Canonical Ground Horn Theories

Maria Paola Bonacina[1*] and Nachum Dershowitz[2]

[1] Dipartimento di Informatica, Università degli Studi di Verona,
Strada Le Grazie 15, I-37134 Verona, Italy
mariapaola.bonacina@univr.it

[2] School of Computer Science, Tel Aviv University, Ramat Aviv 69978, Israel
Nachum.Dershowitz@cs.tau.ac.il

Dedicated to the memory of **Harald Ganzinger**, friend and colleague.

Abstract. An abstract framework of canonical inference based on proof orderings is applied to ground Horn theories with equality. A finite presentation that makes all normal-form proofs available is called *saturated*. To maximize the chance that a saturated presentation be finite, it should also be *contracted*, in which case it is deemed *canonical*. We apply these notions to propositional Horn theories – or equivalently Moore families – presented as *implicational systems* or *associative-commutative rewrite systems*, and ground equational Horn theories, presented as *decreasing conditional rewrite systems*. For implicational systems, we study different notions of optimality and the completion procedures that generate them, and we suggest a new notion of *rewrite-optimality*, that takes contraction by simplification into account. For conditional rewrite systems, we show that reduced (fully normalized) is stronger than contracted (sans redundancy), and accordingly the *perfect* system – complete and reduced – is preferred to the canonical one – saturated and contracted. We conclude with a survey of approaches to normal-form proofs, saturated, or canonical, systems, and decision procedures based on them.

Keywords: Horn theories, conditional theories, Moore families, decision procedures, canonical systems, normal forms, saturation, redundancy

> The first concept is ... the elimination of equations and rules....
> An equation $C \Rightarrow s = t$ can be discarded
> if there is also a proof of the same conditional equation,
> different from the one which led to the construction of the equation.
> In addition, this proof has to be simpler
> with respect to the complexity measure on proofs.
>
> – Harald Ganzinger (1991)

* Research supported in part by Ministero per l'Istruzione, l'Università e la Ricerca (grants no. 2003-097383 and 2007-9E5KM8) and by EU COST Action IC0901.

1 Motivation

We are interested in the study of presentations for theories in Horn logic with equality. We use the term "presentation" to mean a set of formulæ, reserving "theory" for a presentation with all its theorems. Thus, a *Horn presentation* is any set of Horn clauses, while a *Horn theory* is a deductively-closed set of formulæ that can be axiomatized by a Horn presentation. Since a Horn presentation can also be read naturally as a set of instructions for a computer, Horn theories are important in automated reasoning, artificial intelligence, declarative programming and deductive databases. The literature is vast; surveys include those by Apt [1] and Hodges [51]. More specifically, conditional rewriting (and unification) with equational Horn clauses has been proposed as a logic-based programming paradigm in [37,68,49,44,38]; see [50] for a survey.

On account of their double nature – computational and logical – Horn theories, and especially Horn theories with equality, presented by sets of *conditional equations* or *conditional rewrite rules*, played a special rôle in Harald Ganzinger's work (e.g. [48]). Harald's study of them represented the transition phase from his earlier work on compilers and programming languages to his later work in automated deduction.

From the perspective taken here, the quality of presentations depends on the quality of the proofs they make possible: the better are the proofs, the better is the presentation. Proofs are measured by *proof orderings*, and the most desirable ones are those that are minimal in the chosen ordering. Since a minimal proof in a certain presentation may not remain minimal in a presentation expanded by deduction, the best proofs are those that are minimal in deductively-closed presentations. These best proofs are called *normal-form proofs*. However, what is a deductively-closed presentation depends on the choice of *deduction mechanism*. Thus, the choices of notion of normal-form proof and deduction mechanism are intertwined.

One reason for deeming normal-form proofs to be best is their connection with decidability. The archetypal instance of this concept is rewriting for *equational theories*, where normal-form proofs are valley proofs. A *valley proof* of an equational theorem $\forall\bar{x}\ s\simeq t$, where $\bar{x}$ are the variables in $s\simeq t$, is a proof chain $\widetilde{s} \xrightarrow{*} \circ \xleftarrow{*} \widetilde{t}$, where $\widetilde{s}$ and $\widetilde{t}$ are s and t with their variables treated as Skolem constants, and equations only decrease terms. Given a presentation E of universally quantified equations, and a complete simplification ordering $\succ$, an equivalent *ground-convergent* presentation $E^\sharp$ offers a valley proof for every equational theorem $\forall\bar{x}\ s\simeq t$. If $E^\sharp$ is *finite*, it serves as a *decision procedure*, because validity can be decided by *rewriting* $\widetilde{s}$ and $\widetilde{t}$ "blindly" to their $E^\sharp$-normal forms and comparing the results. If $E^\sharp$ is also *reduced*, in the sense that as much as possible is in normal form, it is called *canonical*, and is unique for the given ordering $\succ$, a property first noticed by Mike Ballantyne (see [36]). Procedures to generate *canonical presentations*, which afford normal-form proofs and may be the basis for decision procedures, are called *completion procedures* (cf. [60,56,55,6,19,5]). For more on rewriting, see [33,39,12].

More generally, the notion of *canonicity* can be articulated into three properties of increasing strength, that were defined in the abstract framework of [34,14] as follows:

- A presentation is *complete* if it affords at least one normal-form proof for each theorem.
- A presentation is *saturated* if it supports all normal-form proofs for all theorems.
- A presentation is *canonical* if it is both saturated and contracted, in the sense of containing no redundancies.

If minimal proofs are unique, complete and saturated coincide. For equational theories, contracted means *reduced* and saturated means *convergent*. We call a system *perfect* when it is reduced but not saturated, only complete. A critical question is whether canonical, or perfect, presentations can be *finite* – possibly characterized by some quantitative bound – and/or *unique*. Viewed in this light, one purpose of studying these properties is to balance the strength of the "saturated," or "complete," requirement with that of the "contracted" requirement. On one hand, one wants saturation to be strong enough that a saturated presentation – when finite – yields a decision procedure for validity in the theory. On the other hand, one wants contraction to be as strong as possible, so as to maximize the possibility that the canonical presentation turns out to be finite. Furthermore, it is desirable that the canonical presentation be unique relative to the chosen ordering.

In this article, we present three main contributions:

- a study of canonicity in *propositional Horn theories* (Sect. 3);
- a study of canonicity in *conditional equational theories* in the ground case (Sect. 4);
- a survey of proof normalization and decision procedures based on saturated systems, primarily in Horn theories (Sect. 5).

Propositional Horn theories are the theories presented by sets of propositional Horn implications, known as *implicational systems*. The family of models of a theory of this kind is known as a *Moore family* and has the distinctive property of closure under intersection (see [11,10]). Moore families and implicational systems play a rôle in a variety of fields in computer science, including relational databases, data mining, artificial intelligence, logic programming, lattice theory and abstract interpretations. We refer to [23] and [10] for surveys, including applications, related formalisms and historical notes.

Since a Moore family may be presented by different implicational systems, it makes sense to define and generate implicational systems that are "optimal," or "minimal," or "canonical" in some suitable sense. Bertet and Nebut [11] proposed the notions of *directness* of implicational systems, optimizing computation by forward chaining, and *direct-optimality* of implicational systems, which adds an optimization step based on a symbol count. Bertet and Monjardet [10] considered other candidates and proved them all equal to direct-optimality, which, therefore, earned the appellation *canonical-directness*. Furthermore, they showed that given a Horn function, the Moore family of its models and its associated closure operator, the elements of the corresponding canonical-direct implicational system, read as disjunctions, give the *prime implicates* of the Horn function.

We investigate correspondences between "optimal" implicational systems (direct, direct-optimal) and canonical rewrite systems, by establishing an equivalence between implicational systems and associative-commutative rewrite systems, and by defining and comparing their respective deduction mechanisms and underlying proof orderings. We discover that direct-optimality can be simulated by normalization with respect to a different proof ordering than the one assumed by rewriting, and this discrepancy leads us to introduce a new notion of *rewrite-optimality*. Thus, while directness corresponds to saturation in an expansion-oriented deduction mechanism, rewrite-optimality corresponds to canonicity.

For conditional equational theories, we find that, unlike for equational theories, reduced implies contracted, but the two notions remain distinct. Thus, in the conditional case, perfect differs from canonical in two ways: complete is weaker than saturated, and reduced is stronger than contracted. Since complete/saturated determines how much expansion we need to do in completion, whereas reduced/contracted refers to how much simplification we should have, perfect is doubly preferable to canonical.

This article is organized as follows: Sect. 2 fixes notations and concepts; Sects. 3 and 4 are devoted to propositional Horn theories and to ground conditional equational theories, respectively; and Sect. 5 contains the survey of proof normalization and saturation-based systems. We conclude with a discussion.

2 Background

Horn clauses, the subject of this study, are an important subclass of logical formulæ.

2.1 Preliminaries

Let $\Sigma = \langle X, F, P\rangle$ be a vocabulary, consisting of variables X, function (and constant) symbols F, and predicate symbols P. Although this article is mainly concerned with the ground case, where there are no variables X, we keep basic definitions as general as possible. Let T be the set of atoms over Σ. Identity of terms and atoms will be denoted by $=$. A *context* is a term with a "hole" at some indicated position. The notation $l = t[s]_u$ indicates that term s occurs in term or atom l at position u within context t, and $Var(l)$ is the set of variables occurring in term or atom l. Positions u will henceforth be omitted from the notation.

A *Horn clause*,

$$\neg a_1 \vee \cdots \vee \neg a_n \quad \text{or} \quad \neg a_1 \vee \cdots \vee \neg a_n \vee c \ ,$$

($n \geq 0$) is a clause (set of literals) with at most one positive literal, c, where $\vee$ (disjunction) is commutative and idempotent by nature, and $a_1, \ldots, a_n, c$ are atoms in T. Positive literals (c present and $n = 0$), sometimes called "facts", and negative clauses (c absent and $n > 0$), called "queries" or "goals," are special cases of Horn clauses. Horn clauses that are not queries are termed *definite* Horn clauses. A *Horn presentation* is a set of non-negative Horn clauses.

It is customary to write a Horn clause as the implication or *rule*

$$a_1 \cdots a_n \Rightarrow c \ .$$

A Horn clause is *trivial* if the *conclusion* c is the same as one of the *premises* a_i. The same clause also has n *contrapositive* forms

$$a_1 \cdots a_{j-1} a_{j+1} \cdots a_n \neg c \Rightarrow \neg a_j \ ,$$

for $1 \leq j \leq n$. Facts are written simply as is,

$$c \ ,$$

and queries as

$$a_1 \cdots a_n \Rightarrow \text{FALSE} \ ,$$

or just

$$a_1 \cdots a_n \Rightarrow \ .$$

The main inference rules for Horn-theory reasoning are *forward chaining* and *backward chaining*:

$$\frac{a_1 \cdots a_n \Rightarrow c \quad b_1 \cdots b_m c \Rightarrow d}{a_1 \cdots a_n b_1 \cdots b_m \Rightarrow d} \qquad \frac{a_1 \cdots a_n c \Rightarrow \quad b_1 \cdots b_m \Rightarrow c}{a_1 \cdots a_n b_1 \cdots b_m \Rightarrow} \ .$$

Another way to present a Horn theory is as an "implicational" system (see [11,10]). An *implicational system* S is a binary relation $S \subseteq \mathcal{P}(T) \times \mathcal{P}(T)$, read as a set of implications

$$a_1 \cdots a_n \Rightarrow c_1 \cdots c_m \ ,$$

for $a_i, c_j \in T$, with both sides understood as conjunctions. If all right-hand sides are singletons, S is a *unary* implicational system. Clearly, any definite Horn clause is such a unary implication and vice-versa, and any non-unary implication can be decomposed into a set of m unary implications, or, equivalently, Horn clauses, one for each c_i. Empty sets correspond to TRUE. Conjunctions of facts are written just as

$$c_1 \cdots c_m \ ,$$

instead of as $\emptyset \Rightarrow c_1 \ldots c_m$.

If we focus on propositional logic, atoms are propositional variables, that evaluate to either TRUE or FALSE. A propositional implication $a_1 \cdots a_n \Rightarrow c_1 \cdots c_m$ is equivalent to the bi-implication $a_1 \cdots a_n c_1 \cdots c_m \Leftrightarrow a_1 \cdots a_n$, again with both sides understood as conjunctions. Since one side is greater than the other in any monotonic well-founded ordering, it can also be translated into a rewrite rule

$$a_1 \cdots a_n c_1 \cdots c_m \ \rightarrow \ a_1 \cdots a_n \ ,$$

where juxtaposition stands for the associative-commutative-idempotent (ACI) conjunction operator, and the arrow $\rightarrow$ has the operational semantics of rewriting and the logical semantics of *equivalence* (see, for instance, [28,29,18]).

When dealing with theories with equality, we presume the underlying axioms of equality (which are Horn), and use the predicate symbol $\simeq$ (in P) symmetrically: $l \simeq r$ stands for both $l \simeq r$ and $r \simeq l$. If one views atoms as terms and phrases an atom $r(t_1, \ldots, t_n)$ as an equation $r(t_1, \ldots, t_n) \simeq \text{TRUE}$, where r is a predicate symbol other than $\simeq$, $t_1, \ldots, t_n$ are terms, and TRUE is a special symbol, not in the original vocabulary, then any equational Horn clause can be written interchangeably as a *conditional equation*,

$$p_1 \simeq q_1, \cdots, p_n \simeq q_n \Rightarrow l \simeq r \ ,$$

or as an equational clause

$$p_1 \not\simeq q_1 \vee \cdots \vee p_n \not\simeq q_n \vee l \simeq r \ ,$$

where $p_1, q_1, \ldots, p_n, q_n, l, r$ are terms, and $p \not\simeq q$ stands for $\neg(p \simeq q)$.

A *conjecture* $C \Rightarrow l \simeq r$ is valid in a theory with presentation S, where C is some set (conjunction) of equations, if $l \simeq r$ is valid in $S \cup C$, or, equivalently, $S \cup C \cup \{l \not\simeq r\}$ is unsatisfiable, where $l \not\simeq r$ is the *goal*. A conjecture $p_1 \simeq q_1 \ldots p_n \simeq q_n$ is valid in S if $S \cup \{p_1 \not\simeq q_1 \vee \ldots \vee p_n \not\simeq q_n\}$ is unsatisfiable, in which case $p_1 \not\simeq q_1 \vee \ldots \vee p_n \not\simeq q_n$ is the *goal*.

The purely equational ground case, where all conditions are empty, the propositional case (with rules in the form $a_1 \simeq \text{TRUE}, \ldots, a_n \simeq \text{TRUE} \Rightarrow c \simeq \text{TRUE}$), and the intermediate case $a_1 \simeq \text{TRUE}, \ldots, a_n \simeq \text{TRUE} \Rightarrow l \simeq r$ (where $a_1, \ldots, a_n, c$ are propositional variables and l, r are ground terms), are all covered by the general ground equational Horn presentation case.

2.2 Canonical Systems

In this paper, we apply the framework of [34,14] to proofs made of ground Horn clauses. Let $\mathbb{A}$ be the set of all *ground conditional equations* and $\mathbb{P}$ the set of all *ground Horn proofs*, over signature Σ. Formulæ $\mathbb{A}$ and proofs $\mathbb{P}$ are linked by two functions $Pm \colon \mathbb{P} \rightarrow \mathcal{P}(\mathbb{A})$, that takes a proof p and gives its premises, denoted $[p]^{Pm}$, and $Cl \colon \mathbb{P} \rightarrow \mathbb{A}$, that takes a proof p and gives its conclusion, denoted $[p]_{Cl}$. Both are extended to sets of proofs – termed *justifications* – in the usual fashion. Proofs in $\mathbb{P}$ are ordered by two *well-founded* partial orderings: a *subproof relation* $\trianglerighteq$ and a *proof ordering* $\geq$, which, for convenience, is assumed to compare only proofs with the same conclusion (that is, $p \geq q \Rightarrow [p]_{Cl} = [q]_{Cl}$).

In addition to standard inference rules of the form

$$\frac{A_1 \quad \ldots \quad A_n}{B_1 \quad \ldots \quad B_m}$$

that add inferred formulæ $B_1, \ldots, B_m$ to the set of known theorems, which already include the premises $A_1, \ldots, A_n$, we are interested in rules that delete or simplify already-inferred theorems. We use a "double-ruled" inference rule of the form

$$\frac{A_1 \quad \ldots \quad A_n}{\overline{B_1 \quad \ldots \quad B_m}}$$

meaning that the formulæ (A_i) above the rule are *replaced* by those below (B_j). It is a *deletion* rule if the consequences are a proper subset of the premises; otherwise, it is a *simplification* rule. The challenge is incorporating such rules without endangering completeness of the inference system.

Given a presentation S, the set of all proofs using premises of S is denoted by $Pf(S)$ and defined by[1]

$$Pf(S) \quad \stackrel{!}{=} \quad \{p \in \mathbb{P} : [p]^{Pm} \subseteq S\} \ .$$

A proof is *trivial* if it proves only its single premise ($[p]^{Pm} = \{[p]_{Cl}\}$) and has no subproofs other than itself ($p \trianglerighteq q \Rightarrow p = q$). A trivial proof of $a \in \mathbb{A}$ is denoted by $\widehat{a}$. The theory of S is denoted by $Th\, S$ and defined by

$$Th\, S \quad \stackrel{!}{=} \quad [Pf(S)]_{Cl} \ ,$$

that is, the conclusions of all proofs using any number of premises from S.

Three basic assumptions on $\trianglerighteq$ and $\geq$ are postulated, for all proofs p, q, r and formulæ a:

1. Proofs use their premises:
$$a \in [p]^{Pm} \Rightarrow p \trianglerighteq \widehat{a} \ .$$
2. Subproofs do not use non-extant premises:
$$p \trianglerighteq q \Rightarrow [p]^{Pm} \supseteq [q]^{Pm} \ .$$
3. Proof orderings are monotonic with respect to subproofs:[2]
$$p \trianglerighteq q > r \ \Rightarrow \ \exists v \in Pf([p]^{Pm} \cup [r]^{Pm}).\ p > v \trianglerighteq r \ .$$

(Recall that $p \geq q \Rightarrow [p]_{Cl} = [q]_{Cl}$.)

Since $>$ is well-founded, there exist *minimal* proofs. The set of minimal proofs in a given justification P is defined as

$$\mu P \quad \stackrel{!}{=} \quad \{p \in P : \forall q \in P.\ q \not< p\} \ ,$$

while the *normal-form proofs* of a presentation S are the minimal proofs in the *theory* of S, that is,

$$Nf(S) \quad \stackrel{!}{=} \quad \mu Pf(Th\, S) \ .$$

[1] We use $\stackrel{!}{=}$ to signify definitions.
[2] This is weakened in [22].

This definition is not trivial, because it is not necessarily the case that for all proofs p, $p > \widehat{[p]_{Cl}}$. For instance, for equational theories, and a standard choice of proof ordering (e.g. [5]), $\widetilde{s} \rightarrow \circ \leftarrow \widetilde{t} \not> \widetilde{s} \simeq \widetilde{t}$. In other words, trivial proofs are not normal-form proofs in general.

With these notions in place, the characterizations of presentations introduced in Sect. 1 can be defined formally: The *canonical presentation* is the set of premises of normal-form proofs, or

$$S^\sharp \quad \stackrel{!}{=} \quad [\,\mathit{Nf}(S)\,]^{Pm} \ ,$$

and a presentation S is *canonical* if $S = S^\sharp$. Since trivial proofs are not normal-form proofs in general, $S^\sharp$ is not $\mathit{Th}\, S$. Furthermore, $(S^\sharp)^\sharp = S^\sharp$.

By lifting the proof ordering to justifications and presentations, canonicity can be characterized directly in terms of the ordering. We say that presentation B is *simpler* than a logically equivalent presentation A, denoted by $A \succsim B$, when B provides better proofs than does A, in the sense that

$$\forall p \in \mathit{Pf}(A).\, \exists q \in \mathit{Pf}(B).\ p \geq q \ .$$

Thus, canonicity is characterized in terms of this quasi-ordering, by proving that the canonical presentation is the simplest, or, in other words, that $A \succsim A^\sharp$ [34,14].

In addition to canonical, a presentation S can be:

- *contracted*, if it is made of the premises of minimal proofs, or $S = [\,\mu\mathit{Pf}(S)\,]^{Pm}$;
- *saturated*, if its minimal proofs are exactly the normal-form proofs, or $\mu\mathit{Pf}(S) = \mathit{Nf}(S)$; or
- *complete*, if its set of minimal proofs contains a normal-form proof for every theorem, or $\mathit{Th}\, S = [\,\mathit{Pf}(S) \cap \mathit{Nf}(S)\,]_{Cl}$.

A clause is *redundant* in a presentation if adding it – or removing it – does not affect minimal proofs, and a presentation is *irredundant* if it does not contain anything redundant. A presentation is contracted if and only if it is irredundant, and canonical if and only if it is saturated and contracted [34,14].

A *(one-step) deduction mechanism* $\rightsquigarrow$ is a binary relation over presentations. A deduction step $S \rightsquigarrow S \cup S'$ is an *expansion* provided $S' \subseteq \mathit{Th}\, S$. A deduction step $S \cup S' \rightsquigarrow S$ is a *contraction* provided $S \cup S' \succsim S$. A sequence of deductions $S_0 \rightsquigarrow S_1 \rightsquigarrow \cdots$ is a *derivation*, whose result, or *limit*, is the set of *persisting* formulæ: $S_\infty \stackrel{!}{=} \bigcup_j \bigcap_{i \geq j} S_i$. Since [56], a fundamental requirement of derivations is *fairness*, doing all inferences that are needed to achieve the desired degree of proof normalization. A *fair* derivation generates a complete set in the limit, a *uniformly fair* derivation generates a saturated limit, and a *contracting* derivation generates a contracted limit. We refer to [14] for these definitions and results, as well as historical notes and references on fairness.

2.3 A Clausal Ordering

Modern theorem provers employ orderings to control and limit inference. Let $\succ$ be a *complete simplification ordering* on atoms and terms over Σ, by which we mean that the ordering is total (on ground terms), monotonic (with respect to term structure), stable (with respect to substitutions), and includes the subterm ordering, meaning that $t[s] \succ s$ for any non-empty context t (hence, $\succ$ is well-founded [27]). See [33], for example, for basic definitions.

Various orderings on Horn clause proofs are possible. Suppose we express atoms as equations and let $t \succ \text{TRUE}$ for all terms t over Σ. Literals may be ordered by an ordering $\succ_L$ that measures an equation $l \simeq r$ by the multiset $\{\!\{l, r\}\!\}$ and a disequation $l \not\simeq r$ by the multiset $\{\!\{l, r, l, r\}\!\}$, and compares such multisets by the multiset extension [35] of $\succ$. It follows that $l \not\simeq r \succ_L l \simeq r$, because $\{\!\{l, r, l, r\}\!\}$ is a bigger multiset than is $\{\!\{l, r\}\!\}$, which is desirable, so as to allow $l \simeq r$ to simplify $l \not\simeq r$.

Given this ordering on literals, an ordering $\succ_C$ on clauses is obtained by another multiset extension. An equational clause e of the form $p_1 \simeq q_1, \cdots, p_n \simeq q_n \Rightarrow l \simeq r$, regarded as a multiset of literals, is measured by

$$M(e) \quad \stackrel{!}{=} \quad \{\!\{\{\!\{p_1, q_1, p_1, q_1\}\!\}, \ldots, \{\!\{p_n, q_n, p_n, q_n\}\!\}, \{\!\{l, r\}\!\}\}\!\}$$

and these multisets are compared by the multiset extension of $\succ_L$. Under this ordering, a clause $C \vee p \not\simeq q \vee l \simeq r$ is smaller than a clause $C \vee f[p] \not\simeq f[q] \vee l \simeq r$, because the multiset $M(C) \cup \{\!\{\{\!\{p, q, p, q\}\!\}, \{\!\{l, r\}\!\}\}\!\}$ is smaller than the multiset $M(C) \cup \{\!\{\{\!\{f[p], f[q], f[p], f[q]\}\!\}, \{\!\{l, r\}\!\}\}\!\}$. Similarly, a clause $C \vee l \simeq r$ is smaller than a clause $C \vee f[l] \simeq f[r]$, because the multiset $M(C) \cup \{\!\{\{\!\{l, r\}\!\}\}\!\}$ is smaller than $M(C) \cup \{\!\{\{\!\{f[l], f[r]\}\!\}\}\!\}$. A clause $C \Rightarrow l \simeq r$ is smaller than a clause $B \Rightarrow l \simeq r$, such that $C \subsetneq B$, because the multiset $M(C) \cup \{\!\{\{\!\{l, r\}\!\}\}\!\}$ is smaller than the multiset $M(B) \cup \{\!\{\{\!\{l, r\}\!\}\}\!\}$.

Example 1. If e_1 is $a \simeq b \Rightarrow c \simeq d$, $M(e_1) = \{\!\{\{\!\{a, b, a, b\}\!\}, \{\!\{c, d\}\!\}\}\!\}$. If e_2 is $f(a) \simeq f(b) \Rightarrow c \simeq d$, $M(e_2) = \{\!\{\{\!\{f(a), f(b), f(a), f(b)\}\!\}, \{\!\{c, d\}\!\}\}\!\}$. Since $f(a) \succ a$ and $f(b) \succ b$ in any ordering with the subterm property, $e_2 \succ_C e_1$. $\square$

If S is a set of clauses, we write $M(S)$ also for the multiset of their measures, and $\succ_M$ for the multiset extension of $\succ_C$. Let $>_P$ be the usual proof ordering where proofs are compared by comparing the multisets of their premises: $p >_P q$ if $[p]^{Pm} \succ_M [q]^{Pm}$.

Example 2. Consider the equational theory $\{a \simeq b, b \simeq c, a \simeq c\}$. Different proof orderings induce different canonical presentations.

a. If all proofs are minimal, the canonical saturated presentation is the whole theory, while any pair of equations, like $a \simeq b$ and $b \simeq c$, is sufficient to form a complete presentation, because, in this example, the proof of $a \simeq c$ by transitivity from $\{a \simeq b, b \simeq c\}$ is minimal. Since minimal proofs are not unique, saturated and complete indeed differ.

b. Suppose $a \succ b \succ c$. If all valley proofs are minimal, the whole theory is again the saturated presentation, while the only other complete presentation is $\{a \simeq c, b \simeq c\}$, which gives $a \to c \leftarrow b$ as minimal proof of $a \simeq b$.
c. If $a \succ b \succ c$ and the proof ordering is $>_P$, then minimal proofs are unique. The complete presentation $\{a \simeq c, b \simeq c\}$ is also saturated. The proof of $a \simeq b$ is again $a \to c \leftarrow b$, which is smaller than $a \to b$, since $\{\!\{\{\!\{a, c\}\!\}, \{\!\{b, c\}\!\}\}\!\} \prec_M \{\!\{\{\!\{a, b\}\!\}\}\!\}$.
d. If a and b are incomparable, that is, $a \neq b \wedge a \not\succ b \wedge b \not\succ a$, and all valley proofs are minimal, $a \leftrightarrow b$ is not a minimal proof, and $\{a \simeq c, b \simeq c\}$ is both complete and saturated.
e. On the other hand, if only trivial proofs are minimal, it is the whole theory $\{a \simeq b, b \simeq c, a \simeq c\}$ that is both saturated and complete. □

3 Implicational Systems

In this section we study canonicity for propositional Horn theories. We consider propositional implicational systems, that are sets of implications $A \Rightarrow B$, whose antecedent A and consequent B are conjunctions of distinct propositional variables. The notation $A \Rightarrow_S B$ specifies that $A \Rightarrow B \in S$, for given implicational system S.

Let V be a set of propositional variables. A subset $X \subseteq V$ represents the propositional interpretation that assigns the value TRUE to all elements in X and FALSE to all those in $V \setminus X$. Accordingly, a set X is said to *satisfy* an implication $A \Rightarrow B$ over V if either $B \subseteq X$ or else $A \not\subseteq X$. Similarly, we say that X *satisfies* an implicational system S, or is a *model* of S, denoted by $X \models S$, if X satisfies all implications in S.

3.1 Moore Families

A *Moore family* on a given set V is a family $\mathcal{F}$ of subsets of V that contains V and is closed under intersection [13]. Moore families are in one-to-one correspondence with closure operators, where a *closure operator* on V is an operator $\varphi\colon \mathcal{P}(V) \to \mathcal{P}(V)$ that is

– *isotone*, that is, $X \subseteq X'$ implies $\varphi(X) \subseteq \varphi(X')$,
– *extensive*, that is, $X \subseteq \varphi(X)$, and
– *idempotent*, that is, $\varphi(\varphi(X)) = \varphi(X)$.

The Moore family $\mathcal{F}_\varphi$ associated with a given closure operator φ is the set of all fixed points of φ:

$$\mathcal{F}_\varphi \quad \stackrel{!}{=} \quad \{X \subseteq V : X = \varphi(X)\} \ .$$

The closure operator $\varphi_{\mathcal{F}}$ associated with a given Moore family $\mathcal{F}$ maps any $X \subseteq V$ to the least element of $\mathcal{F}$ that contains X:

$$\varphi_{\mathcal{F}}(X) \quad \stackrel{!}{=} \quad \bigcap\{Y \in \mathcal{F} \colon X \subseteq Y\} \ .$$

The Moore family $\mathcal{F}_S$ associated with a given implicational system S is the family of the *propositional models* of S, in the sense given above:

$$\mathcal{F}_S \quad \stackrel{!}{=} \quad \{X \subseteq V : X \models S\} \ .$$

In turn, every Moore family $\mathcal{F}$ can be presented at least by one implicational system, for instance $\{X \Rightarrow \varphi_{\mathcal{F}}(X) : X \subseteq V\}$. Combining the notions of closure operator for a Moore family, and Moore family associated with an implicational system, the closure operator φ_S for implicational system S maps any $X \subseteq V$ to the least model of S that satisfies X [11]:

$$\varphi_S(X) \quad \stackrel{!}{=} \quad \bigcap\{Y \subseteq V : Y \supseteq X \wedge Y \models S\} \ .$$

Example 3. If $S = \{a \Rightarrow b,\ ac \Rightarrow d,\ e \Rightarrow a\}$ and writing sets as strings, then $\mathcal{F}_S = \{\emptyset, b, c, d, ab, bc, bd, cd, abd, abe, bcd, abcd, abde, abcde\}$ and $\varphi_S(ae) = abe$. □

As noted in Sect. 2.1, there is an obvious syntactic correspondence between Horn presentations and implicational systems. At the semantic level, there is a correspondence between Horn theories and Moore families, since Horn theories are those theories whose models are closed under intersection, a fact due to McKinsey [66] and later Horn himself [52, Lemma 7]. This result is rephrased in [10] in terms of Boolean functions and Moore families: if a Horn function is defined as a Boolean function whose conjunctive normal form is a conjunction of Horn clauses, a Boolean function is Horn if and only if the set of its true points (equivalently, the set of its models) is a Moore family.[3]

Different implicational systems can describe the same Moore family, like different presentations can describe the same theory. Two implicational systems S and S' are said to be *equivalent* if they have the same Moore family, $\mathcal{F}_S = \mathcal{F}_{S'}$.

3.2 Direct Systems

In this section we investigate the relation between the notion of *direct* implicational system and that of *saturated* presentation with respect to an appropriately chosen deduction mechanism. Directness appeared in [11], motivated by finding an implicational system that allows one to compute $\varphi_S(X)$ efficiently for any X:

Definition 1 (Directness [11, Def. 1]). *An implicational system S is* direct *if $\varphi_S(X) = S(X)$, where $S(X) \stackrel{!}{=} X \cup \bigcup\{B : A \Rightarrow_S B \wedge A \subseteq X\}$.*

In other words, a direct implicational system allows one to compute $\varphi_S(X)$ in one single round of forward chaining. In general, $\varphi_S(X) = S^*(X)$, where

$$\begin{aligned} S^0(X) &= X \\ S^{i+1}(X) &= S(S^i(X)) \\ S^*(X) &= \textstyle\bigcup_i S^i(X) \ . \end{aligned}$$

[3] For enumerations of Moore families and related structures, see [32] and Sequences A102894–7 and A108798–801 in [70].

Since S, X and V are all finite, $S^*(X) = S^k(X)$ for the smallest k such that $S^{k+1}(X) = S^k(X)$.

Example 4. The implicational system $S = \{ac \Rightarrow d, e \Rightarrow a\}$ is not direct. Indeed, for $X = ce$, the computation of $\varphi_S(X) = \{acde\}$ requires two rounds of forward chaining, because only after a has been added by $e \Rightarrow a$, can d be added by $ac \Rightarrow d$. That is, $S(X) = \{ace\}$ and $\varphi_S(X) = S^2(X) = S^*(X) = \{acde\}$. $\square$

Generalizing this example, it is sufficient to have two implications $A \Rightarrow_S B$ and $C \Rightarrow_S D$ such that $A \subseteq X$, $C \not\subseteq X$ and $C \subseteq X \cup B$, for $\varphi_S(X)$ to require more than one iteration of forward chaining. Since $A \subseteq X$, but $C \not\subseteq X$, the first round adds B, but not D; since $C \subseteq X \cup B$, D is added in a second round. In the above example, $A \Rightarrow B$ is $e \Rightarrow a$ and $C \Rightarrow D$ is $ac \Rightarrow d$. The conditions $A \subseteq X$ and $C \subseteq X \cup B$ are equivalent to $A \cup (C \setminus B) \subseteq X$, because $C \subseteq X \cup B$ means that whatever is in C and not in B must be in X. Thus, to collapse the two iterations of forward chaining into one, it is sufficient to add the implication $A \cup (C \setminus B) \Rightarrow_S D$. In the example $A \cup (C \setminus B) \Rightarrow_S D$ is $ce \Rightarrow d$. This mechanism can be defined in more abstract terms as the following inference rule:

Implicational overlap

$$\frac{A \Rightarrow BO \quad CO \Rightarrow D}{AC \Rightarrow D} \quad B \cap C = \emptyset \neq O$$

Intuitively, the consequent of the first implication "overlaps" with the antecedent of the second one, whence the conclusion. The condition $O \neq \emptyset$ says that the overlap is non-trivial, and the condition $B \cap C = \emptyset$ says that it is as large as possible. Indeed, if $O = \emptyset$, the conclusion $AC \Rightarrow D$ is subsumed by $C \Rightarrow D$, and if $B \cap C \neq \emptyset$, then an alternate inference is more general. One inference step of this rule will be denoted by $\vdash_{\mathrm{I}}$. Thus, directness can be characterized as follows:

Definition 2 (Generated direct system [11, Def. 4]). *Given an implicational system S, the* direct *implicational system $I(S)$ generated from S is the smallest implicational system containing S and closed with respect to implicational overlap.*

A main theorem of [11] shows that indeed $\varphi_S(X) = I(S)(X)$. What we call "overlap" is called "exchange" in [10], where a system closed with respect to implicational overlap is said to satisfy an "exchange condition."

As we saw in Sect. 2.1, an implicational system can be rewritten as a unary system or a set of Horn clauses, and vice-versa. Recalling that an implication $A \Rightarrow B$ is equivalent to the bi-implication $AB \Leftrightarrow A$, and using juxtaposition for ACI conjunction, we can view the bi-implication as a rewrite rule $AB \rightarrow A$, where $AB \succ A$ in any well-founded ordering with the subterm property. Accordingly, we have the following:

Definition 3 (Associated rewrite system). *The rewrite system R_X* associated *to a set $X \subseteq V$ of variables is $R_X = \{x \rightarrow \textsc{true} \colon x \in X\}$. The rewrite system R_S associated with an implicational system S is $R_S = \{AB \rightarrow A \colon A \Rightarrow_S B\}$. Given S and X we can also form the rewrite system $R_X^S = R_X \cup R_S$.*

Example 5. If $S = \{a \Rightarrow b, ac \Rightarrow d, e \Rightarrow a\}$, then $R_S = \{ab \rightarrow a, acd \rightarrow ac, ae \rightarrow e\}$. If $X = ae$, then $R_X = \{a \rightarrow \text{TRUE}, e \rightarrow \text{TRUE}\}$. Thus, $R_X^S = \{a \rightarrow \text{TRUE}, e \rightarrow \text{TRUE}, ab \rightarrow a, acd \rightarrow ac, ae \rightarrow e\}$. $\square$

We show that there is a correspondence between implicational overlap and the classical notion of overlap between monomials in Boolean rewriting that was developed for theorem proving in both propositional and first-order logic (e.g. [53,54,4,73]) and applied also to declarative programming (e.g. [29,38,18]). Here we are concerned only with its propositional version:

Equational overlap

$$\frac{AO \rightarrow B \quad CO \rightarrow D}{M \rightarrow N} \quad A \cap C = \emptyset \neq O,\ M \succ N$$

where M and N are the normal forms of BC and AD with respect to $\{AO \rightarrow B, CO \rightarrow D\}$, and $\succ$ is some ordering on sets of propositions (with the subterm property).

Intuitively, the left hand sides of the two rules "overlap," yielding the proof $BC \leftarrow AOC \rightarrow AD$, which justifies the conclusion. One inference step of this rule will be denoted by $\vdash_{\mathrm{E}}$. We observe the correspondence first on the implicational system of Example 4:

Example 6. For $S = \{ac \Rightarrow d, e \Rightarrow a\}$, we have $R_S = \{acd \rightarrow ac, ae \rightarrow e\}$, and the overlap of the two rewrite rules gives $ace \leftarrow acde \rightarrow cde$. Hence, the proof $ce \leftarrow ace \leftarrow acde \rightarrow cde$ yields the rewrite rule $cde \rightarrow ce$, which corresponds to the implication $ce \Rightarrow d$ generated by implicational overlap. $\square$

Note how an implicational overlap between consequent and antecedent corresponds to an equational overlap between left hand sides, since both antecedent and consequent appears on the left hand sides of rewrite rules representing bi-implications.

Lemma 1. *If $A \Rightarrow B$ and $C \Rightarrow D$ are two non-trivial Horn clauses ($|B| = |D| = 1$, $B \not\subseteq A$, $D \not\subseteq C$), then if $A \Rightarrow B, C \Rightarrow D \vdash_{\mathrm{I}} E \Rightarrow D$ by implicational overlap, then $AB \rightarrow A, CD \rightarrow C \vdash_{\mathrm{E}} DE \rightarrow E$ by equational overlap, and vice-versa. Furthermore, all other equational overlaps are trivial.*

This result reflects the fact that implicational overlap is designed to produce a direct system $I(S)$, which, once fed with a set X, yields its image $\varphi_{I(S)}(X)$ in a single round of forward chaining. Hence, implicational overlap unfolds the forward chaining in the implicational system. Since forward chaining is complete for Horn logic, it is coherent to expect that the only non-trivial equational overlaps are those corresponding to implicational overlaps.

Proof. (If direction.) Assume $A \Rightarrow B, C \Rightarrow D \vdash_{\mathrm{I}} E \Rightarrow D$. Since B is a singleton by hypothesis, it must be that the consequent of the first implication and the antecedent of the second one overlap on B. Thus, $C \Rightarrow D$ is $BF \Rightarrow D$ and the implicational overlap of $A \Rightarrow B$ and $BF \Rightarrow D$ generates $AF \Rightarrow D$. The corresponding rewrite rules are $AB \rightarrow A$ and $BFD \rightarrow BF$, which also overlap on B yielding the equational overlap

$$AFD \leftarrow ABFD \rightarrow ABF \rightarrow AF ,$$

which generates the corresponding rule $AFD \rightarrow AF$.
(Only if direction.) If $AB \rightarrow A, CD \rightarrow C \vdash_{\mathrm{E}} DE \rightarrow E$, the rewrite rules $AB \rightarrow A$ and $CD \rightarrow C$ can overlap in four ways: $B \cap C \neq \emptyset$, $A \cap D \neq \emptyset$, $A \cap C \neq \emptyset$ and $B \cap D \neq \emptyset$, which we consider in order.

1. $B \cap C \neq \emptyset$: Since B is a singleton, it must be $B \cap C = B$, hence $C = BF$ for some F. Thus, $CD \rightarrow C$ is $BFD \rightarrow BF$, and the overlap of $AB \rightarrow A$ and $BFD \rightarrow BF$ is the same as above, yielding $AFD \rightarrow AF$. The corresponding implications $A \Rightarrow B$ and $BF \Rightarrow D$ generate $AF \Rightarrow D$ by implicational overlap.
2. $A \cap D \neq \emptyset$: Since D is a singleton, it must be $A \cap D = D$ or $A = DF$ for some F. Thus, $AB \rightarrow A$ is $DFB \rightarrow DF$ and the overlap is

$$CF \leftarrow CDF \leftarrow CDFB \rightarrow CFB ,$$

so that $CFB \rightarrow CF$ is generated. The corresponding implications $C \Rightarrow D$ and $DF \Rightarrow B$ overlap on D and generate $CF \Rightarrow B$ by implicational overlap.
3. $A \cap C \neq \emptyset$: Let $A = FO$ and $C = OG$, so that the rules are $FOB \rightarrow FO$ and $OGD \rightarrow OG$, with $O \neq \emptyset$ and $F \cap G = \emptyset$. The resulting equational overlap is trivial: $FOG \leftarrow FOGD \leftarrow FBOGD \rightarrow FBOG \rightarrow FOG$.
4. $B \cap D \neq \emptyset$: Since B and D are singletons, it must be $B \cap D = B = D$, and rules $AB \rightarrow A$ and $CB \rightarrow C$ produce the trivial overlap $AC \leftarrow ABC \rightarrow AC$.

□

The "if" direction holds also for non-Horn clauses: suppose $A \Rightarrow FO, OG \Rightarrow D \vdash_{\mathrm{I}} AG \Rightarrow D$ by implicational overlap, with $O \neq \emptyset = F \cap G$. The corresponding rewrite rules $AFO \rightarrow A$ and $OGD \rightarrow OG$ also overlap on O, yielding

$$AGD \leftarrow AFOGD \rightarrow AFOG \rightarrow AG ,$$

which generates the rule $AGD \rightarrow AG$ corresponding to $AG \Rightarrow D$.
Let $\leadsto_{\mathrm{I}}$ be the deduction mechanism of implicational overlap: $S \leadsto_{\mathrm{I}} S'$ if $S' = S \cup \{A \Rightarrow B\}$ and $A \Rightarrow B$ is generated by implicational overlap from implications in S. Clearly, such a deduction mechanism only features expansion. For propositional Horn theories, it is reasonable to assume that minimal proofs are unique, so that complete and saturated, and fair and uniformly fair, coincide. If minimal proofs are not unique, all our results still hold, provided the hypothesis of fairness of derivations is replaced by uniform fairness. For an expansion-only mechanism such as $\leadsto_{\mathrm{I}}$, fairness simply means performing all applicable implicational overlaps eventually. If we apply these concepts to implicational systems and the $\leadsto_{\mathrm{I}}$ deduction mechanism, we have:

Proposition 1. *Given an implicational system S, for all fair derivations $S = S_0 \rightsquigarrow_{\mathrm{I}} S_1 \rightsquigarrow_{\mathrm{I}} \cdots$, $S_\infty = I(S)$.*

Proof. By fairness, S_∞ is saturated, and therefore closed with respect to implicational overlap. Since $\rightsquigarrow_{\mathrm{I}}$ deletes nothing, S_∞ contains S. Since $\rightsquigarrow_{\mathrm{I}}$ adds nothing beside implicational overlaps, S_∞ is equal to the smallest system with these properties, that is, $S_\infty = I(S)$. □

Let $\rightsquigarrow_{\mathrm{E}}$ be the deduction mechanism of equational overlap: $R \rightsquigarrow_{\mathrm{E}} R'$ if $R' = R \cup \{M \rightarrow N\}$ and $M \rightarrow N$ is generated by equational overlap from rewrite rules in R. Equational overlap combines expansion, in the form of the generation of $BC \leftrightarrow AD$, with contraction – its normalization to $M \rightarrow N$, where $M \succ N$. This sort of contraction applied to normalize a newly generated formula, before it is inserted in the database, is called *forward contraction*, while the contraction applied to reduce an equation that was already established is called *backward contraction*. Thus, $\rightsquigarrow_{\mathrm{E}}$ features expansion and forward contraction, and therefore is expansion-oriented, since contraction is limited to forward contraction. Similar to $\rightsquigarrow_{\mathrm{I}}$, fairness means performing all applicable equational overlaps eventually. Lemma 1 yields the following correspondence between deduction mechanisms:

Lemma 2. *For all implicational systems S, $S \rightsquigarrow_{\mathrm{I}} S'$ if and only if $R_S \rightsquigarrow_{\mathrm{E}} R_{S'}$.*

Proof.

– If $S \rightsquigarrow_{\mathrm{I}} S'$ then $R_S \rightsquigarrow_{\mathrm{E}} R_{S'}$ follows from the if direction of Lemma 1.
– If $R_S \rightsquigarrow_{\mathrm{E}} R'$ then $S \rightsquigarrow_{\mathrm{I}} S'$ and $R' = R_{S'}$ follows from the only-if direction of Lemma 1. □

The next theorem shows that for fair derivations the process of completing S with respect to implicational overlap, and turning the result into a rewrite system, is equivalent to the process of translating S into the rewrite system R_S, and then completing it with respect to equational overlap. In other words, completion and translation commute. For the sake of expressivity, we abuse the notation slightly, and use $(R_S)_\infty$ in lieu of R_∞ for the limit of a derivation $R_0 \rightsquigarrow_{\mathrm{E}} R_1 \rightsquigarrow_{\mathrm{E}} \cdots$ where $R_0 = R_S$.

Theorem 1. *For every implicational system S and for all fair derivations $S = S_0 \rightsquigarrow_{\mathrm{I}} S_1 \rightsquigarrow_{\mathrm{I}} \cdots$ and $R_S = R_0 \rightsquigarrow_{\mathrm{E}} R_1 \rightsquigarrow_{\mathrm{E}} \cdots$, we have*

$$R_{(S_\infty)} = (R_S)_\infty \,.$$

Proof.

(a) $R_{(S_\infty)} \subseteq (R_S)_\infty$: for any $AB \rightarrow A \in R_{(S_\infty)}$, $A \Rightarrow B \in S_\infty$ by Definition 3; then $A \Rightarrow B \in S_j$ for some $j \geq 0$. Let j be the smallest such index. If $j = 0$, or $S_j = S$, $AB \rightarrow A \in R_S$ by Definition 3, and $AB \rightarrow A \in (R_S)_\infty$, because $\rightsquigarrow_{\mathrm{E}}$ features no backward contraction. If $j > 0$, $A \Rightarrow B$ is generated at stage j by implicational overlap. By Lemma 2 and by fairness of $R_0 \rightsquigarrow_{\mathrm{E}} R_1 \rightsquigarrow_{\mathrm{E}} \cdots$, $AB \rightarrow A \in R_k$ for some $k > 0$. Then $AB \rightarrow A \in (R_S)_\infty$, since $\rightsquigarrow_{\mathrm{E}}$ features no backward contraction.

(b) $(R_S)_\infty \subseteq R_{(S_\infty)}$: for any $AB \to A \in (R_S)_\infty$, $AB \to A \in R_j$ for some $j \geq 0$. Let j be the smallest such index. If $j = 0$, or $R_j = R_S$, $A \Rightarrow B \in S$ by Definition 3, and $A \Rightarrow B \in S_\infty$, because $\leadsto_{\mathrm{I}}$ features no backward contraction. Hence $AB \to A \in R_{(S_\infty)}$. If $j > 0$, $AB \to A$ is generated at stage j by equational overlap. By Lemma 2 and by fairness of $S_0 \leadsto_{\mathrm{I}} S_1 \leadsto_{\mathrm{I}} \cdots$, $A \Rightarrow B \in S_k$ for some $k > 0$. Then $A \Rightarrow B \in S_\infty$, since $\leadsto_{\mathrm{I}}$ features no backward contraction, and $AB \to A \in R_{(S_\infty)}$ by Definition 3. □

Since the limit of a fair $\leadsto_{\mathrm{I}}$-derivation is $I(S)$, it follows that:

Corollary 1. *For every implicational system S, and for all fair derivations $S = S_0 \leadsto_{\mathrm{I}} S_1 \leadsto_{\mathrm{I}} \cdots$ and $R_S = R_0 \leadsto_{\mathrm{E}} R_1 \leadsto_{\mathrm{E}} \cdots$, we have*

$$R_{(I(S))} = (R_S)_\infty \, .$$

3.3 Computing Minimal Models

The motivation for generating $I(S)$ from S is to be able to compute, for any subset $X \subseteq V$, its minimal S-model $\varphi_S(X)$ in one round of forward chaining. In other words, one envisions a two-stage process: in the first stage, S is saturated with respect to implicational overlap to generate $I(S)$; in the second stage, forward chaining is applied to $I(S) \cup X$ to generate $\varphi_{I(S)}(X) = \varphi_S(X)$. In the rewrite-based framework, these two stages can be replaced by one. For any $X \subseteq V$ we can compute $\varphi_S(X) = \varphi_{I(S)}(X)$, by giving as input to a completion procedure the rewrite system R^S_X and extracting the rules in the form $x \to \textsc{true}$. For this purpose, the deduction mechanism is enriched with contraction rules, as follows:

Simplification

$$\frac{AC \to B \quad C \to D}{\overline{AD \to B \quad C \to D}} \; AD \succ B \qquad \frac{AC \to B \quad C \to D}{\overline{B \to AD \quad C \to D}} \; B \succ AD$$

$$\frac{B \to AC \quad C \to D}{\overline{B \to AD \quad C \to D}} \, ,$$

where A can be empty, and
Deletion

$$\frac{A \leftrightarrow A}{\overline{}} \, ,$$

which eliminates trivial equalities.

Let $\leadsto_{\mathrm{R}}$ denote the deduction mechanism that extends $\leadsto_{\mathrm{E}}$ with simplification and deletion. Thus, in addition to the simplification applied as forward contraction within equational overlap, there is simplification applied as backward contraction to any rule. Accordingly, we consider derivations that are both fair and contracting, meaning that both expansion and contraction are applied systematically.

The following theorem shows that the completion of R_X^S with respect to $\leadsto_{\mathrm{R}}$ generates a limit that includes the least S-model of X. As before, we use $(R_X^S)_\infty$ in lieu of R_∞ for the limit of a derivation $R_0 \leadsto_{\mathrm{R}} R_1 \leadsto_{\mathrm{R}} \cdots$ where $R_0 = R_X^S$.

Theorem 2. *For all $X \subseteq V$, implicational systems S, and fair and contracting derivations $R_X^S = R_0 \leadsto_{\mathrm{R}} R_1 \leadsto_{\mathrm{R}} \cdots$, if $Y = \varphi_S(X) = \varphi_{I(S)}(X)$, then*

$$R_Y \subseteq (R_X^S)_\infty \ .$$

Proof. By Definition 3, $R_Y = \{x \to \textsc{true} \colon x \in Y\}$. The proof is by induction on the construction of $Y = \varphi_S(X)$.
Base case: If $x \in Y$ because $x \in X$, then $x \to \textsc{true} \in R_X$, $x \to \textsc{true} \in R_X^S$ and $x \to \textsc{true} \in (R_X^S)_\infty$, since a rule in the form $x \to \textsc{true}$ is persistent.
Inductive case: If $x \in Y$ because for some $A \Rightarrow_S B$, $B = x$ and $A \subseteq Y$, then $AB \to A \in R_S$ and $AB \to A \in R_X^S$. By the induction hypothesis, $A \subseteq Y$ implies that, for all $z \in A$, $z \in Y$ and $z \to \textsc{true} \in (R_X^S)_\infty$. Let $j > 0$ be the smallest index in the derivation $R_0 \leadsto_{\mathrm{E}} R_1 \leadsto_{\mathrm{E}} \cdots$ such that for all $z \in A$, $z \to \textsc{true} \in R_j$. Then there is an $i > j$ such that $x \to \textsc{true} \in R_i$, because the rules $z \to \textsc{true}$ simplify $AB \to A$ to $x \to \textsc{true}$. It follows that $x \to \textsc{true} \in (R_X^S)_\infty$, since a rule in the form $x \to \textsc{true}$ is persistent. □

Then, the least S-model of X can be extracted from the saturated set:

Corollary 2. *For all $X \subseteq V$, implicational systems S, and fair and contracting derivations $R_X^S = R_0 \leadsto_{\mathrm{R}} R_1 \leadsto_{\mathrm{R}} \cdots$, if $Y = \varphi_S(X) = \varphi_{I(S)}(X)$, then*

$$R_Y = \{x \to \textsc{true} \colon x \to \textsc{true} \in (R_X^S)_\infty\} \ .$$

Proof. If $x \to \textsc{true} \in (R_X^S)_\infty$, then $x \to \textsc{true} \in R_Y$, and $x \in Y$, by the soundness of equational overlap and simplification. The other direction was established in Theorem 2. □

Example 7. Let $S = \{ac \Rightarrow d, e \Rightarrow a, bd \Rightarrow f\}$ and $X = ce$. Then $Y = \varphi_S(X) = acde$, and $R_Y = \{a \to \textsc{true}, c \to \textsc{true}, d \to \textsc{true}, e \to \textsc{true}\}$. On the other hand, for $R_S = \{acd \to ac, ae \to e, bdf \to bd\}$ and $R_X = \{c \to \textsc{true}, e \to \textsc{true}\}$, completion gives $\left(R_X^S\right)_\infty = \{c \to \textsc{true}, e \to \textsc{true}, a \to \textsc{true}, d \to \textsc{true}, bf \to b\}$, where $a \to \textsc{true}$ is generated by simplification of $ae \to e$ with respect to $e \to \textsc{true}$, $d \to \textsc{true}$ is generated by simplification of $acd \to ac$ with respect to $c \to \textsc{true}$ and $a \to \textsc{true}$, and $bf \to b$ is generated by simplification of $bdf \to bd$ with respect to $d \to \textsc{true}$. Thus, $\left(R_X^S\right)_\infty$ includes R_Y, which is made exactly of the rules in the form $x \to \textsc{true}$ of $\left(R_X^S\right)_\infty$. The direct system $I(S)$ contains the implication $ce \Rightarrow d$, generated by implicational overlap from $e \Rightarrow a$ and $ac \Rightarrow d$. The corresponding equational overlap of $acd \to ac$ and $ae \to e$ gives $ce \leftarrow ace \leftarrow acde \to cde$ and, hence, generates the rule $cde \to ce$. However, this rule is redundant in the presence of $\{c \to \textsc{true}, e \to \textsc{true}, d \to \textsc{true}\}$ and simplification. □

3.4 Direct-Optimal Systems

Bertet and Nebut [11] refined the notion of direct implicational system into that of *direct-optimal* implicational system. In this subsection, we disprove the conjecture that the direct-optimal implicational system corresponds to the canonical rewrite system with respect to equational overlap and contraction.

Optimality is defined with respect to a measure $|\, S\,|$ that counts the sum of the number of occurrences of symbols on each of the two sides of each implication in a system S:

Definition 4 (Optimality [11, Section 2]). *An implicational system S is* optimal *if, for all equivalent implicational system S', $|\, S\,| \leq |\, S'\,|$ where*

$$|\, S\,| \quad \stackrel{!}{=} \quad \sum_{A \Rightarrow_S B} |A| + |B|$$

and $|A|$ is the cardinality of set A.

From an implicational system S, one can generate an equivalent implicational system $D(S)$ that is direct, optimal, and has the following properties, shown to be necessary and sufficient for directness and optimality (cf. [11, Thm. 2]):

- *extensiveness*: for all $A \Rightarrow_{D(S)} B$, $A \cap B = \emptyset$;
- *isotony*: for all $A \Rightarrow_{D(S)} B$ and $C \Rightarrow_{D(S)} D$, if $C \subset A$, then $B \cap D = \emptyset$;
- *premise property*: for all $A \Rightarrow_{D(S)} B$ and $A \Rightarrow_{D(S)} B'$, $B = B'$;
- *non-empty conclusion property*: for all $A \Rightarrow_{D(S)} B$, $B \neq \emptyset$.

This leads to the following characterization:

Definition 5 (Direct-optimal system [11, Def. 5]). *Given a direct system S, the* direct-optimal *system $D(S)$ generated from S contains precisely the implications*

$$A \Rightarrow \bigcup\{B \colon A \Rightarrow_S B\} \setminus \{C \colon D \Rightarrow_S C \wedge D \subsetneq A\} \setminus A\ ,$$

for each set A of propositions – provided the conclusion is non-empty.

From the above four properties, we can deduce an *optimization* procedure, applying – in order – the following rules:

Premise

$$\frac{A \Rightarrow B,\ A \Rightarrow C}{A \Rightarrow BC}\ ,$$

Isotony

$$\frac{A \Rightarrow B,\ AD \Rightarrow BE}{A \Rightarrow B,\ AD \Rightarrow E}\ ,$$

Extensiveness

$$\frac{AC \Rightarrow BC}{AC \Rightarrow B}\ ,$$

Definiteness

$$\frac{A \Rightarrow \emptyset}{\overline{}} \,.$$

The first rule merges all rules with the same antecedent A into one and implements the *premise* property. The second rule removes from the consequent thus generated those subsets B that are already implied by subsets A of AD, to enforce *isotony*. The third rule makes sure that antecedents C do not themselves appear in the consequent to enforce *extensiveness*. Finally, implications with empty consequent are eliminated. This latter rule is called *definiteness*, because it eliminates negative clauses, which, for Horn theories, represent queries and are not "definite" clauses.

Clearly, the changes wrought by the optimization rules do not affect the theory. Application of this optimization to the direct implicational system $I(S)$ yields the direct-optimal system $D(S)$ of S.

The following example shows that this notion of optimization does *not* correspond to elimination of redundancies by contraction in completion:

Example 8. Let $S = \{a \Rightarrow b, ac \Rightarrow d, e \Rightarrow a\}$. Then, $I(S) = \{a \Rightarrow b, ac \Rightarrow d, e \Rightarrow a, e \Rightarrow b, ce \Rightarrow d\}$, where $e \Rightarrow b$ is generated by implicational overlap of $e \Rightarrow a$ and $a \Rightarrow b$, and $ce \Rightarrow d$ is generated by implicational overlap of $e \Rightarrow a$ and $ac \Rightarrow d$. Next, optimization replaces $e \Rightarrow a$ and $e \Rightarrow b$ by $e \Rightarrow ab$, so that $D(S) = \{a \Rightarrow b, ac \Rightarrow d, e \Rightarrow ab, ce \Rightarrow d\}$. If we consider the rewriting side, we have $R_S = \{ab \to a, acd \to ac, ae \to e\}$. Equational overlap of $ae \to e$ and $ab \to a$ generates $be \to e$, and equational overlap of $ae \to e$ and $acd \to ac$ generates $cde \to ce$, corresponding to the two implicational overlaps. Thus, $(R_S)_\infty = \{ab \to a, acd \to ac, ae \to e, be \to e, cde \to ce\}$. The rule corresponding to $e \Rightarrow ab$, namely $abe \to e$, would be redundant if added to $(R_S)_\infty$, because it would be reduced to a trivial equivalence by $ae \to e$ and $be \to e$. Thus, the optimization consisting of replacing $e \Rightarrow a$ and $e \Rightarrow b$ by $e \Rightarrow ab$ does not correspond to a rewriting inference. □

The reason for this discrepancy is the different choice of ordering. The procedure of [11] optimizes the overall size of the system. For the above example, we have $|\{e \Rightarrow ab\}| = 3 < 4 = |\{e \Rightarrow a, e \Rightarrow b\}|$. The corresponding proof ordering measures a proof of a from a set X and an implicational system S by a multiset of pairs $\langle |B|, \#_B S\rangle$, for each $B \Rightarrow_S aC$ such that $B \subseteq X$, where $\#_B S$ is the number of implications in S with antecedent B. A proof of a from $X = \{e\}$ and $\{e \Rightarrow ab\}$ will have measure $\{\!\{\langle 1, 1\rangle\}\!\}$, which is smaller than the measure $\{\!\{\langle 1, 2\rangle\}\!\}$ of a proof of a from $X = \{e\}$ and $\{e \Rightarrow a, e \Rightarrow b\}$.

Completion, on the other hand, optimizes with respect to a complete simplification ordering $\succ$. For $\{abe \to e\}$ and $\{ae \to e, be \to e\}$, we have $ae \prec abe$ and $be \prec abe$ by the subterm property of $\succ$, so $\{\!\{ae, e\}\!\} \prec_L \{\!\{abe, e\}\!\}$ and $\{\!\{be, e\}\!\} \prec_L \{\!\{abe, e\}\!\}$ in the multiset extension $\succ_L$ of $\succ$, and $\{\!\{\{\!\{ae, e\}\!\}, \{\!\{be, e\}\!\}\}\!\} \prec_C \{\!\{\{\!\{abe, e\}\!\}\}\!\}$ in the multiset extension $\succ_C$ of $\succ_L$. Indeed, from a rewriting point of view, it is better to have $\{ae \to e, be \to e\}$ than $\{abe \to e\}$, since rules with smaller left-hand side are more applicable.

3.5 Rewrite Optimality

It is apparent that the differences between direct optimality and completion arise because of the application of the *premise* rule. Accordingly, we propose an alternative definition of optimality, one that does not require the *premise* property, because symbols in repeated antecedents are counted only once:

Definition 6 (Rewrite optimality). *An implicational system S is* rewrite-optimal *if $\| S \| \leq \| S' \|$ for all equivalent implicational system S', where the measure $\| S \|$ is defined by:*

$$\| S \| \quad \stackrel{!}{=} \quad |Ante(S)| + |Cons(S)| \ ,$$

for $Ante(S) \stackrel{!}{=} \{c \in A\colon\ A \Rightarrow_S B\}$, the set *of symbols occurring in antecedents, and $Cons(S) \stackrel{!}{=} \{\!\{c \in B\colon\ A \Rightarrow_S B\}\!\}$, the* multiset *of symbols occurring in consequents.*

Unlike Definition 4, where antecedents and consequents contribute equally, here symbols in antecedents are counted only once, because $Ante(S)$ is defined as a set – hence, without repetitions – while symbols in consequents are counted as many times as they appear, since $Cons(S)$ is a multiset.

Rewrite optimality appears to be an appropriate choice to work with Horn clauses, because the *premise* property conflicts with the decomposition of non-unary implications (e.g., $a_1 \cdots a_n \Rightarrow c_1 \cdots c_m$) into Horn clauses (e.g., $a_1 \cdots a_n \Rightarrow c_i$ for $1 \leq i \leq n$) that we saw in Sect. 2.1. Indeed, if S is a non-unary implicational system, and S_H is the equivalent Horn system obtained by decomposing non-unary implications, the application of the *premise* rule to S_H undoes the decomposition.

Example 9. Applying rewrite optimality to $S = \{a \Rightarrow b, ac \Rightarrow d, e \Rightarrow a\}$ of Example 8, we have $\|\{e \Rightarrow ab\}\| = 3 = \|\{e \Rightarrow a, e \Rightarrow b\}\|$, so that replacing $\{e \Rightarrow a, e \Rightarrow b\}$ by $\{e \Rightarrow ab\}$ is no longer justified. Thus, $D(S) = I(S) = \{a \Rightarrow b, ac \Rightarrow d, e \Rightarrow a, e \Rightarrow b, ce \Rightarrow d\}$, and the rewrite system associated with $D(S)$ is $(R_S)_\infty = \{ab \rightarrow a, acd \rightarrow ac, ae \rightarrow e, be \rightarrow e, cde \rightarrow ce\}$. A proof ordering corresponding to rewrite optimality would measure a proof of a from a set X and an implicational system S by the set of the cardinalities $|B|$, for each $B \Rightarrow_S aC$ such that $B \subseteq X$. Accordingly, a proof of a from $X = \{e\}$ and $\{e \Rightarrow ab\}$ will have measure $\{\!\{1\}\!\}$, which is the same as the measure of a proof of a from $X = \{e\}$ and $\{e \Rightarrow a, e \Rightarrow b\}$. □

Thus, we deem *canonical* the result of optimization without *premise* rule:

Definition 7 (Canonical system). *Given an implicational system S, the* canonical *implicational system $O(S)$ generated from S is the closure of S with respect to implicational overlap, isotony, extensiveness and definiteness.*

Let $\leadsto_{\mathrm{O}}$ denote the deduction mechanism that features implicational overlap as expansion rule and the optimization rules except *premise*, namely *isotony*, *extensiveness* and *definiteness*, as contraction rules. Then, we have:

Proposition 2. *Given an implicational system S, for all fair and contracting derivations $S = S_0 \leadsto_{\mathrm{O}} S_1 \leadsto_{\mathrm{O}} \cdots$, $S_\infty = O(S)$.*

Proof. If the derivation is fair and contracting, both expansion and contraction rules are applied systematically. Hence, the result. □

The following lemma shows that every inference by $\leadsto_{\mathrm{O}}$ is covered by an inference in $\leadsto_{\mathrm{R}}$:

Lemma 3. *For all implicational systems S, if $S \leadsto_{\mathrm{O}} S'$, then $R_S \leadsto_{\mathrm{R}} R_{S'}$.*

Proof. We consider four cases, corresponding to the four inference rules in $\leadsto_{\mathrm{O}}$:

1. *Implicational overlap:* If $S \leadsto_{\mathrm{O}} S'$ by an implicational overlap step, then $R_S \leadsto_{\mathrm{R}} R_{S'}$ by equational overlap, by Lemma 2.
2. *Isotony:* For an application of this rule, $S = S'' \cup \{A \Rightarrow B, AD \Rightarrow BE\}$ and $S' = S'' \cup \{A \Rightarrow B, AD \Rightarrow E\}$. Then, $R_S = R_{S''} \cup \{AB \to A, ADBE \to AD\}$. Simplification applies to R_S using $AB \to A$ to rewrite $ADBE \to AD$ to $ADE \to AD$, yielding $R_{S''} \cup \{AB \to A, ADE \to AD\} = R_{S'}$.
3. *Extensiveness:* When this rule applies, $S = S'' \cup \{AC \Rightarrow BC\}$ and $S' = S'' \cup \{AC \Rightarrow B\}$. Then, $R_S = R_{S''} \cup \{ACBC \to AC\}$. By mere idempotence of juxtaposition, $R_S = R_{S''} \cup \{ABC \to AC\} = R_{S'}$.
4. *Definiteness:* If $S = S' \cup \{A \Rightarrow \emptyset\}$, then $R_S = R_{S'} \cup \{A \leftrightarrow A\}$ and an application of deletion eliminates the trivial equation, yielding $R_{S'}$. □

However, the other direction of this lemma does not hold. Although every equational overlap is covered by an implicational overlap and deletions correspond to applications of the definiteness rules, there are simplifications by $\leadsto_{\mathrm{R}}$ that do not correspond to inferences in $\leadsto_{\mathrm{O}}$:

Example 10. Assume that the implicational system S includes $\{de \Rightarrow b, b \Rightarrow d\}$. Accordingly, R_S contains $\{deb \to de, bd \to b\}$. A simplification inference applies $bd \to b$ to reduce $deb \to de$ to $be \leftrightarrow de$, which is oriented into $be \to de$, if $b \succ d$, and into $de \to be$, if $d \succ b$. (Were $\leadsto_{\mathrm{R}}$ equipped with a cancellation inference rule, $be \leftrightarrow de$ could be rewritten to $b \leftrightarrow d$, whence $b \to d$ or $d \to b$.) The deduction mechanism $\leadsto_{\mathrm{O}}$ can apply implicational overlap to $de \Rightarrow b$ and $b \Rightarrow d$ to generate $de \Rightarrow d$. However, $de \Rightarrow d$ is reduced to $de \Rightarrow \emptyset$ by the extensiveness rule, and $de \Rightarrow \emptyset$ is deleted by the definiteness rule. Thus, $\leadsto_{\mathrm{O}}$ does not generate anything that corresponds to $be \leftrightarrow de$. □

This example can be generalized to provide a simple analysis of simplification steps, one that shows which steps correspond to $\leadsto_{\mathrm{O}}$-inferences and which do not. Assume we have two rewrite rules $AB \to A$ and $CD \to C$, corresponding to non-trivial Horn clauses ($|B| = 1$, $B \not\subseteq A$, $|D| = 1$, $D \not\subseteq C$), and such that $CD \to C$ simplifies $AB \to A$. We distinguish three cases:

1. In the first one, CD appears in AB because CD appears in A. In other words, $A = CDE$ for some E. Then, the simplification step is

$$\frac{CDEB \to CDE,\ CD \to C}{\overline{CEB \to CE,\ CD \to C}}$$

(where simplification is actually applied to both sides). The corresponding implications are $A \Rightarrow B$ and $C \Rightarrow D$. Since $A \Rightarrow B$ is $CDE \Rightarrow B$, implicational overlap applies to generate the implication $CE \Rightarrow B$ that corresponds to $CEB \rightarrow CE$:

$$\frac{C \Rightarrow D,\ CDE \Rightarrow B}{CE \Rightarrow B}\,.$$

The isotony rule applied to $CE \Rightarrow B$ and $CDE \Rightarrow B$ reduces the latter to $CDE \Rightarrow \emptyset$, which is then deleted by the definiteness rule. Thus, a combination of implicational overlap, isotony and definiteness simulates the effects of simplification.

2. In the second case, CD appears in AB because C appears in A, that is, $A = CE$ for some E, and $D = B$. Then, the simplification step is

$$\frac{CEB \rightarrow CE,\ CB \rightarrow C}{\overline{CE \leftrightarrow CE,\ CB \rightarrow C}}\,,$$

and there is an isotony inference

$$\frac{C \Rightarrow B,\ CE \Rightarrow B}{\overline{C \Rightarrow B,\ CE \Rightarrow \emptyset}}\,,$$

which generates the trivial implication $CE \Rightarrow \emptyset$ corresponding to the trivial equation $CE \leftrightarrow CE$. Both are deleted by definiteness and deletion, respectively.

3. The third case is the generalization of Example 10: CD appears in AB because D appears in A, and C is made of B and some F that also appears in A, that is, $A = DEF$ for some E and F, and $C = BF$. The simplification step is

$$\frac{DEFB \rightarrow DEF,\ BFD \rightarrow BF}{\overline{BFE \leftrightarrow DEF,\ BFD \rightarrow BF}}\,.$$

Implicational overlap applies

$$\frac{DEF \Rightarrow B,\ BF \Rightarrow D}{DEF \Rightarrow D}$$

to generate an implication that is first reduced by extensiveness to $DEF \Rightarrow \emptyset$ and then eliminated by definiteness. Thus, nothing corresponding to $BFE \leftrightarrow DEF$ is generated.

It follows that whatever is generated by $\rightsquigarrow_{\mathrm{O}}$ is generated by $\rightsquigarrow_{\mathrm{R}}$, but may become redundant eventually:

Theorem 3. *For every implicational system S, for all fair and contracting derivations $S = S_0 \rightsquigarrow_{\mathrm{O}} S_1 \rightsquigarrow_{\mathrm{O}} \cdots$ and $R_S = R_0 \rightsquigarrow_{\mathrm{R}} R_1 \rightsquigarrow_{\mathrm{R}} \cdots$, for all $FG \rightarrow F \in R_{(S_\infty)}$, either $FG \rightarrow F \in (R_S)_\infty$ or $FG \rightarrow F$ is redundant in $(R_S)_\infty$.*

Proof. For all $FG \to F \in R_{(S_\infty)}$, $F \Rightarrow G \in S_\infty$ by Definition 3, and $F \Rightarrow G \in S_j$ for some $j \geq 0$. Let j be the smallest such index. If $j = 0$, or $S_j = S$, $FG \to F \in R_S = R_0$ by Definition 3. If $j > 0$, $F \Rightarrow G$ was generated by an application of implicational overlap, the isotony rule or extensiveness. By Lemma 3 and the assumption that the $\leadsto_{\mathrm{R}}$-derivation is fair and contracting, $FG \to F \in R_k$ for some $k > 0$. In both cases, $FG \to F \in R_k$ for some $k \geq 0$. If $FG \to F$ persists, then $FG \to F \in (R_S)_\infty$. Otherwise, $FG \to F$ is rewritten by simplification and is therefore redundant in $(R_S)_\infty$. □

Since the limit of a fair and contracting $\leadsto_{\mathrm{O}}$-derivation is $O(S)$, it follows that:

Corollary 3. *For every implicational system S, for all fair and contracting derivations $S = S_0 \leadsto_{\mathrm{O}} S_1 \leadsto_{\mathrm{O}} \cdots$ and $R_S = R_0 \leadsto_{\mathrm{R}} R_1 \leadsto_{\mathrm{R}} \cdots$, and for all $FG \to F \in R_{O(S)}$, either $FG \to F \in (R_S)_\infty$ or $FG \to F$ is redundant in $(R_S)_\infty$.*

4 Conditional Rewrite Systems

In this section we investigate canonicity in *conditional equational theories*, focusing on the ground case. We study conditional reduction and we propose a notion of *reducedness*, where also conditions themselves are subject to reduction, so that it may be possible to "reduce" overly-complex conditions, without affecting the equality relation. It follows that for conditional equational theories, unlike for equational ones, being *reduced* and being *contracted* are distinct. As a consequence, perfect systems – complete and reduced – and canonical systems – saturated and contracted – also differ.

4.1 Decreasing Systems

To use conditional equations for simplification, one needs to establish that the conditions hold. If testing the validity of the conditions yields a problem that is as difficult as the one we would like to solve by applying conditional equations, conditional simplification becomes unpractical. In other words, the complexity of conditions should be bounded. Therefore, we start with a notion of *decreasingness*, which appeared in [30] and ensures that testing conditions does not yield bigger problems:

Definition 8 (Decreasing conditional equation). *A ground conditional equation $p_1 \simeq q_1, \cdots, p_n \simeq q_n \Rightarrow l \simeq r$ is* decreasing *if $l \succ r, p_1, q_1, \ldots, p_n, q_n$; a conditional equation is* decreasing *if all its ground instances are.*

A *decreasing inference* is an application of the following inference rule:

$$\frac{C \Rightarrow l \simeq r \quad w_1 \quad \ldots \quad w_n}{C \setminus \{w_1, \ldots w_n\} \Rightarrow f[l] \simeq f[r]} \qquad f[l] \simeq f[r] \succ_C C$$

where f is any context and $w_1 \ldots w_n$ are equations. If $C \setminus \{w_1, \ldots w_n\} = \emptyset$, an equation is deduced; otherwise, a conditional equation is deduced, where those conditions that are not discharged remain part of the conclusion. Condition $f[l] \simeq f[r] \succ_C C$ characterizes the inference as *decreasing*. Since $\succ$ is a simplification ordering and therefore has the subterm property, $f[l] \simeq f[r] \succ_C l \simeq r$ also holds. Thus, $f[l] \simeq f[r] \succ_C (C \Rightarrow l \simeq r)$ follows. On the other hand, the subproofs of the w_i may contain larger premises.

The *depth of a decreasing inference* is 0 if $f[l] = f[r]$ (a trivial equation is deduced) or $n = 0$ (no subproofs). Otherwise, it is 1. The *depth of a proof* is the sum of the depth of its inferences, that is, the number of non-trivial inferences where a conditional equation is applied and some if its conditions are discharged. Thus, purely equational proofs have depth 0, because they do not have conditions.

Definition 9 (Equivalence). *Given a presentation S of a theory, two terms s and t are* S-equivalent*, written $s \equiv_S t$, if there is a proof p, such that $[p]_{Cl} = s \simeq t$ and $[p]^{Pm} \subseteq S$, made of decreasing inferences.*

We can use minimal elements of S-equivalence classes as their representatives:

Definition 10 (Normal form). *The* S-normal form *of a term t is the $\succ$-minimal element of its S-equivalence class.*

By the same token, a term t is in *normal form* with respect to S, if it is its own S-normal form.

4.2 Reduced Systems

Given a set S of conditional equations, we are interested in a reduced version of S. Computing a reduced system involves deletion of trivial conditional equations, subsumption and simplification, as defined by the following inference rules:

Deletion

$$\frac{C \Rightarrow r \simeq r}{} \qquad \frac{C, l \simeq r \Rightarrow l \simeq r}{}$$

Subsumption

$$\frac{C, D \Rightarrow u[l] \simeq u[r] \quad C \Rightarrow l \simeq r}{C \Rightarrow l \simeq r}$$

Simplification

$$\frac{C, p \simeq q \Rightarrow l[p] \simeq r}{C, p \simeq q \Rightarrow l[q] \simeq r} \quad p \succ q \qquad\qquad \frac{C, p \simeq q, u[p] \simeq v \Rightarrow l \simeq r}{C, p \simeq q, u[q] \simeq v \Rightarrow l \simeq r} \quad p \succ q$$

$$\frac{C, D \Rightarrow l[u] \simeq r \quad C \Rightarrow u \simeq v}{C, D \Rightarrow l[v] \simeq r \quad C \Rightarrow u \simeq v} \quad u \succ v \ ,$$

where the first two simplification rules use a condition to simplify the consequence or another condition of the same conditional equation, while the third one applies a conditional equation $C \Rightarrow u \simeq v$ to simplify another conditional equation whose conditions include C. Inferences shown on the left-hand side of $\simeq$ apply also to the right-hand side, since $\simeq$ is symmetric.

These inference rules produce a reduced system according to the following definition:

Definition 11 (S-reduced). *Let $S = S' \uplus \{e\}$ be a presentation, where $e = (C \Rightarrow l \simeq r)$ is a conditional equation, $C = \{p_i \simeq q_i\}_{i=1}^n$, and, for convenience, $l \succ r$ and $p_i \succ q_i$, for all i, $1 \leq i \leq n$. Then, e is S-*reduced *if*

1. *e is not trivial,*
2. *no conditional equation in S' subsumes e,*
3. *l is in $(S' \cup C)$-normal form,*
4. *r is in $(S \cup C)$-normal form,*
5. *for all i, $1 \leq i \leq n$,*
   (a) *p_i is in $(S' \cup (C \setminus \{p_i \simeq q_i\}))$-normal form and*
   (b) *q_i is in $(S' \cup C)$-normal form.*

The difference between Item 3 and Item 4 is designed to prevent $C \Rightarrow l \simeq r$ from simplifying itself. In Item 5, a condition $p \simeq q \in C$ is normalized also with respect to the other equalities in C, because all equalities in C must be true to apply a conditional equation e. Thus, the notion of reducedness incorporates the notion of reduction with respect to a context as in the *conditional contextual rewriting* proposed by Zhang [74]. The difference between Item 5a and Item 5b is meant to prevent $p_i \simeq q_i$ from simplifying itself. Thus, we can safely define the following:

Definition 12 (Self-reduced). *A conditional equation e is* self-reduced*, if it is $\{e\}$-reduced. The* self-reduced *form of e is denoted $e^\flat$.*

Example 11. For $S = \{e_1,\ e_2\}$, where e_1 is $a \simeq b \Rightarrow c \simeq d$, and e_2 is $f(a) \simeq f(b) \Rightarrow c \simeq d$, as in Example 1, both e_1 and e_2 are S-reduced. □

Definition 13 (Reduced). *A presentation S is* reduced*, if all its elements are S-reduced.*

Definition 14 (Perfect). *A presentation S is* perfect*, if it is complete and reduced.*

Example 12. Let $S = \{e_1, e_2\}$, where e_1 is $a \simeq b \Rightarrow f(a) \simeq c$ and e_2 is $a \simeq b \Rightarrow f(b) \simeq c$, with $f > a > b > c$. The presentation S is not reduced, because clause e_1 is not. Indeed, the normal form of $f(a)$ with respect to $(S \setminus \{e_1\}) \cup \{a \simeq b\}$ is c, and the reduced form of e_1 is the trivial clause $a \simeq b \Rightarrow c \simeq c$. Clause e_2 is reduced. □

Proposition 3. *If S is reduced, then it is contracted.*

Proof. Assume that S is not contracted. Then, there exists an $e \in S$, such that $e \notin [\mu Pf(S)]^{Pm}$ (see Sect. 2.2). In other words, if $e \in [p]^{Pm}$, then $p \notin \mu Pf(S)$. For each such p, there is a $q \in \mu Pf(S)$, such that $p > q$ and $e \notin [q]^{Pm}$. By monotonicity of the proof ordering with respect to subproofs (cf. Property 3 in Sect. 2.2), proof p must contain a subproof involving e, possibly consisting of e itself, which is replaced by a smaller subproof in q. Thus, proof p and premise e must contain at least a term that is not in S-normal form. Hence, e is not in S-reduced form, and S is not reduced. □

On the other hand, a presentation can be contracted but not reduced, as shown in the following example:

Example 13. If $a > b > c$, neither $a \simeq b \Rightarrow b \simeq c$ nor $a \simeq b \Rightarrow a \simeq c$ is decreasing. The presentations $S_1 = \{a \simeq b \Rightarrow b \simeq c\}$, $S_2 = \{a \simeq b \Rightarrow a \simeq c\}$, and $S_3 = \{a \simeq b \Rightarrow b \simeq c,\ a \simeq b \Rightarrow a \simeq c\}$ are equivalent. However, S_1 is reduced, whereas S_2 is not, since the S_2-reduced form of $a \simeq b \Rightarrow a \simeq c$ is $a \simeq b \Rightarrow b \simeq c$. Neither is S_3 reduced, although it is contracted. Indeed, while $a \simeq b \Rightarrow b \simeq c$ is S_3-reduced, the S_3-reduced form of $a \simeq b \Rightarrow a \simeq c$ is the trivial clause $a \simeq c \Rightarrow a \simeq c$. □

Unlike the ground equational case, where contracted and canonical collapse to reduced, because all inferences consist of rewriting, in the conditional case contracted and reduced are different (like S_3 in Example 13). Furthermore, decreasing simplification is "incomplete" with respect to Definition 11, because non-reduced presentations may not be reducible by decreasing simplification, as the clauses are not decreasing (like S_2 and S_3 in Example 13). The following lemma and theorem follow from the definitions.

Lemma 4. *A conditional equation e has a normal-form proof in presentation S, if the S-reduced form of $e^\flat$ is subsumed by a conditional equation in S.*

Theorem 4. *If S is canonical, then S subsumes the S-reduced form of every theorem of S.*

Example 14. Consider again the three presentations of Example 13, $e_1 = a \simeq b \Rightarrow b \simeq c$ and $e_2 = a \simeq b \Rightarrow a \simeq c$. We have $e_1^\flat = a \simeq b \Rightarrow b \simeq c = e_2^\flat$. Thus, both S_1 and S_3 are complete (and saturated), because they contain $a \simeq b \Rightarrow b \simeq c$. On the other hand, S_2 does not, and therefore it is not complete. In summary, S_1 is perfect (reduced and complete), S_3 is canonical (contracted and saturated), whereas S_2 is neither. □

In summary, in the conditional case, a perfect system – complete and reduced – and a canonical system – saturated and contracted – differ in two ways: complete is weaker than saturated, and reduced is stronger than contracted. Since complete/saturated determines how much expansion is required in completion, whereas reduced/contracted refers to how much simplification completion should feature, both discrepancies hint that the perfect system is really the best system, as the name suggests.

5 Horn Normal Forms

In this section, we survey proof normalization and decision procedures, based on canonical systems, in Horn theories and beyond. Since $Th\, S$ is defined based on proofs (cf. Sect. 2.2), the choice of normal-form proofs is intertwined with the choice of the deduction mechanism that generates the proofs. This double choice is guided by the purpose of ensuring that $S^\sharp$ forms the basis for a decision procedure. To achieve decidability, the various notions of normal-form proof aim at minimizing non-deterministic choice-points that require search. Then, Horn proofs may have the following qualities:

- *Linear*: in *linear resolution proofs* at each step a *center clause* is resolved with a *side clause*, to generate the next center clause (see, for instance, Chapter 7 in the book by Chang and Lee [24]). The first center clause, or *top clause*, is the goal given by the problem. Linearity eliminates one choice point, because the main premise of the next step must be whatever was generated by the previous step.
- *Linear input*: the choice of side clauses is restricted to input clauses [24].
- *Reducing*: a linear proof is reducing if each center clause is smaller than its predecessor in the ordering $\succ_C$ – this implies termination [19].
- *Unit-resulting*: each step must generate a unit clause; thus, all literals but one must be resolved away, which eliminates the choice of literal in the center clause, but may require multiple side clauses (traditionally called *satellites* or *electrons* as in the *unit-resulting resolution* of McCharen, Overbeek and Wos [65]).
- *Confluent*: whatever choices are left, such as choice of side premise(s) or choice of subterm, are irrelevant for finding or not finding a proof, which means they will never need to be undone by backtracking.

Valley proofs for purely equational theories satisfy all these properties, some vacuously (like unit-resulting). In this section we survey different choices of normal-form proofs for Horn theories, and we examine how they yield different requirements on canonical presentations, and on the completion procedures that generate them at the limit.

5.1 Trivial Proofs

If trivial proofs are assumed to be normal-form proofs, closure with respect to *forward chaining* gives the canonical presentation. Canonical, saturated and complete coincide. Given a Horn presentation S, $S^\sharp$ is made of all ground facts that follow from S and the axioms of equality by forward chaining. In other words, $S^\sharp$ is the least Herbrand model of S, and, equivalently, the least fixed-point of the mapping associated to program S and the axioms of equality in the fixed point semantics of logic programming (see the aforementioned surveys [1,51] or Lloyd's book [63]).

Existence of the least Herbrand model is a consequence of the defining property of Horn theories, namely closure of the family of models with respect to intersection. This is also the basis upon which to draw a correspondence between Horn clauses with *unary* predicate symbols and certain tree automata, called *two-way alternating tree automata* (cf. [25, Sec. 7.6.3]). Tree automata are automata that accept trees, or, equivalently, terms. Given a Horn presentation S, the predicate symbols in S are the states of the automaton. As usual, a subset of states is defined to be *final*. Then, the essence of the correspondence is that a ground term t is accepted by the automaton if the atom $r(t)$ is in $S^\sharp$ and r is final. The deduction mechanism for computing the accepted terms is still *forward chaining*. It is sufficient to have unary predicate symbols, because the notion of being accepted applies to *one* term at a time. This restriction is advantageous because many properties in the *monadic fragment* are decidable. For the class of two-way alternating tree automata, clauses are further restricted to have one of the following forms:

1. $a_1(x_1), \ldots, a_n(x_n) \Rightarrow c(u)$, where $x_1, \ldots, x_n$ are (not necessarily distinct) variables, u is a linear, non-variable term, and $x_1, \ldots, x_n \in \mathit{Var}(u)$;
2. $a(u) \Rightarrow c(x)$, where u is a linear term and x is a variable; and
3. $a_1(x), \ldots, a_n(x) \Rightarrow c(x)$.

We refer the interested reader to [25] for more details and results.

5.2 Ground-Preserving Linear Input Proofs

According to Kounalis and Rusinowitch [61], normal-form proofs for Horn theories with equality are *linear input* proofs by ordered resolution and ordered superposition, where only maximal literals are resolved upon, and only maximal sides of maximal literals are superposed into and from. Furthermore, in order to have a normal-form proof, all side clauses $p_1 \simeq q_1, \cdots, p_n \simeq q_n \Rightarrow l \simeq r$ in the proof must be *ground-preserving*: $\mathit{Var}(p_i \simeq q_i) \subseteq \mathit{Var}(l \simeq r)$, for all i, $1 \leq i \leq n$, and either $l \succ r$ or $r \succ l$, or $\mathit{Var}(l) = \mathit{Var}(r)$.

A conjecture is a conjunction $\forall \bar{x}\ u_1 \simeq v_1, \ldots, u_k \simeq v_k$, whose negation is a ground (Skolemized) negative clause $\widetilde{u}_1 \not\simeq \widetilde{v}_1 \vee \cdots \vee \widetilde{u}_k \not\simeq \widetilde{v}_k$. If all side clauses are ground-preserving and the top clause is ground, all center clauses will also be ground. This, together with the ordering restrictions on resolution and superposition and the assumption that the ordering is a complete simplification ordering – hence, is total on ground terms, literals and clauses – imply that every center clause is smaller than its parent center clause, so that proofs are *reducing*.

Therefore, a finite presentation that features such a normal-form proof for every conjunction of positive literals is a *decision procedure*. The *Horn completion procedure* of [61], with ordered resolution, superposition, simplification by conditional equations, and subsumption, generates at the limit a *saturated* presentation, which is such a decision procedure, if it is finite and all its clauses are ground-preserving.

5.3 Linear Input Unit-Resulting Proofs

An approach for Horn logic without equality was studied by Baumgartner in his book on *theory reasoning* [9]. Here normal-form proofs of conjunctions of positive literals are *linear input unit-resulting* (UR) resolution proofs. A completion procedure, called *Linearizing Completion*, applies selected resolution inferences and additions of contrapositives to compile the given presentation into one that offers normal-form proofs for all conjunctions of positive literals. The name "Linearizing" evokes the transformation of UR-resolution proofs (not in normal form) into linear UR-resolution proofs (in normal form).

If finite, the resulting saturated presentation is used as a decision procedure for the Horn theory in the context of *partial theory model elimination*. Similar to *partial theory resolution* of Stickel [71], a *decision procedure* that generates conditions for the unsatisfiability of a set of literals in the theory, as opposed to deciding unsatisfiability, suffices. The saturated presentation generated by Linearizing Completion is a decision procedure in this weaker sense.

5.4 Valley Proofs

If the notion of normal-form proof of the unconditional case is generalized to the conditional case, normal-form proofs are valley proofs of depth 0, where all conditions have been solved away. The *Maximal Unit Strategy* of [31] achieves this effect by restricting expansion inferences to have at least one unit premise: it applies superposition between unconditional equations and to superpose unconditional equations into maximal terms of conditions.

In the limit, the saturated set contains all positive unit theorems, or, equivalently, all conditional equations are redundant [19], so that there is a normal-form proof for every theorem. However, such a presentation will be infinite in most cases, so that the *Maximal Unit Strategy* is better seen as a *semi-decision procedure* for forward-reasoning theorem proving, rather than as a generator of decision procedures [30].

5.5 Nested Valley Proofs

In [30,31], a normal-form proof of $s \simeq t$ is a valley proof, in which each subproof is also in normal form, and each term in a subproof is smaller than the greater of s and t. To enforce the latter constraint, only *decreasing instances of conditional equations* are applied. The *Decreasing Strategy* of [30,31] simplifies by *decreasing instances* of conditional equations and applies ordered paramodulation/superposition of decreasing instances, to generate at the limit a saturated presentation that features normal-form proofs for all theorems. Our analysis in Sect. 4 started from this point to develop the notions of reduced and perfect system, showing the incompleteness of decreasing simplification with respect to reducedness and the difference between canonical and perfect system.

If we compare this notion of normal-form proof based on decreasingness with the conditional valley proofs of null depth of Sect. 5.4, we see that decreasingness allows *nested* valley proofs, or, equivalently, it does not require that normal-form proofs have depth 0: this means renouncing linearity.

To compare with the ground-preserving linear input proofs of Sect. 5.2, consider a conditional equation that is not ground-preserving, such as

$p_1 \simeq q_1, \cdots, p_n \simeq q_n \Rightarrow l \simeq r$, where $l \succ r$ and either r, or one of the p_i's, or one of the q_i's, for some i, $1 \leq i \leq n$, contains a variable that does not occur in l. Such a conditional equation cannot be decreasing. However, the motivations for the two conditions are different. The motivation for the ground-preserving property is to ensure that proofs are reducing. The motivation for decreasingness, which improved upon previous suggestions in [58,57], is to capture exactly the finiteness of recursive evaluation of terms.

Another significant difference between decreasingness, on one hand, and earlier requirements, on the other, including the ground-preserving condition and the requirements studied by Kaplan and Rémy [59] or Ganzinger [48], is that they are static properties of conditional rewrite rules or equations, whereas decreasingness is tested dynamically on the applied instances. This difference resembles the one between Knuth-Bendix completion [60], where all equations must be oriented, and Unfailing, or Ordered, Completion, that applies oriented instances of unoriented equations [21,62,55,6,5,19].

5.6 Quasi-Horn Theories

A generalization of the approach of Sect. 5.2 was given by Bachmair and Ganzinger in [7], by considering *quasi-Horn clauses* and replacing the ground-preserving property with the *universally reductive* property.

A clause C is *quasi-Horn* if it has at most one positive equational literal, and, if there is one – say $l \simeq r$, then $(l \simeq r)\sigma$ is maximal in $C\sigma$ for all ground instances $C\sigma$ of C. A general clause C is *universally reductive* if it contains a literal L such that (i) $Var(C) \subseteq Var(L)$, (ii) for all ground substitutions σ, $L\sigma$ is strictly maximal in $C\sigma$, (iii) if L is an equational literal, it is a positive equation $s \simeq t$, such that $Var(s \simeq t) \subseteq Var(s)$ and for all ground substitutions σ, $s\sigma \succ t\sigma$. Clause C is said to be universally reductive for L. Clearly, if a quasi-Horn clause that contains a positive equation is universally reductive, it is universally reductive for the positive equation.

A quasi-Horn clause is more general than a Horn clause, because it allows more than one positive literal, provided they are not equations: if there is a positive equation, then it must be unique and maximal. A quasi-Horn clause C that contains a positive equation $l \simeq r$ will be involved only in superposition inferences into, or from, $l \simeq r$: ordered resolution does not apply to C, because its non-equational literals are not maximal; ordered factoring and equality resolution,[4] are not applicable either, because C has only one positive literal and its negative literals are not maximal. Furthermore, superposition of C into a clause without positive equations will produce another clause without positive equations. In essence, the notion of quasi-Horn clause serves the purpose of making sure that the equational part of the problem is Horn, and can be dealt with separately with respect to the non-equational part, which may be non-Horn and require ordered resolution and ordered factoring.

[4] Equality resolution is ordered resolution with $x \simeq x$.

The notion of goal is generalized from ground negative clause to ground clause without positive equations, and the notion of normal-form proof for such a goal is weakened accordingly: the equational reasoning part by ordered superposition is *linear*, whereas the ordered resolution and ordered factoring part for the non-equational component is not necessarily linear. A finite saturated set of universally-reductive quasi-Horn clauses is a *decision procedure* in that it provides a normal-form proof for all goals in this form.

5.7 Beyond Quasi-Horn

It is well known that the restrictions of general inferences that are complete for Horn logic (including *linear input resolution, unit resolution, forward chaining*) are not complete for full first-order logic (see [24]). In the non-equational case, linear input proofs must be replaced by linear proofs, involving also *factoring* and *ancestor-resolution* inferences. In the presence of equality, one needs to deal with the interplay of the equational and non-equational parts in its full generality. Nevertheless, completion procedures to generate saturated or canonical presentations have been investigated also in the unrestricted first-order context. One purpose is to find whether inference systems or strategies that are not complete for first-order logic, may become complete if a canonical, or at least saturated, presentation is given.

An example is the classical *resolution with set of support* of Wos et al. [72], where the set of support initially contains the goal clauses (those resulting from the negation of the conjecture), its complement contains the presentation, and all generated clauses are added to the set of support. This is complete for resolution in first-order logic, but it is not complete for ordered resolution and superposition in first-order logic with equality. However, it is well known that, if the presentation is saturated, then the set of support strategy is complete also for first-order logic with equality and ordered inferences, for the simple reason that all inferences from the saturated presentation are redundant.

For first-order theories, in general, there is no finite canonical presentation that forms the basis for a decision procedure. Obtaining decision procedures for fragments of first-order logic rests on some combination of saturation by completion and syntactic constraints on the presentation. A survey can be found in [43]. More recent results based on syntactic constraints include those of Comon-Lundh and Courtier in [26]. In [41], Dowek studied proof normalization in the context of a sequent calculus modulo a congruence on terms, where normal-form proofs are cut-free proofs.

Another thread of research on decision procedures is that of *satisfiability modulo a theory* (SMT), where T*-satisfiability* is the problem of deciding satisfiability of a set of ground literals in theory T. Armando et al. [3,2] proved that a rewrite-based inference system for first-order logic with equality is guaranteed to generate finitely many clauses when applied to T-satisfiability problems in several theories of data structures, and any of their combinations. Thus, such an inference system equipped with a fair search plan yields a decision procedure for T-satisfiability in those theories. Bonacina and Echenim generalized this approach to T-satisfiability in theories of *recursive data structures* [15], extended it to decide T-satisfiability of arbitrary ground formulæ [16], and investigated how using a rewrite-based inference system as a pre-processor for an SMT-solver yields decision procedures [17]. Lynch and Morawska [64] combined the approach of [3] with syntactic constraints to obtain complexity bounds for some theories. Bonacina, Lynch and de Moura obtained more decision procedures by equipping a theorem prover that integrates superposition into an SMT-solver with *speculative inferences* [20].

5.8 Knowledge Representation

In the context of knowledge representation, various forms of *knowledge compilation* have been studied to transform a given knowledge base into a normal form that enables efficient reasoning. Roussel and Mathieu [69] investigated the problem of "completing" a knowledge base, so that forward chaining becomes complete also in the first-order case (without equality). An *achieved* knowledge base corresponds to a saturated presentation, and the process that generates it is called *achievement*.

Clearly, in many instances an achieved knowledge base that is equivalent to the original one will be infinite, so that one has to resort to either *partial achievement* or *total achievement* techniques. Partial achievement produces a finite knowledge base by setting a limit on either the depth of instances, or the length of chains of literals, that may be produced. Total achievement relaxes, in a controlled way, the requirement that the achieved base be equivalent to the original one. More recently, Furbach and Obermaier [45] considered knowledge compilation in description logics.

6 Discussion

Knuth-Bendix completion [60,56,55,6,5,19] was designed to derive decision procedures for validity in algebraic theories. Its outstanding feature is the use of inferred rules to continuously reduce equations and rules during the inference process. As a byproduct, the resulting reduced convergent system is unique – given a well-founded ordering of terms for orienting equations into rules [36] – and appropriately viewed as *canonical*.

In the ground equational case, reduction and completion are one and the same [62,46,67,8,14]. The natural next step up is to consider what canonical ground Horn presentations might look like. Here, we take a new look at ground Horn theories from the point of view of the theory of canonical inference initiated in [34,14]. Of course, entailment of equational Horn clauses is also easily decidable in the propositional [42] and ground [47] cases. However, it turns out that reduced and canonical – hence, reduction and completion – are distinct in this case.

For implicational systems, we analyzed the notions of direct and direct-optimal implicational system in terms of completion and canonicity. We compared implicational systems with inference mechanisms featuring implicational overlap and optimization, and rewrite systems with inference mechanisms featuring equational overlap and simplification. We found that a direct implicational system corresponds to the canonical limit of a derivation by completion that features expansion by equational overlap and contraction by forward simplification. When completion also features backward simplification and is given a subset of the alphabet as input, together with the implicational system, it computes the image of the subset with respect to the closure operator associated with the implicational system. In other words, it computes the minimal model that satisfies both the implicational system and the subset. On the other hand, a direct-optimal implicational system does not correspond to the limit of a derivation by completion, because the underlying proof orderings are different and therefore normalization induces two different notions of optimization. Accordingly, we introduced a new notion of optimality for implicational systems, termed *rewrite optimality*, that corresponds to canonicity defined by completion up to redundancy.

Although limited to the propositional level, our analysis is complementary to those of [29,38,18,40] in a few ways. First, previous studies primarily compared answering a query with respect to a program of definite clauses, interpreted by SLD-resolution, as in Prolog, with answering a query with respect to a program of rewrite rules, interpreted by linear completion, with equational overlap, with or without simplification. Thus, from an operational point of view, those analyses focused on *backward reasoning* from the query, whereas ours concentrates on optimizing and completing presentations by *forward reasoning*. Second, SLD-resolution involves no contraction, so that earlier comparisons placed an inference mechanism with contraction (linear completion) side-by-side with one without. The treatment in [18] included the case where the Prolog interpreter is enriched with subsumption, but it was only subsumption between goals, with no contraction of the presentation. Here we have also compared different forms of contraction, putting optimization of implicational systems and simplification of rewrite systems in parallel. The present analysis agrees with prior ones in indicating the rôle of simplification in differentiating reasoning by completion about equivalences from reasoning about implications. Indeed, as we have seen, the canonical rewrite system can be more reduced than the rewrite-optimal implicational system (cf. Theorem 3).

Future work includes generalizing this analysis to non-ground Horn theories, similar to what was done in [22] to extend the application of the abstract framework of [34,14] from ground completion to standard completion of equational theories. Other directions may be opened by exploring new connections between canonical systems and decision procedures.

Acknowledgements. We thank Andreas Podelski, Andrei Voronkov and Reinhard Wilhelm for organizing the workshop in Harald's memory in Saarbrücken in June 2005.

References

1. Krzysztof R. Apt. Logic programming. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Methods and Semantics, chapter 10, pages 493–574. North-Holland, Amsterdam, 1990.
2. Alessandro Armando, Maria Paola Bonacina, Silvio Ranise, and Stephan Schulz. New results on rewrite-based satisfiability procedures. *ACM Transactions on Computational Logic*, 10(1):129–179, January 2009.
3. Alessandro Armando, Silvio Ranise, and Michaël Rusinowitch. A rewriting approach to satisfiability procedures. *Information and Computation*, 183(2):140–164, 2003.
4. Leo Bachmair and Nachum Dershowitz. Inference rules for rewrite-based first-order theorem proving. In *Proceedings of the Second Annual IEEE Symposium on Logic in Computer Science*, pages 331–337. IEEE Computer Society Press, 1987.
5. Leo Bachmair and Nachum Dershowitz. Equational inference, canonical proofs, and proof orderings. *Journal of the Association for Computing Machinery*, 41(2):236–276, 1994.
6. Leo Bachmair, Nachum Dershowitz, and David A. Plaisted. Completion without failure. In Hassam Aït-Kaci and Maurice Nivat, editors, *Resolution of Equations in Algebraic Structures*, volume II: Rewriting Techniques, pages 1–30. Academic Press, 1989.
7. Leo Bachmair and Harald Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4:217–247, 1994.
8. Leo Bachmair, Ashish Tiwari, and Laurent Vigneron. Abstract congruence closure. *Journal of Automated Reasoning*, 31(2):129–168, 2003.
9. Peter Baumgartner. *Theory Reasoning in Connection Calculi*, volume 1527 of *Lecture Notes in Artificial Intelligence*. Springer, Heidelberg, 1998.
10. Karell Bertet and Bernard Monjardet. The multiple facets of the canonical direct implicational basis. Cahiers de la MSE No. b05052, Maison des Sciences Economiques, Université Paris Panthéon-Sorbonne, June 2005. `http://ideas.repec.org/p/mse/wpsorb/b05052.html`.
11. Karell Bertet and Mirabelle Nebut. Efficient algorithms on the Moore family associated to an implicational system. *Discrete Mathematics and Theoretical Computer Science*, 6:315–338, 2004.
12. Marc Bezem, Jan Willem Klop, and Roel de Vrijer (Editors). *TeReSe: Term Rewriting Systems*. Cambridge University Press, Cambridge, UK, 2003.
13. Garrett Birkhoff. *Lattice Theory*. American Mathematical Society, New York, Revised edition, 1948.
14. Maria Paola Bonacina and Nachum Dershowitz. Abstract canonical inference. *ACM Transactions on Computational Logic*, 8(1), January 2007.
15. Maria Paola Bonacina and Mnacho Echenim. Rewrite-based satisfiability procedures for recursive data structures. In Byron Cook and Roberto Sebastiani, editors, *Proceedings of the Fourth Workshop on Pragmatics of Decision Procedures in Automated Reasoning (PDPAR), Fourth Federated Logic Conference (FLoC), August 2006*, volume 174(8) of *Electronic Notes in Theoretical Computer Science*, pages 55–70. Elsevier, Amsterdam, June 2007.
16. Maria Paola Bonacina and Mnacho Echenim. On variable-inactivity and polynomial T-satisfiability procedures. *Journal of Logic and Computation*, 18(1):77–96, February 2008.

17. Maria Paola Bonacina and Mnacho Echenim. Theory decision by decomposition. *Journal of Symbolic Computation*, 45(2):229–260, February 2010.
18. Maria Paola Bonacina and Jieh Hsiang. On rewrite programs: Semantics and relationship with Prolog. *Journal of Logic Programming*, 14(1 & 2):155–180, October 1992.
19. Maria Paola Bonacina and Jieh Hsiang. Towards a foundation of completion procedures as semidecision procedures. *Theoretical Computer Science*, 146:199–242, July 1995.
20. Maria Paola Bonacina, Christopher A. Lynch, and Leonardo de Moura. On deciding satisfiability by theorem proving with speculative inferences. *Journal of Automated Reasoning*, 47(2):161–189, August 2011.
21. Thomas Carl Brown, Jr. *A Structured Design-Method for Specialized Proof Procedures*. PhD thesis, California Institute of Technology, Pasadena, 1975.
22. Guillaume Burel and Claude Kirchner. Completion is an instance of abstract canonical system inference. In Kokichi Futatsugi, Jean-Pierre Jouannaud, and Josè Meseguer, editors, *Algebra, Meaning and Computation – Essays in Honor of Joseph Goguen*, volume 4060 of *Lecture Notes in Computer Science*, pages 497–520. Springer, Heidelberg, 2006.
23. Nathalie Caspard and Bernard Monjardet. The lattice of Moore families and closure operators on a finite set: A survey. *Electronic Notes in Discrete Mathematics*, 2, 1999.
24. Chin-Liang Chang and Richard Char-Tung Lee. *Symbolic Logic and Mechanical Theorem Proving*. Academic Press, 1973.
25. Hubert Comon, Max Dauchet, Rémi Gilleron, Florent Jacquemard, Denis Lugiez, Sophie Tison, and Marc Tommasi. Tree Automata Techniques and Applications. `http://www.grappa.univ-lille3.fr/tata`, 2005.
26. Hubert Comon-Lundh and Veronique Courtier. New decidability results for fragments of first-order logic and application to cryptographic protocols. In Robert Nieuwenhuis, editor, *Proceedings of the Fourteenth Conference on Rewriting Techniques and Applications (RTA)*, volume 2706 of *Lecture Notes in Computer Science*, pages 148–164. Springer, Heidelberg, 2003.
27. Nachum Dershowitz. A note on simplification orderings. *Information Processing Letters*, 9(5):212–215, November 1979.
28. Nachum Dershowitz. Equations as programming language. In *Proceedings of the Fourth Jerusalem Conference on Information Technology*, pages 114–124. IEEE Computer Society Press, May 1984.
29. Nachum Dershowitz. Computing with rewrite systems. *Information and Control*, 64(2/3):122–157, May-June 1985.
30. Nachum Dershowitz. Canonical sets of Horn clauses. In Javier Leach Albert, Burkhard Monien, and Mario Rodríguez Artalejo, editors, *Proceedings of the Eighteenth International Colloquium on Automata, Languages and Programming (ICALP)*, volume 510 of *Lecture Notes in Computer Science*, pages 267–278. Springer, Heidelberg, July 1991.
31. Nachum Dershowitz. Ordering-based strategies for Horn clauses. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 118–124, Sydney, August 1991.
32. Nachum Dershowitz, Guan-Shieng Huang, and Mitchell A. Harris. Enumeration problems related to ground Horn theories. `http://arxiv.org/abs/cs/0610054`, 2008.

33. Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Methods and Semantics, chapter 6, pages 243–320. North-Holland, Amsterdam, 1990.
34. Nachum Dershowitz and Claude Kirchner. Abstract canonical presentations. *Theoretical Computer Science*, 357:53–69, 2006.
35. Nachum Dershowitz and Zohar Manna. Proving termination with multiset orderings. *Communications of the ACM*, 22(8):465–476, 1979.
36. Nachum Dershowitz, Leo Marcus, and Andrzej Tarlecki. Existence, uniqueness, and construction of rewrite systems. *SIAM Journal of Computing*, 17(4):629–639, 1988.
37. Nachum Dershowitz and David A. Plaisted. Logic programming *cum* applicative programming. In *Proceedings of the Symposium on Logic Programming*, pages 54–66. IEEE Computer Society Press, 1985.
38. Nachum Dershowitz and David A. Plaisted. Equational programming. In J. E. Hayes, Donald Michie, and J. Richards, editors, *Machine Intelligence 11: The logic and acquisition of knowledge*, chapter 2, pages 21–56. Oxford University Press, Oxford, 1988.
39. Nachum Dershowitz and David A. Plaisted. Rewriting. In John Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 9, pages 535–610. Elsevier, Amsterdam, 2001.
40. Nachum Dershowitz and Uday Reddy. Deductive and inductive synthesis of equational programs. *Journal of Symbolic Computation*, 15:467–494, 1993.
41. Gilles Dowek. Confluence as a cut elimination property. In Robert Nieuwenhuis, editor, *Proceedings of the Fourteenth Conference on Rewriting Techniques and Applications (RTA)*, volume 2706 of *Lecture Notes in Computer Science*, pages 2–13. Springer, Heidelberg, 2003.
42. William F. Dowling and Jean H. Gallier. Linear-time algorithms for testing the satisfiability of propositional Horn formulæ. *Journal of Logic Programming*, 1(3):267–284, 1984.
43. Christian Fermüller, Alexander Leitsch, Ulrich Hustadt, and Tanel Tammet. Resolution decision procedures. In John Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume II, chapter 25, pages 1793–1849. Elsevier, Amsterdam, 2001.
44. Laurent Fribourg. SLOG—Logic interpreter for equational clauses. In *Proceedings of the Fourth Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, number 415 in Lecture Notes in Computer Science, pages 479–480. Springer, Heidelberg, 1987.
45. Ulrich Furbach and Claudia Obermaier. Knowledge compilation for description logics. In Nachum Dershowitz and Andrei Voronkov, editors, *Proceedings of the Fourteenth International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR), Short papers session*. 2007.
46. Jean Gallier, Paliath Narendran, David A. Plaisted, Stan Raatz, and Wayne Snyder. Finding canonical rewriting systems equivalent to a finite set of ground equations in polynomial time. *Journal of the Association for Computing Machinery*, 40(1):1–16, 1993.
47. Jean H. Gallier. Fast algorithms for testing unsatisfiability of ground Horn clauses with equations. *Journal of Symbolic Computation*, 4:233–254, 1987.
48. Harald Ganzinger. A completion procedure for conditional equations. *Journal of Symbolic Computation*, 11(1 & 2):51–81, 1991.

49. Joseph A. Goguen and José Meseguer. EQLOG: Equality, types, and generic modules for logic programming. In D. DeGroot and G. Lindstrom, editors, *Logic Programming: Functions, Relations, and Equations*, pages 295–363. Prentice-Hall, Englewood Cliffs, 1986.
50. Michael Hanus. The integration of functions into logic programming: From theory to practice. *Journal of Logic Programming*, 19&20:583–628, 1994.
51. Wilfrid Hodges. Logical features of Horn clauses. In Dov M. Gabbay, C. J. Hogger, and John Alan Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume I: Logical Foundations, pages 449–503. Oxford University Press, Oxford, 1993.
52. Alfred Horn. On sentences which are true in direct unions of algebras. *Journal of Symbolic Logic*, 16:14–21, 1951.
53. Jieh Hsiang. Refutational theorem proving using term rewriting systems. *Artificial Intelligence*, 25:255–300, 1985.
54. Jieh Hsiang. Rewrite method for theorem proving in first order theories with equality. *Journal of Symbolic Computation*, 3:133–151, 1987.
55. Jieh Hsiang and Michaël Rusinowitch. On word problems in equational theories. In T. Ottman, editor, *Proceedings of the Fourteenth International Colloquium on Automata, Languages and Programming (ICALP)*, volume 267 of *Lecture Notes in Computer Science*, pages 54–71. Springer, Heidelberg, 1987.
56. Gérard Huet. A complete proof of correctness of the Knuth–Bendix completion algorithm. *Journal of Computer and System Sciences*, 23(1):11–21, August 1981.
57. Jean-Pierre Jouannaud and Bernard Waldmann. Reductive conditional term rewriting systems. In *Proceedings of the Third IFIP Working Conference on Formal Description of Programming Concepts*, Ebberup, Denmark, 1986.
58. Stéphane Kaplan. Simplifying conditional term rewriting systems: Unification, termination, and confluence. *Journal of Symbolic Computation*, 4(3):295–334, 1987.
59. Stéphane Kaplan and Jean-Luc Rémy. Completion algorithms for conditional rewriting systems. In Hassam Aït-Kaci and Maurice Nivat, editors, *Resolution of Equations in Algebraic Structures*, volume II: Rewriting Techniques, pages 141–170. Academic Press, 1989.
60. Donald E. Knuth and Peter B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, Oxford, 1970.
61. Emmanuel Kounalis and Michaël Rusinowitch. On word problems in Horn theories. *Journal of Symbolic Computation*, 11(1 & 2):113–128, 1991.
62. Dallas S. Lankford. Canonical inference. Memo ATP-32, Automatic Theorem Proving Project, University of Texas, Austin, December 1975.
63. John Wylie Lloyd. *Foundations of Logic Programming*. Symbolic Computation Series. Springer, Heidelberg, Second, extended edition, 1987.
64. Christopher Lynch and Barbara Morawska. Automatic decidability. In Gordon Plotkin, editor, *Proceedings of the Seventeenth IEEE Symposium in Logic in Computer Science*. IEEE Computer Society Press, 2002.
65. John D. McCharen, Ross A. Overbeek, and Larry Wos. Complexity and related enhancements for automated theorem proving programs. *Computers and Mathematics with Applications*, 2(1):1–16, 1976.
66. John Charles C. McKinsey. The decision problem for some classes of sentences without quantifiers. *Journal of Symbolic Logic*, 8:61–76, 1943.
67. David A. Plaisted and Andrea Sattler-Klein. Proof lengths for equational completion. *Information and Computation*, 125(2):154–170, 1996.

68. Uday S. Reddy. Narrowing as the operational semantics of functional languages. In *Proceedings of the Symposium on Logic Programming*, pages 138–151. IEEE Computer Society Press, 1985.
69. Olivier Roussel and Philippe Mathieu. Exact knowledge compilation in predicate calculus: the partial achievement case. In William W. McCune, editor, *Proceedings of the Fourteenth Conference on Automated Deduction (CADE)*, volume 1249 of *Lecture Notes in Artificial Intelligence*, pages 161–175. Springer, Heidelberg, 1997.
70. Neil J. A. Sloane. *The On-Line Encyclopedia of Integer Sequences*. 2006. `http://www.research.att.com/~njas/sequences`.
71. Mark E. Stickel. Automated deduction by theory resolution. *Journal of Automated Reasoning*, 1:333–355, 1985.
72. Larry Wos, Daniel F. Carson, and George A. Robinson. Efficiency and completeness of the set of support strategy in theorem proving. *Journal of the Association for Computing Machinery*, 12:536–541, 1965.
73. Hantao Zhang. A new method for the boolean ring based theorem proving. *Journal of Symbolic Computation*, 17(2):189–211, 1994.
74. Hantao Zhang. Contextual rewriting in automated reasoning. *Fundamenta Informaticae*, 24:107–123, 1995.