Marco Volpe

Labeled Natural Deduction for Temporal Logics

Ph.D. Thesis

Università degli Studi di Verona
Dipartimento di Informatica

Advisor:
prof. Luca Viganò

Series N°: **TD-10-10**

Università di Verona
Dipartimento di Informatica
Strada le Grazie 15, 37134 Verona
Italy

Abstract Despite the great relevance of temporal logics in many applications of computer science, their theoretical analysis is far from being concluded. In particular, we still lack a satisfactory proof theory for temporal logics and this is especially true in the case of branching-time logics.

The main contribution of this thesis consists in presenting a modular approach to the definition of labeled (natural) deduction systems for a large class of temporal logics. We start by proposing a system for the minimal Priorean tense logic and show how to modularly enrich it in order to deal with more complex logics, like *LTL*. We also consider the extension to the branching case, focusing on the Ockhamist branching-time logics with a bundled semantics.

A detailed proof-theoretical analysis of the systems is performed. In particular, in the case of discrete-time logics, for which rules modeling an induction principle are required, we define a procedure of normalization inspired to those of systems for Heyting Arithmetic. As a consequence of normalization, we obtain a purely syntactical proof of the consistency of the systems.

Acknowledgements

First of all, I wish to thank my supervisor, Luca Viganò, for the advice and the constant encouragement during the last three years and three months.

This thesis owes really much to many inspiring discussions with Andrea Masini, to his ideas and to his patience.

I also benefited from spending two pleasant and stimulating months at IST in Lisbon, for which I thank Carlos Caleiro and all the Section of Logic and Computation.

Thanks to the reviewers (Carlos Caleiro and Stéphane Demri) and to the other members of the jury of my Ph.D. defense (Andrea Masini, Angelo Montanari and Luca Viganò) for suggesting several improvements to a preliminary version of this document.

As usual, I thank my family and my friends for everything else.

Contents

1

Introduction

1.1 Background and motivation

The history of the philosophical and logical reasoning about time goes back at least to ancient Greece, with the works of Aristotle and Diodorus Cronus. However, the birth of modern (symbolic) temporal logic is mainly connected to the name of Prior, who in the late 1950's developed the so-called *tense logics* on the model of modal logics, in a work significantly titled "Time and Modality" [127].

Since the seminal work of Pnueli in 1977 [124], temporal logic has also gained a great importance in computer science: applications include its use as a tool for the specification and verification of programs and protocols [18], in the study and development of temporal databases [39], as a framework within which to define the semantics of temporal expressions in natural language [90] and as a language for encoding temporal knowledge in artificial intelligence [72].

Many temporal logics have been proposed, varying both in the set of the operators used and in the semantics adopted (see [88] for a survey). Despite the fact that temporal logics have been studied for many years, their theoretical analysis is far from being concluded. In particular, a satisfactory proof-theoretical analysis for temporal logics is still lacking. This is especially true in the case of branching-time logics, as shown by the fact that for one of the most important of such logics, CTL^*, even the problem of finding a complete Hilbert-style axiomatization has been, partially, solved only recently [135]. Furthermore, when deduction systems have been devised in a form that allows for a meta-theoretical and proof-theoretical analysis (e.g., natural deduction, sequent systems), they have been given for specific logics and do not seem to be easily generalizable to a modular treatment of a wide range of logics of time.

The aim of this thesis is to provide a modular approach to the definition of deduction systems for a large class of temporal logics and to their proof-theoretical analysis. We will mainly deal with natural deduction systems [73, 125]. Such systems present an elegant meta-theory in which derivations can be treated as mathematical objects interesting in themselves. It follows that a "good" natural deduction presentation can be seen also as a useful device for understanding a logic better and for reasoning on its properties. Namely, we believe that a good formula-

tion, in a natural deduction setting, of a logic should at least satisfy the following requirements:

(i) for each connective, there is exactly one introduction and one elimination rule[1], which also express, as well illustrated by Prawitz [125], the "meaning" of the connective;
(ii) a normalization theorem holds and, moreover, the structure of normal proofs is informative enough to let one derive important meta-theorems, such as the subformula property or consistency.

There are a number of different reasons for the delay in the development of temporal proof theory, but perhaps the most important one is that temporal logics are (multi) modal logics and modal proof theory is notoriously a difficult subject. For instance, adapting natural deduction systems for classical (or intuitionistic) logic to modal logic is not straightforward and, in fact, it is not trivial to define systems that enjoy properties (i) and (ii) mentioned above.

Fortunately, in the last decades some interesting proposals for modal proof theory have been presented, e.g. [5, 7, 8, 16, 26, 27, 61, 66, 81, 104, 143, 148, 159, 162]. Among these, particularly interesting are the proposals that are based on *labeled deduction* [26,27,66,143,148,159], a framework that has been successfully employed for several non-classical, and in particular modal, logics, since labeling provides a clean and effective way of dealing with modalities and gives rise to deduction systems with good proof-theoretical properties. The basic idea is that labels allow one to explicitly encode additional information, of a semantical or proof-theoretical nature, that is otherwise implicit in the logic one wants to capture. So, for instance, instead of a formula A, one can consider the *labeled formula* $b : A$, which intuitively means that A holds at the world denoted by b within the underlying Kripke semantics. We can also use labels to specify how worlds are related, e.g. the *relational formula* bRc states that the world c is accessible from b.

Such an enrichment of the language allows for defining introduction and elimination rules for modal operators that are extremely clean and follow the "spirit" of natural deduction. For instance, we can express $b : \Box A$ as the metalevel implication $bRb' \Longrightarrow b' : A$ for an arbitrary b' accessible from b to give the rules:

$$\dfrac{\begin{array}{c}[bRb'] \\ \vdots \\ b' : A\end{array}}{b : \Box A}\ \Box I \qquad\qquad \dfrac{b : \Box A \quad bRb'}{b' : A}\ \Box E$$

where the rule $\Box I$ has the side condition that b' is different from b and does not occur in any assumption on which $b' : A$ depends other than bRb'.

Since it is possible to think of a temporal logic (at least the ones we consider in this thesis) as a modal logic, we propose to use the framework of labeled deduction to develop a proof theory for temporal logics. In fact, by following the Priorean approach, mentioned at the beginning, we can see a temporal logic as a modal logic where the worlds in the semantics are time instants and the accessibility relation is

[1] Up to a few standard exceptions, like, e.g., two symmetrical elimination rules for conjunction.

the ordering $<$ between such time instants. In this view, the modalities of *necessity* $\Box$ and *possibility* $\Diamond$ assume the intended meanings of *always* (usually denoted G) and *eventually* (usually denoted F), respectively. An extension considering past operators is also possible.

1.2 Contributions

Table 1.1 presents a, clearly not comprehensive, map of temporal logics, which will help clarify the main contributions of this thesis. The first column presents logics whose underlying flow of time is linear, while in the second and third column we have branching logics, i.e., the flow of time is assumed to have a tree-like structure and the language is extended with an operator $\forall$ that allows for quantifying on the branches. A further classification can be made when reading the table by rows: the first row presents logics where the flow of time is an arbitrary time-line or an arbitrary tree (*general time*); in the second row, we consider *discrete time* logics, and thus also enrich the language with a next-time operator; in the third row, we are still in a discrete-time setting and further extend the language with the operator *until* [96].

With regard to branching logics, we remark that we focus here on the so-called *Ockhamist* ones, whose language allows for a free combination of temporal operators and quantifiers, and distinguish between two forms of semantics: in the third column, we find the standard (*full*) semantics of the well-known CTL^* [55] (and of its general-time corresponding $OBTL$ [136]); in the second column, we have logics originated by using a generalized (*bundled*) semantics obtained by allowing restrictions on the set of branches considered.

In the literature, labeled natural deduction systems have been proposed for linear-time logics [19, 103] and the branching logic CTL [20, 131], which, given its syntactic restrictions on the nesting of operators, is not Ockhamist and thus is not reported in Table 1.1. In this thesis, we propose a modular approach, based on labeling, to natural deduction for (linear and Ockhamist branching) temporal logics and focus on a proof-theoretical analysis of the defined systems. The main difficulties in such a work can be summarized in the following points:

(1) extending the approach from the linear to the branching case, i.e., moving from the first to the second column of Table 1.1;
(2) treating in a proof-theoretically satisfactory way the operator *until*, i.e., moving from the second to the third row[2];
(3) capturing the full semantics of branching logics (by means of a system with finitary rules), i.e., moving from the second to the third column;
(4) defining a normalization procedure in the case of systems for discrete-time logics, which require a rule modeling the induction principle.

In this thesis, we mainly face and solve points (1) and (4) and give a proposal for point (2), thus covering the first two columns of Table 1.1. The very complex

[2] In this thesis, we consider the use of until explicitly only in the case of discrete logics, but indeed the recipe we propose for dealing with such an operator can be easily adapted to the case of general-time logics.

	Linear-Time	**Bundled Ockhamist Branching-Time**	**Full Ockhamist Branching-Time**
General time	Kl	$BOBTL$	$OBTL$
Until-free discrete time	$LTL\_$	$BCTL^*\_$	$CTL^*\_$
Discrete time with until	LTL	$BCTL^*$	CTL^*

Table 1.1. A map of temporal logics.

problem of item (3) (we remind that even finding a finitary Hilbert-style axiomatization for such logics is still a partially open problem) is left for future work. We further analyze these points below.

1.2.1 Labeled natural deduction for linear temporal logics

We have already seen that, at least in the case of the Priorean tense logics, temporal operators are nothing more than modal operators with respect to a Kripke semantics where the worlds are time instants and the accessibility relation is the ordering $<$ between the time instants. It follows that we may apply the same pattern of introduction/elimination rules seen above in the modal case (just replace $\Box$ with G and R with $<$):

$$\frac{\begin{array}{c}[b < b'] \\ \vdots \\ b' : A\end{array}}{b : \mathsf{G}A}\ \mathsf{G}I \qquad \frac{b : \mathsf{G}A \quad b < b'}{b' : A}\ \mathsf{G}E$$

with the usual condition of freshness for b' in $\mathsf{G}I$.

Relational properties specifying a particular flow of time can also be expressed by means of rules that manage relational formulas, along the same line of relational rules of labeled natural deduction systems for modal logics[3] [148,159]. For instance, we can force the flow of time to be transitive by endowing the system with a rule like:

$$\frac{b_1 < b_2 \quad b_2 < b_3 \qquad \begin{array}{c}[b_1 < b_3] \\ \vdots \\ b : A\end{array}}{b : A}\ trans <$$

Some labeled natural deduction systems for linear temporal logics have been proposed [19, 103] by following the ideas sketched above. Our contribution with

[3] Though, as we will see, some of such properties, e.g., expressing a temporal induction principle in the case of discrete time, require a much more complex treatment than that for most common modal logics.

regard to these logics consists mainly in giving a uniform and modular presentation of systems for a large class of linear temporal logics and in performing a proof-theoretical analysis of such systems. Namely, we give a system for the general linear tense logic Kl, consider some of its variants, e.g., Kl with dense time, with first/final point, unbounded, etc., and finally treat the case of the discrete-time logic $LTL_{\_}$. With regard to the last logic, it is easy to observe that the operator X of next-time can be treated exactly in the same way as the operator G, since it can be seen as a $\Box$-like modal operator with respect to the functional relation of being the *immediate predecessor*.

1.2.2 Labeled natural deduction for branching temporal logics

When we are interested in reasoning about concurrent or non-deterministic processes, it can be convenient to refer to richer semantical structures and more expressive languages than those of linear-time logics. Namely, we can consider tree-like structures and exploit the possibility of quantifying over sets of branches of such trees, where a single branch represents a possible computation. In this thesis, we will mainly deal with the so-called *bundled* branching-time logics, which are obtained by considering a generalization of the standard tree-based semantics. The semantics is defined on the larger class of *bundled trees*, where a bundled tree is represented by a (standard) tree and a set of branches, satisfying some closure properties, on it.[4]

Bundled versions of branching logics have been often considered in the literature [31, 139, 150, 167] and, though less popular than the corresponding "full" logics, are relevant both from a philosophical point of view [116, 118] and in the case of applications to computer science, e.g., when we are interested in restricting the set of computations to be taken into consideration; namely, in the case of reasoning under fairness assumptions. In fact, it has been shown in [42] that $BCTL^*$ is equivalent to the logic generated by fair structures, i.e. transition systems endowed with a mechanism for expressing conditions of *generalized fairness* [63].

The extension of the system for linear-time logics to the bundled branching-time logics requires the definition of rules for treating the path quantifier $\forall$. The idea we apply here consists in considering a different, but equivalent, semantical formulation of such logics, given by means of the so-called *Ockhamist frames* [150, 167]. An Ockhamist frame is a Kripke frame with two accessibility relations[5] (say $\prec$ and $\simeq$) obtained from a bundled tree as follows:

- each branch of the tree is a world of the Ockhamist frame;
- $b_1 \prec b_2$ if b_2 is a sub-branch of b_1;
- $b_1 \simeq b_2$ if b_1 and b_2 share the same initial node.

[4] Namely, in the case of $BOBTL$, the set of branches must be closed under sub-branches and super-branches and such that every node of the tree belongs to some branch in the set. In the case of $BCTL^*$, and of its until-free fragment, the bundled semantics is obtained by removing the so-called *limit-closure* condition from the standard semantics of CTL^*. Details in Chapter 2.

[5] In the case of discrete-time logics, we can also consider a relation of *immediate sub-branch* on which the operator X will be defined.

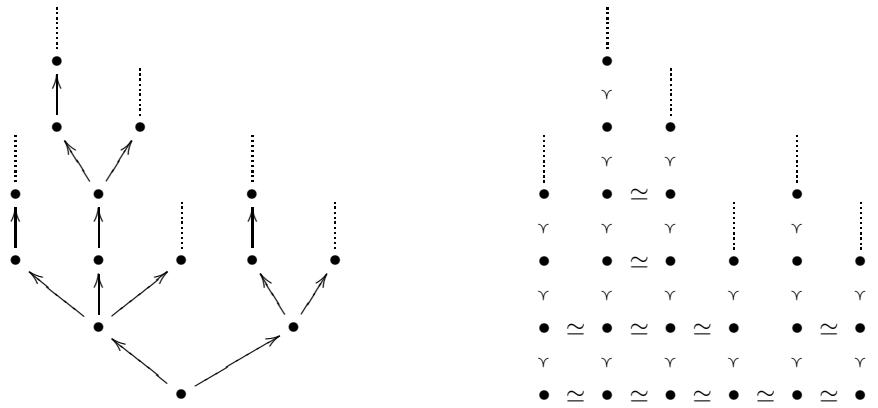


Fig. 1.1. A *bundled tree* (left) and the corresponding *Ockhamist frame* (right).

Figure 1.1 illustrates this correspondence, which, as observed in [167], allows for giving a genuine Kripke-style semantics, where also the path quantifier $\forall$ is seen as a standard ($S5$) modal operator with respect to the equivalence relation $\simeq$.

We have observed above that, when dealing with "pure" modal operators, labeling allows for devising clean and effective introduction and elimination natural deduction rules. And in fact, with this semantics in mind, and by using labels to refer to branches rather than to time instants, we are able to give well-behaved rules for the quantifier $\forall$ as well: just consider the rules for G given above and replace G with $\forall$ and $<$ with $\simeq$.

This leads to a clean and strongly modular deduction system where each basic operator (i.e. G, $\forall$ and, possibly, X) is seen as a modal operator and is endowed with a proper accessibility relation. Interactions between the relations are expressed by means of structural rules that do not involve the operators themselves directly.

A detailed proof-theoretical analysis of the system is also made. Normalization is especially problematic in the case of the logics with both the operators X and G because of the underlying temporal induction principle, which relates the next-time relation and the order relation. Such temporal induction is handled, inside the system, in a way strongly similar to first-order induction of Peano/Heyting Arithmetics and in fact the normalization procedure follows those defined for systems for Heyting Arithmetics in [74, 126, 151]. As is standard in these cases, we present an intuitionistic version of the system and, though the standard subformula property cannot hold, we are able to prove for it *confluence* and *weak normalization*; then we use such results to give a purely syntactical proof of consistency for the intuitionistic system and, via a proper translation, for the classical system as well.

1.2.3 The treatment of until

In the thesis, normalization is studied in the case of systems for until-free logics. In fact, the until U is a quite complex operator, from a proof-theoretical point of view, mainly because of its ambivalent nature of being both "universal" and

"existential"[6]. Indeed, if one is interested in a natural deduction presentation enjoying the properties (*i*) and (*ii*) illustrated in Section 1.1, the solutions given in the literature do not seem to be really satisfactory. Here we give a proposal based on using a slightly more complex labeling discipline than the usual one, so that a formula can be also labeled by a pair of labels, and on introducing a new temporal operator *history* ∇, which allows for a bounded universal quantification between two points. So, for instance, we are allowed to write $bc : \nabla A$ to say that A holds in all the points contained between the instants denoted by b and c. Rules for the new operator can be given in a very clean way, which mirrors the one of the other temporal operators, and until can be clearly expressed in terms of the new operator by exploiting the following equivalence:

$$A \mathsf{U} B \equiv B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A)^{7}\,.$$

In the thesis, we give a system for a variant of *LTL*, obtained by replacing until with history, and prove that such a variant is as expressive as standard *LTL*. We remark, however, that our solution is fully general and can be easily adapted to the case of other (possibly branching) logics with until.

1.2.4 Mosaics for temporal logics

In this thesis we also consider an "orthogonal" model-theoretical topic: the use of the mosaic method in temporal logic [105]. Although the subject is rather different, our contribution, which consists in an extension of the method from the linear to the bundled branching case and is based on the same intuition related to the Ockhamist frames, is in a way similar.

The mosaic method has been introduced in algebraic logic as a way of proving the decidability of the theories of some classes of algebras of relations [114, 115]. The basic idea consists in showing that the existence of a model is equivalent to the existence of a (finite) set of fragments of models (called *mosaics*), satisfying a given number of requirements. From that, we get a decision procedure for the logic, which consists in checking whether such a (finite) set exists or not. The mosaic method has been recently applied [105, 134, 137, 140] to prove decidability, complexity results and completeness of Hilbert-style axiomatizations for several linear temporal logics, namely *Kl* and some of its variants.

Here we propose an extension of the method to the case of bundled branching-time logics, i.e., we move from *Kl* (for which the mosaic method is defined in [105]) to *BOBTL*, and in doing so we also consider a number of intermediate logics. The results concerning decidability and completeness of these logics are already well

[6] In *LTL*, the formula $A \mathsf{U} B$ holds at the current time instant b iff either B holds at b or there *exists* a time instant b' in the future at which B holds and such that A holds in *all* the time instants between the current one and b'. The words in emphasis highlight the dual existential and universal nature of U.

[7] That is: $A \mathsf{U} B$ iff either B holds or there exists a time instant b' in the future (as expressed by the *sometime in the future* operator F) such that (*i*) B holds in the successor time instant, and (*ii*) A holds in all the time instants between the current one and b' (included). The latter conjunct is precisely what the *history* operator ∇ expresses.

known [31], however we believe that the mosaic method is interesting in itself as it provides a uniform way of establishing such results for many logics, by simple and modular modifications of the basic definitions. Moreover, our proposal for this class of branching-time logics can be seen as a basis for dealing with other more interesting logics, for which decidability and complexity results are still missing.

1.3 Synopsis

Part I - Background
- In Chapter 2, we give a brief presentation of modal and temporal logics, focusing on those considered in the thesis.
- In Chapter 3, we introduce labeled natural deduction and describe its use in the case of most common modal logics.

Part II - Labeled Natural Deduction for Temporal Logics
- In Chapter 4, we present and analyze labeled natural deduction systems for linear temporal logics; a proposal for the treatment of until is also given.
- In Chapter 5, we describe labeled natural deduction for a number of bundled branching-time logics, and study normalization, in particular, of the system for $BCTL^*_-$.

Part III - Mosaics for Temporal Logics
- In Chapter 6, we introduce the technique of mosaics in temporal logics and describe an extension to the case of bundled branching Ockhamist logics.

Finally, in Chapter 7, we summarize the contents of the thesis and discuss some possible directions for future work.

In order to ease readability, some of the proofs of Chapter 5 are given in an appendix.

1.4 Publications

Some of the material of this thesis has been published or submitted for publication.

Chapter 4

[160] Luca Viganò and Marco Volpe. *Labeled Natural Deduction Systems for a Family of Tense Logics*. In Stéphane Demri and Christian S. Jensen, editors, Proceedings of the 16th International Symposium on Temporal Representation and Reasoning (TIME-2008), pages 118-126. IEEE Computer Society, 2008.

[110] Andrea Masini, Luca Viganò and Marco Volpe. *A History of Until*. In Thomas Bolander and Torben Braüner, editors, Proceedings of the 6th workshop on Methods for Modalities (M4M-6), volume 262 of *Electronic Notes in Theoretical Computer Science*, pages 189-204, 2010.

Chapter 5

[109] Andrea Masini, Luca Viganò and Marco Volpe. *A Labeled Natural Deduction System for a Fragment of CTL**. In Sergei N. Artëmov and Anil Nerode, editors, Proceedings of the 2009 Symposium on Logical Foundations of Computer Science (LFCS '09), volume 5407 of *Lecture Notes in Computer Science*, pages 338-353. Springer, 2009.

[108] Andrea Masini, Luca Viganò and Marco Volpe. *Labeled Natural Deduction for a Bundled Branching Temporal Logic*. Journal of Logic and Computation (Submitted).

Part I

Background

2

Modal and Temporal Logics

2.1 Introduction

In this chapter, we present the basic notions related to the logics that will be considered in the thesis. We will start introducing the most basic modal logics and then, by enriching the language and by refining the semantical structures considered, we will move to describe a number of linear-time and branching-time temporal logics. For most of the logics, we will also present Hilbert-style axiomatizations, which will turn out to be useful, in the rest of the thesis, in order to prove meta-theoretical properties (typically, completeness) of the natural deduction systems defined.

We remark that in this chapter (as in the rest of the thesis) we restrict to consider only *propositional* modal and temporal logics.

The structure of the chapter is the following:

- in Section 2.2, we introduce the minimal normal modal logic K and some of its most common extensions;
- in Section 2.3, we present linear-time temporal logics;
- in Section 2.4, we describe branching-time temporal logics, focusing on the so-called Ockhamist ones.

2.2 Modal Logics

While classical logic has been devised for dealing with the basic notions of true and false, modal logics allow for *qualifying* the truth of a judgment. This is obtained by using *modal operators*, commonly denoted by $\Box$ and $\Diamond$, with the intended meaning of "necessarily" and "possibly", respectively. There are other possible readings for such modal operators, each of which giving rise to a particular class of modal logics. Some common interpretations are collected in Table 2.1. Modal logics also have important applications in computer science. For an introduction, see [16, 38, 62].

Modal logic	**Interpretation for $\Box A$**
Alethic	A is *necessarily* true
Epistemic	A is *known*
Deontic	it is *obligatory* that A
Temporal	it will *always* be the case that A

Table 2.1. Interpretation of modal operators in most common modal logics.

2.2.1 The minimal normal modal logic *K*

First we introduce syntax and semantics of the minimal normal modal logic K. As we will show in Section 2.2.2, several extensions of K can be obtained by considering the same language but a different semantical characterization.

Syntax

The language of propositional modal logic K consists of a functionally complete set of *classical connectives* (here we will use falsum, denoted by $\bot$, and implication, denoted by $\supset$), a *modal operator* $\Box$ and a denumerable set of *propositional symbols* (or propositional symbols).

Definition 2.1. *Given a set $\mathcal{P}$ of propositional symbols, the set of* (well-formed) modal formulas *is defined by the grammar*

$$A ::= p \mid \bot \mid A \supset A \mid \Box A\,,$$

where $p \in \mathcal{P}$. The set of atomic formulas *is $\mathcal{P} \cup \{\bot\}$. The* complexity *of a formula is the number of occurrences of connectives ($\supset$) and operators ($\Box$).*

The given syntax uses a restricted set of classical connectives and modal operators. As is standard, we can introduce abbreviations and use, e.g., $\neg$, $\wedge$ and $\vee$ for the negation, the conjunction and the disjunction, respectively. For instance, $\neg A \equiv A \supset \bot$. We can also define the dual modal operator of $\Box$, denoted by $\Diamond$, i.e. $\Diamond A \equiv \neg\Box\neg A$.

Semantics

Since the early sixties, semantics for modal logics has been given by means of relational (Kripke) structures, i.e. structures consisting of a set of elements (usually called *worlds*, or points) on which a binary *accessibility relation* is defined.[1] We also associate each relational structure with a *valuation function*, which assigns to every world the set of propositional symbols that are true in it. The truth at every world is defined locally by using the laws of classical logic, while truth for $\Box A$ in a given world w is defined by considering that $\Box A$ is true in w if A is true in every world accessible from w.

[1] As a generalization, we obtain *multi-modal* logics by considering structures with more than one relation (and a distinct modal operator for each relation) and more complex modal logics, e.g. relevance logics, by allowing relations that are not necessarily binary.

Definition 2.2. *A* Kripke frame *is a pair* $\mathcal{F} = (\mathcal{W}, \mathcal{R})$ *where:*

- $\mathcal{W}$ *is a non empty set of* worlds *(or points);*
- $\mathcal{R}$ *is a binary relation on* $\mathcal{W}$*, called* accessibility relation.

Given a set $\mathcal{P}$ *of propositional symbols, a* Kripke structure *(or Kripke model) on* $\mathcal{P}$ *is a triple* $\mathcal{M} = (\mathcal{W}, \mathcal{R}, \mathcal{V})$ *where:*

- $(\mathcal{W}, \mathcal{R})$ *is a Kripke frame;*
- $\mathcal{V} : \mathcal{W} \to 2^{\mathcal{P}}$ *is a (*valuation*) function that assigns to each world in* $\mathcal{W}$ *a (possibly empty) set of propositional symbols.*

Definition 2.3. Truth in the logic K *for a modal formula at a point* w *in a Kripke structure* $\mathcal{M} = (\mathcal{W}, \mathcal{R}, \mathcal{V})$ *is the smallest relation* $\models_K$ *satisfying:*

$$\begin{array}{rll} \mathcal{M}, w \models_K p & \textit{iff} & p \in \mathcal{V}(w) \\ \mathcal{M}, w \models_K A \supset B & \textit{iff} & \mathcal{M}, w \models_K A \textit{ implies } \mathcal{M}, w \models_K B \\ \mathcal{M}, w \models_K \Box A & \textit{iff} & \mathcal{M}, w' \models_K A \textit{ for all } w' \textit{ s.t. } w\mathcal{R}w' \end{array}$$

Note that $\mathcal{M}, w \not\models \bot$ *for every* $\mathcal{M}$ *and* w*. By extension, given a modal formula* A *and a set of modal formulas* Γ*, we write:*

$$\begin{array}{rll} \mathcal{M} \models_K A & \textit{iff} & \mathcal{M}, w \models_K A \textit{ for all } w \in \mathcal{W} \\ \mathcal{M} \models_K \Gamma & \textit{iff} & \mathcal{M} \models_K A \textit{ for all } A \in \Gamma \\ \Gamma \models_K A & \textit{iff} & \mathcal{M} \models_K \Gamma \textit{ implies } \mathcal{M} \models_K A\textit{, for every Kripke structure } \mathcal{M} \\ \models_K A & \textit{iff} & \mathcal{M} \models_K A \textit{ for every Kripke structure } \mathcal{M}. \end{array}$$

We say that:

- *a modal formula* A *is* K-satisfiable in a Kripke structure $\mathcal{M}$ *iff there exists a world* w *in* $\mathcal{M}$ *such that* $\mathcal{M}, w \models_K A$*;*
- *a modal formula* A *is* K-satisfiable *iff* A *is satisfiable in some Kripke structure* $\mathcal{M}$*; otherwise it is* K-unsatisfiable*;*
- *a modal formula* A *is* K-valid in a Kripke structure $\mathcal{M}$ *iff* $\mathcal{M} \models_K A$*;*
- *a modal formula* A *is* K-valid in a Kripke frame $\mathcal{F}$ *iff* $\mathcal{M} \models_K A$ *for every model* $\mathcal{M}$ *defined on the frame* $\mathcal{F}$*;*
- *a modal formula* A *is* K-valid *iff* $\models_K A$*, i.e.* A *is valid in every Kripke structure.*

We can now define the logic K as the set of formulas that are valid according to the semantics given above, i.e. $K = \{A \mid \models_K A\}$.

A Hilbert-style axiomatization

For the minimal modal logic K, we can give the following Hilbert-style axiomatization $\mathcal{H}(K)$:

$$\begin{array}{ll} (CL) & \text{Any tautology instance of classical propositional logic} \\ (K) & \Box(A \supset B) \supset (\Box A \supset \Box B) \end{array}$$

We have also the inference rules of *modus ponens* and *modal necessitation* (or generalization):

$$(MP)\ \text{If}\ A\ \text{and}\ A \supset B\ \text{then}\ B$$
$$(Nec)\ \text{If}\ A\ \text{then}\ \Box A$$

The set of *theorems of* $\mathcal{H}(K)$ is defined as the smallest set of modal formulas containing the set of axioms and closed with respect to the rules of inference above. We denote with $\vdash_K$ the notion of derivability in $\mathcal{H}(K)$, i.e. $\vdash_K A$ iff A is a theorem of $\mathcal{H}(K)$. Furthermore we write $\Gamma \vdash_K A$ (A *follows deductively from* Γ) if A can be derived from all theorems of $\mathcal{H}(K)$ and the formulas in Γ by applying the rule (MP) only.[2]

We can now state a relation between the notions of logical consequence, i.e. $\Gamma \models_K A$, and deductive consequence, i.e. $\Gamma \vdash_K A$,. In fact, by a Henkin-style construction (see, e.g., [89]), it is possible to show the following result of soundness (right-to-left direction) and completeness (left-to-right direction) for the given axiomatization.

Theorem 2.4 (Soundness and completeness). *Given a modal formula A and a set of modal formulas Γ, it holds:*

$$\Gamma \models_K A \quad \Leftrightarrow \quad \Gamma \vdash_K A\,.$$

2.2.2 Axiomatic extensions

Several further modal logics (we call them *frame logics*) can be defined as extensions of the logic K by simply restricting the class of frames we consider. Classes of frames can be distinguished by means of the properties (e.g., reflexivity, transitivity, etc.) of their accessibility relation. Many of the restrictions we are interested in are definable as formulas of first-order logic where the binary predicate $\mathcal{R}(x, y)$ refers to the corresponding accessibility relation.[3] Table 2.2, adapted from [81], summarizes some of the most common frame logics, describing the corresponding frame property. The semantics of a given logic KP can be inferred from the one for K of Definition 2.3: we just consider Kripke models whose accessibility relation satisfies the property P instead of generic Kripke models. This idea can be further generalized by defining a logic $KP_1 \ldots P_n$ as the logic of frames satisfying the set of properties $\{P_1, \ldots, P_n\}$.

At the heart of *correspondence theory* (see [144, 154] for details) lays the fact that particular axioms correspond to particular restrictions on the accessibility relation, i.e. suppose $(\mathcal{W}, \mathcal{R})$ is a frame, then a certain axiom P will be valid on all the models based on $(\mathcal{W}, \mathcal{R})$ if and only if the accessibility relation $\mathcal{R}$ meets a certain condition P (for simplicity, we give the same name to properties of the accessibility relation and axioms).

[2] We remark that, due to the rule of necessitation, the deduction theorem ($\Gamma \vdash_K A \supset B$ iff $\Gamma \cup \{A\} \vdash B$) fails if we adopt the same notion of derivability as in classical Hilbert system formulations (see, e.g., [62] for details).

[3] Note that, for simplicity, we use here the same symbol for denoting both the accessibility relation and the predicate.

Axiom	Condition	First-Order Formula
T	Reflexive	$\forall w : \mathcal{R}(w, w)$
D	Serial	$\forall w \exists w' : \mathcal{R}(w, w')$
4	Transitive	$\forall s, t, u : (\mathcal{R}(s, t) \wedge \mathcal{R}(t, u)) \Rightarrow \mathcal{R}(s, u)$
5	Euclidean	$\forall s, t, u : (\mathcal{R}(s, t) \wedge \mathcal{R}(s, u)) \Rightarrow \mathcal{R}(t, u)$
B	Symmetric	$\forall w, w' : \mathcal{R}(w, w') \Rightarrow \mathcal{R}(w', w)$
2	Weakly-Directed	$\forall s, t, u \exists v : (\mathcal{R}(s, t) \wedge \mathcal{R}(s, u)) \Rightarrow (\mathcal{R}(t, v) \wedge \mathcal{R}(u, v))$
L	Weakly-Connected	$\forall s, t, u : (\mathcal{R}(s, t) \wedge \mathcal{R}(s, u)) \Rightarrow (\mathcal{R}(t, u) \vee t = u \vee \mathcal{R}(u, t))$
X	Dense	$\forall u, v \exists w : (\mathcal{R}(u, v) \Rightarrow (\mathcal{R}(u, w) \wedge \mathcal{R}(w, v)$

Table 2.2. Axioms and corresponding first-order conditions on $\mathcal{R}$.

It is obviously possible to extend the notions of K-satisfiability and K-validity to the case of a logic $KP_1 \dots P_n = \{A \mid \models_{KP_1 \dots P_n} A\}$. The same analogy holds also in considering axiomatic deduction systems: for each property described in Table 2.2, we give a corresponding defining axiom in Table 2.3. Let P be one of such axioms; then, by adding the axiom P to the axiomatization $\mathcal{H}(K)$ we get an axiomatization $\mathcal{H}(KP)$ that is sound and complete for the logic KP.

Traditionally, some of these axiomatic extensions of K have been denoted in the literature with specific names. In particular, the following equivalences hold: $S4 = KT4$, $S5 = KT4B$. In other words, $S4$ denotes the logic of reflexive and transitive frames, while $S5$ denotes the logic of frames whose accessibility relation is an equivalence relation.

Axiom	Defining Formula
K	$\Box(A \supset B) \supset (\Box A \supset \Box B)$
T	$\Box A \supset A$
D	$\Box A \supset \Diamond A$
4	$\Box A \supset \Box\Box A$
5	$\Box A \supset \Box\Diamond A$
B	$A \supset \Box\Diamond A$
2	$\Diamond\Box A \supset \Box\Diamond A$
L	$\Box((A \wedge \Box A) \supset B) \vee \Box((B \wedge \Box B) \supset A)$
X	$\Box\Box A \supset \Box A$

Table 2.3. Modal logics and corresponding defining formulas.

2.3 Linear Temporal Logics

Temporal logics can be seen as a branch of modal logic, where the accessibility relation is used to model the flow of time and each world in a structure corresponds to a time instant. In this section we focus on linear temporal logics, i.e. those whose underlying semantical structures represent flows of time with the shape of a line.

First, we will present some basic tense logic whose definition is due to Prior [128] (see also [34, 68]). Then we will present more interesting logics from a computational point of view, i.e *LTL* [124] and fragments of *LTL*.

2.3.1 The basic tense logic *Kt*

As for modal logics, we begin by fixing a temporal language that will be used first for introducing a basic tense logic, called Kt, and then for considering axiomatic extensions of it, in the vein of the extensions presented in Section 2.2.2.

Syntax

The language of propositional priorean tense logic consists of a functionally complete set of *classical connectives*, two *modal operators* (G and P) and a denumerable set of *propositional symbols*.

Definition 2.5. *Given a set $\mathcal{P}$ of propositional symbols, the set of* (well-formed) tense formulas *is defined by the grammar*

$$A ::= p \mid \bot \mid A \supset A \mid \mathsf{G}A \mid \mathsf{H}A,$$

where $p \in \mathcal{P}$. The set of atomic formulas *is $\mathcal{P} \cup \{\bot\}$. The* complexity *of a formula is the number of occurrences of connectives ($\supset$) and operators (G and H).*

G and H are "universal" modal operators, whose intuitive meaning is *always in the future* and *always in the past*, respectively. Their duals F and P (*eventually in the future* and *sometime in the past*, respectively) can be defined as $\mathsf{F}A \equiv \neg\mathsf{G}\neg A$ and $\mathsf{P}A \equiv \neg\mathsf{H}\neg A$. Other classical connectives can also be defined as usual.

Semantics

Temporal frames and structures are simple adaptations of the standard Kripke ones (Section 2.2.1). Since we are interested in representing a flow of time, from now on we will use the symbol $\prec$ (recalling the idea of an order relation) to denote the accessibility relation $\mathcal{R}$ and the term *instant* instead of world. For the moment we do not make any particular assumption about the nature of the relation $\prec$.[4]

Truth for a tense formula is then defined by letting G behave as the operator $\Box$ and H as its analogous with respect to the symmetric relation $\prec^{-1}$.

Definition 2.6. *A* temporal frame *is a pair $\mathcal{F} = (\mathcal{W}, \prec)$ where:*

- *$\mathcal{W}$ is a non empty set of (time) instants;*
- *$\prec$ is a binary relation on $\mathcal{W}$.*

Given a set $\mathcal{P}$ of propositional symbols, a temporal structure *(model) on $\mathcal{P}$ is a triple $\mathcal{M} = (\mathcal{W}, \prec, \mathcal{V})$ where:*

[4] For convenience, we present Kt in the section devoted to *linear* temporal logics, but indeed there is no assumption of *linearity* in the semantical structures of Kt.

- $(\mathcal{W}, \prec)$ *is a temporal frame;*
- $\mathcal{V} : \mathcal{W} \to 2^{\mathcal{P}}$ *is a (*valuation*) function that assigns to each instant in* $\mathcal{W}$ *a (possibly empty) set of propositional symbols.*

Definition 2.7. Truth in the logic Kt *for a tense formula at an instant* w *in a temporal structure* $\mathcal{M} = (\mathcal{W}, \prec, \mathcal{V})$ *is the smallest relation* $\models_{Kt}$ *satisfying:*

$$\begin{array}{rll} \mathcal{M}, w \models_{Kt} p & \textit{iff} & p \in \mathcal{V}(w) \\ \mathcal{M}, w \models_{Kt} A \supset B & \textit{iff} & \mathcal{M}, w \models_{Kt} A \textit{ implies } \mathcal{M}, w \models_{Kt} B \\ \mathcal{M}, w \models_{Kt} \mathsf{G}A & \textit{iff} & \mathcal{M}, w' \models_{Kt} A \textit{ for all } w' \textit{ s.t. } w \prec w' \\ \mathcal{M}, w \models_{Kt} \mathsf{H}A & \textit{iff} & \mathcal{M}, w' \models_{Kt} A \textit{ for all } w' \textit{ s.t. } w' \prec w \end{array}$$

Note that, as a consequence, we have $\mathcal{M}, w \not\models \bot$ *for every* $\mathcal{M}$ *and* w*. By extension, given a tense formula* A *and a set of tense formulas* Γ*, we write:*

$$\begin{array}{rll} \mathcal{M} \models_{Kt} A & \textit{iff} & \mathcal{M}, w \models_{Kt} A \textit{ for all } w \in \mathcal{W} \\ \mathcal{M} \models_{Kt} \Gamma & \textit{iff} & \mathcal{M} \models_{Kt} A \textit{ for all } A \in \Gamma \\ \Gamma \models_{Kt} A & \textit{iff} & \mathcal{M} \models_{Kt} \Gamma \textit{ implies } \mathcal{M} \models_{Kt} A\textit{, for every linear temporal structure } \mathcal{M} \\ \models_{Kt} A & \textit{iff} & \mathcal{M} \models_{Kt} A \textit{ for every linear temporal structure } \mathcal{M}. \end{array}$$

We say that:

- *a tense formula* A *is* Kt-satisfiable in a temporal structure $\mathcal{M}$ *iff there exists a world* w *in* $\mathcal{M}$ *such that* $\mathcal{M}, w \models_{Kt} A$*;*
- *a tense formula* A *is* Kt-satisfiable *iff* A *is satisfiable in some temporal structure* $\mathcal{M}$*; otherwise it is* Kt-unsatisfiable*;*
- *a tense formula* A *is* Kt-valid in a temporal structure $\mathcal{M}$ *iff* $\mathcal{M} \models_{Kt} A$*;*
- *a tense formula* A *is* Kt-valid in a temporal frame $\mathcal{F}$ *iff* $\mathcal{M} \models_{Kt} A$ *for every model* $\mathcal{M}$ *defined on the frame* $\mathcal{F}$*;*
- *a tense formula* A *is* Kt-valid *iff* $\models_{Kt} A$*, i.e.* A *is valid in every temporal structure.*

As we did for K, we can define the logic Kt as the set of formulas that are Kt-valid according to the semantics given above, i.e. $Kt = \{A \mid \models_{Kt} A\}$.

A Hilbert-style axiomatization

A Hilbert-style axiomatization $\mathcal{H}(Kt)$ for Kt can be easily obtained by adapting the one for K (see, e.g., [75]). An equivalent of the axiom schema K is needed for both the operators G and H, in addition to a couple of axioms stating the relation between the two operators.

$$\begin{array}{ll} (CL) & \text{Any tautology instance of classical propositional logic} \\ (K_{\mathsf{G}}) & \mathsf{G}(A \supset B) \supset (\mathsf{G}A \supset \mathsf{G}B) \\ (K_{\mathsf{H}}) & \mathsf{H}(A \supset B) \supset (\mathsf{H}A \supset \mathsf{H}B) \\ (\mathsf{GP}) & A \supset \mathsf{GP}A \\ (\mathsf{HF}) & A \supset \mathsf{HF}A \end{array}$$

We also need the inference rules of *modus ponens* and *necessitation* (or generalization):

$$(MP) \quad \text{If } A \text{ and } A \supset B \text{ then } B$$
$$(Nec_{\mathsf{G}}) \text{ If } A \text{ then } \mathsf{G}A$$
$$(Nec_{\mathsf{H}}) \text{ If } A \text{ then } \mathsf{H}A$$

As for K, we define the notions of *theorem of* $\mathcal{H}(Kt)$ and *derivability in* $\mathcal{H}(Kt)$ ($\vdash_{Kt}$) and enunciate a theorem of soundness and completeness [75].

Theorem 2.8 (Soundness and completeness). *Given a tense formula A and a set of tense formulas Γ, it holds::*

$$\Gamma \models_{Kt} A \qquad \Leftrightarrow \qquad \Gamma \vdash_{Kt} A\,.$$

2.3.2 Axiomatic extensions

As in Section 2.2.2, we can obtain extensions of the basic logic, in this case Kt, by adding axioms to the given axiomatization $\mathcal{H}(Kt)$. Some of the most interesting axioms (and the corresponding properties) are shown in Table 2.4.

Axiom	Property	Formula
$(REFL_R)$	Right-Reflexivity	$\mathsf{G}A \supset A$
$(REFL_L)$	Left-Reflexivity	$\mathsf{H}A \supset A$
$(TRANS_R)$	Right-Transitivity	$\mathsf{G}A \supset \mathsf{GG}A$
$(TRANS_L)$	Left-Transitivity	$\mathsf{H}A \supset \mathsf{HH}A$
$(CONN_R)$	Right-Linearity	$(\mathsf{H}A \wedge A \wedge \mathsf{G}A) \supset \mathsf{GH}A$
$(CONN_L)$	Left-Linearity	$(\mathsf{H}A \wedge A \wedge \mathsf{G}A) \supset \mathsf{HG}A$
(SER_R)	Right-seriality	$\mathsf{F}\top$
(SER_L)	Left-seriality	$\mathsf{P}\top$
$(FINAL)$	Right-Boundedness	$\mathsf{G}\bot \vee \mathsf{FG}\bot$
$(FIRST)$	Left-Boundedness	$\mathsf{H}\bot \vee \mathsf{PH}\bot$
$(DENS_R)$	Right-Density	$\mathsf{F}A \supset \mathsf{FF}A$
$(DENS_L)$	Left-Density	$\mathsf{P}A \supset \mathsf{PP}A$
$(DISCR_R)$	Right-Discreteness	$(\mathsf{F}\top \wedge A \wedge \mathsf{H}A) \supset \mathsf{FH}A$
$(DISCR_L)$	Left-Discreteness	$(\mathsf{P}\top \wedge A \wedge \mathsf{G}A) \supset \mathsf{PG}A$

Table 2.4. Axioms expressing temporal properties.

Such axioms are obviously not completely independent one of each other. Some combinations give rise to interesting tense logics extending Kt.

In the following, we present explicitly those axiomatic extensions to which we will refer more often in the thesis: the linear tense logic Kl and some of its variants.

The logic *Kl*

The language of the logic Kl is the language of tense formulas defined in Definition 2.5.

Semantics

The semantics is given on a refinement of the temporal structures of Definition 2.7 that takes into account transitivity and linearity (or connectedness) of the flow of time.

Definition 2.9. *A* linear temporal frame *is a pair $\mathcal{F} = (\mathcal{W}, \prec)$, where:*

- *$\mathcal{W}$ is a non-empty set of (time) instants;*
- *$\prec \subseteq \mathcal{W} \times \mathcal{W}$ is a binary relation that satisfies the properties of irreflexivity, transitivity and connectedness, i.e. for all $(w, w') \in \mathcal{W}^2$ we have $w = w'$ or $(w, w') \in \prec$ or $(w', w) \in \prec$.*

Given a set $\mathcal{P}$ of propositional symbols, a linear temporal structure *(model) on $\mathcal{P}$ is a triple $\mathcal{M} = (\mathcal{W}, \prec, \mathcal{V})$ where:*

- *$(\mathcal{W}, \prec)$ is a linear temporal frame;*
- *$\mathcal{V} : \mathcal{W} \to 2^{\mathcal{P}}$ is a (*valuation*) function that assigns to each instant in $\mathcal{W}$ a (possibly empty) set of propositional symbols.*

Truth in the logic *Kl* for a tense formula is defined as in Definition 2.7 where we consider linear temporal structures instead of temporal structures. We also extend the notion of *Kl*-truth to the notions of *Kl*-satisfiability and *Kl*-validity in a standard way and define *Kl* as the set of *Kl*-valid formulas.

A Hilbert-style axiomatization

A Hilbert-style axiomatization $\mathcal{H}(Kl)$ for *Kl* is obtained (see, e.g., [75]) by extending the one for *Kt* of Section 2.3.1 with the following axiom schemata:

$$
\begin{array}{ll}
(TRANS_R) & \mathsf{G}A \supset \mathsf{GG}A \\
(TRANS_L) & \mathsf{H}A \supset \mathsf{HH}A \\
(CONN_R) & \mathsf{H}A \wedge A \wedge \mathsf{G}A \supset \mathsf{GH}A \\
(CONN_L) & \mathsf{H}A \wedge A \wedge \mathsf{G}A \supset \mathsf{HG}A
\end{array}
$$

Axioms $(TRANS_R)$ and $(TRANS_L)$ express the transitivity of $\prec$, while $(CONN_R)$ and $(CONN_L)$ expresses its connectedness.

Kl with unbounded time

We can further restrict the set of linear temporal frames by requiring that they satisfy additional relational properties. For instance, we can express the fact that the sequence of time points is unbounded, towards the future and/or towards the past. This corresponds to adding the conditions of seriality on the right and/or on the left, i.e. every point has a successor and/or a predecessor.

The axioms expressing unboundedness are SER_R and SER_L in Table 2.4, which express, respectively, the following two properties:

- $\forall x \, \exists y \, . \, x \prec y$;
- $\forall x \, \exists y \, . \, y \prec x$.

Kl with a first/final point

The semantics of *Kl* is given by means of temporal structures where nothing is said about the existence of a first or a final point. To express the existence of such points, we add the axioms (*FINAL*) and (*FIRST*) of Table 2.4, which correspond to the properties:

- $\exists x \, \forall y . \, \neg(y \prec x)$;
- $\exists x \, \forall y . \, \neg(x \prec y)$.

Kl with dense time

Another constraint that we can impose on relational structures is that the flow of time is dense, i.e. between any two points we can find a third point:

- $\forall x \forall y \, . \, x \prec y \;\Rightarrow\; \exists z . \, x \prec z$ and $z \prec y$.

This property is represented by the two axioms $DENS_R$ and $DENS_L$.

Kl with discrete time

Finally, we can express discreteness both towards the future:

- for all x, y, if $x \prec y$, then there exists z such that:
  - $x \prec z$; and
  - for all w, $\neg(x \prec w)$ or $\neg(w \prec z)$;

and towards the past:

- for all x, y, if $x \prec y$, then there exists z such that:
  - $z \prec y$; and
  - for all w, $\neg(z \prec w)$ or $\neg(w \prec y)$.

In terms of axiomatization, this corresponds to the addition of the axioms $DISCR_R$ and $DISCR_L$, respectively, to $\mathcal{H}(Kl)$.

2.3.3 Language extensions

Interesting extensions can also be obtained by considering languages enriched with further temporal operators on the semantical structures of Section 2.3.2. In his doctoral dissertation [96], Kamp extended the basic tense language with the binary operator *until* (and its past-oriented version *since*), which has been shown to be very expressive and particularly useful for applications to computer science. In the case of discrete flows of time, it makes also sense to consider an operator of *next-time*. For a description of the more expressive resulting logics, see [68, 75].

Here we will consider both until and next-time in Section 2.3.4, in the specific context of *LTL*, where we will also formalize their semantics.

2.3.4 *LTL*

LTL is probably the most popular linear temporal logic in computer science. It has been proposed in [124] and further developed and studied in [71]. Here we recall the syntax and semantics of *LTL* and give an axiomatization for it.

Syntax

When considering *LTL*, we are used to restrict the attention to the future-oriented operators. The set of basic temporal operators is enriched by the *next-time* (denoted X) and the *until* (denoted U) operators.

Definition 2.10. *Given a set $\mathcal{P}$ of propositional symbols, the set of* (well-formed) *LTL*-formulas *is defined by the grammar*

$$A ::= p \mid \bot \mid A \supset A \mid \mathsf{G}A \mid \mathsf{X}A \mid A\mathsf{U}A$$

where $p \in \mathcal{P}$. The set of LTL-atomic formulas *is $\mathcal{P} \cup \{\bot\}$. The* complexity *of an LTL-formula is the number of occurrences of the connective $\supset$ and of the temporal operators G, X, and U.*

The intuitive meaning of the temporal operators G, X, and U is the standard one:

- $\mathsf{G}A$ states that A holds always in the future;
- $\mathsf{X}A$ states that A holds in the next time instant;
- $A\mathsf{U}B$ states that B holds at the current time instant or there is a time instant w in the future such that B holds in w and A holds in all the time instants between the current one and w.

Semantics

The semantics of *LTL* is defined on structures that are isomorphic to the set of natural numbers. Note that in this case we consider a non-strict order relation $\leq$, as it seems to be more common in the literature when considering *LTL*. So, for example, $\mathsf{G}A$ holds in a time instant w iff A holds in w and in all its successors.

Definition 2.11. *Let $\mathcal{N} = (\mathbb{N}, s : \mathbb{N} \to \mathbb{N}, \leq)$ be the standard structure of natural numbers, where s and $\leq$ are the successor function and the total (reflexive) order relation, respectively. An LTL*-structure *is a pair $\mathcal{M} = (\mathcal{N}, \mathcal{V})$ where $\mathcal{V} : \mathbb{N} \to 2^{\mathcal{P}}$.* Truth *for an LTL-formula at a point $n \in \mathbb{N}$ in an LTL-structure $\mathcal{M} = (\mathcal{N}, \mathcal{V})$ is the smallest relation $\models_{LTL}$ satisfying:*

$$\begin{array}{rll}
\mathcal{M}, n \models_{LTL} p & \textit{iff} & p \in \mathcal{V}(n) \\
\mathcal{M}, n \models_{LTL} A \supset B & \textit{iff} & \mathcal{M}, n \models_{LTL} A \textit{ implies } \mathcal{M}, n \models_{LTL} B \\
\mathcal{M}, n \models_{LTL} \mathsf{G}A & \textit{iff} & \mathcal{M}, m \models_{LTL} A \textit{ for all } m \geq n \\
\mathcal{M}, n \models_{LTL} \mathsf{X}A & \textit{iff} & \mathcal{M}, n+1 \models_{LTL} A \\
\mathcal{M}, n \models_{LTL} A\mathsf{U}B & \textit{iff} & \textit{there exists } n' \geq n \textit{ such that } \mathcal{M}, n' \models_{LTL} B \\
 & & \textit{and } \mathcal{M}, m \models_{LTL} A \textit{ for all } n \leq m < n'
\end{array}$$

Note that $\mathcal{M}, n \not\models_{LTL} \bot$ *for every* $\mathcal{M}$ *and* n*. By extension, we write:*

$$\begin{array}{rll} \mathcal{M} \models_{LTL} A & \textit{iff} & \mathcal{M}, n \models_{LTL} A \textit{ for every natural number } n \\ \mathcal{M} \models_{LTL} \Gamma & \textit{iff} & \mathcal{M} \models_{LTL} A \textit{ for all } A \in \Gamma \\ \Gamma \models_{LTL} A & \textit{iff} & \mathcal{M} \models_{LTL} \Gamma \textit{ implies } \mathcal{M} \models_{LTL} A\textit{, for every LTL-structure } \mathcal{M} \end{array}$$

A Hilbert-style axiomatization

We now present a sound and complete Hilbert-style axiomatization, which we call $\mathcal{H}(LTL)$, for *LTL* (see, e.g., [75]). $\mathcal{H}(LTL)$ consists of the axioms

$$\begin{array}{ll} (A1) & \text{Any tautology instance} \\ (A2) & \mathsf{G}(A \supset B) \supset (\mathsf{G}A \supset \mathsf{G}B) \\ (A3) & (\mathsf{X}\neg A \leftrightarrow \neg \mathsf{X}A) \\ (A4) & \mathsf{X}(A \supset B) \supset (\mathsf{X}A \supset \mathsf{X}B) \\ (A5) & \mathsf{G}A \supset A \wedge \mathsf{XG}A \\ (A6) & \mathsf{G}(A \supset \mathsf{X}A) \supset (A \supset \mathsf{G}A) \\ (A7) & A \mathsf{U} B \leftrightarrow (B \vee (A \wedge \mathsf{X}(A \mathsf{U} B))) \\ (A8) & A \mathsf{U} B \supset \mathsf{F}B \end{array}$$

where we denote with $\leftrightarrow$ the double implication, and of the rules of inference

$$\begin{array}{ll} (MP) & \text{If } A \text{ and } A \supset B \text{ then } B \\ (Nec_X) & \text{If } A \text{ then } \mathsf{X}A \\ (Nec_G) & \text{If } A \text{ then } \mathsf{G}A \end{array}$$

The set of theorems of $\mathcal{H}(LTL)$ is the smallest set containing these axioms and closed with respect to these rules of inference. The notion of derivability in $\mathcal{H}(LTL)$ will be denoted with $\vdash_{LTL}$ and the deductive consequence $\Gamma \vdash_{LTL} A$ is defined as usual.

With regard to $\mathcal{H}(LTL)$, we need to notice that it is possible to express only a result of *weak* completeness, i.e. a result in terms of single valid formulas, or in terms of a consequence relation $\Gamma \models_{LTL} A$ where Γ is a finite set. As $\mathcal{H}(LTL)$ consists of only finitary rules, it cannot be strongly complete and indeed all the finitary deduction systems for temporal logics equipped with at least the operators X and G (and thus not compact) present such a problem; see, e.g., [100, Chapter 6]. In fact, it is easy to check that $\{\mathsf{X}^i A\}_{i<\omega} \models_{LTL} \mathsf{G}A$ but (via soundness) we can see that $\{\mathsf{X}^i A\}_{i<\omega} \not\vdash_{LTL} \mathsf{G}A$, where $\mathsf{X}^0 A$ is just A and $\mathsf{X}^{i+1} A$ stands for $\mathsf{X}\mathsf{X}^i A$. We will return to this point in Chapter 4 when discussing completeness of a natural deduction system for (a fragment of) *LTL*.

Theorem 2.12 (Soundness and completeness). *Let* A *be an LTL-formula and* Γ *a set of LTL-formulas. Then it holds:*

$$\begin{array}{rcl} \Gamma \vdash_{LTL} A & \Rightarrow & \Gamma \models_{LTL} A\,, \\ \models_{LTL} A & \Rightarrow & \vdash_{LTL} A\,. \end{array}$$

Until-free *LTL*: *LTL*$_-$

Since we will consider it in the thesis, we also define here a fragment of LTL named LTL_-. It corresponds to the until-free fragment of LTL.

The syntax is given by the following definition.

Definition 2.13. *Given a set $\mathcal{P}$ of propositional symbols, the set of* (well-formed) LTL_--formulas *is defined by the grammar*

$$A ::= p \mid \bot \mid A \supset A \mid \mathsf{G}A \mid \mathsf{X}A$$

where $p \in \mathcal{P}$.

The semantics is given on LTL-structures and can be inferred from that of LTL, i.e., given an LTL_--formula A and an LTL-structure $\mathcal{M}$, we have $\mathcal{M} \models_{LTL_-} A$ iff $\mathcal{M} \models_{LTL} A$. The notions of validity and consequence relation come from it as is standard.

A sound and weakly complete axiomatization $\mathcal{H}(LTL_-)$ for LTL_- (see, e.g., [75]) is obtained by just removing the axioms $(A7)$ and $(A8)$ (concerning the until) from the axiomatization $\mathcal{H}(LTL)$.

2.4 Branching Temporal Logics

The temporal logics presented so far are of interest for reasoning about single computations. When we are interested in reasoning about concurrent or non-deterministic processes, it is convenient to refer to richer semantical structures and more expressive languages. Namely, we will consider tree-like structures and exploit the possibility of quantifying over sets of branches of such trees, where a single branch represents a possible computation.

The philosophical basis of branching-time logics can be found already in the work of Prior [128]. However their development in computer science is due to [2, 13, 40, 55]. A survey for the "philosophical" branching-time logics is in [167]; for a survey more oriented towards computer science, see [52].

Here we will focus on those branching-time logics according to which the past is determined and cannot be changed (from which the term *historical necessity* derives), while the future is non-deterministic and can take different possible courses. However, before defining the most standard logics of historical necessity, we will also present (by following the taxonomy in [167]) several intermediate logics, whose tree-like branching nature is much weaker.

In particular, we will consider here the logics originated from the so-called *Ockhamist* semantics (see [128, 167]). In an Ockhamist view, the *actual* future is in some way determined, that is temporal formulas are evaluated with respect not just to a given instant but to an instant and a branch beginning from such instant.

First we will present a class of logics, to which we will refer as *bundled Ockhamist logics with general time*, that have been mainly object of philosophical study and in which arbitrary trees are allowed as flows of time. Then we will move to the so-called *computation tree logics*, which are more interesting from a computational point of view: these logics consider flows of time that are discrete ω-height

trees. In both cases, particular attention will be concentrated on the definition of a generalized semantics (usually referred to as *bundled*), in addition to the standard one, since such a generalized semantics will be object of study in the rest of the thesis.

2.4.1 Bundled Ockhamist logics with general time

Syntax

The language of the branching logics considered in this section consists of a set of classical connectives enriched by some linear temporal operators (the ones we have already considered in Section 2.3) and by one or more path quantifiers.

Definition 2.14. *Given a set $\mathcal{P}$ of propositional symbols, the set of* (well-formed) Ockhamist formulas *is defined by the grammar*

$$A ::= p \mid \bot \mid A \supset A \mid \mathsf{G}A \mid \mathsf{H}A \mid \forall A\,,$$

where $p \in \mathcal{P}$. The set of atomic formulas *is $\mathcal{P} \cup \{\bot\}$. The* complexity *of a formula is the number of occurrences of connectives ($\supset$), operators (G, H) and path quantifiers ($\forall$).*

The intuitive meaning of the linear operators G and H is as in linear temporal logics with respect to a single branch of the tree. The path quantifier $\forall$ allows one to switch from a branch to another: intuitively, $\forall A$ holds at a node s iff A holds in all the branches starting from the node s.

Semantics

Semantics in terms of trees

As we anticipated, we consider as branching logics the logics whose semantical structure have a tree-like representation.

Definition 2.15. *A* tree *is an irreflexive ordered set $\mathcal{T} = (T, <)$ in which the set of the $<$-predecessors of any element t of T is linearly ordered by $<$, that is, for all x, y, z in T, if $x < z$ and $y < z$ then either $x < y$ or $y < x$ or $x = y$.*

A path *in a tree $\mathcal{T}$ is a maximal linearly ordered set of nodes. A* branch *in a tree $\mathcal{T}$ is any set of nodes $\{y \mid y \in \pi \text{ and } x < y\}$ for a given path π and a node $x \in \pi$. The least node x of a branch b is the* initial node *of b, denoted by $I(b)$ and b is said to be* stemming from *x. The set of all branches in $\mathcal{T}$ will be denoted by $\mathcal{B}(\mathcal{T})$. If b and c are branches and $b \subseteq c$ then we say that b is a* sub-branch *of c and c is a super-branch of b.*

We will refer to the notion of validity based on trees, as defined above, as *full validity* and to the logic originating from such trees as *OBTL*, or *full Ockhamist logic*. However, in this thesis we will be mainly concerned with the notion of the so-called *bundled validity* and with the bundled logics (introduced in [31]), in which the modal quantification over branches is restricted to a given set.

Definition 2.16. *Given a tree $\mathcal{T}$, a* bundle *B on $\mathcal{T}$ is a subset of $\mathcal{B}(\mathcal{T})$ closed under sub-branches and super-branches and such that every node of $\mathcal{T}$ belongs to some branch in B. A* bundled tree *is a pair $(\mathcal{T}, B)$ where $\mathcal{T}$ is a tree and B is a bundle on $\mathcal{T}$. We say that a bundled tree $(\mathcal{T}, B)$ is* complete *when $B = \mathcal{B}(\mathcal{T})$.*

We can define the semantics for such logics by providing trees with a valuation function. With respect to this point, we notice that different branching-time logics are defined according to the policy we associate to such valuations. Many authors (see, e.g., [128]) assume that propositional symbols refer in some way to the future. A consequence of this assumption is that the valuation of an atom depends not only on the node we are considering but also on a particular branch containing that node. Thus the valuation function is defined in terms of pairs *(branch, instant)*.

A different point of view consists in assuming that propositional symbols contain *no trace of futurity* [136]. This leads to consider all the branches starting from a given instant in a tree-like frame as sharing the same evaluation of every propositional variable.

In the following, we will adopt this *no trace of futurity* approach (we will sometimes also call it *atomic harmony* assumption), since it is more common in computer science-oriented branching temporal logics.[5] Namely, the logics presented in this section are those described in [167] with the only difference that we adopt, as, e.g., in [136], the atomic harmony assumption. As a consequence, we have that the classical substitution rule is not a valid deduction rule in the axiomatizations of our logics, e.g., the validity of the formula $p \supset \forall p$ is not preserved under substitution.

Definition 2.17. *Given a bundled tree $(\mathcal{T}, B)$, a* valuation *$\mathcal{V}$ on $(\mathcal{T}, B)$ is a function assigning a (possibly empty) set of propositional symbols to each branch in B, such that if $I(b) = I(b')$ then $\mathcal{V}(b) = \mathcal{V}(b')$.*

Given a bundled tree $(\mathcal{T}, B)$ and a valuation $\mathcal{V}$ on it, truth for an Ockhamist formula at a branch $b \in B$ is the smallest relation $\models$ defined as follows:

$$
\begin{array}{lll}
\mathcal{M}, b \models p & \textit{iff} & p \in \mathcal{V}(b); \\
\mathcal{M}, b \models A \supset B & \textit{iff} & \mathcal{M}, b \models A \textit{ implies } \mathcal{M}, b \models B; \\
\mathcal{M}, b \models \mathsf{G}A & \textit{iff} & \textit{for all } b' \in B \textit{ s.t. } b \subset b',\ \mathcal{M}, b' \models A; \\
\mathcal{M}, b \models \mathsf{H}A & \textit{iff} & \textit{for all } b' \in B \textit{ s.t. } b' \subset b,\ \mathcal{M}, b' \models A; \\
\mathcal{M}, b \models \forall A & \textit{iff} & \textit{for all } b' \in B \textit{ s.t. } I(b) = I(b'),\ \mathcal{M}, b' \models A.
\end{array}
$$

Semantics in terms of Ockhamist frames

In order to give a semantics to bundled logics in a more traditional Kripke style, we can give a different characterization of bundled trees. Namely we can view a bundled tree $(\mathcal{T}, B)$ as a triple $(\mathcal{W}, \prec, \simeq)$, in which:

- $\mathcal{W}$ is B, i.e. the set of branches of the bundled tree;
- $\prec$ is $\supset$, i.e. the inclusion relation between branches;
- $\simeq$ is the relation of having the same initial point, i.e. $b \simeq c$ iff $I(b) = I(c)$.

The structures that we obtain correspond to the Ockhamist frames of, e.g., [167].

[5] In fact, both the most well-known computation tree logics, *CTL* and *CTL** (see Section 2.4.2), rely on this assumption.

Definition 2.18. *A* basic frame *is a triple* $(\mathcal{W}, \prec, \simeq)$*, where* $\mathcal{W}$ *is a non-empty set,* $\prec$ *is a union of irreflexive linear orders on* $\mathcal{W}$ *and* $\simeq$ *is an equivalence relation on* $\mathcal{W}$*.*

An Ockhamist frame *is a basic frame* $(\mathcal{W}, \prec, \simeq)$*, satisfying the following conditions:*

(Dis) if $x \simeq y$ *then* $x \not\prec y$ *;*
(PI) if $x \simeq y$*, then there exists an order-isomorphism* f *between* $\{z \mid z \prec x\}$ *and* $\{z \mid z \prec y\}$ *such that for all* $z \prec x$*,* $z \simeq f(z)$ *;*
(WDC) if $x \prec y \simeq y'$ *, then there exists* x' *such that* $x \simeq x' \prec y'$ *;*
(MB) if $x \simeq y$ *and* $x \neq y$*, then there exists* $x' \succ x$ *such that for all* $z \succ y$ *not-*$(x' \simeq z)$*.*

(Dis) stays for *disjointness of* $\prec$ *and* $\simeq$ and comes from the irreflexivity of $\prec$. (PI) expresses the *past isomorphism* of two points that are $\simeq$-related, while (WDC) stays for *weak diagram completion* and both properties are consequences of the left linearity of $\prec$. Finally, since two distinct branches in a tree must have disjoint subbranches, a property expressing the *maximality of branches* holds.

It is possible to prove (see [167]) that for every Ockhamist frame there exists a corresponding bundled tree, from which the Ockhamist frame can be built as suggested above. Thus the semantics generated by bundled trees is exactly the same that we get when we consider Ockhamist frames. In the following we choose to refer to Ockhamist frames, since this gives us the possibility of defining the notion of truth in a pure Kripke-style. We anticipate that this possibility is in fact what will allow us, in Chapter 5, to extend the labeled deduction framework used for standard modal logics to the context of these branching-time logics.

Note also that the properties (Dis), (PI), (WDC) and (MB) are not completely independent one of each other, e.g. (Dis) + (WDC) implies (PI). We enumerate all of them because, as in [167], this gives us the possibility of considering several intermediate logics, according to which of the conditions above we require the frames to satisfy. In particular, we will consider, in the rest of the thesis, the following classes of frames.

Definition 2.19. *A* (Dis)-frame *is a basic frame satisfying the condition (Dis). A* (WDC)-frame *is a basic frame satisfying the condition (WDC). A* (Dis+WDC)-frame *is a (Dis)-frame that is also a (WDC)-frame.*

As usual, we can obtain a class of structures from each class of frames considered, by providing the frames with a valuation function. As we remarked above when defining valuation functions for trees, the policy that we follow in this thesis is such that all the points $\simeq$-related in an Ockhamist frame satisfy the same set of atoms.

Definition 2.20. *Let* $\mathcal{P}$ *be a denumerable set of propositional symbols. A* basic (Dis, WDC, Dis+WDC, Ockhamist) structure *is a* 4*-ple* $(\mathcal{W}, \prec, \simeq, \mathcal{V})$*, where* $(\mathcal{W}, \prec, \simeq)$ *is a basic (Dis, WDC, Dis+WDC, Ockhamist) frame and* $\mathcal{V}$ *is a valuation function* $\mathcal{V} : \mathcal{W} \to 2^{\mathcal{P}}$ *such that for all* $u, v \in \mathcal{W}$*, if* $u \simeq v$ *then* $\mathcal{V}(u) = \mathcal{V}(v)$*.*

Now we give the notion of truth with respect to a point in a structure. Note that truth is defined by having the temporal operators G and H operate along the $\prec$-lines of points, and the quantifier $\forall$ within a $\simeq$-equivalence class.

Definition 2.21. *Given a basic (Dis, WDC, Dis+WDC, Ockhamist) structure* $\mathcal{M} = (\mathcal{W}, \prec, \simeq, \mathcal{V})$ *and a point* $u \in \mathcal{W}$ *the corresponding notion of basic (Dis, WDC, Dis+WDC, Ockhamist) truth for a Ockhamist formula is the smallest relation* $\models$ *defined as follows:*

$$\begin{array}{lll} \mathcal{M}, u \models p & \textit{iff} & p \in \mathcal{V}(u); \\ \mathcal{M}, u \models A \supset B & \textit{iff} & \mathcal{M}, u \models A \textit{ implies } \mathcal{M}, u \models B; \\ \mathcal{M}, u \models \mathsf{G}A & \textit{iff} & \textit{for all } v \textit{ s.t. } u \prec v, \mathcal{M}, v \models A; \\ \mathcal{M}, u \models \mathsf{H}A & \textit{iff} & \textit{for all } v \textit{ s.t. } v \prec u, \mathcal{M}, v \models A; \\ \mathcal{M}, u \models \forall A & \textit{iff} & \textit{for all } v \textit{ s.t. } u \simeq v, \mathcal{M}, v \models A; \end{array}$$

As is standard, we can extend this notion of truth to the notions of basic (Dis, WDC, Dis+WDC, Ockhamist) satisfiability and validity.

In the following, we will use the symbols $\models_{Dis}$, $\models_{WDC}$, etc. to refer to the corresponding notions of truth and validity. $\models_{bas}$ will denote basic truth/validity. $\models_{\mathcal{O}}$ will denote Ockhamist truth/validity. We will refer to the logic of Ockhamist frames also as *BOBTL*. Sometimes we will also consider frames and validities originating from other combinations, e.g., (Dis+PI)-validity is the notion of validity determined by (Dis+PI)-frames, i.e. by basic frames satisfying both the properties (Dis) and (PI).

Some interesting results concerning the relations between these notions of validity are described in [167]. First of all, it has been shown that, as long as validity is concerned, the property (MB) can be replaced by:

(MB$^-$) if x is a $\prec$-maximal element, then, for every y, $x \simeq y$ implies $x = y$.

Moreover, if we put ourselves in the no trace of futurity setting, we can further simplify the maximality of branches property as follows:

(MB^{--}) if x is a $\prec$-maximal element, and $x \simeq y$, then y is a $\prec$-maximal element.

We introduce also another property that will be useful in the following sections. It can be seen as a strong form of (WDC) and will be referred to as *strong diagram completion*:

(SDC) if $x \prec y \prec z \simeq z' \succ x' \simeq x$, then there exists y' such that $y' \simeq y$ and $x' \prec y' \prec z'$.

It is interesting because one can prove that the logic determined by (Dis+WDC)-frames and the logic determined by (WDC+SDC)-frames coincide.

We collect in the following lemma some comparison results that can be easily adapted from [167].

Lemma 2.22. *Basic validity and (Dis)-validity coincide. (Dis+WDC)-validity, (Dis+PI)-validity and (WDC+SDC)-validity coincide. (Dis+WDC+MB)-validity, (Dis+WDC+MB*$^-$*)-validity, (Dis+WDC+MB*$^{--}$*)-validity and Ockhamist validity coincide.*

Proof. By trivial adaptations of the analogous results proved in [167] in the case where no assumptions are made about the evaluation of the atoms.

□

Hilbert-style axiomatizations

Hilbert-style axiomatizations for several bundled Ockhamist logics have been proposed in [68, 136, 164, 167]. In this section, we present the ones corresponding to the logics considered above.

Note that for the full Ockhamist logic $OBTL$, i.e. the logic of complete bundled trees, as for its corresponding computation tree logic CTL^*, no finitary complete axiomatization is known.

The logic of basic frames (or (Dis)-frames)

First, we present a Hilbert-style axiomatization $\mathcal{H}(bas)$ (slightly adapted from [167]) for the logic of basic frames (or, which is the same, the logic of Dis-frames). We have that the temporal axioms for linear time, plus the modal axioms for *S5* with respect to the operator $\forall$, plus a rule for atomic harmony (i.e., branches with the same initial point satisfy the same atoms), plus the usual deduction rules form a complete deductive system.

(CL) Any tautology instance of classical propositional logic
(K_{G}) $\mathsf{G}(A \supset B) \supset (\mathsf{G}A \supset \mathsf{G}B)$
(K_{H}) $\mathsf{H}(A \supset B) \supset (\mathsf{H}A \supset \mathsf{H}B)$
$(K_{\forall})$ $\forall(A \supset B) \supset (\forall A \supset \forall B)$
(GP) $A \supset \mathsf{GP}A$
(HF) $A \supset \mathsf{HF}A$
$(L1)$ $\mathsf{F}A \supset \mathsf{G}(\mathsf{F}A \vee A \vee \mathsf{P}A)$
$(L2)$ $\mathsf{P}A \supset \mathsf{H}(\mathsf{F}A \vee A \vee \mathsf{P}A)$
$(L3)$ $\mathsf{G}A \supset \mathsf{GG}A$
$(L4)$ $\mathsf{H}A \supset \mathsf{HH}A$
$(\forall 1)$ $\forall A \supset \forall\forall A$
$(\forall 2)$ $\forall A \supset A$
$(\forall 3)$ $A \supset \forall\exists A$
$(Atom)$ $p \supset \forall p$ for each atomic proposition p

Notice that the axioms above have to be considered axiom schemata: in fact, because of the axiom $(Atom)$, the common rule of substitution does not hold for this logic.

The rules of inference are the following:

(MP) If A and $A \supset B$ then B
(Nec_{G}) If A then $\mathsf{G}A$
(Nec_{H}) If A then $\mathsf{H}A$
$(Nec_{\forall})$ If A then $\forall A$

As usual, we define the notions of *theorem of* $\mathcal{H}(bas)$ and *derivability in* $\mathcal{H}(bas)$ (denoted $\vdash_{bas}$).

The logic of (WDC)-frames

Such an axiomatization can be extended to capture the logic of (WDC)-frames by adding the following axiom (from [167]). We denote with $\mathcal{H}(WDC)$ the resulting axiomatization.

$$(WDC) \qquad \mathsf{P}A \supset \forall \mathsf{P} \exists A$$

The logic of (Dis+WDC)-frames

The logic of (Dis+WDC)-frames is much more difficult to capture by means of Hilbert-style axioms. The use of a form of the Gabbay *irreflexivity rule* [64] as a further deduction rule greatly simplifies the task, as proposed in [68].

In [164], Zanardo proposes the following two rather complex (but with a standard form) Hilbert-style axioms:

$$\begin{aligned}(DW1)\ & \mathsf{P}(\forall A \wedge \mathsf{G}B) \wedge \mathsf{H}\neg(B \wedge \exists C) \\ & \supset \forall[\mathsf{G}A_1 \wedge \mathsf{P}C \supset \mathsf{P}(A \wedge (C \vee \mathsf{P}C)) \wedge \mathsf{G}(C \supset \mathsf{G}A_1)]\end{aligned}$$

$$\begin{aligned}(DW2)\ & [\mathsf{H}A \wedge \mathsf{H}\neg(B \wedge \exists C \wedge \mathsf{F}(B \wedge A \wedge \exists C_1)) \wedge \mathsf{P}(\forall A_1 \wedge \mathsf{G}B)] \\ & \supset \forall[\mathsf{G}B_1 \supset \mathsf{P}(A_1 \wedge \mathsf{G}(C \supset \mathsf{G}(C_1 \supset \mathsf{G}B_1)))]\end{aligned}$$

The addition of them to the ones for the logic of (WDC)-frames gives an axiomatization $\mathcal{H}(Dis + WDC)$ for the (Dis+WDC)-frames logic.

The logic BOBTL of Ockhamist frames

Finally, we get an axiomatization $\mathcal{H}(\mathcal{O})$ for the logic $BOBTL$ by adding the following axiom expressing the maximality of branches.

$$(MB^{--}) \qquad \mathsf{G}\,\bot \supset \forall \mathsf{G}\,\bot$$

Theorem 2.23 (Soundness and completeness). *The Hilbert-style axiomatizations $\mathcal{H}(bas)$, $\mathcal{H}(WDC)$, $\mathcal{H}(Dis + WDC)$ and $\mathcal{H}(\mathcal{O})$ are sound and complete with respect to the corresponding semantics.*

Proof. The axiomatizations are trivial adaptations of the ones given in [164] and [167] for a version of the logics that did not consider atomic harmony. Proofs can be easily adapted to deal with the case in which branches with the same initial node agree on the valuation of propositional symbols.

□

Related logics

Although they will not be explicitly treated in this thesis, it is worth mentioning some variations and extensions of the logics presented above. They include the logics obtained by adding until and since operators [166] and logics originating from allowing the truth of propositional symbols to be dependent both on branches and time-instants [128, 136, 167].

Finally, we remark that we focused here on Ockhamist branching logics. Another important class is that of Peircean branching logics [128, 129, 165], in which

truth of all formulas depends only on the time instant of evaluation and not on a branch. In other words, all the formulas can be considered to be state formulas. An example is represented by a sublanguage of the Ockhamist logics above, obtained by allowing the combination of branching and linear operators only in the form of a single linear operator preceded by a single path quantifier, as in $\forall\mathsf{G}$, $\forall\mathsf{H}$, $\forall\mathsf{F}$, $\forall\mathsf{P}$, $\exists\mathsf{G}$, $\exists\mathsf{H}$, $\exists\mathsf{F}$ and $\exists\mathsf{P}$.

2.4.2 Computation tree logics

In this section, we present some branching temporal logics that are more common in computer science and are usually referred to as computation tree logics.

*CTL**

The logic *CTL** has been introduced in [55] as an extension of the less expressive *CTL*. Here we first define *CTL** and then specify which is the subset corresponding to *CTL*.

Syntax

The language of *CTL** extends that of Ockhamist logics presented in Section 2.4.1 with the linear temporal operator *until* and restricts the attention to future-oriented operators.

Definition 2.24. *Given a set $\mathcal{P}$ of propositional symbols, the set of* (well-formed) *CTL**-formulas *is defined by the grammar*

$$A ::= p \mid \bot \mid A \supset A \mid \mathsf{G}A \mid \mathsf{X}A \mid A\mathsf{U}B \mid \forall A\,,$$

where $p \in \mathcal{P}$. The set of atomic formulas *is $\mathcal{P} \cup \{\bot\}$. The* complexity *of a formula is the number of occurrences of connectives ($\supset$), operators (X, G and U) and path quantifiers ($\forall$).*

Semantics

Several alternative characterizations can be given for *CTL** and the other computation tree logics and some equivalence results have been shown (see, e.g., [51]).

In particular, as for the Ockhamist logics seen in Section 2.4.1, we can give two main notions of validity: the *full validity* and the *bundled validity*; for a detailed account see [52, 135].

The notion of validity underlying the semantics of *CTL** is the full one.

If we define a *transition system* as consisting of a set S of states and of a serial[6] relation $\mathcal{R}$ on S, i.e. a relation such that for every s in S there exists a t in S for which $s\mathcal{R}t$ holds, then the notion of full validity is given by defining the semantics with respect to the set of all the $\mathcal{R}$-generable paths, i.e. of all the ω-sequences $s_1, s_2, ...$ such that $(s_i, s_{i+1}) \in \mathcal{R}$ for all $i \in \mathbb{N}$. The following definitions formalize these notions.

[6] In the computer science literature, the condition of seriality is often referred to as totality.

Definition 2.25. *A* transition system *is a pair $\mathcal{F} = (\mathcal{S}, \mathcal{R})$ where:*

- *$\mathcal{S}$ is a non empty set of states;*
- *$\mathcal{R}$ is a serial binary relation on $\mathcal{S}$, i.e. for each $s \in \mathcal{S}$ there exists $t \in \mathcal{S}$ such that $(s,t) \in \mathcal{R}$.*

Given a set $\mathcal{P}$ of propositional symbols, a labeled transition system *is a triple $\mathcal{M} = (\mathcal{S}, \mathcal{R}, \mathcal{V})$ where:*

- *$(\mathcal{S}, \mathcal{R})$ is a transition system;*
- *$\mathcal{V} : \mathcal{S} \to 2^{\mathcal{P}}$ is a (*labeling*) function that assigns to each state in $\mathcal{S}$ a (possibly empty) set of propositional symbols.*

A fullpath *(or just* path*) in a (labeled) transition system $\mathcal{M} = (\mathcal{S}, \mathcal{R}, \mathcal{V})$ is an infinite sequence $s_0, s_1, s_2, \ldots$ of states in $\mathcal{S}$.*

Given a fullpath $\sigma = s_0, s_1, s_2, \ldots$, we write σ^i to denote the suffix path $s_i, s_{i+1}, s_{i+2}, \ldots$ and $\sigma(i)$ to denote the i-th state of σ, i.e. s_i.

Note that we are considering here the case of monomodal transition systems: the generalization to transition systems with more relations (actions) is straightforward.

It is quite common to present the language of computation tree logics by distinguishing between *state formulas*, which are evaluated with respect to a state, and *path formulas*, which are evaluated with respect to a fullpath.

The distinction between state and path formulas is specified by the following alternative formulation of the language of *CTL**-formulas:

$$S ::= p \mid \bot \mid S \supset S \mid \forall P$$

$$P ::= S \mid P \supset P \mid \mathsf{X}P \mid \mathsf{G}P \mid P \,\mathsf{U}\, P,$$

where S denotes the category of state formulas and P the category of path formulas.

It is also possible to define the notion of truth for a formula just with respect to fullpaths, by assuming that an atomic proposition is true at a fullpath σ iff it is true at the initial state of σ. Note that here, as in *LTL*, and since it seems to be more common in the literature, we assume the relation behind linear temporal operators to be reflexive.

Definition 2.26. Truth in the logic *CTL* for a CTL*-formula at a fullpath σ in a labeled transition system $\mathcal{M} = (\mathcal{S}, \mathcal{R}, \mathcal{V})$ is the smallest relation $\models_{CTL^*}$ satisfying:*

$$\begin{array}{rll}
\mathcal{M}, \sigma \models_{CTL^*} p & \textit{iff} & p \in \mathcal{V}(\sigma(0)) \\
\mathcal{M}, \sigma \models_{CTL^*} A \supset B & \textit{iff} & \mathcal{M}, \sigma \models_{CTL^*} A \textit{ implies } \mathcal{M}, \sigma \models_{CTL^*} B \\
\mathcal{M}, \sigma \models_{CTL^*} \mathsf{G}A & \textit{iff} & \mathcal{M}, \sigma^i \models_{CTL^*} A \textit{ for all } i \geq 0 \\
\mathcal{M}, \sigma \models_{CTL^*} \mathsf{X}A & \textit{iff} & \mathcal{M}, \sigma^1 \models_{CTL^*} A \\
\mathcal{M}, \sigma \models_{CTL^*} A \mathsf{U} B & \textit{iff} & \mathcal{M}, \sigma^j \models_{CTL^*} B \textit{ for some } j \geq 0 \textit{ and} \\
 & & \mathcal{M}, \sigma^k \models_{CTL^*} A \textit{ for every } 0 \leq k < j \\
\mathcal{M}, \sigma \models_{CTL^*} \forall A & \textit{iff} & \mathcal{M}, \tau \models_{CTL^*} A \textit{ for every fullpath } \tau \textit{ s.t. } \tau(0) = \sigma(0)
\end{array}$$

By extension, given a CTL-formula A and a set of CTL*-formulas Γ, we write:*

$$\begin{array}{rll} \mathcal{M} \models_{CTL^*} A & \textit{iff} & \mathcal{M}, \sigma \models_{CTL^*} A \textit{ for every fullpath } \sigma \\ \mathcal{M} \models_{CTL^*} \Gamma & \textit{iff} & \mathcal{M} \models_{CTL^*} A \textit{ for all } A \in \Gamma \\ \Gamma \models_{CTL^*} A & \textit{iff} & \mathcal{M} \models_{CTL^*} \Gamma \textit{ implies } \mathcal{M} \models_{CTL^*} A, \textit{ for every labeled transition system } \mathcal{M} \\ \models_{CTL^*} A & \textit{iff} & \mathcal{M} \models_{CTL^*} A \textit{ for every labeled transition system } \mathcal{M}. \end{array}$$

As in the previous sections, we can generalize this notion of truth to the notions of satisfiability and validity and define CTL^* as the set of formulas that are CTL^*-valid according to the resulting semantics.

We remark that a (kind of "unorthodox") Hilbert-style axiomatization for CTL^* has been provided by Reynolds [135], by using a special *auxiliary atoms* rule, which allows for adding new atoms in a derivation.

CTL

The sublogic CTL is obtained by restricting the syntax of CTL^* to disallow boolean combinations and nestings of linear-time operators, i.e. linear-time operators can appear only immediately preceded by a path quantifier. While CTL^* can be seen as the computational version of Ockhamist branching-time logic, CTL can be considered the computational version of the *Peircean* branching logic (for more details on this, consult, e.g., [79]).

Given this syntactic restriction, the semantics of CTL is trivially inferred from the one of CTL^*, i.e. a CTL-formula is CTL-valid iff it is CTL^*-valid. In other words, CTL^* is a conservative extension of CTL.

Since in the rest of the thesis the focus will be on Ockhamist logics, we do not go into details concerning CTL; the interested reader can see [52]. A further restriction consists in considering the until-free fragment of CTL, presented in [13] with the name of UB.

*BCTL**

As we anticipated when presenting CTL^*, it is possible to give a generalized semantics, by considering more general structures. This gives rise to a logic that is a subset of CTL^* and is usually named $BCTL^*$[7] [139], i.e. *bundled* CTL^*.

The language considered is the same of CTL^* (see Section 2.4.2).

Semantics in terms of transition systems

In order to introduce the semantics of $BCTL^*$, we recall that the semantics of CTL^* is given by considering all the $\mathcal{R}$-generable paths of a transition system. The notion of *bundled validity*, in the context of computation tree logics, is obtained by restricting the set P of admissible paths. The only requirement that such restricted set has to satisfy is given by the following conditions:

[7] This logic coincides with the logic determined by the deductive system $\forall LTFC$ described in [149].

(*i*) *suffix-closure*, i.e. if the path $s_0, s_1, s_2...$ is in P then the path $s_1, s_2, ...$ is also in P; and

(*ii*) *fusion-closure*, i.e. if $s_1, s_2, ..., s_n, s_{n+1}, s_{n+2}, ...$ and $s'_1, s'_2, ..., s'_{n-1}, s_n, s'_{n+1}, s'_{n+2}, ...$ are in P then $s_1, s_2, ..., s_n, s'_{n+1}, s'_{n+2}, ...$ is also in P.

We remark that, in order to retrieve the set of all the $\mathcal{R}$-generable paths, a third condition needs to be added (a proof is in [51]):

(*iii*) *limit-closure*, i.e. if the paths (s_1, σ_1), (s_1, s_2, σ_2), $(s_1, s_2, s_3, \sigma_3)$, etc. are in P then the path $(s_1, s_2, s_3, \ldots)$, which is the limit of the prefixes (s_1), (s_1, s_2), (s_1, s_2, s_3), etc. is also in P.

An example showing that the full and the bundled validity are distinct notions is given by the formula $A \equiv \forall \mathsf{G}(p \supset \exists \mathsf{X} p) \supset (p \supset \exists \mathsf{G} p)$, where p is an atomic formula. It is possible to check (see [135]) that A is valid with respect to the full semantics, i.e. in CTL^*, but not with respect to the bundled one, i.e. in $BCTL^*$.

Thus the notion of truth in $BCTL^*$ can be inferred from that given for CTL^* in Definition 2.32. The only difference is that now we consider not just labeled transition systems but also all the variants of such systems obtained by restricting the set of admissible paths to a subset, satisfying suffix- and fusion-closure, of the set of all paths. This means that we have a greater number of structures, i.e. a smaller set of valid formulas.

In [42], it has been shown that it is possible to give a precise characterization of the family of transition systems giving rise to the logic $BCTL^*$. Such a definition consists in endowing transition systems with a mechanism for excluding those computation paths that do not fit some *fairness* requirements.

Definition 2.27. *A fair* transition system *is a triple $\mathcal{F} = (\mathcal{S}, \mathcal{R}, \mathcal{C})$ where:*

- *$(\mathcal{S}, \mathcal{R})$ is a transition system;*
- *$\mathcal{C} \subseteq 2^{\mathcal{S}} \times 2^{\mathcal{S}}$ is the* fairness condition*.*

$\mathcal{C}$ is a set of pairs (X_i, Y_i) of subsets of S and it is used to define the set of fair paths through $\mathcal{F}$.

A fullpath *is defined as for transition systems. Given a set $X \subseteq \mathcal{S}$ and a fullpath σ, we define the* size *of the intersection of X with σ (denoted $|X \cap \sigma|$) as the cardinality of the set $\{j \in \omega | \sigma(j) \in X\}$. A fullpath σ is* fair *iff, for all pairs $(X_i, Y_i) \in \mathcal{C}$, if $|X_i \cap \sigma|$ is infinite, then $|Y_i \cap \sigma|$ is also infinite.*

Given a set $\mathcal{P}$ of propositional symbols, a fair labeled transition system *is a* 4*-ple $\mathcal{M} = (\mathcal{S}, \mathcal{R}, \mathcal{C}, \mathcal{V})$ where:*

- *$(\mathcal{S}, \mathcal{R}, \mathcal{C})$ is a fair transition system;*
- *$\mathcal{V} : \mathcal{S} \to 2^{\mathcal{P}}$ is a (*labeling*) function that assigns to each state in $\mathcal{S}$ a (possibly empty) set of propositional symbols.*

Then a notion of truth given in terms of fair transition systems can be obtained from Definition 2.26 by letting the quantification range over just fair paths.

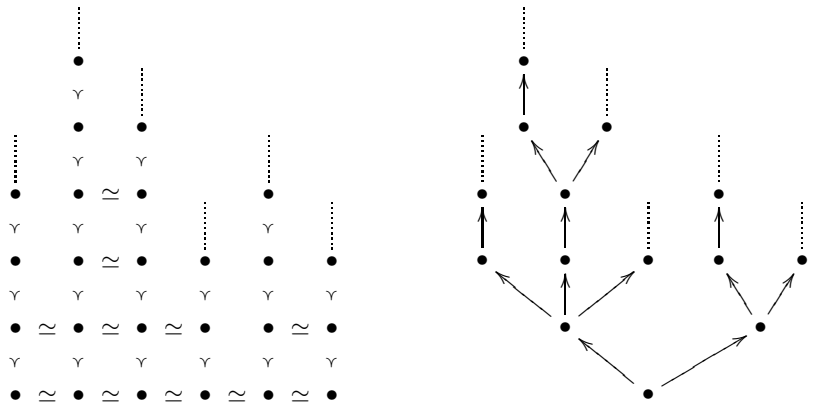


Fig. 2.1. An *Ockhamist frame* (left) and the corresponding *bundled tree* (right).

Semantics in terms of Ockhamist frames

However, here we prefer to consider a different but equivalent semantical formulation given by frames where the basic entities (or *worlds*, in a Kripke-style terminology) are the paths of computation rather than the states. In fact, this view allows us to present a more genuine Kripke-style semantics.

We thus introduce $(\mathbb{N} \times \mathcal{W})$-structures [135], which are closely related to the Kamp and Ockhamist structures, described respectively in [150] and [167], and introduced in Section 2.4.1.

We need to adapt the general notion of Ockhamist frame to a notion that considers the fact that the flow of time behind each computation is now required to be isomorphic to the set of natural numbers.

Definition 2.28. *A* floored Ockhamist frame *(of countable height) is a triple* $(\mathcal{T}, \prec, \simeq)$ *where:*

1. $\mathcal{T}$ *is the set of points;*
2. $\prec$ *is a transitive, anti-symmetric, irreflexive, linear relation on* $\mathcal{T}$*, i.e.:*
   a) $\forall x, y, z.\, ((x \prec y) \wedge (y \prec z)) \Rightarrow (x \prec z)$;
   b) $\forall x, y.\, \neg\, ((x \prec y) \wedge (y \prec x))$;
   c) $\forall x.\, \neg\, (x \prec x)$;
   d) $\forall x, y, z.\, ((x \prec y) \wedge (x \prec z)) \Rightarrow ((z \prec y) \vee (z = y) \vee (y \prec z))$;
   e) $\forall x, y, z.\, ((y \prec x) \wedge (z \prec x)) \Rightarrow ((z \prec y) \vee (z = y) \vee (y \prec z))$;
3. $\{y \mid y \prec x\}$ *is finite for each* $x \in \mathcal{T}$;
4. $\simeq$ *is an equivalence relation such that:*
   a) *if* $x \simeq y$ *then it is not the case that* $x \prec y$;
   b) *if* $x \simeq y$ *and* $u \prec x$ *then there is a* v *such that* $v \prec y$ *and* $u \simeq v$;
5. *there is an element* $0 \in \mathcal{T}$ *such that for each* $w \in \mathcal{T}$*, there is a* $w' \in \mathcal{T}$ *such that* $0 \simeq w'$ *and either* $w' \prec w$ *or* $w' = w$ *(the equivalence class* $0/_{\simeq}$ *is known as the* floor*).*

Intuitively, every Ockhamist point can be thought of as corresponding to a path in a transition system and the relation $\prec$ as the equivalent of the relation "is a prefix of", i.e. $x \prec y$ stands for "the path x is a prefix of the path y". The branching nature of Ockhamist frames is hidden in the $\simeq$-equivalence relation, where the idea is that each $\simeq$-class of points contains all the paths of the corresponding transition system that share a same initial state.

More precisely, there exists an equivalence [138] between Ockhamist frames (or their unwindings into bundled trees, as exemplified in Fig. 2.1) and fair transition systems. Such an equivalence is based on the fact that Ockhamist points correspond to paths in the transition system while points related by $\simeq$ correspond to paths with the same initial state.

In order to give a proper semantics for every linear temporal operator, we require the lines of points defined by $\prec$ to be isomorphic to the natural numbers.

Definition 2.29. *An Ockhamist frame* $(\mathcal{T}, \prec, \simeq)$ *is an* $(\mathbb{N} \times \mathcal{W})$-frame *iff*

(i) there is some set $\mathcal{W}$ *such that* $\mathcal{T} = (\mathbb{N} \times \mathcal{W})$*;*
(ii) the order $\prec$ *is defined by* $(n, u) \prec (m, v)$ *iff* $n < m$ *and* $u = v$*.*

As usual, we obtain a structure by providing the frame a valuation function. In this case, as for the logics of Section 2.4.1, we also require that all points in a $\simeq$-equivalence class satisfy the same set of atoms.

Definition 2.30. *The structure* $(\mathcal{T}, \prec, \simeq, \mathcal{V})$ *is an* $(\mathbb{N} \times \mathcal{W})$-structure *iff* $(\mathcal{T}, \prec, \simeq)$ *is an* $(\mathbb{N} \times \mathcal{W})$*-frame,* $\mathcal{V} : (\mathbb{N} \times \mathcal{W}) \to 2^{\mathcal{P}}$*, and for all* $n \in \mathbb{N}$ *and for all* $u, v \in \mathcal{W}$*, if* $(n, u) \simeq (n, v)$ *then* $\mathcal{V}(n, u) = \mathcal{V}(n, v)$*.*

It is easy to show by induction the following lemma (see [138]), which will be useful later on.

Lemma 2.31. *Given an* $(\mathbb{N} \times \mathcal{W})$-structure $(\mathcal{T}, \prec, \simeq, \mathcal{V})$ *and two points* (n, w) *and* (m, v) *in* $\mathcal{T}$*, if* $(n, w) \simeq (m, v)$ *then* $n = m$*.*

Definition 2.32. *Given an* $(\mathbb{N} \times \mathcal{W})$*-structure* $\mathcal{M} = (\mathcal{T}, \prec, \simeq, \mathcal{V})$*, where* $\mathcal{T} = (\mathbb{N} \times \mathcal{W})$ *for some set* $\mathcal{W}$*,* truth in the logic *BCTL** *for a CTL***-formula at a point* $(n, w) \in \mathcal{T}$ *is the smallest relation* $\models_{BCTL^*}$ *satisfying:*

$$
\begin{array}{rll}
\mathcal{M}, (n, w) \models_{BCTL^*} p & \textit{iff} & p \in \mathcal{V}(n, w) \\
\mathcal{M}, (n, w) \models_{BCTL^*} A \supset B & \textit{iff} & \mathcal{M}, (n, w) \models_{BCTL^*} A \textit{ implies } \mathcal{M}, (n, w) \models_{BCTL^*} B \\
\mathcal{M}, (n, w) \models_{BCTL^*} \mathsf{G}A & \textit{iff} & \mathcal{M}, (m, w) \models_{BCTL^*} A \textit{ for all } m \geq n \\
\mathcal{M}, (n, w) \models_{BCTL^*} \mathsf{X}A & \textit{iff} & \mathcal{M}, (n+1, w) \models_{BCTL^*} A \\
\mathcal{M}, (n, w) \models_{BCTL^*} A \mathsf{U} B & \textit{iff} & \mathcal{M}, (m, w) \models_{BCTL^*} B \textit{ for some } m \geq n \textit{ and} \\
 & & \mathcal{M}, (m', w) \models_{BCTL^*} A \textit{ for every } n \leq m' < m \\
\mathcal{M}, (n, w) \models_{BCTL^*} \forall A & \textit{iff} & \mathcal{M}, (n, v) \models_{BCTL^*} A \textit{ for every point } (n, v) \\
 & & \textit{s.t. } (n, w) \simeq (n, v)
\end{array}
$$

As for *CTL**, we can generalize this notion of truth to the notions of logical consequence ($\Gamma \models_{BCTL^*} A$), satisfiability and validity and define *BCTL** as the set of formulas that are *BCTL**-valid according to the resulting semantics.

$BCTL^*_-$: the until-free version of $BCTL^*$

In the rest of the thesis, we will often refer to a syntactic restriction of $BCTL^*$, obtained by just removing the operator *until*.

Syntax

Definition 2.33. *Given a set $\mathcal{P}$ of propositional symbols, the set of* (well-formed) $BCTL^*_-$-formulas *is defined by the grammar*

$$A ::= p \mid \bot \mid A \supset A \mid \mathsf{G}A \mid \mathsf{X}A \mid \forall A\,,$$

where $p \in \mathcal{P}$.

Semantics

$BCTL^*$ is a conservative extension of $BCTL^*_-$: a $BCTL^*_-$-formula is $BCTL^*_-$-valid iff it is $BCTL^*$-valid. We use the symbol $\models_{BCTL^*_-}$ to denote the notion of truth in $BCTL^*_-$; its extension to express logical consequence is also standard.

A Hilbert-style axiomatization for $BCTL^*_-$

Now we give a Hilbert-style axiomatization, which we call $\mathcal{H}(BCTL^*_-)$, for the logic $BCTL^*_-$. $\mathcal{H}(BCTL^*_-)$ consists of two sets of axioms (axioms for linear temporal formulas and axioms for quantified formulas) and a set of inference rules. For the first set of axioms, we refer to a standard axiomatization for until-free LTL [149]:

$(L1)$ Any tautology instance
$(L2)$ $\mathsf{G}(A \supset B) \supset (\mathsf{G}A \supset \mathsf{G}B)$
$(L3)$ $(\mathsf{X}\neg A \supset \neg\mathsf{X}A) \wedge (\neg\mathsf{X}A \supset \mathsf{X}\neg A)$
$(L4)$ $\mathsf{X}(A \supset B) \supset (\mathsf{X}A \supset \mathsf{X}B)$
$(L5)$ $\mathsf{G}A \supset A \wedge \mathsf{XG}A$
$(L6)$ $\mathsf{G}(A \supset \mathsf{X}A) \supset (A \supset \mathsf{G}A)$

The second set of axioms ensures that the path modality $\forall$ behaves as a $\Box$ in the modal logic $S5$ and defines some interactions between the linear temporal operators and the path quantifier. This set of axioms comes from [135] and is slightly different from, but clearly equivalent to, the one in [149]:

$(K_\forall)$ $\forall(A \supset B) \supset (\forall A \supset \forall B)$
$(\forall 1)$ $\forall A \supset \forall\forall A$
$(\forall 2)$ $\forall A \supset A$
$(\forall 3)$ $A \supset \forall\exists A$
$(Atom)$ $p \supset \forall p$ for each atomic proposition p
$(Fusion)$ $\forall\mathsf{X}A \supset \mathsf{X}\forall A$

Finally, we have the inference rules of modus ponens and temporal and path generalization:

(MP) If A and $A \supset B$ then B
(Nec_X) If A then $\mathsf{X}A$
(Nec_G) If A then $\mathsf{G}A$
$(Nec_\forall)$ If A then $\forall A$

The set of theorems of $\mathcal{H}(BCTL^*_-)$ is the smallest set containing the set of axioms above and closed with respect to the rules of inference. Soundness and weak completeness[8] of this axiomatization can be easily verified by adapting analogous proofs for similar axiom systems, as in the following lemma.

Lemma 2.34. *The axiom system $\mathcal{H}(BCTL^*_-)$ is sound and weakly complete for the logic $BCTL^*_-$, i.e. the set of theorems of $\mathcal{H}(BCTL^*_-)$ coincides with the set $BCTL^*_-$.*

Proof. (Sketch) The proof mirrors the one given in [149] for $BCTL^*$, with respect to which our axiom system only misses the two axioms concerning the operator *until*, namely:

$(L7)$ $A\mathsf{U}B \supset \mathsf{F}B$
$(L8)$ $A\mathsf{U}B \leftrightarrow B \vee (A \wedge \mathsf{X}(A\mathsf{U}B))$

where we denote with $\leftrightarrow$ the double implication.

$\mathcal{H}(BCTL^*_-)$ is sound as it is a subset of the axiomatization in [149] and $BCTL^*_-$ structures coincide with $BCTL^*$ structures. A proof of completeness can be easily obtained by adapting the one in [149], which consists of two parts:

(i) first a Henkin-style proof is given for the LTL axiomatization, by the definition of a canonical model construction;
(ii) then such a construction is extended in order to consider the system for $BCTL^*$.

We can modify such a proof for our case by noticing that in (i) the axioms ($L7$) and ($L8$) are used along the proof only to deal with formulas containing the operator *until*. We can use the same arguments to show that the axioms $(L1) - (L6)$ form a complete axiomatization for until-free LTL (as it is done for example in [71]). It is also easy to observe that the arguments in (ii) do not make use of the axioms ($L7$) and ($L8$). Thus, we can mirror part (ii) of the proof in [149] to extend our canonical model construction for until-free LTL to a canonical model construction for $BCTL^*_-$. The main idea here is to consider the equivalence relation between points of the linear canonical model that satisfy the same state formulas and take such equivalence classes as the points of the branching canonical model.

□

[8] On the impossibility of giving a finitary and strongly complete axiomatization for $BCTL^*_-$, see the discussion about $\mathcal{H}(LTL)$ in Section 2.3.4.

3

Labeled Natural Deduction for Modal Logics

3.1 Introduction

Labeling [10, 61, 66] (sometimes also called prefixing, annotating or subscripting) is a method designed for giving uniform presentations of logics, typically the non-classical ones, such as modal, substructural or non-monotonic logics. Labeling allows one to explicitly encode additional information, of a semantic or proof-theoretical nature, that is otherwise implicit in the logic one wants to capture. Such additional information is typically internalized in the syntax by means of proper labels. So, for instance, we will consider a labeled formula of the form $b : A$ instead of the standard logical formula A. Some possible interpretations of the label b in a formula $b : A$, as suggested by Gabbay in [66], are the following:

- possible world where A holds (modal logics);
- time instant when A holds (temporal logics);
- fuzzy reliability value, i.e. b is a number between 0 and 1 (fuzzy logics);
- origin of A, i.e. b indicates where the input A comes from (databases).

This general approach has then been used [4,9,10,23,43,66,93,103,119,148,159] in the context of several different logics and with respect to different classes of deduction systems: natural deduction, sequent calculus, tableaux methods.

Since in the thesis we will mainly deal with natural deduction systems [73], the rest of this chapter will be devoted to give a general presentation of natural deduction and to consider the specific example of the application of labeling techniques to natural deduction systems. In particular, we will illustrate the use of labeled natural deduction in the case of modal logic. This will provide a basis for the definition of labeled natural deduction systems for temporal logics, which will be treated in Chapters 4 and 5.

The structure of this chapter is the following:

- in Section 3.2, we present the basis of classical natural deduction and give a brief description of normalization in the context of natural deduction;
- in Section 3.3, we discuss the adaptation of natural deduction to the case of modal logics and, in particular, present an approach to natural deduction for modal logics based on labeling.

3.2 Natural Deduction

Natural deduction is the term used to denote a class of deduction systems that have been first proposed by Gentzen [73].

A key property of natural deduction systems is the fact that they formalize intuitive reasoning very closely. This is mainly due to the possibility of reasoning "under assumptions", e.g., in order to prove $A \supset B$ one can assume the truth of A and prove (under such an assumption) the truth of B. During the deduction process, the assumption A is *active* and can be used to derive B. When the derivation of B is concluded, the assumption A may be *cancelled* so to obtain a derivation of $A \supset B$ which does not depend on the truth of A.

From a proof-theoretical point of view, natural deduction systems present an elegant meta-theory in which derivations are treated as mathematical objects interesting in themselves.

We give here a brief presentation of natural deduction, focusing for concreteness on a system for propositional classical logic. For a formal and exhaustive treatment, standard references are [125, 152].

3.2.1 Rules and derivations

A natural deduction system is described by means of a set of *logical rules*. As an example, we give here a set of logical rules for propositional classical logic (where we consider only the constant $\perp$ and the implication $\supset$ explicitly).

$$\frac{\begin{array}{c}[A \supset \perp] \\ \vdots \\ \perp\end{array}}{A}\ \perp E \qquad \frac{\begin{array}{c}[A] \\ \vdots \\ B\end{array}}{A \supset B}\ \supset I \qquad \frac{A \supset B \quad A}{B}\ \supset E$$

The formulas above the line are called *premises* and the one below the line is the *conclusion*.

A *derivation* is a tree-like structure where each node is a formula and such that if A is the child of a set of nodes $\{A_1, \ldots, A_n\}$, then there exists a rule in the system whose premises are $A_1, \ldots, A_n$ and whose conclusion is A. The leaves of a derivation are called *assumptions* and its root is the *conclusion* of the derivation.

As notation, we write

$$\begin{array}{c} A_1 \ldots A_n \\ \Pi \\ A \end{array}$$

to denote that Π is a derivation whose conclusion is A (we also say that Π is a derivation of A) and whose set of assumptions may contain the formulas $A_1, \ldots, A_n$. In some cases, we will also write

$$\frac{\Pi}{A}\ r$$

to denote a derivation of A obtained by applying a rule r to the conclusion of Π.

Some rules ($\perp E$ and $\supset I$ in the system given above) allow for discharging assumptions, e.g., when we apply the rule $\supset I$ and conclude $A \supset B$ we are allowed

(but not obliged) to discharge possible assumptions of the form A[1]. We denote *discharged assumptions* by using square brackets. We can also use an index to relate the assumption to the rule application that discharges it, like in the following example:

$$\dfrac{\begin{array}{c}[A]^1\\ \Pi\\ B\end{array}}{A \supset B}\ \supset I^1$$

Assumptions that are not discharged are said to be *open*.

Given a system $\mathcal{N}$ of natural deduction, we write $\Gamma \vdash_{\mathcal{N}} A$ to say that there exists a derivation of A in the system $\mathcal{N}$ whose open assumptions are all contained in the set of formulas Γ. A derivation of A in $\mathcal{N}$ where all the assumptions are discharged is a *proof* of A in $\mathcal{N}$ and we then say that A is a *theorem* of $\mathcal{N}$ and write $\vdash_{\mathcal{N}} A$.

3.2.2 Normalization

Natural deduction rules are designed to render the intuitive meaning of the connectives as faithfully as possible. Each rule is related to a logical connective and can be classified either as an *introduction rule* or as an *elimination rule*. The premises of an introduction rule can be seen as the "minimal" conditions necessary to derive the conclusion; conversely the conclusion of an elimination rule can be seen as the "maximal" information that can be restored from the premises.

Up to a few standard exceptions, each connective has one introduction rule and one elimination rule. If the system is well-behaved, each elimination rule is dual to the corresponding introduction rule. In elimination rules, the premise containing the connective is called the *major premise*; the other premises, if any, are called the *minor premises*. A formula occurrence is a *maximum formula* in a derivation when it is both the conclusion of an introduction rule and the major premise of an elimination rule.

Corresponding to the notion of maximum formula is that of *detour*, i.e., a pair of introduction/elimination rules such that the application of the elimination rule occurs immediately below the application of the corresponding introduction rule. Intuitively, a detour represents a redundant step in a derivation (it does not seem to be so clever to introduce something and to eliminate it soon after). A process of normalization will consist basically in removing such redundancies, by means of *contraction rules* that transform a derivation into another derivation with the same open assumptions and conclusion. As an example, we show here the contraction rule for a detour $\supset I / \supset E$.

$$\dfrac{\dfrac{\begin{array}{c}[A]^1\\ \Pi_1\\ B\end{array}}{A \supset B}\ \supset I^1 \qquad \begin{array}{c}\Pi_2\\ A\end{array}}{B}\ \supset E \qquad \rightsquigarrow \qquad \begin{array}{c}\Pi_2\\ A\\ \Pi_1\\ B\end{array}$$

[1] We also remark that the rule is applicable even if such a dischargeable assumption is not present, e.g., $\dfrac{B}{A \supset B}$ is a correct derivation.

Contraction rules focus on a subtree of a larger derivation; the rest of the derivation remains unaltered. We can define a *reduction* relation $\Rightarrow$ built on such contractions, i.e., we say that $\Pi \Rightarrow_1 \Pi'$ if Π' is obtained by Π by applying a contraction to a subderivation of Π and that $\Pi_1 \Rightarrow \Pi_n$ if there exists a *reduction sequence* $\Pi_1 \Rightarrow_1 \Pi_2 \Rightarrow_1 \ldots \Rightarrow_1 \Pi_{n-1} \Rightarrow_1 \Pi_n$.

We say that a derivation Π is in *normal form* if there is no Π' such that $\Pi \Rightarrow \Pi'$, i.e., no contractions can be applied to any subderivation of Π or, which is equivalent, Π does not contain any maximum formulas.

We can distinguish between two forms of normalization: we say that $\Rightarrow$ is *weakly normalizing* if every derivation reduces to a normal form and that it is *strongly normalizing* if there are no infinite reduction sequences. Informally speaking, weak normalization states that if we apply the contractions in a proper way, then we will find a normal form; strong normalization says that we will finally get to a normal form no matter how we choose the contractions.

We do not go into the details of the proof here (see, e.g., [126]) and just conclude that a theorem of (strong) normalization can be proved for the system of propositional classical logic given above. Indeed, the relation $\Rightarrow$ also satisfies the *Church-Rosser property* (see [74]): if $\Pi \Rightarrow \Pi'$ and $\Pi \Rightarrow \Pi''$ then there exists Π''' such that $\Pi' \Rightarrow \Pi'''$ and $\Pi'' \Rightarrow \Pi'''$. As a consequence, we have that each derivation reduces to a unique normal form.

Normalization has a great relevance in proof theory since normal derivations usually satisfy several interesting properties, amongst which we mention the *subformula property*: if Π is a normal derivation of A from a set Γ of assumptions, then every formula B occurring in Π is a subformula of $\Gamma \cup \{A\}$.

Structural properties of normal derivations can also be used to prove interesting corollaries, such as the *consistency* of the deduction system, by means of a purely syntactic argument.

For more details on normalization in natural deduction systems for classical (and intuitionistic) logic, one can consult, e.g., [74,125,126,152]. We will return to these matters in Chapters 4 and 5, when discussing normalization of the systems proposed.

3.3 Natural Deduction for Modal Logics

3.3.1 Towards a Natural Deduction for Modal Logics

Traditionally, modal logics have been presented in terms of Hilbert-style axiom systems, but these are notoriously difficult to use in practice. Unfortunately, natural deduction (or sequent) systems are typically badly suited for non-classical logics: a basic reference on the subject is [61]; a more recent survey on natural deduction methods for modal logics is [92].

The reason of such difficulties is well described in [159]. As we remarked in Section 3.2, a nice feature of natural deduction systems is in the possibility of proving under assumptions. This is clearly illustrated by the rule $\supset I$, which is directly related to the *deduction theorem*:

$$A \models B \qquad \Rightarrow \qquad \models A \supset B\,,$$

where $\Rightarrow$ denotes the implication in the meta-language and $\supset$ the implication in the object language.

When we consider logics whose notion of implication is different from the classical (or intuitionistic) one, it is not immediate to retrieve such a connection between the rule and the theorem. In the case of modal logic, for example, the rule would suggest a *global* deduction theorem like the following:

$$(\forall b \in \mathcal{M}(\mathcal{M} \models b : A) \Rightarrow \forall b \in \mathcal{M}(\mathcal{M} \models b : B)) \Rightarrow \forall b \in \mathcal{M}(\mathcal{M} \models b : A \supset B)\,.$$

But in fact the semantics of $\supset$ in modal logic is weaker and gives rise to the following *local* deduction theorem:

$$\forall b \in \mathcal{M}((\mathcal{M} \models b : A \Rightarrow \mathcal{M} \models b : B) \Rightarrow \mathcal{M} \models b : A \supset B)\,.$$

More on this discussion in [159].

As a consequence, we have that rules for modalities need to take into account such a distinction between global and local assumptions and that it is not trivial to design rules that are proof-theoretically well-behaving. In fact, in the literature we find systems with no explicit modal introduction and elimination rules [30] or with a modal rule like the following one, which is neither an introduction nor an elimination rule:

$$\frac{\Box\Gamma \qquad \begin{array}{c}[\Gamma]\\ \vdots\\ A\end{array}}{\Box A}$$

where $\Box\Gamma$ indicates that each assumption in Γ has $\Box$ as its main logical operator.

In [61], Fitting presents systems for a number of modal propositional logics, treating them by a uniform method. Such natural deduction systems are based on the idea of *subordinate proofs*. The solution described is in the style of that introduced in [59] and consists in adding a second level of subordination, to which we give the name of *strict subordinate proof*. A strict subordinate proof does not represent simply a deduction from an assumption, but we can think of it as an argument in an arbitrary alternative world. Additional rules for managing such strict subordinate proofs are needed and their definition also depends on the different modal logic one wants to represent.

Other methods, still not in the range of labeled deduction, proposed for extending deduction systems to modal (or non-classical in general) logics are in [14, 30, 49, 50, 104, 106, 107, 125, 161, 162]. We remark that such methods are typically referred to some specific logic and not easily generalizable to consider a large class of modal logics. For example, Prawitz [125] provides a rather elegant natural deduction presentation for the logics *S4* and *S5*, but such an approach is not generalizable to other modal logics.

3.3.2 Labeled Natural Deduction for Modal Logics

Around the '90s[2], a new interesting approach, based on the ideas of labeling, has been developed for facing such a problem: rather than modifying the structure of

[2] But its origin can be already found in the semantic approaches to tableaux of [99] and to natural deduction of [60].

a natural deduction proof with extra devices and new kinds of rules, we extend the logical language by using labels. This choice leads to labeled deduction systems [66]. Here we will focus on a particular class of labeled deduction systems, those where labels are used to denote worlds in the corresponding Kripke semantics. In particular, the approaches that we will follow more closely, here and in the rest of the thesis, are those presented in [148, 159].

We consider an extended language consisting of two classes of formulas:

1. *labeled formulas* of the form $b : A$, intuitively expressing that the propositional modal formula A holds at the world $\overline{b}$;
2. *relational formulas* of the form bRb', expressing that $\overline{b'}$ is accessible from $\overline{b}$ according to the relation $\mathcal{R}$ of the model.

(Note that here we have used the *overline* to denote the worlds in the semantics and distinguish them from the labels used in the syntax. The idea is that the label b refers to the world $\overline{b}$.)

As an example, given the modal language defined in Definition 2.1, we can define the corresponding labeled language as follows.

Definition 3.1. *Let L be a denumerable set of labels and R a binary relation symbol over L. If b and c are labels in L and A is a modal formula, then bRc is a* relational well-formed formula *(hereafter simply called* relational formula *or* rwff *for short) and $b : A$ is a* labeled well-formed (modal) formula *(hereafter simply called* labeled formula *or* lwff *for short).*

In the rest of the thesis, when not differently specified, we assume that the variables $b, c, \ldots$ range over labels, the variables $A, B, \ldots$ range over formulas of the (not labeled) logics, φ is an arbitrary rwff or lwff. All variables may be annotated with subscripts or superscripts.

Now we can extend the semantics given for the logic K to the labeled modal language defined above; it is necessary to define an interpretation of labels as worlds explicitly.

Definition 3.2 (Interpretation of labels). *Given a denumerable set of labels L and a Kripke structure $\mathcal{M} = (\mathcal{W}, \mathcal{R}, \mathcal{V})$, an* interpretation *is a function $\lambda : L \to \mathcal{W}$ that maps every label in L to a world in $\mathcal{W}$.*

A semantics for the labeled logic can be given now with respect to a structure and an interpretation. We extend the notion $\models_K$ to deal with labeled and relational formulas as follows.

Definition 3.3. *Given a Kripke structure $\mathcal{M} = (\mathcal{W}, \mathcal{R}, \mathcal{V})$, a denumerable set L of labels and an interpretation λ on them,* truth *for a generic formula φ at a pair $(\mathcal{M}, \lambda)$ is the smallest relation $\models_K$ satisfying:*

$$\begin{aligned} \mathcal{M}, \lambda \models_K bRc \quad &\textit{iff} \quad \lambda(b)\, \mathcal{R}\, \lambda(c) \\ \mathcal{M}, \lambda \models_K b : A \quad &\textit{iff} \quad \mathcal{M}, \lambda(b) \models_K A \end{aligned}$$

Given a set Γ of generic formulas and a generic formula φ:

$$\begin{aligned} \mathcal{M}, \lambda \models_K \Gamma \quad &\textit{iff} \quad \mathcal{M}, \lambda \models_K \varphi \textit{ for all } \varphi \in \Gamma \\ \Gamma \models_K \varphi \quad &\textit{iff} \quad \mathcal{M}, \lambda \models_K \Gamma \textit{ implies } \mathcal{M}, \lambda \models_K \varphi \textit{ for all } \mathcal{M} \textit{ and } \lambda \end{aligned}$$

$$\frac{\begin{array}{c}[b : A \supset \bot] \\ \vdots \\ b' : \bot\end{array}}{b : A}\ \bot E \qquad \frac{\begin{array}{c}[b : A] \\ \vdots \\ b : B\end{array}}{b : A \supset B}\ \supset I \qquad \frac{b : A \supset B \quad b : A}{b : B}\ \supset E$$

$$\frac{\begin{array}{c}[bRb'] \\ \vdots \\ b' : A\end{array}}{b : \Box A}\ \Box I \qquad \frac{b : \Box A \quad bRb'}{b' : A}\ \Box E$$

- In $\Box I$, b' is fresh, i.e., it is different from b and does not occur in any assumption on which $b' : A$ depends other than bRb'.

Fig. 3.1. The rules of $\mathcal{N}(K)$.

The enrichment of the language allows us to give introduction and elimination rules for modal operators that are extremely clean and follow the "spirit" of natural deduction. One can observe that these rules are close to the rules for quantifiers in predicate classical logic [125]. In fact, we express $b : \Box A$ as the metalevel implication $bRb' \Longrightarrow b' : A$ for an arbitrary b' accessible from b:

$$\frac{\begin{array}{c}[bRb'] \\ \vdots \\ b' : A\end{array}}{b : \Box A}\ \Box I \qquad \frac{b : \Box A \quad bRb'}{b' : A}\ \Box E$$

where the rule $\Box I$ has the side condition that b' is different from b and does not occur in any assumption on which $b' : A$ depends other than bRb'.

Analogously, for the operator of possibility, introduction and elimination rules can be defined in the following way:

$$\frac{c : A \quad bRc}{b : \Diamond A}\ \Diamond I \qquad \frac{b : \Diamond A \qquad \begin{array}{c}[c : A] \quad [bRc] \\ \vdots \\ d : B\end{array}}{d : B}\ \Diamond E$$

where $\Diamond E$ has the side condition that c is different from b and d and does not occur in any assumption on which the upper occurrence of $d : B$ depends other than $c : A$ or bRc.

In Figure 3.1, we summarize the rules of a natural deduction system $\mathcal{N}(K)$ for the basic modal logic K. Rules for classical connectives can be modified in a straightforward way in order to treat labeled formulas. Just notice that, in the case of $\bot E$, which is a labeled version of *reductio ad absurdum*, we do not enforce Prawitz's side condition that $A \neq \bot$.[3]

[3] See [159] for a detailed discussion on $\bot E$, which in particular explains how, in order to maintain the duality of modal operators like $\Box$ and $\Diamond$, the rule must allow one to derive $w : A$ from a contradiction $\bot$ at a possibly different world w', and thereby discharge the assumption $w : A \supset \bot$.

Given a labeled natural deduction system, the notions of derivation, theorem and derivability are defined as for standard natural deduction systems (Section 3.2). Thus we write $\Gamma \vdash_{\mathcal{N}(K)} b : A$ to say that there exists a derivation of $b : A$ in the system $\mathcal{N}(K)$ whose open assumptions are all contained in the set of formulas Γ.

For the system $\mathcal{N}(K)$, it is possible to state the following result of soundness and completeness (see, e.g., [159] for a proof).

Theorem 3.4. *Let Γ be a set of labeled formulas and $b : A$ a labeled formula. Then*

$$\Gamma \vdash_{\mathcal{N}(K)} b : A \quad \Leftrightarrow \quad \Gamma \models_{K} b : A\,.$$

The system $\mathcal{N}(K)$ can now be extended in order to capture axiomatic extensions of the modal logic K (see Section 2.2.2) simply by formalizing the details of particular accessibility relations.

Further classifications can be made inside the field of labeled deduction systems. In the following, we describe a few methods presented in the literature for dealing with modal logics by using labels. In particular, we focus on the approaches presented in [159] and [148], to which the systems defined in Chapters 4 and 5 are mainly inspired.

Systems with a proper relational theory

In [159], Viganò introduces a general methodology for presenting and working with a large set of non-classical logics, in particular modal and relevance logics. His natural deduction systems consist of two parts:

- a *base system*, whose rules are in the style of the ones presented above for the logic K, for manipulating labeled formulas;
- a *labeling algebra* for reasoning about the labels, i.e. for manipulating relational formulas.

The base system presents the base logic of a family of propositional non-classical logics. The base and the relational systems are separate and communicate through an *interface* provided by the rules for the modal operators; the intuition behind all this is that for a family or class of related logics we keep the same base system and obtain a presentation of the particular logic we want by "plugging in" the appropriate relational theory.

In [159], labeling algebras are restricted to those that can be formulated in a Horn Theory (see [155]).

Definition 3.5 (Horn relational theory). *A* Horn relational formula *is a closed formula of the form*

$$\forall x_1, ..., x_n((s_1 R t_1 \wedge ... \wedge s_m R t_m) \supset s_0 R t_0),$$

where $m \geq 0$, and the s_i and t_i are terms built from the labels $x_1, ..., x_n$ and constant function symbols. Corresponding to each such formula is a Horn relational rule

$$\frac{s_1 R t_1 \quad ... \quad s_m R t_m}{s_0 R t_0},$$

which has no premises when $m = 0$. *A* Horn relational theory *is a theory generated by a set of such rules.*

The use of a Horn theory gives rise to natural deduction systems that enjoy good normalization properties. To give an idea of the way the whole system works, we give here two relational rules: R_T expresses the reflexivity of the relation $\mathcal{R}$ and R_4 expresses the transitivity of $\mathcal{R}$.

$$\frac{}{bRb}\,R_T \qquad \frac{bRc \qquad cRd}{bRd}\,R_4$$

By adding the relational rule R_4 to the base system for K of Figure 3.1, we obtain a sound and complete system for the logic *K4*. If we add also R_T, then we get a system for *S4*.

A nice feature of Viganò's framework is the strict separation between the base systems and the labeling algebras, which is maintained also when building derivations: in the relational theory we reason only on relational formulas, while in the base system we exploit labeled and relational formulas to infer only labeled formulas, so that a derivation in the base system may depend on a derivation in the relational theory but not viceversa. It follows that derivations of labeled formulas consist of a tree built from the base system, which is decorated with relational subderivations. As an example, we show here a derivation of the axiom 4, expressing the property of transitivity (see Section 2.3.2).

$$\dfrac{\dfrac{\dfrac{[b:\Box A]^1 \qquad \dfrac{[bRc]^2 \quad [cRd]^3}{bRd}\,R_4}{d:A}\,\Box E}{c:\Box A}\,\Box I^3}{b:\Box\Box A}\,\Box I^2}{b:\Box A \supset \Box\Box A}\,\supset I^1$$

The strict separation between the base and the relational systems can be exploited to show that these deduction systems enjoy some "good" structural properties, in particular that derivations *normalize* and that normal derivations satisfy some form of the *subformula property*.

In this thesis, we will propose labeled natural deduction systems closely related to this approach in Section 4.3 for a number of linear tense logics.

Systems without a proper relational theory

In [148], Simpson presents a natural deduction system for intuitionistic modal logics, although the technique used for this purpose can also be used to develop systems for classical modal logics. From our point of view, what really differentiates his systems from systems in [159] is the way of treating relational formulas.

Simpson relegates relational formulas to the role of assumptions in the derivation of labeled logical formulas. This is justified by the fact that relational formulas are not part of the logic and thus that one would not expect that a rule of the system concludes with a relational formula.

This approach aims at keeping the system as simple as possible and at avoiding the explicit introduction of an algebra of terms for the labels. As an example we show now the rules R'_T and R'_4, concerning reflexivity and transitivity respectively, expressed in the Simpson-style: while in [159] premises and conclusions are both relational formulas, here relational formulas appear only as premises or discharged assumptions.

$$\dfrac{\begin{array}{c}[bRb]\\ \vdots\\ b' : A\end{array}}{b' : A}\ R'_T \qquad \dfrac{bRc \quad cRd \quad \begin{array}{c}[bRd]\\ \vdots\\ b' : A\end{array}}{b' : A}\ R'_4$$

The following is a proof of the axiom 4 given in a Simpson-style system. One can compare it with the corresponding one given above (in a Viganò-style system) and observe that in this case we lose the strict separation between base and relational subderivations.

$$\dfrac{\dfrac{\dfrac{[bRc]^2 \quad [cRd]^3 \qquad \dfrac{[b : \Box A]^1 \quad [bRd]^4}{d : A}\ \Box E}{d : A}\ R'^4_4}{c : \Box A}\ \Box I^3}{b : \Box\Box A}\ \Box I^2}{b : \Box A \supset \Box\Box A}\ \supset I^1$$

Also Simpson's systems are proved [148] to enjoy good meta and proof-theoretical properties.

In this thesis, we will propose labeled natural deduction systems following this approach in Sections 4.2, 5.2, and 5.3.

Related approaches

The approaches presented above are the ones that will be followed more closely in the definition of labeled natural deduction systems throughout this thesis. However, it is worth mentioning some related works in the field of labeling for modal and non-classical logics.

In [66], Gabbay describes a general and unifying method for presenting a huge variety of logics. The rules of the deduction systems are designed for manipulating the informations in a sort of logical data-base based on diagrams. As an example, we show here a rule for the elimination of $\Diamond$:

$$\dfrac{b : \Diamond A}{\text{Create a new point } b' \text{ with } b\mathcal{R}b' \text{ and deduce } b' : A}$$

In a sense, as noticed in [159], Gabbay manipulates labels metalinguistically by using expressions coming from a sort of programming language, while, in the approaches of [148,159] such commands are expressed directly by using rules defined in a more natural deduction-style (compare the rule for $\Diamond E$ in the system $\mathcal{N}(K)$ with the one above).

The development of [66] follow different directions. In [26, 27, 143] uniform systems for families of modal and non-classical logics are formalized. Labeled sequent

systems are presented in [112] for modal and in [113] for non-classical logics. Labeling is also used in defining tableaux for some substructural logics in [44], for modal logics with richer languages in [6], for modal and description logics in [45]. For a survey on labeled tableaux see [43], and in particular [81]. Also goal-oriented deduction systems for several non-classical logics [69] have been defined by using labeling.

Labeled deduction is clearly related also to semantic embeddings [117] consisting in translating modal formulas into a first-order classical language, where relational statements are expressed by using binary predicates.

Finally, we mention the so-called hybrid logics [5], in which the enrichment of the language with elements coming from the semantics is not just used as a tool for deduction but becomes part of the logic itself. Namely, the language is extended with propositional symbols (*nominals*) of a new sort, such that each symbol is true at exactly one world. This leads to the definition of more expressive logics, which are usually endowed with a "good" proof theory.

Here we focused on works oriented to modal, and in general non-classical, logics. We postpone the analysis of other related works, specific to temporal logics, to Sections 4.2.6, 4.3.5 and 5.5.

Part II

Labeled Natural Deduction for Temporal Logics

4

Labeled Natural Deduction for Linear Temporal Logics

4.1 Introduction

In Chapter 2 we introduced a number of modal and temporal logics, while in Chapter 3 we presented an approach to deduction for non-classical logics based on labeling and described its application in the case of the most common modal logics. In this chapter, we focus on linear temporal logics and define labeled natural deduction systems for several such logics.

When we introduced labeled natural deduction, we distinguished between two possible approaches: in the first one (followed, e.g., by Simpson in [148]) relational formulas are used only as assumptions in the derivation of (labeled) logical formulas, while in the last one (proposed, e.g., by Viganò in [159]) we have a proper relational sub-system where the inference of a relational formula from premises that are also relational formulas is allowed. In this chapter, we will consider both the approaches and analyze benefits and limitations of each of them.

In Section 4.2, we present a labeled natural deduction system in the style of Simpson's approach for the basic tense logic Kt. Then we show how to extend modularly such a system in order to capture the linear tense logic Kl and some of its variants, namely Kl with bounded or unbounded time, Kl with dense time and Kl with discrete time. We show that all such systems are sound and complete with respect to the corresponding semantics. Finally, we describe a further extension leading to a system for the logic $LTL\_$, i.e. the until-free fragment of LTL. We remark that our original contribution in the context of Section 4.2 is mainly in giving a uniform and modular presentation of systems for a large class of linear temporal logics. In fact, the system for Kt is a trivial extension of the ones presented for the modal logic K, the systems for Kl and its variants are a specialization of the ones for axiomatic extensions of modal logics (as described, e.g., in [148]) and the system for $\mathcal{N}(LTL\_)$ is very close to the one described in [103] for the same logic.

In Section 4.3, we use [159], where labeled natural deduction systems are defined for several non-classical logics, as a starting point. As described in Section 3.3.2, systems in [159] are composed of a base system for inferring labeled logical formulas and of a sub-system, consisting of Horn rules, for reasoning on relational formulas. Such a restriction of considering only Horn rules in the relational sub-system allows for well-behaving, from a proof-theoretical point of view, natural

deduction systems, based on the strict separation between the labeled and the relational systems. If we consider linear temporal logics, even a basic logic like Kl, Horn rules do not suffice, because we need to capture a condition of linearity (see Section 2.3 for details), which requires the possibility of expressing at least a disjunction. Thus we need to consider more powerful relational systems. Here we define a sound and complete system for Kl where the relational language is extended to be a full first-order relational language and study the consequences of such an extension. We see that the strict separation between the base and the relational system, typical of the systems in [159], is lost, but also that the resulting systems still enjoy some good structural properties that can be exploited in order to prove some form of normalization. Furthermore the extension of the relational language allows us to capture the axiomatic extensions of Kl presented in Section 2.3.2 in a clean and modular way. The possibility of further extending the system in order to reason on LTL or $LTL\_$ is also discussed. Part of the material of Section 4.3 has been presented in [160].

The systems in Sections 4.2 and 4.3 do not consider the operators since and until. In fact, such operators are quite complex to be treated from a proof-theoretical point of view, e.g., if we are interested in defining a normalization procedure for our systems. In Section 4.4, we propose a solution for the treatment of until (we focus on the future fragment but an extension to the past should not be problematic), consisting in replacing it by a new unary operator, called *history*. As a concrete example, we define a logic LTL_{∇}, obtained by replacing until with history, and showing that the two logics are equally expressive, i.e., that it is possible to define a translation from LTL into LTL_{∇} and, viceversa, a translation from LTL_{∇} into LTL such that the notions of semantical consequence are preserved. Then we define a labeled natural deduction system for LTL_{∇}, where the interesting point is that the rules for the introduction and the elimination of the new operator are very simple and absolutely in the spirit of natural deduction; indeed, they present the same pattern of the rules for the other modal (temporal) operators. The equivalence between the two logics makes the system useful also for reasoning on LTL. Furthermore the approach presented is fully general and can be easily adapted to other linear and branching temporal logics with until.

4.2 Systems for linear temporal logics

In this section, we present sound and complete labeled natural deduction systems in the style of Section 3.3.2 for linear temporal logics.

The structure of this section is the following:

- in Section 4.2.1, we present a labeled natural deduction system for Kt;
- in Section 4.2.2, we extend it to capture the linear tense logic Kl;
- in Section 4.2.3, we consider extensions of the system for some variants of Kl;
- in Section 4.2.4, we give a system for the until-free version of LTL;
- in Section 4.2.5, we briefly discuss normalization matters;
- in Section 4.2.6, we summarize and compare with related work.

4.2.1 A system for *Kt*

The minimal Priorean tense logic Kt presented in Section 2.3.1 is no more than the basic modal logic K (Section 2.2) with a symmetrical modal (temporal) operator directed towards the past. Thus a trivial extension of the labeled base system $\mathcal{N}(K)$ (Section 3.3.2) of [148, 159] will work. As is standard for temporal logics, we use G and F for $\Box$ and $\Diamond$ and denote with $<$ the relational symbol used in the syntax (corresponding to the relation $\prec$ of the semantics; see Section 2.3.1).

A labeled version of *Kt*

As we did in Section 3.3.2 for modal logics, we need to formalize the extension of the language and the adaptations to the semantics required by the labeled deduction setting.

Definition 4.1. *Let L be a denumerable set of labels and $<$ a binary relation symbol over L. If b and c are labels in L and A is a tense formula, then $b < c$ is a* relational well-formed formula *(or relational formula, or rwff for short) and $b : A$ is a* labeled well-formed (tense) formula *(or labeled formula, or lwff for short).*

We remark that the terms labeled formula (or lwff) and relational formula (or rwff) will be often redefined in the thesis and thus will be used with different meanings in the context of different sections. Since each section is typically devoted to a specific labeled system, and consequently deals with a specific labeled language, we believe that this will not generate any confusion.

Definition 4.2. *Given a denumerable set of labels L and a temporal structure $\mathcal{M} = (\mathcal{W}, \prec, \mathcal{V})$, an* interpretation *is a function $\lambda : L \to \mathcal{W}$ that maps every label in L to a time-instant in $\mathcal{W}$.*

Definition 4.3. *Given a temporal structure $\mathcal{M} = (\mathcal{W}, \prec, \mathcal{V})$, a denumerable set L of labels and an interpretation λ on them,* truth *for a labeled or relational formula φ at a pair $(\mathcal{M}, \lambda)$ is the smallest relation $\models_{Kt}$ satisfying:*

$$\begin{array}{rll} \mathcal{M}, \lambda \models_{Kt} b < c & \textit{iff} & \lambda(b) \prec \lambda(c) \\ \mathcal{M}, \lambda \models_{Kt} b : A & \textit{iff} & \mathcal{M}, \lambda(b) \models_{Kt} A \end{array}$$

Given a set Γ of generic formulas and a generic formula φ:

$$\begin{array}{rll} \mathcal{M}, \lambda \models_{Kt} \Gamma & \textit{iff} & \mathcal{M}, \lambda \models_{Kt} \varphi \textit{ for all } \varphi \in \Gamma \\ \Gamma \models_{Kt} \varphi & \textit{iff} & \mathcal{M}, \lambda \models_{Kt} \Gamma \textit{ implies } \mathcal{M}, \lambda \models_{Kt} \varphi \textit{ for all } \mathcal{M} \textit{ and } \lambda \end{array}$$

The system $\mathcal{N}(Kt)$

With respect to $\mathcal{N}(K)$, the extension consists in introducing a pair of introduction/elimination rules for the operator H to the base system $\mathcal{N}(K)$; such rules are just the symmetrical version of $\Box I$ and $\Box E$.

$$\frac{\begin{array}{c}[b_1 : A \supset \bot] \\ \vdots \\ b_2 : \bot\end{array}}{b_1 : A}\ \bot E \qquad \frac{\begin{array}{c}[b : A] \\ \vdots \\ b : B\end{array}}{b : A \supset B}\ \supset I \qquad \frac{b : A \supset B \quad b : A}{b : B}\ \supset E$$

$$\frac{\begin{array}{c}[b_1 < b_2] \\ \vdots \\ b_2 : A\end{array}}{b_1 : \mathsf{G}A}\ \mathsf{G}I \qquad \frac{b_1 : \mathsf{G}A \quad b_1 < b_2}{b_2 : A}\ \mathsf{G}E \qquad \frac{\begin{array}{c}[b_1 < b_2] \\ \vdots \\ b_1 : A\end{array}}{b_2 : \mathsf{H}A}\ \mathsf{H}I \qquad \frac{b_2 : \mathsf{H}A \quad b_1 < b_2}{b_1 : A}\ \mathsf{H}E$$

- In $\mathsf{G}I$, b_2 is *fresh*, i.e., it is different from b_1 and does not occur in any assumption on which $b_2 : A$ depends other than the discharged assumption $b_1 < b_2$.
- In $\mathsf{H}I$, b_1 is *fresh*, i.e., it is different from b_2 and does not occur in any assumption on which $b_1 : A$ depends other than the discharged assumption $b_1 < b_2$.

Fig. 4.1. The rules of $\mathcal{N}(Kt)$.

The set of rules of the system $\mathcal{N}(Kt)$, for which the notion of derivability $\vdash_{\mathcal{N}(Kt)}$ can be defined as usual, is given in Figure 4.1. The notions of *derivation* and *theorem*, here and for the other systems of this section, are the standard ones (see Section 3.2).

We will give concrete examples of derivations in the following. For simplicity, we will sometimes employ the rules for conjunction $\wedge$ and disjunction $\vee$, which are derived from the basic propositional rules as is standard, as well as other derived rules such as those for F and P (see Figure 4.2).

As examples, we show how to derive the rules $\wedge I$, $\mathsf{F}I$ and $\mathsf{F}E$:

$$\frac{b : A \quad b : B}{b : A \wedge B}\ \wedge I \quad \text{abbreviates} \quad \frac{\dfrac{\dfrac{[b : A \supset (B \supset \bot)]^1 \quad b : A}{b : B \supset \bot}\ \supset E \qquad b : B}{b : \bot}\ \supset E}{b : (A \supset (B \supset \bot)) \supset \bot}\ \supset I^1$$

The rule

$$\frac{c : A \qquad b < c}{b : \mathsf{F}A}\ \mathsf{F}I$$

can be derived as follows:

$$\dfrac{\dfrac{\dfrac{[b : \mathsf{G}(A \supset \bot)]^1 \qquad b < c}{c : A \supset \bot}\ \mathsf{G}E \qquad c : A}{c : \bot}\ \supset E}{b : \bot}\ \bot E}{b : \mathsf{G}(A \supset \bot) \supset \bot}\ \supset I^1$$

while an application of $\mathsf{F}E$

$$\frac{c : A \quad b < c}{b : \mathsf{F}A}\ \mathsf{F}I \qquad \frac{b : \mathsf{F}A \qquad \begin{array}{c}[c : A][b < c]\\ \vdots\\ d : B\end{array}}{d : B}\ \mathsf{F}E$$

$$\frac{c : A \quad c < b}{b : \mathsf{P}A}\ \mathsf{P}I \qquad \frac{b : \mathsf{P}A \qquad \begin{array}{c}[c : A][c < b]\\ \vdots\\ d : B\end{array}}{d : B}\ \mathsf{P}E$$

$$\frac{b : A \quad b : B}{b : A \wedge B}\ \wedge I \qquad \frac{b : A \wedge B}{b : A}\ \wedge E_1 \qquad \frac{b : A \wedge B}{b : B}\ \wedge E_2$$

$$\frac{b : A}{b : A \vee B}\ \vee I_1 \qquad \frac{b : B}{b : A \vee B}\ \vee I_2 \qquad \frac{b' : B \vee C \quad \begin{array}{c}[b' : B]\\ \vdots\\ b : A\end{array} \quad \begin{array}{c}[b' : C]\\ \vdots\\ b : A\end{array}}{b : A}\ \vee E$$

- In $\mathsf{F}E$, c is different from b and d, and does not occur in any assumption on which the upper occurrence of $d : B$ depends other than $c : A$ or $b < c$ ($c < b$).
- In $\mathsf{P}E$, c is different from b and d, and does not occur in any assumption on which the upper occurrence of $d : B$ depends other than $c : A$ or $c < b$.

Fig. 4.2. Some derived rules.

$$\frac{\begin{array}{c}[c : A]\,[b < c]\\ \Pi\end{array}}{\begin{array}{c}b : \mathsf{F}A \qquad d : B\\ \hline d : B\end{array}\ \mathsf{F}E}$$

can be replaced by the following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[c : A]^3\,[b < c]^2}{\Pi}}{\dfrac{[d : B \supset \bot]^1 \qquad d : B}{d : \bot}\ \supset E}}{c : \bot}\ \bot E}{c : A \supset \bot}\ \supset I^3}{b : \mathsf{G}(A \supset \bot)}\ \mathsf{G}I^2 \qquad b : \mathsf{G}(A \supset \bot) \supset \bot}{b : \bot}\ \supset E}{d : B}\ \bot E^1$$

Soundness

Theorem 4.4. *Let Γ be a set of labeled and relational tense formulas and $b : A$ a labeled tense formula. Then*

$$\Gamma \vdash_{\mathcal{N}(Kt)} b : A \quad \Rightarrow \quad \Gamma \models_{Kt} b : A\,.$$

Proof. The proof proceeds by induction on the structure of the derivation of $b : A$. The base case is when $b : A \in \Gamma$ and is trivial. There is one step case for every rule. Soundness of the rules for logical connectives can be proved by using standard arguments, while the soundness of the rules of introduction/elimination of temporal operators and quantifiers follows like in other labeled systems for non-classical logics (see, e.g., [148, 159]). We show only the cases of introduction and elimination of G; the cases concerning H can be proved analogously.

Consider an application of the rule $\mathsf{G}I$

$$\frac{\begin{array}{c}[b_1 < b_2]\\ \Pi\\ b_2 : A\end{array}}{b_1 : \mathsf{G}A}\ \mathsf{G}I$$

where Π is a proof of $b_2 : A$ from hypotheses in Γ', with b_2 fresh and with $\Gamma' = \Gamma \cup \{b_1 < b_2\}$. By the induction hypothesis, for all interpretations λ, if $\mathcal{M}, \lambda \models_{Kt} \Gamma'$ then $\mathcal{M}, \lambda \models_{Kt} b_2 : A$. We let λ be any interpretation such that $\mathcal{M}, \lambda \models_{Kt} \Gamma$, and show that $\mathcal{M}, \lambda \models_{Kt} b_1 : \mathsf{G}A$. Let $\lambda(b_1) = n$. Now let us consider a generic successor $n + k$ of n for some $k > 0$. Since λ can be trivially extended to another interpretation (still called λ for simplicity) by setting $\lambda(b_2) = n + k$, the induction hypothesis yields $\mathcal{M}, \lambda \models_{Kt} b_2 : A$, i.e. $\mathcal{M}, n + k \models_{Kt} A$. Given that k is arbitrary we can conclude $\mathcal{M}, \lambda \models_{Kt} b_1 : \mathsf{G}A$.

Consider the case in which the last rule applied is $\mathsf{G}E$:

$$\frac{\begin{array}{cc}\Pi & \\ b_1 : \mathsf{G}A & b_1 < b_2\end{array}}{b_2 : A}\ \mathsf{G}E$$

where Π is a proof of $b_1 : \mathsf{G}A$ from hypotheses in Γ_1, with $\Gamma = \Gamma_1 \cup \{b_1 < b_2\}$ for some set Γ_1 of formulas. By applying the induction hypothesis on Π, we have:

$$\Gamma_1 \models_{Kt} b_1 : \mathsf{G}A\ .$$

From $\Gamma \supset \Gamma_1$, we deduce (by the induction hypothesis) $\mathcal{M}, \lambda \models_{Kt} b_1 : \mathsf{G}A$. Furthermore $\mathcal{M}, \lambda \models \Gamma$ entails $\mathcal{M}, \lambda \models_{Kt} b_1 < b_2$ and thus $\mathcal{M}, \lambda(b_2) \models_{Kt} A$, i.e., by Definition 4.3, $\mathcal{M}, \lambda \models_{Kt} b_2 : A$.

□

Completeness

Theorem 4.5. *Let Γ be a set of labeled tense formulas and $b : A$ a labeled tense formula. Then*

$$\Gamma \models_{Kt} b : A \quad \Rightarrow \quad \Gamma \vdash_{\mathcal{N}(Kt)} b : A\,.$$

Proof. We show that the system $\mathcal{N}(Kt)$ is complete with respect to the semantics of Kt (Definition 2.7) by showing that every axiom and rule of inference in the axiomatization $\mathcal{H}(Kt)$ is provable in $\mathcal{N}(Kt)$.

Firstly, we show by induction on the length of $\mathcal{H}(Kt)$ derivations that it is possible to derive the rules of inference of $\mathcal{H}(Kt)$ in $\mathcal{N}(Kt)$.

(MP)

$$\text{If } \vdash_{Kt} A \text{ and } \vdash_{Kt} A \supset B\,, \text{ then } \vdash_{Kt} B\,.$$

By induction hypothesis, given an arbitrary label b, we have in $\mathcal{N}(Kt)$ a derivation Π_1 of $b : A$ and a derivation Π_2 of $b : A \supset B$. By applying $\supset E$, we obtain:

$$\dfrac{\begin{array}{cc}\Pi_2 & \Pi_1\\ b : A \supset B & b : A\end{array}}{b : B}\ \supset E$$

(Nec_{G})

$$\text{If } \vdash_{Kt} A\,, \text{ then } \vdash_{Kt} \mathsf{G}A\,.$$

Given an arbitrary label b', by induction hypothesis we have a proof Π of $b' : A$ in $\mathcal{N}(Kt)$. Then we can use the arbitrariness of b' and build a proof of $b : \mathsf{G}A$ as follows:

$$\dfrac{\begin{array}{c}[b < b']\\ \Pi\\ b' : A\end{array}}{b : \mathsf{G}A}\ \mathsf{G}I$$

The case of the rule Nec_{H} can be treated in a symmetrical way.

In the following, we give derivations of the axioms K_{G} and GP. We omit the derivations for K_{H} and HF, which are very similar.

(K_{G})

$$\dfrac{\dfrac{\dfrac{[b : \mathsf{G}(A \supset B)]^1 \quad [b < c]^3}{c : A \supset B}\ \mathsf{G}E \quad \dfrac{[b : \mathsf{G}A]^2 \quad [b < c]^3}{c : A}\ \mathsf{G}E}{c : B}\ \supset E}{b : \mathsf{G}B}\ \mathsf{G}I^3}{b : \mathsf{G}A \supset \mathsf{G}B}\ \supset I^2}{b : \mathsf{G}(A \supset B) \supset (\mathsf{G}A \supset \mathsf{G}B)}\ \supset I^1$$

(GP)

$$\dfrac{\dfrac{\dfrac{[b : A]^1 \quad [b < c]^2}{c : \mathsf{P}A}\ \mathsf{P}I}{b : \mathsf{GP}A}\ \mathsf{G}I^2}{b : A \supset \mathsf{GP}A}\ \supset I^1$$

□

4.2.2 A system for *Kl*

When moving from Kt to Kl, we restrict to consider models where the flow of time is irreflexive, transitive and connected (or linear). With regard to irreflexivity, it is well known (see, e.g., [75]) that considering or not such a property does not modify the set of valid formulas and thus, in terms of rules, we can avoid considering it.[1]

[1] We will return to this point in Section 4.3, where, by considering natural deduction systems endowed with a proper first-order relational subsystem, we will be able to capture also irreflexivity.

$$\frac{b_1 < b_2 \quad b_2 < b_3 \quad \begin{array}{c}[b_1 < b_3]\\ \vdots \\ b : A\end{array}}{b : A}\ trans < \qquad \frac{b_1 : B \quad \begin{array}{c}[b_2 : B]\\ \vdots \\ b : A\end{array} \quad \begin{array}{c}[b_1 < b_2]\\ \vdots \\ b : A\end{array} \quad \begin{array}{c}[b_2 < b_1]\\ \vdots \\ b : A\end{array}}{b : A}\ conn <$$

Fig. 4.3. The rules for *transitivity* and *connectedness*.

By enriching the system $\mathcal{N}(Kt)$ of Section 4.2.1 with two further rules, one for transitivity and one for connectedness, we get a system that is sound and complete with respect to Kl.

We use the same labeled language defined for Kt (Section 4.2.1). The definition of interpretation in the case of Kl and the extension of $\models_{Kl}$ to labeled and relational formulas can be easily adapted from Section 4.2.1: just replace *temporal structure* by Kl-structure; we omit the details.

The system $\mathcal{N}(Kl)$

In Figure 4.3, we present the rules *trans* $<$ and *conn* $<$, which capture transitivity and connectedness, respectively. We define $\mathcal{N}(Kl)$ as the system containing the set of rules in $\mathcal{N}(Kt)$ plus *trans* $<$ and *conn* $<$.

With regard to *conn* $<$, we remark that, since we do not treat equality between labels explicitly in our relational language[2], we express it by means of equality of the sets of formulas holding in the labels. Thus, given two instants b_1 and b_2, the rule can be read as stating that one of the following must hold:

1. b_1 and b_2 coincide, and then if a formula B holds in b_1, it must also hold in b_2;
2. b_1 precedes b_2;
3. b_2 precedes b_1.

We also notice that, in the case in which the relation $<$ was assumed to be reflexive (denoted $\leqslant$), the rule *conn* $<$ could be simplified as follows[3]:

$$\frac{\begin{array}{c}[b_1 \leqslant b_2]\\ \vdots \\ b : A\end{array} \quad \begin{array}{c}[b_2 \leqslant b_1]\\ \vdots \\ b : A\end{array}}{b : A}\ conn \leqslant$$

[2] We will consider equality explicitly, by allowing also relational formulas of the form $b = c$, in Section 4.3, where we will investigate benefits and disadvantages of having a much richer relational system.

[3] A system for the "reflexive" version of Kl could then be obtained by simply replacing $<$ by $\leqslant$ in each rule of $\mathcal{N}(Kl)$, by using *conn* $\leqslant$ instead of *conn* $<$ and by adding the following rule for reflexivity:

$$\frac{\begin{array}{c}[b_1 \leqslant b_1]\\ \vdots \\ b : A\end{array}}{b : A}\ refl \leqslant$$

Soundness

Theorem 4.6. *Let Γ be a set of labeled and relational tense formulas and $b : A$ a labeled tense formula. Then*

$$\Gamma \vdash_{\mathcal{N}(Kl)} b : A \quad \Rightarrow \quad \Gamma \models_{Kl} b : A \,.$$

Proof. We extend the proof of Theorem 4.4 by considering the cases regarding the rules *trans* $<$ and *conn* $<$.

(*trans* $<$) Consider the case in which the last rule applied is *trans* $<$:

$$\dfrac{\begin{array}{ccc} & & [b_1 < b_3] \\ & & \Pi \\ b_1 < b_2 & b_2 < b_3 & b : A \end{array}}{b : A}\ \mathit{trans} <$$

where Π is a proof of $b : A$ from hypotheses in Γ_2, with $\Gamma = \Gamma_1 \cup \{b_1 < b_2, b_2 < b_3\}$ and $\Gamma_2 = \Gamma_1 \cup \{b_1 < b_3\}$ for some set Γ_1 of formulas. By applying the induction hypothesis on Π, we have:

$$\Gamma_2 \models_{Kl} b : A \,.$$

We proceed by considering a generic linear temporal structure $\mathcal{M} = (\mathcal{W}, \prec, \mathcal{V})$ and a generic interpretation λ on it such that $\mathcal{M}, \lambda \models_{Kl} \Gamma$ and showing that this entails

$$\mathcal{M}, \lambda \models_{Kl} b : A \,.$$

Let $\lambda(b_1) = w$ for some $w \in \mathcal{W}$. Then, by $\mathcal{M}, \lambda \models_{Kl} \Gamma$, we infer $\lambda(b_1) \prec \lambda(b_2)$ and $\lambda(b_2) \prec \lambda(b_3)$ and, by the definition of a linear temporal frame, we have $\lambda(b_1) \prec \lambda(b_3)$, i.e. $\mathcal{M}, \lambda \models_{Kl} b_1 < b_3$. This implies $\mathcal{M}, \lambda \models_{Kl} \Gamma_2$ and thus $\mathcal{M}, \lambda \models_{Kl} b : A$ by the induction hypothesis.

(*conn* $<$) Now consider the case in which the last rule applied is *conn* $<$:

$$\dfrac{\begin{array}{cccc} [b_2 : B] & & [b_1 < b_2] & [b_2 < b_1] \\ \Pi & & \Pi' & \Pi'' \\ b_1 : B & b : A & b : A & b : A \end{array}}{b : A}\ \mathit{conn} <$$

where Π is a proof of $b : A$ from hypotheses in Γ_2, Π' is a proof of $b : A$ from hypotheses in Γ_2' and Π'' is a proof of $b : A$ from hypotheses in Γ_2'', where $\Gamma = \Gamma_1 \cup \{b_1 : B\}$, $\Gamma_2 = \Gamma_1 \cup \{b_2 : B\}$, $\Gamma_2' = \Gamma_1 \cup \{b_1 < b_2\}$ and $\Gamma_2'' = \Gamma_1 \cup \{b_2 < b_1\}$ for some set Γ_1 of formulas. By applying the induction hypothesis on Π, Π' and Π'' we have (respectively):

$$\Gamma_2 \models_{Kl} b : A \quad , \quad \Gamma_2' \models_{Kl} b : A \quad , \quad \Gamma_2'' \models_{Kl} b : A \quad .$$

We proceed by considering a generic linear temporal structure $\mathcal{M} = (\mathcal{W}, \prec, \mathcal{V})$ and a generic interpretation λ on it such that $\mathcal{M}, \lambda \models_{Kl} \Gamma$ and showing that this entails

$$\mathcal{M}, \lambda \models_{Kl} b : A \,.$$

First notice that $\mathcal{M}, \lambda \models_{Kl} \Gamma$, implies $\mathcal{M}, \lambda \models_{Kl} b_1 : B$. By the condition of linearity on linear temporal models, we have that one of the following must hold:

1. $\lambda(b_1) = \lambda(b_2)$: and then we have that $\mathcal{M}, \lambda \models_{Kl} b_1 : B$ implies $\mathcal{M}, \lambda \models_{Kl} b_2 : B$, from which we infer $\mathcal{M}, \lambda \models_{Kl} \Gamma_2$ and thus $\mathcal{M}, \lambda \models_{Kl} b : A$ by the induction hypothesis;
2. $\lambda(b_1) \prec \lambda(b_2)$: and then we have that $\mathcal{M}, \lambda \models_{Kl} b_1 < b_2$, from which we infer $\mathcal{M}, \lambda \models_{Kl} \Gamma_2'$ and thus $\mathcal{M}, \lambda \models_{Kl} b : A$ by the induction hypothesis;
3. $\lambda(b_2) \prec \lambda(b_1)$: and then, symmetrically, we have that $\mathcal{M}, \lambda \models_{Kl} b_2 < b_1$, from which we infer $\mathcal{M}, \lambda \models_{Kl} \Gamma_2''$ and thus $\mathcal{M}, \lambda \models_{Kl} b : A$ by the induction hypothesis.

□

Completeness

Theorem 4.7. *Let Γ be a set of labeled tense formulas and $b : A$ a labeled tense formula. Then*

$$\Gamma \models_{Kl} b : A \quad \Rightarrow \quad \Gamma \vdash_{\mathcal{N}(Kl)} b : A\,.$$

Proof. We give a derivation of the axioms $TRANS_R$ and $CONN_R$. The proofs for $TRANS_L$ and $CONN_L$ are completely symmetrical and we omit them.

($TRANS_R$)

$$\dfrac{\dfrac{\dfrac{[b < c]^2 \quad [c < d]^3 \quad \dfrac{[b : \mathsf{G}A]^1 \quad [b < d]^4}{d : A}\,\mathsf{G}E}{d : A}\,trans\,{<^4}}{c : \mathsf{G}A}\,\mathsf{G}I^3}{b : \mathsf{G}\mathsf{G}A}\,\mathsf{G}I^2}{b : \mathsf{G}A \supset \mathsf{G}\mathsf{G}A}\,{\supset}I^1$$

($CONN_R$)

We slightly simplify the derivation here, by allowing the application of $\wedge E$ on a premise consisting of three conjuncts.

$$\dfrac{\dfrac{\dfrac{[b : \mathsf{H}A \wedge A \wedge \mathsf{G}A]^1}{b : A}\,{\wedge}E \quad \begin{matrix}\Pi_1 & \Pi_2\\ [d : A]^4 & d : A \quad d : A\end{matrix}}{d : A}\,conn\,{<^4}}{c : \mathsf{H}A}\,\mathsf{H}I^3}{b : \mathsf{G}\mathsf{H}A}\,\mathsf{G}I^2}{b : \mathsf{H}A \wedge A \wedge \mathsf{G}A \supset \mathsf{G}\mathsf{H}A}\,{\supset}I^1$$

where Π_1 is

$$\dfrac{\dfrac{[b : \mathsf{H}A \wedge A \wedge \mathsf{G}A]^1}{b : \mathsf{G}A}\,{\wedge}E \quad [b < d]^4}{d : A}\,\mathsf{G}E$$

and Π_2 is

$$\dfrac{\dfrac{[b : \mathsf{H}A \wedge A \wedge \mathsf{G}A]^1}{b : \mathsf{H}A}\,{\wedge}E \quad [d < b]^4}{d : A}\,\mathsf{H}E$$

□

4.2.3 Systems for axiomatic extensions of *Kl*

Here we consider extensions of the system $\mathcal{N}(Kl)$ aiming at capturing some of the axiomatic extensions presented in Section 2.3.2.

Kl with unbounded time

One of the possible extensions of Kl consists in requiring that the underlying flow of time is unbounded, i.e., every point has a successor and/or a predecessor. We can express such properties by adding the rules ser_R and ser_L below.

$$\frac{\begin{array}{c}[b_1 < b_2]\\ \vdots \\ b : A\end{array}}{b : A}\ ser_R \qquad \frac{\begin{array}{c}[b_1 < b_2]\\ \vdots \\ b : A\end{array}}{b : A}\ ser_L \quad ,$$

where we require that b_2 is fresh in ser_R (i.e., it is different from b_1 and does not occur in any assumption on which $b : A$ depends other than the discharged assumptions $b_1 < b_2$) and that b_1 is fresh in ser_L (i.e., it is different from b_2 and does not occur in any assumption on which $b : A$ depends other than the discharged assumptions $b_1 < b_2$).

A derivation of the axiom SER_R, using the rules of $\mathcal{N}(Kl)$ and ser_R, is the following.

$$\dfrac{\dfrac{\dfrac{\dfrac{[c : A \wedge \neg A]^2}{c : A}\ \wedge E \quad \dfrac{[c : A \wedge \neg A]^2}{c : \neg A}\ \wedge E}{\dfrac{c : \bot}{c : A \vee \neg A}\ \bot E^2}\ \neg E \qquad [b < c]^1}{b : \mathsf{F}(A \vee \neg A)}\ \mathsf{F}I}{b : \mathsf{F}(A \vee \neg A)}\ ser_R^1$$

In a completely symmetrical way, one can obtain a derivation of SER_L, using the rules of $\mathcal{N}(Kl)$ and ser_L.

Kl with first/final point

Conversely, we can require that the flow of time is bounded by a first and/or a final point. It is not trivial to express such a property in our setting, as long as we are interested in keeping the good structural properties of our derivations, i.e., in particular, limiting the introduction/elimination of the operators to the rules devoted to that.

A solution could consist in the use of two special labels as constants, which intuitively denote in the syntax the first and the final point of the flow of time. What we miss is the possibility of deriving a contradiction at a relational level.[4] A rule like the following[5]

[4] In fact, in Section 4.3, we will show that the usage of a first-order relational language makes it simple to capture this extension of the logic Kl.

[5] Plus clearly some other rules modeling the use of the constants.

$$\frac{b_1 < b_2 \quad b_2 < b_1}{b_1 : \bot}$$

would do if we just aim at obtaining soundness and completeness but we are aware of the fact that we lose some of the good properties of the system.

Kl with dense time

A flow of time is dense if between any two points we can find a third point. The following rule captures such a property:

$$\frac{\begin{array}{c}[b_1 < b'] \quad [b' < b_2] \\ \vdots \\ b_1 < b_2 \qquad b : A\end{array}}{b : A}\ dens <$$

where we require that b' is fresh, i.e., it is different from b_1 and b_2 and does not occur in any assumption on which $b : A$ depends other than the discharged assumptions $b_1 < b'$ and $b' < b_2$.

We give here a derivation of the axiom $DENS_R$, using the rules of $\mathcal{N}(Kl)$ and $dens <$. We omit the derivation of the axiom $DENS_L$, which is symmetrical.

$$\dfrac{\dfrac{[b : \mathsf{F}A]^1 \qquad \dfrac{\dfrac{[b < c]^2 \qquad \dfrac{\dfrac{[c : A]^2 \quad [d < c]^3}{d : \mathsf{F}A}\,\mathsf{F}I \qquad [b < d]^3}{b : \mathsf{FF}A}\,\mathsf{F}I}{b : \mathsf{FF}A}\,dens <^3}{b : \mathsf{F}A}\,\mathsf{F}E^2}{b : \mathsf{F}A \supset \mathsf{F}A}\,\supset I^1$$

Kl with discrete time

Finally, we can express discreteness both towards the future and towards the past (see Section 2.3.2 for details).

A solution for capturing such a property consists in introducing into the system a new relational symbol expressing the relation of being the *immediate successor* of another point. We use the symbol $\lhd$ for such a relation and thus extend the relational language by considering as rwffs also formulas of the form $b \lhd c$. We extend the semantics of labeled Kl with the following clause:

$$\mathcal{M}, \lambda \models_{Kl} b \lhd c \quad \text{iff} \quad \lambda(b) \prec \lambda(c) \text{ and there is no } x \in \mathcal{W} \text{ s.t. } \lambda(b) \prec x \prec \lambda(c)\,.$$

Now we introduce some rules for modeling its properties. First of all, we require the relation $\lhd$ to be functional, i.e., if both b_2 and b_3 are the immediate successors of b_1, then b_2 and b_3 must coincide. We define two symmetrical rules, expressing functionality (or linearity, since it prevents in some way the formation of a branch) towards the future and towards the past, respectively.

$$\frac{b_1 \lhd b_2 \quad b_1 \lhd b_3 \quad b_2 : A}{b_3 : A}\ lin\lhd_R \qquad \frac{b_1 \lhd b_2 \quad b_3 \lhd b_2 \quad b_1 : A}{b_3 : A}\ lin\lhd_L$$

Then we need to specify the interaction between the relations $\lhd$ and $<$. First, we have that $\lhd$ is contained in $<$, i.e. if $b_1 \lhd b_2$ holds, then also $b_1 < b_2$ must hold.

$$\frac{\begin{array}{cc} & [b_1 < b_2] \\ & \vdots \\ b_1 \lhd b_2 & b : A \end{array}}{b : A}\ \mathit{base} <$$

We can also state that if a point has a successor (predecessor), then it must also have an immediate successor (predecessor).[6]

$$\frac{\begin{array}{cc} & [b_1 \lhd b'] \\ & \vdots \\ b_1 < b_2 & b : A \end{array}}{b : A}\ \mathit{discr} <_R \qquad \frac{\begin{array}{cc} & [b' \lhd b_2] \\ & \vdots \\ b_1 < b_2 & b : A \end{array}}{b : A}\ \mathit{discr} <_L$$

In both the rules we require that b' is fresh, i.e., it is different from b_1 and b_2 and does not occur in any assumption on which $b : A$ depends other than the discharged assumption.

Finally, we need a rule that allows us to split a statement of the form $b_1 < b_2$ into two cases: either b_2 is the immediate successor of b_1 or b_2 is a successor of the immediate successor of b_1.

$$\frac{\begin{array}{ccc} [b_1 \lhd b_2] & [b_1 \lhd b'] & [b' < b_2] \\ \vdots & & \vdots \\ b_1 < b_2 \quad b : A & & b : A \end{array}}{b : A}\ \mathit{split} <_R ,$$

where we require that b' is fresh, i.e., it is different from b_1 and b_2 and does not occur in any assumption on which $b : A$ depends other than the discharged assumptions $b_1 \lhd b'$ and $b' < b_2$.

Clearly, the same argument holds if we reason (symmetrically) in terms of predecessors and immediate predecessors.

$$\frac{\begin{array}{ccc} [b_1 \lhd b_2] & [b_1 < b'] & [b' \lhd b_2] \\ \vdots & & \vdots \\ b_1 < b_2 \quad b : A & & b : A \end{array}}{b : A}\ \mathit{split} <_L ,$$

where we require that b' is fresh, i.e., it is different from b_1 and b_2 and does not occur in any assumption on which $b : A$ depends other than the discharged assumptions $b_1 < b'$ and $b' \lhd b_2$.

In Figure 4.4, we present a derivation of the axiom DISCR_R, using the rules of $\mathcal{N}(Kl)$ and the ones introduced in this paragraph. A derivation of the axiom DISCR_L can be obtained symmetrically.

[6] Note that in the case of discrete *unbounded* time, we could omit these rules and replace the rules ser_R and ser_L by analogous ones defined on the relation $\lhd$.

$$\dfrac{\dfrac{\dfrac{[b : \mathsf{F}\top \wedge A \wedge \mathsf{H}A]^1}{b : \mathsf{F}\top}\wedge E \quad \dfrac{[b < c]^2 \quad \dfrac{[b \lhd d]^3 \quad \dfrac{\overset{\Pi}{\overset{d : \mathsf{H}A}{\vdots}} \quad [b < d]^3}{b : \mathsf{FH}A}\mathsf{F}I}{b : \mathsf{FH}A}\,base <^4}{b : \mathsf{FH}A}\,discr <^3_R}{b : \mathsf{FH}A}\mathsf{F}E^2}{b : \mathsf{F}\top \wedge A \wedge \mathsf{H}A \supset \mathsf{FH}A}\supset I^1$$

where Π is the following derivation:

$$\dfrac{[e < d]^5 \quad \dfrac{\dfrac{[b \lhd d]^3 \quad [e < d]^6 \quad \dfrac{[b : \mathsf{F}\top \wedge A \wedge \mathsf{H}A]^1}{b : A}\wedge E}{e : A}\,lin\lhd_L \quad \dfrac{[b \lhd d]^3 \quad [f \lhd d]^6 \quad \dfrac{[b : \mathsf{F}\top \wedge A \wedge \mathsf{H}A]^1}{b : \mathsf{H}A}\wedge E}{f : \mathsf{H}A}\,lin\lhd_L \quad [e < f]^6}{e : A}\mathsf{H}E}{\dfrac{e : A}{d : \mathsf{H}A}\mathsf{H}I^5}\,split <^6_L$$

Fig. 4.4. A derivation of the axiom $DISCR_R$.

Soundness and completeness

Theorem 4.8. *The extensions of $\mathcal{N}(Kl)$ presented above are sound and complete with respect to the semantics of the corresponding logics.*

Proof. Soundness of the extended systems is straightforward, since the rules mirror the properties that the models of the extended logics are required to satisfy.

With regard to completeness, we have already presented derivations of the axioms expressing the properties that define each logic when we introduced the rules.

□

4.2.4 A system for until-free *LTL*

In this section, we present a labeled natural deduction system for the logic $LTL_{\_}$ described in Section 2.3.4. The core of the system comes from [103]; we just apply some slight modifications, partly due to the fact that we do not use an explicit relational symbol for equality and partly just for uniformity of treatment with the other systems presented here.

A labeled version of *LTL*_

For clarity, since the language used in the system $\mathcal{N}(LTL)$ is different from the one of previous sections, we define it formally.

As we already did in Section 2.3.4 in presenting the logic, here we restrict to consider only future-time operators. We also remark that in this case, since it seems to be more common in the related literature, we use an order relation that enjoys reflexivity, i.e. $\leq$ instead of $\prec$. We will use $\leqslant$ as its corresponding in the syntax. Like in Section 4.2.3, we use $\lhd$ to denote, in the syntax, the relation of *immediate predecessor*.

Definition 4.9. *Let L be a denumerable set of labels. If b and c are labels in L and A is an $LTL_{\_}$-formula, then $b \leqslant c$ and $b \lhd c$ are* relational well-formed ($LTL_{\_}$) formulas *and $b : A$ is a* labeled well-formed ($LTL_{\_}$) formula.

An interpretation is defined as usual as a function mapping a label into a time-instant. The notion of $\models_{LTL_{\_}}$ can be extended as follows in order to deal with labeled and relational formulas.

Definition 4.10. *Given an LTL-structure $\mathcal{M} = (\mathcal{N}, \mathcal{V})$, a denumerable set L of labels and an interpretation λ on them,* truth *for a generic formula φ at a pair $(\mathcal{M}, \lambda)$ is the smallest relation $\models_{LTL_{\_}}$ satisfying:*

$$
\begin{aligned}
\mathcal{M}, \lambda \models_{LTL_{\_}} b \leqslant c \quad & \textit{iff} \quad \lambda(b) \leq \lambda(c) \\
\mathcal{M}, \lambda \models_{LTL_{\_}} b \lhd c \quad & \textit{iff} \quad \lambda(b) + 1 = \lambda(c) \\
\mathcal{M}, \lambda \models_{LTL_{\_}} b : A \quad & \textit{iff} \quad \mathcal{M}, \lambda(b) \models_{LTL_{\_}} A
\end{aligned}
$$

$$\frac{\begin{array}{c}[b_1 : A \supset \bot] \\ \vdots \\ b_2 : \bot\end{array}}{b_1 : A}\ \bot E \qquad \frac{\begin{array}{c}[b : A] \\ \vdots \\ b : B\end{array}}{b : A \supset B}\ \supset I \qquad \frac{b : A \supset B \quad b : A}{b : B}\ \supset E$$

$$\frac{\begin{array}{c}[b_1 \lhd b_2] \\ \vdots \\ b_2 : A\end{array}}{b_1 : \mathsf{X}A}\ \mathsf{X}I \qquad \frac{b_1 : \mathsf{X}A \quad b_1 \lhd b_2}{b_2 : A}\ \mathsf{X}E \qquad \frac{\begin{array}{c}[b_1 \lhd b_2] \\ \vdots \\ b : A\end{array}}{b : A}\ ser\lhd \qquad \frac{b_1 \lhd b_2 \quad b_1 \lhd b_3 \quad b_2 : A}{b_3 : A}\ lin\lhd$$

$$\frac{\begin{array}{c}[b_1 \leqslant b_2] \\ \vdots \\ b_2 : A\end{array}}{b_1 : \mathsf{G}A}\ \mathsf{G}I \qquad \frac{b_1 : \mathsf{G}A \quad b_1 \leqslant b_2}{b_2 : A}\ \mathsf{G}E \qquad \frac{\begin{array}{c}[b_1 \leqslant b_1] \\ \vdots \\ b : A\end{array}}{b : A}\ refl\leqslant \qquad \frac{b_1 \leqslant b_2 \quad b_2 \leqslant b_3 \quad \begin{array}{c}[b_1 \leqslant b_3] \\ \vdots \\ b : A\end{array}}{b : A}\ trans\leqslant$$

$$\frac{\begin{array}{c}[b_1 \leqslant b_2] \\ \vdots \\ b : A\end{array}}{b_1 \lhd b_2}\ base\leqslant \qquad \frac{b_0 : \alpha \quad b_0 \leqslant b \quad \begin{array}{c}[b_0 \leqslant b_i] \quad [b_i \lhd b_j] \\ \vdots \\ b_j : A\end{array} \quad [b_i : A]}{b : A}\ ind$$

- In $\mathsf{X}I$, b_2 is *fresh*, i.e. it is different from b_1 and does not occur in any assumption on which $b_2 : A$ depends other than the discarded assumption $b_1 \lhd b_2$.
- In $ser\lhd$, b_2 is fresh, i.e. it is different from b and does not occur in any assumption on which $b : A$ depends other than the discarded assumption $b_1 \lhd b_2$.
- In $\mathsf{G}I$, b_2 is fresh, i.e. it is different from b_1 and does not occur in any assumption on which $b_2 : A$ depends other than the discarded assumption $b_1 \leqslant b_2$.
- In *ind*, b_i and b_j are fresh, i.e. they are different from b and do not occur in any assumption on which $b : A$ depends other than the discarded assumptions of the rule.

Fig. 4.5. The rules of $\mathcal{N}(LTL\_)$.

Given a set Γ of generic formulas and a generic formula φ:

$$\mathcal{M}, \lambda \models_{LTL\_} \Gamma \quad \textit{iff} \quad \mathcal{M}, \lambda \models_{LTL\_} \varphi \textit{ for all } \varphi \in \Gamma$$
$$\Gamma \models_{LTL\_} \varphi \quad \textit{iff} \quad \mathcal{M}, \lambda \models_{LTL\_} \Gamma \textit{ implies } \mathcal{M}, \lambda \models_{LTL\_} \varphi \textit{ for all } \mathcal{M} \textit{ and } \lambda$$

The system $\mathcal{N}(LTL\_)$

The set of rules of the system $\mathcal{N}(LTL\_)$, for which the notion of derivability $\vdash_{\mathcal{N}(LTL\_)}$ can be defined as usual, is given in Figure 4.5.

First of all, we have the standard rules for classical connectives seen in the previous sections. With regard to $\mathsf{G}I$ and $\mathsf{G}E$ we just remark that for simplicity we keep using such rule names even if the relational symbol used ($\leqslant$) is different from that of the systems $\mathcal{N}(Kt)$ and $\mathcal{N}(Kl)$. The set of rules for temporal operators

is completed by $\mathsf{X}I$ and $\mathsf{X}E$, that present, with respect to the relation $\lhd$, the same structure of $\mathsf{G}I$ and $\mathsf{G}E$. In fact, they share the same universal formulation.[7]

The rule $ser\lhd$ models the fact that the flow of time is unbounded towards the future. The rule $lin\lhd$ expresses the uniqueness of the immediate successor of a point. *refl* $\leqslant$ and *trans* $\leqslant$ state the reflexivity and the transitivity of the order relation denoted by $\leqslant$, respectively.

Finally, we have two rules modeling the interactions between the relations $\lhd$ and $\leqslant$. We have already encountered *base* $\leqslant$ (though with respect to the relation $<$) in Section 4.2.3: it captures the fact that $\leqslant$ contains $\lhd$. *ind* models the induction principle underlying the relation between $\lhd$ and $\leqslant$. If (base case) A holds in b_0 and if (inductive step) by assuming that A holds in b_i for an arbitrary b_i $\leqslant$-accessible from b_0, we can derive that A holds also in b_j, where b_j is the immediate successor of b_i, then we can conclude that A holds in every b such that b is $\leqslant$-accessible from b_0.[8]

Soundness

Theorem 4.11. *Let Γ be a set of labeled $LTL_{\_}$-formulas and $b : A$ a labeled $LTL_{\_}$-formula. Then*

$$\Gamma \vdash_{\mathcal{N}(LTL_{\_})} b : A \quad \Rightarrow \quad \Gamma \models_{LTL_{\_}} b : A\,.$$

Proof. By induction on the structure of the derivation of $b : A$. The base case is when $b : A \in \Gamma$ and is trivial. There is one step case for every rule. We consider some cases.

Consider an application of the rule $\mathsf{G}I$

$$\frac{\begin{array}{c}[b_1 < b_2]\\ \Pi\\ b_2 : A\end{array}}{b_1 : \mathsf{G}A}\ \mathsf{G}I$$

where Π is a proof of $b_2 : A$ from hypotheses in Γ', with b_2 fresh and with $\Gamma' = \Gamma \cup \{b_1 < b_2\}$. By the induction hypothesis, for all interpretations λ, if $\mathcal{M}, \lambda \models_{LTL_{\_}} \Gamma'$ then $\mathcal{M}, \lambda \models_{LTL_{\_}} b_2 : A$. We let λ be any interpretation such that $\mathcal{M}, \lambda \models_{LTL_{\_}} \Gamma$, and show that $\mathcal{M}, \lambda \models_{LTL_{\_}} b_1 : \mathsf{G}A$. Let $\lambda(b_1) = n$. Now let us

[7] Notice that, since $\lhd$ is functional, an existential formulation of the rules for introduction and elimination of X would also be possible. Thus we could consider, instead of the ones given, the following two rules:

$$\frac{b_2 : A \quad b_1 \lhd b_2}{b_1 : \mathsf{X}A}\ \mathsf{X}I' \qquad \frac{b_1 : \mathsf{X}A \qquad \begin{array}{c}[b_1 \lhd b_2]\quad [b_2 : A]\\ \vdots\\ b : A\end{array}}{b : A}\ \mathsf{X}E'\ ,$$

where b_2 is required to be fresh in $\mathsf{X}E'$.

[8] The rule is given only in terms of relations between labels, since (for proof-theoretical reasons) we restrict the treatment of operators in the system to the specific rules for their introduction and elimination.

consider a generic successor $n+k$ of n for some $k > 0$. Since λ can be trivially extended to another interpretation (still called λ for simplicity) by setting $\lambda(b_2) = n+k$, the induction hypothesis yields $\mathcal{M}, \lambda \models_{_{LTL_{\_}}} b_2 : A$, i.e. $\mathcal{M}, n+k \models_{_{LTL_{\_}}} A$. Given that k is arbitrary we can conclude $\mathcal{M}, \lambda \models_{_{LTL_{\_}}} b_1 : \mathsf{G}A$.

Consider the case in which the last rule applied is $ser\lhd$:

$$\dfrac{\begin{array}{c}[b_1 \lhd b_2] \\ \Pi \\ b : A\end{array}}{b : A}\ ser\lhd$$

where Π is a proof of $b : A$ from hypotheses in Γ_1, with $\Gamma_1 = \Gamma \cup \{b_1 \lhd b_2\}$. By the side-condition on the application of $ser\lhd$, b_2 is fresh in Π and $b_2 \neq b$. Hence, by applying the induction hypothesis on Π, we have:

$$\Gamma_1 \models_{_{LTL_{\_}}} b : A\ .$$

We proceed by considering a generic *LTL*-structure $\mathcal{M} = (\mathcal{N}, \mathcal{V})$ and a generic interpretation λ on it such that $\mathcal{M}, \lambda \models_{_{LTL_{\_}}} \Gamma$ and showing that this entails

$$\mathcal{M}, \lambda \models_{_{LTL_{\_}}} b : A\ .$$

Let $\lambda(b_1) = n$ for some $n \in \mathbb{N}$. Since every element of $\mathbb{N}$ has an immediate successor, we can define an interpretation $\lambda' = \lambda[b_2 \mapsto n+1]$. Given that b_2 is fresh in Π, we can infer $\mathcal{M}, \lambda' \models_{_{LTL_{\_}}} \Gamma$. Furthermore it holds $\mathcal{M}, \lambda' \models_{_{LTL_{\_}}} b_1 \lhd b_2$ and thus we can conclude $\mathcal{M}, \lambda' \models_{_{LTL_{\_}}} \Gamma_1$. The induction hypothesis yields $\mathcal{M}, \lambda' \models_{_{LTL_{\_}}} b : A$. Since $b_2 \neq b$ (by the side condition on $ser\lhd$) and the interpretations λ and λ' differ only in the value assigned to b_2, we have $\mathcal{M}, \lambda \models_{_{LTL_{\_}}} b : A$ as desired.

Now consider the case in which the last rule applied is *ind*:

$$\dfrac{\begin{array}{ccc} & [b_0 \leqslant b_i] \quad [b_i \lhd b_j] & [b_i : A] \\ \Pi' & & \Pi \\ b_0 : A \quad b_0 \leqslant b & & b_j : A\end{array}}{b : A}\ ind$$

where Π is a proof of $b_j : A$ from hypotheses in Γ_2 and Π' is a proof of $b_0 : A$ from hypotheses in Γ_1, with $\Gamma = \Gamma_1 \cup \{b_0 \leqslant b\}$ and $\Gamma_2 = \Gamma_1 \cup \{b_0 \leqslant b_i\} \cup \{b_i \lhd b_j\} \cup \{b_i : A\}$ for some set Γ_1 of formulas. The side-condition on *ind* ensures that b_i and b_j are fresh in Π. Hence, by applying the induction hypothesis on Π and Π', we have:

$$\Gamma_2 \models_{_{LTL_{\_}}} b_j : A \qquad\qquad \Gamma_1 \models_{_{LTL_{\_}}} b_0 : A.$$

We proceed by considering a generic *LTL*-structure $\mathcal{M} = (\mathcal{N}, \mathcal{V})$ and a generic interpretation λ on it such that $\mathcal{M}, \lambda \models_{_{LTL_{\_}}} \Gamma$ and showing that this entails

$$\mathcal{M}, \lambda \models_{_{LTL_{\_}}} b : A\ .$$

First we note that $\Gamma \supset \Gamma_1$ and therefore $\mathcal{M}, \lambda \models_{_{LTL_{\_}}} \Gamma$ implies $\mathcal{M}, \lambda \models_{_{LTL_{\_}}} \Gamma_1$ and, by induction hypothesis on Π', $\mathcal{M}, \lambda \models_{_{LTL_{\_}}} b_0 : A$. Let $\lambda(b_0) = n$ for some

$n \in \mathbb{N}$. From $\mathcal{M}, \lambda \models_{_{LTL\_}} \Gamma$, we can deduce $\mathcal{M}, \lambda \models_{_{LTL\_}} b_0 \leqslant b$ and thus $\lambda(b) = n + k$ for some $k \in \mathbb{N}$. We show by induction on k that $\mathcal{M}, \lambda \models_{_{LTL\_}} b : A$. As a base case, we have $k = 0$; it follows that $\lambda(b) = \lambda(b_0)$ and thus trivially that $\mathcal{M}, \lambda \models_{_{LTL\_}} b_0 : A$ entails $\mathcal{M}, \lambda \models_{_{LTL\_}} b : A$. Let us consider now the induction step. Given a label b_{k-1} such that $\lambda(b_{k-1}) = n + k - 1$ we show that the induction hypothesis $\mathcal{M}, \lambda \models_{_{LTL\_}} b_{k-1} : A$ entails the thesis $\mathcal{M}, \lambda \models_{_{LTL\_}} b : A$. We can build an interpretation λ' that differs from λ only in the points assigned to b_i and b_j, namely $\lambda' = \lambda[b_i \mapsto n + k - 1][b_j \mapsto n + k]$. It is easy to verify that the interpretation λ' is such that the following three conditions hold:

(i) $\mathcal{M}, \lambda' \models_{_{LTL\_}} b_i : A$;
(ii) $\mathcal{M}, \lambda' \models_{_{LTL\_}} b_0 \leqslant b_i$;
(iii) $\mathcal{M}, \lambda' \models_{_{LTL\_}} b_i \lhd b_j$.

Furthermore the side-condition on the rule *ind* ensures that λ and λ' agree on all the labels occurring in Γ_1, from which we can infer that also $\mathcal{M}, \lambda' \models_{_{LTL\_}} \Gamma_1$ must hold. It follows $\mathcal{M}, \lambda' \models_{_{LTL\_}} \Gamma_2$ and thus (by the induction hypothesis on Π) $\mathcal{M}, \lambda' \models_{_{LTL\_}} b_j : A$. We conclude $\mathcal{M}, \lambda \models_{_{LTL\_}} b : A$ by observing that $\lambda'(b_j) = \lambda(b)$.

Soundness of the other rules can be proved by using an analogous way of reasoning and the corresponding relational properties of the structures.

□

Completeness

With regard to completeness, we remark that, since $\mathcal{N}(LTL\_)$ consists of only finitary rules, it cannot be strongly complete.[9] In fact, it is easy to check that $\{b : \mathsf{X}^i A\}_{i<\omega} \models_{_{LTL\_}} b : \mathsf{G}A$ but (via soundness) we can see that $\{b : \mathsf{X}^i A\}_{i<\omega} \not\vdash_{_{\mathcal{N}(LTL\_)}} b : \mathsf{G}A$, where $\mathsf{X}^0 A$ is just A and $\mathsf{X}^{i+1} A$ stands for $\mathsf{X}\mathsf{X}^i A$.

Nevertheless, our system $\mathcal{N}(LTL\_)$ is weakly complete with respect to the semantics of $\mathcal{N}(LTL\_)$, namely:

Theorem 4.12. *Let Γ be a finite set of labeled $LTL\_$-formulas and $b : A$ a labeled $LTL\_$-formula. Then*

$$\Gamma \models_{_{LTL\_}} b : A \quad \Rightarrow \quad \Gamma \vdash_{_{\mathcal{N}(LTL\_)}} b : A\,.$$

Proof. We need to prove that each rule of inference and each axiom of the Hilbert-style axiomatization $\mathcal{H}(LTL\_)$ given in Section 2.3.4 is derivable in $\mathcal{N}(LTL\_)$. The derivation of the rules is analogous to that described in the proof of Theorem 4.5.

With regard to the axioms, derivations for *A2* and *A4* are completely analogous to that of K_{G} of Theorem 4.5. We show derivations of the other axioms.

[9] This is not a problem of our formulation: all the finitary deduction systems for temporal logics equipped with at least the operators X and G (and thus not compact) have such a defect; see, e.g., [100, Chapter 6].

$(A3)$ $(\mathsf{X}\neg A \supset \neg\mathsf{X}A) \wedge (\neg\mathsf{X}A \supset \mathsf{X}\neg A)$

$$\dfrac{\dfrac{\dfrac{\dfrac{[b:\mathsf{X}\neg A]^1 \quad [b \lhd c]^3}{c:\neg A}\,\mathsf{X}E \qquad \dfrac{\dfrac{[b \lhd d]^4 \quad [b \lhd c]^3}{c:A}\,lin\lhd \qquad \dfrac{[b:\mathsf{X}A]^2 \quad [b \lhd d]^4}{d:A}\,\mathsf{X}E}{c:A}\,ser\lhd^4}{c:\bot}\,\supset E}{\dfrac{c:\bot}{b:\neg\mathsf{X}A}\,ser\lhd^3}\,\supset I^2}{b:\mathsf{X}\neg A \supset \neg\mathsf{X}A}\,\supset I^1$$

$$\dfrac{\dfrac{\dfrac{[b:\neg\mathsf{X}A]^1 \qquad \dfrac{\dfrac{[b \lhd c]^2 \quad [b \lhd d]^4 \quad [c:A]^3}{d:A}\,lin\lhd}{b:\mathsf{X}A}\,\mathsf{X}I^4}{b:\bot}\,\supset E}{\dfrac{b:\bot}{c:\neg A}\,\supset^3}\,\mathsf{X}I^2}{b:\neg\mathsf{X}A \supset \mathsf{X}\neg A}\,\supset I^1$$

$(A5)$ $\mathsf{G}A \supset A \wedge \mathsf{XG}A$

$$\dfrac{\dfrac{\dfrac{[b:\mathsf{G}A]^1 \quad [b \leqslant b]^2}{b:A}\,refl\leqslant^2 \qquad \dfrac{\dfrac{[b \leqslant c]^5 \quad [c \leqslant d]^4 \quad \dfrac{[b:\mathsf{G}A]^1 \quad [b \leqslant d]^6}{d:A}\,\mathsf{G}E}{d:A}\,trans\leqslant^6}{d:A}\,base\leqslant^5}{c:\mathsf{G}A}\,\mathsf{G}I^4}{b:\mathsf{XG}A}\,\mathsf{X}I^3 \qquad \wedge I}{b:A \wedge \mathsf{XG}A}\,\supset I^1$$

$(A6)$ $\mathsf{G}(A \supset \mathsf{X}A) \supset (A \supset \mathsf{G}A)$

$$\dfrac{\dfrac{\dfrac{[b:A]^2 \quad [b \leqslant c]^3 \qquad \dfrac{\dfrac{[b:\mathsf{G}(A \supset \mathsf{X}A)]^1 \quad [b \leqslant d]^4}{d:A \supset \mathsf{X}A}\,\mathsf{G}E \qquad [d:A]^4}{d:\mathsf{X}A}\,\supset E \qquad [d:\lhd d']^4}{d':A}\,\mathsf{X}E}{c:A}\,ind^4}{b:\mathsf{G}A}\,\supset^3}{b:A \supset \mathsf{G}A}\,\supset I^2}{b:\mathsf{G}(A \supset \mathsf{X}A) \supset (A \supset \mathsf{G}A)}\,\supset I^1$$

□

Remark 4.13. Note that we could also express completeness for our system as follows:

$$\Gamma \models_{LTL_{\_}} b:A \quad \Rightarrow \quad \Gamma \vdash_{\mathcal{N}(LTL_{\_})} b:A \quad ,$$

where Γ is a finite set of lwffs. We remark, however, that such a result does not hold if Γ contains also rwffs, i.e. our system is not complete with respect to (even a finite set of) relational assumptions. As an example, it is easy to check that $\{b_1 \lhd b_2\,,\, b_2 \lhd b_1\} \models_{LTL_\_} b : \bot$ but $\{b_1 \lhd b_2\,,\, b_2 \lhd b_1\} \not\vdash_{\mathcal{N}(LTL_\_)} b : \bot$.

4.2.5 Normalization

The labeled natural deduction systems presented in this section have been designed with an eye to normalization matters. In particular, we have restricted the treatment of the operators to the specific rules for their introduction and elimination and in fact for each connective and operator (with the only standard exception of $\bot$) we have one introduction and one elimination rule. Moreover, the rules modeling relational properties are defined in such a way that they can be shown to be reduced to have only atomic conclusions and thus they do not compromise the definition of a normalization procedure. The only exception to this is represented by the rule *ind*, for which a particular treatment is required.

In this section, we omit a detailed treatment of normalization. However, in [103] a system for a so-called *small temporal logic*, which corresponds to the until-free fragment of *LTL* with a semantics given on frames where the principle of induction does not hold, and a normalization procedure for such a system is given. Since the systems presented in Sections 4.2.1, 4.2.2 and 4.2.3 present the same main features of the system in [103], we believe that an analogous procedure could be defined for them. Moreover, the use of some of the techniques required will be illustrated in Section 4.3, where we will present a detailed treatment of normalization for a number of systems, where a proper relational labeling algebra is employed, that capture the same logics of Sections 4.2.1, 4.2.2 and 4.2.3.

With regard to the system of Section 4.2.4 for $LTL_\_$, we just say that, as already remarked, the presence of a principle of induction at a semantical level, and thus of the rule *ind* in the system, requires a much more complex analysis of normalization. A standard subformula property cannot hold for such a system, however we are able to show that a normalizing reduction procedure can be defined and that such a procedure allows us to prove, in a purely syntactic way, the consistency of the system. Again, we omit the details here. However, a full description will be given in Section 5.3 with regard to a system (for a branching-time logic) that is an extension of the one presented here for $LTL_\_$. Thus the procedure defined there can be easily applied also to $\mathcal{N}(LTL_\_)$ by just ignoring the treatment of those rules (specific to the branching case) that are not considered in $\mathcal{N}(LTL_\_)$.

4.2.6 Discussion and related works

We have already discussed some works that are related to the labeled natural deduction systems for tense logics that we have given here, for which, summarizing, we have proved soundness and completeness, and sketched some ideas concerning normalization. Our approach is based on the extension of a fixed base system for the temporal operators with relational rules that express the relational properties of the considered logic. This, in particular, allows for uniform and modular proofs of meta-theoretic properties for families of logics. Moreover, it makes our systems

amenable to extensions to other logics, as we have seen for $LTL_{\_}$ and, as we will show in Chapter 5, to branching-time logics also.

Kt and its extensions

The main difficulties in applying deduction in the context of linear tense logics arise from the need of expressing the condition of *connectedness* in the case of the basic linear tense logic Kl (see [93] for a discussion).

In [66,68] a general presentation of the tense logic Kt, of its quantified version and of some of its extensions, in the context of labeled deduction is provided.

A natural deduction system for Kl is given in [93]. It is a labeled and analytical system, that has only elimination rules for temporal operators and can be used as a decision procedure. The system follows the Kalish/Montague variant for Natural Deduction (see [95]), whose main feature is that of explicitly writing down the goal of the derivation at a given stage. However, the system is closer to labeled tableau systems than to standard natural deduction, and indeed the duality introduction/elimination for modal operators, commonly preserved by labeled natural deduction systems, here is lost. The system in [93] is *analytical* in the sense that all the formulas admissible in a proof of the formula A belong to the set of subformulas of A and their single negations, although some of the rules do not satisfy the subformula property "per se". Labels have a rich structure, which helps build a model: they are nonempty finite sequences of natural numbers with 1 as the first digit, marked with an $[F]$ or a $[P]$. We remark that they are used as prefixes of formulas but no operations are made in a specific relational language. The relation between labels is contained in the structure of the label itself: e.g., $1.2[F]$ denotes that the point 1.2 follows the point 1. In the paper, variants are proposed in order to capture extensions of Kl. Properties of the accessibility relation (like reflexivity, having a first or a last point, density, etc.) are expressed by means of rules that follow the corresponding Hilbert-style axioms closely. To give an idea, we show here a rule that captures the reflexivity of the relation.

$$\frac{w : \mathsf{G}A}{w : A}\,(\leq_T)$$

In general, we can say that the paper focuses more on the automatizability of the proof construction than on the theoretical purity of the system.

It is worth mentioning that in [23], Bonnette and Goré give a labeled sequent system for the minimal tense logic Kt that can easily capture any combination of the reflexive, transitive, euclidean, symmetrical and serial extensions of the logic. We have not considered all of these properties of the accessibility relation here, but the missing ones can be added straightforwardly thanks to the modularity of our system, which we exploit to capture the extensions towards LTL we consider in the remainder of this section. The labeling discipline of [23] is different from ours and is tailored to a lean Prolog implementation of their sequent systems. In contrast, we focus here on the proof-theoretical aspects of our natural deduction systems and leave an implementation for future work.

LTL and *LTL_*

In [103], Marchignoli presents labeled natural deduction proof systems for discrete linear temporal logics. His way of dealing with labels is similar to the approach of Simpson: relational formulas are simply used to express assumptions on logical rules and are not provided with a proper algebra. Our presentation of a system for $LTL\_$ is, up to some minor modifications and adaptations, taken from [103]. As we remarked in Section 4.2.4, the most problematic aspect of defining natural deduction systems for *LTL* (and $LTL\_$) is probably the necessity of modeling the induction principle that links the relation of *next* to the ordering $\leq$ on time points. To tackle this problem, Marchignoli first defines a proof system for a simplified logic ("smaller" than *LTL*) for which no induction rule is needed. The resulting proof system is rather simple and, for such a system, Marchignoli proves that standard proof theoretical properties of predicate logic hold. In particular, it is shown that derivations in this system normalize and that the intuitionistic fragment of the system enjoys the disjunction property and the existential property. Then such a system is extended to capture standard $LTL\_$. The new system requires an induction rule (like the rule *ind* we used in Section 4.2.4), which breaks the clean symmetry of introduction/elimination pairs for temporal operators and causes the failure of normalization. It is shown that normalization of derivations can be obtained instead by defining a new proof system with an infinitary rule. The new system is proved to be equivalent to the system based on the inductive rule as long as we consider finite sets of formulas.

In [19], Bolotov et al. also present a natural deduction calculus for *LTL*. It is a labeled system based on the idea of natural deduction with subordinate proofs originally developed by Jaskowski [94], and then improved and simplified by Quine [130] and Fitch [59]. The system is based on the classical separation of formulas into labeled and relational ones and the rules of the labeled system can be separated, like ours, into two main categories:

- rules for the introduction/elimination of logical connectives;
- rules for the introduction/elimination of temporal operators.

Rules in the first category are quite standard. About rules for temporal operators, it has to be remarked the use of a mechanism of *flagging* for the set of labels. By saying that the label w is flagged, we mean that it is bound to a time point and, hence, that it cannot be rebound to some other point. By saying that w is relatively flagged by v (for example by the judgement $w \leq v$), we restrict the range of time points to which w can be mapped. During the construction of a proof, a label cannot be flagged twice and cannot relatively bind itself. This system also presents rules for the operator *until*. When modeled in a natural deduction setting, the until does not behave very well. Even in this case, three introduction and two elimination rules are needed to represent his behavior. As in [103], also the induction principle requires a specific rule to be modeled. Relational properties are expressed by means of purely relational rules, following an approach similar to that of [159]. Relational formulas are not used just as side conditions but become part of a separate system. The system is proved to be sound and complete and is strongly oriented to the development of a proof-searching procedure, based on the goal-directed nature of the proofs. Such an approach is further developed in

the papers [21], in which an automated proof searching technique is presented, and [22], where an optimization of the system is proposed.

Finally, we mention [7], where an extension of standard (non-labeled) natural deduction for logics like $LTL\_$ is presented and a strong normalization theorem for an intuitionistic version of the calculus is proved. As a consequence, the authors get also a proof of the consistency of the system.

Other methods

We just mention some other works where deduction systems, not falling in the scope of natural deduction, for linear temporal logics are described.

Gentzen's sequent systems are introduced in [73] (for a good presentation in classical logic, see, e.g., [87]) and in fact natural deduction can be seen as a variant of sequent calculus. Traditional sequent calculi for linear temporal logics can be seen in [91, 97, 120, 123, 147, 162]. In [8], a sequent calculus with an ω-rule for $LTL\_$ is proposed. Cut-free labeled sequent systems are also in [28] and [24], which extends the work of [112] to linear temporal logics.

Quite popular in the field of temporal logics is also the use of semantic tableaux, introduced in [15, 85] and extended to modal logics in [61, 89]. Overview on the use of tableaux for temporal logics are in [53, 81, 163]. Interesting examples are in [80, 145, 146] and [9, 11], where a labeled tableaux system for a distributed temporal logic that comprises full LTL is given. A labeled tableaux system, based on the technique of mosaics, is in [105]: we will return to this in Chapter 6. In the case of logics of discrete time, a particular way of managing tableaux generation, based on the use of more general graph structures instead of trees, has been often adopted [12, 41, 98].

Finally, we cite the use of the resolution method, described in [142] for classical logic and extended to linear temporal logics in [1, 36, 56, 57, 158].

4.3 Systems with an explicit relational theory

In this section, we propose the definition of systems designed in the style of Section 3.1 for a number of linear temporal logics and discuss benefits and limitations of such an approach. The difference between the systems in Section 4.2 and the ones that will be presented here is in the fact that relational formulas were used there just as assumptions in the derivation of labeled logical formulas, while here we consider also rules concluding with a relational formula. Part of the material of this section has been presented in [160].

4.3.1 Introduction

As illustrated in Section 3.3.2, labeled deduction systems have been given for several non-classical logics. Research has focused not only on the design of systems for specific logics, but also, more generally, on the characterization of the classes of logics that can be formalized this way. General properties and limitations of labeling techniques have also been investigated. For example, [159] highlights an important

trade-off between limitations and properties, which can be roughly summarized as follows. Assume that we have a labeled system like the ones described in Section 3.3.2, i.e., by summing up, a set of rules for reasoning about the introduction and elimination of modal operators in labeled formulas $b : A$ such as the rules for $\Box$ of the system $\mathcal{N}(K)$ and of its extensions. Assume also that we reason on the semantic information provided by labeling using only *Horn-style relational rules* (see Section 3.3.2). While restricting our systems to such Horn rules allows us to present only a subset of all possible non-classical logics, we can still capture several of the most common modal and relevance logics [159], and, more importantly, labeling provides an efficient general method for establishing the metatheoretical properties of these logics, including their completeness, decidability, and computational complexity. This method relies on the separation between the sub-system for reasoning about lwffs and the sub-system for reasoning about rwffs: derivations of lwffs can depend on derivations of rwffs (e.g. via the $\Box$ rules), but rwffs depend only on rwffs (via the Horn rules).

If we are interested now in considering linear temporal logics, it should be immediately clear that Horn rules do not suffice: even a basic tense logic like Kl (see Section 2.3.2) requires its time points to be connected, i.e. for any two points b and c either $b = c$, or b is before c, or c is before b. It is straightforward to see that such a property cannot be captured by a Horn rule; rather, we need non-atomic rwffs, in particular disjunction ($\sqcup$) of relations, and more complex rules built using a full first-order language, such as the axiom

$$\frac{}{\forall b.c.\, b < c \sqcup b = c \sqcup c < b}\ \mathit{conn}\,.$$

A similar situation occurs if we wish to impose irreflexivity of our worlds. And that's not all: as shown in [159] (in the case of modal logics, but the same arguments apply here, mutatis mutandis), if we move to such a first-order language and wish to retain completeness of the resulting systems, then we need to abandon the strict separation between the sub-system for lwffs and that for rwffs (and let derivations of rwffs depend also on lwffs). As we will see in more detail below, this is best achieved by introducing a so-called *universal falsum*, so that a contradiction in a world can be propagated not only to any other world but also to the relational structure to derive any rwff; and, vice versa, from a contradiction in the relational sub-system we can obtain any lwff.

The structure of this section is the following:

- in Section 4.3.2, first we give a brief presentation of the syntax and semantics of a labeled version of the logic Kl; then we give a labeled natural deduction system $\mathcal{N}'(Kl)$ for Kl, which we show to be sound and complete (extending the completeness proofs given for modal logics in [159]); finally, we show that $\mathcal{N}'(Kl)$ possesses a number of useful normalization properties; in particular, derivations reduce to a normal form that enjoys a subformula property;
- in Section 4.3.3, we extend $\mathcal{N}'(Kl)$ to capture some interesting extensions of Kl;
- in Section 4.3.4 we discuss how to extend our systems to capture richer logics like (fragments of) *LTL*.

4.3.2 A system for *Kl*

A labeled version of *Kl*

The definition of the language of tense formulas and of the semantics of Kl has been given in Section 2.3. The extension of the language with labels and relational symbols, as required by the labeled deduction setting, are in the vein of those described in Section 3.3.2. For a greater clarity, in the following subsections, we recall some notions, give a formal definition of the language used in this section and present an adaptation of the semantics of Kl to the labeled language.

Syntax

Definition 4.14. *Let L be a denumerable set of labels and let b and c be labels in L. If A is a well-formed tense formula, then $b : A$ is a* labeled well-formed tense formula *(labeled formula or lwff, for short).*

The set of well-formed relational $(\mathcal{N}'(Kl))$-formulas *(relational formulas or rwffs, for short) is defined as follows:*

$$\rho ::= b < c \mid b = c \mid \emptyset \mid \rho \sqsupset \rho \mid \forall b.\, \rho\,.$$

For simplicity, in this section we will often omit the adjective "tense" and just speak of labeled formulas or lwffs, as well as we will speak just of relational formulas (or rwffs) instead of $\mathcal{N}'(Kl)$-relational formulas. As in Section 3.3.2, φ will denote a generic formula (lwff or rwff). We say that an lwff $b : A$ is *atomic* when A is atomic, i.e. A is a propositional variable or A is $\bot$. An rwff ρ is *atomic* when it does not contain any connective or quantifiers, i.e. ρ is $\emptyset$ or ρ has the form $b < c$ or $b = c$. The *grade* of an lwff or rwff is the number of occurrences of connectives ($\supset$ or $\sqsupset$), operators (G or H), and quantifiers ($\forall$). Finally, given a set of lwffs Γ and a set of rwffs Δ, we call the ordered pair (Γ, Δ) a *proof context*.

$\emptyset$ and $\sqsupset$ denote, respectively, the falsum and the implication in the relational language. Both the languages of labeled and relational formulas present a minimal set of connectives, operators and quantifiers. As usual, we can introduce abbreviations and use, e.g., $\neg$, $\wedge$, $\vee$ and $\sim$, $\sqcap$, $\sqcup$, for the negation, the conjunction, and the disjunction in the labeled language and in the relational one, respectively. For instance, $\neg A \equiv A \supset \bot$ and $\rho' \sqcup \rho'' \equiv (\rho' \sqsupset \emptyset) \sqsupset \rho''$. We can also define $\top \equiv \neg\, \bot$ or other quantifiers, e.g. $\exists b.\, \rho \equiv \sim \forall b.\ \sim \rho$.

Semantics

The notions of Kl-frames and models, together with the semantics of the logic Kl, are given in Section 2.3.2. Here we recall the notion of an interpretation and define the semantics of the labeled logic corresponding to Kl. In particular, we extend the notion of $\models_{Kl}$ defined in Section 2.3.2 with respect to labeled and relational formulas. Note that, for simplicity, we keep using the symbol $\models_{Kl}$ even if the underlying notion is different, as the relational language used is different, from the one of Section 4.2.2.

Definition 4.15. *Given a denumerable set of labels L and a linear temporal structure $\mathcal{M} = (\mathcal{W}, \prec, \mathcal{V})$, an* interpretation *is a function $\lambda : L \to \mathcal{W}$ that maps every label in L to a world in $\mathcal{W}$.*

Given a linear temporal structure $\mathcal{M}$ and an interpretation λ on it, truth *for an rwff or lwff φ is the smallest relation $\models_{Kl}$ satisfying:*

$$
\begin{array}{lll}
\mathcal{M}, \lambda \models_{Kl} b < c & \textit{iff} & (\lambda(b), \lambda(c)) \in \prec; \\
\mathcal{M}, \lambda \models_{Kl} b = c & \textit{iff} & \lambda(b) = \lambda(c); \\
\mathcal{M}, \lambda \models_{Kl} \rho_1 \sqsupset \rho_2 & \textit{iff} & \mathcal{M}, \lambda \models_{Kl} \rho_1 \textit{ implies } \mathcal{M}, \lambda \models_{Kl} \rho_2; \\
\mathcal{M}, \lambda \models_{Kl} \forall b.\, \rho & \textit{iff} & \textit{for all } c,\ \mathcal{M}, \lambda \models_{Kl} \rho[c/b];
\end{array}
$$

$$
\begin{array}{lll}
\mathcal{M}, \lambda \models_{Kl} b : p & \textit{iff} & p \in \mathcal{V}(\lambda(b)); \\
\mathcal{M}, \lambda \models_{Kl} b : A \supset B & \textit{iff} & \mathcal{M}, \lambda \models_{Kl} b : A \textit{ implies } \mathcal{M}, \lambda \models_{Kl} b : B; \\
\mathcal{M}, \lambda \models_{Kl} b : \mathsf{G}A & \textit{iff} & \textit{for all } c,\ \mathcal{M}, \lambda \models_{Kl} b < c \textit{ implies } \mathcal{M}, \lambda \models_{Kl} c : A; \\
\mathcal{M}, \lambda \models_{Kl} b : \mathsf{H}A & \textit{iff} & \textit{for all } c,\ \mathcal{M}, \lambda \models_{Kl} c < b \textit{ implies } \mathcal{M}, \lambda \models_{Kl} c : A.
\end{array}
$$

Hence, $\mathcal{M}, \lambda \not\models b : \bot$ and $\mathcal{M}, \lambda \not\models \emptyset$. When $\mathcal{M}, \lambda \models_{Kl} \varphi$, we say that φ is true *in $\mathcal{M}$ according to the interpretation λ. By extension:*

$$
\begin{array}{lll}
\mathcal{M}, \lambda \models_{Kl} \Gamma & \textit{iff} & \mathcal{M}, \lambda \models_{Kl} b : A \textit{ for all } b : A \in \Gamma; \\
\mathcal{M}, \lambda \models_{Kl} \Delta & \textit{iff} & \mathcal{M}, \lambda \models_{Kl} \rho \textit{ for all } \rho \in \Delta; \\
\mathcal{M}, \lambda \models_{Kl} (\Gamma, \Delta) & \textit{iff} & \mathcal{M}, \lambda \models_{Kl} \Gamma \textit{ and } \mathcal{M}, \lambda \models_{Kl} \Delta; \\
\Gamma, \Delta \models_{Kl} \varphi & \textit{iff} & \mathcal{M}, \lambda \models_{Kl} (\Gamma, \Delta) \textit{ implies } \mathcal{M}, \lambda \models_{Kl} \varphi \\
 & & \textit{for all } \mathcal{M} \textit{ and } \lambda.
\end{array}
$$

Truth for lwffs and rwffs built using other connectives or operators can be defined in the usual manner. As an abbreviation, we will sometimes write $\Gamma, \Delta \models_{Kl}^{\mathcal{M},\lambda} \varphi$ to denote that $\mathcal{M}, \lambda \models_{Kl} (\Gamma, \Delta)$ implies $\mathcal{M}, \lambda \models_{Kl} \varphi$;

An axiomatization of Kl

Several different Hilbert-style axiomatizations have been given for the logic *Kl*. Here we will consider an axiomatization, given in [132], which is slightly different from (but clearly equivalent to) the one presented in Section 2.3.2.

(*G1*) $\mathsf{G}(A \supset B) \supset (\mathsf{G}A \supset \mathsf{G}B)$
(*G2*) $\neg\mathsf{H}\neg\mathsf{G}A \supset A$
(*G3*) $\mathsf{G}A \supset \mathsf{G}\mathsf{G}A$
(*G4*) $[\mathsf{G}(A \vee B) \ \wedge\ \mathsf{G}(A \vee \mathsf{G}B) \ \wedge\ \mathsf{G}(\mathsf{G}A \vee B)] \supset (\mathsf{G}A \vee \mathsf{G}B)$

(Nec_G) If A then $\mathsf{G}A$
(Nec_H) If A then $\mathsf{H}A$
(*MP*) If A and $A \supset B$ then B

The axiom (*G1*) is standard for modal and temporal logics, while (*G2*) sets the dual relation between G and H, (*G3*) expresses the transitivity and (*G4*) the connectedness of G. For brevity, we have omitted the symmetrical axioms (*H1*)-(*H4*) that are obtained by replacing every G by H and vice versa. Moreover, every classical tautology is a tautology, and there are rules for modus ponens and necessitation for both G and H.

Along this section, we denote with $\mathcal{H}'(Kl)$ the axiomatization given above. The set of *theorems of* $\mathcal{H}'(Kl)$ is defined as the smallest set of tense formulas containing the set of axioms and closed with respect to the rules of inference above. We denote with $\vdash_{Kl}$ the notion of derivability in $\mathcal{H}'(Kl)$, i.e. $\vdash_{Kl} A$ iff A is a theorem of $\mathcal{H}'(Kl)$. Furthermore we write $\Gamma \vdash_{Kl} A$ to say that A is derivable in $\mathcal{H}(K)$ from assumptions in Γ.

The system $\mathcal{N}'(Kl)$

Our labeled natural deduction system $\mathcal{N}'(Kl) = \mathcal{N}(Kl_L) + \mathcal{N}(Kl_R) + \mathcal{N}(Kl_G)$ comprises of three sub-systems, whose rules are given in Figure 4.6.

The propositional and temporal rules of $\mathcal{N}(Kl_L)$ allow us to derive lwffs from other lwffs with the help of rwffs. The rules $\supset I$, $\supset E$ and $\bot E$ are just the labeled version of the standard natural deduction rules and are as defined in Section 3.3.2.

The temporal operators G and H share the structure of the basic introduction/elimination rules, with respect to the same accessibility relation $<$. Such rules are analogous to the ones seen in Section 4.2.

The relational rules of $\mathcal{N}(Kl_R)$ allow us to derive rwffs from other rwffs only. The rules $RAA_\emptyset$, $\sqsupset I$, and $\sqsupset E$ are reductio ad absurdum and implication introduction and elimination for rwffs, while $\forall I$ and $\forall E$ are the standard rules for universal quantification, with the usual proviso for $\forall I$. There are also four axiomatic rules (or "axioms", for short) *refl*$=$, *irrefl*$<$, *trans*$<$, and *conn*, which express the properties of $=$[10] and $<$, where, for readability, we employed the symbols for disjunction, conjunction, and negation.

The general rules of $\mathcal{N}(Kl_G)$ allow us to derive lwffs from rwffs and vice versa. The rule *mon* applies monotonicity to an lwff or rwff φ, while the rules $uf1$ and $uf2$ export falsum (and we thus call it a *universal falsum*) from the labeled sub-system to the relational one, and vice versa.[11]

For what concerns this section, we adapt the standard definitions (Section 3.2) of derivation, proof, theorem, etc. as follows.

Definition 4.16. *A* derivation *of a formula (lwff or rwff)* φ *from a proof context* (Γ, Δ) *in* $\mathcal{N}'(Kl)$ *is a tree formed using the rules in* $\mathcal{N}'(Kl)$*, ending with* φ *and*

[10] Note that we do not need further axioms to express symmetry and transitivity of $=$, since the former can be derived by using *mon*, *conn*, and *irrefl*$<$, and the latter by using *mon*.

[11] Note that the presentation of the system could be simplified by introducing a unique symbol for falsum (say $\lightning$), shared by the labeled and the relational sub-systems. In that case, we would not need the rules $uf1$ and $uf2$, while the rules for falsum elimination $\bot E$ and $RAA_\emptyset$ could be replaced by the following rule, where with $-\varphi$ we denote the negation of a generic formula (labeled or relational):

$$\dfrac{\begin{array}{c}[-\varphi]\\ \vdots \\ \lightning\end{array}}{\varphi}\ RAA_{\lightning}$$

However, we prefer to maintain a clear separation between the two sub-systems, as it will allow us to give a simpler presentation of normalization.

$$\frac{\begin{array}{c}[b : A \supset \bot]\\ \vdots\\ c : \bot\end{array}}{b : A}\ \bot E \qquad \frac{\begin{array}{c}[b : A]\\ \vdots\\ b : B\end{array}}{b : A \supset B}\ \supset I \qquad \frac{b : A \supset B \quad b : A}{b : B}\ \supset E$$

$$\frac{\begin{array}{c}[b < c]\\ \vdots\\ c : A\end{array}}{b : \mathsf{G}A}\ \mathsf{G}I \qquad \frac{b : \mathsf{G}A \quad b < c}{c : A}\ \mathsf{G}E \qquad \frac{\begin{array}{c}[c < b]\\ \vdots\\ c : A\end{array}}{b : \mathsf{H}A}\ \mathsf{H}I \qquad \frac{b : \mathsf{H}A \quad c < b}{c : A}\ \mathsf{H}E$$

$$\frac{\begin{array}{c}[\rho \sqsupset \emptyset]\\ \vdots\\ \emptyset\end{array}}{\rho}\ RAA_{\emptyset} \qquad \frac{\begin{array}{c}[\rho_1]\\ \vdots\\ \rho_2\end{array}}{\rho_1 \sqsupset \rho_2}\ \sqsupset I \qquad \frac{\rho_1 \sqsupset \rho_2 \quad \rho_1}{\rho_2}\ \sqsupset E \qquad \frac{\rho}{\forall b.\, \rho}\ \forall I \qquad \frac{\forall b.\, \rho}{\rho[c/b]}\ \forall E$$

$$\frac{}{\forall b.\, b = b}\ refl{=} \qquad \frac{}{\forall b.\ \sim (b < b)}\ irrefl{<}$$

$$\frac{}{\forall b.c.d.\, (b < c \sqcap c < d) \sqsupset b < d}\ trans{<} \qquad \frac{}{\forall b.c.\, b < c \sqcup b = c \sqcup c < b}\ conn$$

$$\frac{\varphi \quad b = c}{\varphi[c/b]}\ mon \qquad \frac{b : \bot}{\emptyset}\ uf1 \qquad \frac{\emptyset}{b : \bot}\ uf2$$

- In $\mathsf{G}I$ (respectively, $\mathsf{H}I$), c is different from b and does not occur in any assumption on which $c : A$ depends other than the discharged assumption $b < c$ (respectively, $c < b$).
- In $\forall I$, the variable b must not occur in any open assumption on which ρ depends.

Fig. 4.6. The rules of $\mathcal{N}'(Kl)$.

$$\frac{\rho_1}{\rho_1 \sqcup \rho_2}\ \sqcup I1 \qquad \frac{\rho_2}{\rho_1 \sqcup \rho_2}\ \sqcup I2 \qquad \frac{\rho_1 \sqcup \rho_2 \quad \begin{array}{c}[\rho_1]\\ \vdots\\ \rho\end{array} \quad \begin{array}{c}[\rho_2]\\ \vdots\\ \rho\end{array}}{\rho}\ \sqcup E$$

$$\frac{\rho[c/b]}{\exists b.\, \rho}\ \exists I \qquad \frac{\exists b.\, \rho \quad \begin{array}{c}[\rho[c/b]]\\ \vdots\\ \rho'\end{array}}{\rho'}\ \exists E$$

Fig. 4.7. Some derived rules.

depending only on a finite subset of $\Gamma \cup \Delta$. We then write $\Gamma, \Delta \vdash_{\mathcal{N}'(Kl)} \varphi$. A derivation of φ in $\mathcal{N}'(Kl)$ depending on the empty set, $\vdash_{\mathcal{N}'(Kl)} \varphi$, is a proof *of φ in $\mathcal{N}'(Kl)$ and we then say that φ is a theorem of $\mathcal{N}'(Kl)$.*

We will give concrete examples of derivations in the following. For simplicity, we will sometimes employ the rules for conjunction $\wedge$, disjunction $\vee$ and the operators F and P which are derived as is standard (Section 4.2), as well as other derived rules such as those for $\sqcup$, and $\exists$ given in Figure 4.7.

Soundness

Here we prove the soundness of the system. The proof follows the standard technique, provided the required adaptation to the labeled case (see also [66,148,159]).

Theorem 4.17. *$\mathcal{N}'(Kl) = \mathcal{N}(Kl_L) + \mathcal{N}(Kl_R) + \mathcal{N}(Kl_G)$ is sound, i.e. it holds:*

(i) $\Gamma, \Delta \vdash_{\mathcal{N}'(Kl)} \rho \quad \Rightarrow \quad \Gamma, \Delta \models_{Kl} \rho$;
(ii) $\Gamma, \Delta \vdash_{\mathcal{N}'(Kl)} b : A \quad \Rightarrow \quad \Gamma, \Delta \models_{Kl} b : A$.

Proof.

(i) The proof is by induction on the structure of the derivation of ρ. The base case is when $\rho \in \Delta$ and is trivial. There is one step case for every axiom or rule. The axioms *conn*, *trans*$<$, and *irrefl*$<$ directly refer to the properties of connectedness, transitivity, and irreflexivity of *Kl* models and thus are trivially sound, while *refl*$=$ and *mon* preserve soundness by definition of $\mathcal{M}, \lambda \models_{Kl} b = c$ (Definition 4.15).
Consider the case of an application of $RAA_{\emptyset}$

$$\dfrac{\begin{array}{c}\Gamma\ \Delta\ [\rho \sqsupset \emptyset]^1 \\ \Pi \\ \emptyset\end{array}}{\rho}\ RAA^1_{\emptyset}$$

where $\Delta_1 = \Delta \cup \{\rho \sqsupset \emptyset\}$. By the induction hypothesis, $\Gamma, \Delta_1 \models_{Kl} \emptyset$. Let us consider an arbitrary model $\mathcal{M}$ and an arbitrary interpretation λ; we assume $\mathcal{M}, \lambda \models_{Kl} (\Gamma, \Delta)$ and prove $\mathcal{M}, \lambda \models_{Kl} \rho$. Since $\mathcal{M}, \lambda \not\models_{Kl} \emptyset$, from the induction hypothesis we obtain $\mathcal{M}, \lambda \not\models_{Kl} (\Gamma, \Delta_1)$, that, given the assumption $\mathcal{M}, \lambda \models_{Kl} (\Gamma, \Delta)$, leads to $\mathcal{M}, \lambda \not\models_{Kl} \rho \sqsupset \emptyset$, i.e. $\mathcal{M}, \lambda \models_{Kl} \rho$ and $\mathcal{M}, \lambda \not\models_{Kl} \emptyset$ by Definition 4.15.
The cases for $\sqsupset I$, $\sqsupset E$, $\forall I$ and $\forall E$ follow by simple adaptations of the standard proofs for classical logic.
Finally, consider the case of an application of $uf1$

$$\dfrac{\begin{array}{c}\Gamma\ \Delta \\ \Pi \\ b :\bot\end{array}}{\emptyset}\ uf1$$

for a proof context (Γ, Δ) and some label b. By the induction hypothesis, we have $\Gamma, \Delta \models_{Kl} b :\bot$. Given a generic model $\mathcal{M}$ and a generic interpretation λ, we can write $\mathcal{M}, \lambda \not\models_{Kl} b :\bot$; it follows that $\mathcal{M}, \lambda \not\models_{Kl} (\Gamma, \Delta)$ and then also $\Gamma, \Delta \models^{\mathcal{M},\lambda}_{Kl} \emptyset$ by Definition 4.15.

(ii) As in (i), by induction on the structure of the derivation of $b : A$. The base case is trivial and there is a step case for every rule of the labeled system. The cases of introduction and elimination of connectives and that of universal falsum are as in (i).
Consider an application of the rule $\mathsf{G}I$

$$\dfrac{\begin{array}{c}\Gamma\ \Delta\ [b < c]^1 \\ \Pi \\ c : A\end{array}}{b : \mathsf{G}A}\ \mathsf{G}I^1$$

where $\Gamma, \Delta_1 \vdash_{\mathcal{N}'(Kl)} c : A$ with c fresh and with $\Delta_1 = \Delta \cup \{b < c\}$. By the induction hypothesis, it holds $\Gamma, \Delta \models_{Kl} c : A$. We let λ be any interpretation such that $\mathcal{M}, \lambda \models_{Kl} (\Gamma, \Delta)$ and show that $\mathcal{M}, \lambda \models_{Kl} b : \mathsf{G}A$. Let w be any world such that $\lambda(b) \prec w$. Since λ can be trivially extended to another interpretation (still called λ for simplicity) by setting $\lambda(c) = w$, the induction hypothesis yields $\mathcal{M}, \lambda \models_{Kl} c : A$, and thus $\mathcal{M}, \lambda \models_{Kl} b : \mathsf{G}A$.
Finally, consider an application of the rule $\mathsf{G}E$

$$\dfrac{\begin{array}{cc}\Gamma_1\ \Delta_1 & \Gamma_2\ \Delta_2 \\ \Pi_1 & \Pi_2 \\ b : \mathsf{G}A & b < c\end{array}}{c : A}\ \mathsf{G}E\,.$$

Let $\mathcal{M}$ be an arbitrary model and λ an arbitrary interpretation. If we assume $\mathcal{M}, \lambda \models_{Kl} (\Gamma_1 \cup \Gamma_2, \Delta_1 \cup \Delta_2)$, then from the induction hypotheses we obtain $\mathcal{M}, \lambda \models_{Kl} b : \mathsf{G}A$ and $\mathcal{M}, \lambda \models_{Kl} b < c$, and thus $\mathcal{M}, \lambda \models_{Kl} c : A$ by Definition 4.15.
The treatment of $\mathsf{H}I$ and $\mathsf{H}E$ is analogous.

□

Completeness

Since the axiomatization of Kl given in Section 4.3.2 is sound and complete, we can prove in $\mathcal{N}'(Kl)$ the axioms and the rules of the axiomatization to establish the completeness of $\mathcal{N}'(Kl)$ indirectly (and we do so in the second part of this section). It seems interesting, however, to give also a direct proof of completeness, by adapting standard proofs for labeled systems (see, e.g., [66, 148, 159]) and in particular by extending those for modal logics in [159], which has been our starting point for the systems in this section, to the case of universal falsum and other general rules that mix derivations of lwffs and rwffs.

Completeness by canonical model construction

In the following, slightly abusing notation, we will write $\varphi \in (\Gamma, \Delta)$ whenever $\varphi \in \Gamma$ or $\varphi \in \Delta$, and write $b \in (\Gamma, \Delta)$ whenever the label b occurs in some $\varphi \in (\Gamma, \Delta)$.

Definition 4.18. *A proof context* (Γ, Δ) *is* $\mathcal{N}'(Kl)$-consistent *iff* $\Gamma, \Delta \nvdash b :\bot$ *for every* b*, and it is* $\mathcal{N}'(Kl)$-inconsistent *otherwise.*

Note that we can have inconsistency also by deriving $\emptyset$ in the relational system; given the rules $uf1$ and $uf2$ for universal falsum, also this case is captured by the previous definition.

For simplicity, in the following we will omit the "$\mathcal{N}'(Kl)$" and simply speak of consistent and inconsistent proof contexts.

Proposition 4.19. *Let* (Γ, Δ) *be a consistent proof context. Then:*

(i) for every b *and every* A*, either* $(\Gamma \cup \{b : A\}, \Delta)$ *is consistent or* $(\Gamma \cup \{b : \neg A\}, \Delta)$ *is consistent;*
(ii) for every relational formula ρ *, either* $(\Gamma, \Delta \cup \{\rho\})$ *is consistent or* $(\Gamma, \Delta \cup \{\sim \rho\})$ *is consistent.*

Proof. (i) Suppose that both $(\Gamma \cup \{b : A\}, \Delta)$ and $(\Gamma \cup \{b : \neg A\}, \Delta)$ are inconsistent. Then from $\Gamma \cup \{b : A\}, \Delta \vdash_{\mathcal{N}'(Kl)} b :\bot$, by applying the rule $\supset I$, we get $\Gamma, \Delta \vdash_{\mathcal{N}'(Kl)} b : \neg A$. Similarly, from $\Gamma \cup \{b : \neg A\}, \Delta \vdash_{\mathcal{N}'(Kl)} b :\bot$, by applying the rule $\bot E$, we get $\Gamma, \Delta \vdash_{\mathcal{N}'(Kl)} b : A$.
But, if both $b : A$ and $b : \neg A$ are derivable in the proof context (Γ, Δ), then it also holds $\Gamma, \Delta \vdash_{\mathcal{N}'(Kl)} b :\bot$, by the rule $\sim E$. It follows that the original proof context (Γ, Δ) had to be inconsistent (contradiction).
(ii) The proof for the relational case is analogous and is obtained by using the corresponding relational rules i.e. $\sqsupset I$, $RAA_{\emptyset}$ and $\sim E$.

□

Definition 4.20. *A proof context* (Γ, Δ) *is* maximally consistent *iff the following three conditions hold:*

1. (Γ, Δ) *is consistent,*
2. for every relational formula ρ*, either* $\rho \in \Delta$ *or* $\sim \rho \in \Delta$*,*
3. for every b *and every* A*, either* $b : A \in \Gamma$ *or* $b : \neg A \in \Gamma$*.*

Completeness follows by a Henkin–style proof, where a canonical model

$$\mathcal{M}^C = (\mathcal{W}^C, \prec^C, \mathcal{V}^C)$$

is built from a proof context (Γ, Δ) to show that $(\Gamma, \Delta) \nvdash \varphi$ implies $\Gamma, \Delta \not\Vdash^{\mathcal{M}^C, \lambda^C} \varphi$ for every formula φ.

In standard proofs for unlabeled modal, temporal, and for other non-classical logics, the set $\mathcal{W}^C$ is obtained by progressively building maximally consistent sets of formulas, where consistency is locally checked within each set. In our case, given the presence of lwffs and rwffs, we modify the Lindenbaum lemma to extend (Γ, Δ) to one single maximally consistent context (Γ^*, Δ^*), where consistency is "globally" checked also against the additional assumptions in Δ.[12] The elements of $\mathcal{W}^C$ are then built by partitioning Γ^* and Δ^* with respect to the labels, and the relation $\prec^C$ between the worlds is defined by exploiting the information in Δ^*.

[12] We consider only consistent proof contexts. If (Γ, Δ) is inconsistent, then $\Gamma, \Delta \vdash_{\mathcal{N}'(Kl)} \varphi$ for all φ, and thus completeness immediately holds for all lwffs and rwffs.

In the Lindenbaum lemma for predicate logic, a maximally consistent and ω-complete set of formulas is inductively built by adding for every formula $\neg\forall b.\, A$ a *witness* to its truth, namely a formula $\neg A[s/b]$ for some new individual constant s. This ensures that the resulting set is ω-complete, i.e. that if, for every closed term t, $A[t/b]$ is contained in the set, then so is $\forall b.\, A$. A similar procedure applies here not only for rwffs $\sim \forall b.\, \rho$, but also in the case of lwffs of the form $b : \neg\mathsf{G}A$. That is, together with $b : \neg\mathsf{G}A$ we consistently add $c : \neg A$ and $b < c$ for some new c, which acts as a *witness world* to the truth of $b : \neg\mathsf{G}A$. This ensures that the maximally consistent context (Γ^*, Δ^*) is such that if $b < d \in (\Gamma^*, \Delta^*)$ implies $d : B \in (\Gamma^*, \Delta^*)$ for every d, then $b : \mathsf{G}B \in (\Gamma^*, \Delta^*)$, as shown in Lemma 4.22 below. Note that in the standard completeness proof for unlabeled modal logics, for instance, one instead considers a canonical model $\mathcal{M}^C$ and shows that if $\mathcal{W}_1 \in \mathcal{W}^C$ and $\mathcal{M}^C, \mathcal{W}_1 \vDash \neg\mathsf{G}A$, then $\mathcal{W}^C$ also contains a world $\mathcal{W}_2$ accessible from $\mathcal{W}_1$ that serves as a witness world to the truth of $\neg\mathsf{G}A$ at $\mathcal{W}_1$, i.e. $\mathcal{M}^C, \mathcal{W}_2 \vDash \neg A$.

Lemma 4.21. *Every consistent proof context (Γ, Δ) can be extended to a maximally consistent proof context (Γ^*, Δ^*).*

Proof. We first extend the language of $\mathcal{N}'(Kl)$ with infinitely many new constants for witness terms and for witness worlds. Let t range over the original terms, s range over the new constants for witness terms, and r range over both; further, let w range over labels, v range over the new constants for witness worlds, and u range over both. All these may be subscripted. Let φ_1, φ_2, ... be an enumeration of all lwffs and rwffs in the extended language; when φ_i is $u : A$, we write $\neg\varphi_i$ for $u : \neg A$.

We iteratively build a sequence of consistent proof contexts by defining $(\Gamma_0, \Delta_0) = (\Gamma, \Delta)$ and $(\Gamma_{i+1}, \Delta_{i+1})$ to be:

- (Γ_i, Δ_i), if $(\Gamma_i \cup \{\varphi_{i+1}\}, \Delta_i)$ is inconsistent; else
- $(\Gamma_i \cup \{u : \neg\mathsf{G}A, v : \neg A\}, \Delta_i \cup \{u < v\})$ for a v not occurring in $(\Gamma_i \cup \{u : \neg\mathsf{G}A\}, \Delta_i)$ if φ_{i+1} is $u : \neg\mathsf{G}A$; else
- $(\Gamma_i \cup \{u : \neg\mathsf{H}A, v : \neg A\}, \Delta_i \cup \{v < u\})$ for a v not occurring in $(\Gamma_i \cup \{u : \neg\mathsf{H}A\}, \Delta_i)$ if φ_{i+1} is $u : \neg\mathsf{H}A$; else
- $(\Gamma_i, \Delta_i \cup \{\sim \forall b.\, \rho, \sim \rho[s/b]\})$ for an s not occurring in $(\Gamma_i, \Delta_i \cup \{\sim \forall b.\, \rho\})$ if φ_{i+1} is $\neg\forall b.\, \rho$; else
- $(\Gamma_i \cup \{\varphi_{i+1}\}, \Delta_i)$ if φ_{i+1} is an lwff or $(\Gamma_i, \Delta_i \cup \{\varphi_{i+1}\})$ if φ_{i+1} is an rwff.

Now define

$$(\Gamma^*, \Delta^*) = (\bigcup_{i\geq 0} \Gamma_i, \bigcup_{i\geq 0} \Delta_i)\,.$$

We show that the proof context (Γ^*, Δ^*) is maximally consistent, i.e. it verifies the three conditions of Definition 4.20.

(i) First we prove that our construction preserves consistency by showing that every (Γ_i, Δ_i) is consistent. The only interesting cases are when φ_{i+1} is one of $\neg\mathsf{G}A$, $\neg\mathsf{H}A$, or $\sim \forall b.\, \rho$. We only consider the first case, since the second one is symmetrical, and the third is very similar.
If $(\Gamma_i \cup \{u : \neg\mathsf{G}A\}, \Delta_i)$ is consistent, then so is $(\Gamma_i \cup \{u : \neg\mathsf{G}A, v : \neg A\})$ for a v not occurring in $(\Gamma_i \cup \{u : \neg\mathsf{G}A\}, \Delta_i)$. By contraposition, suppose that

$$\Gamma_i \cup \{u : \neg \mathsf{G}A, v : \neg A\}\,,\ \Delta_i \cup \{u < v\} \vdash_{\mathcal{N}'(Kl)} u_j :\bot$$

by a derivation Π (where v does not occur in $(\Gamma_i \cup \{u : \neg \mathsf{G}A\}, \Delta_i)$). Then in $\mathcal{N}'(Kl)$ we can have a derivation like the following:

$$\dfrac{\dfrac{\dfrac{\dfrac{u_j :\bot}{v : A}\,\bot E^1}{u : \mathsf{G}A}\,\mathsf{G}I^2 \qquad\qquad u : \neg \mathsf{G}A}{u :\bot}\,\neg E}{\Gamma_i \quad \Delta_i \quad u : \neg \mathsf{G}A \quad [v : \neg A]^1 \quad [u < v]^2}\ \Pi$$

This shows that $(\Gamma_i \cup \{u : \neg \mathsf{G}A\}, \Delta_i)$ is inconsistent, which is not the case.

(ii) Consider an rwff ρ. Suppose that both $\rho \notin \Delta^*$ and $\sim \rho \notin \Delta^*$ hold. Let ρ be φ_{i+1} for some i in our enumeration of formulas and $\sim \rho$ be φ_{j+1}. Now suppose $i < j$ (the other case is symmetrical). $\rho \notin \Delta^*$ implies that $(\Gamma_i, \Delta_i \cup \{\varphi_{i+1}\})$ is inconsistent. Given that in our inductive construction we only add formulas to the proof context, i.e. $\Delta_i \subseteq \Delta_j$, we have that $(\Gamma_j, \Delta_j \cup \{\varphi_{i+1}\})$ is also inconsistent. Then, by Proposition 4.19(ii), $(\Gamma_j, \Delta_j \cup \{\varphi_{j+1}\})$ has to be consistent and φ_{j+1} is added by definition to Δ_j. This implies $\varphi_{j+1} \in \Delta^*$, i.e. $\sim \rho \in \Delta^*$.

(iii) The proof for labeled formulas is the same as in the previous case and proceeds by contraposition by using Proposition 4.19(i).

□

Lemma 4.22. *Let (Γ, Δ) be a maximally consistent proof context. Then:*

(i) $\Gamma, \Delta \vdash_{\mathcal{N}'(Kl)} \varphi$ *iff* $\varphi \in (\Gamma, \Delta)$*;*
(ii) $\rho_1 \sqsupset \rho_2 \in \Delta$ *iff* $\rho_1 \in \Delta$ *implies* $\rho_2 \in \Delta$*;*
(iii) $\forall b.\, \rho \in \Delta$ *iff* $\rho[c/b] \in \Delta$ *for all* c*;*
(iv) $u : A \supset B \in \Gamma$ *iff* $u : A \in \Gamma$ *implies* $u : B \in \Gamma$*;*
(v) $u_1 : \mathsf{G}A \in \Gamma$ *iff* $u_1 < u_2 \in \Delta$ *implies* $u_2 : A \in \Gamma$ *for all* u_2*;*
(vi) $u_1 : \mathsf{H}A \in \Gamma$ *iff* $u_2 < u_1 \in \Delta$ *implies* $u_2 : A \in \Gamma$ *for all* u_2*.*

Proof. We treat only some cases, the others are similar and follow by maximality and consistency of (Γ, Δ).

(i) The proof is analogous for rwffs and lwffs, we see the first case.
($\Leftarrow$) If $\varphi \in (\Gamma, \Delta)$, then trivially $\Gamma, \Delta \vdash_{\mathcal{N}'(Kl)} \varphi$.
($\Rightarrow$) Consider an rwff φ such that $\varphi \notin (\Gamma, \Delta)$. Then, by Definition 4.20, $\sim \varphi \in (\Gamma, \Delta)$. It follows trivially that $\Gamma, \Delta \vdash_{\mathcal{N}'(Kl)} \sim \varphi$ holds. By hypothesis, $\Gamma, \Delta \vdash_{\mathcal{N}'(Kl)} \varphi$ and thus by using $\sim E$ we get $\Gamma, \Delta \vdash_{\mathcal{N}'(Kl)} \emptyset$, that contradicts the consistency of (Γ, Δ).

(v) ($\Leftarrow$) Suppose $u_1 : \mathsf{G}A \notin \Gamma$ and $u_2 : A \in \Gamma$ for every u_2 such that $u_1 < u_2 \in \Delta$. Then, by maximality of (Γ, Δ), $u_1 : \neg \mathsf{G}A \in \Gamma$. Now suppose there exists a u_3 such that $u_1 < u_3 \in \Delta$ and $u_3 : \neg A \in \Gamma$. Then, by hypothesis, we know $u_3 : A \in \Gamma$ and this leads to a contradiction. Otherwise, if such a u_3 does not exist, we can conclude $u_1 : \mathsf{G}A \in \Gamma$ that leads to a contradiction as well.

($\Rightarrow$) We show it by contraposition. Suppose $u_1 : \mathsf{G}A \in \Gamma$, $u_1 < u_2 \in \Delta$ and $u_2 : A \notin \Gamma$. By maximality of (Γ, Δ), we have $u_2 : \neg A \in \Gamma$. Then the following is an $\mathcal{N}'(Kl)$ proof that shows (Γ, Δ) is inconsistent.

$$\dfrac{\dfrac{u_1 : \mathsf{G}A \qquad u_1 < u_2}{u_2 : A}\,\mathsf{G}E \qquad u_2 : \neg A}{u : \bot}\,\neg E$$

□

Our construction of maximally consistent proof contexts (Lemma 4.21) does not exclude the presence of two labels b and c that are related by the relation $b = c$. Now we want to derive a model from such a construction. Since we know from Definition 4.15 that $\mathcal{M}, \lambda \models_{Kl} b = c$ holds only if $\lambda(b) = \lambda(c)$, we need to state an equivalence relation between labels on which the function λ can be defined.

Definition 4.23. *Let $C = (\Gamma, \Delta)$ be a maximally consistent proof context and L^C the set of labels occurring in it, we define the binary relation $\equiv^C$ on L^C as follows: for every $u_1, u_2 \in L^C$,*

$$u_1 \equiv^C u_2 \quad \textit{iff} \quad u_1 = u_2 \in \Delta.$$

Proposition 4.24. *Given a maximally consistent proof context C, the relation $\equiv^C$ is an equivalence relation.*

Proof. It follows trivially by the maximality of C and by the rules *refl* $=$, *mon*, *irrefl* $<$ and *conn*.

□

It follows from Proposition 4.24 that every maximally consistent proof context C determines a partition of the set L^C of labels occurring in it. In the following, we will also use the notation $[u]^C$ to indicate the equivalence class containing the label u, i.e.

$$[u]^C = \{u' \mid u \equiv^C u'\}.$$

Definition 4.25. *Let $C = (\Gamma, \Delta)$ be a maximally consistent proof context and L^C be the set of labels occurring in it. We define the* canonical model $\mathcal{M}^C = (\mathcal{W}^C, \prec^C, \mathcal{V}^C)$ *as follows:*

- $\mathcal{W}^C = \{[u]^C \mid u \in L^C\}$;
- $([u_i]^C, [u_j]^C) \in \prec^C$ *iff* $u_i < u_j \in \Delta$;
- $\mathcal{V}^C([u]^C, p) = 1$ *iff* $u : p \in \Gamma$.

We define the canonical interpretation $\lambda^C : L^C \to \mathcal{W}^C$ *as follows:*

$$\lambda^C(u) = [u]^C \textit{ for every } u \in L^C.$$

Remark 4.26. Note that in the previous definition $\prec^C$ and $\mathcal{V}^C$ are well defined, since it is easy to verify that for every $u_1, u_2 \in L^C$ it holds:

- $u_1 \equiv^C u_2$ implies for every $u_3 \in L^C$, $u_1 < u_3 \in \Delta$ iff $u_2 < u_3 \in \Delta$;
- $u_1 \equiv^C u_2$ implies for every $u_3 \in L^C$, $u_3 < u_1 \in \Delta$ iff $u_3 < u_2 \in \Delta$;

$$\dfrac{\dfrac{\dfrac{}{\forall b.c.\, b < c \sqcup b = c \sqcup c < b}\, conn}{b < c \sqcup b = c \sqcup c < b}\,\forall E \qquad \Pi \qquad \dfrac{[c < b]^1 \qquad \sim (c < b)}{\emptyset}\,\sim E}{\emptyset}\,\sqcup E^1$$

where Π is

$$\dfrac{[b < c \sqcup b = c]^1 \qquad \dfrac{[b < c]^2 \qquad \sim (b < c)}{\emptyset}\,\sim E \qquad \dfrac{[b = c]^2 \qquad \sim (b = c)}{\emptyset}\,\sim E}{\emptyset}\,\sqcup E^2$$

Fig. 4.8. Proof for connectedness of canonical models.

- $u_1 \equiv^C u_2$ implies for every $p \in \mathcal{P}$, $u_1 : p \in \Gamma$ iff $u_2 : p \in \Gamma$.

Proposition 4.27. *Given a maximally consistent proof context $C = (\Gamma, \Delta)$, the canonical model $\mathcal{M}^C$ is a Kripke model for Kl.*

Proof. It suffices to show that $\mathcal{M}^C$ is irreflexive, transitive and connected.

Suppose there exist three worlds $\mathcal{W}_1$, $\mathcal{W}_2$, and $\mathcal{W}_3$ in $\mathcal{W}^C$ such that $(\mathcal{W}_1, \mathcal{W}_2) \in \prec^C$ and $(\mathcal{W}_2, \mathcal{W}_3) \in \prec^C$, but $(\mathcal{W}_1, \mathcal{W}_3) \notin \prec^C$. By definition 4.25, this implies there exist at least three labels b, c and d such that $\lambda(b) = \mathcal{W}_1$, $\lambda(c) = \mathcal{W}_2$, $\lambda(d) = \mathcal{W}_3$, $b < c \in \Delta$ and $c < d \in \Delta$, but $b < d \notin \Delta$, i.e. by the maximality of C, $\sim (b < d) \in \Delta$. But this leads to the inconsistency of (Γ, Δ), as shown by the following derivation.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\forall b.c.d.\, (b < c \sqcap c < d) \sqsupset b < d}\, trans<}{(b < c \sqcap c < d) \sqsupset b < d}\,\forall E \qquad \dfrac{b < c \quad c < d}{b < c \sqcap c < d}\,\sqcap I}{b < d}\,\sqsupset E \qquad \sim (b < d)}{\emptyset}\,\sim E$$

Connectedness of $\mathcal{M}^C$ can be proved in a similar way by using the rule *conn*. Suppose there exist two distinct worlds $\mathcal{W}_1$ and $\mathcal{W}_2$ in $\mathcal{W}^C$ such that $(\mathcal{W}_1, \mathcal{W}_2) \notin \prec^C$ and $(\mathcal{W}_2, \mathcal{W}_1) \notin \prec^C$. By definition 4.25, this implies there exist at least two labels b and c such that $\lambda(b) = \mathcal{W}_1$, $\lambda(c) = \mathcal{W}_2$, $b = c \notin \Delta$, $b < c \notin \Delta$ and $c < b \notin \Delta$, i.e. by the maximality of C, $\sim (b = c) \in \Delta$, $\sim (b < c) \in \Delta$ and $\sim (c < b) \in \Delta$. But this leads to the inconsistency of (Γ, Δ), as shown by the derivation in Figure 4.8.

Irreflexivity of $\mathcal{M}^C$ can be shown in a similar way.

□

Lemma 4.28. *Let $C = (\Gamma, \Delta)$ be a maximally consistent proof context, $\mathcal{M}^C$ the canonical model and λ^C the canonical interpretation built on C as in Definition 4.25. Then:*

(i) $\rho \in \Delta$ iff $\Gamma, \Delta \models^{\mathcal{M}^C, \lambda^C}_{Kl} \rho$;

(ii) $u : A \in \Gamma$ *iff* $\Gamma, \Delta \models^{\mathcal{M}^C,\lambda^C}_{Kl} u : A$.

Proof. (i) ($\Rightarrow$) By hypothesis, $\rho \in \Delta$. Then, if we assume $\mathcal{M}^C, \lambda^C \models_{Kl} (\Gamma, \Delta)$, it immediately follows $\mathcal{M}^C, \lambda^C \models_{Kl} \rho$.
($\Leftarrow$) By hypothesis, $\Gamma, \Delta \models^{\mathcal{M}^C,\lambda^C}_{Kl} \rho$. Let us suppose $\rho \notin \Delta$. By maximality of (Γ, Δ), it follows $\sim \rho \in \Delta$. Then we have also $\Gamma, \Delta \models^{\mathcal{M}^C,\lambda^C}_{Kl} \sim \rho$ (see direction ($\Rightarrow$)). But, since we have by hypothesis $\Gamma, \Delta \models^{\mathcal{M}^C,\lambda^C}_{Kl} \rho$, this yields the absurd $\Gamma, \Delta \models^{\mathcal{M}^C,\lambda^C}_{Kl} \emptyset$.
(ii) The proof for labeled formulas is analogous.

□

Theorem 4.29. $\mathcal{N}'(Kl) = \mathcal{N}(Kl_L) + \mathcal{N}(Kl_R) + \mathcal{N}(Kl_G)$ *is complete, i.e. it holds:*

(i) if $\Gamma, \Delta \nvdash w : A$ *, then there exist a Kl model* $\mathcal{M}^C$ *and an interpretation* λ^C *such that* $\Gamma, \Delta \not\models^{\mathcal{M}^C,\lambda^C} w : A$*;*
(ii) if $\Gamma, \Delta \nvdash \rho$ *, then there exist a Kl model* $\mathcal{M}^C$ *and an interpretation* λ^C *such that* $\Gamma, \Delta \not\models^{\mathcal{M}^C,\lambda^C} \rho$*.*

Proof. (i) If $\Gamma, \Delta \nvdash w : A$, then $(\Gamma \cup \{w : \neg A\}, \Delta)$ is consistent; otherwise there exists a w_i such that $\Gamma \cup \{w : \neg A\}, \Delta \vdash_{\mathcal{N}'(Kl)} w_i : \bot$, and then $\Gamma, \Delta \vdash_{\mathcal{N}'(Kl)} w : A$. Therefore, by Lemma 4.21, $(\Gamma \cup \{w : \neg A\}, \Delta)$ is included in a maximally consistent proof context $C = ((\Gamma \cup \{w : \neg A\})^*, \Delta^*)$. Let $\mathcal{M}^C$ be the canonical model for C. It suffices to find an interpretation according to which $\mathcal{M}^C$ is not a model for $w : A$. By Lemma 4.28, $(\Gamma \cup \{w : \neg A\})^*, \Delta^* \models^{\mathcal{M}^C,\lambda^C}_{Kl} w : \neg A$, where $\mathcal{M}^C$ is a Kl model by Proposition 4.27. It follows $\Gamma \cup \{w : \neg A\})^*, \Delta^* \not\models^{\mathcal{M}^C,\lambda^C} w : A$, and thus $\Gamma, \Delta \not\models^{\mathcal{M}^C,\lambda^C} w : A$.
(ii) We can repeat the same proof for relational formulas. If $\Gamma, \Delta \nvdash \rho$, then $(\Gamma, \Delta \cup \{\sim \rho\})$ is consistent. Then we can build a maximally consistent proof context $\Gamma^*, (\Delta \cup \{\sim \rho\})^*$ such that $\Gamma^*, (\Delta \cup \{\sim \rho\})^* \not\models^{\mathcal{M}^C,\lambda^C} \rho$, and thus $\Gamma, \Delta \not\models^{\mathcal{M}^C,\lambda^C} \rho$.

□

Completeness by axioms

It is possible to give an indirect proof of completeness (Theorem 4.29) by showing that all the rules of inference and axioms of $\mathcal{H}'(Kl)$ (Section 4.3.2) derivable in $\mathcal{N}'(Kl)$. In the following derivations, for simplicity, we will sometimes use derived operators and derived rules (see Figure 4.7), and exploit trivial equivalences between formulas implicitly.

A derivation for $(G1)$ is obtained as in the systems of Section 4.2. The following is a derivation of $(G2)$:

$$\dfrac{\dfrac{[t : \mathsf{PG}A]^1 \qquad \dfrac{[s : \mathsf{G}A]^2 \qquad [s < t]^2}{t : A}\,\mathsf{G}E}{t : A}\,\mathsf{P}E^2}{t : \mathsf{PG}A \supset A}\supset I^1$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\forall b.c.d.\,(b < c \sqcap c < d) \sqsupset b < d}\,trans<}{\forall c.d.\,(t < c \sqcap c < d) \sqsupset t < d}\,\forall E}{\forall d.\,(t < s \sqcap s < d) \sqsupset t < d}\,\forall E}{(t < s \sqcap s < r) \sqsupset t < r}\,\forall E \qquad \dfrac{[t < s]^2 \qquad [s < r]^3}{t < s \sqcap s < r}\,\sqcap I}{t < r}\,\sqsupset E \qquad [t : \mathsf{G}A]^1}{\dfrac{\dfrac{\dfrac{r : A}{s : \mathsf{G}A}\,\mathsf{G}I^3}{t : \mathsf{GG}A}\,\mathsf{G}I^2}{t : \mathsf{G}A \supset \mathsf{GG}A}\,\supset I^1}\,\mathsf{G}E$$

Fig. 4.9. Derivation of the axiom (*G3*).

The derivation for (*G3*) is shown in Figure 4.9, while the derivation for (*G4*) is in Figures 4.10 and 4.11. We omit here the derivations for the symmetrical axioms (*H1*)-(*H4*).

Normalization

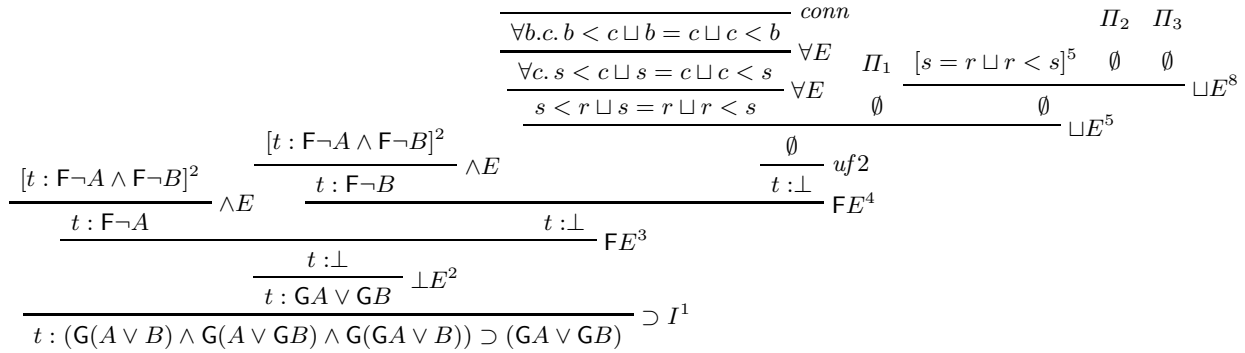
Derivations in normal form

We will now show that the system $\mathcal{N}'(Kl)$ possesses a number of useful normalization properties. To that end, we will follow the classical normalization process of [125] as much as possible, while some adaptations are inspired by [159]. We begin by simplifying the proofs by restricting the applications of some of the rules.

Lemma 4.30. *If $\Gamma, \Delta \vdash_{\mathcal{N}'(Kl)} \varphi$, then there exists a derivation of φ from (Γ, Δ) where: (i) the conclusions of applications of $\bot E$, $RAA_\emptyset$, and mon are atomic; (ii) mon is not applied to lwffs of the form $b : \bot$.*

Proof. (*i*) We show that any application of $\bot E$, $RAA_\emptyset$, and *mon* with a non-atomic conclusion can be replaced with a derivation in which such rules are applied only to formulas of smaller grade by the set of transformations given below. By iterating these transformations, we get a derivation of φ from Γ, Δ where the conclusions of applications of $\bot E$, $RAA_\emptyset$, and *mon* are atomic.

(1) First, we consider applications of $\bot E$. There are three possible cases, depending on whether the conclusion is $b : B \supset C$, $b : \mathsf{G}B$, or $b : \mathsf{H}B$. Note that in the following transformations we only show the part of the derivation where the reduction, denoted by $\rightsquigarrow$, actually takes place; the missing parts remain unchanged.



where Π_1 is:

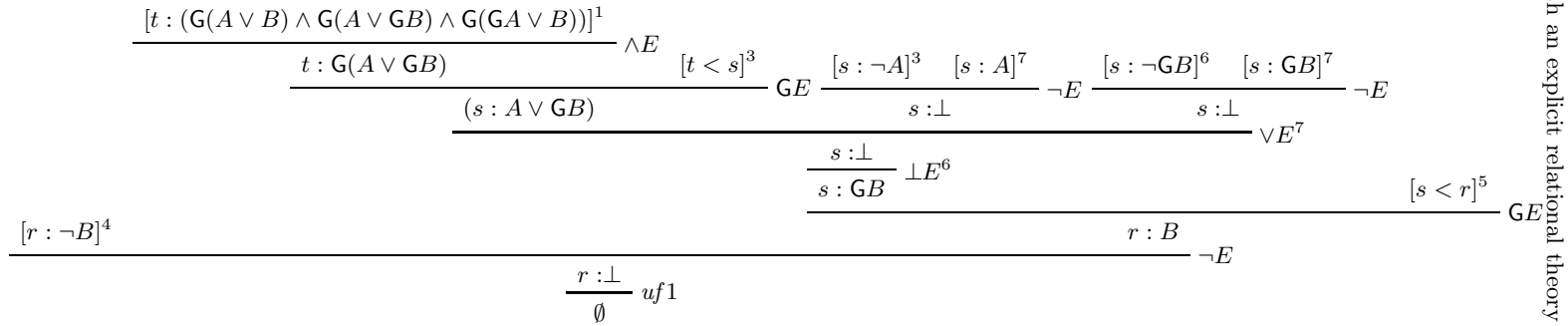


Fig. 4.10. Derivation of the axiom $(G4)$ (1/2).

Π_2 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{[t:(\mathsf{G}(A\vee B)\wedge\mathsf{G}(A\vee\mathsf{G}B)\wedge\mathsf{G}(\mathsf{G}A\vee B))]^1}{t:\mathsf{G}(A\vee B)}\wedge E \quad [t<s]^3}{s:A\vee B}\mathsf{G}E \quad \dfrac{[s:\neg A]^3 \quad [s:A]^{12}}{s:\bot}\neg E \quad \dfrac{[s:\neg B]^{11} \quad [s:B]^{12}}{s:\bot}\neg E}{s:\bot}\vee E^{12}}{s:B}\bot E^{11} \quad [s=r]^8}{r:B}\,mon \quad [r:\neg B]^4}{r:\bot}\neg E}{\emptyset}\,uf1$$

and Π_3 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{[t:(\mathsf{G}(A\vee B)\wedge\mathsf{G}(A\vee\mathsf{G}B)\wedge\mathsf{G}(\mathsf{G}A\vee B))]^1}{t:\mathsf{G}(\mathsf{G}A\vee B)}\wedge E \quad [t<r]^4}{r:\mathsf{G}A\vee B}\mathsf{G}E \quad \dfrac{[r:\neg\mathsf{G}A]^9 \quad [r:\mathsf{G}A]^{10}}{r:\bot}\neg E \quad \dfrac{[r:\neg B]^4 \quad [r:B]^{10}}{r:\bot}\neg E}{r:\bot}\vee E^{10}}{r:\mathsf{G}A}\bot E^9 \quad [r<s]^8}{s:A}\mathsf{G}E \quad [s:\neg A]^3}{s:\bot}\neg E}{\emptyset}\,uf1$$

Fig. 4.11. Derivation of the axiom $(G4)$ (2/2).

(Case 1)

$$\dfrac{\dfrac{[b : (B \supset C) \supset \bot]}{\dfrac{\Pi}{\dfrac{c : \bot}{b : B \supset C}\,\bot E}}}{}$$

$\rightsquigarrow$

$$\dfrac{\dfrac{\dfrac{[b : C \supset \bot]^2 \quad \dfrac{[b : B \supset C]^1 \quad [b : B]^3}{b : C}\,\supset E}{b : \bot}\,\supset E}{b : (B \supset C) \supset \bot}\,\supset I^1}{\dfrac{\dfrac{\Pi}{c : \bot}}{b : C}\,\bot E^2}\,\supset I^3$$

(Case 2)

$$\dfrac{\dfrac{[b : \mathsf{G}B \supset \bot]}{\dfrac{\Pi}{\dfrac{c : \bot}{b : \mathsf{G}B}\,\bot E}}}{}$$

$\rightsquigarrow$

$$\dfrac{\dfrac{\dfrac{[c : B \supset \bot]^2 \quad \dfrac{[b : \mathsf{G}B]^1 \quad [b < c]^3}{c : B}\,\mathsf{G}E}{c : \bot}\,\supset E}{b : \bot}\,\bot E}{\dfrac{\dfrac{\Pi}{c : \bot}}{c : B}\,\bot E^2}\,\mathsf{G}I^3}{b : \mathsf{G}B}\,\supset I^1$$

Case 3 concerns formulas of the form $c : \mathsf{H}A$; it is analogous to the previous one and we omit the reduction for it.

(2) Applications of $RAA_\emptyset$ can be reduced to applications on formulas of lower grade, following an approach analogous to that of $\bot E$. It is easy to see that in this case, we can also restrict to applications of $RAA_\emptyset$ in which the conclusion is not $\emptyset$. We have to consider two possibilities: formulas of the form $\rho_1 \sqsupset \rho_2$ and formulas of the form $\forall b.\, \rho$. We consider only the second case, since the first one is analogous to the case of implication for labeled formulas:

$$\dfrac{\dfrac{[\forall b.\, \rho \sqsupset \emptyset]}{\dfrac{\Pi}{\dfrac{\emptyset}{\forall b.\, \rho}\,RAA_\emptyset}}}{}$$

$\rightsquigarrow$

$$\dfrac{\dfrac{\dfrac{[\rho \sqsupset \emptyset]^1}{\forall b.\, \rho \sqsupset \emptyset}\,\forall I}{\dfrac{\Pi}{\dfrac{\emptyset}{\rho}\,RAA^1_\emptyset}}}{\forall b.\, \rho}\,\forall I$$

(3) Finally, we consider applications of the rule *mon*. We have five cases depending on the form of the formula that is the major premise of the *mon* application:

(a) $b : A \supset B$

(b) $b : \mathsf{G}A$
(c) $b : \mathsf{H}A$
(d) $\rho_1 \sqsupset \rho_2$
(e) $\forall b.\, \rho$

(Case a)

$$\dfrac{\dfrac{b : A \supset B \quad b = c}{c : A \supset B}\, mon}{} \rightsquigarrow \dfrac{\dfrac{b : A \supset B \qquad \dfrac{[c : A]^1 \quad b = c}{b : A}\, mon}{b : B} \supset E \qquad b = c}{\dfrac{c : B}{c : A \supset B} \supset I^1}\, mon$$

(Case b)

$$\dfrac{b : \mathsf{G}A \quad b = c}{c : \mathsf{G}A}\, mon \rightsquigarrow \dfrac{b : \mathsf{G}A \qquad \dfrac{[c < d]^1 \quad b = c}{b < d}\, mon}{\dfrac{d : A}{c : \mathsf{G}A}\, \mathsf{G}I^1}\, \mathsf{G}E$$

(Case e)

$$\dfrac{\forall b.\, \rho \quad c = d}{\forall b.\, \rho[d/c]}\, mon \rightsquigarrow \dfrac{\dfrac{\forall b.\, \rho}{\rho}\, \forall E \quad c = d}{\dfrac{\rho[d/c]}{\forall b.\, \rho[d/c]}\, \forall I}\, mon$$

The case (c) is analogous to (b), while the transformation for the case (d) is as in (a) where $\sqsupset$ plays the role of $\supset$.

(ii) We show that every application of *mon* on a lwff of the form $b :\bot$ can be replaced by an application of $\bot E$ that does not discharge any assumption:

$$\dfrac{\begin{array}{cc} \Pi & \Pi' \\ b :\bot & b = c \end{array}}{c :\bot}\, mon \quad \rightsquigarrow \quad \dfrac{\begin{array}{c} \Pi \\ b :\bot \end{array}}{c :\bot}\, \bot E$$

□

The system obtained from $\mathcal{N}'(Kl)$ by restricting the rules $\bot E$, $RAA_{\emptyset}$, and *mon* according to this lemma is equivalent to $\mathcal{N}'(Kl)$. From now on, we will thus consider only this restricted system and keep calling it $\mathcal{N}'(Kl)$.

The natural deduction systems given in [159] for families of modal and relevance logics are based on a strict separation between the labeled and the relational subsystems (i.e. derivations of lwffs can depend on derivations of rwffs, but not vice versa). This separation is possible thanks to the restriction to relational theories that are Horn theories. Our system $\mathcal{N}'(Kl)$ does not allow for such a separation, since the rules for universal falsum let relational derivations depend also on labeled ones. Thus, more complex derivations are possible, which implies that with respect to [159] we need to consider more forms of detours and hence more forms of reductions. We adapt to our case the definitions given in Section 3.2.

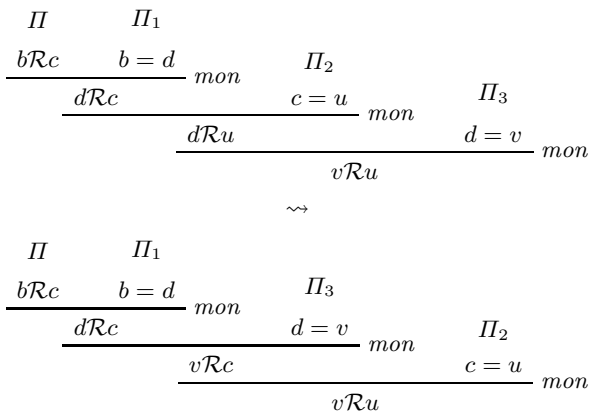


Fig. 4.12. Rule permutation for the ordering of *mon* applications.

Definition 4.31. *We say that a formula φ is a* maximum formula *in a derivation when it is both the conclusion of an introduction rule and the major premise of an elimination rule.*

We define the notion of label position *for labels occurring in a formula φ to which the rule mon is applied. By the restrictions of Lemma 4.30, φ can have the form $(i)\, b : p$, $(ii)\, b < c$, or $(iii)\, b = c$. We say that b has label position 1 in (i), (ii) and (iii), and c has label position 2 in (ii) and (iii).*

A derivation is in pre-normal form *(is a* pre-normal derivation*) if it has no maximum formulas and in every sequence of mon applications, all the applications which concern variables with the same label position occur consecutively.*

The notion of pre-normal derivation embodies the elimination of standard detours (given by a couple of introduction/elimination rule applications on the same connective or operator) and an ordering of *mon* applications that aims at eliminating *mon* detours, i.e. two or more applications of *mon* which concern variables with the same label position. Note that, since *mon* is only applied to atomic formulas of the form described above, once we have eliminated maximum formulas, the case of a sequence of *mon* applications is the only case in which we can have this kind of detour.

Lemma 4.32. *Every derivation in $\mathcal{N}'(Kl)$ reduces to a derivation in pre-normal form.*

Proof. We follow the procedure based on proper reductions used in [159] and we only treat the cases $\supset I/\supset E$, $\mathsf{G}I/\mathsf{G}E$ and $\forall I/\forall E$. The transformations for the detours $\sqsupset I/\sqsupset E$ and $\mathsf{H}I/\mathsf{H}E$ can be easily inferred from these. Any formula φ in a derivation is the root of a tree of rule applications leading back to assumptions. We call *side formulas* of φ the formulas in this tree other than φ. In order to eliminate maximum formulas from a derivation, it suffices to apply the transformations listed below, picking in the set of maximum formulas the formula with the highest grade that has only maximum formulas of lower grade as side formulas, and iterating this process until there are no more maximum formulas in the proof. The process

ends because at every step no new maximum formula as large as (or larger than) the eliminated one is introduced.

$$(i)\quad \dfrac{\dfrac{\begin{array}{c}[b:A]\\ \Pi_1\\ b:B\end{array}}{b:A\supset B}\supset I \qquad \begin{array}{c}\Pi_2\\ b:A\end{array}}{b:B}\supset E \qquad \rightsquigarrow \qquad \begin{array}{c}\Pi_2\\ b:A\\ \Pi_1\\ b:B\end{array}$$

$$(ii)\quad \dfrac{\dfrac{\begin{array}{c}[b<c]\\ \Pi\\ c:A\end{array}}{b:\mathsf{G}A}\mathsf{G}I \qquad b<d}{d:A}\mathsf{G}E \qquad \rightsquigarrow \qquad \begin{array}{c}b<d\\ \Pi[d/c]\\ d:A\end{array}$$

$$(iii)\quad \dfrac{\dfrac{\begin{array}{c}\Pi\\ \rho\end{array}}{\forall b.\,\rho}\forall I}{\rho[c/b]}\forall E \qquad \rightsquigarrow \qquad \begin{array}{c}\Pi[c/b]\\ \rho[c/b]\end{array}$$

Finally, in Fig. 4.12 we show how to permute applications of rules in order to get a derivation where, given a sequence of *mon* applications, the ones on the same label position occur one immediately below the other. We denote with $\mathcal{R}$ a relational symbol that can stay both for $<$ and for $=$. In the derivation on the left, the first and the third application of *mon* refer to the same label position and thus are moved one immediately below the other. The derivations obtained in this way will then be further simplified during the normalization process.

□

Definition 4.33. *We call* falsum-rules *the rules $\bot E$, $RAA_\emptyset$, $uf1$, and $uf2$. We say that a formula φ is a* redundant formula *in a derivation when: (i) φ is both the conclusion and the premise of a falsum-rule; or (ii) φ is both the conclusion and the major premise of a mon carrying out two substitutions in the same label position.*

A derivation is in normal form *(is a* normal derivation*) iff it is in pre-normal form and does not contain any redundant formula.*

Theorem 4.34. *Every derivation in $\mathcal{N}'(Kl)$ reduces to a derivation in normal form.*

Proof. First, we observe that by Lemma 4.32 we can obtain a derivation in pre-normal form. Now let us show how to remove redundant formulas. We know from Lemma 4.30 that every application of a falsum-rule has an atomic formula as a conclusion. Thus it is sufficient to consider the following transformations:

$$(i)\ \dfrac{\dfrac{\begin{array}{c}\Gamma\ \Delta\\ \Pi\\ b:\bot\end{array}}{c:\bot}\,\bot E}{d:A}\,\bot E \quad\rightsquigarrow\quad \dfrac{\begin{array}{c}\Gamma\ \Delta\\ \Pi\\ b:\bot\end{array}}{d:A}\,\bot E$$

where A is $\bot$ or an atomic formula. Note that if the formula $d:A \supset \bot$ is contained in Γ and discharged by the second application of $\bot E$ in the derivation on the left, then the same can be done in the derivation on the right.

$$(ii)\ \dfrac{\dfrac{\begin{array}{c}\Pi\\ b:\bot\end{array}}{c:\bot}\,\bot E}{\emptyset}\,uf1 \quad\rightsquigarrow\quad \dfrac{\begin{array}{c}\Pi\\ b:\bot\end{array}}{\emptyset}\,uf1$$

$$(iii)\ \dfrac{\dfrac{\begin{array}{c}\Pi\\ b:\bot\end{array}}{\emptyset}\,uf1}{c:\bot}\,uf2 \quad\rightsquigarrow\quad \dfrac{\begin{array}{c}\Pi\\ b:\bot\end{array}}{c:\bot}\,\bot E$$

$$(iv)\ \dfrac{\dfrac{\begin{array}{c}\Pi\\ \emptyset\end{array}}{b:\bot}\,uf2}{\emptyset}\,uf1 \quad\rightsquigarrow\quad \begin{array}{c}\Pi\\ \emptyset\end{array}$$

For the rule *mon*, given the ordering of *mon* applications obtained by permutations defined in Lemma 4.32, the only case we have to treat is when two applications of *mon* working on the same label position of a formula occur consecutively. Then we simply exploit the transitivity of $=$ (obtained by using *mon*). Note that, by Lemma 4.30, in the following reduction φ is an atomic formula.

$$\dfrac{\dfrac{\begin{array}{cc}\Pi_1 & \Pi_2\\ \varphi & b=c\end{array}}{\varphi[c/b]}\,mon \quad \begin{array}{c}\Pi_3\\ c=d\end{array}}{\varphi[d/b]}\,mon \quad\rightsquigarrow\quad \dfrac{\begin{array}{cc}\Pi_1 & \dfrac{\begin{array}{cc}\Pi_2 & \Pi_3\\ b=c & c=d\end{array}}{b=d}\,mon\\ \varphi & \end{array}}{\varphi[d/b]}\,mon$$

□

Normal derivations in $\mathcal{N}'(Kl)$ have a well-defined structure that has a number of desirable properties. In particular, there is an ordering on the application of the rules, which we can exploit to prove a subformula property for our system. To that end, we adapt the standard definitions of subformula and track as follows:

Definition 4.35. *B is a* subformula *of A iff (i) A is B; (ii) A is $A_1 \supset A_2$ and B is a subformula of A_1 or A_2; (iii) A is $\mathsf{G}A_1$ and B is a subformula of A_1; or (iv) A is $\mathsf{H}A_1$ and B is a subformula of A_1. We say that $c : B$ is a* subformula *of $b : A$ iff B is a subformula of A.*

ρ_2 is a subformula *of ρ_1 iff (i) ρ_1 is ρ_2; (ii) ρ_1 is $\rho_1' \sqsupset \rho_1''$ and ρ_2 is a subformula of ρ_1' or ρ_1''; or (iii) ρ_1 is $\forall b.\, \rho$ and ρ_2 is a subformula of ρ.*

Given a derivation Π in $\mathcal{N}'(Kl)$, a track *in Π is a sequence of formulas $\varphi_1, \ldots, \varphi_n$ such that:*

(i) φ_1 is an assumption of Π, an axiom, or the conclusion of a universal falsum rule ($uf1$ or $uf2$);

(ii) φ_i stands immediately above φ_{i+1} and is the major (or the only) premise of a rule for $1 \leq i < n$;

(iii) φ_n is the conclusion of Π, the premise of a universal falsum rule, or the minor premise of a rule.

We call a track $\varphi_1, \ldots, \varphi_n$ a labeled track *when each φ_i is an lwff and a* relational track *when each φ_i is an rwff.*

In other words, a track can only pass through the major premises of rules and it ends at the first minor premise of a rule, or at an application of universal falsum, or at the conclusion of Π. The following lemmas formalize properties of the structure of the tracks and specify the way in which the tracks are linked one to each other.

Lemma 4.36. *Let Π be a normal derivation, and let t be a track $\varphi_1, \ldots, \varphi_n$ in Π. Then t consists of three (possibly empty) parts: (1) an* elimination part*, (2) a* central part*, and (3) an* introduction part *(see Figure 4.13) where:*

(i) each φ_i in the elimination part is the major premise of an elimination rule and contains φ_{i+1} as a subformula;

(ii) each φ_j in the introduction part except the last one is the premise of an introduction rule and is a subformula of φ_{j+1};

(iii) each φ_k in the central part is atomic and is the premise of a falsum-rule or the major premise of a mon;

(iv) the central part contains at most one application of falsum-rules;

(v) tracks originating from an application of $uf1$ or $uf2$ have an empty elimination part;

(vi) tracks ending in an application of $uf1$ or $uf2$ have an empty introduction part.

Proof. (i) and (ii) follow from the absence of maximum formulas in a normal derivation: in a track t, no introduction rule application can precede an application of an elimination rule. In other words, a track in a normal derivation is such that the elimination part (when not empty) starts with a non-atomic formula and consists of some applications of elimination-rules; if the elimination part ends with an atomic formula, then the central part (when not empty) consists of some applications of rules whose conclusion is still an atomic formula; the introduction part (when not empty) starts with an atomic formula and consists of some applications of introduction rules (see Fig 4.13).

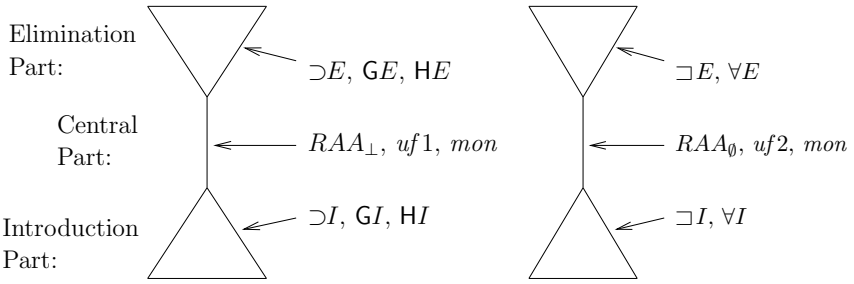


Fig. 4.13. The structure of a labeled track (left) and that of a relational track (right).

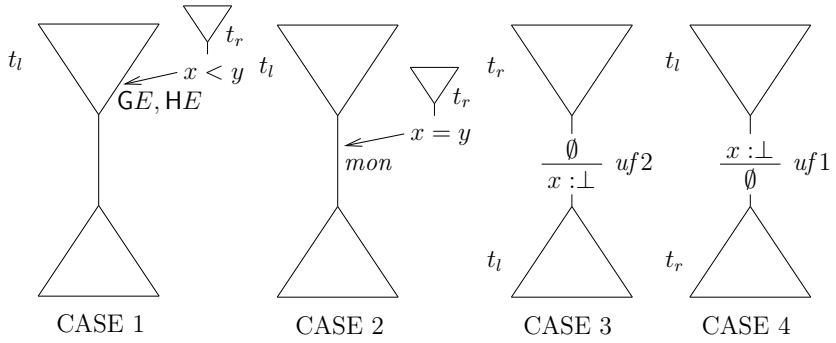


Fig. 4.14. Possible connections between labeled tracks t_l and relational tracks t_r.

(iii) comes from the fact that in a normal derivation a falsum-rule and the *mon*-rule can be applied only to atomic formulas.

(iv) follows directly from the absence of redundant formulas in a normal derivation (see Theorem 4.34).

For (v) and (vi), observe that tracks originating from an application of $uf1$ or $uf2$ start with an atomic formula and thus cannot have an elimination part, while tracks ending in an application of $uf1$ or $uf2$ end with an atomic formula and thus their introduction part must be empty.

□

Lemma 4.37. *Let t_l be a labeled track and t_r a relational track in a derivation Π. Then t_l and t_r can be connected in one of the following ways (shown in Figure 4.14):*

(i) the last formula in t_r is the minor premise of a $\mathsf{G}E$ or of a $\mathsf{H}E$ whose major premise is a formula in the elimination part of t_l;

(ii) the last formula in t_r is the minor premise of a mon whose major premise is a formula in the central part of t_l;

(iii) t_r ends with an application of $uf2$ and the conclusion of that application is the first formula in t_l;

(iv) t_l ends with an application of $uf1$ and the conclusion of that application is the first formula in t_r.

Proof. The statement follows trivially by observing that $\mathsf{G}E$, $\mathsf{H}E$, *mon*, *uf*1, and *uf*2 are the only rules that mix labeled and relational formulas and that, by Lemma 4.36, such rules can be applied only in a specific part of a track.

□

The subformula property

To prove a subformula property for $\mathcal{N}'(Kl)$, we adapt further standard definitions:

Definition 4.38. *Given a derivation Π in $\mathcal{N}'(Kl)$, the* main thread *is the sequence $t_1, \ldots, t_n$ of tracks such that: (1) the first formula in t_1 is an assumption or an axiom; (2) t_i and t_{i+1} are connected by means of an application of uf*1 *or uf*2*, for $1 \leq i \leq (n-1)$; and (3) the last formula in t_n is the conclusion of Π.*

Let Π be a derivation of φ from (Γ, Δ) in $\mathcal{N}'(Kl)$, S_L be the set of subformulas of the formulas in Γ (or in $\Gamma \cup \{\varphi\}$ if φ is a labeled formula), and S_R be the set of subformulas of the formulas in $\Delta \cup Ax$ (or in $\Delta \cup Ax \cup \{\varphi\}$ if φ is a relational formula), where Ax is the set of axioms used in Π. We say that Π enjoys the subformula property *iff*

1. *for all lwffs $c : B$ used in the derivation Π:*
   (i) $B \in S_L$; or
   (ii) B is an assumption $D \supset \bot$ discharged by an application of $\bot E$ where $D \in S_L$; or
   (iii) B is an occurrence of $\bot$ obtained by $\supset E$ from an assumption $D \supset \bot$ discharged by an application of $\bot E$, where $D \in S_L$; or
   (iv) B is an occurrence of $\bot$ obtained by an application of $\bot E$ that does not discharge any assumption; or
   *(v) B is an occurrence of $\bot$ obtained by an application of uf*2*;*
2. *for all rwffs ρ used in the derivation Π:*
   (i) $\rho \in S_R$; or
   (ii) ρ is an assumption $\rho_1 \sqsupset \bot$ discharged by an application of $RAA_\emptyset$ where $\rho_1 \in S_R$; or
   (iii) ρ is an occurrence of $\emptyset$ obtained by $\sqsupset E$ from an assumption $\rho' \sqsupset \emptyset$ discharged by an application of $RAA_\emptyset$, where $\rho' \in S_R$; or
   *(iv) ρ is an occurrence of $\emptyset$ obtained by an application of uf*1*; or*
   (v) ρ is obtained by an application of mon.

Lemma 4.39. *Every normal derivation in $\mathcal{N}'(Kl)$ satisfies the subformula property.*

Proof. This follows immediately from the standard proof [125], which is based on the introduction of an ordering of the tracks in a normal derivation depending on their distance from a main thread. In our case, a main thread contains not only labeled formulas and we have to consider more cases than in the standard proof, given that the central part of a track can have a more complex structure (as it can also contain applications of *uf*1, *uf*2, and *mon*).

□

This lemma shows that although normal derivations in $\mathcal{N}'(Kl)$ have a more complex structure than normal derivations in natural deduction systems for classical logic [125] and natural deduction systems for families of modal and relevance logics [159], they have still a well-defined structure and satisfy a subformula property. It is important to remark that the special cases added to the definition of subformula property (i.e. formulas can be derived by applications of $uf1$, $uf2$, or mon) do not compromise automatic proof search completely, given that such cases can occur only in a limited section of a normal derivation (i.e. the central part of a track).

We also note that the presence of axioms (and in particular the fact that they are expressed in a full first-order language) makes our proof of normalization more complex and our results weaker. Thus, it is not possible to use it as a means to show the consistency of the system or the validity of an interpolation theorem, as can be done for systems in [159], where relational properties are expressed by Horn rules and we have only atomic axioms.

4.3.3 Systems for axiomatic extensions of *Kl*

The basic linear tense logic Kl leaves unanswered many fundamental natural questions about the structure of time. However, the labeling framework allows us to express several further relational properties in a straightforward and clean way, i.e. by only adding the corresponding relational axioms to the relational subsystem. In particular, we will now show how to extend $\mathcal{N}'(Kl)$ to capture the extensions of Kl described in Section 2.3.2, i.e., Kl with:

- unbounded time;
- a first/final point;
- dense time;
- discrete time.

To help the reader, we recall in Figure 4.15 the axioms corresponding to such extensions.

Kl with unbounded time

In the case of an unbounded flow of time, we can add two relational axioms corresponding to the axioms for left and right seriality given in Figure 4.15:

$$\frac{}{\forall b.\exists c.\, c < b}\ lser \qquad \frac{}{\forall b.\exists c.\, b < c}\ rser\,.$$

As an example, we show how to derive the axiom for *(right-seriality)*, where Π is some proof of $s : \top$ based on a proof of $\top$ or $A \vee \neg A$ in classical logic (see, e.g., [125, 152]):

$$\dfrac{\dfrac{\dfrac{}{\forall b.\exists c.\, b < c}\ rser}{\exists c.\, t < c}\ \forall E \qquad \dfrac{\begin{array}{cc} \Pi & \\ s : \top & [t < s]^1 \end{array}}{t : \mathsf{F}\top}\ \mathsf{F}I}{t : \mathsf{F}\top}\ \exists E^1$$

(having a first point)	$\mathsf{H}\bot \vee \mathsf{PH}\bot$	*(left-density)*	$\mathsf{P}A \supset \mathsf{PP}A$
(having a final point)	$\mathsf{G}\bot \vee \mathsf{FG}\bot$	*(right-density)*	$\mathsf{F}A \supset \mathsf{FF}A$
(left-seriality)	$\mathsf{P}\top$	*(left-discreteness)*	$(\mathsf{P}\top \wedge A \wedge \mathsf{G}A) \supset (\mathsf{PG}A)$
(right-seriality)	$\mathsf{F}\top$	*(right-discreteness)*	$(\mathsf{F}\top \wedge A \wedge \mathsf{H}A) \supset (\mathsf{FH}A)$

Fig. 4.15. Some axioms for extensions of Kl.

Kl with a first/final point

To express the existence of a first or of a final point, we can add the following axioms[13] to the relational sub-systems:

$$\frac{}{\exists b.\, \forall c.\ \sim (c < b)}\ \textit{first} \qquad \frac{}{\exists b.\, \forall c.\ \sim (b < c)}\ \textit{final}\,.$$

The two axioms do not affect each other; thus we can decide to add both or just one of them to the system, according to the logic we want to represent. A derivation of the axiom for *first point* is given in Figure 4.16.

Kl with dense time

Having a dense flow of time corresponds to require that between any two points we can find a third point:

$$\frac{}{\forall b.c.\, b < c \sqsupset \exists d.\, b < d \sqcap d < c}\ \textit{dens}\,.$$

Figure 4.17 shows the proof of the axiom for *(right-density)*; the proof for *(left-density)* can be obtained in a symmetrical way by using the same axiom (*dens*).

Kl with discrete time

Finally, we can express discreteness by means of the following axiomatic rules:

$$\frac{}{\forall b.c.\, b < c \sqsupset \exists d.\, d < c \sqcap \sim \exists u.\, (d < u \sqcap u < c)}\ \textit{ldiscr}$$

$$\frac{}{\forall b.c.\, b < c \sqsupset \exists d.\, b < d \sqcap \sim \exists u.\, (b < u \sqcap u < d)}\ \textit{rdiscr}\,.$$

In Figure 4.18, we show how to derive the axiom for *right-discreteness*.

Soundness and completeness

Theorem 4.40. *The extensions of $\mathcal{N}'(Kl)$ presented above are sound and complete with respect to the semantics of the corresponding logics.*

[13] The existence of a first (or a final) point is often expressed by adding a constant to the language. For example, we could introduce a constant 0 for the first point and an axiom stating that $\forall c.\ \sim (c < 0)$. We prefer not to modify the language and keep the treatment of this property closer to that of other ones.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\forall b.c.\, b < c \sqcup b = c \sqcup c < b}\, conn}{\forall c.\, t < c \sqcup t = c \sqcup t < b}\, \forall E}{t < s \sqcup t = s \sqcup s < t}\, \forall E \qquad \dfrac{\dfrac{[\forall c.\ \sim (c < s)]^2}{\sim (t < s)}\, \forall E \qquad [t < s]^3}{\emptyset}\, \sim E \qquad \dfrac{[t = s \sqcup s < t]^3 \qquad \Pi_1 \quad \emptyset \qquad \Pi_2 \quad \emptyset}{\emptyset}\, \sqcup E^4}{\emptyset}\, \sqcup E^3}{\exists b. \forall c.\ \sim (c < b)}\, first \qquad \emptyset}{\emptyset}\, \exists E^2}{\dfrac{\emptyset}{t : \bot}\, uf2}{t : \mathsf{H} \bot \vee \mathsf{PH} \bot}\, \bot E^1$$

where Π_1 is:

$$\dfrac{\dfrac{\dfrac{[t : \mathsf{P}\top \wedge \mathsf{HP}\top]^1}{\sim (t : \mathsf{P}\top)}\, \wedge E \qquad [t = s]^4}{(s : \mathsf{P}\top)}\, mon \qquad \dfrac{\dfrac{[\forall c.\ \sim (c < s)]^2}{\sim (q < s)}\, \forall E \qquad [q < s]^5}{\emptyset}\, \sim E}{s : \bot}\, uf2}{\dfrac{s : \bot}{\emptyset}\, uf1}\, \mathsf{P}E^5$$

and Π_2 is:

$$\dfrac{\dfrac{\dfrac{[t : \mathsf{P}\top \wedge \mathsf{HP}\top]^1}{t : \mathsf{HP}\top}\, \wedge E \qquad [s < t]^4}{s : \mathsf{P}\top}\, \mathsf{H}E \qquad \dfrac{\dfrac{[\forall c.\ \sim (c < s)]^2}{\sim (r < s)}\, \forall E \qquad [r < s]^6}{\emptyset}\, \sim E}{s : \bot}\, uf2}{\dfrac{s : \bot}{\emptyset}\, uf1}\, \mathsf{P}E^6$$

Fig. 4.16. Derivation of the modal axiom for first point.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\forall b.c.\, b < c \sqsupset \exists d.\, (b < d \sqcap d < c)}\, dens}{\forall c.\, (t < c \sqsupset \exists d.\, (t < d \sqcap d < c))}\, \forall E}{t < s \sqsupset \exists d.\, (t < d \sqcap d < s)}\, \forall E \qquad \dfrac{}{[t < s]^2}}{\exists d.\, (t < d \sqcap d < s)}\, \sqsupset E \qquad \dfrac{}{\Pi}\ \dfrac{}{\emptyset}}{\dfrac{\dfrac{\emptyset}{t : \bot}\, uf2}{t : \mathsf{F}\mathsf{F}A}\, \bot E^3}\, \exists E^4}{\dfrac{t : \mathsf{F}\mathsf{F}A}{t : \mathsf{F}A \supset \mathsf{F}\mathsf{F}A}\, \supset I^1}\, \mathsf{F}E^2 \qquad [t : \mathsf{F}A]^1$$

where Π is:

$$\dfrac{\dfrac{[s : A]^2 \qquad \dfrac{[t < r \sqcap r < s]^4}{r < s}\, \sqcap E}{r : \mathsf{F}A}\, \mathsf{F}I \qquad \dfrac{[t < r \sqcap r < s]^4}{t < r}\, \sqcap E}{t : \mathsf{F}\mathsf{F}A}\, \mathsf{F}I \qquad [t : \neg\mathsf{F}\mathsf{F}A]^3}{\dfrac{t : \bot}{\emptyset}\, uf1}\, \neg E$$

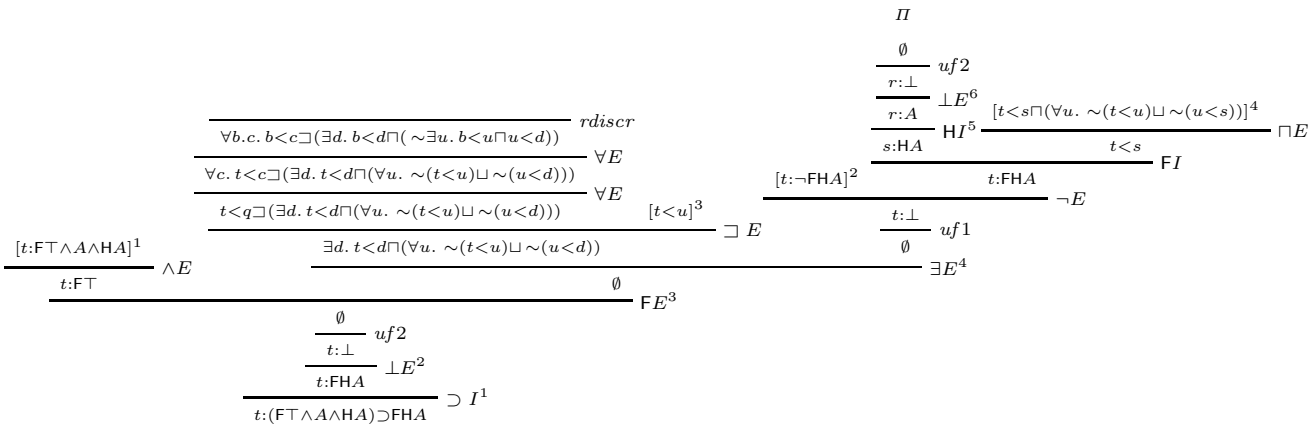
Fig. 4.17. Derivation of the modal axiom for right-density.

Proof. Soundness of the extended systems is straightforward. We have just to extend the proof of Theorem 4.17 by considering the axiomatic rules extending the relational sub-system. But they are trivially sound since they mirror the properties that the models of the extended logic are required to satisfy.

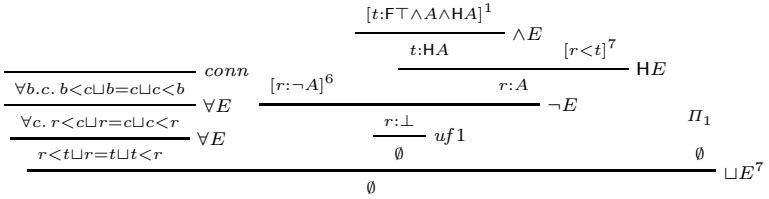
To show completeness, it suffices to extend the canonical model construction presented for $\mathcal{N}'(Kl)$ to consider also the new relational axioms. Alternatively, we can simply prove completeness by proving the corresponding Hilbert-style axioms. We have already proved the axioms for *right-seriality*, *having a first point*, *right-density* and *right-discreteness*. Derivations of the other axioms (*final point*, *left-seriality*, *left-density*, *left-discreteness*) are symmetrical and we thus omit them. □

4.3.4 Towards *LTL*

We have seen that natural deduction systems for several extensions of Kl can be given by extending the "base system" $\mathcal{N}'(Kl)$. This is not the case for all the possible extensions, however, as some properties, e.g. continuity or finite intervals, are second-order properties [156] and thus require an appropriate higher-order relational language. We now briefly discuss whether (and how) it is possible to extend $\mathcal{N}'(Kl)$ to capture a richer logic like (fragments of) *LTL*.



where Π is:



and Π_1 is:

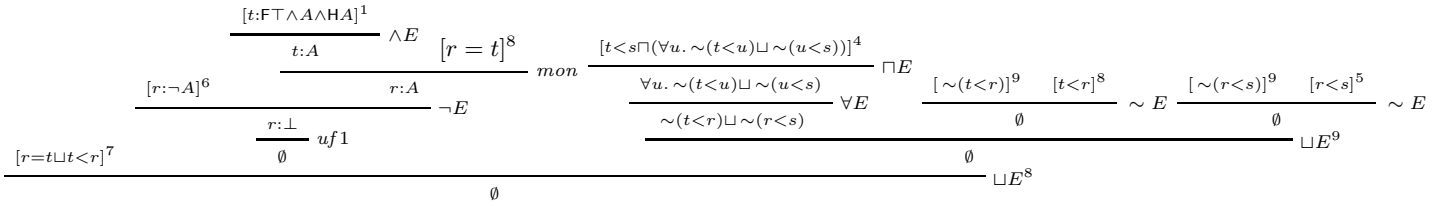


Fig. 4.18. Derivation of the modal axiom for right-discreteness.

MTL: a subset of *LTL*_

For brevity, we restrict our attention to future temporal operators only (but the extension to the past is straightforward) and begin by considering the system $\mathcal{N}'(Kl)$ extended with the axioms *rdiscr* and *rser* so that the flow of time is discrete and unbounded towards the future (in this case, the presence of *rser* allows us to simplify *rdiscr* to $\forall b.\, \exists d.\, b < d \sqcap \sim \exists u.\, (b < u \sqcap u < d)$). We can express in our syntax the relation *next* in terms of the relation $<$ (see, e.g., [76]), i.e. we can introduce, as in Section 4.2.3, a relational symbol $\lhd$ (with the meaning of *immediately precedes*) as an abbreviation:

$$s \lhd t \equiv s < t \sqcap \forall b.\ \sim (s < b) \sqcup \sim (b < t)\,.$$

This allows us to enrich the language with an operator X, as in the system for *LTL*_ of Section 4.2.4, whose semantics can be given without having to introduce a specific relation for it in the definition of a model. We just need to require that models for this logic are linear temporal structures where $\prec$ is also discrete and serial on the right, and extend the definition of truth with:

$$\mathcal{M}, \lambda \models_{Kl} b : \mathsf{X}A \quad \text{iff} \quad \mathcal{M}, \lambda \models_{Kl} b \lhd c \ \text{ and } \mathcal{M}, \lambda \models_{Kl} c : A\,.$$

Rules for introduction and elimination of X can now be given in a clean way, with the usual freshness proviso for $\mathsf{X}I$:[14]

$$\frac{\begin{array}{c}[b \lhd c] \\ \vdots \\ c : A\end{array}}{b : \mathsf{X}A}\ \mathsf{X}I\ (c \text{ fresh}) \qquad \frac{b : \mathsf{X}A \quad b \lhd c}{c : A}\ \mathsf{X}E\,.$$

The logic that we capture in this extended system, which we call $\mathcal{N}(K_{MTL})$, is not *LTL*_ yet. We are able to express the existence of an immediate successor, but we miss a way to say that between any two points (related by $\prec$) there can be only a finite sequence of points related one to each other by the relation *next*. We would need to express the finite interval property, but this is a second-order property, as observed above.

In [103], a subset of *LTL*_ called *Small Temporal Logic*, or *STL* for short, is introduced and given a natural deduction system. The reasons behind the definition of *STL* are the difficulties arising from dealing with the induction principle (relating $\lhd$ and $<$) that is needed in order to represent *LTL*_. While the semantics of *LTL*_ can be given by considering Kripke structures defined over a relation of successor (denoted by N) and by defining $\prec$ as the least transitive closure of N, in the semantics of *STL* the relation $\prec$ is just required to contain N. It follows that a rule for induction is not needed in a system for *STL*.

It is easy to verify that $\mathcal{N}(K_{MTL})$ is complete with respect to the semantics of *STL*. Moreover, it can be proven to correspond to a logic "larger" than *STL*

[14] The fact that every time point has one (and only one) immediate successor follows from right-discreteness, right-seriality, and connectedness, and it allows one to express rules for X both in a universal and in an existential formulation. We give here the universal one.

for which the condition of linearity (or connectedness) on the relation $\prec$ holds: we call this logic *Medium Temporal Logic MTL*.[15]

*LTL_*

In Section 4.2.4, we defined a labeled natural deduction system for $LTL_{\_}$, with an induction rule (borrowed from [103]) like the following (where b' and b'' are fresh)

$$\frac{\begin{array}{cc} & [b < b'][b' \lhd b''][b' : A] \\ & \vdots \\ b : A \quad b < c & b'' : A \end{array}}{c : A}\ ind$$

which does not operate at a purely relational level. Some remarks are worth about a solution like this. First of all, the rule *ind* adds some more points of contact between the labeled and the relational sub-systems and leads to a failure of normalization. Moreover, one can show that the axiom of connectedness is not needed anymore since it is in a way "contained" in the induction principle. In fact, the axiom (3)

$$\neg \mathsf{G}(\mathsf{G}A \supset B) \supset \mathsf{G}(\mathsf{G}B \supset A)$$

of *weak connectedness* must obviously hold in *LTL* (and thus $LTL_{\_}$), for it can be subsumed by the induction axiom (see, e.g., [75]). Thus, in the case we want to use a rule like *ind* to capture $LTL_{\_}$, it seems more reasonable to follow a different approach that avoids both the extension of the relational language to a first-order language and the introduction of the universal falsum. In other words, we can have a system for $LTL_{\_}$ which uses only Horn rules in the relational theory (from which it follows that we have only atomic rwffs and no relational falsum) but extends the labeled sub-systems with a rule for induction that mixes labeled and relational premises.

4.3.5 Discussion and related works

In this section, we have given labeled natural deduction systems for a family of tense logics and we have proved not only soundness and completeness, but also a number of useful proof-theoretical properties. We have also discussed possible extensions leading up to *LTL*.

An analysis of related works has been already done in Section 4.2.6. Here we just remark that, as discussed in Section 4.3.4, the approach followed in this section and based on the use of a (first-order) relational sub-system allows us to express all the first-order relational properties of structures in a clean and modular way.

[15] An axiomatization of *MTL* can be obtained, as shown in [76], by adding the following axioms to those given for future-time *Kl*:

(K_{X}) $\quad \mathsf{X}(A \supset B) \supset (\mathsf{X}A \supset \mathsf{X}B)$
$(FUNC)$ $\quad (\mathsf{X}\neg A \supset \neg \mathsf{X}A) \wedge (\neg \mathsf{X}A \supset \mathsf{X}\neg A)$
(REC_{G}) $\quad (\mathsf{G}A \supset \mathsf{X}(A \wedge \mathsf{G}A)) \wedge (\mathsf{X}(A \wedge \mathsf{G}A) \supset \mathsf{G}A)$

When we consider the case of (fragments of) LTL, however, we need to express the induction principle, which is a second-order property, and thus it is not possible (at least in our formulation) to do it at a purely relational level. Thus, in such a case, the complexity introduced by using a relational sub-system like the ones of this section seems not to be justified. This is the reason why, in Chapter 5, when moving to consider branching-time logics, we will prefer not to use an explicit relational sub-system. In particular, in Section 5.3, in order to define a natural deduction system for a temporal logic that is based on a branching extension of the linear $LTL_{\_}$, we will use as a base system the one for $LTL_{\_}$ presented in Section 4.2.4 and extend it with rules capturing the branching nature of the logic.

4.4 A proposal for the treatment of *until*

In Sections 4.3 and 4.2, we presented natural deduction systems for a large number of linear temporal logics. However, all of them did not consider the operator until. The reason for such a choice is that until is a notoriously difficult temporal operator to deal with from a proof-theoretical point of view. Thus, at first stage, we have preferred to focus on the definition of well-behaved deduction systems for the until-free fragments of the logics considered. In this section, we propose a solution for the treatment of until in the context of labeled natural deduction. A slightly different version of the material of this section has been presented in [110].

4.4.1 Introduction

The operator until has an "ambivalent" nature, for it can be seen both as an existential and a universal operator at the same time: $A\mathsf{U}B$ holds at the current time instant w iff either B holds at w or there *exists* a time instant w' in the future at which B holds and such that A holds in *all* the time instants between the current one and w'. The words in emphasis highlight the dual existential and universal nature of U, which poses a significant challenge when attempting to give deduction rules for until, so that deduction systems for temporal logics either deliberately exclude until from the set of operators considered or devise clever ways to formalize reasoning about until. And even if one manages to give rules, these often come at the price of additional difficulties for, or even the impossibility of, proving useful metatheoretic properties, such as normalization or the subformula property. (This is even more so in the case of Hilbert-style axiomatizations, which provide axioms for until, but are not easily usable for proof construction.) See, for instance, [9, 21, 58, 81, 83, 146], where techniques for formalizing suitable inference rules include introducing additional information (such as the use of a Skolem function $f(A\mathsf{U}B)$ to name the time instant where B begins to hold), or exploiting the standard recursive unfolding of until (corresponding to the axiom $(A7)$ of Section 2.3.4)

$$A\mathsf{U}B \equiv B \vee (A \wedge \mathsf{X}(A\mathsf{U}B)) \tag{4.1}$$

which says that $A\mathsf{U}B$ iff either B holds or A holds and in the successor time instant (as expressed by the *next* operator X) we have again $A\mathsf{U}B$.

The problem

Let us illustrate more precisely the problem in the context of labeled natural deduction. For concreteness, we can consider the system $\mathcal{N}(LTL_{\_})$, defined in Section 4.2.4 for $LTL_{\_}$. A straightforward way to have a complete system for full LTL is to extend $\mathcal{N}(LTL_{\_})$ with the following three axioms:

1. $b : A\mathsf{U}B \supset \neg\mathsf{G}\neg B$;
2. $b : A\mathsf{U}B \supset (\neg B \supset (A \wedge \mathsf{X}(A\mathsf{U}B)))$;
3. $b : (\neg B \supset (A \wedge \mathsf{X}(A\mathsf{U}B))) \supset A\mathsf{U}B$.

It is however evident that this solution is not proof-theoretically acceptable, as it would make the system of no use in terms of normalization properties.

Unfortunately, finding a proof-theoretically satisfactory solution for the treatment of until is an extremely challenging task. To illustrate this, let us consider a simplified version U^{X} of until with the following semantics:

$$\begin{array}{l}\mathcal{M},\lambda \models_{LTL} b : A\mathsf{U}^{\mathsf{X}}B \;\; \textit{iff} \;\; \text{there are } b_1, b_2 \textit{ such that} \\ \qquad\qquad \mathcal{M},\lambda \models_{LTL} b \leqslant b_1 \textit{ and} \\ \qquad\qquad \mathcal{M},\lambda \models_{LTL} b_1 \lhd b_2 \textit{ and} \\ \qquad\qquad \mathcal{M},\lambda \models_{LTL} b_2 : B \textit{ and} \\ \qquad\qquad \textit{for all } b', \textit{ if } \mathcal{M},\lambda \models_{LTL} b \leqslant b' \textit{ and } \mathcal{M},\lambda \models_{LTL} b' \leqslant b_1 \\ \qquad\qquad \textit{then } \mathcal{M},\lambda \models_{LTL} b' : A\end{array}$$

The standard until formula $A\mathsf{U}B$ is then simply equivalent to the formula $\neg B \supset A\mathsf{U}^{\mathsf{X}}B$.

In the spirit of labeled natural deduction, we could use this semantics to define the following "good" rules for the introduction and elimination of U^{X}:

$$\frac{\begin{array}{cccc} & & & \begin{array}{c}[b \leqslant b'] \\ \vdots \\ b' : A\end{array} \\ b \leqslant b_1 & b_1 \lhd b_2 & b_2 : B & \end{array}}{b : A\mathsf{U}^{\mathsf{X}}B}\ \mathsf{U}^{\mathsf{X}}I$$

$$\frac{\begin{array}{ccccc} b : A\mathsf{U}^{\mathsf{X}}B & & [b \leqslant b_1]\ [b_1 \lhd b_2]\ [b_2 : B]\ \Big[\bigwedge b'\,((b \leqslant b' \,\&\, b' \leqslant b_1) \Longrightarrow b' : A)\Big] & & \\ & & \vdots & & \\ & & b'' : C & & \end{array}}{b'' : C}\ \mathsf{U}^{\mathsf{X}}E$$

where b' is fresh in $\mathsf{U}^{\mathsf{X}}I$ and b_1, b_2 are fresh in $\mathsf{U}^{\mathsf{X}}E$, and where we employ the symbols $\Longrightarrow$, & and $\bigwedge$ to denote the usual semantical operators for implication, conjunction and universal quantification, respectively.

The rule $\mathsf{U}^{\mathsf{X}}I$ is fully standard with respect to our labeled framework, whereas $\mathsf{U}^{\mathsf{X}}E$ falls outside of it. In fact, in order to eliminate the until, we have formalized the semantical condition

$$\textit{for all } b', \textit{ if } \mathcal{M},\lambda \models_{LTL} b \leqslant b' \textit{ and } \mathcal{M},\lambda \models_{LTL} b' \leqslant b_1 \textit{ then } \mathcal{M},\lambda \models_{LTL} b' : A\,,$$

using the conditional assumption

$$\bigwedge b' \left((b \leqslant b' \,\&\, b' \leqslant b_1) \Longrightarrow b' : A \right).$$

Unfortunately, this conditional assumption is not expressible, neither directly nor indirectly, in our labeled framework. More generally, we cannot express conditional hypotheses where the conditions are a conjunction of relational formulas (namely, assumptions of the kind $b_1 \leqslant b_2 \,\&\, \ldots \,\&\, b_{k-1} \leqslant b_k \Longrightarrow b : A$), nor can we express the universal quantification in the hypothesis. We leave for future work the investigation of extensions of our approach in order to deal with such new kinds of hypotheses.

Our proposal

In the solution proposed in this section, we try to make explicit the duality of until by introducing a new temporal operator ∇ that allows us to formalize the "history" of until, i.e., the fact that when we have $A \mathsf{U} B$ the formula A holds in all the time instants between the current one and the one where B holds. We express this "historic" universal quantification by means of a new temporal operator ∇ with respect to the following intuitive translation:

$$A \mathsf{U} B \equiv B \vee \mathsf{F}(\mathsf{X} B \wedge \nabla A) \tag{4.2}$$

That is: $A \mathsf{U} B$ iff either B holds or there exists a time instant w' in the future (as expressed by the *sometime in the future* operator F) such that

- B holds in the successor time instant, and
- A holds in all the time instants between the current one and w' (included).

The latter conjunct is precisely what the *history* operator ∇ expresses[16]. This is better seen when introducing labeling: since ∇ actually quantifies over the time instants in an interval (delimited by the current instant and the one where the B of the until holds), we adopt a labeling discipline that is slightly different from the more customary one of labeled deduction seen in Sections 4.3 and 4.2.

In fact, considering labels that consist of a single time instant is not enough for ∇, as the operator is explicitly designed to speak about an interval. We thus consider labels that are possibly built out of a pair of time instants, so that we can write $b_1 b_3 : \nabla A$ to express, intuitively, that A holds in the interval between the time instants b_1 and b_3. This allows us to give the natural deduction elimination rule

$$\frac{b_1 b_3 : \nabla A \quad b_1 \leqslant b_2 \quad b_2 \leqslant b_3}{b_2 : A} \; \nabla E$$

that says that if ∇A holds in the interval delimited by b_1 and b_3 and if b_2 is in-between b_1 and b_3, as expressed by the relational formulas with the accessibility relation $\leqslant$, then we can conclude that A holds at b_2.

Dually, we can introduce ∇A at the pair (b_1, b_3) whenever from the assumptions $b_1 \leqslant b_2$ and $b_2 \leqslant b_3$ for a *fresh* b_2 we can infer $b_2 : A$:

[16] This is in contrast to the unfolding (4.1). The decoupling of U that we achieve with ∇ is precisely what allows us to give well-behaved (in a sense made clearer below) natural deduction rules.

$$\dfrac{\begin{array}{c}[b_1 \leqslant b_2]\,[b_2 \leqslant b_3] \\ \vdots \\ b_2 : A\end{array}}{b_1 b_3 : \nabla A}\ \nabla I$$

The adoption of pairs of time instant for labels has thus allowed us to give rules for ∇ that are well-behaved in the spirit of natural deduction [125]: there is precisely one introduction and one elimination rule for ∇, as well as for the other connectives and temporal operators ($\supset$, G, and X). This paves the way to a proof-theoretical analysis of the resulting natural deduction systems, e.g., to show proof normalization and other useful meta-theoretical analysis.

Moreover, the rules ∇I and ∇E provide a clean-cut way of reasoning about until, according to the translation (4.2), provided that we also give rules for F and X. These operators have a local nature, in the sense that they speak not about intervals (pairs of time instants) but about single time instants. Still, we can easily give natural deduction rules for them by generalizing the more standard "single-time instant" rules of Sections 4.3 and 4.2 using our labeling with (possibly) pairs of time instants. As we will discuss in more detail below, if we collapse the pairs of time instants to consider only the final time instant in the pair, then these rules reduce to the standard ones. For instance, for the *always in the future* operator G (the dual of F) and X, with the corresponding *successor relation* $\lhd$, we can give the elimination rules

$$\dfrac{b_1 : \mathsf{G}A \quad b_1 \leqslant b_2}{b_1 b_2 : A}\ \mathsf{G}E \qquad \text{and} \qquad \dfrac{b_1 : \mathsf{X}A \quad b_1 \lhd b_2}{b_1 b_2 : A}\ \mathsf{X}E$$

The corresponding introduction rules are given in Section 4.4.4, together with a revised version of the usual rules for $\bot$ and the connective $\supset$, as well as a rule for induction on the underlying linear ordering and rules expressing the properties of the relations $\leqslant$ and $\lhd$. Moreover, the fact that we consider labels that are not necessarily single time instants requires us to consider some structural rules to express properties of such labels (with respect to formulas).

This approach thus provides the basis for formalizing deduction systems for temporal logics endowed with the until operator. For concreteness, we give here a labeled natural deduction system for a linear-time logic endowed with the new history operator ∇ and show that, via a proper translation, such a system is also sound and complete with respect to the linear temporal logic *LTL* with until (Section 2.3.4). (We do not consider past explicitly here, but adding operators and rules for it should be unproblematic, e.g., as in Section 4.3.)

The structure of this section is the following:

- in Section 4.4.2, we define LTL_∇, the logic that is obtained from *LTL* by replacing U with the operator *history* ∇;
- in Section 4.4.3, we provide a translation $(\cdot)^*$ from the language of *LTL* into the language of LTL_∇ and an inverse translation $(\cdot)^\bullet$ from LTL_∇ into *LTL*. Since both the translations can be shown to preserve the validity of formulas, we will finally prove that the two logics are equally expressive;
- in Section 4.4.4, we give a labeled natural deduction system $\mathcal{N}(LTL_\nabla)$ for LTL_∇;

- in Section 4.4.5, we show that $\mathcal{N}(LTL_\nabla)$ is sound with respect to the semantics of LTL_∇ and that, via the translation $(\cdot)^\bullet$, it can be also used to capture reasoning in LTL, with respect to which it is sound too;
- in Section 4.4.6, we prove that $\mathcal{N}(LTL_\nabla)$ is complete, via the translation $(\cdot)^*$, with respect to LTL; by using a double translation $((\cdot)^\bullet)^*$, we also prove a form of completeness with respect to LTL_∇;
- in Section 4.4.7, we summarize and compare with related work.

4.4.2 LTL_∇: LTL with history

Syntax and semantics of LTL, together with a Hilbert-style axiomatization of the logic, have been described in Section 2.3.4. Here we introduce the linear temporal logic LTL_∇, which is obtained from LTL by replacing the operator U with a new unary temporal operator ∇, called *history*. The definition of the semantics of LTL_∇ requires a notion of truth given with respect to points that are possibly pairs of time instants rather than just time instants.

Syntax and semantics

Definition 4.41. *Given a set $\mathcal{P}$ of propositional symbols, the set of* (well-formed) LTL_∇-formulas *is defined by the grammar*

$$A ::= p \mid \bot \mid A \supset A \mid \mathsf{G}A \mid \mathsf{X}A \mid \nabla A$$

where $p \in \mathcal{P}$. The set of LTL_∇-atomic formulas *is $\mathcal{P} \cup \{\bot\}$. The* complexity *of an LTL_∇-formula is the number of occurrences of the connective $\supset$ and of the temporal operators X, G, and ∇.*

The intuitive meaning of the operators G and X is the same as for LTL, while ∇A intuitively states that A holds at any instant of a particular time interval (but here we see that we need more than just time instants to formalize the semantics of the history operator, as we anticipated in Section 4.4.1). Again, we can define other connectives and operators as abbreviations, e.g., $\neg$, $\vee$, $\wedge$, F and so on.

As usual, in order to define a labeled deduction system for the logic LTL_∇, we extend the language with a set of labels and introduce the new notions of labeled formula and relational formula.

Definition 4.42. *Let L be a denumerable set of labels. We say that a* prefix *is a single label b or a pair of labels bc, where $b, c \in L$. If A is an LTL_∇-formula and α is a prefix, then $\alpha : A$ is a* labeled (well-formed) LTL_∇-formula *(*lwff *for short). The set of* relational (well-formed) LTL_∇-formulas *(*rwffs *for short) is the set of expressions of the form $b \leqslant c$ or $b \lhd c$, where $b, c \in L$.*

In the rest of this section, we will assume given a fixed denumerable set L of labels and we will use $b, c, d, \ldots$ to denote labels and $\alpha, \beta, \gamma, \ldots$ to denote prefixes. We will sometimes use parentheses and write, e.g., $(b)c$ to denote a prefix where b is not necessarily present. Furthermore, we will write Λ to denote a *set of LTL-formulas* and Γ to denote a *set of LTL_∇-formulas*. For simplicity, we will often

omit the term LTL_∇ when referring to labeled or relational formulas. So a *labeled formula*, in the context of this section, is always a labeled LTL_∇-formula and a *relational formula* is a relational LTL_∇-formula. φ will denote a *generic formula* (either labeled or relational) and Γ a *set of generic formulas*.

Truth for an LTL_∇-formula is defined by using the same models of *LTL* (see Section 2.3.4), i.e. structures that are isomorphic to the set of natural numbers, but with respect to points that are not necessarily single natural numbers. As anticipated in 4.4.1, we will sometimes need to store elements of the model in order to give a proper interpretation of a formula.

Definition 4.43. *A* time instant *is a natural number n. A* time instant with a store *is a pair of natural numbers (m, n). An* observation point *is a time instant or a time instant with a store.*

We will denote observation points by using square brackets and a comma to separate the possible two values; so we will write, e.g., $[n]$ to indicate a time instant and $[m, n]$ to indicate a time instant with a store. The intuitive interpretation of a time instant with a store $[m, n]$ is that the last element (n) represents the instant where the formula has to be actually evaluated, while the first element (m) represents an instant that we need to store (in order to give an interpretation to formulas with ∇). We will use parentheses, like in $[(m,)n]$, to denote an observation point that may possibly contain a store.

Definition 4.44. Truth *for an LTL_∇-formula at an observation point σ in an LTL-model $\mathcal{M} = (\mathcal{N}, \mathcal{V})$ is the smallest relation $\models_\nabla$ satisfying :*

$$
\begin{array}{rll}
\mathcal{M}, [(m,)n] \models_\nabla p & \textit{iff} & p \in \mathcal{V}(n) \\
\mathcal{M}, \sigma \models_\nabla A \supset B & \textit{iff} & \mathcal{M}, \sigma \models_\nabla A \textit{ implies } \mathcal{M}, \sigma \models_\nabla B \\
\mathcal{M}, [(m,)n] \models_\nabla \mathsf{G}A & \textit{iff} & \mathcal{M}, [n, i] \models_\nabla A \textit{ for all } i \geq n \\
\mathcal{M}, [(m,)n] \models_\nabla \mathsf{X}A & \textit{iff} & \mathcal{M}, [n, n+1] \models_\nabla A \\
\mathcal{M}, [m, n] \models_\nabla \nabla A & \textit{iff} & \mathcal{M}, [i] \models_\nabla A \textit{ for all } m \leq i \leq n \\
\mathcal{M}, [n] \models_\nabla \nabla A & \textit{iff} & \mathcal{M}, [n] \models_\nabla A
\end{array}
$$

By extension, we write:

$$
\begin{array}{rll}
\mathcal{M} \models_\nabla A & \textit{iff} & \mathcal{M}, [n] \models_\nabla A \textit{ for every } n \in \mathbb{N} \\
\mathcal{M} \models_\nabla \Gamma & \textit{iff} & \mathcal{M} \models_\nabla A \textit{ for all } A \in \Gamma \\
\Gamma \models_\nabla A & \textit{iff} & \mathcal{M} \models_\nabla \Gamma \textit{ implies } \mathcal{M} \models_\nabla A, \textit{ for every LTL-model } \mathcal{M}
\end{array}
$$

Notice that the notion of validity in a model ($\mathcal{M} \models_\nabla A$) is given by considering only those observation points consisting of a single instant. This emphasizes the fact that the use of observation points consisting of a time instant plus a store can be seen as just an auxiliary technical device, i.e. in order to evaluate a formula at a given single time instant, we possibly need to consider the evaluation of some of its subformulas at observation points that are endowed with a store. The following example shows that the notion of validity given with respect to single time instants and the notion of validity given with respect to all the observation points are different.

Example 4.45. Let us define a new notion of validity $\models^{\triangledown}$ as follows:

$$\mathcal{M} \models^{\triangledown} A \quad \text{iff} \quad \mathcal{M}, \sigma \models_{\triangledown} A \text{ for every observation point } \sigma$$

$\mathcal{M} \models^{\triangledown} \Gamma$ and $\Gamma \models^{\triangledown} A$ can be defined consequently. Now let $A = p \vee \neg p$, $A_1 = \nabla(p \vee \neg p)$ and $A_2 = \nabla p \vee \nabla(\neg p)$. Then A_1 is semantically equivalent to A (and thus valid) according to both the notion of validities, while A_2 is semantically equivalent to A (and thus valid) only according to the notion of validity $\models_{\triangledown}$. In fact, we have $\models^{\triangledown} A_2$ iff $\models^{\triangledown} p$ or $\models^{\triangledown} \neg p$ and thus A_2 is not valid according to $\models^{\triangledown}$.

Now we introduce the notion of interpretation of labels and prefixes and define, in terms of it, the notion of truth for labeled and relational formulas.

Definition 4.46. *Given an LTL-model $\mathcal{M}$ and a set L of labels, an* interpretation *$\lambda : L \to \mathbb{N}$ is a function mapping each label to a natural number. Let $Pref$ be the set of prefixes defined on L and Σ the set of observation points on $\mathcal{M}$. We define the extension of λ, denoted $\lambda^+ : Pref \to \Sigma$, as follows:*

$\lambda^+(n) = [\lambda(n)]$;
$\lambda^+(n_1\, n_2) = [\lambda(n_1), \lambda(n_2)]$.

Given an LTL-model $\mathcal{M}$, a set L of labels and an interpretation λ on them, truth *for a generic formula φ in a pair $(\mathcal{M}, \lambda)$ is the smallest relation $\models_{\triangledown}$ satisfying:*

$$\begin{array}{lll} \mathcal{M}, \lambda \models_{\triangledown} b \leqslant c & \textit{iff} & \lambda(b) \leq \lambda(c) \\ \mathcal{M}, \lambda \models_{\triangledown} b \lhd c & \textit{iff} & \lambda(c) = \lambda(b) + 1 \\ \mathcal{M}, \lambda \models_{\triangledown} \alpha : A & \textit{iff} & \mathcal{M}, \lambda^+(\alpha) \models_{\triangledown} A \end{array}$$

Note that $\mathcal{M}, \sigma \not\models_{\triangledown} \bot$ and $\mathcal{M}, \lambda \not\models_{\triangledown} \alpha : \bot$ for every $\mathcal{M}$, σ and λ.

Given a set Γ of generic formulas and a generic formula φ:

$$\begin{array}{rll} \mathcal{M}, \lambda \models_{\triangledown} \Gamma & \textit{iff} & \mathcal{M}, \lambda \models_{\triangledown} \varphi \textit{ for all } \varphi \in \Gamma \\ \Gamma \models_{\triangledown} \varphi & \textit{iff} & \mathcal{M}, \lambda \models_{\triangledown} \Gamma \textit{ implies } \mathcal{M}, \lambda \models_{\triangledown} \varphi \textit{ for all } \mathcal{M} \textit{ and } \lambda \end{array}$$

4.4.3 The equivalence of *LTL* and LTL_{∇}

We introduced a variant of *LTL* based on replacing the operator U with the operator ∇, whose interpretation has been described in Section 4.4.1. Here we study the relation between *LTL* and LTL_{∇} and prove that the two logics are indeed equally expressive. Such a proof is given by defining a translation from *LTL* into LTL_{∇} and an inverse one from LTL_{∇} into *LTL*. Both the translations are proved to preserve the validity of formulas.

A translation from *LTL* into LTL_{∇}

We proceed as follows: first, we define a translation $(\cdot)^*$ from *LTL* into LTL_{∇}. Then, in Lemma 4.48, we will show that if an LTL_{∇}-formula corresponds to the translation of some *LTL*-formula, then it can be interpreted "locally", i.e., its truth value with respect to an observation point depends only on the last element and not on the store. Finally, in Lemma 4.50 and Theorem 4.51, we will use this result to prove that the translation preserves the validity of formulas.

Definition 4.47. *We define the translation* $(\cdot)^*$ *from the language of LTL into the language of* LTL_∇ *inductively as follows:*

$$\begin{array}{ll}(p)^* & = p\,,\ \textit{for } p \textit{ atomic}\\ (\mathsf{G}A)^* & = \mathsf{G}\,(A)^*\\ (\bot)^* & = \bot\\ (\mathsf{X}A)^* & = \mathsf{X}\,(A)^*\\ (A \supset B)^* & = (A)^* \supset (B)^*\\ (A\mathsf{U}B)^* & = (B)^* \vee (\,\mathsf{F}\,(\,\mathsf{X}\,(B)^* \wedge \nabla(A)^*\,))\end{array}$$

We extend $(\cdot)^*$ *to sets of formulas in the obvious way:* $\Lambda^* = \{(A)^* \mid A \in \Lambda\}$.

In the following, when not confusing, we will sometimes omit parentheses and write, e.g., A^* instead of $(A)^*$.

Lemma 4.48. *Let* $\mathcal{M}$ *be an LTL-model,* $[(m,)n]$ *an observation point and* A *an LTL-formula. Then*

$$\mathcal{M}, [(m,)n] \models_\nabla A^* \qquad \Leftrightarrow \qquad \mathcal{M}, [(i,)n] \models_\nabla A^* \textit{ for every natural number } i\,.$$

Proof. By induction on the complexity of A. The base case is when $A = p$ or $A = \bot$ and is trivial. There is one inductive step case for each connective and temporal operator.

$A = B \supset C$. Then the translation of A is $A^* = B^* \supset C^*$. By Definition 4.44, we obtain $\mathcal{M}, [(m,)n] \models_\nabla B^* \supset C^*$ iff $\mathcal{M}, [(m,)n] \models_\nabla B^*$ implies $\mathcal{M}, [(m,)n] \models_\nabla C^*$. By the induction hypothesis, we see that this holds iff $\mathcal{M}, [(i,)n] \models_\nabla B^*$ implies $\mathcal{M}, [(i,)n] \models_\nabla C^*$ for every natural number i and thus, by Definition 4.44, iff for every natural number i, $\mathcal{M}, [(i,)n] \models_\nabla B^* \supset C^*$.

$A = \mathsf{G}B$. Then $A^* = \mathsf{G}B^*$. In this case, we do not even use the induction hypothesis. Just observe that, by Definition 4.44, the possible value of m is not involved in the evaluation of the formula. Thus we have $\mathcal{M}, [(m,)n] \models_\nabla \mathsf{G}B^*$ iff $\forall l \geq n.\, \mathcal{M}, [n,l] \models_\nabla B^*$ iff $\mathcal{M}, [(i,)n] \models_\nabla \mathsf{G}B^*$, for every natural number i.

$A = \mathsf{X}B$. This case is very similar to the previous one and we omit it.

$A = B\mathsf{U}C$. Then $A^* = C^* \vee (\mathsf{F}(\mathsf{X}C^* \wedge \nabla B^*))$. By Definition 4.44, we have $\mathcal{M}, [(m,)n] \models_\nabla A^*$ iff $(\mathcal{M}, [(m,)n] \models_\nabla C^*$ or $\mathcal{M}, [(m,)n] \models_\nabla \mathsf{F}(\mathsf{X}C^* \wedge \nabla B^*))$ iff $(\mathcal{M}, [(m,)n] \models_\nabla C^*$ or $\exists l \geq n.\, (\mathcal{M}, [n,l] \models_\nabla \mathsf{X}C^* \wedge \nabla B^*))$ iff $(\mathcal{M}, [(m,)n] \models_\nabla C^*$ or $\exists l \geq n.\, (\mathcal{M}, [n,l] \models_\nabla \mathsf{X}C^*$ and $\mathcal{M}, [n,l] \models_\nabla \nabla B^*))$ iff $(\mathcal{M}, [(m,)n] \models_\nabla C^*$ or $\exists l \geq n.\, (\mathcal{M}, [l,l+1] \models_\nabla C^*$ and $\forall l'.\, n \leq l' \leq l$ implies $\mathcal{M}, [l'] \models_\nabla B^*))$ iff (by the induction hypothesis) for every natural number i, we have $(\mathcal{M}, [(i,)n] \models_\nabla C^*$ or $\exists l \geq n.\, (\mathcal{M}, [l,l+1] \models_\nabla C^*$ and $\forall l'.\, n \leq l' \leq l$ implies $\mathcal{M}, [l'] \models_\nabla B^*))$ iff (by Definition 4.44) $\mathcal{M}, [(i,)n] \models_\nabla C^* \vee (\mathsf{F}(\mathsf{X}C^* \wedge \nabla B^*))$ for every natural number i.

□

Corollary 4.49. *Let* $\mathcal{M}$ *be an LTL-model,* $[(m,)n]$ *an observation point, and* A *an LTL-formula. Then* $\mathcal{M}, [(m,)n] \models_\nabla A^*$ *iff* $\mathcal{M}, [n] \models_\nabla A^*$.

Proof. Immediate, by Lemma 4.48.

□

Lemma 4.50. *Let $\mathcal{M}$ be an LTL-model, n a natural number and A an LTL-formula. Then*

$$\mathcal{M}, n \models_{LTL} A \qquad \Leftrightarrow \qquad \mathcal{M}, [n] \models_{\nabla} A^* .$$

Proof. By induction on the complexity of A. The base case is when $A = p$ or $A = \bot$ and is trivial. As inductive step, we have a case for each connective and temporal operator.

$A = B \supset C$. Then $A^* = B^* \supset C^*$. We have $\mathcal{M}, n \models_{LTL} B \supset C$ iff (by Definition 2.11) $\mathcal{M}, n \models_{LTL} B$ implies $\mathcal{M}, n \models_{LTL} C$ iff (by the induction hypothesis) $\mathcal{M}, [n] \models_{\nabla} B^*$ implies $\mathcal{M}, [n] \models_{\nabla} C^*$ iff (by Definition 4.44) $\mathcal{M}, [n] \models_{\nabla} B^* \supset C^*$.

$A = \mathsf{G}B$. Then $A^* = \mathsf{G}B^*$. We have $\mathcal{M}, n \models_{LTL} \mathsf{G}B$ iff (by Definition 2.11) $\forall m \geq n. \mathcal{M}, m \models_{LTL} B$ iff (by the induction hypothesis) $\forall m \geq n. \mathcal{M}, [m] \models_{\nabla} B^*$ iff (by Lemma 4.48) $\forall m \geq n. \mathcal{M}, [n, m] \models_{\nabla} B^*$ iff (by Definition 4.44) $\mathcal{M}, [n] \models_{\nabla} \mathsf{G}B^*$.

$A = \mathsf{X}B$. This case is very similar to the previous one and we omit it.

$A = B\mathsf{U}C$. Then $A^* = C^* \vee (\mathsf{F}(\mathsf{X}C^* \wedge \nabla B^*))$. We have $\mathcal{M}, n \models_{LTL} A$ iff (by Definition 2.11) $\exists m \geq n. \mathcal{M}, m \models_{LTL} C$ and $\forall n'. n \leq n' < m$ implies $\mathcal{M}, n' \models_{LTL} B$ iff $\mathcal{M}, n \models_{LTL} C$ or $(\exists m > n. \mathcal{M}, m \models_{LTL} C$ and $\forall n'. n \leq n' < m$ implies $\mathcal{M}, n' \models_{LTL} B)$ iff (by the induction hypothesis) $\mathcal{M}, [n] \models_{\nabla} C^*$ or $(\exists m > n. \mathcal{M}, [m] \models_{\nabla} C^*$ and $\forall n'. n \leq n' < m$ implies $\mathcal{M}, [n'] \models_{\nabla} B^*)$ iff (by simple rewriting) $\mathcal{M}, [n] \models_{\nabla} C^*$ or $(\exists l \geq n. \mathcal{M}, [l+1] \models_{\nabla} C^*$ and $\forall n'. n \leq n' \leq l$ implies $\mathcal{M}, [n'] \models_{\nabla} B^*)$ iff (by Lemma 4.48) $\mathcal{M}, [n] \models_{\nabla} C^*$ or $(\exists l \geq n. \mathcal{M}, [l, l+1] \models_{\nabla} C^*$ and $\forall n'. n \leq n' \leq l$ implies $\mathcal{M}, [n'] \models_{\nabla} B^*)$ iff (by Definition 4.44) $\mathcal{M}, [n] \models_{\nabla} C^*$ or $(\exists l \geq n. \mathcal{M}, [n, l] \models_{\nabla} \mathsf{X}C^* \wedge \nabla B^*)$ iff (by Definition 4.44) $\mathcal{M}, [n] \models_{\nabla} C^* \vee \mathsf{F}(\mathsf{X}C^* \wedge \nabla B^*)$.

□

Theorem 4.51. *Let Λ be a set of LTL-formulas and A an LTL-formula. Then*

$$\Lambda \models_{LTL} A \qquad \Leftrightarrow \qquad \Lambda^* \models_{\nabla} A^* .$$

Proof. By Definition 2.11, $\Lambda \models_{LTL} A$ iff $\forall \mathcal{M}. \mathcal{M} \models_{LTL} \Lambda$ implies $\mathcal{M} \models_{LTL} A$ iff $\forall \mathcal{M}. (\forall B \in \Lambda. \forall n. \mathcal{M}, n \models_{LTL} B$ implies $\forall n. \mathcal{M}, n \models_{LTL} A)$ iff (by Lemma 4.50) $\forall \mathcal{M}. (\forall B \in \Lambda. \forall n. \mathcal{M}, [n] \models_{\nabla} B^*$ implies $\forall n. \mathcal{M}, [n] \models_{\nabla} A^*)$ iff (by Lemma 4.48) $\forall \mathcal{M}. (\forall B \in \Lambda. \forall \sigma. \mathcal{M}, \sigma \models_{\nabla} B^*$ implies $\forall \sigma. \mathcal{M}, \sigma \models_{\nabla} A^*)$ iff (by Definition 4.44) $\forall \mathcal{M}. (\forall B \in \Lambda. \mathcal{M} \models_{\nabla} B^*$ implies $\mathcal{M} \models_{\nabla} A^*)$ iff $\forall \mathcal{M}. (\mathcal{M} \models_{\nabla} \Lambda^*$ implies $\mathcal{M} \models_{\nabla} A^*)$ iff $\Lambda^* \models_{\nabla} A^*$.

□

A translation from LTL_{∇} into LTL

Defining a translation from LTL_{∇} into LTL is a much trickier task. Typically, translations are defined recursively: we have a case for each possible main connective of a formula and in all of these cases the translation is given in terms of the translation of its subformulas. A similar recursive definition, when translating LTL_{∇} into LTL, needs to take into account some subtleties.

Clearly, the interesting case in the translation is that of formulas containing the operator ∇. Furthermore, by observing the semantics of LTL_{∇} (Section 4.4.2), one can conclude (we will prove it formally below) that:

- when ∇ is in the scope of another ∇, it can be ignored, e.g., $\nabla\nabla A \equiv \nabla A$;
- when ∇ is not in the scope of any temporal operator, it does not alter the evaluation of the formula, e.g., $\nabla A \equiv A$.

Thus the crucial case is when ∇ is in the scope of a different temporal operator: X or G (or F, if we consider it explicitly).[17]

We have seen that, in order to define the semantics of LTL_{∇}, we need to consider pairs of instants, such that one instant (the second one) is where the evaluation actually takes place and the other (the first one) is a kind of pointer to some other instant in the flow of time. By reading Definition 4.44, we deduce that this pointer is in fact only needed to evaluate a restricted class of LTL_{∇}-formulas.

Namely, we can divide LTL_{∇}-formulas into two classes:

1. the class of *history-independent* formulas, whose evaluation only depends on the last element of an observation point;
2. the class of *history-dependent* formulas, whose evaluation depends also on the first element (the pointer, or the store) of an observation point.

By observing the semantics of LTL_{∇}, one can easily check that the history-dependent formulas are indeed those where the ∇ operator is not in the scope of any different temporal operator. As an example, we have that the formula $\mathsf{G}\nabla p$ is history-independent, but its subformula ∇p is history-dependent.

All these arguments lead to the intuition that the translation of a formula of the form $\mathsf{X}A$ or $\mathsf{G}A$ should depend on the nature of the subformula A. If the formula A is history-independent, then we can give for it a simple recursive definition, otherwise we need to consider a translation that mimics in some way the action of the pointer. In this second case, considering a (disjunctive) normal form for LTL_{∇}-formulas will help define the translation.

In the following paragraphs, we formalize all these ideas and prove that the resulting translation preserves the validity of formulas.

An alternative grammar for LTL_{∇}-formulas

Here we give an alternative grammar for LTL_{∇}-formulas with the intent of making the separation between history-independent and history-dependent formulas clear. Since it allows for a simpler presentation of the translation, we give the grammar by considering $\neg$, $\wedge$, $\vee$, X and F as primitive connectives. $\bot$, $\supset$ and G can be defined in terms of these in the standard way.

Definition 4.52. *Given a set $\mathcal{P}$ of propositional symbols, the set of* (well-formed) LTL^{∇}-formulas *is defined by the grammar*

$$A ::= \gamma \mid \delta$$

[17] Indeed, even the case of a ∇ in the scope of an X could be simplified by splitting it into two elementary subcases, e.g., $\mathsf{X}\nabla A \equiv A \wedge \mathsf{X}A$. Thus, in conclusion, the case of a ∇ in the scope of a G (or of an F) is the one that really matters.

$$\gamma ::= p \mid \gamma \wedge \gamma \mid \gamma \vee \gamma \mid \neg\gamma \mid \mathsf{X}\gamma \mid \mathsf{F}\gamma \mid \mathsf{X}\delta \mid \mathsf{F}\delta$$
$$\delta ::= \nabla A \mid \neg\delta \mid A \wedge \delta \mid \delta \wedge A \mid A \vee \delta \mid \delta \vee A$$

where $p \in \mathcal{P}$. We call (LTL^{∇}) history-independent formulas the formulas belonging to the syntactic category γ and (LTL^{∇}) history-dependent formulas the formulas belonging to the syntactic category δ.

Lemma 4.53. *The language of LTL^{∇}-formulas and the language of LTL_{∇}-formulas coincide.*

Proof. We have to show that: (*i*) each LTL^{∇}-formula is also an LTL_{∇}-formula; and, viceversa, (*ii*) each LTL_{∇}-formula is also an LTL^{∇}-formula. The proof proceeds by structural induction in both directions; we omit the details. □

Because of Lemma 4.53, from now on, for simplicity, we will speak of LTL_{∇}-formulas also when referring to formulas originating from the grammar in Definition 4.52.

A normal form for LTL_{∇}-formulas

Considering a normal form for LTL_{∇}-formulas will help define the translation. The first step will consist in eliminating some redundant occurrences of ∇: intuitively, those occurrences falling directly into the scope of another ∇. Some proper terminology needs to be introduced.

Definition 4.54. *Let A be an LTL_{∇}-formula of the form $\mathsf{X}A'$ (or $\mathsf{G}A'$, or $\nabla A'$) and let us denote with h that occurrence of X (or of G, or of ∇, respectively). Then for each occurrence h' of a temporal operator in A', we say that h' is in the* temporal scope *of h.*

Given an LTL_{∇}-formula A, we say that an occurrence h of a temporal operator in A is in the strict temporal scope *of an occurrence h' of a temporal operator in A iff:*

1. *h is in the temporal scope of h'; and*
2. *for each occurrence h'' of a temporal operator in A:*
   a) *h is not in in the temporal scope of h''; or*
   b) *h' is in the temporal scope of h''.*

We also say that an occurrence of a ∇ in an LTL_{∇}-formula A is redundant *if it is in the strict temporal scope of another occurrence of ∇.*

Example 4.55. Consider the formula $\mathsf{XG}(\nabla p \wedge q)$. The occurrence of ∇ (not redundant) is in the temporal scope of the occurrences of both X and G, and in the strict temporal scope of the occurrence of G.

In $\mathsf{X}\nabla(\nabla p \wedge q)$, the second occurrence of ∇ is in the strict temporal scope of the first one and thus it is redundant.

Lemma 4.56. *Let A be an LTL_{∇}-formula and B be the formula obtained by removing all the redundant occurrences of the operator ∇. Then A and B are semantically equivalent.*

Proof. By observing the semantics given in Definition 4.44, we can first notice that the evaluation of a formula of the form ∇A at an observation point that is a single time instant (without a store) corresponds to the evaluation of the formula A at the same point. Now observe that if an occurrence of ∇ is in the strict temporal scope of another occurrence of ∇, then its evaluation is performed in a single time instant-observation point. This implies that the removal of the inner-most ∇ does not alter the evaluation.

□

In order to get a normal form, we require, in addition to the removal of redundant occurrences of ∇, that each history-dependent subformula is written in a particular form. The following definition, lemma and example clarify and formalize the form of normal LTL_{∇}-formulas.

Definition 4.57. *Given an LTL_{∇}-formula A, we say that δ is a* history-dependent subformula *of A iff δ is a subformula of A and is a history-dependent formula.*

Definition 4.58. *A* δ-disjunctive normal form clause *(δ-DNF clause, for short) is an LTL_{∇}-formula consisting of a conjunction of formulas that are:*

1. *history-independent formulas; or*
2. *history-dependent formulas of the form $\nabla\gamma$ or $\neg\nabla\gamma$ for some history-independent formula γ.*

An LTL_{∇}-formula A is in δ-disjunctive normal form *(in δ-DNF, for short) if:*

1. *A does not contain any redundant occurrence of a ∇; and*
2. *for each history-dependent subformula δ of A, δ is the disjunction of δ-DNF clauses.*

Lemma 4.59. *For every LTL_{∇}-formula A, there exists an equivalent LTL_{∇}-formula A' such that A' is in δ-DNF.*

Proof. We prove the statement by describing a procedure for transforming a generic LTL_{∇}-formula A into an LTL_{∇}-formula A' that is in δ-DNF.

First, we remove all the occurrences of the operator ∇ that are in the strict temporal scope of another occurrence of ∇. Lemma 4.56 ensures that after this process we have an equivalent formula.

Then we observe that, once we have removed the redundant occurrences of ∇, given a subformula δ of A, the process of reducing δ to a disjunction of conjunctions (as required by Definition 4.58) is equivalent to the process of reducing a formula of propositional classical logic into the standard disjunctive normal form (see, e.g., [155]), where we consider as literals:

1. history-independent formulas; or
2. history-dependent formulas of the form $\nabla\gamma$ or $\neg\nabla\gamma$ for some history-independent formula γ.

Thus, in order to transform an LTL_{∇}-formula without redundant occurrences of ∇ into a formula in δ-DNF, we can iteratively apply the following procedure, corresponding (mutatis mutandis) to the one defined for producing a disjunctive normal form, to each history dependent subformula of A, starting from the inner-most one.

1. we iteratively apply the so-called double negation and De Morgan's laws (see [155]) in order to get a formula where we have only single negations and they occur just before the atoms (where we consider history-independent formulas or history-dependent formulas of the form $\nabla\gamma$ as atoms);
2. we iteratively apply distributivity laws in order to get a disjunction of conjunctions.

The proof that the resulting formula is equivalent to the original one is a trivial adaptation (again, mutatis mutandis) of the proof [155] given for transformations into the standard disjunctive normal form in the case of propositional classical logic.

□

Example 4.60. Let us consider the LTL_{∇}-formula

$$A \equiv p_1 \wedge \neg\mathsf{F}(\mathsf{X}\nabla p_2 \wedge \neg(p_3 \vee \nabla\mathsf{F}\nabla(p_4 \vee p_5))) .$$

First, we eliminate the redundant occurrences of ∇ and obtain

$$A_1 \equiv p_1 \wedge \neg\mathsf{F}(\mathsf{X}\nabla p_2 \wedge \neg(p_3 \vee \nabla\mathsf{F}(p_4 \vee p_5))) .$$

Then we consider the history-dependent subformulas of A'. The inner-most ones are ∇p_2 and $\nabla\mathsf{F}(p_4 \vee p_5)$, which are already in normal form. Then we consider $\neg(p_3 \wedge \nabla\mathsf{F}(p_4 \vee p_5))$, to which we can apply De Morgan laws and obtain

$$A_2 \equiv p_1 \wedge \neg\mathsf{F}(\mathsf{X}\nabla p_2 \wedge (\neg p_3 \vee \neg\nabla\mathsf{F}(p_4 \vee p_5))) .$$

Finally, by applying distributivity laws to $\mathsf{X}\nabla p_2 \wedge (\neg p_3 \vee \neg\nabla\mathsf{F}(p_4 \vee p_5))$, we get

$$A_3 \equiv p_1 \wedge \neg\mathsf{F}((\mathsf{X}\nabla p_2 \wedge \neg p_3) \vee (\mathsf{X}\nabla p_2 \wedge \neg\nabla\mathsf{F}(p_4 \vee p_5))) ,$$

which is in δ-DNF.

The translation $(\cdot)^{\bullet}$

Since Lemma 4.59 holds, we can, with no loss of generality, restrict the attention to LTL_{∇}-formulas that are in δ-DNF and define the translation $(\cdot)^{\bullet}$ from LTL_{∇} into LTL in terms of this class of formulas. We also remark, as it will be useful in defining the translation and in proving some statements, that given an LTL_{∇}-formula A in δ-DNF, every its subformula of the form ∇B is such that B is history-independent. Such a fact is a direct consequence of the absence of redundant occurrences of ∇ in a formula in δ-DNF form.

Definition 4.61. *We define the translation* $(\cdot)^{\bullet}$ *from the language of* LTL_{∇}*-formulas in* δ*-DNF form into the language of LTL inductively as follows. (Note that, as in Definition 4.52, we use* A*,* γ *and* δ *(possibly subscripted) to denote a generic* LTL_{∇}*-formula, a history-independent formula and a history-dependent formula, respectively.)*

$$
\begin{array}{ll}
(p)^\bullet & = p\,, \textit{ for } p \textit{ atomic} \\
(A_1 \wedge A_2)^\bullet & = (A_1)^\bullet \wedge (A_2)^\bullet \\
(A_1 \vee A_2)^\bullet & = (A_1)^\bullet \vee (A_2)^\bullet \\
(\neg A)^\bullet & = \neg\,(A)^\bullet \\
(\mathsf{X}\gamma)^\bullet & = \mathsf{X}\,(\gamma)^\bullet \\
(\mathsf{F}\gamma)^\bullet & = \mathsf{F}\,(\gamma)^\bullet \\
(\nabla A)^\bullet & = (A)^\bullet \\
(\mathsf{X}\delta)^\bullet & = (C_1)^{\mathsf{X}} \vee \ldots \vee (C_n)^{\mathsf{X}} \\
(\mathsf{F}\delta)^\bullet & = (C_1)^{\mathsf{F}} \vee \ldots \vee (C_n)^{\mathsf{F}}
\end{array}
$$

where $\delta \equiv C_1 \vee \ldots \vee C_n$ for $C_1, \ldots, C_n$ δ-DNF clauses and $(\cdot)^{\mathsf{X}}$ and $(\cdot)^{\mathsf{F}}$ are auxiliary translations defined from the set of δ-DNF clauses into the set of LTL-formulas as specified below.

Let C be a δ-DNF clause. Since the order of the elements of a conjunction does not alter its evaluation, we can always write it as:

$$C \;\equiv\; (\gamma_1 \wedge \ldots \wedge \gamma_n) \;\wedge\; (\nabla\gamma'_1 \wedge \ldots \wedge \nabla\gamma'_m) \;\wedge\; (\neg\nabla\gamma''_1 \wedge \ldots \wedge \neg\nabla\gamma''_l)\,.$$

Furthermore, let $\gamma \equiv \gamma_1 \wedge \ldots \wedge \gamma_n$ and $\gamma_\nabla \equiv \gamma'_1 \wedge \ldots \wedge \gamma'_m$. For greater convenience, we also define another version of the operator until *on LTL-formulas:*

$$A \underline{\mathsf{U}} B \;\equiv\; (A \wedge B) \wedge ((A \wedge \mathsf{X}A)\,\mathsf{U}\,B)\,,$$

where the idea is that now A holds also in the instant where B holds.

Then we define $(\cdot)^{\mathsf{X}}$ and $(\cdot)^{\mathsf{F}}$ as follows:

$$(C)^{\mathsf{X}} = \mathsf{X}(\gamma)^\bullet \;\wedge\; (\gamma_\nabla)^\bullet \wedge \mathsf{X}(\gamma_\nabla)^\bullet \;\wedge\; (\neg(\gamma''_1)^\bullet \vee \neg\mathsf{X}(\gamma''_1)^\bullet) \wedge \ldots \wedge (\neg(\gamma''_l)^\bullet \vee \neg\mathsf{X}(\gamma''_l)^\bullet)$$

$$(C)^{\mathsf{F}} = \mathsf{F}(\gamma)^\bullet \wedge ((\gamma_\nabla)^\bullet \,\underline{\mathsf{U}}\, (\gamma)^\bullet) \wedge \neg((\gamma''_1)^\bullet \,\underline{\mathsf{U}}\, (\gamma)^\bullet) \wedge \ldots \wedge \neg((\gamma''_l)^\bullet \,\underline{\mathsf{U}}\, (\gamma)^\bullet)$$

We extend $(\cdot)^\bullet$ to sets of formulas in the obvious way: $\Gamma^\bullet = \{(A)^\bullet \mid A \in \Gamma\}$.

In the following, when not confusing, we will sometimes omit parentheses and write, e.g., $A^\bullet$, C^{X} and C^{F} instead of $(A)^\bullet$, $(C)^{\mathsf{X}}$ and $(C)^{\mathsf{F}}$, respectively.

Properties of the translation

Here we show that the translation $(\cdot)^\bullet$ preserves the validity of formulas. Along the proofs of the following lemmas, $\gamma, \gamma_1, \gamma_2, \ldots$ will denote history-independent formulas, $\delta, \delta_1, \delta_2, \ldots$ history-dependent formulas and $A, A_1, A_2, \ldots$ generic LTL_∇-formulas.

Lemma 4.62. *Let $\mathcal{M}$ be an LTL-model, $m, n \in \mathbb{N}$ and γ a history-independent formula. Then*

$$\mathcal{M}, [(m,)n] \models_\nabla \gamma \qquad \Leftrightarrow \qquad \mathcal{M}, [(m',)n] \models_\nabla \gamma\,, \textit{ for all } m' \in \mathbb{N}\,.$$

Proof. The proof is by induction on the complexity of the formula γ. The base case is when $\gamma = p$ and is trivial. There is one inductive step case for each other formation case coming from the recursive definition of the grammar in Definition 4.52. Along the proof, γ, γ_1 and γ_2 denote history-independent formulas while A denotes a generic LTL_∇-formula.

$\gamma = \gamma_1 \wedge \gamma_2$. By Definition 4.44, we have $\mathcal{M}, [(m,)n] \models_\nabla \gamma_1 \wedge \gamma_2$ iff $\mathcal{M}, [(m,)n] \models_\nabla \gamma_1$ and $\mathcal{M}, [(m,)n] \models_\nabla \gamma_2$. By the induction hypothesis, this holds iff $\mathcal{M}, [(m',)n] \models_\nabla \gamma_1$ and $\mathcal{M}, [(m',)n] \models_\nabla \gamma_2$ for every natural number m', and thus, by Definition 4.44, iff for every natural number m', $\mathcal{M}, [(m',)n] \models_\nabla \gamma_1 \wedge \gamma_2$.

$\gamma = \gamma_1 \vee \gamma_2$. By Definition 4.44, we have $\mathcal{M}, [(m,)n] \models_\nabla \gamma_1 \vee \gamma_2$ iff $\mathcal{M}, [(m,)n] \models_\nabla \gamma_1$ or $\mathcal{M}, [(m,)n] \models_\nabla \gamma_2$. By the induction hypothesis, this holds iff $\mathcal{M}, [(m',)n] \models_\nabla \gamma_1$ or $\mathcal{M}, [(m',)n] \models_\nabla \gamma_2$ for every natural number m', and thus, by Definition 4.44, iff for every natural number m', $\mathcal{M}, [(m',)n] \models_\nabla \gamma_1 \vee \gamma_2$.

$\gamma = \neg\gamma_1$. By Definition 4.44, we have $\mathcal{M}, [(m,)n] \models_\nabla \neg\gamma_1$ iff $\mathcal{M}, [(m,)n] \not\models_\nabla \gamma_1$. By the induction hypothesis, this holds iff $\mathcal{M}, [(m',)n] \not\models_\nabla \gamma_1$ for every natural number m', and thus, by Definition 4.44, iff for every natural number m', $\mathcal{M}, [(m',)n] \models_\nabla \neg\gamma_1$.

$\gamma = \mathsf{X}A$. We treat at the same time the cases where A is a history-independent and A is a history-dependent formula, and we do not need to use the induction hypothesis. By Definition 4.44, we have $\mathcal{M}, [(m,)n] \models_\nabla \mathsf{X}A$ iff $\mathcal{M}, [n, n+1] \models_\nabla A$. Again, by Definition 4.44, this holds iff for every natural number m', $\mathcal{M}, [(m',)n] \models_\nabla \mathsf{X}A$.

$\gamma = \mathsf{F}A$. Again, we do not use the induction hypothesis. By Definition 4.44, we have $\mathcal{M}, [(m,)n] \models_\nabla \mathsf{F}A$ iff there exists $i \geq n$ such that $\mathcal{M}, [n,i] \models_\nabla A$. By Definition 4.44, this holds iff for every natural number m', $\mathcal{M}, [(m',)n] \models_\nabla \mathsf{F}A$.

□

Lemma 4.63. *Let $\mathcal{M}$ be an LTL-model, $n \in \mathbb{N}$ and A an LTL$_\nabla$-formula. Then*

$$\mathcal{M}, [n] \models_\nabla A \qquad \Leftrightarrow \qquad \mathcal{M}, n \models_{LTL} A^\bullet .$$

Proof. The proof is by structural induction on A.

$A = p$. By Definition 4.61, $A^\bullet = p$. We have $\mathcal{M}, [n] \models_\nabla p$ iff (by Definition 4.44) $p \in \mathcal{V}(n)$ iff (by Definition 2.11) $\mathcal{M}, n \models_{LTL} p$.

$A = A_1 \wedge A_2$. By Definition 4.61, $A^\bullet = A_1^\bullet \wedge A_2^\bullet$. We have $\mathcal{M}, [n] \models_\nabla A_1 \wedge A_2$ iff (by Definition 4.44) $\mathcal{M}, [n] \models_\nabla A_1$ and $\mathcal{M}, [n] \models_\nabla A_2$ iff (by the induction hypothesis) $\mathcal{M}, n \models_{LTL} A_1^\bullet$ and $\mathcal{M}, n \models_{LTL} A_2^\bullet$ iff (by Definition 2.11) $\mathcal{M}, n \models_{LTL} A_1^\bullet \wedge A_2^\bullet$.

$A = A_1 \vee A_2$. By Definition 4.61, $A^\bullet = A_1^\bullet \vee A_2^\bullet$. We have $\mathcal{M}, [n] \models_\nabla A_1 \vee A_2$ iff (by Definition 4.44) $\mathcal{M}, [n] \models_\nabla A_1$ or $\mathcal{M}, [n] \models_\nabla A_2$ iff (by the induction hypothesis) $\mathcal{M}, n \models_{LTL} A_1^\bullet$ or $\mathcal{M}, n \models_{LTL} A_2^\bullet$ iff (by Definition 2.11) $\mathcal{M}, n \models_{LTL} A_1^\bullet \vee A_2^\bullet$.

$A = \neg A_1$. By Definition 4.61, $A^\bullet = \neg(A_1^\bullet)$. We have $\mathcal{M}, [n] \models_\nabla \neg A_1$ iff (by Definition 4.44) $\mathcal{M}, [n] \not\models_\nabla A_1$ iff (by the induction hypothesis) $\mathcal{M}, n \not\models_{LTL} A_1^\bullet$ iff (by Definition 2.11) $\mathcal{M}, n \models_{LTL} \neg(A_1^\bullet)$.

$A = \mathsf{X}\gamma$. By Definition 4.61, $A^\bullet = \mathsf{X}(\gamma^\bullet)$. We have $\mathcal{M}, [n] \models_\nabla \mathsf{X}\gamma$ iff (by Definition 4.44) $\mathcal{M}, [n, n+1] \models_\nabla \gamma$ iff (by Lemma 4.63) $\mathcal{M}, [n+1] \models_\nabla \gamma$ iff (by the induction hypothesis) $\mathcal{M}, n+1 \models_{LTL} \gamma^\bullet$ iff (by Definition 2.11) $\mathcal{M}, n \models_{LTL} \mathsf{X}(\gamma^\bullet)$.

$A = \mathsf{F}\gamma$. By Definition 4.61, $A^\bullet = \mathsf{F}(\gamma^\bullet)$. We have $\mathcal{M}, [n] \models_\nabla \mathsf{F}\gamma$ iff (by Definition 4.44) there exists $i \geq n$ such that $\mathcal{M}, [n, i] \models_\nabla \gamma$ iff (by Lemma 4.63) there exists $i \geq n$ such that $\mathcal{M}, [i] \models_\nabla \gamma$ iff (by the induction hypothesis) there exists $i \geq n$ such that $\mathcal{M}, i \models_{LTL} \gamma^\bullet$ iff (by Definition 2.11) $\mathcal{M}, n \models_{LTL} \mathsf{F}(\gamma^\bullet)$.

$A = \mathsf{X}\delta$. By Definition 4.61, $A^\bullet = (C_1)^\mathsf{X} \vee \ldots \vee (C_m)^\mathsf{X}$, where $\delta \equiv C_1 \vee \ldots \vee C_m$ and $C_1, \ldots, C_m$ are δ-DNF clauses. For $1 \leq i \leq m$, we can write $C_i = \gamma_1 \wedge \ldots \wedge \gamma_{k_i} \wedge (\nabla\gamma'_1 \wedge \ldots \wedge \nabla\gamma'_{j_i}) \wedge (\neg\nabla\gamma''_1 \wedge \ldots \wedge \neg\nabla\gamma''_{l_i})$. For convenience, we also define $\gamma_{\wedge_i} = \gamma_1 \wedge \ldots \wedge \gamma_{k_i}$ and $\gamma_{\nabla_i} = \gamma'_1 \wedge \ldots \wedge \gamma'_{j_i}$.

First, we prove that, for $1 \leq i \leq m$, $\mathcal{M}, [n, n+1] \models_\nabla C_i$ iff $\mathcal{M}, n \models_{LTL} (C_i)^\mathsf{X}$. We have: $\mathcal{M}, [n, n+1] \models_\nabla C_i$

- iff (by Definition 4.44) $\mathcal{M}, [n, n+1] \models_\nabla \gamma_h$ for all h s.t. $1 \leq h \leq k_i$ and $\mathcal{M}, [n, n+1] \models_\nabla \nabla\gamma'_h$ for all h s.t. $1 \leq h \leq j_i$ and $\mathcal{M}, [n, n+1] \models_\nabla \neg\nabla\gamma''_h$ for all h s.t. $1 \leq h \leq l_i$
- iff (by Lemma 4.62) $\mathcal{M}, [n+1] \models_\nabla \gamma_h$ for all h s.t. $1 \leq h \leq k_i$ and $\mathcal{M}, [n, n+1] \models_\nabla \nabla\gamma'_h$ for all h s.t. $1 \leq h \leq j_i$ and $\mathcal{M}, [n, n+1] \models_\nabla \neg\nabla\gamma''_h$ for all h s.t. $1 \leq h \leq l_i$
- iff (by Definition 4.44) $\mathcal{M}, [n+1] \models_\nabla \gamma_h$ for all h s.t. $1 \leq h \leq k_i$ and $(\mathcal{M}, [n] \models_\nabla \gamma'_h$ and $\mathcal{M}, [n+1] \models_\nabla \gamma'_h)$ for all h s.t. $1 \leq h \leq j_i$ and $(\mathcal{M}, [n] \not\models_\nabla \gamma''_h$ or $\mathcal{M}, [n+1] \not\models_\nabla \gamma''_h)$ for all h s.t. $1 \leq h \leq l_i$
- iff (by the induction hypothesis) $\mathcal{M}, n+1 \models_{LTL} \gamma_h^\bullet$ for all h s.t. $1 \leq h \leq k_i$ and $(\mathcal{M}, n \models_{LTL} \gamma_h'^\bullet$ and $\mathcal{M}, n+1 \models_{LTL} \gamma_h'^\bullet)$ for all h s.t. $1 \leq h \leq j_i$ and $(\mathcal{M}, n \not\models_{LTL} \gamma_h''^\bullet$ or $\mathcal{M}, n+1 \not\models_{LTL} \gamma_h''^\bullet)$ for all h s.t. $1 \leq h \leq l_i$
- iff (by Definition 2.11) $\mathcal{M}, n+1 \models_{LTL} \gamma_1^\bullet \wedge \ldots \wedge \gamma_{k_i}^\bullet$ and $(\mathcal{M}, n \models_{LTL} \gamma_1'^\bullet \wedge \ldots \wedge \gamma_{j_i}'^\bullet$ and $\mathcal{M}, n+1 \models_{LTL} \gamma_1'^\bullet \wedge \ldots \wedge \gamma_{j_i}'^\bullet)$ and $(\mathcal{M}, n \not\models_{LTL} \gamma_h''^\bullet$ or $\mathcal{M}, n \not\models_{LTL} \mathsf{X}(\gamma_h''^\bullet))$ for all h s.t. $1 \leq h \leq l_i$
- iff (by Definition 4.61) $\mathcal{M}, n+1 \models_{LTL} (\gamma_{\wedge_i})^\bullet$ and $(\mathcal{M}, n \models_{LTL} (\gamma_{\nabla_i})^\bullet$ and $\mathcal{M}, n+1 \models_{LTL} (\gamma_{\nabla_i})^\bullet)$ and $(\mathcal{M}, n \not\models_{LTL} \gamma_h''^\bullet$ or $\mathcal{M}, n \not\models_{LTL} \mathsf{X}(\gamma_h''^\bullet))$ for all h s.t. $1 \leq h \leq l_i$
- iff (by Definition 2.11) $\mathcal{M}, n \models_{LTL} \mathsf{X}(\gamma_{\wedge_i})^\bullet$ and $\mathcal{M}, n \models_{LTL} (\gamma_{\nabla_i})^\bullet$ and $\mathcal{M}, n \models_{LTL} \mathsf{X}((\gamma_{\nabla_i})^\bullet)$ and $(\mathcal{M}, n \models_{LTL} \neg(\gamma_h''^\bullet)$ or $\mathcal{M}, n \models_{LTL} \neg\mathsf{X}(\gamma_h''^\bullet))$ for all h s.t. $1 \leq h \leq l_i$
- iff $\mathcal{M}, n \models_{LTL} (C_i)^\mathsf{X}$.

Now we use this result to prove the main statement. Namely we have: $\mathcal{M}, [n] \models_\nabla \mathsf{X}\delta$

- iff (by Definition 4.44) $\mathcal{M}, [n, n+1] \models_\nabla \delta$
- iff (by Definition 4.44) $\mathcal{M}, [n, n+1] \models_\nabla C_1$ or ... or $\mathcal{M}, [n, n+1] \models_\nabla C_m$
- iff (by the result above) $\mathcal{M}, n \models_{LTL} (C_1)^\mathsf{X}$ or ... or $\mathcal{M}, n \models_{LTL} (C_m)^\mathsf{X}$
- iff (by Definition 2.11) $\mathcal{M}, n \models_{LTL} (C_1)^\mathsf{X} \vee \ldots \vee (C_m)^\mathsf{X}$
- iff $\mathcal{M}, n \models_{LTL} A^\bullet$.

$A = \mathsf{F}\delta$. By Definition 4.61, $A^\bullet = (C_1)^\mathsf{F} \vee \ldots \vee (C_m)^\mathsf{F}$, where $\delta \equiv C_1 \vee \ldots \vee C_m$ and $C_1, \ldots, C_m$ are δ-DNF clauses. For $1 \leq i \leq m$, we can write $C_i = \gamma_1 \wedge \ldots \wedge \gamma_{k_i} \wedge (\nabla\gamma'_1 \wedge \ldots \wedge \nabla\gamma'_{j_i}) \wedge (\neg\nabla\gamma''_1 \wedge \ldots \wedge \neg\nabla\gamma''_{l_i})$. For convenience, we also define, as above, $\gamma_{\wedge_i} = \gamma_1 \wedge \ldots \wedge \gamma_{k_i}$ and $\gamma_{\nabla_i} = \gamma'_1 \wedge \ldots \wedge \gamma'_{j_i}$.

First, we prove that, for $1 \leq i \leq m$, there exists $n' \geq n$ such that $\mathcal{M}, [n, n'] \models_\nabla C_i$ iff $\mathcal{M}, n \models_{LTL} (C_i)^\mathsf{F}$. In fact, we have: there exists $n' \geq n$ such that $\mathcal{M}, [n, n'] \models_\nabla C_i$

iff (by Definition 4.44) there exists $n' \geq n$ such that $\mathcal{M}, [n, n'] \models_{\nabla} \gamma_h$ for all h s.t. $1 \leq h \leq k_i$ and $\mathcal{M}, [n, n'] \models_{\nabla} \nabla\gamma'_h$ for all h s.t. $1 \leq h \leq j_i$ and $\mathcal{M}, [n, n'] \models_{\nabla} \neg\nabla\gamma''_h$ for all h s.t. $1 \leq h \leq l_i$

iff (by Lemma 4.62) there exists $n' \geq n$ such that $\mathcal{M}, [n'] \models_{\nabla} \gamma_h$ for all h s.t. $1 \leq h \leq k_i$ and $\mathcal{M}, [n, n'] \models_{\nabla} \nabla\gamma'_h$ for all h s.t. $1 \leq h \leq j_i$ and $\mathcal{M}, [n, n'] \models_{\nabla} \neg\nabla\gamma''_h$ for all h s.t. $1 \leq h \leq l_i$

iff (by Definition 4.44) there exists $n' \geq n$ such that $\mathcal{M}, [n'] \models_{\nabla} \gamma_h$ for all h s.t. $1 \leq h \leq k_i$ and $(\mathcal{M}, [n''] \models_{\nabla} \gamma'_h$ for all n'' s.t. $n \leq n'' \leq n'$ and for all h s.t. $1 \leq h \leq j_i)$ and (for all h s.t. $1 \leq h \leq l_i$ there exists n'' s.t. $n \leq n'' \leq n'$ for which $\mathcal{M}, [n''] \not\models_{\nabla} \gamma''_h)$

iff (by the induction hypothesis) there exists $n' \geq n$ such that $\mathcal{M}, n' \models_{LTL} \gamma^{\bullet}_h$ for all h s.t. $1 \leq h \leq k_i$ and $(\mathcal{M}, n'' \models_{LTL} \gamma'^{\bullet}_h$ for all n'' s.t. $n \leq n'' \leq n'$ and for all h s.t. $1 \leq h \leq j_i)$ and (for all h s.t. $1 \leq h \leq l_i$ there exists n'' s.t. $n \leq n'' \leq n'$ for which $\mathcal{M}, n'' \not\models_{LTL} \gamma''^{\bullet}_h)$

iff (by Definition 2.11) there exists $n' \geq n$ such that $(\mathcal{M}, n' \models_{LTL} \gamma^{\bullet}_1 \wedge \ldots \wedge \gamma^{\bullet}_{k_i})$ and $(\mathcal{M}, n'' \models_{LTL} \gamma'^{\bullet}_1 \wedge \ldots \wedge \gamma'^{\bullet}_{j_i}$ for all n'' s.t. $n \leq n'' \leq n')$ and (for all h s.t. $1 \leq h \leq l_i$ there exists n'' s.t. $n \leq n'' \leq n'$ for which $\mathcal{M}, n'' \not\models_{LTL} \gamma''^{\bullet}_h)$

iff (by Definition 4.61) there exists $n' \geq n$ such that $(\mathcal{M}, n' \models_{LTL} (\gamma_{\wedge_i})^{\bullet})$ and $(\mathcal{M}, n'' \models_{LTL} (\gamma_{\nabla_i})^{\bullet}$ for all n'' s.t. $n \leq n'' \leq n')$ and (for all h s.t. $1 \leq h \leq l_i$ there exists n'' s.t. $n \leq n'' \leq n'$ for which $\mathcal{M}, n'' \not\models_{LTL} \gamma''^{\bullet}_h)$

iff (by Definition 2.11) $\mathcal{M}, n \models_{LTL} \mathsf{F}(\gamma_{\wedge_i})^{\bullet}$ and $\mathcal{M}, n \models_{LTL} ((\gamma_{\nabla_i})^{\bullet} \underline{\mathsf{U}} (\gamma_{\wedge_i})^{\bullet})$ and for all h s.t. $1 \leq h \leq l_i$, $\mathcal{M}, n \models_{LTL} \neg((\gamma''_h)^{\bullet} \underline{\mathsf{U}} (\gamma_{\wedge_i})^{\bullet})$

iff $\mathcal{M}, n \models_{LTL} (C_i)^{\mathsf{F}}$.

Now we use this result to prove the main statement. Namely we have: $\mathcal{M}, [n] \models_{\nabla} \mathsf{F}\delta$

iff (by Definition 4.44) there exists $n' \geq n$ such that $\mathcal{M}, [n, n'] \models_{\nabla} \delta$

iff (by Definition 4.44) there exists $n' \geq n$ such that $\mathcal{M}, [n, n'] \models_{\nabla} C_1$ or $\ldots$ or $\mathcal{M}, [n, n'] \models_{\nabla} C_m$

iff (by the result above) $\mathcal{M}, n \models_{LTL} (C_1)^{\mathsf{F}}$ or $\ldots$ or $\mathcal{M}, n \models_{LTL} (C_m)^{\mathsf{F}}$

iff (by Definition 2.11) $\mathcal{M}, n \models_{LTL} (C_1)^{\mathsf{F}} \vee \ldots \vee (C_m)^{\mathsf{F}}$

iff $\mathcal{M}, n \models_{LTL} A^{\bullet}$.

□

Proposition 4.64. *Let $\mathcal{M}$ be an LTL-model and γ a history-independent formula. Then*

$$\mathcal{M} \models_{\nabla} \gamma \qquad \Leftrightarrow \qquad \mathcal{M} \models_{LTL} \gamma^{\bullet}.$$

Proof. By Definition 4.44, $\mathcal{M} \models_{\nabla} \gamma$ iff $\mathcal{M}, [n] \models_{\nabla} \gamma$ for all $n \in \mathbb{N}$ iff (by Lemma 4.63) $\mathcal{M}, n \models_{LTL} \gamma^{\bullet}$ for all $n \in \mathbb{N}$ iff (by Definition 2.11) $\mathcal{M} \models_{LTL} \gamma^{\bullet}$.

□

Proposition 4.65. *Let $\mathcal{M}$ be an LTL-model and δ a history-dependent formula. Then*

$$\mathcal{M} \models_{\nabla} \delta \qquad \Leftrightarrow \qquad \mathcal{M} \models_{LTL} \delta^{\bullet}.$$

Proof. By Definition 4.44, $\mathcal{M} \models_{\nabla} \delta$ iff $\mathcal{M}, [n] \models_{\nabla} \delta$ for all $n \in \mathbb{N}$ iff (by Lemma 4.63) $\mathcal{M}, n \models_{LTL} \delta^{\bullet}$ for all $n \in \mathbb{N}$ iff (by Definition 2.11) $\mathcal{M} \models_{LTL} \delta^{\bullet}$.

□

Theorem 4.66. *Let Γ be a set of LTL_∇-formulas and A an LTL_∇-formula. Then*

$$\Gamma \models_\nabla A \qquad \Leftrightarrow \qquad \Gamma^\bullet \models_{LTL} A^\bullet .$$

Proof. We have $\Gamma \models_\nabla A$ iff (by Definition 4.44) for every *LTL*-model $\mathcal{M}$, ($\mathcal{M} \models_\nabla \Gamma$ implies $\mathcal{M} \models_\nabla A$) iff (by Definition 4.44) for every *LTL*-model $\mathcal{M}$, (($\mathcal{M} \models_\nabla B$ for every LTL_∇-formula $B \in \Gamma$) implies $\mathcal{M} \models_\nabla A$) iff (by Propositions 4.64 and 4.65) for every *LTL*-model $\mathcal{M}$, (($\mathcal{M} \models_{LTL} B^\bullet$ for every LTL_∇-formula $B \in \Gamma$) implies $\mathcal{M} \models_{LTL} A^\bullet$) iff (by Definition 2.11) $\Gamma^\bullet \models_{LTL} A^\bullet$.

□

4.4.4 $\mathcal{N}(LTL_\nabla)$: a labeled natural deduction system for LTL_∇

In this section, we will first define a labeled natural deduction system $\mathcal{N}(LTL_\nabla)$ on the language of LTL_∇-formulas. By considering the translations $(\cdot)^*$ and $(\cdot)^\bullet$, in the next sections we will show how it is possible to use such a system also for reasoning on *LTL*.

The rules of $\mathcal{N}(LTL_\nabla)$

The rules of $\mathcal{N}(LTL_\nabla)$ are given in Figure 4.19. The core is the system $\mathcal{N}(LTL_\_)$; thus, there are no rules whose conclusion is an rwff.

The rules $\supset I$, $\supset E$ and $\bot E$ are just an adaptation of those of $\mathcal{N}(LTL_\_)$ to the case of prefixes that are not necessarily single labels.

The rules for the introduction and the elimination of G and X share the same structure. Consider, for instance, G and the corresponding relation $\leqslant$. The idea underlying the introduction rule $\mathsf{G}I$ is that the meaning of $b_1 : \mathsf{G}A$ is given by the metalevel implication $b_1 \leqslant b_2 \Longrightarrow b_1 b_2 : A$ for an arbitrary b_2 $\leqslant$-accessible from b_1 (where the arbitrariness of b_2 is ensured by the side-condition on the rule). As we remarked above, the operators G and X have a local nature, in that when we write $(b_1)b_2 : \mathsf{G}A$ we are stating that $\mathsf{G}A$ holds at time instant b_2, which is the last in the observation point. Hence, the elimination rule $\mathsf{G}E$ says that if b_2 is $\leqslant$-accessible from b_1 (i.e., $b_1 \leqslant b_2$), then we can conclude that A holds for the sequence $b_1 b_2$. Similar observations hold for X and the corresponding relation $\lhd$.

As in the previous sections, the rule *ser*$\lhd$ models the fact that every time instant has an immediate successor, while the rule *lin*$\lhd$ specifies that such a successor must be unique.

Similarly, the rules *refl* $\leqslant$ and *trans* $\leqslant$ state the reflexivity and transitivity of $\leqslant$, while *eq* $\leqslant$ captures substitution of equals.[18] The rule *split* $\leqslant$ states that if $b_1 \leqslant b_2$, then either $b_1 = b_2$ or $b_1 < b_2$. The rule thus works in the style of a disjunction elimination: if by assuming either of the two cases, we can derive a formula $\alpha : A$, then we can discharge the assumptions and conclude $\alpha : A$.

The rule *base* $\leqslant$ expresses the fact that $\leqslant$ contains $\lhd$, while the rule *ind* models the induction principle underlying the relation between $\lhd$ and $\leqslant$.

[18] Recall that in this system we use rwffs only as assumptions for the derivation of lwffs, so we do not need a more general rule that concludes $\varphi[b_2/b_1]$ from φ, $b_1 \leqslant b_2$ and $b_2 \leqslant b_1$.

$$\frac{\begin{array}{c}[\alpha_1 : A \supset \bot] \\ \vdots \\ \alpha_2 : \bot\end{array}}{\alpha_1 : A}\ \bot E \qquad \frac{\begin{array}{c}[\alpha : A] \\ \vdots \\ \alpha : B\end{array}}{\alpha : A \supset B}\ \supset I \qquad \frac{\alpha : A \supset B \quad \alpha : A}{\alpha : B}\ \supset E$$

$$\frac{\begin{array}{c}[b_1 \leqslant b_2] \\ \vdots \\ b_1 b_2 : A\end{array}}{b_1 : \mathsf{G}A}\ \mathsf{G}I \qquad \frac{(b) b_1 : \mathsf{G}A \quad b_1 \leqslant b_2}{b_1 b_2 : A}\ \mathsf{G}E \qquad \frac{b_1 \lhd b_2 \quad b_1 \lhd b_3 \quad \varphi \qquad \begin{array}{c}[\varphi[b_3/b_2]] \\ \vdots \\ \alpha : A\end{array}}{\alpha : A}\ lin\lhd$$

$$\frac{\begin{array}{c}[b_1 \lhd b_2] \\ \vdots \\ b_1 b_2 : A\end{array}}{b_1 : \mathsf{X}A}\ \mathsf{X}I \qquad \frac{(b) b_1 : \mathsf{X}A \quad b_1 \lhd b_2}{b_1 b_2 : A}\ \mathsf{X}E \qquad \frac{b_1 \leqslant b_2 \quad b_2 \leqslant b_3 \qquad \begin{array}{c}[b_1 \leqslant b_3] \\ \vdots \\ \alpha : A\end{array}}{\alpha : A}\ trans\leqslant$$

$$\frac{\begin{array}{c}[b_1 \leqslant b_2]\,[b_2 \leqslant b_3] \\ \vdots \\ b_2 : A\end{array}}{b_1 b_3 : \nabla A}\ \nabla I \qquad \frac{b_1 b_3 : \nabla A \quad b_1 \leqslant b_2 \quad b_2 \leqslant b_3}{b_2 : A}\ \nabla E \qquad \frac{(b_1) b : A}{(b_2) b : A}\ last \qquad \frac{\begin{array}{c}[b_1 \leqslant b_1] \\ \vdots \\ \alpha : A\end{array}}{\alpha : A}\ refl\leqslant$$

$$\frac{b_1 \leqslant b_2 \quad \varphi \qquad \begin{array}{c}[\varphi[b_2/b_1]] \\ \vdots \\ \alpha : A\end{array} \qquad \begin{array}{c}[b_1 \lhd b'][b' \leqslant b_2] \\ \vdots \\ \alpha : A\end{array}}{\alpha : A}\ split\leqslant \qquad \frac{\begin{array}{c}[b_1 \leqslant b_2] \\ \vdots \\ b_1 \lhd b_2 \qquad \alpha : A\end{array}}{\alpha : A}\ base\leqslant \qquad \frac{\begin{array}{c}[b_1 \lhd b_2] \\ \vdots \\ \alpha : A\end{array}}{\alpha : A}\ ser\lhd$$

$$\frac{b_1 \leqslant b_2 \quad b_2 \leqslant b_1 \quad (b') b_1 : A}{(b') b_2 : A}\ eq\leqslant \qquad \frac{(b') b_0 : A \quad b_0 \leqslant b \qquad \begin{array}{c}[b_0 \leqslant b_i]\,[b_i \lhd b_j]\,[(b') b_i : A] \\ \vdots \\ (b') b_j : A\end{array}}{(b') b : A}\ ind$$

The rules have the following side conditions:

- In $\mathsf{X}I$ ($\mathsf{G}I$), b_2 is *fresh*, i.e., it is different from b_1 and does not occur in any assumption on which $\alpha b_1 b_2 : A$ depends other than the discharged assumption $b_1 \lhd b_2$ ($b_1 \leqslant b_2$).
- In ∇I, b_2 is *fresh*, i.e., it is different from b_1 and b_3, and does not occur in any assumption on which $\alpha b_1 b_2 : A$ depends other than the discharged assumptions $b_1 \leqslant b_2$ and $b_2 \leqslant b_3$.
- In *last*, the formula A must be history-independent (see Definition 4.52).
- In $ser\lhd$, b_2 is fresh, i.e., it is different from b and does not occur in any assumption on which $\alpha : A$ depends other than the discharged assumption $b_1 \lhd b_2$.
- In *split* $\leqslant$, b' is fresh, i.e., it is different from b_1 and b_2 and does not occur in any assumption on which $\alpha : A$ depends other than the discharged assumptions $b_1 \lhd b'$ and $b' \leqslant b_2$.
- In *ind*, b_i and b_j are fresh, i.e., they are different from each other and from b and b_0, and do not occur in any assumption on which $\alpha b_0 b_j : A$ depends other than the discharged assumptions of the rule.
- In *ind* and *eq* $\leqslant$, the use of the parentheses has to be intended as follows: b' is either present in all the prefixes where it occurs between parentheses or in none of them.

Fig. 4.19. The rules of $\mathcal{N}(LTL_{\nabla})$.

Finally, we have three rules that speak about the history and the observation points: the rules ∇I and ∇E, which we already described in the introduction, and *last*. This rule expresses what we also anticipated in Sections 4.4.1 and 4.4.3: the standard operators (and connectives) of *LTL* only speak about single time instants, and thus if a formula A is history-independent (see Definition 4.52), then given a lwff $(b_1)b : A$ we can safely replace the possible store b_1 of our observation point by any other time instant b_2 and conclude that A holds at $(b_2)b$.

We write $\Phi \vdash_{\nabla} \alpha : A$ to say that there exists a derivation of $\alpha : A$ in the system $\mathcal{N}(LTL_{\nabla})$ whose open assumptions are all contained in the set of formulas Φ.

4.4.5 Soundness

In this section we discuss the soundness of the system $\mathcal{N}(LTL_{\nabla})$. First, we show that it is sound with respect to the semantics of LTL_{∇}. Then we extend this result to *LTL* and prove that $\mathcal{N}(LTL_{\nabla})$ is also sound, by means of the translation $(\cdot)^{\bullet}$, with respect to the semantics of *LTL*.

Theorem 4.67. *For every set Φ of labeled and relational formulas and every labeled formula $\alpha : A$,*

$$\Phi \vdash_{\nabla} \alpha : A \qquad \Rightarrow \qquad \Phi \models_{\nabla} \alpha : A\,.$$

Proof. The proof proceeds by induction on the structure of the derivation of $\alpha : A$. The base case is when $\alpha : A \in \Phi$ and is trivial. There is one step case for every rule and we show here the most representative cases.

First, consider the case in which the last rule application is a ∇I, where $\alpha = b_1b_3$, $A = \nabla B$, and Π is a proof of $b_2 : B$ from hypotheses in Φ', with b_2 fresh and with $\Phi' = \Phi \cup \{b_1 \leqslant b_2\} \cup \{b_2 \leqslant b_3\}$.

$$\dfrac{\begin{array}{c}[b_1 \leqslant b_2] \quad [b_2 \leqslant b_3] \\ \Pi \\ b_2 : B\end{array}}{b_1b_3 : \nabla B}\ \nabla I$$

By the induction hypothesis, for every interpretation λ, if $\mathcal{M}, \lambda \models_{\nabla} \Phi'$, then $\mathcal{M}, \lambda \models_{\nabla} b_1b_2 : B$. We let λ be any interpretation such that $\mathcal{M}, \lambda \models_{\nabla} \Phi$, and show that $\mathcal{M}, \lambda \models_{\nabla} b_1b_3 : \nabla B$. Let $\lambda(b_1) = n$ and $\lambda(b_3) = m$. Since b_2 is fresh, we can extend λ to an interpretation (still called λ for simplicity) such that $\lambda(b_2) = n + i$ for an arbitrary $0 \leq i \leq m$. The induction hypothesis yields $\mathcal{M}, \lambda \models_{\nabla} b_2 : B$, i.e., $\mathcal{M}, [n + i] \models_{\nabla} B$, and thus, since i is an arbitrary point between 0 and m, we obtain $\mathcal{M}, [n, n + m] \models_{\nabla} \nabla B$. It follows $\mathcal{M}, \lambda \models_{\nabla} \beta b_1b_3 : \nabla B$.

Now consider the case in which the last rule applied is ∇E and $\alpha = b_2$:

$$\dfrac{\begin{array}{c}\Pi \\ b_1b_3 : \nabla A\end{array} \quad b_1 \leqslant b_2 \quad b_2 \leqslant b_3}{b_2 : A}\ \nabla E$$

where Π is a proof of $b_1b_3 : \nabla A$ from hypotheses in Φ_1, with $\Phi = \Phi_1 \cup \{b_1 \leqslant b_2\} \cup \{b_2 \leqslant b_3\}$ for some set Φ_1 of formulas. By applying the induction hypothesis on Π, we have:

$$\Phi_1 \models_\nabla b_1 b_3 : \nabla A \, .$$

We proceed by considering a generic *LTL*-model $\mathcal{M}$ and a generic interpretation λ on it such that $\mathcal{M}, \lambda \models_\nabla \Phi$ and showing that this entails

$$\mathcal{M}, \lambda \models_\nabla b_2 : A \, .$$

Since $\Phi_1 \subset \Phi$, we deduce $\mathcal{M}, \lambda \models_\nabla \Phi_1$ and, from the induction hypothesis, $\mathcal{M}, \lambda \models_\nabla b_1 b_3 : \nabla A$. Furthermore $\mathcal{M}, \lambda \models_\nabla \Phi$ entails $\mathcal{M}, \lambda \models_\nabla b_1 \leqslant b_2$ and $\mathcal{M}, \lambda \models_\nabla b_2 \leqslant b_3$. Then, by Definition 4.44, we obtain $\mathcal{M}, \lambda \models_\nabla b_2 : A$.

Consider the case in which the last rule application is a $\mathsf{G}I$, where $\alpha = b_1$ and $A = \mathsf{G}B$:

$$\dfrac{\begin{array}{c}[b_1 \leqslant b_2] \\ \Pi \\ b_1 b_2 : B\end{array}}{b_1 : \mathsf{G}B} \; \mathsf{G}I$$

where Π is a proof of $b_1 : \mathsf{G}B$ from hypotheses in Φ', with b_2 fresh and with $\Phi' = \Phi \cup \{b_1 \leqslant b_2\}$. By the induction hypothesis, for all interpretations λ, if $\mathcal{M}, \lambda \models_\nabla \Phi'$, then $\mathcal{M}, \lambda \models_\nabla b_1 b_2 : B$. We let λ be any interpretation such that $\mathcal{M}, \lambda \models_\nabla \Phi$, and show that $\mathcal{M}, \lambda \models_\nabla b_1 : \mathsf{G}B$. Let $\lambda(b_1) = n$. Since b_2 is fresh, we can extend λ to an interpretation (still called λ for simplicity) such that $\lambda(b_2) = n + m$ for an arbitrary $m > 0$. The induction hypothesis yields $\mathcal{M}, \lambda \models_\nabla b_1 b_2 : B$, i.e., $\mathcal{M}, [n, n+m] \models_\nabla B$, and thus, since m is arbitrary, we obtain $\mathcal{M}, [n] \models_\nabla \mathsf{G}B$. It follows $\mathcal{M}, \lambda \models_\nabla b_1 : \mathsf{G}B$.

Now consider the case in which the last rule applied is $\mathsf{G}E$ and $\alpha = b_1 b_2$:

$$\dfrac{\begin{array}{c}\Pi \\ (b)b_1 : \mathsf{G}A \quad b_1 \leqslant b_2\end{array}}{b_1 b_2 : A} \; \mathsf{G}E$$

where Π is a proof of $(b)b_1 : \mathsf{G}A$ from hypotheses in Φ_1, with $\Phi = \Phi_1 \cup \{b_1 \leqslant b_2\}$ for some set Φ_1 of formulas. By applying the induction hypothesis on Π, we have:

$$\Phi_1 \models_\nabla (b)b_1 : \mathsf{G}A \, .$$

We proceed by considering a generic *LTL*-model $\mathcal{M}$ and a generic interpretation λ on it such that $\mathcal{M}, \lambda \models_\nabla \Phi$ and showing that this entails

$$\mathcal{M}, \lambda \models_\nabla b_1 b_2 : A \, .$$

Since $\Phi_1 \subset \Phi$, we deduce $\mathcal{M}, \lambda \models_\nabla \Phi_1$ and, from the induction hypothesis, $\mathcal{M}, \lambda \models_\nabla (b)b_1 : \mathsf{G}A$. Furthermore $\mathcal{M}, \lambda \models_\nabla \Phi$ entails $\mathcal{M}, \lambda \models_\nabla b_1 \leqslant b_2$. Then, by Definition 4.44, we obtain $\mathcal{M}, \lambda \models_\nabla b_1 b_2 : A$.

Now consider the case in which the last rule applied is *last* and $\alpha = (b_2)b$, where Π is a proof of $(b_1)b : A$ from hypotheses in Φ. By applying the induction hypothesis on Π, we have $\Phi \models_\nabla (b_1)b : A$.

$$\dfrac{\begin{array}{c}\Pi \\ (b_1)b : A\end{array}}{(b_2)b : A} \; \mathit{last}$$

We proceed by considering a generic *LTL*-model $\mathcal{M}$ and a generic interpretation λ on it such that $\mathcal{M}, \lambda \models_\triangledown \Phi$ and showing that this entails $\mathcal{M}, \lambda \models_\triangledown (b_2)b : A$. By the induction hypothesis, $\mathcal{M}, \lambda \models_\triangledown (b_1)b : A$, i.e., $\mathcal{M}, \lambda^+((b_1)b) \models_\triangledown A$ by Definition 4.44. Since A is a history-independent formula, by the side condition of the rule, and the two observation sequences $\lambda^+((b_1)b)$ and $\lambda^+((b_2)b)$ share the same last element $\lambda(b)$, we can apply Lemma 4.62 and obtain $\mathcal{M}, \lambda^+((b_2)b) \models_\triangledown A$, i.e., $\mathcal{M}, \lambda \models_\triangledown (b_2)b : A$ by Definition 4.44.

Finally, consider the case in which the last rule applied is *ind* and $\alpha = (b')b$:

$$\dfrac{\begin{array}{cc} & [b_0 \leqslant b_i] \quad [b_i \lhd b_j] \quad [(b')b_i : A] \\ \Pi' & \Pi \\ (b')b_0 : A \quad b_0 \leqslant b & (b')b_j : A \end{array}}{(b')b : A}\ ind$$

where Π is a proof of $(b')b_j : A$ from hypotheses in Φ_2 and Π' is a proof of $(b')b_0 : A$ from hypotheses in Φ_1, with $\Phi = \Phi_1 \cup \{b_0 \leqslant b\}$ and $\Phi_2 = \Phi_1 \cup \{b_0 \leqslant b_i\} \cup \{b_i \lhd b_j\} \cup \{(b')b_i : A\}$ for some set Φ_1 of formulas. The side-condition on *ind* ensures that b_i and b_j are fresh in Π. Hence, by applying the induction hypothesis on Π and Π', we have:

$$\Phi_2 \models_\triangledown (b')b_j : A \qquad \text{and} \qquad \Phi_1 \models_\triangledown (b')b_0 : A\,.$$

We proceed by considering a generic *LTL*-model $\mathcal{M}$ and a generic interpretation λ on it such that $\mathcal{M}, \lambda \models_\triangledown \Phi$ and showing that this entails

$$\mathcal{M}, \lambda \models_\triangledown (b')b : A\,.$$

First, we note that $\Phi_1 \subset \Phi$ and therefore $\mathcal{M}, \lambda \models_\triangledown \Phi$ implies $\mathcal{M}, \lambda \models_\triangledown \Phi_1$ and, by the induction hypothesis on Π', $\mathcal{M}, \lambda \models_\triangledown (b')b_0 : A$. Now let $\lambda(b_0) = n$ for some natural number n. From $\mathcal{M}, \lambda \models_\triangledown \Phi$, we deduce $\mathcal{M}, \lambda \models_\triangledown b_0 \leqslant b$ and thus $\lambda(b) = n + k$ for some $k \in \mathbb{N}$. We show by induction on k that $\mathcal{M}, \lambda \models_\triangledown (b')b : A$. As a base case, we have $k = 0$; it follows that $\lambda(b) = \lambda(b_0)$ and thus trivially that $\mathcal{M}, \lambda \models_\triangledown (b')b_0 : A$ entails $\mathcal{M}, \lambda \models_\triangledown (b')b : A$. Let us consider now the induction step. Given a label b_{k-1} such that $\lambda(b_{k-1}) = n+k-1$, we show that the induction hypothesis $\mathcal{M}, \lambda \models_\triangledown (b')b_{k-1} : A$ entails the thesis $\mathcal{M}, \lambda \models_\triangledown (b')b : A$. We can build an interpretation λ' that differs from λ only in the points assigned to b_i and b_j, namely, $\lambda' = \lambda[b_i \mapsto n + k - 1][b_j \mapsto n + k]$. It is easy to verify that the interpretation λ' is such that the following three conditions hold:

(*i*) $\mathcal{M}, \lambda' \models_\triangledown (b')b_i : A$;
(*ii*) $\mathcal{M}, \lambda' \models_\triangledown b_0 \leqslant b_i$;
(*iii*) $\mathcal{M}, \lambda' \models_\triangledown b_i \lhd b_j$.

Furthermore, the side-condition on the rule *ind* ensures that λ and λ' agree on all the labels occurring in Φ_1, from which we can infer $\mathcal{M}, \lambda' \models_\triangledown \Phi_1$. It follows $\mathcal{M}, \lambda' \models_\triangledown \Phi_2$ and thus, by the induction hypothesis on Π, $\mathcal{M}, \lambda' \models_\triangledown (b')b_j : A$. We conclude $\mathcal{M}, \lambda' \models_\triangledown (b')b : A$ by observing that $\lambda'(b_j) = \lambda(b)$.

□

We have proved the soundness of the system in terms of the labeled language. It is trivial to infer from it a result of soundness in terms of the logic LTL_{∇}, by focusing on those derivations where both the conclusion and all the open assumptions are lwffs prefixed by the same single label.

Corollary 4.68. *Let $\Gamma = \{A_1, \ldots, A_n\}$ be a set of LTL_{∇}-formulas, A an LTL_{∇}-formula and b a label. Then*

$$b : A_1, \ldots, b : A_n \vdash_{\nabla} b : A \qquad \Rightarrow \qquad \Gamma \models_{\nabla} A\,.$$

Proof. By Theorem 4.67, $b : A_1, \ldots, b : A_n \vdash_{\nabla} b : A$ implies $b : A_1, \ldots, b : A_n \models_{\nabla} b : A$. By Definition 4.46, $b : A_1, \ldots, b : A_n \models_{\nabla} b : A$ implies $\Gamma \models_{\nabla} A$. □

Now, by exploiting the translation $(\cdot)^{\bullet}$ defined in Section 4.4.3, we extend this result to a form of soundness with respect to *LTL*.

Theorem 4.69. *Let $\Gamma = \{A_1, \ldots, A_n\}$ be a set of LTL_{∇}-formulas, A an LTL_{∇}-formula and b a label. Then*

$$b : A_1, \ldots, b : A_n \vdash_{\nabla} b : A \qquad \Rightarrow \qquad \Gamma^{\bullet} \models_{LTL} A^{\bullet}\,.$$

Proof. By Corollary 4.68, $b : A_1, \ldots, b : A_n \vdash_{\nabla} b : A$ implies $\Gamma \models_{\nabla} A$. By Theorem 4.66, $\Gamma \models_{\nabla} A$ implies $\Gamma^{\bullet} \models_{LTL} A^{\bullet}$. □

4.4.6 Completeness

In order to prove the completeness of the system $\mathcal{N}(LTL_{\nabla})$, we can exploit the equivalence shown in Section 4.4.3 and use the Hilbert-style axiomatization $\mathcal{H}(LTL)$ of Section 2.3.4. The proposed natural deduction system consists of only finitary rules; consequently, it cannot be strongly complete for *LTL* (see also the discussion in Section 4.2). Nevertheless, by using the translation $(\cdot)^{*}$, we can give a proof of weak completeness for it. First, we introduce a lemma that will be useful in proving completeness.

Lemma 4.70. *If A is an LTL-formula, then A^{*} is a history- independent formula.*

Proof. It follows easily from Definition 4.52. The proof proceeds by induction on the complexity of the formula A. □

Theorem 4.71. *Let A be an LTL-formula and b a label. Then*

$$\models_{LTL} A \qquad \Rightarrow \qquad \vdash_{\nabla} b : A^{*}\,.$$

Proof. We can prove the theorem by showing that $\mathcal{N}(LTL_{\nabla})$ is complete with respect to the (translation of the) axiomatization $\mathcal{H}(LTL)$ given in Section 2.3.4, which is sound and complete for the logic *LTL*. That is, we need to prove that: (i) the translation, via $(\cdot)^{*}$, of every axiom of $\mathcal{H}(LTL)$ is provable in $\mathcal{N}(LTL_{\nabla})$ by

means of an *LTL*-derivation, and (ii) the notion of $\vdash_{LTL}$ is closed under the (labeled equivalent of the) rules of inference of $\mathcal{H}(LTL)$.

We focus on (i); showing (ii) is straightforward and we omit it here.

Note that, for simplicity, we use also some rules (i.e., $\mathsf{F}I$, $\mathsf{F}E$, $\vee I$, $\vee E$, $\wedge I$ and $\wedge E$) concerning derived operators. They can be easily derived from the set of rules in Figure 4.19.

We also remark that, by Lemma 4.70, our use of the rule *last* in the following derivations respects the side-conditions of the rule, i.e. the premise (and thus the conclusion) of each application of *last* is a history-independent labeled formula.

(A2)

$$\dfrac{\dfrac{\dfrac{[b : \mathsf{G}(A \supset B)]^1 \quad [b \leqslant c]^3}{bc : A \supset B}\,\mathsf{G}E \quad \dfrac{[b : \mathsf{G}A]^2 \quad [b \leqslant c]^3}{bc : A}\,\mathsf{G}E}{bc : B}\supset E}{b : \mathsf{G}B}\,\mathsf{G}I^3}{b : \mathsf{G}A \supset \mathsf{G}B}\,\supset I^2}{b : \mathsf{G}(A \supset B) \supset (\mathsf{G}A \supset \mathsf{G}B)}\,\supset I^1$$

(A3)

$(\mathsf{X}\neg A \leftrightarrow \neg \mathsf{X}A)$

$$\dfrac{\dfrac{\dfrac{[b : \mathsf{X}\neg A]^1 \quad [b \lhd c]^2}{bc : \neg A}\,\mathsf{X}E \quad \dfrac{[b : \mathsf{X}A]^3 \quad [b \lhd c]^2}{bc : A}\,\mathsf{X}E}{bc : \bot}\supset E}{b : \neg \mathsf{X}A}\,\bot E^3}{b : \neg \mathsf{X}A}\,ser\lhd^2}{b : \mathsf{X}\neg A \supset \neg \mathsf{X}A}\,\supset I^1$$

$$\dfrac{\dfrac{\dfrac{[b : \neg \mathsf{X}A]^1 \qquad \dfrac{\dfrac{[b \lhd c]^2 \quad [b \lhd d]^4 \quad [bc : A]^3}{bd : A}\,lin\lhd}{b : \mathsf{X}A}\,\mathsf{X}I^4}{b : \bot}\supset E}{bc : \neg A}\,\supset^3}{b : \mathsf{X}\neg A}\,\mathsf{X}I^2}{b : \neg \mathsf{X}A \supset \mathsf{X}\neg A}\,\supset I^1$$

(A4)

This proof is very similar to the one for *(A2)* and we thus omit it.

(A5)

$$\dfrac{\dfrac{[b : \mathsf{G}A]^1 \quad [b \leqslant b]^2}{\dfrac{\dfrac{bb : A}{b : A}\, last}{b : A}\, refl \leqslant^2}\, \mathsf{G}E \qquad \dfrac{\dfrac{\dfrac{[b \lhd c]^3 \quad \dfrac{\dfrac{[b \leqslant c]^5 \quad [c \leqslant d]^4}{bd : A}\, trans \leqslant^6 \quad \dfrac{[b : \mathsf{G}A]^1 \quad [b \leqslant d]^6}{bd : A}\, \mathsf{G}E}{bd : A}\, base \leqslant^5}{bd : A}\, last}{cd : A}\, \mathsf{G}I^4}{c : \mathsf{G}A}\, last}{bc : \mathsf{G}A}\, last}{b : \mathsf{X}\mathsf{G}A}\, \mathsf{X}I^3}{b : A \wedge \mathsf{X}\mathsf{G}A}\, \wedge I}{b : \mathsf{G}A \supset (A \wedge \mathsf{X}\mathsf{G}A)}\, \supset I^1$$

$(A6)$

$$\dfrac{\dfrac{\dfrac{[b : A]^2 \quad [b \leqslant c]^3 \qquad \dfrac{\dfrac{\dfrac{[b : \mathsf{G}(A \supset \mathsf{X}A)]^1 \quad [b \leqslant b_i]^4}{bb_i : A \supset \mathsf{X}A}\, \mathsf{G}E \quad \dfrac{[b_i : A]^4}{bb_i : A}\, last}{bb_i : \mathsf{X}A}\, \supset E \qquad [b_i \lhd b_j]^4}{b_i b_j : A}\, \mathsf{X}E}{b_j : A}\, last}{c : A}\, ind^4}{bc : A}\, last}{b : \mathsf{G}A}\, \supset^3}{b : A \supset \mathsf{G}A}\, \supset I^2}{b : \mathsf{G}(A \supset \mathsf{X}A) \supset (A \supset \mathsf{G}A)}\, \supset I^1$$

$(A7)$

Derivations are presented in Figures 4.20 and 4.21. Note that, for brevity, we give a derivation of a, clearly equivalent, simplified version of the translation of $(A7)$. Namely, we consider $\mathsf{F}(\mathsf{X}B \wedge \nabla A) \supset (A \wedge \mathsf{X}(B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A)))$ instead of $B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A) \supset B \vee (A \wedge \mathsf{X}(B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A)))$.

$(A8)$

A proof of the axiom $(A8)$ is given in Figure 4.22.

□

Theorem 4.73 below expresses a form of completeness with regard to LTL_∇. It is based upon the composed translation $((\cdot)^\bullet)^*$, going first from LTL_∇ into LTL and then back into LTL_∇. We need to remark that the result of such a translation is a formula that is semantically (but not necessarily syntactically) equivalent to the original one, as shown by the following example.

Example 4.72. Consider the formula $A = \nabla p \vee \nabla q$. Then we have $A^\bullet = p \vee q$ and $(A^\bullet)^* = p \vee q$. A and $(A^\bullet)^*$, though syntactically different, are semantically equivalent.

In other words, if we are interested in reasoning on LTL_∇, we can reduce the problem of finding a derivation for a given LTL_∇-formula A to the problem of finding a derivation for the formula $(A^\bullet)^*$, which is semantically equivalent to A and for which a derivation in $\mathcal{N}(LTL_\nabla)$ exists.

Left-to-right direction:

$$\dfrac{[b : \mathsf{F}(\mathsf{X}B \wedge \nabla A)]^1 \qquad \dfrac{\dfrac{\dfrac{[bc : \mathsf{X}B \wedge \nabla A]^2}{bc : \nabla A}\wedge E \quad \dfrac{[b \leqslant b]^3 \quad [b \leqslant c]^2}{\dfrac{b : A}{b : A}\, refl \leqslant^3}\nabla E \qquad \overset{\Pi_1}{b : \mathsf{X}(B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A))}}{b : A \wedge \mathsf{X}(B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A))}\wedge I}{b : A \wedge \mathsf{X}(B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A))}\mathsf{F}E^2}{b : \mathsf{F}(\mathsf{X}B \wedge \nabla A) \supset (A \wedge \mathsf{X}(B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A)))}\supset I^1$$

where Π_1 is the following derivation:

$$\dfrac{\dfrac{[b \leqslant c]^2 \quad [b \lhd b']^4 \qquad \dfrac{\dfrac{\dfrac{[bc : \mathsf{X}B \wedge \nabla A]^2}{bc : \mathsf{X}B}\wedge E \quad [c \lhd b']^5}{\dfrac{cb' : B}{bb' : B}\, last}\mathsf{X}E}{bb' : B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A)}\vee I \qquad \overset{\Pi_2}{\dfrac{b \lhd b' \quad b \lhd b'' \quad bb'' : B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A) \quad [bb' : B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A)]^6}{bb' : B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A)}\, lin\lhd^6}}{bb' : B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A)}\, split \leqslant^5}{b : \mathsf{X}(B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A))}\mathsf{X}I^4$$

and Π_2 is the following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{[bc : \mathsf{X}B \wedge \nabla A]^2}{bc : \mathsf{X}B}\wedge E \quad [c \lhd c']^7}{\dfrac{cc' : B}{c : \mathsf{X}B}\mathsf{X}I^7}\mathsf{X}E \qquad \dfrac{[b \lhd b'']^5 \qquad \dfrac{\dfrac{[b \leqslant b'']^9 \quad [b'' \leqslant d]^8}{d : A}\, base \leqslant^9 \qquad \dfrac{\dfrac{[bc : \mathsf{X}B \wedge \nabla A]^2}{bc : \nabla A}\wedge E \quad \dfrac{[b \leqslant d]^{10} \quad [d \leqslant c]^8}{d : A}\nabla E}{d : A}\, trans \leqslant^{10}}{\dfrac{d : A}{b'' c : \nabla A}\nabla I^8}}{b'' c : \mathsf{X}B \wedge \nabla A}\wedge I}{\dfrac{b'' : \mathsf{F}(\mathsf{X}B \wedge \nabla A)}{b' : B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A)}\vee I}\, last}{bb'' : B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A)}\, last \qquad [b'' \leqslant c]^5}\mathsf{F}I$$

Fig. 4.20. Proof of the Axiom ($A7$): left-to-right direction.

Right-to-left direction: in the following derivations, we denote with φ the formula $b : A \wedge \mathsf{X}(B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A))$.

$$\dfrac{\dfrac{\dfrac{\dfrac{[\varphi]^1}{b : \mathsf{X}(B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A))}\ \wedge E \quad [b \lhd e]^2}{be : B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A)}\ \mathsf{X}E \quad \begin{matrix}[be : B]^3 \\ \Pi_1 \\ b : \mathsf{F}(\mathsf{X}B \wedge \nabla A)\end{matrix} \quad \begin{matrix}[be : \mathsf{F}(\mathsf{X}B \wedge \nabla A)]^3 \\ \Pi_2 \\ b : \mathsf{F}(\mathsf{X}B \wedge \nabla A)\end{matrix}}{b : \mathsf{F}(\mathsf{X}B \wedge \nabla A)}\ \vee E^3}{b : \mathsf{F}(\mathsf{X}B \wedge \nabla A)}\ ser\lhd^2}{b : (A \wedge \mathsf{X}(B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A))) \supset \mathsf{F}(\mathsf{X}B \wedge \nabla A)}\ \supset I^1$$

where Π_1 is the following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{[b \lhd e]^2 \quad [b \lhd f]^5 \quad [be : B]^3 \quad [bf : B]^6}{bf : B}\ lin\lhd^6}{b : \mathsf{X}B}\ \mathsf{X}I^5}{bb : \mathsf{X}B}\ last \quad \dfrac{\dfrac{[b \leqslant b']^7 \quad [b' \leqslant b]^7 \quad \dfrac{[\varphi]^1}{b : A}\ \wedge E}{b' : A}\ eq \leqslant}{bb : \nabla A}\ \nabla I^7}{bb : \mathsf{X}B \wedge \nabla A}\ \wedge I \quad [b \leqslant b]^4}{b : \mathsf{F}(\mathsf{X}B \wedge \nabla A)}\ \mathsf{F}I$$
$$\dfrac{}{b : \mathsf{F}(\mathsf{X}B \wedge \nabla A)}\ refl \leqslant^4$$

Π_2 is the following derivation:

$$\dfrac{[be : \mathsf{F}(\mathsf{X}B \wedge \nabla A)]^3 \quad \dfrac{[b \lhd e]^2 \quad \dfrac{[b \leqslant e]^9 \quad [e \leqslant c]^8 \quad \dfrac{\dfrac{\dfrac{\dfrac{[c : \mathsf{X}B \wedge \nabla A]^8}{c : \mathsf{X}B}\ \wedge E \quad [c \lhd f]^{11}}{cf : B}\ \mathsf{X}I^{11}}{c : \mathsf{X}B}\ last \quad \begin{matrix}\Pi_3 \\ bc : \nabla A\end{matrix}}{bc : \mathsf{X}B \wedge \nabla A}\ \wedge I \quad [b \leqslant c]^{10}}{b : \mathsf{F}(\mathsf{X}B \wedge \nabla A)}\ \mathsf{F}I}{b : \mathsf{F}(\mathsf{X}B \wedge \nabla A)}\ trans \leqslant^{10}}{b : \mathsf{F}(\mathsf{X}B \wedge \nabla A)}\ base \leqslant^9}{b : \mathsf{F}(\mathsf{X}B \wedge \nabla A)}\ \mathsf{F}E^8$$

and Π_3 is the following derivation:

$$\dfrac{[b \leqslant d]^{12} \quad \dfrac{[\varphi]^1}{b : A}\ \wedge E \quad [d : A]^{13} \quad \dfrac{[b \lhd f]^{13} \quad [b \lhd e]^2 \quad [f \leqslant d]^{13} \quad \dfrac{\dfrac{[ec : \mathsf{X}B \wedge \nabla A]^8}{ec : \nabla A}\ \wedge E \quad [e \leqslant d]^{14} \quad [d \leqslant c]^{12}}{d : A}\ \nabla E}{d : A}\ lin\lhd^{14}}{d : A}\ split \leqslant^{13}}{bc : \nabla A}\ \nabla I^{12}$$

Fig. 4.21. Proof of the Axiom ($A7$): right-to-left direction.

$$\dfrac{\dfrac{[b : B \vee (\mathsf{F}(\mathsf{X}B \wedge \nabla A))]^1 \qquad \dfrac{\dfrac{\dfrac{[b : B]^2}{bb : B}\ last \qquad [b \leqslant b]^3}{b : \mathsf{F}B}\ \mathsf{F}I}{b : \mathsf{F}B}\ refl \leqslant^3 \qquad \dfrac{[b : \mathsf{F}(\mathsf{X}B \wedge \nabla A)]^2 \qquad \dfrac{\dfrac{\overset{\Pi}{[c \lhd d]^5 \quad b : \mathsf{F}B}}{b : \mathsf{F}B}\ ser\lhd^5}{b : \mathsf{F}B}\ \mathsf{F}E^4}{b : \mathsf{F}B}\ \vee E^2}{b : B \vee (\mathsf{F}(\mathsf{X}B \wedge \nabla A)) \supset \mathsf{F}B}\ \supset I^1$$

where Π is the following derivation:

$$\dfrac{[b \leqslant c]^4 \quad [c \leqslant d]^6 \qquad \dfrac{\dfrac{\dfrac{[bc : \mathsf{X}B \wedge \nabla A]^4}{bc : \mathsf{X}B}\ \wedge E \qquad [c \lhd d]^5}{bcd : B}\ \mathsf{X}E}{bd : B}\ last \qquad [b \leqslant d]^7}{b : \mathsf{F}B}\ \mathsf{F}I}{b : \mathsf{F}B}\ trans \leqslant^7$$

Fig. 4.22. Proof of the axiom ($A8$).

Theorem 4.73. *Let A be an LTL_{∇}-formula and b a label. Then*

$$\models_{\nabla} A \qquad \Rightarrow \qquad \vdash_{\nabla} b : (A^{\bullet})^{*} .$$

Proof. By Theorem 4.66, $\models_{\nabla} A$ implies $\models_{LTL} A^{\bullet}$. By Theorem 4.71, $\models_{LTL} A^{\bullet}$ implies $\vdash_{\nabla} b : (A^{\bullet})^{*}$.

□

4.4.7 Discussion and related works

The introduction of the operator ∇ has allowed us to formalize the "history" of until and thus, via a proper translation, to give a labeled natural deduction system for a linear-time logic endowed with ∇ that is also sound and complete with respect to *LTL* with until. We remark that the "recipe" for dealing with until that we gave here is abstract and general, and thus provides the basis for formalizing deduction systems for temporal logics endowed with U, both linear and branching time.

In this section, we did not address normalization matters explicitly. However the well-behaved nature of this approach, where each connective and operator has one introduction and one elimination rule, paves the way to a proof-theoretical analysis of the resulting natural deduction systems, e.g., to show proof normalization and other useful meta-theoretical properties. In fact, the procedure of normalization presented in Section 4.3 for linear-time logics (and the one that will be presented in Section 5.3 with regard to a branching-time logic as well) can be easily adapted to deal also with the rules for ∇ given here.

With regard to the discussion on the rule *last*, we believe that the restriction we imposed, i.e., the rule can only be applied to history-independent formulas, is closely related, at least in spirit, to the focus on *persistent* formulas when combining intuitionistic and classical logic so as to avoid the collapse of the two logics into one, see [46] but also [35,67]. We are, after all, considering here formulas stemming from two classes (if not two logics altogether), and it makes thus sense that they require different labeling (single instants and pairs of time instants).

In [101], an extension of a linear-time temporal logic with past is presented, where a unary operator *now* is used in order to fix a point of evaluation. When used in combination with past operators, *now* allows to "forget" part of the past. The resulting logic is proved to be equally expressive to, but more succinct than, *LTL* with past[19].

A class of logics extending the expressivity of standard temporal logics is that of *hybrid temporal logics*, where the possibility of referring to worlds (instants) of a model is internalized in the syntax of the logic itself and not just used as a technical device when performing deduction like we use to do in our systems. Early examples are in [29, 128]; more recent works in [3, 17, 77]. Many other works have proposed interesting extensions of temporal logics with new operators, e.g., [48,77].

Finally, it is worth observing that several works have considered *interval temporal logics*, e.g., [25,37,78,84,145]. While these works consider intervals explicitly,

[19] We also remark that [71] proves that *LTL* is expressively complete, thus as expressive as *PLTL*, i.e., *LTL* with past [96]. Furthermore, [65] presents an algorithm for the translation of a *PLTL* formula into an *LTL* one that is initially equivalent.

we have used them somehow implicitly here, as a means to formalize the dual nature of until via the history ∇.

5

Labeled Natural Deduction for Branching Temporal Logics

5.1 Introduction

In Chapter 4, we presented labeled natural deduction systems for a wide range of linear temporal logics and shown that such systems are well-behaved, in the sense that their derivations enjoy some godo structural properties. In this chapter, we propose extensions of the systems already presented in order to capture branching temporal logics. In particular, we will use as a starting point the framework adopted in Section 4.2, i.e., we will define systems without an explicit relational labeling algebra.

The extension to branching logics will require the definition of rules for treating the path quantifier $\forall$ (or, equivalently, its dual $\exists$). The intuition we move from is that, as shown by the systems in Chapters 3 and 4, labeling allows for devising clean and effective natural deduction rules for the introduction and elimination of operators, at least as long as we are able to consider them as "pure" modal operators. We have seen in Section 2.4.1 that the semantics of bundled branching logics can be given in terms of Ockhamist frames and that this gives us the possibility of defining the notion of truth in a purely Kripkean style, according to an interpretation that sees the branches as the worlds of our structures. With such an interpretation in mind, we can consider also the path quantifier $\forall$ as a standard ($S5$) modal operator with respect to the accessibility (equivalence) relation, defined on branches, of having the same initial node.

It follows that the rules for introduction and elimination of $\forall$ can be given by following the same pattern of the other modal and temporal operators, i.e.,

$$\frac{\begin{array}{c}[b_1 \bullet b_2] \\ \vdots \\ b_2 : A\end{array}}{b_1 : \forall A}\ \forall I \qquad \frac{b_1 : \forall A \quad b_1 \bullet b_2}{b_2 : A}\ \forall E$$

where we use $\bullet$ as the syntactic corresponding of $\simeq$ and impose the standard condition that b_2 is fresh in $\forall I$. Relational properties of $\simeq$, i.e., reflexivity, symmetry and transitivity, are also easy to capture by means of labeled natural deduction rules (see the analogous rules presented in Chapter 4 with respect to other relations).

Finally, we need rules expressing the interactions between the relations $\simeq$ and $\prec$ (and/or $\lhd$ if we are in the discrete case) and thus expressing the branching nature of the particular logic we want to capture. Such rules are devised in such a way that operators are neither introduced nor eliminated.

This approach makes it easy and natural to define labeled natural deduction systems for *Ockhamist* branching temporal logics, i.e., for those branching-time logics where there are no restrictions on the nesting of the operators. In fact, in this chapter, we will define natural deduction systems for several such logics.[1]

In Section 5.2, we will start by defining a sound and complete system for a simple logic (the logic of basic frames [167]), where we have no interdependencies between $\simeq$ and $\prec$. Then we will proceed by modularly enriching such a system with rules specifying interaction properties in order to capture other bundled Ockhamist logics.

In Section 5.3, we will consider computation tree logics and define a sound and (weakly) complete system for the logic $BCTL^*_{\_}$. A detailed proof-theoretical analysis of the system will be also made. As already remarked in Section 4.2.4, when we considered a system for $LTL_{\_}$, the main problem in considering normalization of systems for logics with both the operators X and G arises from dealing with the underlying induction principle, which relates the next-time relation and the order relation. Such temporal induction is handled, inside the system, in a way strongly similar to first-order induction of Peano/Heyting Arithmetics and in fact the normalization procedure will follow those defined for systems for Heyting Arithmetics in [74,126,151]. We will present an intuitionistic version[2] of the system and prove its *confluence* and *weak normalization*; consequently, we will use such results to give a purely syntactical proof of consistency (for both the intuitionistic and classical versions) of the deduction system.

We remark that here we limit ourselves to consider *bundled* branching temporal logics. In fact, considering the *full* semantics, both in the case of the "philosophical" logics of Section 5.2 and in the case of computation tree logics, introduces a complexity that we are not able to deal with in terms of finitary natural deduction rules. Indeed, as discussed in Section 5.5, even the definition of (standard) finitary Hilbert-style axiomatizations for the full Ockhamist logic and for CTL^* are still open problems.

Finally, we remark that in this chapter, for simplicity, we will consider only until-free logics. We recall, however, that the recipe for the treatment of the operator until, formalized in Section 4.4 in the case of a linear-time logic, is general and can be easily adapted to the branching case.

[1] We note anyway that Peircean logics can be obtained by the Ockhamist ones by just imposing a restriction on the language. Thus our systems can be also used for reasoning on Peircean logics, e.g., by considering a restriction on the set of admissible derivations.

[2] Moving to intuitionistic systems for studying normalization is also standard in such cases. The results obtained, e.g., a proof of consistency, can be then extended to the classical system by considering a proper translation.

5.2 Systems for bundled Ockhamist logics with general time

In this section, we first define a labeled natural deduction system $\mathcal{N}(bas)$ for the logics of basic frames and (Dis)-frames (which are proved to be equivalent in [167]). Then we extend such a system in order to consider other bundled Ockhamist logics (see Section 2.4.1). All the systems are shown to be sound and complete.

5.2.1 A system for the logic of basic frames

A labeled version of the logic of basic frames

As usual, we need to formalize the extension of the language and the adaptations to the semantics required by the labeled deduction setting. We use $<$ to denote the order relation between points of a basic frame and, as indicated above, $\bullet$ for the corresponding of $\simeq$.

Definition 5.1. *Let L be a denumerable set of labels, $<$ and $\bullet$ two binary relation symbols over L. If b and c are labels in L and A is an Ockhamist formula, then $b < c$ and $b \bullet c$ are* relational well-formed (Ockhamist) formulas *(or relational formulas, or rwffs for short) and $b : A$ is a* labeled well-formed (Ockhamist) formula *(or labeled formula, or lwff for short).*

The notion of interpretation can be adapted to the case of this logic in a standard way.

Definition 5.2. *Given a denumerable set of labels L and a basic structure $\mathcal{M} = (\mathcal{W}, \prec, \simeq, \mathcal{V})$, an* interpretation *is a function $\lambda : L \to \mathcal{W}$ that maps every label in L to an element of $\mathcal{W}$.*

Definition 5.3. *Given a basic structure $\mathcal{M} = (\mathcal{W}, \prec, \simeq, \mathcal{V})$, a denumerable set L of labels and an interpretation λ on them,* truth *for a labeled or relational formula φ at a pair $(\mathcal{M}, \lambda)$ is the smallest relation $\models_{bas}$ satisfying:*

$$\begin{array}{rll} \mathcal{M}, \lambda \models_{bas} b < c & \textit{iff} & \lambda(b) \prec \lambda(c) \\ \mathcal{M}, \lambda \models_{bas} b \bullet c & \textit{iff} & \lambda(b) \simeq \lambda(c) \\ \mathcal{M}, \lambda \models_{bas} b : A & \textit{iff} & \mathcal{M}, \lambda(b) \models_{bas} A \end{array}$$

Given a set Γ of generic formulas and a generic formula φ, we say that:

$$\begin{array}{rll} \mathcal{M}, \lambda \models_{bas} \Gamma & \textit{iff} & \mathcal{M}, \lambda \models_{bas} \varphi \textit{ for all } \varphi \in \Gamma \\ \Gamma \models_{bas} \varphi & \textit{iff} & \mathcal{M}, \lambda \models_{bas} \Gamma \textit{ implies } \mathcal{M}, \lambda \models_{bas} \varphi \textit{ for all } \mathcal{M} \textit{ and } \lambda \end{array}$$

The system $\mathcal{N}(bas)$

The complete set of rules of the system $\mathcal{N}(bas)$ is presented in Figure 5.1. Rules for classical connectives and linear temporal operators are as seen in Chapter 4. Indeed, the core of the system is given by the rules of $\mathcal{N}(Kl)$ (Section 4.2.2) which cover the linear part of the logic. We just remark that, when dealing with branching

logics, the condition of linearity captured by the rule *conn* $<$ requires a slightly more complex formulation[3] and needs to be split into two rules, one related to the future and one related to the past. The one related to the future, *conn* $<_R$, says that if both b_2 and b_3 are sub-branches of b_1, then one of the following facts holds:

1. $b_2 = b_3$, and then if a formula B holds in b_2 it must also hold in b_3 (again, as in Section 4.2, we express equality indirectly); or
2. $b_2 \prec b_3$; or
3. $b_3 \prec b_2$.

The structure of *conn* $<_L$ is symmetrical.

As anticipated in Section 5.1, the rules for introduction and elimination of $\forall$ mirror those for X and G. The rule *atom*$\bullet$ captures the property of basic structures according to which if $u \simeq v$ then $\mathcal{V}(u) = \mathcal{V}(v)$ (see Definition 2.20) and is the equivalent of the axiom $(Atom)$ in $\mathcal{H}(bas)$.

The set of rules of $\mathcal{N}(bas)$ is completed by *refl*$\bullet$, *symm*$\bullet$ and *trans*$\bullet$, which express reflexivity, symmetry and transitivity of the relation $\simeq$, respectively.

As is standard, $\vdash_{\mathcal{N}(bas)}$ denotes the notion of derivability in the system $\mathcal{N}(bas)$. The notions of *derivation* and *theorem* are also standard (see Section 3.2).

In addition to the derived rules for other classical connectives and temporal operators given in Section 4.2.1, we will use sometimes the following derived rules for introduction/elimination of $\exists$, which is the dual of the path quantifier $\forall$:

$$\frac{b_2 : A \quad b_1 \bullet b_2}{b_2 : \exists A}\ \exists I \qquad \frac{c : \exists A \qquad \begin{array}{c}[c \bullet c'] \quad [c' : A] \\ \vdots \\ b : A\end{array}}{b : A}\ \exists E$$

where c' is required to be fresh in $\exists E$.

Soundness

Theorem 5.4. *Let Γ be a set of labeled and relational Ockhamist formulas and $b : A$ a labeled Ockhamist formula. Then*

$$\Gamma \vdash_{\mathcal{N}(bas)} b : A \quad \Rightarrow \quad \Gamma \models_{bas} b : A\,.$$

Proof. The proof is by induction on the length of the derivation. We have one case for each rule; some have already been treated for the analogous rules of the systems in Chapter 4. As further examples, w e show here some new cases: the rules for the quantifier $\forall$, though they can be treated in a way similar to that of the rules for the other temporal operators, and the rule *atom*$\bullet$.

$(\forall I)$ Consider an application of the rule $\forall I$

$$\frac{\begin{array}{c}[b_1 \bullet b_2] \\ \Pi \\ b_2 : A\end{array}}{b_1 : \forall A}\ \forall I$$

[3] The reason is that given two worlds b and c of an Ockhamist structure, it is not true that either $b \prec c$ holds or $c \prec b$ holds; they may also be $\prec$-unrelated.

$$\frac{\begin{array}{c}[b_1 : A \supset \bot]\\ \vdots\\ b_2 : \bot\end{array}}{b_1 : A}\ \bot E \qquad \frac{\begin{array}{c}[b : A]\\ \vdots\\ b : B\end{array}}{b : A \supset B}\ \supset I \qquad \frac{b : A \supset B \quad b : A}{b : B}\ \supset E$$

$$\frac{\begin{array}{c}[b_1 < b_2]\\ \vdots\\ b_2 : A\end{array}}{b_1 : \mathsf{G}A}\ \mathsf{G}I \qquad \frac{b_1 : \mathsf{G}A \quad b_1 < b_2}{b_2 : A}\ \mathsf{G}E \qquad \frac{\begin{array}{c}[b_1 < b_2]\\ \vdots\\ b_1 : A\end{array}}{b_2 : \mathsf{H}A}\ \mathsf{H}I \qquad \frac{b_2 : \mathsf{H}A \quad b_1 < b_2}{b_1 : A}\ \mathsf{H}E$$

$$\frac{b_1 < b_2 \quad b_2 < b_3 \qquad \begin{array}{c}[b_1 < b_3]\\ \vdots\\ b : A\end{array}}{b : A}\ \mathit{trans} <$$

$$\frac{b_1 < b_2 \quad b_1 < b_3 \quad b_2 : B \qquad \begin{array}{c}[b_3 : B]\\ \vdots\\ b : A\end{array} \qquad \begin{array}{c}[b_2 < b_3]\\ \vdots\\ b : A\end{array} \qquad \begin{array}{c}[b_3 < b_2]\\ \vdots\\ b : A\end{array}}{b : A}\ \mathit{conn} <_R$$

$$\frac{b_2 < b_1 \quad b_3 < b_1 \quad b_2 : B \qquad \begin{array}{c}[b_3 : B]\\ \vdots\\ b : A\end{array} \qquad \begin{array}{c}[b_2 < b_3]\\ \vdots\\ b : A\end{array} \qquad \begin{array}{c}[b_3 < b_2]\\ \vdots\\ b : A\end{array}}{b : A}\ \mathit{conn} <_L$$

$$\frac{\begin{array}{c}[b_1 \bullet b_2]\\ \vdots\\ b_2 : A\end{array}}{b_1 : \forall A}\ \forall I \qquad \frac{b_1 : \forall A \quad b_1 \bullet b_2}{b_2 : A}\ \forall E \qquad \frac{b_1 : p \quad b_1 \bullet b_2}{b_2 : p}\ \mathit{atom}\bullet$$

$$\frac{\begin{array}{c}[b_1 \bullet b_1]\\ \vdots\\ b : A\end{array}}{b : A}\ \mathit{refl}\bullet \qquad \frac{b_1 \bullet b_2 \qquad \begin{array}{c}[b_2 \bullet b_1]\\ \vdots\\ b : A\end{array}}{b : A}\ \mathit{symm}\bullet \qquad \frac{b_1 \bullet b_2 \quad b_2 \bullet b_3 \qquad \begin{array}{c}[b_1 \bullet b_3]\\ \vdots\\ b : A\end{array}}{b : A}\ \mathit{trans}\bullet$$

- In $\mathsf{G}I$, b_2 is *fresh*, i.e., it is different from b_1 and does not occur in any assumption on which $b_2 : A$ depends other than the discharged assumption $b_1 < b_2$.
- In $\mathsf{H}I$, b_1 is *fresh*, i.e., it is different from b_2 and does not occur in any assumption on which $b_1 : A$ depends other than the discharged assumption $b_1 < b_2$.
- In $\forall I$, b_2 is *fresh*, i.e., it is different from b_1 and does not occur in any assumption on which $b_2 : A$ depends other than the discharged assumption $b_1 \bullet b_2$.
- In $\mathit{atom}\bullet$, p is an atomic proposition.

Fig. 5.1. The rules of $\mathcal{N}(\mathit{bas})$.

where Π is a proof of $b_2 : A$ from hypotheses in Γ', with b_2 fresh and with $\Gamma' = \Gamma \cup \{b_1 \bullet b_2\}$. By the induction hypothesis, for all interpretations λ, if $\mathcal{M}, \lambda \models_{bas} \Gamma'$ then $\mathcal{M}, \lambda \models_{bas} b_2 : A$. We let $\mathcal{M} = (\mathcal{W}, \prec, \simeq, \mathcal{V})$ and λ be any basic structure and interpretation such that $\mathcal{M}, \lambda \models_{bas} \Gamma$, and show that $\mathcal{M}, \lambda \models_{bas} b_1 : \forall A$. Let $\lambda(b_1) = w$, for some world w in the set $\mathcal{W}$. Now let us consider a generic world w' such that $w \simeq w'$. Since λ can be trivially extended to another interpretation (still called λ for simplicity) by setting $\lambda(b_2) = w'$, the induction hypothesis yields $\mathcal{M}, \lambda \models_{bas} b_2 : A$, i.e. $\mathcal{M}, w' \models_{bas} A$. Given that w' is an arbitrary world $\simeq$-related to w, we can conclude $\mathcal{M}, \lambda \models_{bas} b_1 : \forall A$.

($\forall E$) Consider the case in which the last rule applied is $\forall E$:

$$\frac{\begin{array}{cc}\Pi & \\ b_1 : \forall A & b_1 \bullet b_2\end{array}}{b_2 : A}\ \forall E$$

where Π is a proof of $b_1 : \forall A$ from hypotheses in Γ_1, with $\Gamma = \Gamma_1 \cup \{b_1 \bullet b_2\}$ for some set Γ_1 of formulas. By applying the induction hypothesis on Π, we have:

$$\Gamma_1 \models_{bas} b_1 : \forall A\ .$$

Now we proceed by considering a generic basic structure $\mathcal{M} = (\mathcal{W}, \prec, \simeq, \mathcal{V})$ and a generic interpretation λ on it such that $\mathcal{M}, \lambda \models_{bas} \Gamma$ and showing that this entails

$$\mathcal{M}, \lambda \models_{bas} b_2 : A\ .$$

From $\Gamma \supset \Gamma_1$, we deduce (by the induction hypothesis) $\mathcal{M}, \lambda \models_{bas} b_1 : \forall A$. Furthermore $\mathcal{M}, \lambda \models_{bas} \Gamma$ entails $\mathcal{M}, \lambda \models_{bas} b_1 \bullet b_2$ and thus $\mathcal{M}, \lambda(b_2) \models_{bas} A$, i.e., by Definition 5.3, $\mathcal{M}, \lambda \models_{bas} b_2 : A$.

($atom\bullet$) Consider the case in which the last rule applied is $atom\bullet$:

$$\frac{\begin{array}{cc}\Pi & \\ b_1 : A & b_1 \bullet b\end{array}}{b : A}\ atom\bullet$$

where Π is a proof of $b_1 : A$ from hypotheses in Γ_1, with $\Gamma = \Gamma_1 \cup \{b_1 \bullet b\}$ for some set Γ_1 of formulas and A is an atomic proposition. By applying the induction hypothesis on Π, we have:

$$\Gamma_1 \models_{bas} b_1 : A\ .$$

Now we proceed by considering a generic basic structure $\mathcal{M} = (\mathcal{W}, \prec, \simeq, \mathcal{V})$ and a generic interpretation λ on it such that $\mathcal{M}, \lambda \models_{bas} \Gamma$ and showing that this entails

$$\mathcal{M}, \lambda \models_{bas} b : A\ .$$

By $\mathcal{M}, \lambda \models_{bas} \Gamma$ we deduce:

(i) $\mathcal{M}, \lambda \models_{bas} \Gamma_1$;
(ii) $\lambda(b_1) \simeq \lambda(b)$.

By the induction hypothesis, (i) yields $\mathcal{M}, \lambda \models_{bas} b_1 : A$. By Definition 2.20, (ii) yields $\mathcal{V}(\lambda(b_1)) = \mathcal{V}(\lambda(b))$. Since A is atomic, from $\mathcal{M}, \lambda \models_{bas} b_1 : A$ we can conclude $\mathcal{M}, \lambda \models_{bas} b : A$.

□

Completeness

Theorem 5.5. *Let Γ be a set of labeled Ockhamist formulas and $b : A$ a labeled Ockhamist formula. Then*

$$\Gamma \models_{bas} b : A \quad \Rightarrow \quad \Gamma \vdash_{\mathcal{N}(bas)} b : A\,.$$

Proof. In order to show that the system $\mathcal{N}(bas)$ is complete with respect to the semantics of the logic of basic frames (Definition 2.21), we need to prove that every axiom and rule of inference in the axiomatization $\mathcal{H}(bas)$ is provable in $\mathcal{N}(bas)$.

We omit the proofs for rules of inference, which are standard (see Section 4.2.1).

As for the axioms, we give derivations of the ones concerning linearity and of the ones related to the quantifier $\forall$. For the other axioms, the reader is referred to the proofs given in the case of the system $\mathcal{N}(Kt)$ (Section 4.2.1).

($K_\forall$)

$$\dfrac{\dfrac{\dfrac{\dfrac{[b : \forall(A \supset B)]^1 \quad [b \bullet c]^3}{c : A \supset B}\,\forall E \quad \dfrac{[b : \forall A]^2 \quad [b \bullet c]^3}{c : A}\,\forall E}{c : B}\,\supset E}{b : \forall B}\,\forall I^3}{b : \forall A \supset \forall B}\,\supset I^2}{b : \forall(A \supset B) \supset (\forall A \supset \forall B)}\,\supset I^1$$

($L1$)

$$\dfrac{[b : \mathsf{F}A]^1 \qquad \dfrac{[b < d]^3 \quad [b < c]^2 \quad [d : A]^3 \quad \dfrac{[c : A]^4}{\varphi}\,\vee I \quad \begin{matrix}\Pi_1\\ \varphi\end{matrix} \quad \begin{matrix}\Pi_2\\ \varphi\end{matrix}}{c : \mathsf{F}A \vee A \vee \mathsf{P}A}\,conn <^4_R}{\dfrac{\dfrac{c : \mathsf{F}A \vee A \vee \mathsf{P}A}{b : \mathsf{G}(\mathsf{F}A \vee A \vee \mathsf{P}A)}\,\mathsf{G}I^2}{b : \mathsf{F}A \supset \mathsf{G}(\mathsf{F}A \vee A \vee \mathsf{P}A)}\,\supset I^1}\,\mathsf{F}E^3$$

where Π_1 is

$$\dfrac{\dfrac{[d : A]^3 \quad [d < c]^4}{c : \mathsf{P}A}\,\mathsf{P}I}{\varphi}\,\vee I$$

and Π_2 is

$$\dfrac{\dfrac{[d : A]^3 \quad [c < d]^4}{c : \mathsf{F}A}\,\mathsf{F}I}{\varphi}\,\vee I$$

Note that we have used the abbreviation $\varphi \equiv c : \mathsf{F}A \vee A \vee \mathsf{P}A$ and we have slightly simplified the proof by using a generic $\vee I$ rule. The axiom ($L2$) can be derived in a symmetrical way.

($\forall 1$)

$$\dfrac{\dfrac{\dfrac{[b \bullet c]^2 \quad [c \bullet d]^3 \qquad \dfrac{[b : \forall A]^1 \quad [b \bullet d]^4}{d : A}\,\forall E}{d : A}\,trans\bullet^4}{c : \forall A}\,\forall I^3}{b : \forall\forall A}\,\forall I^2}{b : \forall A \supset \forall\forall A}\,\supset I^1$$

$(\forall 2)$

$$\dfrac{\dfrac{\dfrac{[b : \forall A]^1 \quad [b \bullet b]^2}{b : A}\,\forall E}{b : A}\,refl\bullet^2}{b : \forall A \supset A}\,\supset I^1$$

$(\forall 3)$

$$\dfrac{\dfrac{\dfrac{\dfrac{[c : \forall\neg A]^4 \quad [c \bullet b]^3}{b : \neg A}\,\forall E \qquad [b : A]^1}{b : \bot}\,\supset E}{c : \bot}\,\bot E}{c : \neg\forall\neg A}\,\supset I^4}{c : \neg\forall\neg A}\,symm\bullet^3}{b : \forall\neg\forall\neg A}\,\forall I^2}{b : A \supset \forall\neg\forall\neg A}\,\supset I^1$$

$(Atom)$

$$\dfrac{\dfrac{\dfrac{[b : p]^1 \quad [b \bullet c]^2}{c : p}\,atom\bullet}{b : \forall p}\,\forall I^2}{b : p \supset \forall p}\,\supset I^1$$

□

5.2.2 Systems for other bundled Ockhamist logics

In this section, we consider extensions of the system $\mathcal{N}(bas)$ aiming at capturing some of the extensions of the logic of basic frames presented in Section 2.4.1, namely the logic of (WDC)-frames, the logic of (Dis+WDC)-frames and the logic *BOBTL* of Ockhamist frames. We will show that a modular enrichment of the base system $\mathcal{N}(bas)$ with specific rules capturing the new properties will work.

We use the same labeled language defined for $\mathcal{N}(bas)$ (Section 5.2.1). The definition of interpretation and the notions of truth and validity are also standard and can be easily inferred from those of Section 5.2.1: just replace *basic structure* by the proper structure; we omit the details.

The logic of (WDC)-frames

In basic frames there is no interaction between the relations $\prec$ and $\simeq$. The first extension that we consider consists in requiring that the basic frames satisfy the property WDC (we recall it here for convenience):

(WDC) If $x \prec y \simeq y'$, then there exists x' such that $x \simeq x' \prec y'$.

We can capture such a property by adding the rule *wdc* below.

$$\frac{c \bullet c' \quad d < c \qquad\qquad \begin{array}{c}[d' < c'] \quad [d \bullet d'] \\ \vdots \\ b : A\end{array}}{b : A}\ wdc$$

where we require that d' is fresh.

A derivation of the axiom (*WDC*) (see Section 2.4.1), obtained by using the rules of $\mathcal{N}(bas)$ and *wdc*, is the following.

$$\dfrac{\dfrac{[b : \mathsf{P}A]^1 \qquad \dfrac{[b \bullet c]^2 \quad [d < b]^3 \qquad \dfrac{\dfrac{[d : A]^3 \quad [d \bullet d']^4}{d' : \exists A}\ \exists I \qquad [d' < c]^4}{c : \mathsf{P}\exists A}\ \mathsf{P}I}{c : \mathsf{P}\exists A}\ wdc^4}{c : \mathsf{P}\exists A}\ \mathsf{P}E^3}{b : \forall \mathsf{P}\exists A}\ \forall I^2}{b : \mathsf{P}A \supset \forall \mathsf{P}\exists A}\ \supset I^1$$

The logic of (Dis+WDC)-frames

As stated in Lemma 2.22, (Dis+WDC)-validity and (WDC+SDC)-validity coincide.

We recall here the property SDC, which is the one on which we will build our deduction rule:

(SDC) if $x \prec y \prec z \simeq z' \succ x' \simeq x$, then there exists y' such that $y' \simeq y$ and $x' \prec y' \prec z'$.

The following rule *sdc* models such a property.

$$\frac{b < c \quad c < d \quad b' \bullet b \quad d' \bullet d \quad b' < d' \qquad\qquad \begin{array}{c}[c' \bullet c] \quad [b' < c'] \quad [c' < d'] \\ \vdots \\ b : A\end{array}}{b : A}\ sdc$$

where we require that c' is fresh.

We present in Figures 5.2 and 5.3 a derivation of the axiom (*DW1*) (see Section 2.4.1), obtained by using the rules of $\mathcal{N}(bas)$, *wdc* and *sdc*. We omit the derivation of the axiom (*DW2*), which can be obtained similarly by using *conn* $<_R$ instead of *conn* $<_L$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Pi_1 \qquad \Pi_2}{c : \mathsf{P}(A \wedge (C \vee \mathsf{P}C)) \quad c : \mathsf{G}(C \supset \mathsf{G}A_1)}\wedge I}{c : \mathsf{P}(A \wedge (C \vee \mathsf{P}C)) \wedge \mathsf{G}(C \supset \mathsf{G}A_1)}\supset I^3}{c : (\mathsf{G}A_1 \wedge \mathsf{P}C) \supset \mathsf{P}(A \wedge (C \vee \mathsf{P}C)) \wedge \mathsf{G}(C \supset \mathsf{G}A_1)}\forall I^2}{b : \forall(\mathsf{G}A_1 \wedge \mathsf{P}C \supset \mathsf{P}(A \wedge (C \vee \mathsf{P}C)) \wedge \mathsf{G}(C \supset \mathsf{G}A_1))}\supset I^1}{b : (\mathsf{P}(\forall A \wedge \mathsf{G}B) \wedge \mathsf{H}\neg(B \wedge \exists C)) \supset \forall(\mathsf{G}A_1 \wedge \mathsf{P}C \supset \mathsf{P}(A \wedge (C \vee \mathsf{P}C)) \wedge \mathsf{G}(C \supset \mathsf{G}A_1))}$$

where Π_1 is the following derivation:

$$\dfrac{\dfrac{[b : \mathsf{P}(\forall A \wedge \mathsf{G}B) \wedge \mathsf{H}\neg(B \wedge \exists C)]^1}{b : \mathsf{P}(\forall A \wedge \mathsf{G}B)}\wedge E \quad \dfrac{[f < b]^8 \quad [b \bullet c]^2 \qquad \dfrac{\dfrac{\dfrac{\dfrac{[f : \forall A \wedge \mathsf{G}B]^8}{f : \forall A}\wedge I \quad [f \bullet d]^9}{d : A}\forall E \quad \dfrac{\Pi_3}{d : C \vee \mathsf{P}C}}{d : A \wedge (C \vee \mathsf{P}C)}\wedge I \quad [d < c]^9}{c : \mathsf{P}(A \wedge (C \vee \mathsf{P}C))}\mathsf{P}I}{c : \mathsf{P}(A \wedge (C \vee \mathsf{P}C))}wdc^9}{c : \mathsf{P}(A \wedge (C \vee \mathsf{P}C))}\mathsf{P}E^8$$

and Π_3 is the following derivation:

$$\dfrac{\dfrac{[c : \mathsf{G}A_1 \wedge \mathsf{P}C]^3}{c : \mathsf{P}C}\wedge E \quad \dfrac{[e < c]^{10} \quad [d < c]^9 \quad [e : C]^{10} \quad \dfrac{[d : C]^{11}}{d : C \vee \mathsf{P}C}\vee I \qquad \dfrac{\dfrac{[e : C]^{10} \quad [e < d]^{11}}{d : \mathsf{P}C}\mathsf{P}I}{d : C \vee \mathsf{P}C}\vee I \qquad \dfrac{\Pi_4}{d : C \vee \mathsf{P}C}}{d : C \vee \mathsf{P}C}conn <_L^{11}}{d : C \vee \mathsf{P}C}\mathsf{P}E^{10}$$

Fig. 5.2. A derivation of the axiom *DW1* (1/2).

Π_4 is the following derivation:

$$\dfrac{\dfrac{[d<e]^{11}\quad[e<c]^{10}\quad[f\bullet d]^{9}\quad[b\bullet c]^{2}\quad[f<b]^{8}}{d:C\vee \mathsf{P}C}\quad \dfrac{\dfrac{\dfrac{[b:\mathsf{P}(\forall A\wedge \mathsf{G}B)\wedge \mathsf{H}\neg(B\wedge \exists C)]^{1}}{b:\mathsf{H}(\neg B\vee \forall\neg C)}\wedge E\quad [e'<b]^{12}}{e':\neg B\vee\forall\neg C}\mathsf{H}E\quad \dfrac{\dfrac{[e':\neg B]^{13}\quad \dfrac{\dfrac{[f:\forall A\wedge \mathsf{G}B]^{8}}{f:\mathsf{G}B}\wedge E\quad [f<e']^{12}}{e:B}\mathsf{G}E}{e':\bot}\supset E}{d:C\vee \mathsf{P}C}\bot E\quad \dfrac{\Pi_5}{d:C\vee \mathsf{P}C}}{d:C\vee \mathsf{P}C}\vee E^{13}}{d:C\vee \mathsf{P}C}\,sdc^{12}$$

Π_5 is the following derivation:

$$\dfrac{\dfrac{[e':\forall\neg C]^{13}\quad [e'\bullet e]^{12}}{e:\neg C}\forall E\quad [e:C]^{10}}{d:C\vee \mathsf{P}C}\supset E$$

and Π_2 is the following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{[c:\mathsf{G}A_1\wedge \mathsf{P}C]^{3}}{c:\mathsf{G}A_1}\wedge E\quad [c<e]^{7}}{e:A_1}\mathsf{G}E}{\dfrac{[c<d]^{4}\quad [d<e]^{6}}{e:A_1}\,trans<^{7}}}{\dfrac{e:A_1}{d:\mathsf{G}A_1}\mathsf{G}I^{6}}}{\dfrac{d:C\supset \mathsf{G}A_1}{c:\mathsf{G}(C\supset \mathsf{G}A_1)}\mathsf{G}I^{4}}\supset I^{5}$$

Fig. 5.3. A derivation of the axiom *DW1* (2/2).

The logic *BOBTL*

Finally, we obtain Ockhamist frames by requiring the set of frames to satisfy also the property MB^{--}.

(MB^{--}) if x is a $\prec$-maximal element, and $x \simeq y$, then y is a $\prec$-maximal element.

A further extension of the system will contain the following rule:

$$\dfrac{\begin{array}{ccc} & & [d < d'] \\ & & \vdots \\ d \bullet c & c < c' & b : A \end{array}}{b : A}\ mb$$

where we require that d' is fresh.

A derivation of the corresponding axiom MB^{--} (see Section 2.4.1) in the extended system is the following:

$$\dfrac{\dfrac{\dfrac{[b : \exists \mathsf{F}\top]^2 \qquad \dfrac{\dfrac{[d : \mathsf{F}\top]^3 \qquad \dfrac{[b \bullet d]^3 \quad [d < e]^4 \qquad \dfrac{[b : \mathsf{G}\ \bot]^1 \quad [b < c]^5}{c : \bot}\ \mathsf{G}E}{c : \bot}\ mb^5}{c : \bot}\ \mathsf{F}E^4}{c : \bot}\ \exists E^3}{c : \bot}\ \bot E^2}{b : \mathsf{G}\ \bot \supset \forall \mathsf{G}\ \bot}\ \supset I^1$$

Soundness and completeness

Theorem 5.6. *The extensions of $\mathcal{N}(bas)$ presented in Section 5.2.2 are sound and complete with respect to the semantics of the corresponding logics.*

Proof. Soundness of the extended systems is easy to prove, since the rules mirror the properties that the frames of the extended logics are required to satisfy. We do not go into details and just remark that the proof is in the style of those of Section 4.2.

With regard to completeness, we have already presented derivations of (most of) the axioms expressing the properties that define each logic when we introduced the rules.

□

5.2.3 Normalization

The labeled natural deduction systems defined in this section present the same features of those of Section 4.2. In particular, we have restricted the introduction/elimination of the operators to the specific rules $\mathsf{G}I$, $\mathsf{G}E$, $\forall I$, $\forall E$. Moreover, as in Section 4.2, relational rules can be reduced to have only atomic conclusions. Here we omit an explicit analysis of normalization, however the nature of the system is such that an adaptation of the techniques used in similar labeled systems (see, e.g., [103, 148]) does not seem to be difficult.

In particular, we remark that a standard procedure, defined by induction on the complexity of the maximum formulas to be removed (see also Section 3.2 for a brief introduction to normalization in natural deduction), is expected to work in this case. A deeper proof-theoretical analysis will be performed in Section 5.3 in the case of a system for the logic $BCTL^*_-$, for which a more complex treatment will be required.

5.3 A System for $BCTL^*_-$

5.3.1 Introduction

In this section, we consider computation tree logics. One of the most popular of such logics is the so-called CTL^* (see Section 2.4.2), which has been shown to be especially useful in developing and checking the correctness of reactive systems (see, e.g., [70, 102]). In spite of its great relevance, the problem of presenting a satisfactory deduction system or even a Hilbert-style axiomatization for such a logic has been, partially, solved only recently in [135]. However, it is a non-standard automata-based axiomatization, which makes use of "an unusual and unorthodox rule of inference" (as stated by Reynolds himself in [139]).

The main difficulty encountered in finding a *finitary axiomatization* of CTL^* (and, in fact, such an axiomatization is still unknown, as discussed in, e.g., [135]) resides in the extreme difficulty to master the so-called *limit-closure property* of the standard CTL^* validity semantics.

For this reason, a number of interesting sublogics of CTL^* have been proposed in the literature. Amongst these logics, a special role is played by $BCTL^*$ [139]. The logic $BCTL^*$, is obtained by referring to a more general semantics than that of CTL^*, where we only require that the set of paths in a model is closed under taking suffixes (i.e. is *suffix-closed*) and is closed under putting together a finite prefix of one path with the suffix of any other path beginning at the same state where the prefix ends (i.e. is *fusion-closed*). In other words, this logic does not enjoy the limit-closure property (see Section 2.4.2 for details).

It is important to stress that $BCTL^*$ is not merely a kind of escape from CTL^*. It is also relevant in itself when we are interested in restricting the set of computations to be taken into consideration; namely, in the case of reasoning under fairness assumptions. In fact, as described in Section 2.4.2, it has been shown [42] that $BCTL^*$ is equivalent to the logic generated by fair structures, i.e. transition systems endowed with a mechanism for expressing conditions of *generalized fairness* [63].

In this section, we present a labeled natural deduction system $\mathcal{N}(BCTL^*_-)$ for the bundled computation tree logic $BCTL^*_-$, which is the until-free version of $BCTL^*$. In defining such a system, we adapt the ideas laying behind the formulation of systems for Ockhamist logics of Section 5.2 to the discrete case. Excluding until from the set of considered operators makes an analysis of normalization easier. We remark, however, that the solution proposed in Section 4.4 for the treatment of until is pretty general and thus could be easily adapted to the case of this section. With regard to possible extensions towards CTL^* (and in general towards

capturing a *full* semantics, also in the case of *OBTL*) some ideas will be sketched in Section 5.5. Part of the material of this section has been presented in [109].

The structure of this section is the following:

– in Section 5.3.2, we define a labeled version of $BCTL^*_-$ and specify its semantics;
– in Section 5.3.3, we present and briefly describe the rules of the natural deduction system;
– in Section 5.3.4, we prove that the system is sound with respect to the given semantics;
– in Section 5.3.5, we give a proof of weak completeness by using a given Hilbert-style axiomatization for the logic.

Normalization of the system will be treated in Section 5.4.

5.3.2 A labeled version of $BCTL^*_-$

It is not difficult to adapt the extension to a labeled version of the logic of basic frames provided in Section 5.2.1 to the case of $BCTL^*_-$. First, we need to add a further relational symbol: we will use $\lhd$, as in Chapter 4, to denote, in the syntax, the relation of immediate successor, upon which the operator X is defined.

In this section, the terms *labeled formula* and *relational formula* will correspond to the following notions. We also remark that, in order to give a more complete treatment of normalization (Section 5.4), in this case we consider also the conjunction $\wedge$ as a primitive connective.

Definition 5.7. *Let L be a denumerable set of labels, $<$ and $\bullet$ two binary relation symbols over L. If b and c are labels in L and A is a $BCTL^*_-$ formula, then $b \lhd c$, $b \leqslant c$ and $b \bullet c$ are* relational well-formed ($BCTL^*_-$) formulas *(or relational formulas, or rwffs for short) and $b : A$ is a* labeled well-formed ($BCTL^*_-$) formula *(or labeled formula, or lwff for short).*

If we reason in terms of transition frames, the intended meaning of an lwff $b : A$ is that:

- A holds in the initial state of b when A is a state formula, and that
- A holds in the path b when A is a path formula.

In the rwffs, we use $\lhd$, $\leqslant$ and $\bullet$ with the following intended meaning:

- $b_1 \leqslant b_2$ states that b_2 is a suffix of b_1, i.e. if $b_1 = s_1, s_2, ...$ then $b_2 = s_i, s_{i+1}, ...$ for some $i \geq 1$;
- $b_1 \lhd b_2$ states that b_2 is the maximal proper suffix of b_1, i.e. if $b_1 = s_1, s_2, s_3, ...$ then $b_2 = s_2, s_3, ...$;
- $b_1 \bullet b_2$ states that b_1 and b_2 share the same initial state, i.e. if $b_1 = s_1, s_2, s_3, ...$ and $b_2 = s'_1, s'_2, s'_3, ...$ then $s_1 = s'_1$.

For the sake of clarity, we define explicitly the notion of truth for labeled and relational formulas as follows. The notion of interpretation is adapted to the case of $BCTL^*_-$ from the standard one (Section 3.3.2) in the obvious way.

Definition 5.8. *Given an* $(\mathbb{N} \times \mathcal{W})$*-structure* $\mathcal{M} = (\mathcal{T}, \prec, \simeq, \mathcal{V})$*, where* $\mathcal{T} = (\mathbb{N} \times \mathcal{W})$ *for some set* $\mathcal{W}$*, and an interpretation* λ *on it,* truth *for a formula* φ *(relational or labeled) is the relation* $\models_{BCTL^*_-}$ *defined as follows:*

$\mathcal{M}, \lambda \not\models_{BCTL^*_-} b : \bot$;

$\mathcal{M}, \lambda \models_{BCTL^*_-} b_1 \lhd b_2$ *iff* *there exist* $n \in \mathbb{N}$ *and* $w \in \mathcal{W}$ *such that* $\lambda(b_1) = (n, w)$ *and* $\lambda(b_2) = (n+1, w)$;

$\mathcal{M}, \lambda \models_{BCTL^*_-} b_1 \leqslant b_2$ *iff* $\lambda(b_1) = \lambda(b_2)$ *or* $\lambda(b_1) \prec \lambda(b_2)$;

$\mathcal{M}, \lambda \models_{BCTL^*_-} b_1 \bullet b_2$ *iff* $\lambda(b_1) \simeq \lambda(b_2)$;

$\mathcal{M}, \lambda \models_{BCTL^*_-} b : p$ *iff* $p \in \mathcal{V}(\lambda(b))$;

$\mathcal{M}, \lambda \models_{BCTL^*_-} b : A \supset B$ *iff* $\mathcal{M}, \lambda \models_{BCTL^*_-} b : A$ *implies* $\mathcal{M}, \lambda \models_{BCTL^*_-} b : B$;

$\mathcal{M}, \lambda \models_{BCTL^*_-} b : A \wedge B$ *iff* $\mathcal{M}, \lambda \models_{BCTL^*_-} b : A$ *and* $\mathcal{M}, \lambda \models_{BCTL^*_-} b : B$;

$\mathcal{M}, \lambda \models_{BCTL^*_-} b : \mathsf{X}A$ *iff* *for all* b', $\mathcal{M}, \lambda \models_{BCTL^*_-} b \lhd b'$ *implies* $\mathcal{M}, \lambda \models_{BCTL^*_-} b' : A$;

$\mathcal{M}, \lambda \models_{BCTL^*_-} b : \mathsf{G}A$ *iff* *for all* b', $\mathcal{M}, \lambda \models_{BCTL^*_-} b \leqslant b'$ *implies* $\mathcal{M}, \lambda \models_{BCTL^*_-} b' : A$;

$\mathcal{M}, \lambda \models_{BCTL^*_-} b : \forall A$ *iff* *for all* b', $\mathcal{M}, \lambda \models_{BCTL^*_-} b \bullet b'$ *implies* $\mathcal{M}, \lambda \models_{BCTL^*_-} b' : A$.

When $\mathcal{M}, \lambda \models_{BCTL^*_-} \varphi$*, we say that* φ *is* true *in* $\mathcal{M}$ *according to* λ*. By extension:*

$\mathcal{M}, \lambda \models_{BCTL^*_-} \Gamma$ *iff* $\mathcal{M}, \lambda \models_{BCTL^*_-} \varphi$ *for all* $\varphi \in \Gamma$;

$\mathcal{M} \models_{BCTL^*_-} \varphi$ *iff* *for every interpretation* λ, $\mathcal{M}, \lambda \models_{BCTL^*_-} \varphi$;

$\mathcal{M} \models_{BCTL^*_-} \Gamma$ *iff* *for every interpretation* λ, $\mathcal{M}, \lambda \models_{BCTL^*_-} \Gamma$;

$\Gamma \models_{BCTL^*_-} \varphi$ *iff* *for every* $(\mathbb{N} \times \mathcal{W})$*-structure* $\mathcal{M}$ *and interpretation* λ, $\mathcal{M}, \lambda \models_{BCTL^*_-} \Gamma$ *implies* $\mathcal{M}, \lambda \models_{BCTL^*_-} \varphi$.

5.3.3 The System $\mathcal{N}(BCTL^*_-)$

In this section, we give a labeled natural deduction system, which we call $\mathcal{N}(BCTL^*_-)$, for the logic $BCTL^*_-$.

The rules of $\mathcal{N}(BCTL^*_-)$ are given in Fig. 5.4. The notion of derivability in $\mathcal{N}(BCTL^*_-)$ (denoted $\vdash_{\mathcal{N}(BCTL^*_-)}$) can be defined in the usual way (see Section 3.2). Rules for logical connectives and for temporal operators are in the same vein of those of the systems already presented (Chapter 4 and Section 5.2). Here we briefly describe the other rules, trying to clarify also their interpretation in terms of paths in a transition system.

Rules for $\lhd$

The rule *ser*$\lhd$ models the fact that every world has an immediate successor and thus ensures that the suffix-closure property (as described in Section 2.4.2) is satisfied. The rule *lin*$\lhd$ specifies that such a successor must be unique.

$$\frac{\begin{array}{c}[b_1 : A \supset \bot] \\ \vdots \\ b_2 : \bot\end{array}}{b_1 : A}\ \bot E \qquad \frac{\begin{array}{c}[b : A] \\ \vdots \\ b : B\end{array}}{b : A \supset B}\ \supset I \qquad \frac{b : A \supset B \quad b : A}{b : B}\ \supset E$$

$$\frac{b : A \quad b : B}{b : A \wedge B}\ \wedge I \qquad \frac{b : A \wedge B}{b : A}\ \wedge E_1 \qquad \frac{b : A \wedge B}{b : B}\ \wedge E_2$$

$$\frac{\begin{array}{c}[b_1 \lhd b_2] \\ \vdots \\ b_2 : A\end{array}}{b_1 : \mathsf{X}A}\ \mathsf{X}I \qquad \frac{b_1 : \mathsf{X}A \quad b_1 \lhd b_2}{b_2 : A}\ \mathsf{X}E \qquad \frac{\begin{array}{c}[b_1 \lhd b_2] \\ \vdots \\ b : A\end{array}}{b : A}\ ser\lhd \qquad \frac{b_1 \lhd b_2 \quad b_1 \lhd b_3 \quad b_2 : A}{b_3 : A}\ lin\lhd$$

$$\frac{\begin{array}{c}[b_1 \leqslant b_2] \\ \vdots \\ b_2 : A\end{array}}{b_1 : \mathsf{G}A}\ \mathsf{G}I \qquad \frac{b_1 : \mathsf{G}A \quad b_1 \leqslant b_2}{b_2 : A}\ \mathsf{G}E \qquad \frac{\begin{array}{c}[b_1 \leqslant b_1] \\ \vdots \\ b : A\end{array}}{b : A}\ refl\leqslant \qquad \frac{b_1 \leqslant b_2 \quad b_2 \leqslant b_3 \quad \begin{array}{c}[b_1 \leqslant b_3] \\ \vdots \\ b : A\end{array}}{b : A}\ trans\leqslant$$

$$\frac{\begin{array}{c}[b_1 \bullet b_2] \\ \vdots \\ b_2 : A\end{array}}{b_1 : \forall A}\ \forall I \qquad \frac{b_1 : \forall A \quad b_1 \bullet b_2}{b_2 : A}\ \forall E \qquad \frac{\begin{array}{c}[b_1 \bullet b_1] \\ \vdots \\ b : A\end{array}}{b : A}\ refl\bullet \qquad \frac{b_1 \bullet b_2 \quad \begin{array}{c}[b_2 \bullet b_1] \\ \vdots \\ b : A\end{array}}{b : A}\ symm\bullet$$

$$\frac{b_1 \bullet b_2 \quad b_2 \bullet b_3 \quad \begin{array}{c}[b_1 \bullet b_3] \\ \vdots \\ b : A\end{array}}{b : A}\ trans\bullet \qquad \frac{b_1 : p \quad b_1 \bullet b_2}{b_2 : p}\ atom\bullet \qquad \frac{b_1 \lhd b_2 \quad \begin{array}{c}[b_1 \leqslant b_2] \\ \vdots \\ b : A\end{array}}{b : A}\ base\leqslant$$

$$\frac{b_1 \lhd b_2 \quad b_2 \bullet b_3 \quad \begin{array}{c}[b' \bullet b_1] \quad [b' \lhd b_3] \\ \vdots \\ b : A\end{array}}{b : A}\ fusion \qquad \frac{b_0 : A \quad b_0 \leqslant b \quad \begin{array}{c}[b_0 \leqslant b_i] \quad [b_i \lhd b_j] \quad [b_i : A] \\ \vdots \\ b_j : A\end{array}}{b : A}\ ind$$

- In $\mathsf{X}I$ (respectively $\mathsf{G}I$, $\forall I$), b_2 is *fresh*, i.e. it is different from b_1 and does not occur in any assumption on which $b_2 : A$ depends other than the discharged assumption $b_1 \lhd b_2$ (respectively $b_1 \leqslant b_2$, $b_1 \bullet b_2$).
- In $ser\lhd$, b_2 is fresh, i.e. it is different from b and does not occur in any assumption on which $b : A$ depends other than the discharged assumption $b_1 \lhd b_2$.
- In $atom\bullet$, p is an atomic proposition.
- In *fusion*, b' is fresh, i.e. it is different from b, b_1, b_2 and b_3, and does not occur in any assumption on which $b : A$ depends other than the discharged assumptions $b' \bullet b_1$ and $b' \lhd b_3$.
- In *ind*, b_i and b_j are fresh, i.e. they are different from each other and from b and b_0, and do not occur in any assumption on which $b_j : A$ depends other than the discharged assumptions of the rule.

Fig. 5.4. The rules of $\mathcal{N}(BCTL^*_-)$.

Rules for $\leqslant$

We recall that $b_1 \leqslant b_2$ intuitively means that b_2 is a suffix of b_1. In terms of the given semantics, $\leqslant$ denotes in the syntax the reflexive and transitive closure of $\prec$ (see Definition 2.28). The rules *refl* $\leqslant$ and *trans* $\leqslant$ state respectively the reflexivity and transitivity of $\leqslant$.

Rules for $\bullet$

We recall from Section 5.3.2 that the symbol $\bullet$ in the syntax corresponds to the accessibility relation $\simeq$ in the semantics. $\simeq$ is defined as an equivalence relation and thus we have the rules *refl*$\bullet$, *symm*$\bullet$ and *trans*$\bullet$ that express reflexivity, symmetry and transitivity of $\bullet$, respectively. It follows that $\forall$ behaves as the modal operator $\Box$ does in the modal logic *S5*.

Finally, *atom*$\bullet$ mirrors the property of $(\mathbb{N} \times \mathcal{W})$-structures according to which if $x \simeq y$ then $\mathcal{V}(x) = \mathcal{V}(y)$ (see Definition 2.30). Intuitively, with regard to transition structures, it models the idea that two paths having the same initial state must satisfy the same set of atomic propositions and is the equivalent of the axiom (*Atom*) in the axiomatization $\mathcal{H}(BCTL^*_-)$ given in Section 2.4.2.

Rules for Interactions between the Relations

The rule *base* $\leqslant$ expresses the fact that the relation corresponding to $\leqslant$ contains the relation corresponding to $\lhd$: in the "path terminology", it says that every path b is a prefix of its maximal proper suffix.

The rule *fusion* strictly corresponds to the *fusion-closure* property (see Section 2.4.2) of transition systems, according to which the set of paths must be closed under putting together a finite prefix of one path with the suffix of any other path such that the prefix ends at the same state as the suffix begins. In terms of the given semantics, it roughly corresponds to condition 4(b) in the definition of an Ockhamist frame (Definition 2.28). In terms of the axiomatization $\mathcal{H}(BCTL^*_-)$, it is the equivalent of the axiom (*Fusion*).

Finally, we have a rule *ind* modeling the induction principle underlying the relation between $\lhd$ and $\leqslant$. It comes from the definition of $(\mathbb{N} \times \mathcal{W})$-frame (Definition 2.29), which requires the vertical lines of points to be isomorphic to the natural numbers.

5.3.4 Soundness

Theorem 5.9. *For every set* Γ *of labeled and relational formulas and every labeled formula* $b : A$*, it holds that*

$$\Gamma \vdash_{\mathcal{N}(BCTL^*_-)} b : A \quad \Rightarrow \quad \Gamma \models_{BCTL^*_-} b : A \,.$$

Proof. The proof proceeds by induction on the structure of the derivation of $b : A$. The base case is when $b : A \in \Gamma$ and is trivial. There is one step case for every rule: most of them can be treated in a way similar to the analogous rules given for the other systems; we show only five representative cases.

Consider an application of the rule $\mathsf{X}I$:

$$\dfrac{\begin{array}{c}[b \lhd b'] \\ \Pi \\ b' : A\end{array}}{b : \mathsf{X}A}\ \mathsf{X}I$$

where Π is a proof of $b' : A$ from hypotheses in Γ', with b' fresh and with $\Gamma' = \Gamma \cup \{b \lhd b'\}$. By the induction hypothesis, for all interpretations λ, if $\mathcal{M}, \lambda \models_{BCTL^*_{\_}} \Gamma'$ then $\mathcal{M}, \lambda \models_{BCTL^*_{\_}} b' : A$. We let λ be any interpretation such that $\mathcal{M}, \lambda \models_{BCTL^*_{\_}} \Gamma$, and show that $\mathcal{M}, \lambda \models_{BCTL^*_{\_}} b : \mathsf{X}A$. Let (n, w) be any point such that $\lambda(b) = (n, w)$. Since λ can be trivially extended to another interpretation (still called λ for simplicity) by setting $\lambda(b') = (n + 1, w)$, the induction hypothesis yields $\mathcal{M}, \lambda \models_{BCTL^*_{\_}} b' : A$, i.e. $\mathcal{M}, (n + 1, w) \models_{BCTL^*_{\_}} A$, and thus $\mathcal{M}, \lambda \models_{BCTL^*_{\_}} b : \mathsf{X}A$.

Consider an application of the rule $\forall I$:

$$\dfrac{\begin{array}{c}[b \bullet b'] \\ \Pi \\ b' : A\end{array}}{b : \forall A}\ \forall I$$

where Π is a proof of $b' : A$ from hypotheses in Γ', with b' fresh and with $\Gamma' = \Gamma \cup \{b \bullet b'\}$. By the induction hypothesis, for all interpretations λ, if $\mathcal{M}, \lambda \models_{BCTL^*_{\_}} \Gamma'$ then $\mathcal{M}, \lambda \models_{BCTL^*_{\_}} b' : A$. We let λ be any interpretation such that $\mathcal{M}, \lambda \models_{BCTL^*_{\_}} \Gamma$, and show that $\mathcal{M}, \lambda \models_{BCTL^*_{\_}} b : \forall A$. Let (n, w) be any point such that $\lambda(b) = (n, w)$. Now let us consider an arbitrary point (n, w') for some w'. Since λ can be trivially extended to another interpretation (still called λ for simplicity) by setting $\lambda(b') = (n, w')$, the induction hypothesis yields $\mathcal{M}, \lambda \models_{BCTL^*_{\_}} b' : A$, i.e. $\mathcal{M}, (n, w') \models_{BCTL^*_{\_}} A$. Given that w' is arbitrary we can conclude $\mathcal{M}, \lambda \models_{BCTL^*_{\_}} b : \forall A$.

Consider the case in which the last rule applied is $\mathsf{G}E$:

$$\dfrac{\begin{array}{c}\Pi \\ b' : \mathsf{G}A\end{array} \quad b' \leqslant b}{b : A}\ \mathsf{G}E$$

where Π is a proof of $b' : \mathsf{G}A$ from hypotheses in Γ_1, with $\Gamma = \Gamma_1 \cup \{b' \leqslant b\}$ for some set Γ_1 of formulas. By applying the induction hypothesis on Π, we have:

$$\Gamma_1 \models_{BCTL^*_{\_}} b' : \mathsf{G}A\ .$$

We proceed by considering a generic $(\mathbb{N} \times \mathcal{W})$-structure $\mathcal{M} = (\mathcal{T}, \prec, \simeq, \mathcal{V})$ and a generic interpretation λ on it such that $\mathcal{M}, \lambda \models_{BCTL^*_{\_}} \Gamma$ and showing that this entails

$$\mathcal{M}, \lambda \models_{BCTL^*_{\_}} b : A\ .$$

Since $\Gamma_1 \subset \Gamma$, from the induction hypothesis we deduce $\mathcal{M}, \lambda \models_{BCTL^*_{\_}} b' : \mathsf{G}A$. Furthermore $\mathcal{M}, \lambda \models_{BCTL^*_{\_}} \Gamma$ entails $\mathcal{M}, \lambda \models_{BCTL^*_{\_}} b' \leqslant b$. Then, by Definition 2.32, we obtain $\mathcal{M}, \lambda \models_{BCTL^*_{\_}} b : A$.

Let an application of *fusion* be the last rule application in the derivation of $b : A$:

$$\dfrac{\begin{array}{cc} & \begin{array}{cc}[b' \bullet b_1] & [b' \lhd b_3]\end{array} \\ b_1 \lhd b_2 \quad b_2 \bullet b_3 & \begin{array}{c}\Pi \\ b : A\end{array}\end{array}}{b : A}\ \mathit{fusion}$$

where Π is a proof of $b : A$ from hypotheses in Γ_2, with $\Gamma = \Gamma_1 \cup \{b_1 \lhd b_2\} \cup \{b_2 \bullet b_3\}$ and $\Gamma_2 = \Gamma_1 \cup \{b' \bullet b_1\} \cup \{b' \lhd b_3\}$ for some set Γ_1 of formulas. The side-condition ensures that b' is fresh in Π. Hence, by applying the induction hypothesis on Π, we have

$$\Gamma_2 \models_{BCTL^*_-} b : A\,.$$

We proceed by considering a generic $(\mathbb{N} \times \mathcal{W})$-structure $\mathcal{M} = (\mathcal{T}, \prec, \simeq, \mathcal{V})$ and a generic interpretation λ on it such that $\mathcal{M}, \lambda \models_{BCTL^*_-} \Gamma$ and showing that this entails

$$\mathcal{M}, \lambda \models_{BCTL^*_-} b : A\,.$$

From $\mathcal{M}, \lambda \models_{BCTL^*_-} \Gamma$, we deduce:

(*i*) there exists a point $(n, w) \in \mathcal{T}$ such that $\lambda(b_1) = (n, w)$ and $\lambda(b_2) = (n + 1, w)$;
(*ii*) $\lambda(b_2) \simeq \lambda(b_3)$.

We know from Lemma 2.31 that $\lambda(b_3) = (n + 1, v)$ for some $(n + 1, v) \in \mathcal{T}$. Then by the property 4(*b*) of Ockhamist frames (Definition 2.28), the point (n, v) is such that $(n, v) \simeq (n, w) = \lambda(b_1)$. Now let us consider an interpretation λ' which differs from λ only for the point assigned to b', namely $\lambda' = \lambda[b' \mapsto (n, v)]$. Note that we have defined λ' in a way such that $\mathcal{M}, \lambda' \models_{BCTL^*_-} b' \bullet b_1$ and $\mathcal{M}, \lambda' \models_{BCTL^*_-} b' \lhd b_3$. Since b' does not occur in Γ (by the side-condition on the application of *fusion*), we have $\mathcal{M}, \lambda' \models_{BCTL^*_-} \Gamma_1$ and thus also $\mathcal{M}, \lambda' \models_{BCTL^*_-} \Gamma_2$. Then, by the induction hypothesis, $\mathcal{M}, \lambda' \models_{BCTL^*_-} b : A$. We conclude $\mathcal{M}, \lambda \models_{BCTL^*_-} b : A$ by observing that the side-condition $b' \neq b$ ensures $\lambda(b) = \lambda'(b)$.

Finally, consider the case in which the last rule applied is *ind*:

$$\dfrac{\begin{array}{ccc} & \begin{array}{ccc}[b_0 \leqslant b_i] & [b_i \lhd b_j] & [b_i : A]\end{array} \\ \begin{array}{c}\Pi' \\ b_0 : A\end{array} \quad b_0 \leqslant b & & \begin{array}{c}\Pi \\ b_j : A\end{array}\end{array}}{b : A}\ \mathit{ind}$$

where Π is a proof of $b_j : A$ from hypotheses in Γ_2 and Π' is a proof of $b_0 : A$ from hypotheses in Γ_1, with $\Gamma = \Gamma_1 \cup \{b_0 \leqslant b\}$ and $\Gamma_2 = \Gamma_1 \cup \{b_0 \leqslant b_i\} \cup \{b_i \lhd b_j\} \cup \{b_i : A\}$ for some set Γ_1 of formulas. The side-condition on *ind* ensures that b_i and b_j are fresh in Π. Hence, by applying the induction hypothesis on Π and Π', we have:

$$\Gamma_2 \models_{BCTL^*_-} b_j : A \qquad \text{and} \qquad \Gamma_1 \models_{BCTL^*_-} b_0 : A\,.$$

We proceed by considering a generic $(\mathbb{N} \times \mathcal{W})$-structure $\mathcal{M} = (\mathcal{T}, \prec, \simeq, \mathcal{V})$ and a generic interpretation λ on it such that $\mathcal{M}, \lambda \models_{BCTL^*_-} \Gamma$ and showing that this entails

$$\mathcal{M}, \lambda \models_{BCTL^*_-} b : A \,.$$

First, we note that $\Gamma_1 \subset \Gamma$ and therefore $\mathcal{M}, \lambda \models_{BCTL^*_-} \Gamma$ implies $\mathcal{M}, \lambda \models_{BCTL^*_-} \Gamma_1$ and, by the induction hypothesis on Π', $\mathcal{M}, \lambda \models_{BCTL^*_-} b_0 : A$. Let $\lambda(b_0) = (n, w)$ for some $(n, w) \in \mathcal{T}$. From $\mathcal{M}, \lambda \models_{BCTL^*_-} \Gamma$, we deduce $\mathcal{M}, \lambda \models_{BCTL^*_-} b_0 \leqslant b$ and thus $\lambda(b) = (n + k, w)$ for some $k \in \mathbb{N}$. We show by induction on k that $\mathcal{M}, \lambda \models_{BCTL^*_-} b : A$. As a base case, we have $k = 0$; it follows that $\lambda(b) = \lambda(b_0)$ and thus trivially that $\mathcal{M}, \lambda \models_{BCTL^*_-} b_0 : A$ entails $\mathcal{M}, \lambda \models_{BCTL^*_-} b : A$. Let us consider now the induction step. Given a label b_{k-1} such that $\lambda(b_{k-1}) = (n + k - 1, w)$, we show that the induction hypothesis $\mathcal{M}, \lambda \models_{BCTL^*_-} b_{k-1} : A$ entails the thesis $\mathcal{M}, \lambda \models_{BCTL^*_-} b : A$. We can build an interpretation λ' that differs from λ only in the points assigned to b_i and b_j, namely $\lambda' = \lambda[b_i \mapsto (n+k-1, w)][b_j \mapsto (n+k, w)]$. It is easy to verify that the interpretation λ' is such that the following three conditions hold:

(*i*) $\mathcal{M}, \lambda' \models_{BCTL^*_-} b_i : A$;
(*ii*) $\mathcal{M}, \lambda' \models_{BCTL^*_-} b_0 \leqslant b_i$;
(*iii*) $\mathcal{M}, \lambda' \models_{BCTL^*_-} b_i \lhd b_j$.

Furthermore, the side-condition on the rule *ind* ensures that λ and λ' agree on all the labels occurring in Γ_1, from which we can infer that also $\mathcal{M}, \lambda' \models_{BCTL^*_-} \Gamma_1$ must hold. It follows that $\mathcal{M}, \lambda' \models_{BCTL^*_-} \Gamma_2$ and thus, by the induction hypothesis on Π, that $\mathcal{M}, \lambda' \models_{BCTL^*_-} b_j : A$. We conclude $\mathcal{M}, \lambda \models_{BCTL^*_-} b : A$ by observing that $\lambda'(b_j) = \lambda(b)$.

□

5.3.5 Completeness

The proposed natural deduction system $\mathcal{N}(BCTL^*_-)$ consists of only finitary rules; consequently, it cannot be strongly complete (see the discussion on the failure of compactness in Section 2.3.4). Nevertheless, our system $\mathcal{N}(BCTL^*_-)$ is weakly complete with respect to $BCTL^*_-$, namely:

Theorem 5.10. *For every labeled formula $b : A$ it holds:*

$$\models_{BCTL^*_-} b : A \quad \Rightarrow \quad \vdash_{\mathcal{N}(BCTL^*_-)} b : A \,.$$

Proof. The most "economic" way to prove the theorem is to show that $\mathcal{N}(BCTL^*_-)$ is complete with respect to the axiomatization $\mathcal{H}(BCTL^*_-)$ given in Section 2.4.2, which is sound and complete for the logic $BCTL^*_-$. Most of the axioms can be proved in a way analogous to the proof for $\mathcal{N}(LTL_-)$ (for the linear part) or $\mathcal{N}(bas)$ (for the branching part). Here we just show a derivation for the axiom (*Fusion*).

$$\dfrac{\dfrac{\dfrac{\dfrac{[b : \forall \mathsf{X} A]^1 \quad [b \bullet b']^4}{b' : \mathsf{X} A}\,\forall E \qquad \dfrac{[b' \lhd d]^4}{d : A}\,\mathsf{X}E}{d : A}\,\mathit{fusion}^4}{\dfrac{d : A}{c : \forall A}\,\forall I^3}}{\dfrac{c : \forall A}{b : \mathsf{X}\forall A}\,\mathsf{X}I^2}}{b : \forall \mathsf{X} A \supset \mathsf{X}\forall A}\,\supset I^1 \qquad [b \lhd c]^2 \quad [c \bullet d]^3$$

□

5.4 Normalization of the system for $BCTL^*_-$

In this section, we describe a process of normalization for an intuitionistic version of the system $\mathcal{N}(BCTL^*_-)$.

In studying normalization for $\mathcal{N}(BCTL^*_-)$, the main difficulties arise from the presence of a rule (*ind*) modeling the induction principle. This suggests an analogy with deduction systems for Peano Arithmetic. Though it can be proved that a standard subformula property does not hold for such systems, it is possible to consider forms of normalization that are "good" enough to get a syntactic proof of consistency.

There exist several translations from classical Peano arithmetic to intuitionistic Heyting arithmetic such that if a contradiction is provable in the former, then it is also provable in the latter. This implies that the consistency of intuitionistic arithmetic guarantees also the consistency of classical arithmetic. Thus, in order to simplify the treatment of normalization, it is quite standard to focus on intuitionistic versions of the system.

We will follow the same way of reasoning here. The structure of the proof mirrors those given in [74, 126, 151] with regard to natural deduction systems for Heyting arithmetic. Some adaptations required by the labeled nature of our system are inspired by normalization presented for labeled systems in [148, 159].

We will obtain a result of consistency for the intuitionistic version of our system. Moreover, we will extend it, via a proper translation, to a result of consistency for the classical system $\mathcal{N}(BCTL^*_-)$ presented in Section 5.3.3.

The structure of this section is the following:

- in Section 5.4.1, we introduce the intuitionistic version $\mathcal{N}(BCTL^*_{-i})$ of the system, for which normalization will be studied;
- in Section 5.4.2, we define a normal form for $\mathcal{N}(BCTL^*_{-i})$ derivations, where the idea is that we do not have detours and allow only some forms of *ind*-applications;
- in Section 5.4.3, we define a reduction relation $\Rightarrow$ between derivations, where, in addition to standard contractions, we consider some contractions for *ind*-applications;
- in Section 5.4.4, we prove that $\mathcal{N}(BCTL^*_{-i})$ derivations enjoy the Church-Rosser property with regard to the relation $\Rightarrow$, i.e. if a derivation Π is such that $\Pi \Rightarrow \Pi'$ and $\Pi \Rightarrow \Pi''$ then there exists a derivation Π''' such that $\Pi' \Rightarrow \Pi'''$ and $\Pi'' \Rightarrow \Pi'''$;

– in Section 5.4.5, we exploit the Church-Rosser property to prove a theorem of weak normalization for the system $\mathcal{N}(BCTL^*_{-i})$, i.e. we show that every derivation reduces (by $\Rightarrow$) to a derivation in normal form; the proof is not by induction on the complexity of the maximum formulas to be removed but follows the schema of normalization procedures of natural deduction systems for Heyting arithmetics (in particular, [74]);
– in Section 5.4.6, we analyze the structure of normal derivations;
– in Section 5.4.7, we use the structural properties of normal derivations to give a (syntactic) proof of the consistency of $\mathcal{N}(BCTL^*_{-i})$ and indirectly, by using a translation from the classical to the intuitionistic version of the logic, of $\mathcal{N}(BCTL^*_{-})$;
– finally, in Section 5.4.8, we show that the system $\mathcal{N}(BCTL^*_{-i})$, with respect to the normalization defined, does not enjoy the subformula property.

In order to ease readability, some of the proofs are given in Appendix A.
The content of this section has been submitted in [108].

5.4.1 The intuitionistic system $\mathcal{N}(BCTL^*_{-i})$

Here we define the intuitionistic system $\mathcal{N}(BCTL^*_{-i})$ for which we will study normalization. First, we show that some conditions hold on the use of labels; then we introduce some modifications to the rules of $\mathcal{N}(BCTL^*_{-})$ in order to get an intuitionistic version of the system and some restrictions in order to simplify the normalization procedure; finally, we present a translation from the classical $(BCTL^*_{-})$ into the intuitionistic $(BCTL^*_{-i})$ version of the logic, which will be used to extend to $\mathcal{N}(BCTL^*_{-})$ the result of consistency proved for $\mathcal{N}(BCTL^*_{-i})$.

In order to carry out the process of normalization described in the following, we need to note that some conditions on variables hold in the system $\mathcal{N}(BCTL^*_{-})$ (and also in its intuitionistic version). In particular, we adapt the standard definition of proper parameter from [125, 153] and prove a lemma on parameters.

Definition 5.11. *A label b is said to be the* proper parameter *of an application r of $\mathsf{X}I$, $\mathsf{G}I$, $\forall I$, $ser\lhd$, fusion or ind if b is the label that is required to be fresh in the dischargeable assumption of r. A label b is said to be a proper parameter in a derivation Π if it is the proper parameter of some rule application in Π.*

Lemma 5.12. *If $b : A$ is derivable, then there exists a derivation Π of $b : A$ from Γ where:*

(i) each proper parameter of Π is a proper parameter of a single rule application;
(ii) the proper parameter of an application r of $\mathsf{X}I$, $\mathsf{G}I$ or $\forall I$ occurs only above the conclusion of r.
(iii) the proper parameter of an application r of $ser\lhd$, fusion or ind occurs only above one of the premises of r.

Proof. By induction on Π, by systematically renaming proper parameters starting with the uppermost applications of $\mathsf{X}I$, $\mathsf{G}I$, $\forall I$, $ser\lhd$, *fusion* and *ind*.

□

In order to simplify the analysis, we modify the system $\mathcal{N}(BCTL^*_-)$ by adding a rule $lin{\lhd}_{\mathcal{R}}$ that exploits linearity of $\lhd$ also in the context of rwffs:

$$\dfrac{\begin{array}{cccc} b_1 \lhd b_2 & b_1 \lhd b_3 & \rho & \begin{array}{c}\rho[b_3/b_2]\\ \vdots \\ b : A\end{array}\end{array}}{b : A}\ lin{\lhd}_{\mathcal{R}} \quad ,$$

where ρ is an rwff. The system obtained is equivalent, with respect to the set of derivable formulas, to the previous one, as shown in the following lemma.

Lemma 5.13. *The system $\mathcal{N}(BCTL^*_-)$ with the addition of the rule $lin{\lhd}_{\mathcal{R}}$ is sound with respect to the semantics of $BCTL^*_-$.*

Proof. We show that for every set Γ of labeled and relational formulas and every labeled formula $b : A$, if $b : A$ is derivable in the system by using assumptions in Γ then $\Gamma \models_{_{BCTL^*_-}} b : A$ holds. The proof is by induction on the structure of the derivations: with respect to Theorem 5.9, we have to consider just one additional case. Let an application of $lin{\lhd}_{\mathcal{R}}$ be the last rule application in the derivation of $b : A$:

$$\dfrac{\begin{array}{cccc} b_1 \lhd b_2 & b_1 \lhd b_3 & \rho & \begin{array}{c}\rho[b_3/b_2]\\ \Pi \\ b : A\end{array}\end{array}}{b : A}\ lin{\lhd}_{\mathcal{R}} \quad ,$$

where Π is a proof of $b : A$ from hypotheses in Γ_2, with $\Gamma = \Gamma_1 \cup \{b_1 \lhd b_2\} \cup \{b_1 \lhd b_3\} \cup \{\rho\}$ and $\Gamma_2 = \Gamma_1 \cup \{\rho[b_3/b_2]\}$ for some set Γ_1 of formulas. By applying the induction hypothesis on Π, we have

$$\Gamma_2 \models_{_{BCTL^*_-}} b : A\ .$$

We proceed by considering a generic $(\mathbb{N} \times \mathcal{W})$-structure $\mathcal{M} = (\mathcal{T}, \prec, \simeq, \mathcal{V})$ and a generic interpretation λ on it such that $\mathcal{M}, \lambda \models_{_{BCTL^*_-}} \Gamma$ and showing that this entails

$$\mathcal{M}, \lambda \models_{_{BCTL^*_-}} b : A\ .$$

From $\mathcal{M}, \lambda \models_{_{BCTL^*_-}} \Gamma$, we deduce:

(*i*) there exists a point $(n, w) \in \mathcal{T}$ such that $\lambda(b_1) = (n, w)$ and $\lambda(b_2) = (n + 1, w)$;
(*ii*) there exists a point $(m, v) \in \mathcal{T}$ such that $\lambda(b_1) = (m, v)$ and $\lambda(b_3) = (m + 1, v)$.

Since λ is a function, (n, w) and (m, v) must coincide. It follows that also $\lambda(b_2)$ and $\lambda(b_3)$ coincide. But then, from $\mathcal{M}, \lambda \models_{_{BCTL^*_-}} \rho$, we deduce that also $\mathcal{M}, \lambda \models_{_{BCTL^*_-}} \rho[b_3/b_2]$ holds, whatever ρ is.

□

In the following, we call *relational rules* the rules *ser*$\lhd$, *lin*$\lhd$, *lin*$\lhd_{\mathcal{R}}$, *base* $\leqslant$, *refl* $\leqslant$, *trans* $\leqslant$, *refl*$\bullet$, *symm*$\bullet$, *trans*$\bullet$, *atom*$\bullet$ and *fusion*.

An intuitionistic version $\mathcal{N}(BCTL^*_{-i})$ of the system is now obtained by substituting the rule $\bot E$ with its intuitionistic version $\bot E_i$:

$$\frac{b_2 : \bot}{b_1 : A}\ \bot E_i \quad .$$

We can also restrict relational rules and $\bot E_i$ so that they have only atomic conclusions.

Lemma 5.14. *If $\Gamma \vdash_{\mathcal{N}(BCTL^*_{-i})} b : A$, then there exists a derivation in $\mathcal{N}(BCTL^*_{-i})$ of $b : A$ from Γ where all the applications of relational rules and of $\bot E_i$ have an atomic conclusion.*

Proof. We can give rules that systematically reduce the complexity of the formula that is the conclusion of the application. As an example, we show the reductions for the rule *base* $\leqslant$, when the main connective of the conclusion is $\supset$ or G, respectively:

$$\frac{\begin{array}{c}[b_1 \leqslant b_2]^1 \\ \Pi \\ b_1 \lhd b_2 \quad b : A \supset B\end{array}}{b : A \supset B}\ \textit{base}\ \leqslant^1 \quad \rightsquigarrow \quad \frac{\begin{array}{c}[b_1 \leqslant b_2]^2 \\ \Pi \\ b_1 \lhd b_2 \quad \dfrac{b : A \supset B \quad [b : A]^1}{b : B}\ \supset E\end{array}}{\dfrac{b : B}{b : A \supset B}\ \supset I^1}\ \textit{base}\ \leqslant^2 \quad ,$$

$$\frac{\begin{array}{c}[b_1 \leqslant b_2]^1 \\ \Pi \\ b_1 \lhd b_2 \quad b : \mathsf{G}A\end{array}}{b : \mathsf{G}A}\ \textit{base}\ \leqslant^1 \quad \rightsquigarrow \quad \frac{\begin{array}{c}[b_1 \leqslant b_2]^2 \\ \Pi \\ b_1 \lhd b_2 \quad \dfrac{b : \mathsf{G}A \quad [b \leqslant b']^1}{b' : A}\ \mathsf{G}E\end{array}}{\dfrac{b' : A}{b : \mathsf{G}A}\ \mathsf{G}I^1}\ \textit{base}\ \leqslant^2 \quad .$$

The reductions for other relational rules are very similar. We show instead reductions for $\bot E_i$:

$$\frac{\begin{array}{c}\Pi \\ b_1 : \bot\end{array}}{b : A \supset B}\ \bot E_i \quad \rightsquigarrow \quad \frac{\begin{array}{c}[b : A]^1 \\ \Pi \\ b_1 : \bot\end{array}}{b : B}\ \bot E_i}{b : A \supset B}\ \supset I^1 \quad ,$$

$$\frac{\begin{array}{c}\Pi \\ b_1 : \bot\end{array}}{b : \mathsf{G}A}\ \bot E_i \quad \rightsquigarrow \quad \frac{\begin{array}{c}[b \leqslant b_2]^1 \\ \Pi \\ b_1 : \bot\end{array}}{b_2 : A}\ \bot E_i}{b : \mathsf{G}A}\ \mathsf{G}I^1 \quad .$$

□

Thus we can consider a system where the application of relational rules and of $\bot E_i$ is restricted to have atomic conclusions. For simplicity, we will keep calling this system $\mathcal{N}(BCTL^*_{-i})$.

By summing up, in the following we will study normalization for the system obtained by modifying $\mathcal{N}(BCTL^*_{-})$ as specified by the following definition.

Definition 5.15. *The system $\mathcal{N}(BCTL^*_{-i})$ is obtained by modifying the system $\mathcal{N}(BCTL^*_-)$, presented in Fig. 5.4, as follows:*

(i) we add the rule $lin{\lhd_{\mathcal{R}}}$:

$$\frac{b_1 \lhd b_2 \quad b_1 \lhd b_3 \quad \rho \quad \begin{array}{c}\rho[b_3/b_2]\\ \vdots \\ b : A\end{array}}{b : A}\ lin{\lhd_{\mathcal{R}}} \quad ,$$

where ρ is an rwff;

(ii) we replace the rule $\bot E$ with the rule $\bot E_i$:

$$\frac{b_2 : \bot}{b_1 : A}\ \bot E_i \quad ;$$

(iii) we restrict the application of relational rules and of $\bot E_i$ to atomic conclusions.

We remark that the intuitionistic nature of the system is given by modification (ii), while (i) and (iii) are just introduced in order to simplify the normalization procedure.

We can now adapt the Gödel-Gentzen negative translation $(\cdot)^g$ (see, e.g., [152]) to our case.

Definition 5.16. *For all formulas of $BCTL^*_-$ the* negative translation $(\cdot)^g$ *is defined inductively as follows:*

$$\begin{array}{lcl}
(p)^g & = & \neg\neg p\,, \text{ for } p \text{ atomic and } p \neq \bot\,; \\
(\bot)^g & = & \bot\,; \\
(A \supset B)^g & = & (A)^g \supset (B)^g\,; \\
(A \wedge B)^g & = & (A)^g \wedge (B)^g\,; \\
(\mathsf{X}A)^g & = & \mathsf{X}(A)^g\,; \\
(\mathsf{G}A)^g & = & \mathsf{G}(A)^g\,; \\
(\forall A)^g & = & \forall (A)^g\,.
\end{array}$$

By extension, we define the negative translation for lwffs and rwffs as follows:

$$\begin{array}{lcl}
(b : A)^g & = & b : (A)^g\,; \\
(\rho)^g & = & \rho\,.
\end{array}$$

Lemma 5.17. *Given a set Γ of lwffs and rwffs and an lwff $b : A$, it holds*

$$\Gamma \vdash_{\mathcal{N}(BCTL^*_-)} b : A \qquad \textit{iff} \qquad (\Gamma)^g \vdash_{\mathcal{N}(BCTL^*_{-i})} (b : A)^g \quad ,$$

where $\Gamma^g = \{(\varphi)^g \mid \varphi \in \Gamma\}$.

Proof. By induction on the length of the derivation, we show that for every proof in $\mathcal{N}(BCTL^*_-)$ there exists an equivalent derivation in $\mathcal{N}(BCTL^*_{-i})$. The only interesting case is when the last rule applied is $\bot E$. Let Π be the following derivation:

$$\dfrac{\begin{array}{c}[b:\neg A]\\ \Pi_1\\ b:\bot\end{array}}{b:A}\ \bot E \quad .$$

By the induction hypothesis, there exists an $\mathcal{N}(BCTL^*_{-i})$ derivation Π_1^g equivalent to Π_1. Then we can obtain the following $\mathcal{N}(BCTL^*_{-i})$ derivation, which is equivalent to Π:

$$\dfrac{\dfrac{\Pi'}{b:\neg\neg A^g \supset A^g} \quad \dfrac{\begin{array}{c}[\neg A^g]\\ \Pi_1^g\\ b:\bot\end{array}}{b:\neg\neg A^g}\ \supset I}{b:A^g}\ \supset E \quad ,$$

where Π' is some proof of $b:\neg\neg A^g \supset A^g$, which is clearly provable in $\mathcal{N}(BCTL^*_{-i})$. As an example, we show a derivation of $b:\neg\neg A^g \supset A^g$ in the case when $A = p$

$$\dfrac{\dfrac{\dfrac{[b:\neg p]^2 \quad [b:\neg\neg p]^3}{b:\bot}\ \supset E}{b:\neg\neg\neg p}\ \supset I^3 \qquad [b:\neg\neg\neg\neg p]^1}{\dfrac{\dfrac{b:\bot}{b:\neg\neg p}\ \supset I^2}{b:\neg\neg\neg\neg p \supset \neg\neg p}\ \supset I^1}\ \supset E$$

and further note that $(\cdot)^g$ preserves intuitionistic provability.

□

In Section 5.4.7, we will prove consistency for the system $\mathcal{N}(BCTL^*_{-i})$. By Lemma 5.17, such a result can be also used to prove the consistency of the classical version $\mathcal{N}(BCTL^*_{-})$ of the system.

5.4.2 The normal form of derivations

Derivations in normal form

In normalizing derivations of $\mathcal{N}(BCTL^*_{-i})$, we have to consider some more forms of detours than in standard natural deduction normalization processes for classical or intuitionistic logic (see, e.g., [125]). In particular, in order to get a normal form that allows us to prove the consistency of the system, we need to reduce (as in [74, 151]) some applications of *ind*; namely, those applications in which the relational premise, say $b_0 \leqslant b_n$, is "obtained" by a chain of labels, leading from b_0 to b_n, where every element of the chain is linked to the next one by the relational symbol $\lhd$. In the following definitions, we will clarify what we mean with "obtained".

Definition 5.18. *We call* $\leqslant$-formulas *the rwffs of the form* $b_1 \leqslant b_2$ *and* $\lhd$-formulas *the rwffs of the form* $b_1 \lhd b_2$. *Let* Π *be a* $\mathcal{N}(BCTL^*_{-i})$ *derivation. We say that a discharged* $\leqslant$*-formula occurrence* immediately depends on *an rwff occurrence* ρ' *if* ρ *is discharged by an application of base* $\leqslant$, *trans* $\leqslant$ *or lin*$\lhd_{\mathcal{R}}$ *that contains* ρ' *as a premise. We also say that* ρ depends on ρ' *if there exists a sequence (possibly*

of length 1*)* $\rho_1, \ldots, \rho_n$ *such that* $\rho_1 \equiv \rho$, $\rho_n \equiv \rho'$, *and* ρ_i *immediately depends on* ρ_{i+1} *for each* $1 \leq i < n$.

The dependence tree *of* ρ *is the tree of rwff occurrences, whose root is* ρ *and such that every rwff in the tree has the rwffs on which it immediately depends as children.*

In other words, the relation *depends* is the reflexive and transitive closure of the relation *immediately depends*.

Definition 5.19. *Let* ρ *be a* $\leqslant$*-formula occurrence in a derivation* Π*. We say that* ρ *is* unfoldable *in* Π *if each leaf of its dependence tree is:*

(*i*) *a* $\lhd$*-formula; or*
(*ii*) *a* $\leqslant$*-formula of the form* $b \leqslant b$ *for some label* b.

Definition 5.20. *Let* Π *be a derivation obtained by applying* r *to the conclusion of a derivation* Π_1*, for which a formula* ρ *is an assumption, and possibly discharging (by* r*) some of the assumptions of* Π_1*:*

$$\Pi \;=\; \dfrac{\begin{array}{c}[\rho]\\ \Pi_1 \\ b : A\end{array}}{}\, r \quad .$$

We say that r unfolds ρ *if* ρ *is unfoldable in* Π *but not in* Π_1.

In order to give the definition of normal form, it is convenient to extend the notion of unfoldability also to applications of *ind*.

Definition 5.21. *Let* s *be an application of ind. Then* s *will have the following form (for some labels* b_0, b_n, b_i, b_j*):*

$$\dfrac{\begin{array}{cccc} & [b_0 \leqslant b_i] & [b_i \lhd b_j] & [b_i : A] \\ & & \vdots & \\ b_0 : A \quad b_0 \leqslant b_n & & b_j : A & \end{array}}{b_n : A}\, s \quad .$$

We call the premises $b_0 : A$, $b_0 \leqslant b_n$ *and* $b_j : A$ *the* base premise*, the* ending premise *and the* inductive premise *of* s*, respectively.* b_0 *and* b_n *are called respectively the* base label *and the* ending label *of* s.

We say that an ind-application s *is* unfoldable *in a derivation* Π *if its ending premise is unfoldable in* Π*. Finally, we say that an application* r *unfolds* s *if* r *unfolds the ending premise of* s.

We can now adapt the standard definitions of maximum formula and normal form (see, e.g., [125]) to our case.

Definition 5.22. *A formula occurrence* $b : A$ *is a* maximum formula *in* Π *if it is:*

(*i*) *both the conclusion of an introduction rule application and the major premise of an elimination rule application; or*
(*ii*) *the conclusion of an unfoldable application of ind.*

Note that while (i) is standard, (ii) is specific to our case. As we will show in Section 5.4.7, this further condition is necessary in order to get a normal form that allows for proving the consistency of the system.

Definition 5.23. *A derivation Π is in* normal form *(is a* normal derivation*) if Π contains no maximum formulas.*

Derivations in standard form

In contracting applications of *ind*, we have to deal with a further technical complication. Namely, such contractions (see Section 5.4.3) will require the addition of some relational assumptions to the fragment of derivation involved in the contraction. In order to make the contraction admissible we need to be sure that all such assumptions are "justified", i.e. they are either dischargeable or open assumptions already occurring in the original derivation. We will show that for every $\mathcal{N}(BCTL^*_{-i})$ derivation there exists an equivalent one in such a form (we will call it a standard form) that all the assumptions of this kind can be in fact justified. We formalize all these notions as follows.

Definition 5.24. *Given a derivation Π and a $\leqslant$-formula ρ in it, we say that an rwff ρ' is* dischargeable above ρ *(in Π) if:*

(i) *it is an open assumption of Π; or*
(ii) *it is dischargeable by one of the rule applications occurring in Π below ρ.*

Note that, according to the previous definition, a formula ρ' dischargeable above a formula ρ in a derivation Π must not necessarily occur in Π.

Definition 5.25. *Given a $\leqslant$-formula $\rho \equiv b_0 \leqslant b_n$, a* chain *for ρ is a sequence (possibly of length 0, i.e. b_0 and b_n coincide) of rwffs*

$$b_0 \lhd b_1, b_1 \lhd b_2, \ldots, b_{n-1} \lhd b_n$$

for some labels $b_1, \ldots, b_{n-1}$. We say that a chain is dischargeable above *a formula ρ in a derivation Π if every formula in the chain is dischargeable above ρ in Π.*

In the following lemma, we prove that if a $\leqslant$-formula ρ is unfoldable in a derivation Π, then there is a chain for ρ in Π which is dischargeable above ρ. It might be the case that such a chain is "hidden" in the derivation, i.e. it is in some way inferable but not actually dischargeable in Π. Anyway it is always possible to make it "explicit" by means of $lin\lhd_{\mathcal{R}}$-applications.

As an example, consider a derivation Π such that a $\leqslant$-formula $\rho = b_0 \leqslant b_n$ occurs in it and a set of assumptions $\{b_0 \lhd b_1, b_1 \lhd b_2, b'_2 \lhd b_3, \ldots, b_{n-1} \lhd b_n\} \cup \{\overline{b} \lhd b_2, \overline{b} \lhd b'_2\}$ is dischargeable above ρ. It is immediate to observe that, from such a set, one can infer a chain for ρ by just adding an application of $lin\lhd_{\mathcal{R}}$.[4]

Lemma 5.26. *If a $\leqslant$-formula occurrence $\rho \equiv b_0 \leqslant b_n$ is unfoldable in a derivation Π then there is a derivation Π' equivalent to Π (i.e. with the same open assumptions and the same conclusion) that is obtained from Π by only inserting into Π a number of applications of $lin\lhd_{\mathcal{R}}$ (possibly none) and such that there exists a chain for ρ that is dischargeable above ρ in Π'.*

[4] Note that it is necessary to include such cases in the definition of unfoldability in order to have a normal form that allows us to prove consistency.

Proof. The definition of unfoldable formula implies that every $\leqslant$-formula in the dependence tree of ρ is unfoldable as well. By observing the rules *base* $\leqslant$, *trans* $\leqslant$ and *lin*$\lhd_{\mathcal{R}}$, we also notice that the top of the dependence tree of an unfoldable formula is composed by a subtree of $\leqslant$-formulas. The proof is by induction on the height of such a subtree. As a base case, if the $\leqslant$-subtree has height 1, then ρ is either:

(i) such that b_0 and b_n coincide and then an empty chain for ρ is trivially dischargeable above ρ; or
(ii) discharged by an application of *base* $\leqslant$ and then a chain for ρ is trivially dischargeable above ρ in Π by the *base* $\leqslant$-application itself.

If ρ is discharged by an application of *trans* $\leqslant$ whose other premises are ρ_1 and ρ_2 then, by the induction hypothesis, we have that there exists a Π' in which both a chain for ρ_1 and a chain for ρ_2 are dischargeable above ρ. But then their composition gives a chain for ρ that is still dischargeable above ρ in Π'. Finally, let ρ be discharged by an application of *lin*$\lhd_{\mathcal{R}}$: we have two cases according to the fact that the substitution is applied to b_0 or to b_n. Let us consider the first one; the other is analogous. The application discharging $b_0 \leqslant b_n$ will have the following form:

$$\dfrac{\widehat{b} \lhd b'_0 \quad \widehat{b} \lhd b_0 \quad b'_0 \leqslant b_n \qquad \begin{array}{c}[b_0 \leqslant b_n]\\ \Pi_1 \\ b : A\end{array}}{b : A}\ \mathit{lin}\lhd_{\mathcal{R}} \quad ,$$

for some labels $\widehat{b}$ and b'_0. By the induction hypothesis, there is a Π' equivalent to Π in which a chain $b'_0 \lhd b_1, b_1 \lhd b_2, \ldots, b_{n-1} \lhd b_n$ is dischargeable above the occurrence of $b'_0 \leqslant b_n$ (and thus above ρ). Since the dischargeability of a chain for a $\leqslant$-formula depends only on the rule applications below that formula, we can assume, without loss of generality, that the fragment of derivation shown above occurs in Π' also. But then by replacing that fragment of derivation by the following one, where we only add a further *lin*$\lhd_{\mathcal{R}}$ application:

$$\dfrac{\widehat{b} \lhd b'_0 \quad \widehat{b} \lhd b_0 \quad b'_0 \leqslant b_n \qquad \dfrac{\widehat{b} \lhd b'_0 \quad \widehat{b} \lhd b_0 \quad b'_0 \lhd b_1 \qquad \begin{array}{c}[b_0 \leqslant b_n]\\ \Pi_1 \\ b : A\end{array}}{b : A}\ \mathit{lin}\lhd_{\mathcal{R}}}{b : A}\ \mathit{lin}\lhd_{\mathcal{R}} \quad ,$$

we get a new derivation Π'' that is still equivalent to Π and in which a chain for ρ is dischargeable above ρ.

□

Lemma 5.27. *If a rule application r unfolds a formula ρ, then r is an application of base* $\leqslant$ *or lin*$\lhd_{\mathcal{R}}$.

Proof. By inspecting the rules of $\mathcal{N}(BCTL^*_{-i})$ and the definition of a dependence tree (Definition 5.18), one can observe that *base* $\leqslant$ and *lin*$\lhd_{\mathcal{R}}$ are the only rules that can introduce in a derivation, as a premise, a $\lhd$-formula on which ρ depends.

□

As we will define formally in Section 5.4.3, given an unfoldable *ind*-application s in Π and a chain $b_0 \lhd b_1\,,\, b_1 \lhd b_2\,,\,\ldots\,,\, b_{n-1} \lhd b_n$ for it, in order to replace s we need to be sure that every rwff of the form $b_0 \leqslant b_i$ (and not only the chain itself), for $0 \leq i < n$, is dischargeable in Π. This could require the addition of some applications of *trans* $\leqslant$, *base* $\leqslant$ or *refl* $\leqslant$. For this reason, in the following we will:

(*i*) define a standard form for derivations, where the intuitive idea is that a standard derivation contains all such further applications;
(*ii*) show that every derivation is equivalent to (and can be transformed into) a derivation in standard form; and
(*iii*) study normalization with respect to the set of standard derivations.

Definition 5.28. *A derivation Π is in* standard form *(is a* standard derivation*) if for each unfoldable $\leqslant$-formula $\rho \equiv b_0 \leqslant b_n$ occurring in Π:*

(i) a chain μ for ρ is dischargeable above ρ in Π; and
(ii) for each b_i occurring in μ, the rwff $b_0 \leqslant b_i$ is dischargeable above ρ in Π.

Lemma 5.29. *Given a derivation Π in $\mathcal{N}(BCTL^*_{-i})$, it is always possible to define an equivalent derivation Π' that is in standard form.*

Proof. For each unfoldable $\leqslant$-formula ρ in Π, Lemma 5.26 suggests a way of obtaining a derivation Π' equivalent to the original one and such that a chain $\mu \equiv b_0 \lhd b_1, b_1 \lhd b_2, \ldots, b_{n-1} \lhd b_n$ for ρ is dischargeable above ρ in Π', i.e. we only have to add $lin{\lhd_{\mathcal{R}}}$-applications in the way suggested in the lemma. In order to satisfy also condition (*ii*) of Definition 5.28, we apply the following procedure, which (possibly) enriches the original derivation with further applications of relational rules:

(*i*) if $b_0 \leqslant b_0$ is not dischargeable above ρ, then we add an application of *refl* $\leqslant$ discharging $b_0 \leqslant b_0$;
(*ii*) for each $b_i \lhd b_{i+1}$ in μ, if $b_i \leqslant b_{i+1}$ is not dischargeable above ρ, then we add an application of *base* $\leqslant$ discharging $b_i \leqslant b_{i+1}$;
(*iii*) for $1 < i < n$, if $b_0 \leqslant b_i$ is not dischargeable above ρ, then we add an application of *trans* $\leqslant$ discharging $b_0 \leqslant b_i$ (and whose premises are $b_0 \leqslant b_{i-1}$ and $b_{i-1} \leqslant b_i$) .

Such rules can always be applied above the uppermost atomic lwff occurring below ρ (at least one such an lwff does exist since the application that unfolds ρ has an atomic conclusion).

It is easy to check that the algorithm described above is well-defined (every step provides the premises needed for the subsequent ones) and gives a derivation in standard form equivalent to the original one as a result.

□

Fig. 5.5 gives an example of a transformation of a derivation into a standard form. In the starting phase, we add a $lin{\lhd_{\mathcal{R}}}$-application (denoted by 7) discharging $b_0 \lhd b_1$, as specified in Lemma 5.26, in order to get a derivation in which a chain for the $\leqslant$-formula $\rho \equiv b_0 \leqslant b_3$ is dischargeable above ρ. Then (step (*i*) of the procedure described in the proof of Lemma 5.29) we add a *refl* $\leqslant$-application

(8) that discharges $b_0 \leqslant b_0$. With regard to step (ii), we only add the *base* $\leqslant$-application 9. Finally, in step (iii) we make the formula $b_0 \leqslant b_2$ dischargeable by adding the *trans* $\leqslant$-application 10.

Since Lemma 5.29 holds, we can (and in the following will) restrict our attention to derivations in standard form.

5.4.3 Reduction of derivations

Here we present the contractions that will be used in our normalization process in order to remove the maximum formulas and we define a reduction relation ($\Rightarrow$) based on such contractions. We have two classes of contractions:

(i) proper contractions; and
(ii) induction contractions.

Such contractions are operations transforming a derivation (i) ending with the application of an elimination rule on a maximum formula, or (ii) containing an unfoldable induction into another derivation with the same conclusion. After a contraction, when needed, we can always rename labels in order to satisfy the conditions of Lemma 5.12.

Proper contractions

Proper contractions remove maximum formulas from a derivation. We have a contraction for each detour: $\supset I/\supset E$, $\wedge I/\wedge E$, $\mathsf{X}I/\mathsf{X}E$, $\mathsf{G}I/\mathsf{G}E$, $\forall I/\forall E$. Such contractions are quite standard [148, 159] and, as examples, we give here the ones for the cases $\supset I/\supset E$ and $\mathsf{X}I/\mathsf{X}E$:

$$\dfrac{\dfrac{\begin{array}{c}[b:A]^1\\ \Pi_1\\ b:B\end{array}}{b:A\supset B}\supset I^1 \quad \begin{array}{c}\Pi_2\\ b:A\end{array}}{b:B}\supset E \quad \rightsquigarrow \quad \begin{array}{c}\Pi_2\\ b:A\\ \Pi_1\\ b:B\end{array} \quad ,$$

$$\dfrac{\dfrac{\begin{array}{c}[b_1\lhd b_2]^1\\ \Pi\\ b_2:A\end{array}}{b_1:\mathsf{X}A}\mathsf{X}I^1 \quad b_1\lhd b_3}{b_3:A}\mathsf{X}E \quad \rightsquigarrow \quad \begin{array}{c}b_1\lhd b_3\\ \Pi[b_3/b_2]\\ b_3:A\end{array} \quad ,$$

where the condition of the rule $\mathsf{X}I$ according to which b_2 is fresh in Π ensures that the substitution with b_3 does not produce any undesired side-effect.

Induction contractions

We consider contractions for unfoldable applications of *ind*. The first contraction reduces *ind*-applications where the base label and the ending label coincide. The intuition behind this contraction is that when the chain consists of just one element we can replace the induction application with its base case:

$$\dfrac{\begin{array}{ccc}\Pi_0 & & [b_0\leqslant b_i]\,[b_i:A]\,[b_i\lhd b_j]\\ b_0:A & b_0\leqslant b_0 & \Pi_1\\ & & b_j:A\end{array}}{b_0:A}\,r \quad \rightsquigarrow \quad \begin{array}{c}\Pi_0\\ b_0:A\end{array} \quad .$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{b'_0 \lhd b_1}}{b : p}\ base \leqslant^1 \quad [b_0 \leqslant b_1]^1}{b_1 \lhd b_2}}{b : p}\ base \leqslant^2 \quad [b_1 \leqslant b_2]^2}{b_2 \lhd b_3}}{b : p}\ base \leqslant^3 \quad [b_2 \leqslant b_3]^3}{b : p}\ trans \leqslant^4 \quad [b'_0 \leqslant b_2]^4}{b : p}\ trans \leqslant^5 \quad [b_0 \leqslant b_3]^5 \quad \widehat{b} \lhd b_0 \quad \widehat{b} \lhd b'_0}{b : p}\ lin \lhd^6_{\mathcal{R}} \quad \begin{array}{c}[b_0 \leqslant b_3]^6 \\ \Pi_1 \\ b : p\end{array}$$

has the standard form

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{b'_0 \lhd b_1}}{b : p}\ base \leqslant^1 \quad [b_0 \leqslant b_1]^1}{b_1 \lhd b_2}}{b : p}\ base \leqslant^2 \quad [b_1 \leqslant b_2]^2}{b_2 \lhd b_3}}{b : p}\ base \leqslant^3 \quad [b_2 \leqslant b_3]^3}{b : p}\ trans \leqslant^4 \quad [b'_0 \leqslant b_2]^4}{b : p}\ trans \leqslant^5 \quad [b_0 \leqslant b_3]^5 \quad \widehat{b} \lhd b_0 \quad \widehat{b} \lhd b'_0}{b : p}\ lin \lhd^6_{\mathcal{R}} \quad \dfrac{}{b : p}\ refl \leqslant^8 \quad b'_0 \lhd b_1}{b : p}\ lin \lhd^7_{\mathcal{R}} \quad \dfrac{\dfrac{}{[b_0 \lhd b_1]^7}}{b : p}\ base \leqslant^9 \quad \dfrac{\dfrac{}{[b_0 \leqslant b_1]^9} \quad [b_1 \leqslant b_2]^2}{b : p}\ trans \leqslant^{10} \quad \begin{array}{c}[b_0 \leqslant b_3]^6 \\ \Pi_1 \\ b : p\end{array}$$

Fig. 5.5. An example of a transformation into a standard form.

The second contraction is applied when an *ind*-application is unfolded by a *base* $\leqslant$ or by a $lin{\lhd_{\mathcal{R}}}$. Here the main idea is that if there exists a chain, say of length n, leading from the base label to the ending label and composed by labels that are one the immediate successor of the other, then we do not need to use the induction principle; namely, we can replace the *ind*-application by a subderivation built up by applying the inductive step n times (one for every label in the chain).

Note that the structure of relational rules in $\mathcal{N}(BCTL^*_{-_i})$ is such that rwffs depend, in the sense of Definition 5.18, on formulas that occur below in a derivation. This forces us to consider the context in which an *ind* is applied along a derivation. We show here the case *base* $\leqslant$; the case $lin{\lhd_{\mathcal{R}}}$ is treated in an analogous way, by simply substituting the last rule. We denote with s the *ind*-application and with r the *base* $\leqslant$-application that unfolds s. The fact that we deal with derivations in standard form ensures that all the assumptions of the form $b_0 \leqslant b_i$ and $b_i \lhd b_{i+1}$ added by the contraction are either open assumptions already occurring below s or dischargeable (and in this case discharged by the contraction step) by some rule application in Π_2 or by r. We denote this by using the symbol †.

$$\dfrac{\dfrac{\dfrac{\Pi_0}{b_0 : A} \quad \dfrac{[b_0 \leqslant b_n] \quad [b_0 \leqslant b_i]^1 \; [b_i : A]^1 \; [b_i \lhd b_j]^1}{\dfrac{\Pi_1}{b_j : A}}}{b_n : A}\, s^1 \qquad \dfrac{\Pi_2}{b : p}}{b : p}\, r$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Pi_0}{b_0 \leqslant b_0^\dagger} \quad \dfrac{\Pi_0}{b_0 : A} \quad \dfrac{\Pi_0}{b_0 \lhd b_1^\dagger}}{\Pi_1[b_0/b_i][b_1/b_j]}}{b_0 \leqslant b_1^\dagger} \quad \dfrac{\Pi_1[b_0/b_i][b_1/b_j]}{b_1 : A} \quad \dfrac{\Pi_1[b_0/b_i][b_1/b_j]}{b_1 \lhd b_2^\dagger}}{\Pi_1[b_1/b_i][b_2/b_j]}}{b_2 : A}}{\vdots}}{b_0 \leqslant b_{n-1}^\dagger} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Pi_0}{b_0 : A} \quad \dfrac{\Pi_0}{b_0 \lhd b_1^\dagger}}{\Pi_1[b_0/b_i][b_1/b_j]}}{b_1 : A} \quad \dfrac{\Pi_1[b_0/b_i][b_1/b_j]}{b_1 \lhd b_2^\dagger}}{\Pi_1[b_1/b_i][b_2/b_j]}}{b_2 : A}}{\vdots}}{b_{n-1} : A} \quad \dfrac{\dfrac{\dfrac{\dfrac{\Pi_0}{b_0 : A} \quad \dfrac{\Pi_0}{b_0 \lhd b_1^\dagger}}{\Pi_1[b_0/b_i][b_1/b_j]}}{b_1 : A} \quad \dfrac{\Pi_1[b_0/b_i][b_1/b_j]}{b_1 \lhd b_2^\dagger}}{\Pi_1[b_1/b_i][b_2/b_j]}}{b_2 : A}}{\vdots}}{b_{n-1} \lhd b_n^\dagger}}{b_n : A}}{\dfrac{\Pi_2}{b : p}}}{b : p}\, r$$

The reduction relation $\Rightarrow$

We define now a reduction relation between derivations built on the proper and induction contractions described above. It is important to notice that such contractions preserve the standard form of derivations and thus the whole process of normalization is in fact defined over the set of standard derivations.

Definition 5.30. *A* reduction sequence *is a sequence $\Pi_1, ..., \Pi_n$ of derivations such that Π_i is obtained from Π_{i-1} by applying a single proper or induction contraction to a subderivation of Π_i for $1 < i \leq n$. We say that Π* reduces *to Π', and we write $\Pi \Rightarrow \Pi'$, if there exists a reduction sequence (possibly of length 1) $\Pi_1, \ldots, \Pi_n$ such that $\Pi_1 = \Pi$ and $\Pi_n = \Pi'$.*

We also say that Π reduces to a normal form *(*has a normal form*) if there exists a Π' such that $\Pi \Rightarrow \Pi'$ and Π' is in normal form.*

It is immediate to observe that a derivation is in normal form iff it is not possible to apply any proper or inductive contractions to any of its subderivations.

5.4.4 The Church-Rosser property

Here we show that the Church-Rosser property, with regard to the relation $\Rightarrow$, holds for $\mathcal{N}(BCTL^*_{-i})$ derivations. We follow mainly [74], but some non-trivial adaptations are required by the presence of the rule *ind*.

The structure of the proof is the following:

- (*i*) first we will define a relation $\Rightarrow_1$ between derivations, where the idea is that $\Rightarrow_1$ builds $\Rightarrow$, i.e. $\Pi \Rightarrow \Pi'$ iff there exists a sequence of reductions $\Pi = \Pi_1 \Rightarrow_1 \ldots \Rightarrow_1 \Pi_n = \Pi'$ for some n;
- (*ii*) then we will prove that one-step confluence holds, i.e. if $\Pi \Rightarrow_1 \Pi'$ and $\Pi \Rightarrow_1 \Pi''$ then there exists a Π''' such that $\Pi' \Rightarrow_1 \Pi'''$ and $\Pi'' \Rightarrow_1 \Pi'''$;
- (*iii*) finally, as standard, we will use the previous result to prove confluence of $\Rightarrow$, i.e. if $\Pi \Rightarrow \Pi'$ and $\Pi \Rightarrow \Pi''$ then there exists a Π''' such that $\Pi' \Rightarrow \Pi'''$ and $\Pi'' \Rightarrow \Pi'''$.

While the proof of step (*iii*) is standard (once (*ii*) is given) some technical complications arise in proving (*ii*); we give here an intuition of the problem and a sketch of the solution that will be formalized in the following.

In (*ii*), in order to use an inductive argument, we need to prove the result with regard to a larger set of reductions. Namely, the problem comes from the definition of induction contractions (Section 5.4.3) that are not strictly local and in applying which we are required to consider at least a fragment of derivation (the one containing a chain) below the *ind*-application. This "non-locality" gives rise to difficulties when providing inductive definitions and when using inductive arguments within the proofs.

Thus we will make use of a further relation $\rightarrowtail_1$ (containing $\Rightarrow_1$) for which an inductive definition is provided. The definition is such that, given a derivation concluding with an *ind*-application r, a step of $\rightarrowtail_1$ allows us to mimic the application of an induction contraction on it, regardless both of the length of a possible chain for r and of the labels used in it, i.e. we will have a formation rule in the inductive

definition of $\rightarrowtail_1$ that will allow us to unfold an *ind*-application into a chain of length n for each possible n and for each possible choice of labels to be used in the unfolding. Clearly, such a rule implies that not every step of $\rightarrowtail_1$ will correspond to contractions as defined in Section 5.4.3.

We manage this by keeping track of those unfoldings of *ind*-applications that are not "justified" by the presence of an appropriate chain below them; we will call these unfoldings *defects*. Technically, the set of defects associated to a pair in $\rightarrowtail_1$ is defined inductively by adding a defect every time we unfold an *ind*-application and by removing it only when all the relational assumptions introduced by the unfolding are discharged. In order to keep track of the relational formulas introduced by the unfolding of some *ind*-application, we use a marking mechanism that consists in marking each such formula with a same symbol; when all the rwffs marked with a same symbol get discharged we can conclude that the unfolding is no longer a defect and we can remove it from the set.

Then the relation $\Rightarrow_1$ will be defined as the subset of $\rightarrowtail_1$ containing pairs with no defects.

We go now into technical details; first we need to formalize the notion of marking.

Definition 5.31. *Given a derivation Π and a set of marks Σ, a* marking *l for Π is a function that associates a mark in Σ to each rule application in Π and a mark in Σ to some of the rwffs in Π. A marking l for Π is said to be* standard *if l associates a different mark to each rule application of Π and no marks to any rwff of Π. A* marked derivation *is a pair (Π, l) where Π is a derivation and l is a marking for Π.*

As notation, we denote the mark associated to a rule application with a symbol between parentheses on the right of the application line and the mark associated to an rwff as a subscript of the formula. For simplicity, in the rest of this section, we will often omit to specify the marking and just use the symbol Π (possibly subscripted or superscripted) to denote also a marked derivation. The context will clarify whether we are referring to a marked or to an unmarked derivation.

Now we define the relation $\rightarrowtail_1$ between marked derivations, where the idea is that in one step of $\rightarrowtail_1$ we are allowed to perform at the same time more than one contraction, provided that they do not interfere with each other. Note that, as explained above, in the case of a derivation ending with an *ind*-application we are allowed to unfold it in any way and that the case corresponding to an induction contraction is the only one that introduces new marked rwffs. When a rule application introduces a relational open assumption as a premise of the rule, then possible marked rwffs of the same form can be made unmarked in the result of the transformation.

At the same time, we also define inductively the set of defects associated to each pair of derivations in $\rightarrowtail_1$. A defect is introduced when an induction contraction is performed (case [IndContr] below) and removed when all the marked rwffs introduced with such a contraction have been discharged. We remark that, as notation, in the following definition we specify the mark associated to a relational formula only when it seems to be relevant, i.e., when the mark is introduced. In the other cases, one can assume that the $\rightarrowtail_1$-step does not modify the marks,

i.e., each occurrence of a relational formula is either unmarked both on the left and on the right side of $\rightarrowtail_1$ or marked with the same symbol on both sides.

Definition 5.32. *Let Σ be a set of marks. We define the binary relation $\rightarrowtail_1$ between $\mathcal{N}(BCTL^*_{-i})$ derivations marked with symbols in Σ inductively as follows. Contextually, we define a function δ that maps every element of $\rightarrowtail_1$ into a subset of Σ.*

PASSIVE CLAUSES

(i) $[BC]$
$\Pi \rightarrowtail_1 \Pi$ *and* $\delta(\Pi, \Pi) = \emptyset$.

(ii) $[\supset I]$

$$\textit{If} \quad \begin{array}{c} b:A \\ \Pi_1 \\ b:B \end{array} \rightarrowtail_1 \begin{array}{c} b:A \\ \Pi_1' \\ b:B \end{array} ,$$

$$\textit{then} \quad \Pi = \dfrac{\begin{array}{c}[b:A] \\ \Pi_1 \\ b:B\end{array}}{b:A \supset B}\ \supset I\,(r) \quad \rightarrowtail_1 \quad \Pi' = \dfrac{\begin{array}{c}[b:A] \\ \Pi_1' \\ b:B\end{array}}{b:A \supset B}\ \supset I\,(r)$$

and $\delta(\Pi, \Pi') = \delta(\Pi_1, \Pi_1')$.

(iii) $[\wedge I]$

$$\textit{If} \quad \begin{array}{c} \Pi_1 \\ b:A \end{array} \rightarrowtail_1 \begin{array}{c} \Pi_1' \\ b:A \end{array} \quad \textit{and} \quad \begin{array}{c} \Pi_2 \\ b:B \end{array} \rightarrowtail_1 \begin{array}{c} \Pi_2' \\ b:B \end{array} ,$$

$$\textit{then} \quad \Pi = \dfrac{\begin{array}{cc}\Pi_1 & \Pi_2 \\ b:A & b:B\end{array}}{b:A \wedge B}\ \wedge I\,(r) \quad \rightarrowtail_1 \quad \Pi' = \dfrac{\begin{array}{cc}\Pi_1' & \Pi_2' \\ b:A & b:B\end{array}}{b:A \wedge B}\ \wedge I\,(r)$$

and $\delta(\Pi, \Pi') = \delta(\Pi_1, \Pi_1') \cup \delta(\Pi_2, \Pi_2')$.

(iv) $[\mathsf{X}I]$

$$\textit{If} \quad \begin{array}{c} b_1 \lhd b_2 \\ \Pi_1 \\ b_2:A \end{array} \rightarrowtail_1 \begin{array}{c} b_1 \lhd b_2 \\ \Pi_1' \\ b_2:A \end{array} ,$$

$$\textit{then} \quad \Pi = \dfrac{\begin{array}{c}[b_1 \lhd b_2] \\ \Pi_1 \\ b_2:A\end{array}}{b_1 : \mathsf{X}A}\ \mathsf{X}I\,(r) \quad \rightarrowtail_1 \quad \Pi' = \dfrac{\begin{array}{c}[b_1 \lhd b_2] \\ \Pi_1' \\ b_2:A\end{array}}{b_1 : \mathsf{X}A}\ \mathsf{X}I\,(r)$$

and $\delta(\Pi, \Pi') = \delta(\Pi_1, \Pi_1') \setminus \{s \in \Sigma \mid$ *all the rwffs marked with s, if any, are discharged in Π'*$\}$.

(v) $[\mathsf{G}I]$

$$\textit{If} \quad \begin{array}{c} b_1 \leqslant b_2 \\ \Pi_1 \\ b_2:A \end{array} \rightarrowtail_1 \begin{array}{c} b_1 \leqslant b_2 \\ \Pi_1' \\ b_2:A \end{array} ,$$

then $\Pi = \dfrac{\begin{matrix}[b_1 \leqslant b_2] \\ \Pi_1 \\ b_2 : A\end{matrix}}{b_1 : \mathsf{G}A}\ \mathsf{G}I\,(r) \rightarrowtail_1 \Pi' = \dfrac{\begin{matrix}[b_1 \leqslant b_2] \\ \Pi'_1 \\ b_2 : A\end{matrix}}{b_1 : \mathsf{G}A}\ \mathsf{G}I\,(r)$

and $\delta\,(\Pi, \Pi') = \delta\,(\Pi_1, \Pi'_1) \setminus \{s \in \Sigma \mid$ *all the rwffs marked with* s*, if any, are discharged in* $\Pi'\}$.

(vi) $[\forall I]$

If $\begin{matrix} b_1 \bullet b_2 \\ \Pi_1 \\ b_2 : A \end{matrix} \rightarrowtail_1 \begin{matrix} b_1 \bullet b_2 \\ \Pi'_1 \\ b_2 : A \end{matrix}$,

then $\Pi = \dfrac{\begin{matrix}[b_1 \bullet b_2] \\ \Pi_1 \\ b_2 : A\end{matrix}}{b_1 : \forall A}\ \forall I\,(r) \rightarrowtail_1 \Pi' = \dfrac{\begin{matrix}[b_1 \bullet b_2] \\ \Pi'_1 \\ b_2 : A\end{matrix}}{b_1 : \forall A}\ \forall I\,(r)$

and $\delta\,(\Pi, \Pi') = \delta\,(\Pi_1, \Pi'_1)$.

(vii) $[\supset E]$

If $\begin{matrix}\Pi_1 \\ b : A \supset B\end{matrix} \rightarrowtail_1 \begin{matrix}\Pi'_1 \\ b : A \supset B\end{matrix}$ *and* $\begin{matrix}\Pi_2 \\ b : A\end{matrix} \rightarrowtail_1 \begin{matrix}\Pi'_2 \\ b : A\end{matrix}$,

then $\Pi = \dfrac{\begin{matrix}\Pi_1 \\ b : A \supset B\end{matrix} \quad \begin{matrix}\Pi_2 \\ b : A\end{matrix}}{b : B}\ \supset E\,(r) \rightarrowtail_1 \Pi' = \dfrac{\begin{matrix}\Pi'_1 \\ b : A \supset B\end{matrix} \quad \begin{matrix}\Pi'_2 \\ b : A\end{matrix}}{b : B}\ \supset E\,(r)$

and $\delta\,(\Pi, \Pi') = \delta\,(\Pi_1, \Pi'_1) \cup \delta\,(\Pi_2, \Pi'_2)$.

(viii) $[\wedge E_1]$

If $\begin{matrix}\Pi_1 \\ b : A \wedge B\end{matrix} \rightarrowtail_1 \begin{matrix}\Pi'_1 \\ b : A \wedge B\end{matrix}$,

then $\Pi = \dfrac{\begin{matrix}\Pi_1 \\ b : A \wedge B\end{matrix}}{b : A}\ \wedge E_1\,(r) \rightarrowtail_1 \Pi' = \dfrac{\begin{matrix}\Pi'_1 \\ b : A \wedge B\end{matrix}}{b : A}\ \wedge E_1\,(r)$

and $\delta\,(\Pi, \Pi') = \delta\,(\Pi_1, \Pi'_1)$.

(ix) $[\wedge E_2]$

If $\begin{matrix}\Pi_1 \\ b : A \wedge B\end{matrix} \rightarrowtail_1 \begin{matrix}\Pi'_1 \\ b : A \wedge B\end{matrix}$,

then $\Pi = \dfrac{\begin{matrix}\Pi_1 \\ b : A \wedge B\end{matrix}}{b : B}\ \wedge E_2\,(r) \rightarrowtail_1 \Pi' = \dfrac{\begin{matrix}\Pi'_1 \\ b : A \wedge B\end{matrix}}{b : B}\ \wedge E_2\,(r)$

and $\delta\,(\Pi, \Pi') = \delta\,(\Pi_1, \Pi'_1)$.

(x) $[\mathsf{X}E]$

If $\begin{array}{c}\Pi_1\\ b_1 : \mathsf{X}A\end{array} \rightarrowtail_1 \begin{array}{c}\Pi_1'\\ b_1 : \mathsf{X}A\end{array}$,

then $\Pi = \dfrac{\begin{array}{c}\Pi_1\\ b_1 : \mathsf{X}A\end{array} \quad b_1 \lhd b_2}{b_2 : A}\ \mathsf{X}E\,(r) \ \rightarrowtail_1\ \Pi' = \dfrac{\begin{array}{c}\Pi_1'\\ b_1 : \mathsf{X}A\end{array} \quad b_1 \lhd b_2}{b_2 : A}\ \mathsf{X}E\,(r)$

where if there are marked occurrences of $b_1 \lhd b_2$ *in* Π_1*, then their corresponding occurrences in* Π_1' *can be unmarked and* $\delta\,(\Pi, \Pi') = \delta\,(\Pi_1, \Pi_1') \setminus \{s \in \Sigma \mid$ *all the rwffs marked with* s*, if any, are discharged in* $\Pi'\}$*.*

(xi) $[\mathsf{G}E]$

If $\begin{array}{c}\Pi_1\\ b_1 : \mathsf{G}A\end{array} \rightarrowtail_1 \begin{array}{c}\Pi_1'\\ b_1 : \mathsf{G}A\end{array}$,

then $\Pi = \dfrac{\begin{array}{c}\Pi_1\\ b_1 : \mathsf{G}A\end{array} \quad b_1 \leqslant b_2}{b_2 : A}\ \mathsf{G}E\,(r) \ \rightarrowtail_1\ \Pi' = \dfrac{\begin{array}{c}\Pi_1'\\ b_1 : \mathsf{G}A\end{array} \quad b_1 \leqslant b_2}{b_2 : A}\ \mathsf{G}E\,(r)$

where if there are marked occurrences of $b_1 \leqslant b_2$ *in* Π_1*, then their corresponding occurrences in* Π_1' *can be unmarked and* $\delta\,(\Pi, \Pi') = \delta\,(\Pi_1, \Pi_1') \setminus \{s \in \Sigma \mid$ *all the rwffs marked with* s*, if any, are discharged in* $\Pi'\}$*.*

(xii) $[\forall E]$

If $\begin{array}{c}\Pi_1\\ b_1 : \forall A\end{array} \rightarrowtail_1 \begin{array}{c}\Pi_1'\\ b_1 : \forall A\end{array}$,

then $\Pi = \dfrac{\begin{array}{c}\Pi\\ b_1 : \forall A\end{array} \quad b_1 \bullet b_2}{b_2 : A}\ \forall E\,(r) \ \rightarrowtail_1\ \Pi' = \dfrac{\begin{array}{c}\Pi'\\ b_1 : \forall A\end{array} \quad b_1 \bullet b_2}{b_2 : A}\ \forall E\,(r)$

and $\delta\,(\Pi, \Pi') = \delta\,(\Pi_1, \Pi_1')$ *.*

(xiii) $[\bot E]$

If $\begin{array}{c}\Pi_1\\ b_1 : \bot\end{array} \rightarrowtail_1 \begin{array}{c}\Pi_1'\\ b_1 : \bot\end{array}$,

then $\Pi = \dfrac{\begin{array}{c}\Pi_1\\ b_1 : \bot\end{array}}{b : A}\ \bot E\,(r) \ \rightarrowtail_1\ \Pi' = \dfrac{\begin{array}{c}\Pi_1'\\ b_1 : \bot\end{array}}{b : A}\ \bot E\,(r)$

and $\delta\,(\Pi, \Pi') = \delta\,(\Pi_1, \Pi_1')$ *.*

(xiv) [*base* $\leqslant$], [*lin*$\lhd_{\mathcal{R}}$], [*ser*$\lhd$], [*lin*$\lhd$], [*refl* $\leqslant$], [*trans* $\leqslant$], [*refl*$\bullet$], [*symm*$\bullet$], [*trans*$\bullet$], [*atom*$\bullet$], [*fusion*]
These cases are all very similar. When relational open assumptions are introduced as premises of the rule application, possible marked rwffs of the same form can be made unmarked in the result of the transformation. As example cases, we show base $\leqslant$ *and refl* $\leqslant$*.*

If $\begin{array}{c} b_1 \leqslant b_2 \\ \Pi_1 \\ b : A \end{array} \rightarrowtail_1 \begin{array}{c} b_1 \leqslant b_2 \\ \Pi_1' \\ b : A \end{array}$,

then

$$\Pi = \dfrac{\begin{array}{cc} & [b_1 \leqslant b_2]^1 \\ & \Pi_1 \\ b_1 \lhd b_2 & b : A \end{array}}{b : A}\ base \leqslant^1\ (r)$$

$$\rightarrowtail_1$$

$$\Pi' = \dfrac{\begin{array}{cc} & [b_1 \leqslant b_2]^1 \\ & \Pi_1' \\ b_1 \lhd b_2 & b : A \end{array}}{b : A}\ base \leqslant^1\ (r)$$

where if there are marked occurrences of $b_1 \lhd b_2$ *in* Π_1*, then their corresponding occurrences in* Π_1' *can be unmarked and* $\delta(\Pi, \Pi') = \delta(\Pi_1, \Pi_1') \setminus \{s \in \Sigma \mid$ *all the rwffs marked with* s*, if any, are discharged in* $\Pi'\}$*.*

If $\begin{array}{c} b_1 \leqslant b_1 \\ \Pi_1 \\ b : A \end{array} \rightarrowtail_1 \begin{array}{c} b_1 \leqslant b_1 \\ \Pi_1' \\ b : A \end{array}$,

then $\Pi = \dfrac{\begin{array}{c} [b_1 \leqslant b_1]^1 \\ \Pi_1 \\ b : A \end{array}}{b : A}\ refl \leqslant^1\ (r) \rightarrowtail_1 \Pi' = \dfrac{\begin{array}{c} [b_1 \leqslant b_1]^1 \\ \Pi_1' \\ b : A \end{array}}{b : A}\ refl \leqslant^1\ (r)$

where $\delta(\Pi, \Pi') = \delta(\Pi_1, \Pi_1') \setminus \{s \in \Sigma \mid$ *all the rwffs marked with* s*, if any, are discharged in* $\Pi'\}$*.*

(xv) [*ind*]

If $\begin{array}{c} \Pi_0 \\ b_0 : A \end{array} \rightarrowtail_1 \begin{array}{c} \Pi_0' \\ b_0 : A \end{array}$

and $\begin{array}{ccc} b_0 \leqslant b_i & b_i : A & b_i \lhd b_j \\ & \Pi_1 & \\ & b_j : A & \end{array} \rightarrowtail_1 \begin{array}{ccc} b_0 \leqslant b_i & b_i : A & b_i \lhd b_j \\ & \Pi_1' & \\ & b_j : A & \end{array}$

then

$$\Pi = \dfrac{\begin{array}{ccc} \Pi_0 & & [b_0 \leqslant b_i]\,[b_i : A]\,[b_i \lhd b_j] \\ b_0 : A & b_0 \leqslant b & \Pi_1 \\ & & b_j : A \end{array}}{b : A}\ ind\,(r)$$

$$\rightarrowtail_1$$

$$\Pi' = \dfrac{\begin{array}{ccc} & & [b_0 \leqslant b_i]\,[b_i : A]\,[b_i \lhd b_j] \\ \Pi'_0 & & \Pi'_1 \\ b_0 : A \quad b_0 \leqslant b & & b_j : A \end{array}}{b : A}\ ind\,(r)$$

where if there are marked occurrences of $b_0 \leqslant b$ *in* Π_0 *or* Π_1*, then their corresponding occurrences in* Π'_0 *and* Π'_1 *can be unmarked and* $\delta\,(\Pi, \Pi') = \delta\,(\Pi_0, \Pi'_0) \cup \delta\,(\Pi_1, \Pi'_1) \setminus \{s \in \Sigma \mid$ *all the rwffs marked with* s*, if any, are discharged in* $\Pi'\}$*.*

ACTIVE CLAUSES

(xvi) $[IndContr]$

If $\begin{array}{c} \Pi_0 \\ b_0 : A \end{array} \rightarrowtail_1 \begin{array}{c} \Pi'_0 \\ b_0 : A \end{array}$

and $\begin{array}{ccccc} & b_0 \leqslant b_i & b_i : A & b_i \lhd b_j & \\ & & \Pi_1 & & \\ & & b_j : A & & \end{array} \rightarrowtail_1 \begin{array}{ccccc} & b_0 \leqslant b_i & b_i : A & b_i \lhd b_j & \\ & & \Pi'_1 & & \\ & & b_j : A & & \end{array}$

then for each n *and for every choice of labels* $b_1, \ldots, b_{n-1}$

$$\Pi = \dfrac{\begin{array}{ccc} & & [b_0 \leqslant b_i]\,[b_i : A]\,[b_i \lhd b_j] \\ \Pi_0 & & \Pi_1 \\ b_0 : A \quad b_0 \leqslant b & & b_j : A \end{array}}{b : A}\ ind\,(r)$$

$$\rightarrowtail_1$$

$$\Pi' = \begin{array}{c} \Pi'_0 \\ b_0 \leqslant b_{0(r)} \quad b_0 : A \quad b_0 \lhd b_{1(r)} \\ \Pi'_1[b_0/b_i][b_1/b_j] \\ b_0 \leqslant b_{1(r)} \quad b_1 : A \quad b_1 \lhd b_{2(r)} \\ \Pi'_1[b_1/b_i][b_2/b_j] \\ b_2 : A \\ \vdots \\ b_0 \leqslant b_{n-1(r)} \quad b_{n-1} : A \quad b_{n-1} \lhd b_{(r)} \\ \Pi'_1[b_{n-1}/b_i][b/b_j] \\ b : A \end{array}$$

where if there are marked occurrences of $b_0 \leqslant b$ *in* Π_0 *or* Π_1*, then their corresponding occurrences in* Π'_0 *and* Π'_1 *can be unmarked and* $\delta(\Pi, \Pi') = \delta(\Pi_0, \Pi'_0) \cup \delta(\Pi_1, \Pi'_1) \cup \{r\} \setminus \{s \in \Sigma \mid$ *all the rwffs marked with* s*, if any, are discharged in* $\Pi'\}$*.*

$(xvii)$ $[\supset I/\supset E]$

If $\quad \begin{matrix}\Pi_1\\ b:A\end{matrix} \rightarrowtail_1 \begin{matrix}\Pi'_1\\ b:A\end{matrix}$ *and* $\begin{matrix}b:A\\ \Pi_2\\ b:B\end{matrix} \rightarrowtail_1 \begin{matrix}b:A\\ \Pi'_2\\ b:B\end{matrix}$,

then $\quad \Pi = \dfrac{\Pi_1 \quad \dfrac{\begin{matrix}[b:A]\\ \Pi_2\\ b:B\end{matrix}}{b:A \supset B}\supset I}{b:B}\supset E \quad \rightarrowtail_1 \quad \Pi' = \begin{matrix}\Pi'_1\\ b:A\\ \Pi'_2\\ b:B\end{matrix}$

and $\delta(\Pi, \Pi') = \delta(\Pi_1, \Pi'_1) \cup \delta(\Pi_2, \Pi'_2)$*.*

$(xviii)$ $[\wedge I/\wedge E_1]$

If $\quad \begin{matrix}\Pi_1\\ b:A\end{matrix} \rightarrowtail_1 \begin{matrix}\Pi'_1\\ b:A\end{matrix}$ *and* $\begin{matrix}\Pi_2\\ b:B\end{matrix} \rightarrowtail_1 \begin{matrix}\Pi'_2\\ b:B\end{matrix}$,

then $\quad \Pi = \dfrac{\begin{matrix}\Pi_1\\ b:A\end{matrix} \quad \begin{matrix}\Pi_2\\ b:B\end{matrix}}{\dfrac{b:A \wedge B}{b:A}\wedge E_1}\wedge I \quad \rightarrowtail_1 \quad \Pi' = \begin{matrix}\Pi'_1\\ b:A\end{matrix}$

and $\delta(\Pi, \Pi') = \delta(\Pi_1, \Pi'_1)$*.*

(xix) $[\wedge I/\wedge E_2]$

If $\quad \begin{matrix}\Pi_1\\ b:A\end{matrix} \rightarrowtail_1 \begin{matrix}\Pi'_1\\ b:A\end{matrix}$ *and* $\begin{matrix}\Pi_2\\ b:B\end{matrix} \rightarrowtail_1 \begin{matrix}\Pi'_2\\ b:B\end{matrix}$,

then $\quad \Pi = \dfrac{\begin{matrix}\Pi_1\\ b:A\end{matrix} \quad \begin{matrix}\Pi_2\\ b:B\end{matrix}}{\dfrac{b:A \wedge B}{b:B}\wedge E_2}\wedge I \quad \rightarrowtail_1 \quad \Pi' = \begin{matrix}\Pi'_2\\ b:B\end{matrix}$

and $\delta(\Pi, \Pi') = \delta(\Pi_2, \Pi'_2)$*.*

(xx) $[\mathsf{X}I/\mathsf{X}E]$

If $\quad \begin{matrix}b_1 \lhd b_2\\ \Pi_1\\ b_2:A\end{matrix} \rightarrowtail_1 \begin{matrix}b_1 \lhd b_2\\ \Pi'_1\\ b_2:A\end{matrix}$,

then $\quad \Pi = \dfrac{\dfrac{\begin{matrix}[b_1 \lhd b_2]\\ \Pi_1\\ b_2:A\end{matrix}}{b_1:\mathsf{X}A}\mathsf{X}I \quad b_1 \lhd b}{b:A}\mathsf{X}E \quad \rightarrowtail_1 \quad \Pi' = \begin{matrix}b_1 \lhd b\\ \Pi'_1[b/b_2]\\ b:A\end{matrix}$

where if there are marked occurrences of $b_1 \lhd b$ *in* Π_1*, then their corresponding occurrences in* Π'_1 *can be unmarked and* $\delta(\Pi, \Pi') = \delta(\Pi_0, \Pi'_0) \cup$

$\delta(\Pi_1, \Pi_1') \setminus \{s \in \Sigma \mid$ *all the rwffs marked with* s*, if any, are discharged in* $\Pi'\}$.

(xxi) $[\mathsf{G}I/\mathsf{G}E]$

$$\textit{If} \quad \begin{array}{c} b_1 \leqslant b_2 \\ \Pi_1 \\ b_2 : A \end{array} \rightarrowtail_1 \begin{array}{c} b_1 \leqslant b_2 \\ \Pi_1' \\ b_2 : A \end{array} ,$$

$$\textit{then} \quad \Pi = \dfrac{\dfrac{\begin{array}{c}[b_1 \leqslant b_2] \\ \Pi_1 \\ b_2 : A \end{array}}{b_1 : \mathsf{G}A}\ \mathsf{G}I \qquad b_1 \leqslant b}{b : A}\ \mathsf{G}E \quad \rightarrowtail_1 \quad \Pi' = \begin{array}{c} b_1 \leqslant b \\ \Pi_1'[b/b_2] \\ b : A \end{array}$$

where if there are marked occurrences of $b_1 \leqslant b$ *in* Π_1*, then their corresponding occurrences in* Π_1' *can be unmarked and* $\delta(\Pi, \Pi') = \delta(\Pi_0, \Pi_0') \cup \delta(\Pi_1, \Pi_1') \setminus \{s \in \Sigma \mid$ *all the rwffs marked with* s*, if any, are discharged in* $\Pi'\}$.

$(xxii)$ $[\forall I/\forall E]$

$$\textit{If} \quad \begin{array}{c} b_1 \bullet b_2 \\ \Pi_1 \\ b_2 : A \end{array} \rightarrowtail_1 \begin{array}{c} b_1 \bullet b_2 \\ \Pi_1' \\ b_2 : A \end{array} ,$$

$$\textit{then} \quad \Pi = \dfrac{\dfrac{\begin{array}{c}[b_1 \bullet b_2] \\ \Pi_1 \\ b_2 : A \end{array}}{b_1 : \forall A}\ \forall I \qquad b_1 \bullet b}{b : A}\ \forall E \quad \rightarrowtail_1 \quad \Pi' = \begin{array}{c} b_1 \bullet b \\ \Pi_1'[b/b_2] \\ b : A \end{array}$$

and $\delta(\Pi, \Pi') = \delta(\Pi_1, \Pi_1')$.

We illustrate the mechanism of marking by means of an example. Let Π be the following derivation:

$$\Pi = \dfrac{\dfrac{b_0 \lhd b_1 \qquad \dfrac{b_0 : A \quad [b_0 \leqslant b_1]^2}{b_1 : A}\ \textit{base}\leqslant^2 (r)}{\dfrac{b_1 : A}{b_1 : A}\ \textit{refl}\leqslant^1 (q)} \qquad \dfrac{[b_0 \leqslant b_1]^3 \quad [b_i : A]^3 \quad [b_i \lhd b_j]^3}{\begin{array}{c}\Pi_1 \\ b_j : A\end{array}}\ \textit{ind}^3(s)$$

with $\Pi_1 \rightarrowtail_1 \Pi_1'$ and $\delta(\Pi_1, \Pi_1') = \emptyset$ for some Π_1'. Then, by Definition 5.32, we have:

$$\Pi_2 = \dfrac{b_0 : A \quad b_0 \leqslant b_1 \qquad \dfrac{[b_0 \leqslant b_i]^3 \quad [b_i : A]^3 \quad [b_i \lhd b_j]^3}{\begin{array}{c}\Pi_1 \\ b_j : A\end{array}}\ \textit{ind}^3(s)}{b_1 : A}$$

$$\rightarrowtail_1$$

$$\Pi'_2 = \begin{array}{c} b_0 \leqslant b_{0(s)} \quad b_0 : A \quad b_0 \lhd b_{1(s)} \\ \Pi'_1[b_0/b_i][b_1/b_j] \\ b_1 : A \end{array},$$

with $\delta(\Pi_2, \Pi'_2) = \{s\}$ and consequently:

$$\Pi = \dfrac{\dfrac{\dfrac{\Pi_2}{b_0 \lhd b_1 \quad b_1 : A}\ base \leqslant (r)}{b_1 : A}\ refl \leqslant^1 (q)}{b_1 : A}$$

$$\rightarrowtail_1$$

$$\Pi' = \dfrac{\dfrac{\dfrac{[b_0 \leqslant b_{0(s)}]^1 \quad b_0 : A \quad b_0 \lhd b_1}{b_0 \lhd b_1 \qquad\qquad b_1 : A}\ base \leqslant (r)}{b_1 : A}\ refl \leqslant^1 (q)}{b_1 : A}$$

with $\delta(\Pi, \Pi') = \emptyset$. Now we can use the relation $\rightarrowtail_1$ to define the 1-reduction $\Rightarrow_1$. Namely, $\Rightarrow_1$ contains those pairs in $\rightarrowtail_1$ whose set of defects is empty. Note that in this case we give the definition directly for unmarked derivations.

Definition 5.33. *We define the* 1-reduction *relation (denoted by $\Rightarrow_1$) between $\mathcal{N}(BCTL^*_{-i})$ derivations Π and Π' as follows: $\Pi \Rightarrow_1 \Pi'$ iff for every standard marking l for Π, there exists a marking l' for Π' such that $(\Pi, l) \rightarrowtail_1 (\Pi', l')$ and $\delta((\Pi, l), (\Pi', l')) = \emptyset$.*

*By extension, we define the n-*reduction *(denoted by $\Rightarrow_n$) inductively as follows:*

(i) $\Pi \Rightarrow_0 \Pi$;
(ii) if $\Pi \Rightarrow_n \Pi'$ and $\Pi' \Rightarrow_1 \Pi''$ then $\Pi \Rightarrow_{n+1} \Pi''$.

Lemma 5.34. *Let Π and Π' be two derivations. $\Pi \Rightarrow \Pi'$ if and only if there exists a positive integer n such that $\Pi \Rightarrow_n \Pi'$.*

Proof. Immediate, by observing that the contractions on which $\Rightarrow$ is based can be "reproduced" by $\rightarrowtail_1$-reductions with no defects. Concerning the other direction, every $\rightarrowtail_1$-reduction without any defect corresponds to the application of one or more contractions of Section 5.4.3.

□

A result of confluence holds for $\Rightarrow_1$; the details of the proof are in Appendix A.1.

Lemma 5.35. *Let Π, Π' and Π'' be derivations. If $\Pi \Rightarrow_1 \Pi'$ and $\Pi \Rightarrow_1 \Pi''$, then there exists a derivation Π''' such that $\Pi' \Rightarrow_1 \Pi'''$ and $\Pi'' \Rightarrow_1 \Pi'''$.*

Theorem 5.36. *Let Π, Π' and Π'' be $\mathcal{N}(BCTL^*_{-i})$ derivations. If $\Pi \Rightarrow \Pi'$ and $\Pi \Rightarrow \Pi''$, then there exists a derivation Π''' such that $\Pi' \Rightarrow \Pi'''$ and $\Pi'' \Rightarrow \Pi'''$.*

Proof. By Lemma 5.34, there exist two sequences of 1-reductions $\Pi_{00} \Rightarrow_1 \Pi_{01} \ldots \Rightarrow_1 \Pi_{0n}$ and $\Pi_{00} \Rightarrow_1 \Pi_{10} \ldots \Rightarrow_1 \Pi_{m0}$ such that $\Pi_{00} = \Pi$, $\Pi_{0n} = \Pi'$ and $\Pi_{m0} = \Pi''$. Repeated applications of Lemma 5.35 let us build an $(n \times m)$-grid of derivations (see Fig. 5.6), where for each $0 \leq i < n$ and $0 \leq j < m$, there exists a derivation $\Pi_{(i+1)(j+1)}$ such that $\Pi_{i(j+1)} \Rightarrow_1 \Pi_{(i+1)(j+1)}$ and $\Pi_{(i+1)j} \Rightarrow_1 \Pi_{(i+1)(j+1)}$. We can now take $\Pi''' = \Pi_{mn}$. By Lemma 5.34, we conclude $\Pi' \Rightarrow \Pi'''$ and $\Pi'' \Rightarrow \Pi'''$.

□

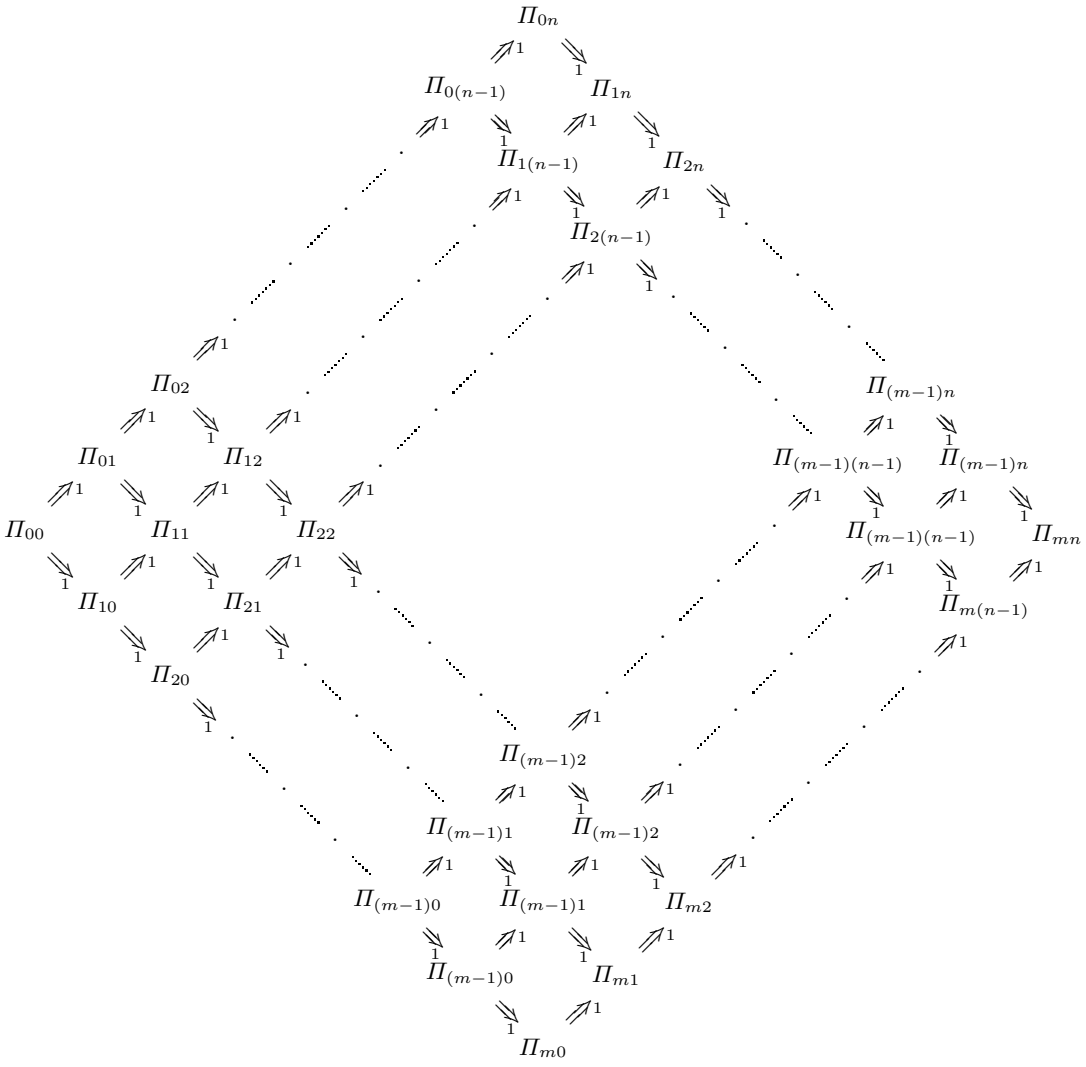


Fig. 5.6. The Church-Rosser theorem.

5.4.5 The normalization theorem

We are now in a position to establish a normalization theorem with regard to the definition of normal form given in Definition 5.23 and to the reduction $\Rightarrow$ based on proper and induction contractions.

Similar to what Girard noted in [74] with regard to his natural deduction system for Heyting arithmetic, also for $\mathcal{N}(BCTL^*_{-i})$ it is not possible to give a proof of normalization by induction on the complexity of the maximum formulas to be removed. As an example, consider the following contraction:

$$\dfrac{\dfrac{\begin{array}{c}[b_1 \leqslant b']^1 \\ \Pi \\ b' : A\end{array}}{b_1 : \mathsf{G}A}\ \mathsf{G}I^1 \qquad b_1 \leqslant b_2}{b_2 : A}\ \mathsf{G}E \quad \rightsquigarrow \quad \begin{array}{c} b_1 \leqslant b_2 \\ \Pi[b_2/b'] \\ b_2 : A \end{array} \quad .$$

If Π contains an *ind*-application r whose ending label is b', then in $\Pi[b_2/b']$, r will have b_2 as ending label. But, unlike b', b_2 is not required to be a proper parameter and thus the application r, which is not unfoldable in Π, could be unfoldable in $\Pi[b_2/b']$. This is an example of a contraction that can give rise to a new maximum formula, about whose complexity we cannot say anything.

Similar to [74], we thus introduce a notion of reducibility for derivations in $\mathcal{N}(BCTL^*_{-i})$. The general schema of the proof of normalization will then consist in showing that:

(i) every $\mathcal{N}(BCTL^*_{-i})$ derivation is reducible (Corollary 5.44); and
(ii) every reducible derivation reduces to a normal form (Theorem 5.41, property $Red1$).

From (i) and (ii), it trivially follows that every $\mathcal{N}(BCTL^*_{-i})$ derivation has a normal form, which is what we wish to prove.

With respect to the case of systems for Heyting arithmetic, a further problem that we have to face here comes again from dealing with *ind*-applications whose contractions are not strictly local. As we did for the Church-Rosser theorem, the solution will consist in proving the statement with regard to a larger class of reductions, according to which we are allowed to unfold an *ind*-application into a subderivation (of the form specified by the induction contractions of Section 5.4.3) whose length and whose set of labels are arbitrary. This idea will be formalized inside the notion of reducibility under substitution (Definition 5.42): in fact, instead of proving directly (i), we will introduce this stronger notion of reducibility and prove that every $\mathcal{N}(BCTL^*_{-i})$ derivation is actually reducible under substitution (Theorem 5.43).

A number of auxiliary lemmas will be used along the proof. In particular, in order to prove (i), we will need to show that the notion of reducibility is preserved by the addition of applications with atomic conclusions (Lemma 5.38), and that it is strictly connected to the relation $\Rightarrow$, namely if $\Pi \Rightarrow \Pi'$ then Π and Π' are either both reducible or both non-reducible (Lemma 5.39). In Definition 5.40, we also introduce another characterization of $\mathcal{N}(BCTL^*_{-i})$ derivations: the set of $\mathcal{S}$-derivations. We will show that this set is contained in the set of reducible

derivations (Theorem 5.41, property *Red*2) and this characterization will turn out to be useful in the case of *ind*-applications.

Finally, both in proving (*i*) and (*ii*) we will use the Church-Rosser property shown in Theorem 5.36.

Definition 5.37. *Let Π be a derivation of $b : A$. We define the notion of* reducibility *by induction on the complexity of A as follows:*

(*i*) *if A is an atomic formula, then Π is reducible iff it reduces to a normal form;*
(*ii*) *if A is $A_1 \supset A_2$, then Π is reducible iff for all reducible derivations Π_1 of $b : A_1$, the derivation*

$$\dfrac{\begin{matrix}\Pi & \Pi_1 \\ b : A_1 \supset A_2 & b : A_1\end{matrix}}{b : A_2}\ \supset E$$

is reducible;
(*iii*) *if A is $A_1 \wedge A_2$, then Π is reducible iff*

$$\dfrac{\begin{matrix}\Pi \\ b : A_1 \wedge A_2\end{matrix}}{b : A_1}\ \wedge E_1 \qquad \textit{and} \qquad \dfrac{\begin{matrix}\Pi \\ b : A_1 \wedge A_2\end{matrix}}{b : A_2}\ \wedge E_2$$

are reducible;
(*iv*) *if A is $\mathsf{X}B$, then Π is reducible iff for each label b' the derivation*

$$\dfrac{\begin{matrix}\Pi \\ b : \mathsf{X}B \quad b \lhd b'\end{matrix}}{b' : B}\ \mathsf{X}E$$

is reducible;
(*v*) *if A is $\mathsf{G}B$, then Π is reducible iff for each label b' the derivation*

$$\dfrac{\begin{matrix}\Pi \\ b : \mathsf{G}B \quad b \leqslant b'\end{matrix}}{b' : B}\ \mathsf{G}E$$

is reducible;
(*vi*) *if A is $\forall B$, then Π is reducible iff for each label b' the derivation*

$$\dfrac{\begin{matrix}\Pi \\ b : \forall B \quad b \bullet b'\end{matrix}}{b' : B}\ \forall E$$

is reducible.

We begin our proof by showing a useful lemma:

Lemma 5.38. *Applications of ser$\lhd$, lin$\lhd$, refl $\leqslant$, trans $\leqslant$, refl$\bullet$, symm$\bullet$, trans$\bullet$, atom$\bullet$, fusion and $\bot E_i$ preserve reducibility.*

Proof. We use the facts that the conclusions of such rules are atomic and that they cannot introduce any maximum formulas. As an example (the other cases are analogous), let us consider a derivation Π whose last application is a *ser*$\lhd$:

$$\dfrac{\begin{array}{c}[b_1 \lhd b_2]\\ \Pi_1\\ b : p\end{array}}{b : p}\ ser{\lhd}\quad .$$

For derivations of atomic formulas, by Definition 5.37, reducibility coincides with normalizability. We conclude by noticing that if Π_1 has a normal form Π_1', then Π has the following derivation as a normal form:

$$\dfrac{\begin{array}{c}[b_1 \lhd b_2]\\ \Pi_1'\\ b : p\end{array}}{b : p}\ ser{\lhd}\quad .$$

□

In Lemma 5.38, we do not consider applications of the relational rules $lin{\lhd_{\mathcal{R}}}$ and *base* $\leqslant$, which can introduce maximum formulas and have to be treated differently.

Lemma 5.39. *Let Π and Π' be $\mathcal{N}(BCTL^*_{-i})$ derivations. If $\Pi \Rightarrow \Pi'$ then Π is reducible iff Π' is reducible.*

Proof. By induction on the complexity of the conclusion $b : A$ of Π.

(*i*) $b : A$ is atomic.
For derivations of atomic formulas, reducibility coincides with normalizability. *(Left-to-right implication)* If Π has a normal form Π'' then by Theorem 5.36 there exists a Π''' to which both Π' and Π'' reduce. Since Π'' is normal, Π'' and Π''' must coincide, i.e. also Π' has a normal form. *(Right-to-left implication)* If $\Pi \Rightarrow \Pi'$ and Π' has a normal form Π'', then $\Pi \Rightarrow \Pi''$.

(*ii*) $b : A$ is $b : A_1 \supset A_2$.
Assume $\Pi \Rightarrow \Pi'$ and consider the derivations

$$\Pi_1 = \dfrac{\begin{array}{cc}\Pi & \widehat{\Pi}\\ b : A_1 \supset A_2 & b : A_1\end{array}}{b : A_2}\ {\supset}E \qquad \text{and} \qquad \Pi_2 = \dfrac{\begin{array}{cc}\Pi' & \widehat{\Pi}\\ b : A_1 \supset A_2 & b : A_1\end{array}}{b : A_2}\ {\supset}E$$

where $\widehat{\Pi}$ is some reducible derivation. We prove both directions simultaneously. By definition of reducibility, Π is reducible iff Π_1 is reducible. But $\Pi_1 \Rightarrow \Pi_2$ and thus, by the induction hypothesis, Π_1 is reducible iff Π_2 is reducible. Finally, by definition of reducibility, Π_2 is reducible iff Π' is reducible.

(*iii*) $b : A$ is $b : A_1 \wedge A_2$.
Assume $\Pi \Rightarrow \Pi'$ and consider the derivations

$$\Pi_1 = \dfrac{\begin{array}{c}\Pi\\ b : A_1 \wedge A_2\end{array}}{b : A_1}\ {\wedge}E_1 \qquad \text{and} \qquad \Pi_1' = \dfrac{\begin{array}{c}\Pi'\\ b : A_1 \wedge A_2\end{array}}{b : A_1}\ {\wedge}E_1 \quad .$$

We prove both directions simultaneously. By definition of reducibility, Π is reducible iff Π_1 is reducible. But $\Pi_1 \Rightarrow \Pi_1'$. It follows that, by the induction hypothesis, Π_1 is reducible iff Π_1' is reducible. By definition of reducibility, Π_1' is reducible iff Π' is reducible. We proceed similarly for ${\wedge}E_2$.

(*iv*) $b : A$ is $b : \mathsf{X}B$ or $b : \mathsf{G}B$ or $b : \forall B$.
We consider the case $b : \mathsf{X}B$; the other ones are analogous. Assume $\Pi \Rightarrow \Pi'$ and consider the derivations

$$\Pi_1 = \dfrac{\overset{\Pi}{b : \mathsf{X}B} \quad b \lhd b'}{b' : B}\ \mathsf{X}E \qquad \text{and} \qquad \Pi_2 = \dfrac{\overset{\Pi'}{b : \mathsf{X}B} \quad b \lhd b'}{b' : B}\ \mathsf{X}E$$

for some label b'. By definition of reducibility, Π is reducible iff Π_1 is reducible. But $\Pi_1 \Rightarrow \Pi_2$ and thus, by the induction hypothesis, Π_1 is reducible iff Π_2 is reducible. Again, by definition of reducibility, Π_2 is reducible iff Π' is reducible.

□

As in [74], we also define a subset of reducible derivations, which will be useful in the following.

Definition 5.40. *We say that a* $\mathcal{N}(BCTL^*_{\_i})$ *derivation is* $\mathcal{S}$-reducible *if it belongs to the set* $\mathcal{S}$ *defined inductively as follows:*

(1) *A derivation consisting of just an assumption is in* $\mathcal{S}$.
(2) *If a derivation* Π *of* $b : A \supset B$ *is in* $\mathcal{S}$ *and if a derivation* Π' *of* $b : A$ *has a normal form, then the derivation*

$$\dfrac{\overset{\Pi}{b : A \supset B} \quad \overset{\Pi'}{b : A}}{b : B}\ \supset E$$

is in $\mathcal{S}$;
(3) *If a derivation* Π *of* $b : A \wedge B$ *is in* $\mathcal{S}$, *then*

$$\dfrac{\overset{\Pi}{b : A \wedge B}}{b : A}\ \wedge E_1 \qquad \textit{and} \qquad \dfrac{\overset{\Pi}{b : A \wedge B}}{b : B}\ \wedge E_2$$

are in $\mathcal{S}$;
(4) *If a derivation* Π *of* $b : \mathsf{X}A$ *is in* $\mathcal{S}$, *then the derivation*

$$\dfrac{\overset{\Pi}{b : \mathsf{X}A} \quad b \lhd b'}{b' : A}\ \mathsf{X}E$$

is in $\mathcal{S}$ *for every label* b';
(5) *If a derivation* Π *of* $b : \mathsf{G}A$ *is in* $\mathcal{S}$, *then the derivation*

$$\dfrac{\overset{\Pi}{b : \mathsf{G}A} \quad b \leqslant b'}{b' : A}\ \mathsf{G}E$$

is in $\mathcal{S}$ *for every label* b';
(6) *If a derivation* Π *of* $b : \forall A$ *is in* $\mathcal{S}$, *then the derivation*

$$\dfrac{\overset{\Pi}{b : \forall A} \quad b \bullet b'}{b' : A}\ \forall E$$

is in $\mathcal{S}$ *for every label* b';

(7) *If the derivations*

$$\begin{array}{c} \Pi_0 \\ b_0 : A \end{array} \qquad \textit{and} \qquad \begin{array}{c} \\ \Pi_1 \\ b_j : A \end{array} \qquad \begin{array}{c} b_0 \leqslant b_i \quad b_i : A \quad b_i \lhd b_j \\ \\ \end{array}$$

have normal forms, then the derivation

$$\dfrac{\begin{array}{c} \Pi_0 \\ b_0 : A \end{array} \quad b_0 \leqslant b_n \qquad \begin{array}{c} [b_0 \leqslant b_i] \quad [b_i : A] \quad [b_i \lhd b_j] \\ \Pi_1 \\ b_j : A \end{array}}{b_n : A}\ \textit{ind}$$

is in $\mathcal{S}$.

(8) *No other derivation belongs to* $\mathcal{S}$.

We prove now some properties of reducible and $\mathcal{S}$-reducible derivations that will be used in the subsequent proofs.

Theorem 5.41. *Reducible derivations enjoy the following properties:*

(*Red*1) *If* Π *is reducible, then* Π *reduces to a normal form.*
(*Red*2) *If* Π *is* $\mathcal{S}$*-reducible, then* Π *is reducible.*

Proof. We proceed by induction on the complexity of the conclusion $b : A$ of Π.

(i) $b : A$ is atomic
- (*Red*1) By definition of reducibility.
- (*Red*2) Let us consider the inductive definition of $\mathcal{S}$-reducibility. If in each application of step (2) we replace Π' by its normal form, and in each application of (7) we replace Π_0 and Π_1 by their normal forms, then it is clear that all the $\mathcal{S}$-reducible derivations are normalizable.

(ii) $b : A$ is $b : A_1 \supset A_2$
- (*Red*1) Let Π'_0 be the following derivation:

$$\dfrac{\begin{array}{c}\Pi \\ b : A_1 \supset A_2\end{array} \quad b : A_1}{b : A_2}\ \supset E \quad .$$

By the induction hypothesis, there exists a reduction sequence $\Pi'_0, \ldots, \Pi'_n$ such that Π'_n is normal. We have two cases:

(a) If all the contractions in the reduction sequence are applied on strict subproofs, then we have that Π'_i is

$$\dfrac{\begin{array}{c}\Pi_i \\ b : A_1 \supset A_2\end{array} \quad b : A_1}{b : A_2}\ \supset E \quad ,$$

for each $0 \leq i < n$, and we can write $\Pi \Rightarrow \Pi_n$, where Π_n is normal.

(b) Otherwise, we can choose the minimum i such that the contraction is not made on a strict subderivation of Π'_i:

$$\Pi_i \;=\; \dfrac{\dfrac{\begin{array}{c}[b:A_1]\\ \widehat{\Pi}\\ b:A_2\end{array}}{b:A_1\supset A_2}\supset I \qquad b:A_1}{b:A_2}\supset E \quad .$$

Then $\Pi'_0 \Rightarrow \widehat{\Pi}$ and, by Theorem 5.36, $\widehat{\Pi} \Rightarrow \Pi'_n$; hence:

$$\Pi \;\Rightarrow\; \dfrac{\begin{array}{c}[b:A_1]\\ \Pi'_n\\ b:A_2\end{array}}{b:A_1\supset A_2}\supset I \quad ,$$

which is normal.

($Red2$) Assume that Π is $\mathcal{S}$-reducible and consider the following derivation:

$$\Pi' \;=\; \dfrac{\begin{array}{cc}\Pi & \widehat{\Pi}\\ b:A_1\supset A_2 & b:A_1\end{array}}{b:A_2}\supset E \quad ,$$

where $\widehat{\Pi}$ is reducible. By the induction hypothesis on ($Red1$), $\widehat{\Pi}$ is normalizable. Then Π' also is $\mathcal{S}$-reducible and, by the induction hypothesis, reducible. By definition of reducibility, we conclude that Π is reducible.

(iii) $b:A$ is $b:A_1 \wedge A_2$

($Red1$) Let Π'_0 and Π''_0 be the following derivations:

$$\Pi'_0 \;=\; \dfrac{\begin{array}{c}\Pi\\ b:A_1\wedge A_2\end{array}}{b:A_1}\wedge E_1 \qquad , \qquad \Pi''_0 \;=\; \dfrac{\begin{array}{c}\Pi\\ b:A_1\wedge A_2\end{array}}{b:A_2}\wedge E_2 \quad .$$

By the induction hypothesis, there exist two reduction sequences $\Pi'_0, \ldots, \Pi'_n$ and $\Pi''_0, \ldots, \Pi''_m$ such that Π'_n and Π''_m are normal. We have two cases:

(a) If all the contractions in the reduction sequence $\Pi'_0, \ldots, \Pi'_n$ are applied on strict subproofs, then we have that Π'_i can be written as

$$\dfrac{\begin{array}{c}\Pi_i\\ b:A_1\wedge A_2\end{array}}{b:A_1}\wedge E_1 \quad ,$$

for each $0 \leq i < n$, and we can conclude $\Pi \Rightarrow \Pi_n$, where Π_n is normal.

(b) Otherwise, we can choose the minimum i such that the contraction is not applied to a strict subderivation of Π'_i:

$$\Pi'_i \;=\; \dfrac{\dfrac{\begin{array}{cc}\widehat{\Pi_1} & \widehat{\Pi_2}\\ b:A_1 & b:A_2\end{array}}{b:A_1\wedge A_2}\wedge I}{b:A_1}\wedge E_1 \quad .$$

Then $\Pi'_0 \Rightarrow \widehat{\Pi_1}$ and $\Pi''_0 \Rightarrow \widehat{\Pi_2}$. By Theorem 5.36, $\widehat{\Pi_1} \Rightarrow \Pi'_n$ and $\widehat{\Pi_2} \Rightarrow \Pi''_m$; hence:

$$\Pi \;\Rightarrow\; \dfrac{\begin{array}{cc}\Pi'_n & \Pi''_m \\ b : A_1 & b : A_2\end{array}}{b : A_1 \wedge A_2}\ \wedge I \quad ,$$

which is normal.

($Red2$) Assume that Π is $\mathcal{S}$-reducible and consider the following derivations:

$$\Pi' \;=\; \dfrac{\begin{array}{c}\Pi \\ b : A_1 \wedge A_2\end{array}}{b : A_1}\ \wedge E_1 \qquad \text{and} \qquad \Pi'' \;=\; \dfrac{\begin{array}{c}\Pi \\ b : A_1 \wedge A_2\end{array}}{b : A_2}\ \wedge E_2 \quad .$$

By Definition 5.40, Π' and Π'' are $\mathcal{S}$-reducible and thus, by the induction hypothesis, reducible. Then, by definition of reducibility, we conclude that Π is reducible.

(iv) $b : A$ is $b : \mathsf{X}B$

($Red1$) Assume that Π is reducible. Then, by definition of reducibility, there exists a reducible derivation Π'_0 such as:

$$\dfrac{\begin{array}{cc}\Pi & \\ b : \mathsf{X}B & b \lhd b'\end{array}}{b' : B}\ \mathsf{X}E \quad ,$$

for some label b'. By the induction hypothesis on Π'_0, there exists a reduction sequence $\Pi'_0, \ldots, \Pi'_n$ such that Π'_n is normal. We have two cases:

(a) If all the contractions in the reduction sequence are made on strict subderivations, then we have that Π'_i is

$$\dfrac{\begin{array}{cc}\Pi_i & \\ b : \mathsf{X}B & b \lhd b'\end{array}}{b' : B}\ \mathsf{X}E \quad ,$$

for each $0 \leq i < n$, and we can write $\Pi \Rightarrow \Pi_n$, where Π_n is clearly normal.

(b) Otherwise, there exists a minimum i such that the contraction is not made on a strict subderivation of Π'_i:

$$\Pi'_i \;=\; \dfrac{\dfrac{\begin{array}{c}[b \lhd b'']^1 \\ \widehat{\Pi} \\ b'' : B\end{array}}{b : \mathsf{X}B}\ \mathsf{X}I^1 \qquad b \lhd b'}{b' : B}\ \mathsf{X}E \quad ,$$

for some b'' fresh in $\widehat{\Pi}$. But $\Pi'_i \rightsquigarrow \widehat{\Pi}[b'/b'']$ and thus we have $\Pi'_0 \Rightarrow \widehat{\Pi}[b'/b'']$. We know that

$$\Pi \;\Rightarrow\; \dfrac{\begin{array}{c}[b \lhd b'']^1 \\ \widehat{\Pi} \\ b'' : B\end{array}}{b : \mathsf{X}B}\ \mathsf{X}I^1 \quad .$$

By Theorem 5.36, $\widehat{\Pi}[b'/b''] \Rightarrow \Pi'_n$, which is normal. By the freshness of b'' in $\widehat{\Pi}$, we have that if Π'_n is a normal form for $\widehat{\Pi}[b'/b'']$ then $\Pi'_n[b''/b']$ is a normal form for $\widehat{\Pi}$. Thus we have:

$$\Pi \;\Rightarrow\; \dfrac{\dfrac{\Pi'_n[b''/b']}{b'' : B}}{b : \mathsf{X}B}\,\mathsf{X}I \quad ,$$

which is normal.

($Red2$) Assume that Π is $\mathcal{S}$-reducible and consider the derivation:

$$\Pi' \;=\; \dfrac{\dfrac{\Pi}{b : \mathsf{X}B} \quad b \lhd b'}{b' : B}\,\mathsf{X}E \quad ,$$

for some label b'. By definition of $\mathcal{S}$-reducibility, if Π is $\mathcal{S}$-reducible, then also Π' is $\mathcal{S}$-reducible. But then, by the induction hypothesis, Π' is reducible. We conclude that Π is reducible by definition of reducibility.

(v) $b : A$ is $b : \mathsf{G}B$

(vi) $b : A$ is $b : \forall B$

Proofs for the cases (v) and (vi) are analogous to those for the case (iv).

□

Now we introduce the stronger notion of reducibility under substitution and show that every derivation is in fact reducible under substitution.

Definition 5.42. *A derivation Π is* reducible under substitution *if:*

(i) for each substitution of labels that are not proper parameters;

(ii) for each replacement of open hypotheses by reducible derivations of such hypotheses; and

(iii) for each replacement of a subderivation of Π, whose last application is an ind-application s, like the following

$$\dfrac{\begin{array}{cc}\Pi_0 & \\ b_0 : A \quad b_0 \leqslant b & \end{array} \qquad \begin{array}{c}[b_0 \leqslant b_1]^1\,[b_i : A]^1\,[b_i \lhd b_j]^1 \\ \Pi_1 \\ b_j : A\end{array}}{b : A}\,s^1$$

by a subderivation like the following (for each n and for every choice of labels $b_1, \ldots, b_{n-1}$)

$$\begin{array}{c}\Pi_0 \\ b_0 \leqslant b_0 \qquad b_0 : A \qquad b_0 \lhd b_1 \\ \Pi_1[b_0/b_i][b_1/b_j] \\ b_0 \leqslant b_1 \qquad b_1 : A \qquad b_1 \lhd b_2 \\ \Pi_1[b_1/b_i][b_2/b_j] \\ b_2 : A \\ \vdots \\ b_0 \leqslant b_{n-1} \qquad b_{n-1} : A \qquad b_{n-1} \lhd b \\ \Pi_1[b_{n-1}/b_i][b/b_j] \\ b : A\end{array}$$

the resulting derivation is reducible.

We say that a derivation Π^* is obtained by substitution *from Π if Π^* is obtained from Π by applying zero or more substitutions and/or replacements as specified by the items (i), (ii) and (iii).*

Lemma 5.43. *Every $\mathcal{N}(BCTL^*_{-i})$ derivation is reducible under substitution.*

Proof. The proof proceeds by induction on the length of the derivation Π. If Π is just an assumption, then it is clearly reducible under substitution. As an inductive step, we have a case for every possible rule.

$(\supset I)$
Let $\supset I$ be the last rule applied in Π:

$$\dfrac{\begin{array}{c}[b:A]\\ \Pi_1\\ b:B\end{array}}{b:A\supset B}\ \supset I \quad .$$

By the induction hypothesis, Π_1 is reducible under substitution. Without loss of generality, we consider now a derivation Π^* obtained by substitution from Π and prove that it is reducible. Π^* will have the form:

$$\dfrac{\begin{array}{c}[b^*:A]\\ \Pi_1^*\\ b^*:B\end{array}}{b^*:A\supset B}\ \supset I \quad .$$

Note that Π_1^* is obtained by substitution from Π_1. By Definition 5.37, we need to show that for all reducible derivations Π' of $b^*:A$, the derivation

$$\dfrac{\Pi' \quad \dfrac{\begin{array}{c}[b^*:A]\\ \Pi_1^*\\ b^*:B\end{array}}{b^*:A\supset B}\ \supset I}{b^*:B}\ \supset E$$

is reducible. But it is enough to notice that this derivation reduces to

$$\begin{array}{c}\Pi'\\ b^*:A\\ \Pi_1^*\\ b^*:B\end{array} \quad ,$$

which is reducible as it is obtained by substitution from Π_1 (that is reducible under substitution by the induction hypothesis). By Lemma 5.39, we have the thesis.

$(\wedge I)$
Let $\wedge I$ be the last rule applied in Π:

$$\dfrac{\Pi_1 \quad \Pi_2}{b:A \quad b:B}\ \wedge I \quad .$$

By the induction hypothesis, Π_1 and Π_2 are reducible under substitution. Without loss of generality, we consider now a derivation Π^* obtained by substitution from Π and prove that it is reducible. Π^* will have the form:

$$\dfrac{\begin{array}{cc}\Pi_1^* & \Pi_2^* \\ b^* : A & b^* : B\end{array}}{b^* : A \wedge B}\ \wedge I \quad .$$

Note that Π_1^* and Π_2^* are obtained by substitution from Π_1 and Π_2^* respectively. By Definition 5.37, we need to show that the derivations

$$\dfrac{\dfrac{\begin{array}{cc}\Pi_1^* & \Pi_2^* \\ b^* : A & b^* : B\end{array}}{b^* : A \wedge B}\ \wedge I}{b^* : A}\ \wedge E_1 \qquad \text{and} \qquad \dfrac{\dfrac{\begin{array}{cc}\Pi_1^* & \Pi_2^* \\ b^* : A & b^* : B\end{array}}{b^* : A \wedge B}\ \wedge I}{b^* : B}\ \wedge E_2$$

are reducible. But it is enough to notice that they reduce to

$$\begin{array}{c}\Pi_1^* \\ b^* : A\end{array} \qquad \text{and} \qquad \begin{array}{c}\Pi_2^* \\ b^* : B\end{array} \quad ,$$

which are reducible as they are obtained by substitution from Π_1 and Π_2, respectively (and Π_1 and Π_2 are reducible under substitution by the induction hypothesis). By Lemma 5.39, we obtain the thesis.

$(\mathsf{X}I)$, $(\mathsf{G}I)$, $(\forall I)$

We consider here the case of $\mathsf{X}I$; the other cases are analogous. Let $\mathsf{X}I$ be the last rule applied in Π:

$$\dfrac{\begin{array}{c}[b \lhd b_1] \\ \Pi_1 \\ b_1 : A\end{array}}{b : \mathsf{X}A}\ \mathsf{X}I \quad .$$

We consider now a generic Π^* obtained by substitution from Π and prove that it is reducible. By Definition 5.37, we need to show that for each label b' the derivation

$$\dfrac{\dfrac{\begin{array}{c}[b^* \lhd b_1] \\ \Pi_1^* \\ b_1 : A\end{array}}{b^* : \mathsf{X}A}\ \mathsf{X}I \qquad b^* \lhd b'}{b' : A}\ \mathsf{X}E$$

is reducible. But this derivation reduces to

$$\begin{array}{c}b^* \lhd b' \\ \Pi_1^*[b'/b_1] \\ b' : A\end{array} \quad ,$$

which is reducible as it is obtained by substitution from Π_1, which is reducible under substitution by the induction hypothesis. (Note that b_1 is a proper parameter in Π but not in Π_1.) By Lemma 5.39, we have the thesis.

$(\supset E)$, $(\wedge E_1)$, $(\wedge E_2)$, $(\mathsf{X}E)$, $(\mathsf{G}E)$, $(\forall E)$

The definition of reducibility is given in such a way that elimination rules clearly

preserve reducibility. Since elimination rules do not introduce proper parameters, do not close any assumption and do not solve any *ind*-application, the set of possible substitutions on Π is exactly the same as in the subderivations obtained from Π by removing the last rule application. Such subderivations are reducible under substitution by the induction hypothesis. Thus we have the thesis.

($ser\lhd$), ($lin\lhd$), ($refl\leqslant$), ($trans\leqslant$), ($refl\bullet$), ($symm\bullet$), ($trans\bullet$), ($atom\bullet$), (*fusion*), ($\bot E_i$)

As in the previous case, these rules do not introduce proper parameters, do not close any assumption and do not solve any *ind*-application. Thus the thesis follows directly from Lemma 5.38.

(*ind*)

Let *ind* be the last rule applied in Π:

$$\dfrac{\begin{array}{ccc}\Pi_0 & & \begin{array}{c}[b_0 \leqslant b_i]\ [b_i : A]\ [b_i \lhd b_j]\\ \Pi_1\end{array}\\ b_0 : A \quad b_0 \leqslant b & & b_j : A\end{array}}{b : A}\ ind \quad .$$

Let us consider a derivation Π^* obtained by substitution from Π and show that it is reducible. We have two cases:

(a) Π^* is obtained without replacing the last *ind*-application in Π:

$$\dfrac{\begin{array}{ccc}\Pi_0^* & & \Pi_1^*\\ b_0^* : A \quad b_0^* \leqslant b^* & & b_j^* : A\end{array}}{b^* : A}\ ind \quad .$$

By the induction hypothesis, Π_0^* and Π_1^* are reducible and thus, by *Red*1 of Theorem 5.41, they have a normal form. By Definition 5.40, it follows that Π^* is $\mathcal{S}$-reducible. Then, by *Red*2 of Theorem 5.41, we can conclude that Π^* is reducible.

(b) Π^* is obtained by replacing (also) the last *ind*-application in Π:

$$\Pi^* \quad = \quad \begin{array}{ccc} & \Pi_0^* & \\ b_0^* \leqslant b_1 & b_0^* : A & b_1 \lhd b_2 \\ & \Pi_1^*[b_1/b_i][b_2/b_j] & \\ b_0^* \leqslant b_2 & b_1 : A & b_2 \lhd b_3 \\ & \vdots & \\ b_0^* \leqslant b_{n-1} & b_{n-1} : A & b_{n-1} \lhd b^* \\ & \Pi_1^*[b_{n-1}/b_i][b^*/b_j] & \\ & b^* : A & \end{array}$$

for some n and some set of labels $\{b_1, \ldots, b_{n-1}\}$. By the induction hypothesis, Π_0 and Π_1 are reducible under substitution. From this, by induction on the value of n, it follows that Π^* is reducible whatever n is.

(*base* $\leqslant$), (*lin*$\lhd_{\mathcal{R}}$)
We consider the case *base* $\leqslant$; the other one is analogous. Let r be the last rule application in Π and let it be an application of *base* $\leqslant$:

$$\dfrac{\begin{array}{cc} & [b_{m-1} \leqslant b_m] \\ & \Pi' \\ b_{m-1} \lhd b_m & b : p \end{array}}{b : p}\ r \qquad .$$

We consider now a generic derivation Π^* obtained by substitution from Π and show that it is reducible. If r does not solve any *ind*-application, then we can simply use the induction hypothesis on Π' and Lemma 5.38. Otherwise, let s be an *ind*-application unfolded by r by a chain of rwffs

$$b_0 \lhd b_1\,,\ b_1 \lhd b_2\,,\ \ldots\,,\ b_{n-1} \lhd b_n\,,$$

for some n; Π will have the following form:

$$\dfrac{\begin{array}{cc} & [b_0 \leqslant b_i]\,[b_i : A]\,[b_i \lhd b_j] \\ \Pi_0 & \Pi_1 \\ b_0 : A \quad [b_0 \leqslant b_n] & b_j : A \end{array}}{\begin{array}{c} b_n : A \\ \Pi_2 \\ b : p \end{array}}\ s \qquad .$$
$$\dfrac{b_{m-1} \lhd b_m \qquad\qquad}{b : p}\ r$$

If in the derivation Π^*, obtained by substitution from Π, the application s is replaced, then we just apply the induction hypothesis on Π' and we are done. Otherwise, we have $\Pi^* \Rightarrow \widehat{\Pi}^*$:

$$\widehat{\Pi}^* = \dfrac{\begin{array}{cc} & \widehat{\Pi}'^* \\ b_{m-1} \lhd b_m & b^* : p \end{array}}{b^* : p}\ r$$

where $\widehat{\Pi}'^*$ is

$$\begin{array}{c} \Pi_0^* \\ b_0^* : A \\ \Pi_1^* \\ b_1^* : A \\ \vdots \\ \Pi_1^* \\ b_n^* : A \\ \Pi_2^* \\ b^* : p \end{array} \qquad .$$

Note that $\widehat{\Pi}'^*$ can be obtained by substitution from Π'. Thus from the reducibility under substitution of Π' we infer the reducibility of $\widehat{\Pi}'^*$. Furthermore, by Lemma 5.38, r preserves reducibility. It follows that $\widehat{\Pi}^*$ is reducible and thus, by Lemma 5.39, that also Π^* is reducible. We conclude that Π is reducible under substitution.
□

Corollary 5.44. *Every $\mathcal{N}(BCTL^*_{-_i})$ derivation is reducible.*

Proof. Immediate, by Lemma 5.43. We just notice that, according to Definitions 5.37 and 5.42, the notion of reducibility under substitution clearly implies that of reducibility, i.e. a derivation reducible under substitution is reducible.

□

Theorem 5.45. *Every $\mathcal{N}(BCTL^*_{-_i})$ derivation has a normal form.*

Proof. The thesis follows easily by Corollary 5.44, i.e. every derivation is reducible, and by property *Red*1 of Theorem 5.41, i.e. every reducible derivation has a normal form.

□

5.4.6 The form of normal derivations

Here we investigate the structure of normal derivations in $\mathcal{N}(BCTL^*_{-_i})$. We adapt from [151] the definition of spine.

Definition 5.46. *Given a derivation Π, a* spine *is a sequence of lwffs $b_1 : A_1, b_2 : A_2, \ldots, b_n : A_n$ such that:*

(i) $b_n : A_n$ is the conclusion of Π;
(ii) $b_{i+1} : A_{i+1}$ occurs immediately below $b_i : A_i$, for $1 \leq i < n$;
(iii) $b_i : A_i$ is the major (or the only) premise of a rule, for $1 \leq i < n$;
(iv) $b_1 : A_1$ is an assumption of Π or the conclusion of an ind-application.

Lemma 5.47. *In a normal derivation, a spine $b_1 : A_1, b_2 : A_2, \ldots, b_n : A_n$ can be divided into three parts:*

(i) an elimination part *$b_1 : A_1, \ldots, b_{m-1} : A_{m-1}$ where each $b_j : A_j$, for $1 \leq j < m-1$ is the major premise of an elimination rule with conclusion $b_{j+1} : A_{j+1}$;*
(ii) a minimum part *$b_m : A_m, \ldots, b_{m+k-1} : A_{m+k-1}$, where each formula except the last one is premise of $\perp E_i$ or of a relational rule;*
(iii) an introduction part *$b_{m+k} : A_{m+k}, \ldots, b_n : A_n$, where each $b_j : A_j$, for $m+k \leq j < n$ is premise of an introduction rule with conclusion $b_{j+1} : A_{j+1}$.*

Proof. Straightforward, by the definition of normal form, which requires the absence of maximum formulas in Π.

□

5.4.7 Consistency

We can exploit the structural properties of normal derivations to prove the consistency of $\mathcal{N}(BCTL^*_{-_i})$.

Theorem 5.48. *The system $\mathcal{N}(BCTL^*_{-_i})$ is consistent, i.e. $b : \perp$ is not derivable in $\mathcal{N}(BCTL^*_{-_i})$.*

Proof. We proceed by showing that a derivation concluding with $b :\bot$ must have at least one open assumption. Since each $\mathcal{N}(BCTL^*_{-i})$ derivation has a normal form (Theorem 5.45), we can restrict the analysis to normal derivations. Let Π be a normal derivation of $b :\bot$ and $b_1 : A_1, b_2 : A_2, \ldots, b_n : A_n \equiv b :\bot$ a spine of Π. First we note that Π has an atomic conclusion and thus, by Lemma 5.47, cannot contain introductions. Moreover, by the definition of a spine (Definition 5.46), there are no *ind*-applications below $b_1 : A_1$. Given that only introduction rules and *ind* can discharge lwffs, we have that $b_1 : A_1$ cannot be a discharged assumption. By Definition 5.46, we have two cases left:

(*i*) $b_1 : A_1$ is an open assumption, and then we are done; or

(*ii*) $b_1 : A_1$ is the conclusion of an *ind*-application s. Then let ρ be the ending premise of s. We will assume that ρ does not depend on any open assumption and show that this leads to a contradiction. We know that there are no $\mathsf{G}I$ and no *ind*-applications below s. Since these are the only rules that can discharge a formula of the form $b' \leqslant b''$ where b' and b'' do not coincide, we can conclude that all the leaves of the dependence tree of ρ are either $\lhd$-formulas or $\leqslant$-formulas of the form $b' \leqslant b'$ for some label b'. By Definition 5.19, it follows that ρ is unfoldable. But then s is unfoldable and Π is not normal (contradiction).

□

Corollary 5.49. *The system $\mathcal{N}(BCTL^*_{-})$ is consistent, i.e. $b :\bot$ is not derivable in $\mathcal{N}(BCTL^*_{-})$.*

Proof. By Lemma 5.17, $b :\bot$ is derivable in $\mathcal{N}(BCTL^*_{-})$ if and only if it is derivable in $\mathcal{N}(BCTL^*_{-i})$. By Theorem 5.48, we have the thesis.

□

5.4.8 The failure of the subformula property

Theorem 5.48 shows that the procedure of normalization that we have defined for $\mathcal{N}(BCTL^*_{-i})$ is good enough to get as a consequence a proof, by purely syntactic means, of the consistency of the system. However, as in normalization of natural deduction systems for Heyting arithmetic (see [152, Chapter 10.4.12] for an example), we do not have a subformula property. Namely, it is possible to show examples of $\mathcal{N}(BCTL^*_{-i})$ derivations that are normal with respect to Definition 5.23 but in which formulas occur that are neither subformulas of the conclusion nor of any of the open assumptions.

In Fig. 5.7, we give, as an example, an $\mathcal{N}(BCTL^*_{-i})$ derivation of $\{b : A, b : \mathsf{X}A, b : \mathsf{G}(A \supset \mathsf{X}\mathsf{X}A)\} \vdash_{\mathcal{N}(BCTL^*_{-i})} b : \mathsf{G}A$. The derivation is clearly in normal form and the formula $b : A \wedge \mathsf{X}A$, which occurs in it, is not a subformula of any of the open assumptions or of the conclusion, according to any reasonable definition of subformula for our labeled logic.

5.5 Discussion and related works

In this chapter, we have given labeled natural deduction systems for the until-free versions of a number of Ockhamist branching-time logics.

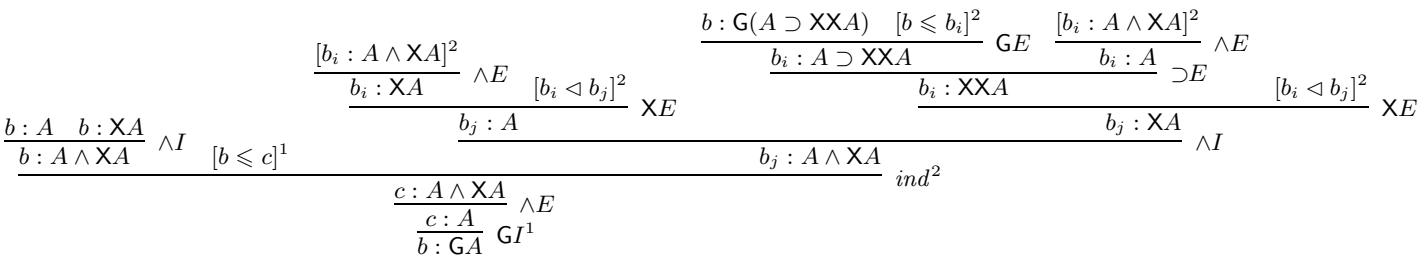


Fig. 5.7. An example of the failure of the subformula property.

Both in the case of general time (Section 5.2) and of discrete time (Section 5.3), we have considered a generalized version of the semantics giving rise to branching sublogics usually known as "bundled". We remark that this limitation is common also in the field of Hilbert-style axiomatizations, where an axiomatization has been only recently given for CTL^* [135] and announced for $OBTL$ by Reynolds. Moreover the one for CTL^* is a not completely standard axiomatization, which makes use of a rule involving the addition of fresh atoms in a proof (similar to the rule IRR of Gabbay [64]). The problem can be summarized, in the case of CTL^*, in the difficulty of capturing the limit-closure property (see Section 2.4.2), which is clearly a second-order property. An extension towards the logics endowed with such a semantics is left for future work. A first step could consist in considering a system with infinitary rules. It would be also interesting to consider an extension of our approach to $PCTL^*$, i.e. CTL^* with past, for which a completely standard Hilbert-style axiomatization has been provided [138].

We wish to remark, however, that $BCTL^*$ is relevant in itself when studying applications in which fairness constraints are considered [42]. Some authors [116, 118] also assert that bundled validity represents a more correct interpretation of human reasoning about time from a philosophical point of view.

We have already discussed a number of relevant related works in the previous sections. In the case of Ockhamist logics of general time, the only known deduction systems are Hilbert-style axiomatizations [68, 136, 164, 167].

Labeled natural deduction systems have been proposed for the logic CTL. Renteria and Haesler [131] present a system where logical formulas are labeled but no relational rules are given and indeed not even a notion of relational formula is used, since informations about the relations between labels are contained in the structure of the labels itself. The system is presented by restricting the attention to a minimal set of three temporal operators, for which introduction, elimination and "hybrid" (neither introduction nor elimination) rules are given. Some of these rules resemble reasoning similar to arithmetic induction. Both soundness and completeness are proved for the system. The presence of "hybrid" rules makes an analysis of normalization quite complex and unnatural.

In [19], Bolotov et al. also extend the approach presented in [19] for LTL in order to capture CTL. The same mechanism of flagging for labels is used, but in this case we have two separate classes of labels:

1. *state labels*, which are interpreted over time points;
2. *path labels*, which are interpreted over branches.

A further classification separates labels into *universal* and *rigid* with the idea that a universal label refers to a generic state (or path) and a rigid label to a specific state (or path). The authors consider the combination of a path quantifier and of a linear time operator as a unique temporal operator. For each of these operators, one or more introduction and one or more elimination rules are given. We give here the example of $(\forall \mathsf{G}_E)$ and $(\exists \mathsf{G}_E)$ which allows us also to show the use of labels in this context.

$$\frac{i : \forall \mathsf{G} A}{(i \leq j)_{\varphi^U}, \quad j : A} \, (\forall \mathsf{G}_E) \qquad \frac{i : \exists \mathsf{G} A}{(i \leq j)_{\varphi^R}, \quad j : A} \, (\exists \mathsf{G}_E)$$

Rules are given in Jaskowski style [94]: in this case we have rules with one premise and two conclusions. The two elimination rules are analogous but they differ in the fact that φ is a universal path label in $(\forall \mathsf{G}_E)$ (and thus refers to a generic path) and a rigid path label in $(\exists \mathsf{G}_E)$ (and thus refers to a particular path). As in [19], relational rules belong to a separate relational system and specific rules to model induction are required.

In [139], a tableau-based decision procedure for $BCTL^*$ is given. The tableau construction differs from the traditional tree-shaped one and consists, like for other tableau systems for temporal logics, e.g. [54, 163], in starting with a graph and iteratively pruning away some nodes until a success or a failure condition is reached. We remark that the focus of our work, instead, mainly concerns the definition of a deduction system with good proof-theoretical properties.

Part III

Mosaics for Temporal Logics

6

The Mosaic Method for Temporal Logics

6.1 Introduction

The mosaic method has been introduced in algebraic logic as a way of proving the decidability of the theories of some classes of algebras of relations [114, 115]. The basic idea consists in showing that the existence of a model is equivalent to the existence of a (finite) set of fragments of models (called *mosaics*). There are of course several conditions to be satisfied: first of all, every single mosaic needs to satisfy some local *coherency conditions*; furthermore, the set of mosaics is required to be closed with respect to a number of *saturation conditions*.

The usefulness of mosaics comes from the fact that, given a formula, we do not need to generate a full model in order to prove its satisfiability: it is enough to show that there exists such a saturated set of mosaics. Thus we have a decision procedure for the logic, which consists in checking whether such a (finite) set exists or not. The mosaic method has been recently applied to prove decidability, complexity results and completeness of Hilbert-style axiomatizations for several modal logics [86, 111, 157].

With regard to temporal logics, a first work considering an adaptation of the technique to the linear temporal logic Kl is [105]. In this paper, the authors give a proper definition of mosaics for the logic Kl and prove that the existence of a saturated set of mosaics for a formula is indeed equivalent to the existence of a model for that formula. Then they apply this result to prove the decidability of the logic and the completeness of a given Hilbert-style axiomatization. A mosaic-based labeled tableau construction is also presented, and the ideas behind that are used to provide a method for automated theorem-proving. Finally, a generalization of these results to the case of several variants of Kl is sketched by suggesting possible modifications of the conditions defining mosaics and saturated sets of mosaics.

Further works using mosaics in temporal logics established complexity results for the logic of *until* over general linear time [137] and the logic using both *since* and *until* over the reals [134] (See also [140,141] for more recent and general accounts on mosaics and complexity topics.) In [133], a variant of the mosaic method has been used to prove decidability of a so-called temporal logic of parallelism, mentioned also in [150]. This logic consists in a simple combination of the temporal operators F and P with a modal operator $\Diamond$. The semantics is given on rectangular frames

consisting of the cross product of a (vertical) linear order and a (horizontal) non-empty set. F and P operate along the vertical lines and $\Diamond$ acts horizontally as an *S5* existential operator but there is no dependence between the vertical and the horizontal relations. In the paper, it is also shown that this logic does not enjoy the finite model property and thus that the mosaic method is in some cases a more powerful tool for proving decidability.

In this chapter, first we briefly recall, mainly from [105], the definitions and results concerning mosaics for linear temporal logics. Then we propose an extension of the mosaic method to the case of branching-time logics. Here we will limit our extension to the case of the bundled Ockhamist branching logics presented in Section 2.4.1, i.e., to *BOBTL* and some of its sublogics. The results concerning decidability and completeness of these logics are already well-known, however we believe that the mosaic method is interesting in itself as it provides a uniform way of establishing such results for a large class of logics, by simple and modular modifications of the basic definitions. Moreover, our proposal for this class of branching-time logics can be seen as a basis for dealing with other more interesting logics, for which decidability and complexity results are still missing.

We also remark that, in this thesis, we do not consider extensions of the mosaic-based techniques to more complex linear-time logics, like *LTL*, or branching-time logics, like computation tree logics, for which further work is required.

The structure of the chapter is the following:

- in Section 6.2, we consider the use of mosaics in the case of linear-time logics. We recall the results from [105] and adapt them to the case of some logics not explicitly considered there;
- in Section 6.3, an extension for the branching-time bundled logics of Section 2.4.1 is proposed.

6.2 Mosaics for linear temporal logics

In this section, we define mosaics in the case of the basic priorean tense logics of Sections 2.3.1 and 2.3.2. Most of the results presented in this section come from [105], where the definition of mosaics for *Kl* and other temporal logics with the operators F and P over linear flows of time is given.

6.2.1 Mosaics for the basic priorean tense logics

In this section, and in general when dealing with mosaics, we will consider as primitive connectives $\wedge$ and $\neg$, instead of $\supset$ and $\bot$. Intuitively, temporal mosaics can be seen as pairs (M, M') where the two elements M and M' refer to two points in a temporal structure, such that the point associated to M precedes (by the relation $\prec$) the one associated to M'. An element M is indeed a set of formulas, namely the set of formulas that are evaluated true at that point.

Given this basic intuition, it seems reasonable to require that mosaics satisfy some local coherency conditions: as an example, given a mosaic (M, M'), we want that if $\mathsf{G}A \in M$, then $A \in M'$. Moreover, we are interested in considering particular sets of mosaics, saturated in such a way that we are able to build a complete model

by just composing the mosaics contained in a given set of that kind. This means we need to define the saturation conditions that a "good" set of mosaics is required to satisfy. Basically, this amounts to making sure that each counterexample occurring in the model we are building can be "cured". In this context, a counterexample consists in the presence of a point w labeled with a formula of the form $\mathsf{F}A$ such that all of its successors are labeled with $\neg A$. By "curing" it, we mean adding a new point w' in the structure (as a successor of w) such that the labeling set of w' contains A.

These ideas are formalized in the following definitions and theorems.

Mosaics for *Kt*

For completeness, and uniformity with previous chapters, here we adapt the definition of a mosaic, given in [105] for the logic Kl, to the case of the simpler Kt. Note that, as in [105], our definition also admits the presence of mosaics that are singletons: we need them in order to consider the existence of single-point models (or, possibly, of models containing disconnected points, i.e. points that are not related to any other point).

Definition 6.1. *Let Δ be a set of formulas closed under subformulas and single negation, in the language of tense formulas (Section 2.3.1). A* mosaic *(on Δ) is a pair (M_0, M_1) or a singleton (M_0), where $M_0, M_1 \subseteq \Delta$, satisfying the following* coherency conditions*.*

COHERENCY CONDITIONS

For every formula $A, B \in \Delta$ and $i \in \{0, 1\}$,

(CL1) $A \in M_i$ iff $\neg A \notin M_i$;
(CL2) $A \wedge B \in M_i$ iff $\{A, B\} \subseteq M_i$;
(CL3) if $A = \mathsf{G}A' \in M_0$, then $A' \in M_1$;
(CL4) if $A = \mathsf{H}A' \in M_1$, then $A' \in M_0$.

In the case of a mosaic being a singleton only conditions $CL1$ and $CL2$ need to be satisfied.

Definition 6.2. *Let S be a set of mosaics on Δ. Then the set of* points of *S is the set $Points(S) = \{M \subseteq \Delta \mid$ there exists $(M_0, M_1) \in S$ or $(M_0) \in S$ s.t. $M_0 = M$ or $M_1 = M\}$.*

Definition 6.3. *A set S of mosaics is a Kt-*saturated set of mosaics *(a Kt-SSM for short) if it satisfies the following* saturation conditions*.*

SATURATION CONDITIONS

For every point $M \in Points(S)$,

(SL1) if $\mathsf{F}A \in M$, then there exists $(M, M') \in S$ s.t. $A \in M'$;
(SL2) if $\mathsf{P}A \in M$, then there exists $(M', M) \in S$ s.t. $A \in M'$;

Theorem 6.4. *For any set Γ of tense formulas, Γ is Kt-satisfiable iff there exists a Kt-SSM for Γ.*

Proof. ($\Rightarrow$) Let $\mathcal{M} = (\mathcal{W}, \prec, \mathcal{V})$ be a temporal structure satisfying Γ and let $u \in \mathcal{W}$ be a point such that $\mathcal{M}, u \models \Gamma$. Given a set Δ, which contains Γ and is closed under subformulas and single negations, we can associate a subset of Δ to every point of $\mathcal{W}$, i.e. for every $w \in \mathcal{W}$ we define $M_w = \{A \in \Delta \mid \mathcal{M}, w \models A\}$. Then we can define the set $S = \{(M_w, M_{w'}) \mid w, w' \in \mathcal{W} \text{ and } w \prec w'\} \cup \{(M_w) \mid w \in \mathcal{W} \text{ and for all } w' \in \mathcal{W} \text{ we have } w \not\prec w' \text{ and } w' \not\prec w\}$. It is easy to verify that every element of S is indeed a mosaic and that the set S is a Kt-SSM. In fact, coherence and saturation conditions are clearly satisfied since the definition of each point in S comes from the labeling of the corresponding point in a temporal structure. Furthermore, S is a Kt-SSM for Γ since $\Gamma \subseteq M_u$ and $M_u \in \mathit{Points}(S)$.

($\Leftarrow$) Let S be a Kt-SSM for Γ and Δ the set of formulas (containing Γ) on which mosaics are defined, i.e. S is a set of mosaics on Δ. Then, in order to obtain a temporal structure satisfying Γ, we just define a set of instants $\mathcal{W}$ isomorphic to the set of points of S, i.e. $\mathcal{W} = \{w_M \mid M \in \mathit{Points}(S)\}$. Then we set $\prec = \{(w_M, w_{M'}) \mid (M, M') \in S\}$ and $\mathcal{V}(w_M) = M$ for every $M \in \mathit{Points}(S)$. □

Mosaics for *Kl*

Things get more interesting when we consider more specific flows of time. In [105], mosaics for the logic Kl, i.e. the logic of irreflexive, transitive and connected orderings (see 2.3.2 for details), are defined. In this case, coherence conditions are enriched by a new one capturing the transitivity of Kl-frames: so, for instance, if $\mathsf{G}A$ is in a point M, then it must also be in all the points M' such that (M, M') is a mosaic. Linearity is obtained by adding a further saturation condition, which says that if (M, M') is a mosaic in our set such that $\mathsf{F}A$ is in M but $\mathsf{F}A \notin M'$, then there must be an intermediate point (a point between M and M') satisfying A.

Definition 6.5. *Let Δ be a set of formulas closed under subformulas and single negation, in the language of tense formulas (Section 2.3.1). A* mosaic *(on Δ) is a pair (M_0, M_1) or a singleton (M_0), where $M_0, M_1 \subseteq \Delta$, satisfying the following* coherency conditions*.*

COHERENCY CONDITIONS

For every formula $A, B \in \Delta$ and $i \in \{0, 1\}$

(CL1), (CL2), (CL3) and (CL4) as defined in Definition 6.1;

(CL5) if $A = \mathsf{G}A' \in M_0$, then $\mathsf{G}A' \in M_1$;
(CL6) if $A = \mathsf{H}A' \in M_1$, then $\mathsf{H}A' \in M_0$.

In the case of a mosaic being a singleton, only conditions $CL1$ and $CL2$ need to be satisfied.

The set of points of a given set of mosaics is defined as before. We express now the saturation conditions.

Definition 6.6. *A set S of mosaics is a* Kl-saturated set of mosaics *if it satisfies the following* saturation conditions*.*

SATURATION CONDITIONS

For every mosaic $(M_0, M_1) \in S$,

(SL1) and (SL2) as in Definition 6.3;

(SL3) if $\mathsf{F}A \in M_0$, *then:*
(i) $\mathsf{F}A \in M_1$; *or*
(ii) there exist $(M_0', M_1'), (M_0'', M_1'') \in S$ *s.t.* $M_0 = M_0'$, $M_1 = M_1''$ *and* $A \in M_1' = M_0''$;
(SL4) if $\mathsf{P}A \in M_1$, *then:*
(i) $\mathsf{P}A \in M_0$; *or*
(ii) there exist $(M_0', M_1'), (M_0'', M_1'') \in S$ *s.t.* $M_0 = M_0'$, $M_1 = M_1''$ *and* $A \in M_1' = M_0''$;

Theorem 6.7. *For any set* Γ *of tense formulas,* Γ *is Kl-satisfiable iff there exists a Kl-SSM for* Γ.

Proof. ($\Rightarrow$) As in the proof of Theorem 6.4.

($\Leftarrow$) We give here just a sketch of the proof; full details can be found in [105]. Given a *Kl*-SSM S for Γ we build a structure satisfying Γ step by step, by using the mosaics in S as building blocks. We begin with a mosaic containing Γ in one of its points and at each step we cure a defect of the construction, where a defect is represented by some point labeled with a formula of the form $\mathsf{F}A$ such that none of its successors is labeled with A (or by the symmetric situation with regard to the past). Saturation conditions ensure that such a curing is always possible, i.e. that it is always possible to provide a proper witness. The construction is an ω-construction. At the ω-step, we obtain a labeled structure that is a *Kl*-model where no defects occur. Furthermore, as required, such a structure satisfies Γ.

□

6.2.2 Applications

Completeness via mosaics

One of the possible applications of the mosaic method is its use in proving the completeness of a given Hilbert-style axiomatization. In fact, Theorems 6.4 and 6.7 can be used to simplify the standard proofs of completeness: given a consistent[1] set of formulas we do not need to create a model satisfying it; an SSM will suffice.

Theorem 6.8. *For any set* Γ *of tense formulas,* Γ *is Kt-consistent (Kl-consistent) iff there exists a Kt-SSM (a Kl-SSM) for* Γ.

Proof. ($\Rightarrow$) Given a consistent set Γ of formulas, we can build a saturated set of mosaics as follows. As labeling set we use the set of all formulas in the language and we consider maximal consistent sets on this language with respect to the axiomatization $\mathcal{H}(Kt)$ ($\mathcal{H}(Kl)$, respectively) of Section 2.3. Then,

[1] We recall that a set Γ of formulas is consistent with respect to an inference system iff it is impossible to derive contradictions from Γ by using the inference system.

in the case of Kt, we define the set S of mosaics as the set $S = \{(M, M') \mid M, M'$ are MCSs and for every $\mathsf{G}A \in M$, $A \in M'\}$. In the case of Kl, we need to consider also transitivity and thus the set of mosaics is $S = \{(M, M') \mid M, M'$ are MCSs and for every $\mathsf{G}A \in M$, $\{A, \mathsf{G}A\} \subseteq M'\}$. One can prove that each element of S is indeed a mosaic, i.e. that the coherency conditions are satisfied, and that S is saturated, i.e. that the saturation conditions are satisfied. More details in [105].

($\Leftarrow$) If there exists an SSM for Γ, then Γ is satisfiable, and hence consistent by the soundness of $\mathcal{H}(Kt)$ (or $\mathcal{H}(Kl)$).

□

Decidability via mosaics

The most typical use of the mosaic method, however, is in showing the decidability of a given logic. Although decidability of the logics Kt and Kl is already well-known, here we sketch a proof obtained by using mosaics. We remark that, as in proving completeness, our work is simplified with respect to standard proofs of decidability (e.g. via the finite model property) by the results of Theorems 6.4 and 6.7. Further details can be found in [105].

Theorem 6.9. *Given a tense formula A, checking its satisfiability (with respect to Kt or Kl semantics) is decidable.*

Proof. By Theorems 6.4 and 6.7, we only need to show that the task of checking whether there is an SSM (a Kt-SSM or a Kl-SSM, according to which case we are interested in) for A is decidable. We use the set of the subformulas of A and their single negations as labeling set. The number of possible mosaics on that labeling set is finite and checking the saturation conditions is clearly decidable (both for Kt and for Kl). Thus it is also decidable whether any subset of the set of all mosaics form an SSM for A.

□

6.2.3 Mosaics for other linear flows of time

It is possible to adapt the definition of mosaic and SSM in order to capture variants of Kl, i.e. other axiomatic extensions presented in Section 2.3.2. Some of them are described in [105]. We list them here. These changes will require in some cases trivial extensions of the labeling set in order to keep it closed under subformulas and single negations.

Substructures of the whole numbers In condition SL3, we require not only $A \in M'_1$ but also $\neg\mathsf{F}A \in M'_1$. (An analogous modification can be made for the symmetric condition SL4.) This implies that once we insert a point satisfying a given A as a witness for an $\mathsf{F}A$-defect, in our construction $\mathsf{F}A$-defects will no longer occur. Since there are only finitely many $\mathsf{F}A$ (and $\mathsf{P}A$) in our labeling set[2], we will insert only finitely many points into the linear order under construction.

[2] Note that this modification works only when the labeling set is supposed to be finite, e.g. in proving decidability or weak (but not strong) completeness.

Without endpoint We add as a further coherence condition that $\mathsf{F}\top \in M_i$.
With endpoint We add as a further coherence condition that $\mathsf{FG}\ \bot \in M_i$.
Without beginning point We add as a further coherence condition that $\mathsf{P}\top \in M_i$.
With beginning point We add as a further coherence condition that $\mathsf{PH}\ \bot \in M_i$.
Dense We require in the definition of a *Kl*-SSM that, for every mosaic, there exist mosaics that can be inserted in-between, like in saturation conditions SL3 and SL4. Then, in the construction of the model from the *Kl*-SSM, in each step we insert the provided points between all neighboring points. In the limit step, there will be no immediate successors and predecessors.

Finally we remark that a refined definition of mosaics covering also the case of the operators since and until is proposed in [105, 134, 137].

6.3 Mosaics for branching temporal logics

Here we extend the definition of the mosaic method for a linear tense logic (see Section 6.2) to the case of several bundled branching logics. We will start by giving the definition of mosaics for the logic of basic frames (see Section 2.4.1) and, by following the classification of [167], by extending it to other more complex and, in a sense, more "branching" logics. Throughout this section, the formulas that we consider belong to the Ockhamist language defined in Section 2.14.

As remarked in Section 2.4, we consider in this thesis branching logics where the evaluation of atoms depends only on the state we are considering and not on the path we are going to follow (*no trace of futurity* assumption) and this assumption is crucial in our extension of the mosaic method to the branching case.

We keep here the intuition behind linear temporal mosaics: we still deal with pairs (M, M') of sets of formulas, such that each set refers to a point in a structure and such that the point referred from M $\prec$-precedes the one referred from M'. As in the linear case, in our key theorem we need to show how to build a full structure from a (saturated) set of mosaics. In other words, we need to define a proper way of combining mosaics, both vertically and horizontally. Vertical combinations are defined as in the linear case: we iteratively provide witnesses for linear counterexamples, where a linear counterexample is a point labeled with a formula of the form $\mathsf{F}A$ such that none of its successors is labeled with A. In the case of branching logics, we need to consider also branching counterexamples (and thus horizontal combinations of mosaics): given a point w labeled with a formula of the form $\exists A$, we add in the structure a new point w', which satisfies A and is in some way "compatible" with w. Since we follow the *no trace of futurity* approach, we can let such a compatibility consist basically in the fact that w and w' satisfy the same set of state formulas.

6.3.1 Mosaics for the logic of basic frames

We distinguish between linear and branching, both for coherency and for saturation conditions. Linear conditions are as expressed in Section 6.2; for clarity, we recall them in the following definition.

Definition 6.10. *Let Δ be a set of formulas closed under subformulas and single negation, in the language of Ockhamist formulas (Section 2.4.1). A* mosaic *(on Δ) is a pair (M_0, M_1) or a singleton (M_0), where $M_0, M_1 \subseteq \Delta$, satisfying the following* coherency conditions.

COHERENCY CONDITIONS

For every formula $A, B \in \Delta$ and $i \in \{0, 1\}$,

LINEAR CONDITIONS

(CL1) $A \in M_i$ iff $\neg A \notin M_i$;
(CL2) $A \wedge B \in M_i$ iff $\{A, B\} \subseteq M_i$;
(CL3) if $A = \mathsf{G}A' \in M_0$, then $A' \in M_1$;
(CL4) if $A = \mathsf{H}A' \in M_1$, then $A' \in M_0$;
(CL5) if $A = \mathsf{G}A' \in M_0$, then $\mathsf{G}A' \in M_1$;
(CL6) if $A = \mathsf{H}A' \in M_1$, then $\mathsf{H}A' \in M_0$.

BRANCHING CONDITIONS

(CB1) if $A = \forall A' \in M_i$, then $A' \in M_i$.

In the case of a mosaic being a singleton only conditions $CL1$, $CL2$ and $CB1$ need to be satisfied.

Definition 6.11. *The set of (Ockhamist)* state formulas *is defined recursively as follows:*

1. *all atomic formulas are state formulas;*
2. *if A and B are state formulas, then $A \wedge B$ is a state formula;*
3. *if A is a state formula, then $\neg A$ is a state formula;*
4. *if A is an Ockhamist formula, then $\forall A$ is a state formula.*

Definition 6.12. *Let Δ be a set of formulas closed under subformulas and single negation and $M, M' \in \Delta$. We say that M and M' are* state-equivalent *(and we write $M \sim_s M'$) if for each Ockhamist state formula $A \in \Delta$, $A \in M$ if and only if $A \in M'$.*

In the following definition, we will also use the notion of *points* of a set of mosaics defined in Section 6.3.1.

Definition 6.13. *A set S of mosaics is a* basic saturated set of mosaics *(a basic SSM for short) if it satisfies the following* saturation conditions.

SATURATION CONDITIONS

For every mosaic $(M_0, M_1) \in S$,

LINEAR CONDITIONS

(SL1) if $\mathsf{F}A \in M_1$, then there exists $(M_0', M_1') \in S$ s.t. $M_1 = M_0'$ and $A \in M_1'$;
(SL2) if $\mathsf{P}A \in M_0$, then there exists $(M_0', M_1') \in S$ s.t. $M_0 = M_1'$ and $A \in M_0'$;
(SL3) if $\mathsf{F}A \in M_0$, then:
(i) $\mathsf{F}A \in M_1$; or

(ii) there exist $(M_0', M_1'), (M_0'', M_1'') \in S$ *s.t.* $M_0 = M_0'$, $M_1 = M_1''$ *and* $A \in M_1' = M_0''$;

(SL4) if $\mathsf{P}A \in M_1$, *then:*

(i) $\mathsf{P}A \in M_0$; *or*

(ii) there exist $(M_0', M_1'), (M_0'', M_1'') \in S$ *s.t.* $M_0 = M_0'$, $M_1 = M_1''$ *and* $A \in M_1' = M_0''$;

BRANCHING CONDITIONS

(SB1) if $M \in Points(S)$ *and* $\exists A \in M$, *then there exists* $M' \in Points(S)$ *s.t.* $M \sim_s M'$ *and* $A \in M'$.

Given an SSM S *and a set of formulas* Γ, *we say that* S *is a* basic SSM for Γ *if there exists* $M \in Points(S)$ *such that* $\Gamma \subseteq M$.

Theorem 6.14. *For any set* Γ *of formulas,* Γ *is* (*Basic*)*-satisfiable iff there exists a basic SSM for* Γ.

Proof. ($\Rightarrow$) Let $\mathcal{M} = (\mathcal{T}, \prec, \simeq, \mathcal{V})$ be a basic structure satisfying Γ and let $u \in \mathcal{T}$ be a point such that $\mathcal{M}, u \models \Gamma$. Given a set Δ, which contains Γ and is closed under subformulas and single negations, we can associate a subset of Δ to every point of $\mathcal{T}$, i.e. for every $v \in \mathcal{T}$ we define $M_v = \{A \in \Delta : \mathcal{M}, v \models A\}$. Then we can define the set $S = \{(M_v, M_v') : v, v' \in \mathcal{T} \text{ and } v \prec v'\} \cup \{(M_v) : v \in \mathcal{T} \text{ and for all } v' \in \mathcal{T} \text{ we have } v \not\prec v' \text{ and } v' \not\prec v\}$. It is easy to verify that every element of S is indeed a mosaic and that the set S is a basic SSM. In fact coherence and saturation conditions are clearly satisfied since the definition of each point in S comes from the labeling of the corresponding point in a basic structure. Furthermore S is a basic SSM for Γ since $\Gamma \subseteq M_u$ and $M_u \in Points(S)$.

($\Leftarrow$) Let S be a basic SSM for Γ and Δ the set of formulas (containing Γ) on which mosaics are defined, i.e. S is a set of mosaics on Δ. As in [105], we will build a model for Γ step by step by using the mosaics in S as building blocks. The structure that we are going to construct can be seen as a grid composed by a countable set of vertical lines, where each vertical line is a substructure of the rational numbers and every point in the structure is associated with a set of formulas (a subset of Δ).

Formally a *labeled structure* L has the form $(H, \{V_h, <_h\}_{h \in H}, \equiv, \mathcal{L})$, where:

1. $H \subseteq \mathbb{N}$;
2. $V_h \subseteq \mathbb{Q}$ for every $h \in H$
   ((h, v) is said to be a *point* of L if $h \in H$ and $v \in V_h$);
3. $<_h$ is the order defined on rational numbers restricted to V_h for every $h \in H$;
4. $\equiv$ is an equivalence relation defined between points of L; and
5. $\mathcal{L}$ is a labeling function which associates a subset of Δ to every point of L[3].

The construction proceeds by "curing" at every step one of the *defects* in the structure. First we enumerate all the possible defects. They are of three kinds:

1. *linear future defects* of the form $\langle (h, v), \mathsf{F}A \rangle$, where (h, v) represents a point in the structure and $\mathsf{F}A$ is a formula in Δ;

[3] In order to simplify the notation, in the following, given a point (h, v), we will write $\mathcal{L}(h, v)$ instead of $\mathcal{L}((h, v))$.

2. *linear past defects* of the form $\langle (h,v), \mathsf{P}A\rangle$, where (h,v) represents a point in the structure and $\mathsf{P}A$ is a formula in Δ;
3. *branching defects* of the form $\langle (h,v), \exists A\rangle$, where (h,v) represents a point in the structure and $\exists A$ is a formula in Δ.

Since the language contains at most countably many atoms, also the number of defects is countable. Thus we can set an enumeration over $\mathbb{N}$ of the following set D of possible defects:

$$D = \{\langle (h,v), \mathsf{F}A\rangle, \langle (h,v), \mathsf{P}A\rangle, \langle (h,v), \exists A\rangle : h \in \mathbb{N}, v \in \mathbb{Q} \text{ and } \mathsf{F}A, \mathsf{P}A, \exists A \in \Delta\}\,.$$

Given a labeled structure $L = (H, \{V_h, <_h\}_{h\in H}, \equiv, \mathcal{L})$, we say that an element $\langle (h,v), \mathsf{F}A\rangle$ of D is a *linear future defect of* L if:

1. (h,v) is a point of L;
2. $\mathsf{F}A \in \mathcal{L}(h,v)$;
3. for every (h,v') such that $v' \in V_h$ and $v <_h v'$, we have $A \notin \mathcal{L}(h,v')$.

In a similar way, we say that $\langle (h,v), \mathsf{P}A\rangle$ of D is a *linear future defect of* L if:

1. (h,v) is a point of L;
2. $\mathsf{P}A \in \mathcal{L}(h,v)$;
3. for every (h,v') such that $v' \in V$ and $v' <_h v$, we have $A \notin \mathcal{L}(h,v')$.

Finally, $\langle (h,v), \exists A\rangle \in D$ is a *branching defect of* L if:

1. (h,v) is a point of L;
2. $\exists A \in \mathcal{L}(h,v)$;
3. for every point (h',v') of L, if $(h,v) \equiv (h',v')$ then $A \notin (h',v')$.

Furthermore, we will say that L is *coherent* if the following conditions (analogous of the coherency conditions in Definition 6.10) are satisfied by every point (h,v) of L:

1. $A \in \mathcal{L}(h,v)$ iff $\neg A \notin \mathcal{L}(h,v)$;
2. $A \wedge B \in \mathcal{L}(h,v)$ iff $\{A,B\} \subseteq \mathcal{L}(h,v)$;
3. if $\mathsf{G}A \in \mathcal{L}(h,v)$, then $\{A, \mathsf{G}A\} \subseteq \mathcal{L}(h,v')$ for every $v' \in V_h$ such that $v <_h v'$;
4. if $\mathsf{H}A \in \mathcal{L}(h,v)$, then $\{A, \mathsf{H}A\} \subseteq \mathcal{L}(h,v')$ for every $v' \in V_h$ such that $v' <_h v$;
5. if $\forall A \in \mathcal{L}(h,v)$, then $A \in \mathcal{L}(h,v)$.

Our construction is such that at every step $n < \omega$ we will have a labeled structure $L_n = (H_n, \{V_{h_n}, <_{h_n}\}_{h\in H_n}, \equiv_n, \mathcal{L}_n)$ satisfying the following *formation conditions*:

(F1) L_n is coherent;
(F2) for every $h \in H_n$, $(V_{h_n}, <_{h_n})$ determines a finite linear order of rational numbers $\langle i_{0_h} < i_{1_h} < \ldots < i_{k_h}\rangle$ such that, for every j, $(\mathcal{L}_n(h, i_{j_h}), \mathcal{L}_n(h, i_{j+1_h}))$ is a mosaic in S;
(F3) if $(h,v) \equiv_n (h',v)$[4] then $\mathcal{L}_n(h,v)$ and $\mathcal{L}_n(h',v)$ are state-equivalent.

[4] Note that our construction will be such that whenever two points (h_1,v_1) and (h_2,v_2) are $\equiv$-equivalent at some stage j, i.e. $(h_1,v_1) \equiv_j (h_2,v_2)$, then v_1 and v_2 must coincide. Viceversa, having at some stage j two points (h_1,v_1) and (h_2,v_2) such that $v_1 = v_2$ does not imply $(h_1,v_1) \equiv_j (h_2,v_2)$.

Note that the condition (F3) is the analogous of the branching saturation condition (SB1) of Definition 6.13.

We will use a scheduling function $\sigma : \omega \to \omega$ such that, for every $j \in \omega$, there are infinitely many k such that $\sigma(k) = j$. At the n-th step we will cure the $\sigma(n)$-th defect in our enumeration of D. In the following we describe our limit construction of a model for Γ.

[*STEP* 0] First let us consider a mosaic $\mu \in S$ such that μ is a mosaic for Γ (since S is a basic SSM for Γ, such a mosaic exists). If $\mu = (M_0)$ is a singleton, then we can define an L_0 such that $H_0 = \{0\}$, $V_{0_0} = \{0\}$, $<_{0_0} = \emptyset$, $\equiv_0 = \{((0,0),(0,0))\}$, $\mathcal{L}_0(0,0) = M_0$. If $\mu = (M_0, M_1)$, then L_0 is such that $H_0 = \{0\}$, $V_{0_0} = \{0,1\}$, $<_{0_0} = \langle 0,1\rangle$, $\equiv_0 = \{((0,0),(0,0)), ((0,1),(0,1))\}$, $\mathcal{L}_0(0,0) = M_0$, $\mathcal{L}_0(0,1) = M_1$. Note that in both cases L_0 trivially satisfies formation conditions.

[*STEP* $n+1$] Assume that we have already defined a labeled structure L_n satisfying the formation conditions. Then we consider the $\sigma(n+1)$-th defect d in our enumeration of D. If d is not an actual defect of L_n, then we just set $L_{n+1} = L_n$. Otherwise we have three cases:

(i) $d = \langle (h,v), \mathsf{F}A\rangle$ is a linear future defect. Then let v' be the greatest element of V_{h_n} with respect to the order $<_{h_n}$ such that $\mathsf{F}A \in (h,v')$. Since d is an actual defect of L_n, such v' exists. We have two subcases:

(a) v' is the greatest element of V_{h_n} according to $<_{h_n}$. Then by the saturation condition (SL1) there is a mosaic (M_0', M_1') in S such that $M_0' = \mathcal{L}_n(h,v')$ and $A \in M_1'$. We add a new element $(v'+1)$ to V_{h_n} and define $<_{h_{n+1}}$ as the restriction to $V_{h_{n+1}}$ of the usual order $<$ on rational numbers. Formally, we define:
- $H_{n+1} = H_n$;
- $V_{h_{n+1}} = V_{h_n} \cup \{v'+1\}$;
- $\mathcal{L}_{n+1}(h, v'+1) = M_1'$;
- $V_{i_{n+1}} = V_{i_n}$ for every $i \in H_{n+1}$ such that $i \neq h$;
- $<_{i_{n+1}}$ for every $i \in H_{n+1}$ is the restriction to $V_{i_{n+1}}$ of the usual order $<$ on rational numbers;
- $\equiv_n = \equiv_{n+1} \cup \{((h,v'+1),(h,v'+1))\}$;
- $\mathcal{L}_{n+1}(i,j) = \mathcal{L}_n(i,j)$ for every point (i,j) of L_n.

(b) v' is not the greatest element of V_{h_n}. Then there exists an element $v'' \in V_{h_n}$ such that v'' is the immediate successor of v', according to the relation $<_{h_n}$, and, by the maximality of v', $\neg\mathsf{F}A \in \mathcal{L}_n(h,v'')$. By the condition (SL3), there exist two mosaics (M_0, M), $(M, M_1) \in S$ such that $M_0 = \mathcal{L}_n(h,v')$, $M_1 = \mathcal{L}_n(h,v'')$ and $A \in M$. Then we insert a point v^* between v' and v'' and label (h,v^*) with M.
By summing up, we define L_{n+1} as follows:
- $H_{n+1} = H_n$;
- $V_{h_{n+1}} = V_{h_n} \cup \{v^*\}$, where v^* is a rational number such that $v' < v^* < v''$;
- $\mathcal{L}_{n+1}(h,v^*) = M$ where M is obtained as described above;
- $<_{i_{n+1}}$ for every $i \in H_{n+1}$ is the restriction to $V_{i_{n+1}}$ of the usual order $<$ on rational numbers;
- $V_{i_{n+1}} = V_{i_n}$ for every $i \in H_{n+1}$ such that $i \neq h$;
- $\equiv_n = \equiv_{n+1} \cup \{((h,v'+1),(h,v'+1))\}$;

- $\mathcal{L}_{n+1}(i,j) = \mathcal{L}_n(i,j)$ for every point (i,j) of L_n;

(ii) $d = \langle (h,v), \mathsf{P}A\rangle$ is a linear past defect. Then the treatment of such defects exploits the saturation conditions (SL2) and (SL4) of the basic SSM S and is completely symmetrical to that of future defects; we omit a detailed description;

(iii) $d = \langle (h,v), \exists A\rangle$ is a branching defect. By the saturation conditions in Definition 6.13, we know that there exists $M' \in \mathit{Points}(S)$ such that $\mathcal{L}_n((h,v)) \sim_s M'$ and $A \in M'$. Then we add a new vertical line (say with index $n+1$) consisting of a single element (say with index v) labeled with M'. Formally, we define L_{n+1} as follows:

- $H_{n+1} = H_n \cup \{n+1\}$;
- $V_{n+1_{n+1}} = \{v\}$;
- $<_{n+1_{n+1}} = \emptyset$;
- $V_{i_{n+1}} = V_{i_n}$ for every $i \in H_n$;
- $<_{i_{n+1}} = <_{i_n}$ for every $i \in H_n$;
- $\mathcal{L}_{n+1}((n+1,v)) = M'$;
- $\mathcal{L}_{n+1}((i,j)) = \mathcal{L}_n((i,j))$ for every point (i,j) of L_n.

The construction is such that in all the cases we get a labeled structure L_{n+1} which satisfies formation conditions F1, F2 and F3 and where d is no longer a defect. In order to ensure that the limit construction is well defined, it is also important to remark that the new labeling $\mathcal{L}_{n+1}$ is just an extension of the old $\mathcal{L}_n$ and that the defect d (once cured) cannot occur in any expansion of the structure.

[*STEP* ω] Now we can just take the union $L = (H, \{V_h, <_h\}_{h\in H}, \equiv, \mathcal{L})$ of the labeled structures defined so far. L is a coherent labeled structure that does not contain any defect, since the scheduling function σ ensures that if a defect becomes actual at some step, then we cure it in a later step.

We can then build a basic structure satisfying Γ by using the labeled structure L. Namely, we define a structure $\mathcal{M} = (\mathcal{T}, \prec, \simeq, \mathcal{V})$ such that:

1. $\mathcal{T} = \{u : u \text{ is a point of } L\}$;
2. $\prec = \bigcup_{h\in H} <_h$;
3. $\simeq = \equiv$;
4. for all $u \in \mathcal{T}$, $p \in \mathcal{V}(u)$ iff $p \in \mathcal{L}(u)$.

It is easy to observe that $\mathcal{M}$ is well defined and is indeed a basic structure which satisfies Γ.[5]

□

[5] We remark that, as observed in [167], basic frames and (Dis)-frames generate the same logic. This means that we could have written down an equivalent statement of the lemma by considering (Dis)-frames instead of basic frames. Indeed, one can notice that our construction in the proof of the lemma is such that we finally get a (Dis)-structure. This comes from the strategy adopted in curing branching defects, which consists here in adding a new point in any case. Different strategies could be adopted. For example, we could cure branching defects by (i) linking (i.e. by $\equiv$-relating) the point where the defect arises to some other point (already present in the labeled structure) providing a counterexample to the defect, if such a point exists and (ii) adding a new point, only if such a point does not exist. In this way we would finally get a basic structure that does not necessarily enjoy the property (Dis).

6.3.2 Mosaics for the logic of (WDC)-frames

Here we show how to extend and modify the definitions of Section 6.3.1 for the logic resulting from considering (WDC)-frames.

First of all, we can enrich the definition of an SSM with a branching saturation condition that is the analogous of the property (WDC):

(SB2) if $M, M', M_0 \in \mathit{Points}(S)$, $M \sim_s M'$ and $(M_0, M) \in S$, then there exists $M'_0 \in \mathit{Points}(S)$ s.t. $M_0 \sim_s M'_0$ and $(M'_0, M') \in S$.

In building a structure from an SSM, now the idea is to consider also WDC-defects, i.e. triples of points in the labeled structure under-construction such that they violate the property WDC.

However, the addition of (SB2) is not sufficient, as shown by the following example. Let x, y and y' be three points representing a counterexample to the property WDC in our labeled structure, i.e. $x < y \equiv y'$ but there is no x' in the structure such that $x \equiv x' < y'$. If $\mathcal{L}(x)$, $\mathcal{L}(y)$ and $\mathcal{L}(z)$ are the sets of formulas labeling the points x, y and z, respectively, then, by construction, there must be a mosaic $(\mathcal{L}(x), \mathcal{L}(y))$ and a point $\mathcal{L}(z)$ in our SSM. By exploiting the condition (SB2), we would be able to add a point x' in the structure such that $x \equiv x' < y'$ holds and that $(\mathcal{L}(x'), \mathcal{L}(y'))$ is a mosaic in the SSM. However this could violate connectedness of the relation $<$: in fact, the structure we are building could contain a point $z < y'$ such that $\mathcal{L}(z) \neq \mathcal{L}(x')$ and none of $z < x'$ and $x' < z$ is coherent.

We can solve this problem by forcing an SSM to satisfy stronger conditions. The addition of the following saturation condition allows us to retrieve connectedness:

(SL5) if $(M_0, M_1), (M'_0, M_1) \in S$ and $M_0 \neq M'_0$, then either $(M_0, M'_0) \in S$ or $(M'_0, M_0) \in S$.

Definition 6.15. *A set S of mosaics is a* (WDC)-saturated set of mosaics *(a (WDC)-SSM for short) if it satisfies the following* saturation conditions.

SATURATION CONDITIONS

For every mosaic $(M_0, M_1) \in S$,

LINEAR CONDITIONS

(SL1), (SL3), (SL2) and (SL4) as defined in Definition 6.13;
(SL5) as defined above;

BRANCHING CONDITIONS

(SB1) as in Definition 6.13;
(SB2) as defined above.

Given a (WDC)-SSM S and a set of formulas Γ, we say that S is a (WDC)-SSM for Γ *if there exists $M \in \mathit{Points}(S)$ such that $\Gamma \subseteq M$.*

Theorem 6.16. *For any set Γ of formulas, Γ is (WDC)-satisfiable iff there exists a (WDC)-SSM for Γ.*

Proof. ($\Rightarrow$) As in the proof of Theorem 6.14.
($\Leftarrow$) By a limit construction, as in the proof of Theorem 6.14 with some adaptations. In particular, we consider now also (WDC)-defects and cure them by exploiting conditions (SL5) and (SB2) on (WDC)-SSMs.

□

6.3.3 Mosaics for the logic of (Dis+WDC)-frames

We recall from Section 6.3.1 the notions of *mosaic*, *state-equivalence* and *points* of a set of mosaics. A definition of (Dis+WDC)-saturated sets of mosaics is introduced in the following. The linear saturation conditions are analogous to the ones given for the logic of basic frames in Definition 6.13. With regard to the branching conditions, we recall SB1 and SB2 from Definition 6.15 for the logic of (WDC)-frames. We need to add a further branching condition, denoted with SB3 below, which can be seen as corresponding to the property SDC on frames (see Section 2.4.1). In fact, we know from Lemma 2.22 that the logic of (Dis+WDC)-frames and the logic of (WDC+SDC)-frames coincide. We remark that after the addition of condition SB3 we do not longer need condition SL5 of Section 6.3.2, since WDC+SDC imply the linearity of the relation $\prec$; more details in the proof of Theorem 6.18.

Definition 6.17. *A set S of mosaics is a* (Dis+WDC)-saturated set of mosaics *(a (Dis+WDC)-SSM for short) if it satisfies the following* saturation conditions.

SATURATION CONDITIONS

For every mosaic $(M_0, M_1) \in S$,

LINEAR CONDITIONS
(SL1), (SL3), (SL2) and (SL4) as defined in Definition 6.13;

BRANCHING CONDITIONS

(SB1) and (SB2) as defined above;

(SB3) let M_0, M_1, M_2, M_0' and M_2' be points of S s.t.
(i) $M_0 \sim_s M_0'$;
(ii) $M_2 \sim_s M_2'$; and
(iii) $(M_0, M_1), (M_1, M_2) \in S$;
then there exists $M_1' \in Points(S)$ s.t. $M_1 \sim_s M_1'$ and $(M_0', M_1'), (M_1', M_2') \in S$.

Given a (Dis+WDC)-SSM S and a set of formulas Γ, we say that S is a (Dis+WDC)-SSM for Γ *if there exists $M \in Points(S)$ such that $\Gamma \subseteq M$.*

Now we present the key theorem concerning mosaics for the logic of (Dis+WDC)-frames. In Section 6.3.2, when we sketched the analogous theorem for the (WDC)-logic, we suggested considering three classes of defects: linear, branching and WDC. In this case, it seems convenient to move back to the classification of Section 6.3.1, distinguishing only between linear and branching defects. Possible WDC-defects and SDC-defects are cured "indirectly" when we treat branching defects. Namely,

when we add a new vertical line, we make sure that we add all the points necessary for letting the structure enjoy the properties (WDC) and (SDC). Conditions SB2 and SB3 ensure that this is always possible.

Theorem 6.18. *For any set Γ of formulas, Γ is (Dis+WDC)-satisfiable iff there exists a (Dis+WDC)-SSM for Γ.*

Proof. ($\Rightarrow$) Let $\mathcal{M} = (\mathcal{T}, \prec, \simeq, \mathcal{V})$ be a (Dis+WDC)-structure satisfying Γ and let $u \in \mathcal{T}$ be a point such that $\mathcal{M}, u \models \Gamma$. Given a set Δ', which contains Γ and is closed under subformulas and single negations, we can associate a different *fresh atom*, i.e. an atom that is not in Δ', to each $\simeq$-equivalence class. Let Δ'' be the set containing such atoms and their negations and $\Delta = \Delta' \cup \Delta''$. We associate a subset of Δ to every point of $\mathcal{T}$ as follows: for every $v \in \mathcal{T}$ we define $M_v = \{A \in \Delta' : \mathcal{M}, v \models A\} \cup \{p_v\} \cup \{\neg p : p \in \Delta'' \text{ and } p \neq p_v\}$, where p_v is the atomic proposition associated to the equivalence class $[v]$. Then we define the set $S = \{(M_v, M_{v'}) : v, v' \in \mathcal{T} \text{ and } v \prec v'\} \cup \{(M_v) : v \in \mathcal{T} \text{ and for all } v' \in \mathcal{T} \text{ we have } v \not\prec v' \text{ and } v' \not\prec v\}$. It is easy to verify that every element of S is indeed a mosaic and that the set S is an SSM. In fact coherence and saturation conditions are clearly satisfied since the definition of each point in S comes from the labeling of the corresponding point in an (Dis+WDC)-structure. In particular, the use of fresh atoms ensures that points of $\mathcal{T}$ that are state-equivalent but not $\simeq$-equivalent give rise to distinct points in S and thus that the saturation conditions (SB2) and (SB3) are satisfied by S. Furthermore S is an SSM for Γ since $\Gamma \subseteq M_u$ and $M_u \in \mathit{Points}(S)$.

($\Leftarrow$) As in the case of basic frames, we will build a model for Γ step by step by using the mosaics in S as building blocks. We recall from the proof of Theorem 6.14 the notions of (coherent) labeled structure and (linear and branching) defect, set an enumeration D of all the possible defects and a scheduling function $\sigma : \omega \to \omega$ such that, for every $j \in \omega$, there are infinitely many k such that $\sigma(k) = j$.

Our construction is such that at every step $n < \omega$ we will have a labeled structure $L_n = (H_n, \{V_{h_n}, <_{h_n}\}_{h \in H_n}, \equiv_n, \mathcal{L}_n)$ satisfying the following *formation conditions*:

(F1) L_n is coherent;
(F2) for every $h \in H_n$, $(V_{h_n}, <_{h_n})$ determines a finite linear order of rational numbers $\langle i_{0_h} < i_{1_h} < \ldots < i_{k_h} \rangle$ such that, for every j, $(\mathcal{L}_n(h, i_{j_h}), \mathcal{L}_n(h, i_{j+1_h}))$ is a mosaic in S;
(F3) if $(h, v) \equiv_n (h', v)$ then $\mathcal{L}_n(h, v)$ and $\mathcal{L}_n(h', v)$ are state-equivalent;
(F4) if $(h, v) \equiv_n (h', v)$ and (h, v') is a point of L_n for some $v' < v$, then there exists (h', v') in L_n such that $(h, v') \equiv_n (h', v')$;
(F5) if (h, v), (h, v'), (h, v''), (h', v) and (h', v'') are points in L_n such that $v < v' < v''$, $(h, v) \equiv_n (h', v)$ and $(h, v'') \equiv_n (h', v'')$, then there exists (h', v') in L_n such that $(h, v') \equiv_n (h', v')$.

Conditions (F1), (F2) and (F3) above are the same as in the proof of Theorem 6.14. (F4) and (F5) are, respectively, the analogous of the branching saturation conditions (SB2) and (SB3) of Definition 6.17.

In the following, we will describe our limit construction of a (Dis+WDC)-model for Γ.

[*STEP* 0] As in the proof of Theorem 6.14.

[*STEP* $n+1$] Assume that we have already defined a labeled structure L_n satisfying the formation conditions. Then we consider the $\sigma(n+1)$-th defect d in our enumeration of D. If d is not an actual defect of L_n, then we just set $L_{n+1} = L_n$. Otherwise we have three cases:

(i) $d = \langle (h, v), \mathsf{F}A \rangle$ is a linear future defect. Then let v' be the greatest element of V_{h_n} with respect to the order $<_{h_n}$ such that $\mathsf{F}A \in (h, v')$. Since d is an actual defect of L_n, such v' exists. We have two subcases:

(a) v' is the greatest element of V_{h_n} according to $<_{h_n}$. Then by the saturation condition (SL1), there is a mosaic (M'_0, M'_1) in S such that $M'_0 = \mathcal{L}_n(h, v')$ and $A \in M'_1$. We add a new element $(v'+1)$ to V_{h_n} and define $<_{h_{n+1}}$ as the restriction to $V_{h_{n+1}}$ of the usual order $<$ on rational numbers. Formally, we define:

- $H_{n+1} = H_n$;
- $V_{h_{n+1}} = V_{h_n} \cup \{v'+1\}$;
- $\mathcal{L}_{n+1}(h, v'+1) = M'_1$ for an M'_1 obtained as described above;
- $V_{i_{n+1}} = V_{i_n}$ for every $i \in H_{n+1}$ such that $i \neq h$;
- $<_{i_{n+1}}$ is, for every $i \in H_{n+1}$, the restriction to $V_{i_{n+1}}$ of the usual order $<$ on rational numbers;
- $\equiv_n = \equiv_{n+1} \cup \{((h, v'+1), (h, v'+1))\}$;
- $\mathcal{L}_{n+1}(i, j) = \mathcal{L}_n(i, j)$ for every point (i, j) of L_n.

(b) v' is not the greatest element of V_{h_n}. Then there exists an element $v'' \in V_{h_n}$ such that v'' is the immediate successor of v', according to the relation $<_{h_n}$, and, by the maximality of v', $\neg\mathsf{F}A \in \mathcal{L}_n(h, v'')$. By the condition (SL3), there exist two mosaics (M_0, M), $(M, M_1) \in S$ such that $M_0 = \mathcal{L}_n(h, v')$, $M_1 = \mathcal{L}_n(h, v'')$ and $A \in M$. Then we insert a point v^* between v' and v'' and label (h, v^*) with M. In order to let L_{n+1} satisfy the formation condition (F5), in this case we need also to consider all the points of L_n that are $\equiv_n$-related to (h, v''). Let (h', v'') be one such point. The formation conditions on L_n ensure that there exist two points $M'_0 = \mathcal{L}_n(h', v')$ and $M'_1 = \mathcal{L}_n(h', v'')$ in $\mathit{Points}(S)$ such that $M_0 \sim_s M'_0$, $M_1 \sim_s M'_1$ and $(M_0, M_1) \in S$. Furthermore, by the saturation condition (SB3) on S, there exists $M' \in \mathit{Points}(S)$ such that $M \sim_s M'$, $(M'_0, M') \in S$ and $(M', M'_1) \in S$. Then we add v^* to the set $V_{h'_n}$ and label it with M'.

By summing up, we define L_{n+1} as follows:

- $H_{n+1} = H_n$;
- $V_{h_{n+1}} = V_{h_n} \cup \{v^*\}$, where v^* is a rational number such that $v' < v^* < v''$;
- $\mathcal{L}_{n+1}(h, v^*) = M$ where M is obtained as described above;
- for every $i \in H_{n+1}$ such that $i \neq h$, if (i, v'') is a point of L_n and $(h, v'') \equiv_n (i, v'')$, then $V_{i_{n+1}} = V_{i_n} \cup \{v^*\}$ and $\mathcal{L}_{n+1}(i, v^*) = M'$ for a set $M' \sim_s M$ obtained as described above; otherwise $V_{i_{n+1}} = V_{i_n}$;
- $<_{i_{n+1}}$ is, for every $i \in H_{n+1}$, the restriction to $V_{i_{n+1}}$ of the usual order $<$ on rational numbers;

- $\equiv_{n+1}$ is the transitive closure of $\equiv_n \cup\{((h_1,\ v_1),\ (h_2, v_2))\ \mid\ (h_1, v_1)$ and (h_2, v_2) are (not necessarily distinct) points of L_{n+1} but not of $L_n\}$;
- $\mathcal{L}_{n+1}(i, j) = \mathcal{L}_n(i, j)$ for every point (i, j) of L_n.

(ii) $d = \langle (h, v), \mathsf{P}A\rangle$ is a linear past defect. The treatment of such defects is symmetrical to that of future defects, though some subtleties need to be taken into account. Let v' be the lowest element of V_{h_n} with respect to the order $<_{h_n}$ such that $\mathsf{P}A \in (h, v')$. Since d is an actual defect of L_n, such v' exists. We have two subcases:

(a) v' is the lowest element of V_{h_n} according to $<_{h_n}$. Then by the saturation condition (SL2) there is a mosaic (M'_0, M'_1) in S such that $M'_1 = \mathcal{L}_n(h, v')$ and $A \in M'_0$. We add a new element $(v' - 1)$ to V_{h_n} and define $<_{h_{n+1}}$ as the restriction to $V_{h_{n+1}}$ of the usual order $<$ on rational numbers. Unlike the symmetrical case concerning linear future defects treated above, here we need also to ensure that the formation condition (F4) is satisfied. Namely, let (h', v') be a point in L_n such that $(h, v') \equiv_n (h', v')$. Then, by the formation condition (F3), there exists a point $M''_1 = \mathcal{L}_n(h', v') \in \mathit{Points}(S)$ such that $M''_1 \sim_s M'_1$ and, by the saturation condition (SB2) on S, there exists a mosaic (M''_0, M''_1) such that $M''_0 \sim_s M'_0$. Then we add $(v' - 1)$ to $V_{h'_{n+1}}$, set $lab_{n+1}(h', v' - 1) = M''_0$ and put $(h', v' - 1) \equiv_{n+1} (h, v' - 1)$. By summing up, we have:
- $H_{n+1} = H_n$;
- $V_{h_{n+1}} = V_{h_n} \cup \{v' - 1\}$;
- $\mathcal{L}_{n+1}(h, v' - 1) = M'_0$ for an M'_0 obtained as described above;
- for every $i \in H_{n+1}$ such that $i \neq h$, if (i, v') is a point of L_n and $(h, v') \equiv_n (i, v')$, then $V_{i_{n+1}} = V_{i_n} \cup \{v' - 1\}$ and $\mathcal{L}_{n+1}(i, v' - 1) = M''_0$ for an $M''_0 \sim_s M'_0$ obtained as described above; otherwise $V_{i_{n+1}} = V_{i_n}$;
- for every $i \in H_{n+1}$, $<_{i_{n+1}}$ is the restriction to $V_{i_{n+1}}$ of the usual order $<$ on rational numbers;
- $\equiv_{n+1}$ is the transitive closure of $\equiv_n \cup\{((h_1,\ v_1),\ (h_2, v_2))\ \mid\ (h_1, v_1)$ and (h_2, v_2) are (not necessarily distinct) points of L_{n+1} but not of $L_n\}$;
- $\mathcal{L}_{n+1}(i, j) = \mathcal{L}_n(i, j)$ for every point (i, j) of L_n.

(b) v' is not the lowest element of V_{h_n}. Then there exists an element $v'' \in V_{h_n}$ such that v'' is the immediate predecessor of v', according to the relation $<_{h_n}$, and, by the maximality of v', $\neg\mathsf{P}A \in \mathcal{L}_n(h, v'')$. By the condition (SL4), there exist two mosaics (M_0, M), $(M, M_1) \in S$ such that $M_0 = \mathcal{L}_n(h, v'')$, $M_1 = \mathcal{L}_n(h, v')$ and $A \in M$. Then we insert a point v^* between v'' and v' and label (h, v^*) with M. In order to let L_{n+1} satisfy the formation condition (F5), we need to consider all the points of L_n that are $\equiv_n$-related to (h, v'). Let (h', v') be one such point. The formation conditions on L_n ensure that there exist two points $M'_0 = \mathcal{L}_n(h', v'')$ and $M'_1 = \mathcal{L}_n(h', v')$ in $\mathit{Points}(S)$ such that $M_0 \sim_s M'_0$, $M_1 \sim_s M'_1$ and $(M_0, M_1) \in S$. Furthermore, by the saturation condition (SB3) on S, there exists $M' \in \mathit{Points}(S)$ such that $M \sim_s M'$, $(M'_0, M') \in S$ and $(M', M'_1) \in S$. Then we add v^* to the set $V_{h'_n}$ and label it with M'.
By summing up, we define L_{n+1} as follows:

- $H_{n+1} = H_n$;
- $V_{h_{n+1}} = V_{h_n} \cup \{v^*\}$, where v^* is a rational number such that $v'' < v^* < v'$;
- $\mathcal{L}_{n+1}(h, v^*) = M$ where M is obtained as described above;
- for every $i \in H_{n+1}$ such that $i \neq h$, if (i, v') is a point of L_n and $(h, v') \equiv_n (i, v')$, then $V_{i_{n+1}} = V_{i_n} \cup \{v^*\}$ and $\mathcal{L}_{n+1}(i, v^*) = M'$ for a set $M' \sim_s M$ obtained as described above; otherwise $V_{i_{n+1}} = V_{i_n}$;
- for every $i \in H_{n+1}$, $<_{i_{n+1}}$ is the restriction to $V_{i_{n+1}}$ of the usual order $<$ on rational numbers;
- $\equiv_{n+1}$ is the transitive closure of the set $\equiv_n \cup\{((h_1, v_1), (h_2, v_2)) \mid (h_1, v_1)$ and (h_2, v_2) are (not necessarily distinct) points of L_{n+1} but not of $L_n\}$;
- $\mathcal{L}_{n+1}(i, j) = \mathcal{L}_n(i, j)$ for every point (i, j) of L_n.

(iii) $d = \langle (h, v), \exists A \rangle$ is a branching defect. By the saturation condition SB1, we know that there exists $M' \in \mathit{Points}(S)$ such that $\mathcal{L}_n(h, v) \sim_s M'$ and $A \in M'$. Then we add a new vertical line (with a fresh index, say $n + 1$) consisting of a single element (with index v) labeled with M', i.e. we add a new point $(n+1, v)$ to L_{n+1}, and set $(h, v) \equiv_{n+1} (n+1, v)$. We will possibly need to add some further points in order to let L_{n+1} satisfy the formation condition (F4). Namely, if L_n contains some point below (h, v), then the idea consists in enriching the labeled structure by adding below $(n + 1, v)$ a linearly ordered set of points isomorphic to the set of predecessors of (h, v) and such that all the corresponding points are state-equivalent. We proceed as follows. Let (h, v') be the point in L_n that is the immediate predecessor of (h, v) according to $<_{h_n}$. By the formation condition (F2), $(\mathcal{L}_n(h, v'), \mathcal{L}_n(h, v))$ is a mosaic in S. Then the saturation condition (SB2) on S ensures that there exists a mosaic $(M, M') \in \mathit{Points}(S)$ such that $M \sim_s \mathcal{L}_n(h, v')$ and thus, by the formation condition (F3), such that $M \sim_s \mathcal{L}_n(h', v')$ for each $(h', v') \equiv_n (h, v')$. Then we add v' to $V_{n+1_{n+1}}$ and set $\mathcal{L}_{n+1}(n + 1, v') = M$ and $(n + 1, v') \equiv_{n+1} (h', v')$ for each $(h', v') \equiv_n (h, v')$. Then we consider the immediate predecessor (h, v'') of (h, v') and repeat the same procedure with respect to these two points. Then again with respect to (h, v'') and its predecessor and so on. By summing up, we define L_{n+1} as follows:

- $H_{n+1} = H_n \cup \{n + 1\}$;
- $V_{i_{n+1}} = V_{i_n}$ for every $i \in H_n$;
- $V_{n+1_{n+1}} = \{\overline{v} \mid \overline{v} \in V_{h_n}$ and $\overline{v} < v\}$;
- for every $i \in H_{n+1}$, $<_{i_{n+1}}$ is the restriction to $V_{i_{n+1}}$ of the usual order $<$ on rational numbers;
- $\mathcal{L}_{n+1}(i, j) = \mathcal{L}_n(i, j)$ for every point (i, j) of L_n.
- $\mathcal{L}_{n+1}(n + 1, v) = M'$, where M' is obtained as described above;
- for every $\overline{v} \in V_{n+1_{n+1}}$ such that $\overline{v} \neq v$, $\mathcal{L}_{n+1}(n + 1, \overline{v}) = M$ for a set $M \sim_s \mathcal{L}_n(h, \overline{v})$ obtained as described above;
- $\equiv_{n+1}$ is the reflexive, symmetric and transitive closure of the set $\equiv_n \cup\{((h, \overline{v}), (n + 1, \overline{v})) \mid \overline{v} \in V_{n+1_{n+1}}\}$.

The construction is such that in all the cases we get a labeled structure L_{n+1} which satisfies formation conditions and where d is no longer a defect. As in the proof of Theorem 6.14, we have that the new labeling $\mathcal{L}_{n+1}$ is just an extension

of the old $\mathcal{L}_n$ and that the defect d (once cured) cannot occur in any expansion of the structure.

[*STEP* ω] We take the union $L = (H, \{V_h, <_h\}_{h\in H}, \equiv, \mathcal{L})$ of the labeled structures defined so far and define a structure $\mathcal{M} = (\mathcal{T}, \prec, \simeq, \mathcal{V})$ such that:

1. $\mathcal{T} = \{u : u \text{ is a point of } L\}$;
2. $\prec = \bigcup_{h\in H} <_h$;
3. $\simeq = \equiv$;
4. for all $u \in \mathcal{T}$, $p \in \mathcal{V}(u)$ iff $p \in \mathcal{L}(u)$.

By construction, $\mathcal{M}$ is a (Dis+WDC)-structure that satisfies Γ.

□

6.3.4 Mosaics for the logic *BOBTL* of Ockhamist frames

The definition of (Dis+WDC)-SSM is still not strong enough in order to get an Ockhamist structure. What we still miss is the property of maximality of branches (MB), which in our case can also be expressed (see Lemma 2.22) by the conditions (MB$^-$) or (MB^{--}).

In the second part of the proof of Theorem 6.18, we used the mosaics contained in a (Dis+WDC)-SSM to build a (Dis+WDC)-structure. If we are interested in building an Ockhamist structure, we need a way to ensure that a $<$-maximal point of a vertical line is $\equiv$-related only to $<$-maximal points.

It is enough to add a branching coherence condition to the definition of a mosaic.

Definition 6.19. *Let Δ be a set of formulas closed under subformulas and single negation, in the language of Ockhamist formulas. An* (MB)-mosaic *(on Δ) is a mosaic (M_0, M_1) or (M_0) on Δ such that the following condition holds:*

(CB2) Let $i \in \{0, 1\}$. If for all $\mathsf{F}A \in \Delta$, $\mathsf{F}A \notin M_i$, then for all $\exists A \in M_i$, $A \in M_i$.

Definition 6.20. *An* Ockhamist SSM *is a set of (MB)-mosaics satisfying the conditions (SL1), (SL2), (SL3), (SL4), (SB1), (SB2) and (SB3), where in each condition (MB)-mosaics replace mosaics.*

Given a set Γ of branching formulas, an Ockhamist SSM is an Ockhamist SSM *for Γ if there exists $M \in Points(S)$ such that $\Gamma \subseteq M$.*

Theorem 6.21. *For any set Γ of Ockhamist formulas, Γ is Ockhamist-satisfiable iff there exists an Ockhamist SSM for Γ.*

Proof. ($\Rightarrow$) As in the proof of Theorem 6.18.
($\Leftarrow$) The construction of a structure from the SSM mirrors that of the proof of Theorem 6.18. The condition CB2 ensures that if we have a point where no future defects can occur, then at that point also the occurrence of branching defects is excluded. It follows that, given a $<$-maximal point, the construction will not generate for it any $\equiv$-related point distinct from itself.

□

6.3.5 Discussion

Related works concerning the use of the mosaic method in temporal logics have been already described in Section 6.1. Most of such works present definitions and techniques for linear tense logics. Our contribution consists in the extension of such techniques (in particular of those presented in [105]) to the case of the bundled branching Ockhamist logic *BOBTL* [167] and some of its sublogics.

The extension is mainly based on the fact that $\simeq$-related points in a (possibly generalized) Ockhamist structure satisfy the same set of atomic propositions and thus the same set of state formulas. The saturation conditions of the linear case are enriched with a further condition requiring that if a point M in the set of mosaics contains a formula of the form $\exists A$, then a point M' state-equivalent to M, i.e. satisfying the same set of state formulas, and containing A must also be in the set. Such a condition allows for capturing the so-called logic of basic frames [167]. Further refinements of the definition of a saturated set of mosaics are required in order to consider *BOBTL* and other intermediate logics.

In this section, we have focused on providing proper definitions of mosaics and saturated sets of mosaics for the case considered and on proving the key theorem relating the satisfiability of a set of formulas to the existence of a saturated set of mosaics. An analysis of possible applications is left for future work; here we just sketch some ideas concerning the use of mosaics in proving completeness of a Hilbert-style axiomatization and decidability.

With regard to completeness, we notice that the use of mosaics allows for simplifying the standard proofs [167] of completeness of Hilbert-style axiomatizations for these logics. Such proofs consist in considering maximal consistent sets and defining two relations $\prec^M$ and $\simeq^M$ on them, based on the formulas they contain, i.e.,

$$\Gamma \prec^M \Delta \text{ iff } \{A \mid \mathsf{G}A \in \Gamma\} \subseteq \Delta \quad , \quad \Gamma \simeq^M \Delta \text{ iff } \{A \mid \forall A \in \Gamma\} \subseteq \Delta \, .$$

The idea is that such relations can be used as the basis for building a structure by a procedure of elimination of counterexamples [32,33,167]. If we use mosaics, then part of this procedure is already contained in the theorems of Sections 6.3.1-6.3.4 and it suffices to show that the set of all pairs (M_1, M_2) such that M_1 and M_2 are maximal consistent sets and $M_1 \prec^M M_2$ form a saturated set of mosaics[6]

Particular attention is required in the case of (Dis+WDC) and Ockhamist frames, since the property (SDC) fails in the set of maximal consistent sets for the corresponding axiomatizations. However we believe that techniques analogous to those described in [164] for proving completeness should help prove that a saturated set of mosaics can be retrieved from a set of pairs of maximal consistent sets defined as above.

By adapting the considerations above, we observe that a proof of completeness for the natural deduction systems defined in the previous chapters could also be obtained via mosaics.

With regard to decidability, we notice that decidability of the logics considered in this section follows from the results of Burgess in [31] (see also [68]). It should

[6] Note that we do not need to consider the relation $\simeq^M$ explicitly since we treat branching counterexamples by using the notion of state-equivalence.

be possible to give a proof of decidability via mosaics (as in Theorem 6.9) by considering that the set of subformulas and single negations of a given formula is finite and that checking saturation conditions on a finite set is decidable. A detailed treatment is left for future work.

7

Conclusions

7.1 Summary of contributions

Despite the fact that temporal logics have been studied for decades and despite their great relevance in many applications of computer science, their theoretical analysis is far from being concluded. In particular, we believe that we still lack a satisfactory proof-theoretical analysis for temporal logics.

The main contribution of this thesis is in the presentation of an approach for the definition of modular natural deduction systems for a large class of, both linear and branching, temporal logics and in their proof-theoretical analysis. Our approach is based on the framework of labeling, which has been successfully employed in the case of proof theory for modal, and in general non-classical, logics.

We started by defining a basic system for the minimal tense logic Kt and, by modular enrichments of the system, we have been able to capture other more complex logics, like the linear tense logic Kl, some of its variants, and finally the until-free fragment of LTL.

The extension to the branching case is limited to the so-called bundled branching logics, obtained by a generalization of the standard semantics for CTL^* or for its corresponding general-time logic. The semantics of bundled logics can be formulated in terms of Ockhamist frames [139, 167] rather than tree-like frames. Ockhamist frames allow for the definition of a pure Kripke-style semantics, where also the path quantifier $\forall$ can be seen as a modal operator, endowed with a proper accessibility relation. As a consequence, we have that we are able to exploit the well-known good behavior of labeled deduction systems for modal logics also in the case of such branching-time logics.

The modularity of the approach is in the fact that each connective (operator, quantifier) has its own accessibility relation, its own rules for defining the properties of such a relation and its own rules for introduction and elimination. Possible interactions between the relations are managed by means of rules not involving the operators themselves, whose introduction and elimination is restricted to the specific rules.

The result is a clean natural deduction system, for which it is possible to define a procedure of normalization. In particular, we have studied normalization in the case of the system for $BCTL^*_{\_}$, where the presence of a rule for induction makes an

analogy with systems for Peano/Heyting Arithmetic. We have proved a result of weak normalization and obtained a syntactical proof of consistency as a corollary.

The proof-theoretical analysis has mainly focused on systems for until-free logics. In fact, until is a very complex operator from a proof-theoretical point of view. In this thesis, we have proposed a solution for its treatment, which is based on the usage of a labeling discipline different from the most standard one and on replacing the until with a new operator, which is easier to treat and in terms of which the until can be defined.

Finally, we have proposed an extension of the mosaic method, presented in the literature [105] in the case of several (non-discrete) linear temporal logics to the corresponding bundled branching logics. The mosaic method can be used for proving decidability, complexity results or completeness of Hilbert-style axiomatizations of a given logic.

7.2 Future work

As usual, much is still to be done.

The most complex, and at the same time most stimulating, direction is represented by an extension towards the "full semantics" branching-time logics, *OBTL* and CTL^*. We recall that such logics represent a, partially still, open problem even when considering Hilbert-style axiomatizations [135, 136]. A first step could consist in providing a system with an infinitary rule, able to capture the so-called limit-closure property.

It is interesting to observe that, if we add past operators to CTL^*, then we get a more expressive logic for which the definition of a standard and complete Hilbert-style axiomatization is easier and has been in fact given by Reynolds [138]. Considering an extension of our system to deal with such a logic is another possible direction of research. The definition of a system for CTL^* with past could shed some light to the case of standard CTL^* as well.

In this thesis, we dealt with Ockhamist branching-time logics, whose language allows for a free combination of quantifiers and operators. We note anyway that Peircean logics, like *CTL*, can be obtained by the Ockhamist ones by just imposing a restriction on the language. Thus our systems can be also used for reasoning on Peircean logics, e.g., by considering a restriction on the set of admissible derivations. Although our approach, based on a strict separation between the operators, seems to lead more naturally to work with the language of Ockhamist logics, it would be interesting to consider possible adaptations explicitly designed for *CTL*-like logics, as in such cases it is also typically less complex to capture the full semantics.

We also plan to extend our work towards the investigation of practical applications of our systems. In particular, we believe that the one for $BCTL^*_{\_}$ can be interesting to reason about fairness, along the lines of [42, 63]. To that end, it will be especially important to mechanize reasoning as much as possible by providing automated reasoning procedures or employing interactive theorem provers, e.g. encoding our systems into a logical framework such as Isabelle [121, 122].

With regard to our proposal for the treatment of until, we notice that here we used the logic based on the new operator *history* mainly as a service-logic for

reasoning on the standard logic with until. Future work will be oriented towards an analysis of the real "meaning" and expressiveness of the new operator. Furthermore, although the introduction of history has been motivated by proof-theoretical considerations and we expect such an operator to be rather well-behaved, a detailed analysis of normalization for history-based logics has been left for future work.

Finally, the extension of the technique of mosaics to the case of the bundled branching Ockhamist logic *BOBTL* can be seen as just a first step towards a more general definition of the method in the context of other, more interesting and complex, logics, for which decidability and complexity results are still missing. In this thesis, we have proved completeness of the deduction systems indirectly by exploiting given Hilbert-style axiomatizations for the same logics (with the only exception of Section 4.3). We believe that a direct proof for (some of) the natural deduction systems defined here could be provided by using the mosaic method, thus creating also a stronger connection between the two tracks of this thesis.

A

Appendix

A.1 Proofs of Chapter 5

The Church-Rosser property

Lemma A.1. *Let* Π_1 *and* Π_2 *be two marked derivations such that* $\begin{matrix}\Pi_1\\ b_1 : A\end{matrix} \rightarrowtail_1 \begin{matrix}\Pi_1'\\ b_1 : A\end{matrix}$ *and* $\begin{matrix}b_1 : A\\ \Pi_2\\ b_2 : B\end{matrix} \rightarrowtail_1 \begin{matrix}b_1 : A\\ \Pi_2'\\ b_2 : B\end{matrix}$. *Then* $\Pi = \begin{matrix}\Pi_1\\ b_1 : A\\ \Pi_2\\ b_2 : B\end{matrix} \rightarrowtail_1 \Pi' = \begin{matrix}\Pi_1'\\ b_1 : A\\ \Pi_2'\\ b_2 : B\end{matrix}$ *and* $\delta(\Pi, \Pi') = \delta(\Pi_1, \Pi_1') \cup \delta(\Pi_2, \Pi_2')$.

Proof. The proof proceeds by induction on the definition of $\Pi_2 \rightarrowtail_1 \Pi_2'$.

(*i*) [*BC*]
If $\Pi_2 = \Pi_2'$, then we easily obtain the thesis by using the passive clauses in the definition of $\rightarrowtail_1$.

(*ii*) [$\supset I$]
Let

$$\Pi_2 = \dfrac{\begin{matrix}b_1 : A & [b_2 : B_1]^1\\ \Pi_3\\ b_2 : B_2\end{matrix}}{b_2 : B_1 \supset B_2}\ \supset I^1 \quad \rightarrowtail_1 \quad \Pi_2' = \dfrac{\begin{matrix}b_1 : A & [b_2 : B_1]^1\\ \Pi_3'\\ b_2 : B_2\end{matrix}}{b_2 : B_1 \supset B_2}\ \supset I^1 \ ,$$

where $B = B_1 \supset B_2$ and $\Pi_3 \rightarrowtail_1 \Pi_3'$. Then, by the induction hypothesis:

$$\widehat{\Pi_3} = \begin{matrix}\Pi_1\\ b_1 : A \quad b_2 : B_1\\ \Pi_3\\ b_2 : B_2\end{matrix} \quad \rightarrowtail_1 \quad \widehat{\Pi_3}' = \begin{matrix}\Pi_1'\\ b_1 : A \quad b_2 : B_1\\ \Pi_3'\\ b_2 : B_2\end{matrix} \ ,$$

where $\delta(\widehat{\Pi_3}, \widehat{\Pi_3}') = \delta(\Pi_1, \Pi_1') \cup \delta(\Pi_3, \Pi_3')$. By Definition 5.32, we conclude:

$$\Pi = \dfrac{\begin{matrix}\Pi_1\\ b_1 : A & [b_2 : B_1]^1\\ \Pi_3\\ b_2 : B_2\end{matrix}}{b_2 : B_1 \supset B_2}\ \supset I^1 \quad \rightarrowtail_1 \quad \Pi' = \dfrac{\begin{matrix}\Pi_1'\\ b_1 : A & [b_2 : B_1]^1\\ \Pi_3'\\ b_2 : B_2\end{matrix}}{b_2 : B_1 \supset B_2}\ \supset I^1 \ ,$$

where

$$\begin{array}{rcl}\delta\left(\Pi,\Pi'\right) & = & \delta\left(\widehat{\Pi_3},\widehat{\Pi_3}'\right) = \\ & = & \delta\left(\Pi_1,\Pi_1'\right)\cup\delta\left(\Pi_3,\Pi_3'\right) = \\ & = & \delta\left(\Pi_1,\Pi_1'\right)\cup\delta\left(\Pi_2,\Pi_2'\right).\end{array}$$

$(iii-xv)$ Cases concerning the remaining passive clauses are treated as (ii) above.

(xvi) $[IndContr]$
Let

$$\Pi_2 = \left\{\begin{array}{c} b_1:A \qquad [c_0\leqslant c_i]^1 \quad [c_i:B]^1 \quad [c_i\lhd c_j]^1 \quad b_1:A \\ \Pi_3 \qquad\qquad\qquad\qquad \Pi_4 \\ \dfrac{c_0:B \quad c_0\leqslant b_2 \qquad\qquad\qquad\qquad c_j:B}{b_2:B}\,(r)^1 \end{array}\right.$$

$$\rightarrowtail_1$$

$$\Pi_2' = \left\{\begin{array}{c} b_1:A \\ \Pi_3' \\ c_0\leqslant c_{0(r)} \qquad c_0:B \qquad c_0\lhd c_{1(r)} \qquad b_1:A \\ \Pi_4'[c_0/c_i][c_1/c_j] \\ c_0\leqslant c_{1(r)} \qquad c_1:B \qquad c_1\lhd c_{2(r)} \qquad b_1:A \\ \Pi_4'[c_1/c_i][c_2/c_j] \\ c_2:B \\ \vdots \\ c_0\leqslant c_{n-1(r)} \qquad c_{n-1}:B \qquad c_{n-1}\lhd b_{2(r)} \qquad b_1:A \\ \Pi_4'[c_{n-1}/c_i][b_2/c_j] \\ b_2:B \end{array}\right.,$$

where r is an application of ind, $\Pi_3\rightarrowtail_1\Pi_3'$ and $\Pi_4\rightarrowtail_1\Pi_4'$. Then, by the induction hypothesis:

$$\widehat{\Pi_3} = \begin{array}{c}\Pi_1\\ b_1:A\\ \Pi_3\\ c_0:B\end{array} \quad\rightarrowtail_1\quad \widehat{\Pi_3}' = \begin{array}{c}\Pi_1'\\ b_1:A\\ \Pi_3'\\ c_0:B\end{array},$$

where $\delta\left(\widehat{\Pi_3},\widehat{\Pi_3}'\right) = \delta\left(\Pi_1,\Pi_1'\right)\cup\delta\left(\Pi_3,\Pi_3'\right)$ and

$$\widehat{\Pi_4} = \begin{array}{c}\Pi_1\\ b_1:A \quad c_0\leqslant c_i \quad c_i:A \quad c_i\lhd c_j\\ \Pi_4\\ c_j:B\end{array}$$

$$\rightarrowtail_1$$

$$\widehat{\Pi_4}' = \begin{array}{c} \Pi_1' \\ b_1 : A \quad c_0 \leqslant c_i \quad c_i : A \quad c_i \lhd c_j \\ \Pi_4' \\ c_j : B \end{array} ,$$

where $\delta\left(\widehat{\Pi_4}, \widehat{\Pi_4}'\right) = \delta\left(\Pi_1, \Pi_1'\right) \cup \delta\left(\Pi_4, \Pi_4'\right)$. By Definition 5.32, we conclude:

$$\Pi = \left\{ \begin{array}{c} \begin{array}{ccccc} \Pi_1 & & & & \Pi_1 \\ b_1 : A & & [c_0 \leqslant c_i]^1 \quad [c_i : B]^1 \quad [c_i \lhd c_j]^1 & & b_1 : A \end{array} \\ \begin{array}{cccc} \Pi_3 & & & \Pi_4 \\ c_0 : B \quad c_0 \leqslant b_2 & & & c_j : B \end{array} \\ \hline b_2 : B \end{array} (r)^1 \right.$$

$$\rightarrowtail_1$$

$$\Pi' = \left\{ \begin{array}{c} \Pi_1' \\ b_1 : A \\ \begin{array}{cccc} \Pi_3' & & \Pi_1' & \\ c_0 \leqslant c_{0(r)} \quad c_0 : B & c_0 \lhd c_{1(r)} & b_1 : A & \end{array} \\ \Pi_4'[c_0/c_i][c_1/c_j] \\ \begin{array}{cccc} & & \Pi_1' & \\ c_0 \leqslant c_{1(r)} \quad c_1 : B & c_1 \lhd c_{2(r)} & b_1 : A & \end{array} \\ \Pi_4'[c_1/c_i][c_2/c_j] \\ c_2 : B \\ \vdots \\ \begin{array}{cccc} & & & \Pi_1' \\ c_0 \leqslant c_{n-1(r)} \quad c_{n-1} : B & c_{n-1} \lhd b_{2(r)} & & b_1 : A \end{array} \\ \Pi_4'[c_{n-1}/c_i][b_2/c_j] \\ b_2 : B \end{array} \right. ,$$

where

$$\begin{array}{rcl} \delta\left(\Pi, \Pi'\right) & = & \delta\left(\widehat{\Pi_3}, \widehat{\Pi_3}'\right) \cup \delta\left(\widehat{\Pi_4}, \widehat{\Pi_4}'\right) \cup \{r\} \; = \\ & = & \delta\left(\Pi_1, \Pi_1'\right) \cup \delta\left(\Pi_3, \Pi_3'\right) \cup \delta\left(\Pi_1, \Pi_1'\right) \cup \delta\left(\Pi_4, \Pi_4'\right) \cup \{r\} \; = \\ & = & \delta\left(\Pi_1, \Pi_1'\right) \cup \delta\left(\Pi_3, \Pi_3'\right) \cup \delta\left(\Pi_4, \Pi_4'\right) \cup \{r\} \; = \\ & = & \delta\left(\Pi_1, \Pi_1'\right) \cup \delta\left(\Pi_2, \Pi_2'\right) . \end{array}$$

(*xvii*) $[\supset I/\supset E]$
Let

$$\Pi_2 = \dfrac{\dfrac{\begin{array}{c} b_1 : A \\ \Pi_3 \\ b_2 : B_1 \end{array} \qquad \dfrac{\begin{array}{c}[b_2 : B_1]^1 \quad b_1 : A \\ \Pi_4 \\ b_2 : B \end{array}}{b_2 : B_1 \supset B}\supset I^1}{b_2 : B}\supset E \quad \rightarrowtail_1 \quad \Pi_2' = \begin{array}{c} b_1 : A \\ \Pi_3' \\ b_2 : B_1 \\ \Pi_4' \\ b_2 : B \end{array} \quad ,$$

where $\Pi_3 \rightarrowtail_1 \Pi_3'$ and $\Pi_4 \rightarrowtail_1 \Pi_4'$. Then, by the induction hypothesis:

$$\widehat{\Pi_3} = \begin{array}{c} \Pi_1 \\ b_1 : A \\ \Pi_3 \\ b_2 : B_1 \end{array} \quad \rightarrowtail_1 \quad \widehat{\Pi_3}' = \begin{array}{c} \Pi_1' \\ b_1 : A \\ \Pi_3' \\ b_2 : B_1 \end{array} \quad ,$$

where $\delta\left(\widehat{\Pi_3}, \widehat{\Pi_3}'\right) = \delta\left(\Pi_1, \Pi_1'\right) \cup \delta\left(\Pi_3, \Pi_3'\right)$ and

$$\widehat{\Pi_4} = \begin{array}{c} \Pi_1 \\ b_2 : B_1 \quad b_1 : A \\ \Pi_4 \\ b_2 : B \end{array} \quad \rightarrowtail_1 \quad \widehat{\Pi_4}' = \begin{array}{c} \Pi_1' \\ b_2 : B_1 \quad b_1 : A \\ \Pi_4' \\ b_2 : B \end{array} \quad ,$$

where $\delta\left(\widehat{\Pi_4}, \widehat{\Pi_4}'\right) = \delta\left(\Pi_1, \Pi_1'\right) \cup \delta\left(\Pi_4, \Pi_4'\right)$. By Definition 5.32, we conclude:

$$\Pi = \dfrac{\dfrac{\begin{array}{c} \Pi_1 \\ b_1 : A \\ \Pi_3 \\ b_2 : B_1 \end{array} \qquad \dfrac{\begin{array}{c} \Pi_1 \\ [b_2 : B_1]^1 \quad b_1 : A \\ \Pi_4 \\ b_2 : B \end{array}}{b_2 : B_1 \supset B}\supset I^1}{b_2 : B}\supset E \quad \rightarrowtail_1 \quad \Pi' = \begin{array}{c} \Pi_1' \\ b_1 : A \\ \Pi_3' \quad \Pi_1' \\ b_2 : B_1 \quad b_1 : A \\ \Pi_4' \\ b_2 : B \end{array} \quad ,$$

where

$$\begin{array}{rcl} \delta\left(\Pi, \Pi'\right) & = & \delta\left(\widehat{\Pi_3}, \widehat{\Pi_3}'\right) \cup \delta\left(\widehat{\Pi_4}, \widehat{\Pi_4}'\right) = \\ & = & \delta\left(\Pi_1, \Pi_1'\right) \cup \delta\left(\Pi_3, \Pi_3'\right) \cup \delta\left(\Pi_1, \Pi_1'\right) \cup \delta\left(\Pi_4, \Pi_4'\right) = \\ & = & \delta\left(\Pi_1, \Pi_1'\right) \cup \delta\left(\Pi_2, \Pi_2'\right) . \end{array}$$

(*xviii*) $[\wedge I / \wedge E_1]$
Let

$$\Pi_2 = \dfrac{\dfrac{\begin{array}{c} b_1 : A \\ \Pi_3 \\ b_2 : B \end{array} \quad \begin{array}{c} b_1 : A \\ \Pi_4 \\ b_2 : C \end{array}}{b_2 : B \wedge C}\wedge I}{b_2 : B}\wedge E_1 \quad \rightarrowtail_1 \quad \Pi_2' = \begin{array}{c} b_1 : A \\ \Pi_3' \\ b_2 : B \end{array} \quad ,$$

where $\Pi_3 \rightarrowtail_1 \Pi_3'$. Then, by the induction hypothesis:

$$\widehat{\Pi_3} = \begin{array}{c} \Pi_1 \\ b_1 : A \\ \Pi_3 \\ b_2 : B \end{array} \quad \rightarrowtail_1 \quad \widehat{\Pi_3}' = \begin{array}{c} \Pi_1' \\ b_1 : A \\ \Pi_3' \\ b_2 : B \end{array} \quad ,$$

where $\delta\left(\widehat{\Pi_3}, \widehat{\Pi_3}'\right) = \delta\left(\Pi_1, \Pi_1'\right) \cup \delta\left(\Pi_3, \Pi_3'\right)$. By Definition 5.32, we conclude:

$$\Pi = \dfrac{\dfrac{\begin{array}{c}\Pi_1\\ b_1 : A\end{array} \quad \begin{array}{c}\Pi_1\\ b_1 : A\end{array}}{\begin{array}{c}\Pi_3\\ b_2 : B\end{array} \quad \begin{array}{c}\Pi_4\\ b_2 : C\end{array}}\ \wedge I}{b_2 : B \wedge C}\ \wedge E_1 \quad \rightarrowtail_1 \quad \Pi' = \begin{array}{c}\Pi'_1\\ b_1 : A\\ \Pi'_3\\ b_2 : B\end{array} \quad ,$$

where

$$\begin{array}{rcl}\delta\,(\Pi, \Pi') & = & \delta\,(\widehat{\Pi_3}, \widehat{\Pi_3}') \ = \\ & = & \delta\,(\Pi_1, \Pi'_1) \cup \delta\,(\Pi_3, \Pi'_3) \ = \\ & = & \delta\,(\Pi_1, \Pi'_1) \cup \delta\,(\Pi_2, \Pi'_2)\ .\end{array}$$

(xix) $[\wedge I/\wedge E_2]$
Analogous to the previous case.

(xx) $[\mathsf{X}I/\mathsf{X}E]$
Let

$$\Pi_2 = \dfrac{\dfrac{\begin{array}{c}[b \lhd b']^1\\ b' : B\end{array} \quad b_1 : A}{\begin{array}{c}\Pi_3\\ b : \mathsf{X}B\end{array}\ \mathsf{X}I^1 \quad b \lhd b_2}\ \mathsf{X}E}{b_2 : B} \quad \rightarrowtail_1 \quad \Pi'_2 = \begin{array}{c}b \lhd b_2 \quad b_1 : A\\ \Pi'_3[b_2/b']\\ b_2 : B\end{array} \quad ,$$

where $\Pi_3 \rightarrowtail_1 \Pi'_3$. Then, by the induction hypothesis:

$$\widehat{\Pi_3} = \begin{array}{c}\Pi_1\\ b \lhd b' \quad b_1 : A\\ \Pi_3\\ b' : B\end{array} \quad \rightarrowtail_1 \quad \widehat{\Pi_3}' = \begin{array}{c}\Pi'_1\\ b \lhd b' \quad b_1 : A\\ \Pi'_3\\ b' : B\end{array} \quad ,$$

where $\delta\,(\widehat{\Pi_3}, \widehat{\Pi_3}') = \delta\,(\Pi_1, \Pi'_1) \cup \delta\,(\Pi_3, \Pi'_3)$. By Definition 5.32, we conclude:

$$\Pi = \dfrac{\dfrac{\begin{array}{c}[b \lhd b']^1\\ b' : B\end{array} \quad \begin{array}{c}\Pi_1\\ b_1 : A\end{array}}{\begin{array}{c}\Pi_3\\ b : \mathsf{X}B\end{array}\ \mathsf{X}I^1 \quad b \lhd b_2}\ \mathsf{X}E}{b_2 : B} \quad \rightarrowtail_1 \quad \Pi' = \begin{array}{c}\Pi'_1\\ b \lhd b_2 \quad b_1 : A\\ \Pi'_3[b_2/b']\\ b_2 : B\end{array} \quad ,$$

where

$$\begin{array}{rcl}\delta\,(\Pi, \Pi') & = & \delta\,(\widehat{\Pi_3}, \widehat{\Pi_3}') \ = \\ & = & \delta\,(\Pi_1, \Pi'_1) \cup \delta\,(\Pi_3, \Pi'_3) \ = \\ & = & \delta\,(\Pi_1, \Pi'_1) \cup \delta\,(\Pi_2, \Pi'_2)\ .\end{array}$$

$(xxi) - (xxii)$ $[\mathsf{G}I/\mathsf{G}E]$ and $[\forall I/\forall E]$ are treated as (xx) above.

□

Lemma 5.34 shows that there is a strict correspondence between the contractions defined in Section 5.4.3 and $\rightarrowtail_1$-reductions with no defects. This correspondence does not hold if we consider all the $\rightarrowtail_1$-reductions. In particular, given $\Pi \rightarrowtail_1 \Pi'$ and $\Pi \rightarrowtail_1 \Pi''$, we cannot say that Π' and Π'' converge to a common Π'''. This is true only if $\Pi \rightarrowtail_1 \Pi'$ and $\Pi \rightarrowtail_1 \Pi''$ are in some way "compatible".

Intuitively, we need to require that, when we reduce an *ind*-application r both in deriving Π' and in deriving Π'', we "unfold" it in the same way, i.e. with respect to a chain of the same length and by using the same sequence of labels. The following definition formalizes this idea.

Definition A.2. *Let Π, Π' and Π'' be marked derivations such that $\Pi \rightarrowtail_1 \Pi'$ and $\Pi \rightarrowtail_1 \Pi''$. We say that (Π, Π') and (Π, Π'') are* compatible *if and only if one of the following cases holds:*

(i) $\Pi' = \Pi$ *or* $\Pi'' = \Pi$.

(ii) $\Pi = \dfrac{\begin{array}{c}[b:A]\\ \Pi_1 \\ b:B\end{array}}{b:A \supset B}\supset I$, $\Pi' = \dfrac{\begin{array}{c}[b:A]\\ \Pi'_1 \\ b:B\end{array}}{b:A \supset B}\supset I$, $\Pi'' = \dfrac{\begin{array}{c}[b:A]\\ \Pi''_1 \\ b:B\end{array}}{b:A \supset B}\supset I$

and (Π_1, Π'_1) and (Π_1, Π''_1) are compatible.

(iii) $\Pi = \dfrac{\begin{array}{cc}\Pi_1 & \Pi_2\\ b:A & b:B\end{array}}{b:A \wedge B}\wedge I$, $\Pi' = \dfrac{\begin{array}{cc}\Pi'_1 & \Pi'_2\\ b:A & b:B\end{array}}{b:A \wedge B}\wedge I$, $\Pi'' = \dfrac{\begin{array}{cc}\Pi''_1 & \Pi''_2\\ b:A & b:B\end{array}}{b:A \wedge B}\wedge I$,

(Π_1, Π'_1) and (Π_1, Π''_1) are compatible and (Π_2, Π'_2) and (Π_2, Π''_2) are compatible.

(iv) $\Pi = \dfrac{\begin{array}{c}[b_1 \lhd b_2]\\ \Pi_1 \\ b_2:A\end{array}}{b_1 : \mathsf{X}A}\mathsf{X}I$, $\Pi' = \dfrac{\begin{array}{c}[b_1 \lhd b_2]\\ \Pi'_1 \\ b_2:A\end{array}}{b_1 : \mathsf{X}A}\mathsf{X}I$, $\Pi'' = \dfrac{\begin{array}{c}[b_1 \lhd b_2]\\ \Pi''_1 \\ b_2:A\end{array}}{b_1 : \mathsf{X}A}\mathsf{X}I$

and (Π_1, Π'_1) and (Π_1, Π''_1) are compatible.
Analogously for the cases in which the last application of Π is a $\mathsf{G}I$ or a $\forall I$.

(v) $\Pi = \dfrac{\begin{array}{cc}\Pi_1 & \Pi_2\\ b:A \supset B & b:A\end{array}}{b:B}\supset E$, $\Pi' = \dfrac{\begin{array}{cc}\Pi'_1 & \Pi'_2\\ b:A \supset B & b:A\end{array}}{b:B}\supset E$,

$\Pi'' = \dfrac{\begin{array}{cc}\Pi''_1 & \Pi''_2\\ b:A \supset B & b:A\end{array}}{b:B}\supset E$,

(Π_1, Π'_1) and (Π_1, Π''_1) are compatible and (Π_2, Π'_2) and (Π_2, Π''_2) are compatible.

(vi) $\Pi = \dfrac{\begin{array}{c}\Pi_1 \\ b_1 : \mathsf{X}A \quad b_1 \lhd b_2\end{array}}{b_2 : A}\mathsf{X}E$, $\Pi' = \dfrac{\begin{array}{c}\Pi'_1 \\ b_1 : \mathsf{X}A \quad b_1 \lhd b_2\end{array}}{b_2 : A}\mathsf{X}E$,

$\Pi'' = \dfrac{\begin{array}{c}\Pi''_1 \\ b_1 : \mathsf{X}A \quad b_1 \lhd b_2\end{array}}{b_2 : A}\mathsf{X}E$ *and (Π_1, Π'_1) and (Π_1, Π''_1) are compatible.*

Analogously for the cases in which the last application of Π is a $\mathsf{G}E$ or a $\forall E$.

(*vii*) $\Pi = \dfrac{\begin{array}{c}\Pi_1\\ b_1 : \bot\end{array}}{b : A}\ \bot E$, $\Pi' = \dfrac{\begin{array}{c}\Pi_1'\\ b_1 : \bot\end{array}}{b : A}\ \bot E$, $\Pi'' = \dfrac{\begin{array}{c}\Pi_1''\\ b_1 : \bot\end{array}}{b : A}\ \bot E$

and (Π_1, Π_1') *and* (Π_1, Π_1'') *are compatible.*

(*viii*) $\Pi = \dfrac{\begin{array}{c}[b_1 \lhd b_1]\\ \Pi_1\\ b_2 : A\end{array}}{b_2 : A}\ ser\lhd$, $\Pi' = \dfrac{\begin{array}{c}[b_1 \lhd b_1]\\ \Pi_1'\\ b_2 : A\end{array}}{b_2 : A}\ ser\lhd$, $\Pi'' = \dfrac{\begin{array}{c}[b_1 \lhd b_1]\\ \Pi_1''\\ b_2 : A\end{array}}{b_2 : A}\ ser\lhd$

and (Π_1, Π_1') *and* (Π_1, Π_1'') *are compatible.*
Analogously for the cases in which the last application of Π *is one of the following relational or structural rules: base* $\leqslant$, *lin*$\lhd$, *lin*$\lhd_{\mathcal{R}}$, *refl* $\leqslant$, *trans* $\leqslant$, *refl*$\bullet$, *symm*$\bullet$, *trans*$\bullet$, *atom*$\bullet$, *fusion.*

(*ix*) $\Pi = \dfrac{\begin{array}{ccc}\Pi_0 & [b_0 \leqslant b_i]\,[b_i : A]\,[b_i \lhd b_j] & \Pi_1\\ b_0 : A \quad b_0 \leqslant b & & b_j : A\end{array}}{b : A}\ ind$,

$\Pi' = \dfrac{\begin{array}{ccc}\Pi_0' & [b_0 \leqslant b_i]\,[b_i : A]\,[b_i \lhd b_j] & \Pi_1'\\ b_0 : A \quad b_0 \leqslant b & & b_j : A\end{array}}{b : A}\ ind$

$\Pi'' = \dfrac{\begin{array}{ccc}\Pi_0'' & [b_0 \leqslant b_i]\,[b_i : A]\,[b_i \lhd b_j] & \Pi_1''\\ b_0 : A \quad b_0 \leqslant b & & b_j : A\end{array}}{b : A}\ ind$,

(Π_0, Π_0') *and* (Π_0, Π_0'') *are compatible and* (Π_1, Π_1') *and* (Π_1, Π_1'') *are compatible.*

(*x*) $\Pi = \dfrac{\begin{array}{ccc}\Pi_0 & [b_0 \leqslant b_i]\,[b_i : A]\,[b_i \lhd b_j] & \Pi_1\\ b_0 : A \quad b_0 \leqslant b & & b_j : A\end{array}}{b : A}\ ind\,(r)$,

$\Pi' = \dfrac{\begin{array}{ccc}\Pi_0' & [b_0 \leqslant b_i]\,[b_i : A]\,[b_i \lhd b_j] & \Pi_1'\\ b_0 : A \quad b_0 \leqslant b & & b_j : A\end{array}}{b : A}\ ind\,(r)$,

$$\Pi'' = \left\{ \begin{array}{c} \Pi_0'' \\ b_0 \leqslant b_{0(r)} \qquad b_0 : A \qquad b_0 \lhd b_{1(r)} \\ \Pi_1''[b_0/b_i][b_1/b_j] \\ b_0 \leqslant b_{1(r)} \qquad b_1 : A \qquad b_1 \lhd b_{2(r)} \\ \Pi_1''[b_1/b_i][b_2/b_j] \\ b_2 : A \\ \cdot \\ \cdot \\ \cdot \\ b_0 \leqslant b_{n-1(r)} \qquad b_{n-1} : A \qquad b_{n-1} \lhd b_{(r)} \\ \Pi_1''[b_{n-1}/b_i][b/b_j] \\ b : A \end{array} \right. ,$$

(Π_0, Π_0') and (Π_0, Π_0'') are compatible and (Π_1, Π_1') and (Π_1, Π_1'') are compatible.

$$(xi)\ \Pi = \left\{ \begin{array}{c} [b_0 \leqslant b_i]\,[b_i : A]\,[b_i \lhd b_j] \\ \Pi_0 \qquad\qquad \Pi_1 \\ \dfrac{b_0 : A \quad b_0 \leqslant b \qquad\qquad b_j : A}{b : A}\ ind\,(r) \end{array} \right. ,$$

$$\Pi' = \left\{ \begin{array}{c} \Pi_0' \\ b_0 \leqslant b_{0(r)} \qquad b_0 : A \qquad b_0 \lhd b_{1(r)} \\ \Pi_1'[b_0/b_i][b_1/b_j] \\ b_0 \leqslant b_{1(r)} \qquad b_1 : A \qquad b_1 \lhd b_{2(r)} \\ \Pi_1'[b_1/b_i][b_2/b_j] \\ b_2 : A \\ \cdot \\ \cdot \\ \cdot \\ b_0 \leqslant b_{n-1(r)} \qquad b_{n-1} : A \qquad b_{n-1} \lhd b_{(r)} \\ \Pi_1'[b_{n-1}/b_i][b/b_j] \\ b : A \end{array} \right. ,$$

$$\Pi'' = \left\{ \begin{array}{lll} & & \Pi_0'' \\ b_0 \leqslant b_{0(r)} & b_0 : A & b_0 \lhd b_{1(r)} \\ & & \Pi_1''[b_0/b_i][b_1/b_j] \\ b_0 \leqslant b_{1(r)} & b_1 : A & b_1 \lhd b_{2(r)} \\ & & \Pi_1''[b_1/b_i][b_2/b_j] \\ & & b_2 : A \\ & & \vdots \\ b_0 \leqslant b_{n-1(r)} & b_{n-1} : A & b_{n-1} \lhd b_{(r)} \\ & & \Pi_1''[b_{n-1}/b_i][b/b_j] \\ & & b : A \end{array} \right. ,$$

(Π_0, Π_0') and (Π_0, Π_0'') are compatible and (Π_1, Π_1') and (Π_1, Π_1'') are compatible.

(*xii*) $$\Pi = \dfrac{\dfrac{\begin{array}{c}[b:A]\\ \Pi_1 \\ b:B\end{array}}{b:A \supset B} \supset I \quad \begin{array}{c}\Pi_2 \\ b:A\end{array}}{b:B} \supset E \quad , \quad \Pi' = \dfrac{\dfrac{\begin{array}{c}[b:A]\\ \Pi_1' \\ b:B\end{array}}{b:A \supset B} \supset I \quad \begin{array}{c}\Pi_2' \\ b:A\end{array}}{b:B} \supset E \quad ,$$

$$\Pi'' = \begin{array}{c}\Pi_2'' \\ b:A \\ \Pi_1'' \\ b:B\end{array} \quad ,$$

(Π_1, Π_1') and (Π_1, Π_1'') are compatible and (Π_2, Π_2') and (Π_2, Π_2'') are compatible.

(*xiii*) $$\Pi = \dfrac{\dfrac{\begin{array}{c}[b:A]\\ \Pi_1 \\ b:B\end{array}}{b:A \supset B} \supset I \quad \begin{array}{c}\Pi_2 \\ b:A\end{array}}{b:B} \supset E \quad , \quad \Pi' = \begin{array}{c}\Pi_2' \\ b:A \\ \Pi_1' \\ b:B\end{array} \quad , \quad \Pi'' = \begin{array}{c}\Pi_2'' \\ b:A \\ \Pi_1'' \\ b:B\end{array} \quad ,$$

(Π_1, Π_1') and (Π_1, Π_1'') are compatible and (Π_2, Π_2') and (Π_2, Π_2'') are compatible.

(*xiv*) $$\Pi = \dfrac{\dfrac{\begin{array}{c}\Pi_1 \\ b:A\end{array} \quad \begin{array}{c}\Pi_2 \\ b:B\end{array}}{b:A \wedge B} \wedge I}{b:A} \wedge E_1 \quad , \quad \Pi' = \dfrac{\dfrac{\begin{array}{c}\Pi_1' \\ b:A\end{array} \quad \begin{array}{c}\Pi_2' \\ b:B\end{array}}{b:A \wedge B} \wedge I}{b:A} \wedge E_1 \quad , \quad \Pi'' = \begin{array}{c}\Pi_1'' \\ b:A\end{array} \quad ,$$

and (Π_1, Π_1') and (Π_1, Π_1'') are compatible. There is an analogous case with $\wedge E_2$.

$$(xv)\ \Pi = \dfrac{\dfrac{\begin{matrix}\Pi_1\\ b:A\end{matrix} \quad \begin{matrix}\Pi_2\\ b:B\end{matrix}}{b:A\wedge B}\wedge I}{b:A}\wedge E_1 \quad , \quad \Pi' = \begin{matrix}\Pi_1'\\ b:A\end{matrix} \quad , \quad \Pi'' = \begin{matrix}\Pi_1''\\ b:A\end{matrix}$$

and (Π_1, Π_1') *and* (Π_1, Π_1'') *are compatible. There is an analogous case with* $\wedge E_2$.

$$(xvi)\ \Pi = \dfrac{\dfrac{\begin{matrix}[b_1 \lhd b_2]\\ \Pi_1\\ b_2 : A\end{matrix}}{b_1 : \mathsf{X}A}\mathsf{X}I \qquad b_1 \lhd b}{b:A}\mathsf{X}E \quad , \quad \Pi' = \dfrac{\dfrac{\begin{matrix}[b_1 \lhd b_2]\\ \Pi_1'\\ b_2 : A\end{matrix}}{b_1 : \mathsf{X}A}\mathsf{X}I \qquad b_1 \lhd b}{b:A}\mathsf{X}E \quad ,$$

$$\Pi'' = \begin{matrix} b_1 \lhd b \\ \Pi_1''[b/b_2] \\ b : A \end{matrix} \quad \textit{and} \quad (\Pi_1, \Pi_1')\ \textit{and}\ (\Pi_1, \Pi_1'')\ \textit{are compatible.}$$

Analogously for the cases in which the last application of Π *is a* $\mathsf{G}E$ *or a* $\forall E$.

$$(xvii)\ \Pi = \dfrac{\dfrac{\begin{matrix}[b_1 \lhd b_2]\\ \Pi_1\\ b_2 : A\end{matrix}}{b_1 : \mathsf{X}A}\mathsf{X}I \qquad b_1 \lhd b}{b:A}\mathsf{X}E \quad , \quad \Pi' = \begin{matrix} b_1 \lhd b \\ \Pi_1'[b/b_2] \\ b : A \end{matrix} \quad ,$$

$$\Pi'' = \begin{matrix} b_1 \lhd b \\ \Pi_1''[b/b_2] \\ b : A \end{matrix} \quad \textit{and} \quad (\Pi_1, \Pi_1')\ \textit{and}\ (\Pi_1, \Pi_1'')\ \textit{are compatible.}$$

Analogously for the cases in which the last application of Π *is a* $\mathsf{G}E$ *or a* $\forall E$.

Lemma A.3. *Let* Π, Π' *and* Π'' *be marked derivations such that* $\Pi \rightarrowtail_1 \Pi'$, $\Pi \rightarrowtail_1 \Pi''$, $\delta(\Pi, \Pi') = \emptyset$ *and* $\delta(\Pi, \Pi'') = \emptyset$. *Then* (Π, Π') *and* (Π, Π'') *are compatible.*

Proof. By observing the inductive definition in Definition A.2, one can notice that the only source of incompatibility comes by unfolding in two different ways some application of *ind*. But having no defects implies that the $\rightarrowtail_1$-steps correspond to a number of $\Rightarrow$-contractions (see Lemma 5.34) and thus that all the possible unfolded *ind*-applications have been treated in the same way.

□

We can now prove that whenever two $\rightarrowtail_1$-steps diverge but are compatible, then there exists some marked derivation to which their results converge.

Lemma A.4. *Let* Π, Π' *and* Π'' *be marked derivations. If* $\Pi \rightarrowtail_1 \Pi'$ *and* $\Pi \rightarrowtail_1 \Pi''$ *and* (Π, Π') *and* (Π, Π'') *are compatible, then there exists a marked derivation* Π''' *such that* $\Pi', \Pi'' \rightarrowtail_1 \Pi'''$, $\delta(\Pi', \Pi''') \subseteq \delta(\Pi, \Pi'')$ *and* $\delta(\Pi'', \Pi''') \subseteq \delta(\Pi, \Pi')$.

Proof. Let n' and n'' be the number of times the clauses in Definition 5.32 have been applied in order to get $\Pi_1 \rightarrowtail_1 \Pi'$ and $\Pi_1 \rightarrowtail_1 \Pi''$, respectively. The proof proceeds by induction on $n' + n''$. We show here the main cases.

(*i*) If $\Pi = \Pi'$ (by clause $[BC]$), then just take $\Pi''' = \Pi''$. Analogously, if $\Pi = \Pi''$ then take $\Pi''' = \Pi'$.

(*ii*) The cases in which the last clause application, both in deriving $\Pi \rightarrowtail_1 \Pi'$ and in deriving $\Pi \rightarrowtail_1 \Pi''$, is a passive clause are all very similar. We show the case $[\supset I]$ as an example. Let Π be the derivation

$$\dfrac{\begin{array}{c}[b:A]^1\\ \Pi_1\\ b:B\end{array}}{b:A \supset B}\ \supset I^1$$

and let Π' and Π'' be

$$\Pi' = \dfrac{\begin{array}{c}[b:A]^1\\ \Pi'_1\\ b:B\end{array}}{b:A \supset B}\ \supset I^1 \qquad \text{and} \qquad \Pi'' = \dfrac{\begin{array}{c}[b:A]^1\\ \Pi''_1\\ b:B\end{array}}{b:A \supset B}\ \supset I^1 \quad .$$

In the derivations above, we have $\Pi_1 \rightarrowtail_1 \Pi'_1$ and $\Pi_1 \rightarrowtail_1 \Pi''_1$ in less than n' and less than n'' clause applications, respectively. If (Π, Π') and (Π, Π'') are compatible then, by Definition A.2, we have that (Π_1, Π'_1) and (Π_1, Π''_1) are compatible. By the induction hypothesis, we can infer $\Pi'_1 \rightarrowtail_1 \Pi'''_1$ and $\Pi''_1 \rightarrowtail_1 \Pi'''_1$ for some Π'''_1 such that $\delta(\Pi'_1, \Pi'''_1) \subseteq \delta(\Pi_1, \Pi''_1)$ and $\delta(\Pi''_1, \Pi'''_1) \subseteq \delta(\Pi_1, \Pi'_1)$. Then given

$$\Pi''' = \dfrac{\begin{array}{c}[b:A]^1\\ \Pi'''_1\\ b:B\end{array}}{b:A \supset B}\ \supset I^1 \quad ,$$

we have, by Definition 5.32, $\Pi' \rightarrowtail_1 \Pi'''$ and $\Pi'' \rightarrowtail_1 \Pi'''$. Furthermore we have:

$$\begin{array}{l}\delta(\Pi', \Pi''') = \delta(\Pi'_1, \Pi'''_1) \subseteq \delta(\Pi_1, \Pi''_1) = \delta(\Pi, \Pi'')\,;\\ \delta(\Pi'', \Pi''') = \delta(\Pi''_1, \Pi'''_1) \subseteq \delta(\Pi_1, \Pi'_1) = \delta(\Pi, \Pi')\,.\end{array}$$

(*iii*) Let Π be the following derivation:

$$\dfrac{\dfrac{\begin{array}{c}[b:A]^1\\ \Pi_1\\ b:B\end{array}}{b:A \supset B}\ \supset I^1 \quad \begin{array}{c}\Pi_2\\ b:A\end{array}}{b:B}\ \supset E \quad .$$

Then, by Definition 5.32, we can have a derivation Π' obtained by applying $[\supset E]$ as the last clause and a derivation Π'' obtained by applying $[\supset I/\ \supset E]$ as the last clause, where Π' and Π'' are as follows:

$$\Pi' = \dfrac{\begin{array}{cc}\Pi'_3 & \Pi'_2\\ b:A \supset B & b:A\end{array}}{b:B}\ \supset E \quad , \qquad \Pi'' = \begin{array}{c}\Pi''_2\\ b:A\\ \Pi''_1\\ b:B\end{array} \quad .$$

In the derivations above, we have $\Pi_2 \rightarrowtail_1 \Pi_2'$, $\Pi_2 \rightarrowtail_1 \Pi_2''$ and $\Pi_1 \rightarrowtail_1 \Pi_1''$. Furthermore

$$\dfrac{\begin{array}{c}[b:A]\\ \Pi_1\\ b:B\end{array}}{b:A\supset B}\supset I \rightarrowtail_1 \Pi_3' \quad , \qquad \text{where} \qquad \Pi_3' = \dfrac{\begin{array}{c}[b:A]\\ \Pi_1'\\ b:B\end{array}}{b:A\supset B}\supset I$$

for some Π_1' such that $\Pi_1 \rightarrowtail_1 \Pi_1'$ with less than n' clauses applications. If (Π, Π') and (Π, Π'') are compatible then, by Definition A.2, we have that (Π_1, Π_1') and (Π_1, Π_1'') are compatible and that (Π_2, Π_2') and (Π_2, Π_2'') are compatible. By the induction hypothesis, we can infer $\Pi_2' \rightarrowtail_1 \Pi_2'''$, $\Pi_2'' \rightarrowtail_1 \Pi_2'''$, $\Pi_1' \rightarrowtail_1 \Pi_1'''$ and $\Pi_1'' \rightarrowtail_1 \Pi_1'''$ for some Π_2''' and Π_1''' such that $\delta(\Pi_1', \Pi_1''') \subseteq \delta(\Pi_1, \Pi_1'')$ and $\delta(\Pi_1'', \Pi_1''') \subseteq \delta(\Pi_1, \Pi_1')$. Then, given

$$\Pi''' = \begin{array}{c}\Pi_2'''\\ b:A\\ \Pi_1'''\\ b:B\end{array} \quad ,$$

we have, by Definition 5.32, $\Pi' \rightarrowtail_1 \Pi'''$ and, by Lemma A.1, $\Pi'' \rightarrowtail_1 \Pi'''$. Furthermore we have:

$$\begin{array}{rcl}\delta(\Pi', \Pi''') & = & \delta(\Pi_1', \Pi_1''') \cup \delta(\Pi_2', \Pi_2''') \subseteq \\ & \subseteq & \delta(\Pi_1, \Pi_1'') \cup \delta(\Pi_2, \Pi_2'') = \\ & = & \delta(\Pi, \Pi'') \, .\end{array}$$

Analogously:

$$\begin{array}{rcl}\delta(\Pi'', \Pi''') & = & \delta(\Pi_1'', \Pi_1''') \cup \delta(\Pi_2'', \Pi_2''') \subseteq \\ & \subseteq & \delta(\Pi_1, \Pi_1') \cup \delta(\Pi_2, \Pi_2') = \\ & = & \delta(\Pi, \Pi') \, .\end{array}$$

(*iv*) We show here only the case $\wedge E_1$, as the case $\wedge E_2$ is symmetrical. Let Π be the following derivation:

$$\dfrac{\dfrac{\begin{array}{cc}\Pi_1 & \Pi_2\\ b:A & b:B\end{array}}{b:A\wedge B}\wedge I}{b:A}\wedge E_1 \quad .$$

Then, by Definition 5.32, we can have a derivation Π' obtained by applying $[\wedge E_1]$ as the last clause and a derivation Π'' obtained by applying $[\wedge I / \wedge E_2]$ as the last clause, where Π' and Π'' are as follows:

$$\Pi' = \dfrac{\dfrac{\begin{array}{cc}\Pi_1' & \Pi_2'\\ b:A & b:B\end{array}}{b:A\wedge B}\wedge I}{b:A}\wedge E_1 \quad , \qquad \Pi'' = \begin{array}{c}\Pi_1''\\ b:A\end{array} \quad ,$$

where $\Pi_1 \rightarrowtail_1 \Pi_1'$, $\Pi_1 \rightarrowtail_1 \Pi_1''$ and $\Pi_2 \rightarrowtail_1 \Pi_2'$. If (Π, Π') and (Π, Π'') are compatible then, by Definition A.2, we have that (Π_1, Π_1') and (Π_1, Π_1'') are compatible. By the induction hypothesis, we can infer $\Pi_1' \rightarrowtail_1 \Pi_1'''$

and $\Pi_1'' \rightarrowtail_1 \Pi_1'''$ for some Π_1''' such that $\delta(\Pi_1', \Pi_1''') \subseteq \delta(\Pi_1, \Pi_1'')$ and $\delta(\Pi_1'', \Pi_1''') \subseteq \delta(\Pi_1, \Pi_1')$. Then given

$$\Pi''' = \begin{array}{c} \Pi_1''' \\ b : A \end{array} \quad ,$$

we have, by Definition 5.32, $\Pi' \rightarrowtail_1 \Pi'''$ and $\Pi'' \rightarrowtail_1 \Pi'''$. Furthermore we have:

$$\delta(\Pi', \Pi''') = \delta(\Pi_1', \Pi_1''') \subseteq \delta(\Pi_1, \Pi_1'') = \delta(\Pi, \Pi'') .$$

Analogously:

$$\delta(\Pi'', \Pi''') = \delta(\Pi_1'', \Pi_1''') \subseteq \delta(\Pi_1, \Pi_1') = \delta(\Pi, \Pi') .$$

(v) The cases in which the last rule is $\mathsf{X}E$, $\mathsf{G}E$ or $\forall E$ are all analogous. We show the first one as an example. Let Π be the following derivation:

$$\dfrac{\dfrac{\begin{array}{c}[b_1 \lhd b]^1 \\ \Pi_1 \\ b : A\end{array}}{b_1 : \mathsf{X}A}\ \mathsf{X}I^1 \qquad b_1 \lhd b_2}{b_2 : A}\ \mathsf{X}E \quad .$$

Then, by Definition 5.32, we can have a Π' obtained by applying $[\mathsf{X}E]$ as the last clause and Π'' obtained by applying $[\mathsf{X}I/\mathsf{X}E]$ as the last clause, where Π' and Π'' are as follows:

$$\Pi' = \dfrac{\begin{array}{c}\Pi_2' \\ b_1 : \mathsf{X}A \qquad b_1 \lhd b_2\end{array}}{b_2 : A}\ \mathsf{X}E \qquad , \qquad \Pi'' = \begin{array}{c} b_1 \lhd b_2 \\ \Pi_1''[b_2/b] \\ b_2 : A \end{array}$$

where

$$\Pi_2' = \dfrac{\begin{array}{c} b_1 \lhd b \\ \Pi_1' \\ b : A \end{array}}{b_1 : \mathsf{X}A}\ \mathsf{X}I$$

for some Π_1' such that $\Pi_1 \rightarrowtail_1 \Pi_1'$ by less than n' clause applications. We also have

$$\dfrac{\begin{array}{c}[b_1 \lhd b]^1 \\ \Pi_1 \\ b : A\end{array}}{b_1 : \mathsf{X}A}\ \mathsf{X}I^1 \rightarrowtail_1 \Pi_2' \qquad \text{and} \qquad \Pi_1 \rightarrowtail_1 \Pi_1'' \quad ,$$

in less than n' and less than n'' clause applications respectively. If (Π, Π') and (Π, Π'') are compatible then, by Definition A.2, we have that (Π_1, Π_1') and (Π_1, Π_1'') are compatible. By the induction hypothesis, we can infer $\Pi_1' \rightarrowtail_1 \Pi_1'''$ and $\Pi_1'' \rightarrowtail_1 \Pi_1'''$ for some Π_1''' such that $\delta(\Pi_1', \Pi_1''') \subseteq \delta(\Pi_1, \Pi_1'')$ and $\delta(\Pi_1'', \Pi_1''') \subseteq \delta(\Pi_1, \Pi_1')$. We conclude:

$$\Pi', \Pi'' \rightarrowtail_1 \Pi''' = \begin{array}{c} b_1 \lhd b_2 \\ \Pi_1'''[b_2/b] \\ b_2 : A \end{array} \quad .$$

Furthermore we have:

$$\delta(\Pi', \Pi''') = \delta(\Pi'_1, \Pi'''_1) \subseteq \delta(\Pi_1, \Pi''_1) = \delta(\Pi, \Pi'') ;$$
$$\delta(\Pi'', \Pi''') = \delta(\Pi''_1, \Pi'''_1) \subseteq \delta(\Pi_1, \Pi'_1) = \delta(\Pi, \Pi') .$$

(*vi*) Now let the last rule application of Π be a *ind* and Π be the following derivation:

$$\dfrac{\begin{array}{ccc} \Pi_0 & & [b_0 \leqslant b_i]\,[b_i : A]\,[b_i \lhd b_j] \\ b_0 : A \quad b_0 \leqslant b & & \Pi_1 \\ & & b_j : A \end{array}}{b : A}\ ind\,(r) \quad .$$

Then, by Definition 5.32, we can have $\Pi \rightarrowtail_1 \Pi'$ and $\Pi \rightarrowtail_1 \Pi''$ such that Π' and Π'' are obtained by applying respectively $[ind]$ or $[IndContr]$ as the last clause. Π' and Π'' will have the following form:

$$\Pi' = \left\{ \dfrac{\begin{array}{ccc} & & [b_0 \leqslant b_i]\,[b_i : A]\,[b_i \lhd b_j] \\ \Pi'_0 & & \Pi'_1 \\ b_0 : A \quad b_0 \leqslant b & & b_j : A \end{array}}{b : A}\ ind\,(r) \right. \quad ,$$

$$\Pi'' = \left\{ \begin{array}{lll} & \Pi''_0 & \\ b_0 \leqslant b_{0(r)} & b_0 : A & b_0 \lhd b_{1(r)} \\ & \Pi''_1[b_0/b_i][b_1/b_j] & \\ b_0 \leqslant b_{1(r)} & b_1 : A & b_1 \lhd b_{2(r)} \\ & \Pi''_1[b_1/b_i][b_2/b_j] & \\ & b_2 : A & \\ & \vdots & \\ b_0 \leqslant b_{n-1(r)} & b_{n-1} : A & b_{n-1} \lhd b_{(r)} \\ & \Pi''_1[b_{n-1}/b_i][b/b_j] & \\ & b : A & \end{array} \right. \quad ,$$

where $\Pi_0 \rightarrowtail_1 \Pi'_0$ and $\Pi_0 \rightarrowtail_1 \Pi''_0$ with less than n' and less than n'' clause applications, respectively, and $\Pi_1 \rightarrowtail_1 \Pi'_1$ and $\Pi_1 \rightarrowtail_1 \Pi''_1$ with less than n' and less than n'' clause applications, respectively. If (Π, Π') and (Π, Π'') are compatible then, by Definition A.2, we have that (Π_0, Π'_0) and (Π_0, Π''_0) are compatible and that (Π_1, Π'_1) and (Π_1, Π''_1) are compatible. By the induction hypothesis, we can infer $\Pi'_0 \rightarrowtail_1 \Pi'''_0$, $\Pi''_0 \rightarrowtail_1 \Pi'''_0$, $\Pi'_1 \rightarrowtail_1 \Pi'''_1$ and $\Pi''_1 \rightarrowtail_1 \Pi'''_1$ for some Π'''_0 and Π'''_1 such that $\delta(\Pi'_0, \Pi'''_0) \subseteq \delta(\Pi_0, \Pi''_0)$, $\delta(\Pi''_0, \Pi'''_0) \subseteq \delta(\Pi_0, \Pi'_0)$, $\delta(\Pi'_1, \Pi'''_1) \subseteq \delta(\Pi_1, \Pi''_1)$ and $\delta(\Pi''_1, \Pi'''_1) \subseteq \delta(\Pi_1, \Pi'_1)$.
If we define:

$$\Pi''' = \left\{ \begin{array}{lll} & & \Pi'''_0 \\ b_0 \leqslant b_0 & b_0 : A & b_0 \lhd b_1 \\ & & \Pi'''_1[b_0/b_i][b_1/b_j] \\ b_0 \leqslant b_1 & b_1 : A & b_1 \lhd b_2 \\ & & \Pi'''_1[b_1/b_i][b_2/b_j] \\ & & b_2 : A \\ & & \vdots \\ b_0 \leqslant b_{n-1} & b_{n-1} : A & b_{n-1} \lhd b \\ & & \Pi'''_1[b_{n-1}/b_i][b/b_j] \\ & & b : A \end{array} \right. ,$$

then we have $\Pi' \rightarrowtail_1 \Pi'''$ by applying $[IndContr]$ as the last clause. And it is easy to observe that

$$\begin{array}{rcl} \delta(\Pi', \Pi''') & = & \delta(\Pi'_0, \Pi'''_0) \cup \delta(\Pi'_1, \Pi'''_1) \cup \{r\} \subseteq \\ & \subseteq & \delta(\Pi_0, \Pi''_0) \cup \delta(\Pi_1, \Pi''_1) \cup \{r\} = \\ & = & \delta(\Pi, \Pi'') . \end{array}$$

Analogously, $\Pi'' \rightarrowtail_1 \Pi'''$ by n applications of Lemma A.1 and we have:

$$\begin{array}{rcl} \delta(\Pi'', \Pi''') & = & \delta(\Pi''_0, \Pi'''_0) \cup \delta(\Pi''_1, \Pi'''_1) \subseteq \\ & \subseteq & \delta(\Pi_0, \Pi'_0) \cup \delta(\Pi_1, \Pi'_1) = \\ & = & \delta(\Pi, \Pi') . \end{array}$$

□

The confluence for $\Rightarrow_1$ (Lemma 5.35) follows as a corollary of Lemma A.4.

References

1. Martín Abadi and Zohar Manna. Nonclausal temporal deduction. In Rohit Parikh, editor, *Logic of Programs*, volume 193 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 1985.
2. Karl R. Abrahamson. Modal logic of concurrent nondeterministic programs. In *Semantics of Concurrent Computation*, pages 21–33, 1979.
3. James F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.
4. Alan Ross Anderson, Nuel D. Belnap, Jr., and J. Michael Dunn. *Entailment, The Logic of Relevance and Necessity*, volume 2. Princeton University Press, Princeton, New Jersey, 1992.
5. Carlos Areces and Balder ten Cate. Hybrid logics. In Patrick Blackburn, Frank Wolter, and Johan van Benthem, editors, *Handbook of Modal Logics*. Elsevier, 2006.
6. Philippe Balbiani and Stéphane Demri. Prefixed tableaux systems for modal logics with enriched languages. In *IJCAI (1)*, pages 190–195, 1997.
7. Stefano Baratella and Andrea Masini. A proof-theoretic investigation of a logic of positions. *Ann. Pure Appl. Logic*, 123(1-3):135–162, 2003.
8. Stefano Baratella and Andrea Masini. An approach to infinitary temporal proof theory. *Arch. Math. Log.*, 43(8):965–990, 2004.
9. David Basin, Carlos Caleiro, Jaime Ramos, and Luca Vigano. Labelled tableaux for distributed temporal logic. *Journal of Logic and Computation*, 19:1245–1279.
10. David Basin, Marcello D'Agostino, Dov M. Gabbay, Seán Matthews, and Luca Viganò, editors. *Labelled Deduction*. Kluwer Academic Publishers, Dordrecht, 2000.
11. David A. Basin, Carlos Caleiro, Jaime Ramos, and Luca Viganò. A labeled tableaux systemfor the distributed temporal logic dtl. In Demri and Jensen [47], pages 101–109.
12. Mordechai Ben-Ari. *Mathematical logic for computer science*. Prentice Hall, 1993.
13. Mordechai Ben-Ari, Amir Pnueli, and Zohar Manna. The temporal logic of branching time. *Acta Inf.*, 20:207–226, 1983.
14. Mario R. F. Benevides and Thomas S. E. Maibaum. A constructive presentation for the modal connective of necessity. *J. Log. Comput.*, 2(1):31–50, 1992.
15. Evert Willem Beth. Semantic entailment and formal derivability. *Mededelingen van de Koninklijke Nederlandse Akademie van Wetenschappen, Afdeling Letterkunde*, 18(13):309–342, 1955.
16. Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Scie*. Cambridge University Press, Cambridge, 2001.

17. Patrick Blackburn and Miroslava Tzakova. Hybrid languages and temporal logic. *Logic Journal of the IGPL*, 7(1):27–54, 1999.
18. Leonard Bolc and Andrzej Szałas. *Time and Logic. A Computational Approach*. UCL Press Ltd., London, 1995.
19. Alexander Bolotov, Artie Basukoski, Oleg Grigoriev, and Vasilyi Shangin. Natural deduction calculus for linear-time temporal logic. In Michael Fisher, Wiebe van der Hoek, Boris Konev, and Alexei Lisitsa, editors, *JELIA*, volume 4160 of *Lecture Notes in Computer Science*, pages 56–68. Springer, 2006.
20. Alexander Bolotov, Oleg Grigoriev, and Vasilyi Shangin. Natural deduction calculus for computation tree logic. In *John Vincent Atanasoff Symposium*, pages 175–183. IEEE Computer Society, 2006.
21. Alexander Bolotov, Oleg Grigoriev, and Vasilyi Shangin. Automated natural deduction for propositional linear-time temporal logic. In *TIME*, pages 47–58. IEEE Computer Society, 2007.
22. Alexander Bolotov, Oleg Grigoriev, and Vasilyi Shangin. A simpler formulation of natural deduction calculus for linear-time temporal logic. In *IICAI*, pages 1253–1266, 2007.
23. Nicolette Bonnette and Rajeev Goré. A labelled sequent system for tense logic $\mathrm{k_t}$. In Grigoris Antoniou and John K. Slaney, editors, *Australian Joint Conference on Artificial Intelligence*, volume 1502 of *Lecture Notes in Computer Science*, pages 71–82. Springer, 1998.
24. Bianca Boretti. *Proof Analysis in Temporal Logic*. PhD thesis, Università degli Studi di Milano, 2008.
25. Howard Bowman and Simon J. Thompson. A decision procedure and complete axiomatization of finite interval temporal logic with projection. *J. Log. Comput.*, 13(2):195–239, 2003.
26. Krysia Broda, Dov M. Gabbay, Luís C. Lamb, and Alessandra Russo. Labelled natural deduction for conditional logics of normality. *Logic Journal of the IGPL*, 10(2):123–163, 2002.
27. Krysia Broda, Dov M. Gabbay, Luís C. Lamb, and Alessandra Russo. *Compiled Labelled Deductive Systems: A Uniform Presentation of Non-Classical Logics*. Research Study Press, 2004.
28. Kai Brünnler and Martin Lange. Cut-free sequent systems for temporal logic. *J. Log. Algebr. Program.*, 76(2):216–225, 2008.
29. R. A. Bull. An approach to tense logic. *Theoria*, 36:282–300, 1970.
30. Robert Bull and Krister Segerberg. Basic modal logic. In Dov M. Gabbay and Franz Guenthner, editors, *Handbook of philosophical logic*, volume 2, pages 1–88. Kluwer Academic Publishers, Dordrecht ; Boston, 2nd edition, 2001.
31. John P. Burgess. Logic and time. *J. Symb. Log.*, 44(4):566–582, 1979.
32. John P. Burgess. Decidability for branching time. *Studia Logica*, 39(2-3):203–218, 1980.
33. John P. Burgess. Axioms for tense logic. I. "Since" and "until". *Notre Dame J. Formal Logic*, 23(4):367–374, 1982.
34. John P. Burgess. Basic tense logic. In Dov Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic: Volume II: Extensions of Classical Logic*, pages 89–133. Reidel, Dordrecht, 1984.
35. Carlos Caleiro and Jaime Ramos. Combining classical and intuitionistic implications. In Boris Konev and Frank Wolter, editors, *FroCos*, volume 4720 of *Lecture Notes in Computer Science*, pages 118–132. Springer, 2007.
36. Ana R. Cavalli and Luis Fariñas del Cerro. A decision method for linear temporal logic. In Robert E. Shostak, editor, *CADE*, volume 170 of *Lecture Notes in Computer Science*, pages 113–127. Springer, 1984.

37. Serenella Cerrito and Marta Cialdea Mayer. Labelled tableaux for propositional linear time logic over finite frames. In David Basin, Marcello D'Agostino, Dov M. Gabbay, Seán Matthews, and Luca Viganò, editors, *Labelled Deduction*, pages 135–159. Kluwer Academic Publishers, Norwell, MA, USA, 2000.
38. Brian F. Chellas. *Modal Logic: an introduction*. Cambridge University Press, 1980.
39. Jan Chomicki and David Toman. Temporal logic in information systems. In Jan Chomicki and Gunter Saake, editors, *Logics for Databases and Information Systems*, pages 31–70. Kluwer, 1998.
40. Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logic of Programs*, pages 52–71, 1981.
41. Edmund M. Clarke and Bernd-Holger Schlingloff. Model checking. In John Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, pages 1635–1790. Elsevier and MIT Press, 2001.
42. Costas Courcoubetis, Moshe Y. Vardi, and Pierre Wolper. Reasoning about fair concurrent programs. In *STOC*, pages 283–294. ACM, 1986.
43. Marcello D'Agostino, Dov Gabbay, Reiner Hahnle, and Joachim Posegga, editors. *Handbook of Tableau Methods*. Kluwer Academic Publishers, 1999.
44. Marcello D'Agostino and Dov M. Gabbay. A generalization of analytic deduction via labelled deductive systems. part i: Basic substructural logics. *J. Autom. Reasoning*, 13(2):243–281, 1994.
45. Luis Fariñas del Cerro, David Fauthoux, Olivier Gasquet, Andreas Herzig, Dominique Longin, and Fabio Massacci. Lotrec : The generic tableau prover for modal and description logics. In Goré et al. [82], pages 453–458.
46. Luis Fariñas del Cerro and Andreas Herzig. Combinig classical and intuitionistic logic, or: Intuitionistic implication as a conditional. In *Frontiers of Combining Systems (FroCos)*, pages 93–102, 1996.
47. Stéphane Demri and Christian S. Jensen, editors. *15th International Symposium on Temporal Representation and Reasoning, TIME 2008, Université du Québec à Monteéal, Canada, 16-18 June 2008*. IEEE Computer Society, 2008.
48. Stéphane Demri and Ranko Lazić. Ltl with the freeze quantifier and register automata. *ACM Trans. Comput. Logic*, 10(3):1–30, 2009.
49. Kosta Doen. Sequent-systems for modal logic. *The Journal of Symbolic Logic*, 50(1):149–168, 1985.
50. Jon M. Dunn. Relevance logic and entailment. In D. Gabbay and F. Günther, editors, *Handbook of Philosophical Logic*, volume III, pages 117–224. Reidel Publication Company, 1986.
51. E. Allen Emerson. Alternative semantics for temporal logics. *Theor. Comput. Sci.*, 26:121–130, 1983.
52. E. Allen Emerson. Temporal and modal logic. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 995–1072. 1990.
53. E. Allen Emerson. Automated temporal reasoning about reactive systems. In Faron Moller and Graham M. Birtwistle, editors, *Banff Higher Order Workshop*, volume 1043 of *Lecture Notes in Computer Science*, pages 41–101. Springer, 1995.
54. E. Allen Emerson and Joseph Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *J. Comput. Syst. Sci.*, 30(1):1–24, 1985.
55. E. Allen Emerson and Joseph Y. Halpern. "sometimes" and "not never" revisited: on branching versus linear time temporal logic. *J. ACM*, 33(1):151–178, 1986.
56. Michael Fisher. A resolution method for temporal logic. In *IJCAI*, pages 99–104, 1991.

57. Michael Fisher, Clare Dixon, and Martin Peim. Clausal temporal resolution. *ACM Trans. Comput. Log.*, 2(1):12–56, 2001.
58. Michael Fisher, Dov Gabbay, and Lluis Vila. *Handbook of Temporal Reasoning in Artificial Intelligence (Foundations of Artificial Intelligence (Elsevier))*. Elsevier Science Inc., New York, NY, USA, 2005.
59. Frederic Brenton Fitch. *Symbolic Logic*. New York, Roland Press, 1952.
60. Frederic Brenton Fitch. Tree proofs in modal logic. *Journal of Symbolic Logic*, 31, 1966.
61. Melvin Fitting. *Proof methods for modal and intuitionistic logics*, volume 169 of *Synthese Library*. D. Reidel Publishing Co., Dordrecht, 1983.
62. Melvin Fitting. Basic modal logic. In *Handbook of logic in artificial intelligence and logic programming, Vol. 1*, Oxford Sci. Publ., pages 365–448. Oxford Univ. Press, New York, 1993.
63. Nissim Francez. *Fairness*. Springer-Verlag, 1986.
64. Dov M. Gabbay. An irreflexivity lemma with applications to axiomatizations of conditions on tense frames. Aspects of philosophical logic, Synth. Libr. 147, 67-89., 1981.
65. Dov M. Gabbay. The declarative past and imperative future: Executable temporal logic for interactive systems. In *Temporal Logic in Specification*, pages 409–448, London, UK, 1987. Springer-Verlag.
66. Dov M. Gabbay. *Labelled Deductive Systems*. Clarendon Press, 1996.
67. Dov M. Gabbay. An overview of fibred semantics and the combination of logics. In *Frontiers of Combining Systems (FroCos)*, pages 1–55, 1996.
68. Dov M. Gabbay, Ian Hodkinson, and Mark Reynolds. *Temporal logic (vol. 1): mathematical foundations and computational aspects*. Oxford University Press, Inc., New York, NY, USA, 1994.
69. Dov M. Gabbay and Nicola Olivetti. *Goal-directed proof theory*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.
70. Dov M. Gabbay and Amir Pnueli. A sound and complete deductive system for ctl* verification. *Logic Journal of the IGPL*, 16(6):499–536, 2008.
71. Dov M. Gabbay, Amir Pnueli, Saharon Shelah, and Jonathan Stavi. On the temporal analysis of fairness. In *POPL '80: Proceedings of the 7th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 163–173, New York, NY, USA, 1980. ACM.
72. Antony Galton. Time and change for AI. In *Handbook of logic in artificial intelligence and logic programming, Vol. 4*, Oxford Sci. Publ., pages 175–240. Oxford Univ. Press, New York, 1995.
73. Gerhard Gentzen. Investigations into logical deductions, 1935. In M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North-Holland Publishing Co., Amsterdam, 1969.
74. Jean-Yves Girard. *Proof Theory and Logical Complexity*, volume 1. Bibliopolis, 1987.
75. Robert I. Goldblatt. *Logics of Time and Computation*. CSLI Lecture Notes, 1987.
76. Valentin Goranko. Temporal logics of computations (manuscript). 2000.
77. Valentin Goranko. Temporal logics with reference pointers and computation tree logics. *Journal of Applied Non-Classical Logics*, 10(3-4), 2000.
78. Valentin Goranko, Angelo Montanari, Pietro Sala, and Guido Sciavicco. A general tableau method for propositional interval temporal logics: Theory and implementation. *J. Applied Logic*, 4(3):305–330, 2006.
79. Valentin Goranko and Alberto Zanardo. From linear to branching-time temporal logics: Transfer of semantics and definability. *Logic Journal of the IGPL*, 15(1):53–76, 2007.

80. Rajeev Goré. Cut-free sequent and tableau systems for propositional diodorean modal logics. *Studia Logica*, 53(3):433–458, 1994.
81. Rajeev Goré. Tableau methods for modal and temporal logics. In M. D'Agostino, D. Gabbay, R. Hahnle, and J.Posegga, editors, *Handbook of Tableau Methods*, pages 297–396. Kluwer Academic Publishers, 1999.
82. Rajeev Goré, Alexander Leitsch, and Tobias Nipkow, editors. *Automated Reasoning, First International Joint Conference, IJCAR 2001, Siena, Italy, June 18-23, 2001, Proceedings*, volume 2083 of *Lecture Notes in Computer Science*. Springer, 2001.
83. Graham D. Gough. Decision procedures for temporal logic. Technical Reports UMCS-89-10-1, Department of Computer Science, University of Manchester, oct 1989.
84. Joseph Y. Halpern and Yoav Shoham. A propositional modal logic of time intervals. *J. ACM*, 38(4):935–962, 1991.
85. Jaakko Hintikka. Form and content in quantification theory. *Acta Philosophica Fennica*, 8:7–55, 1955.
86. Robin Hirsch, Ian Hodkinson, Maarten Marx, Szabolsc Mikulás, and Mark Reynolds. Mosaics and step-by-step. Remarks on "A modal logic of relations". In E. Orlowska, editor, *Logic at Work. Essays Dedicated to the Memory of Helena Rasiowa*, volume 24 of *Studies in Fuzziness and Soft Computing*, pages 158–167. Springer-Verlag, 1999.
87. Wilfrid Hodges. Elementary predicate logic. In *Handbook of philosophical logic, Vol. 1*, pages 1–129. Kluwer Acad. Publ., Dordrecht, 2001.
88. Ian Hodkinson and Mark Reynolds. Temporal logic. In Johan van Benthem Patrick Blackburn and Frank Wolter, editors, *Handbook of Modal Logic*, chapter 11, pages 655–720. Elsevier Science, New York, NY, USA, 2007.
89. G. E. Hughes and M. J. Cresswell. *A New Introduction to Modal Logic*. Routledge, 1996.
90. James Pustejovsky Inderjeet Mani and Rob Gaizauskas (eds.). *The Language of Time: A Reader*. Oxford University Press, 2005.
91. Andrzej Indrzejczak. Multiple sequent calculus for tense logics. In H. Wansing and F. Wolter, editors, *ICTL 2000: Proceedings of ICTL 2000*.
92. Andrzej Indrzejczak. A survey of natural deduction systems for modal logics. *Logica trianguli*, (3):55–83, 1999.
93. Andrzej Indrzejczak. A labelled natural deduction system for linear temporal logic. *Studia Logica*, 75(3):345–376, 2003.
94. Stanislaw Jaskowski. On the rules of suppositions in formal logic. In *Polish Logic (1920-1939)*, pages 232–258. Oxford University Press, 1967.
95. Donald Kalish and Richard Montague. *Logic: Techniques of formal reasoning*. Harcourt Brace and World, New York, 1964.
96. Johan A. W. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles, 1968.
97. Ryo Kashima. Cut-free sequent calculi for some tense logics. *Studia Logica*, 53(1):119–136, 1994.
98. Yonit Kesten, Zohar Manna, Hugh McGuire, and Amir Pnueli. A decision algorithm for full propositional temporal logic. In Costas Courcoubetis, editor, *CAV*, volume 697 of *Lecture Notes in Computer Science*, pages 97–109. Springer, 1993.
99. Saul A. Kripke. A semantical analysis of modal logic I: Normal modal propositional calculi. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.
100. Fred Kroger. *Temporal Logic of Programs*. Springer-Verlag, 1987.
101. François Laroussinie, Nicolas Markey, and Ph. Schnoebelen. Temporal logic with forgettable past. In *LICS*, pages 383–392. IEEE Computer Society, 2002.

102. Zohar Manna and Amir Pnueli. Completing the temporal picture. *Theor. Comput. Sci.*, 83(1):97–130, 1991.
103. Davide Marchignoli. *Natural Deduction Systems for Temporal Logics*. PhD thesis, Dipartimento di Informatica, Università di Pisa, February 2002.
104. Simone Martini and Andrea Masini. A computational interpretation of modal proofs. In *Proof Theory of Modal Logics*, pages 213–241. Kluwer, 1994.
105. Maarten Marx, Szabolcs Mikulás, and Mark Reynolds. The mosaic method for temporal logics. In Roy Dyckhoff, editor, *TABLEAUX*, volume 1847 of *Lecture Notes in Computer Science*, pages 324–340. Springer, 2000.
106. Andrea Masini. 2-sequent calculus: A proof theory of modalities. *Ann. Pure Appl. Logic*, 58(3):229–246, 1992.
107. Andrea Masini. 2-sequent calculus: Intuitionism and natural deduction. *J. Log. Comput.*, 3(5):533–562, 1993.
108. Andrea Masini, Luca Viganò, and Marco Volpe. Labeled natural deduction for a bundled branching temporal logic. *Journal of Logic and Computation (Submitted)*.
109. Andrea Masini, Luca Viganò, and Marco Volpe. A labeled natural deduction system for a fragment of ctl*. In Sergei N. Artëmov and Anil Nerode, editors, *LFCS*, volume 5407 of *Lecture Notes in Computer Science*, pages 338–353. Springer, 2009.
110. Andrea Masini, Luca Viganò, and Marco Volpe. A history of until. *Electr. Notes Theor. Comput. Sci.*, 262:189–204, 2010.
111. Szabolcs Mikulás. Taming first-order logic. *Journal of the IGPL*, 6(2):305–316, 1998.
112. Sara Negri. Proof analysis in modal logic. *J. Philos. Logic*, 34(5-6):507–544, 2005.
113. Sara Negri. Proof analysis in non-classical logics. In *Logic Colloquium 2005*, volume 28 of *Lect. Notes Log.*, pages 107–128. Assoc. Symbol. Logic, Urbana, IL, 2008.
114. István Németi. *Free Algebras and Decidability in Algebraic Logic*. PhD thesis, Hungarian Academy of Sciences, Budapest, 1986.
115. István Németi. Decidable versions of first order logic and cylindric-relativized set algebras. In M. de Rijke L. Csirmaz, D. Gabbay, editor, *Logic Colloquium '92*, pages 171–241. CSLI Publications, 1995.
116. Hirokazu Nishimura. Is the semantics of branching structures adequate for non-metric Ockhamist tense logics? *J. Philos. Logic*, 8(4):477–478, 1979.
117. Hans Jürgen Ohlbach. Semantics-based translation methods for modal logics. *J. Log. Comput.*, 1(5):691–746, 1991.
118. Peter Ohrstrom and Per F.V. Hasle. *Temporal Logic from Ancient Ideas to Artificial Intelligence*. Kluwer Academic Publishers, Dordrecht, 1995.
119. Ewa Orłowska. Relational proof systems for modal logics. In H. Wansing, editor, *Proof Theory of Modal Logic*, pages 55–78. Kluwer Academic Publisher, 1996.
120. Barbara Paech. Gentzen-systems for propositional temporal logics. In Egon Börger, Hans Kleine Büning, and Michael M. Richter, editors, *CSL*, volume 385 of *Lecture Notes in Computer Science*, pages 240–253. Springer, 1988.
121. Lawrence C. Paulson. *Isabelle: a Generic Theorem Prover*. LNCS 828. Springer-Verlag, 1994.
122. Frank Pfenning. Logical frameworks. In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, chapter 17, pages 1063–1147. Elsevier Science and MIT Press, 2001.
123. Regimantas Pliuskevicius. Deduction-based decision procedure for a clausal miniscoped fragment of ftl. In Goré et al. [82], pages 107–120.
124. Amir Pnueli. The temporal logic of programs. In *FOCS*, pages 46–57. IEEE, 1977.
125. Dag Prawitz. *Natural Deduction: a Proof-Theoretical Study*. Number 3 in Stockholm Studies in Philosophy. Almquist and Wiskell, 1965.

126. Dag Prawitz. Ideas and results in proof theory. In J.E. Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symposium*, pages 235–307. North-Holland, 1971.
127. Arthur Prior. *Time and Modality*. Oxford University Press, 1957.
128. Arthur Prior. *Past, Present and Future*. Oxford University Press, 1967.
129. Arthur Prior. *Papers on Time and Tense*. Oxford University Press, 1968.
130. Willard V. O. Quine. On natural deduction. *Journal of Symbolic Logic*, 15(2):93–102, 1950.
131. Christian Jacques Rentería and Edward Hermann Haeusler. A natural deduction system for ctl. *Bulletin of the Section of Logic*, 31(4):231–240, 2002.
132. Nicholas Rescher and Alasdair Urquhart. *Temporal Logic*. Springer-Verlag, 1971.
133. Mark Reynolds. A decidable temporal logic of parallelism. *Notre Dame Journal of Formal Logic*, 38:419–436, 1996.
134. Mark Reynolds. The complexity of temporal logic over the reals. *CoRR*, cs.LO/9910012, 1999.
135. Mark Reynolds. An axiomatization of full computation tree logic. *Journal of Symbolic Logic*, 66(3):1011–1057, 2001.
136. Mark Reynolds. Axioms for branching time. *J. Log. Comput.*, 12(4):679–697, 2002.
137. Mark Reynolds. The complexity of the temporal logic with "until" over general linear time. *J. Comput. Syst. Sci.*, 66(2):393–426, 2003.
138. Mark Reynolds. An axiomatization of PCTL*. *Inf. Comput.*, 201(1):72–119, 2005.
139. Mark Reynolds. A tableau for bundled CTL*. *J. Log. Comput.*, 17(1):117–132, 2007.
140. Mark Reynolds. Dense time reasoning via mosaics. In *TIME '09: Proceedings of the 2009 16th International Symposium on Temporal Representation and Reasoning*, pages 3–10, Washington, DC, USA, 2009. IEEE Computer Society.
141. Mark Reynolds. The complexity of decision problems for linear temporal logics. *Journal of Studies in Logic*, 2010.
142. John Alan Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12(1):23–41, 1965.
143. Alessandra Russo. Generalising propositional modal logic using labelled deductive systems. In *Frontiers of Combining Systems (FroCos)*, pages 57–73, 1996.
144. Henrik Sahlqvist. Completeness and correspondence in first and second order semantics for modal logic. In North Holland S. Kanger, editor, *Proceedings of the Third Scandinavian Logic Symposium*, pages 110–143, 1975.
145. Peter H. Schmitt and Jean Goubault-Larrecq. A tableau system for linear-time temporal logic. In Ed Brinksma, editor, *TACAS*, volume 1217 of *Lecture Notes in Computer Science*, pages 130–144. Springer, 1997.
146. Stefan Schwendimann. A new one-pass tableau calculus for pltl. In Harrie C. M. de Swart, editor, *TABLEAUX*, volume 1397 of *Lecture Notes in Computer Science*, pages 277–292. Springer, 1998.
147. Tatsuya Shimura. Cut-free systems for the modal logic s4.3 and s4.3grz. *Reports on Mathematical Logic*, 25, 1991.
148. Alex K. Simpson. *The Proof Theory and Semantics of Intuitionistic Modal Logic*. PhD thesis, College of Science and Engineering, School of Informatics, University of Edinburgh, 1994.
149. Colin Stirling. Modal and temporal logics. In *Handbook of logic in computer science, Vol. 2*, volume 2 of *Handb. Log. Comput. Sci.*, pages 477–563. Oxford Univ. Press, New York, 1992.
150. Richmond H. Thomason. Combinations of tense and modality. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic: Extensions of Classical Logic*, pages 135–165. Reidel, Dordrecht, 1984.

151. Anne Sjerp Troelstra. Metamathematical investigation of intuitionistic arithmetic and analysis. volume 344 of *Lecture Notes in Mathematics*, Berlin, 1973. Springer-Verlag.
152. Anne Sjerp Troelstra and Helmut Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, 2000.
153. Anne Sjerp Troelstra and Dirk van Dalen. *Constructivism in Mathematics: An introduction*, volume II. North-Holland, Amsterdam, 1988.
154. Johan van Benthem. Correspondence theory. In Dov M. Gabbay and Franz Guenthner, editors, *Handbook of philosophical logic*, volume 4, pages 162–167. D. Reidel, Dordrecht, 1983.
155. Dirk van Dalen. *Logic and Structure*. Springer-Verlag, 1980.
156. Yde Venema. Temporal logic. In L. Goble, editor, *The Blackwell Guide to Philosophical Logic*, pages 203–223. Blackwell Publishers, Malden, USA, 2001.
157. Yde Venema and Maarten Marx. A modal logic of relations. In E. Orlowska, editor, *Logic at Work: Essays Dedicated to the Memory of Helena Rasiowa*. Springer-Verlag, 1999.
158. G. Venkatesh. A decision method for temporal logic based on resolution. In S. N. Maheshwari, editor, *FSTTCS*, volume 206 of *Lecture Notes in Computer Science*, pages 272–289. Springer, 1985.
159. Luca Viganò. *Labelled Non-Classical Logics*. Kluwer Academic Publishers, 2000.
160. Luca Viganò and Marco Volpe. Labeled natural deduction systems for a family of tense logics. In Demri and Jensen [47], pages 118–126.
161. Heinrich Wansing. Sequent calculi for normal modal proposisional logics. *J. Log. Comput.*, 4(2):125–142, 1994.
162. Heinrich Wansing. *Displaying Modal Logics*. Kluwer Academic Publishers, 1999.
163. Pierre Wolper. The tableau method for temporal logic: An overview. *Logique et Analyse*, (110–111):119–136, 1985.
164. Alberto Zanardo. A finite axiomatization of the set of strongly valid ockhamist formulas. *Journal of Philosophical Logic*, 14:447–468, 1985.
165. Alberto Zanardo. Axiomatization of 'peircean' branching-time logic. *Studia Logica*, 1990.
166. Alberto Zanardo. A complete deductive system for since-until branching time logic. *Journal of Philosophical Logic*, 1991.
167. Alberto Zanardo. Branching-time logic with quantification over branches: The point of view of modal logic. *Journal of Symbolic Logic*, 61(1):1–39, 1996.