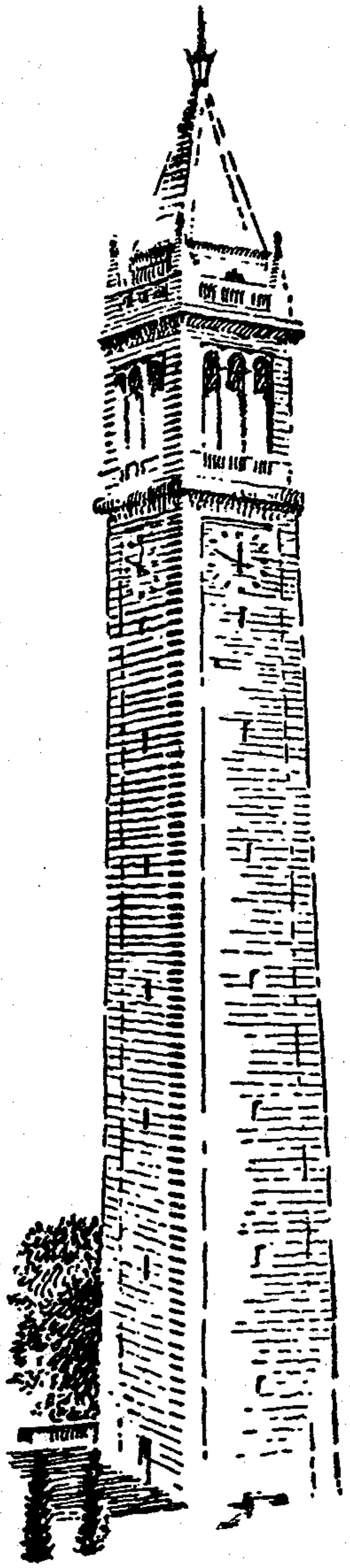

GENERATION OF A MINIMAL STG FROM AN IMPLICIT COVER

by

Luca Carloni, Tiziano Villa, Timothy Kam, Robert Brayton, and Alberto Sangiovanni-Vincentelli

Memorandum No. UCB/ERL M96/40

25 June 1996

GENERATION OF A MINIMAL STG FROM AN IMPLICIT COVER

by

Luca Carloni, Tiziano Villa, Timothy Kam, Robert K. Brayton
and Alberto Sangiovanni-Vincentelli

Memorandum No. UCB/ERL M96/40

25 June 1996

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Generation of a Minimal STG from an Implicit Cover

Luca Carloni Tiziano Villa Timothy Kam Robert Brayton
Alberto Sangiovanni-Vincentelli

Department of EECS, University of California at Berkeley,
Berkeley, CA 94720

June 21, 1996

Abstract

This report describes how to produce a state transition table (or state transition graph) of an FSM reduced by using implicit minimization algorithms. The cases of ISFSM's (minimized implicitly by the program ISM) and PNDFSM's (minimized implicitly by the program PND_REDUCE) are considered. A technique to reduce the size of the FSM table description is presented.

1 Introduction

Programs to perform implicit state minimization of incompletely specified FSM's (ISFSM's) and pseudo-nondeterministic FSM's (PNDFSM's) have been reported, respectively, in [2] and in [3]. These programs read a tabular description of the table and build as an internal representation the reduced ordered binary decision diagrams (ROBDD's or simply BDD's) of the characteristic functions of the next state and output relations representing an FSM.

The computations reported in [2, 3] go as far as computing a minimum closed cover of compatibles, but the transition relation of a reduced machine is not defined. Moreover, a conversion from the relational domain back into a compact tabular representation (or state transition graph, STG) of the reduced FSM is missing. The conversion is necessary to inspect the solution and to prepare an input for subsequent optimization steps.

In this report we present a solution to these unaddressed problems and show experimental data of a benchmark set of reduced PNDFSM's.

2 Implicit Representation of FSM's

We will use the unified implicit framework proposed in [2] [1]. Implicit techniques are based on the idea of operating on discrete sets by their characteristic functions represented by binary decision diagrams (BDD's) [1]. For example, the state transition relation of an FSM is represented by the BDD of the characteristic function of its transition relation. We may have a unique transition relation $\mathcal{T}(i,p,n,o)$, or both a next state relation $\mathcal{T}(i,p,n)$ and an output relation $\mathcal{O}(i,p,o)$.

To perform state minimization, one needs to represent and manipulate efficiently sets of sets of states. With n states, each subset of states is represented in **positional-set** form, using a set of n Boolean variables,

[1] $\exists x(\mathcal{F})$ ($\forall x(\mathcal{F})$) denotes the existential (universal) quantification of function $\mathcal{F}$ over variables x; $\Rightarrow$ denotes Boolean implication; $\Leftrightarrow$ denotes XNOR; $\neg$ denotes NOT.

$x = x_1x_2\ldots x_n$. The presence of a state s_k in the set is denoted by the fact that variable x_k takes the value 1 in the positional-set, whereas x_k takes the value 0 if state s_k is not a member of the set. For example, if $n = 6$, the set with a single state s_4 is represented by 000100 while the set of states $s_2s_3s_5$ is represented by 011010.

A set of sets of states is represented as a set S of positional-sets by a BDD characteristic function $\chi_S : B^n \to B$ as: $\chi_S(x) = 1$ if and only if the set of states represented by the positional-set x is in the set S. A BDD representing $\chi_S(x)$ will contain minterms, each corresponding to a state set in S. As an example, $Tuple_{n,k}(x)$ denotes all positional-sets x with exactly k states in them (i.e. $|x| = k$). For instance, the set of singleton states is $Tuple_{n,1}(x)$, the set of state pairs is $Tuple_{n,2}(x)$, the set of full states is $Tuple_{n,n}(x)$, and the set of empty states is $Tuple_{n,0}(x)$. An alternative notation for $Tuple_{n,k}(x)$ is $Tuple_k(x)$.

Lemma 2.1 *Set* **equality, containment** *and* **strict-containment** *between two positional-sets x and y are expressed by:* $(x = y) = \prod_{k=1}^{n}(x_k \Leftrightarrow y_k)$; $(x \supseteq y) = \prod_{k=1}^{n}(y_k \Rightarrow x_k)$; *and* $(x \supset y) = (x \supseteq y) \cdot (x \neq y)$.

Lemma 2.2 *Given two sets of positional-sets,* **complementation, union, intersection,** *and* **sharp** *can be performed on them as logical operations* $(\neg, +, \cdot, \cdot\neg)$ *on their characteristic functions.*

Lemma 2.3 *Given a characteristic function $\chi_A(x)$ representing a set A of positional-sets,* **set union** *defines a positional-set y which represents the union of all state sets in A, and can be computed by:*

$$Union_{x\to y}(\chi_A) = \prod_{k=1}^{n}(y_k \Leftrightarrow \exists x\ [\chi_A(x) \cdot x_k])$$

3 Transition Relation of Reduced FSM

In Section 3.1 we discuss how to determine the transition relation of a reduced FSM starting from the implicit representation of a closed cover of compatibles as found in the program ISM [2], which minimizes the number of states of an ISFSM. In Section 3.2 we do the same for the program PND_REDUCE [2] [3], which minimizes the number of states of a PNDFSM.

The logic variables $c, c', c'', \tilde{c}', d, p, n$ denote sets of state variables, while i is a set of input variables and $o, \tilde{o}$ are sets of output variables. Let $S(c)$ be the the relation of the compatibles chosen in a minimum solution and $r(p)$ be the relation of the reset state(s).

3.1 Reduction of ISFSM's

The following equations compute the transition relation $T_{red}(i, c, c', o)$ and the reset state $r_{red}(c)$ of a reduced FSM, starting from a minimum closed cover of compatibles $S(c)$ computed by ISM for an ISFSM represented by the next state and output relations $T(i, p, n)$, $O(i, p, o)$ with initial state $r(p)$.

$$\begin{aligned}
T^{red}(i,c,d) &= Union_{n\to d}\{\exists p\ [S(c)(c \supseteq p) \cdot T(i,p,n)]\} && (1)\\
T^{red}(i,c,c') &= \exists d\{T^{red}(i,c,d)S(c')(c' \supseteq d) - \exists c''\ [S(c'')(c'' \prec c')(c'' \supseteq d)]\} && (2)\\
T^{red}(i,c,c',o) &= T^{red}(i,c,c')\ \forall p\ [Tuple_1(p)(c \supseteq p) \Rightarrow O(i,p,o)] && (3)\\
r_a^{red}(c) &= \exists p\ [(r(p)S(c)(c \supseteq p)] && (4)\\
r^{red}(c) &= r_a^{red}(c) - \exists c'[c \to c']r_a^{red}(c)(c' \prec c) && (5)
\end{aligned}$$

[2]Referred to as ISM2 in [3].

This second set of equations is equivalent to the previous one, except that it works with a unique transition relation $\mathcal{T}(i,p,n,o)$, instead of the next state and output relations $\mathcal{T}(i,p,n)$ and $\mathcal{O}(i,p,o)$.

$$\begin{aligned}
T^{red}(i,c,d,o) &= Union_{n \to d}\{\exists p\,[S(c)(c \supseteq p) \cdot \mathcal{T}(i,p,n,o)]\} && (6)\\
T^{red}(i,c,c',o) &= \exists d\{T^{red}(i,c,d,o)S(c')(c' \supseteq d) - \exists c''\,[S(c'')(c'' \prec c')(c'' \supseteq d)]\} && (7)\\
r_a^{red}(c) &= \exists p\,[(r(p)S(c)(c \supseteq p)] && (8)\\
r^{red}(c) &= r_a^{red}(c) - \exists c'[c \to c']r_a^{red}(c)(c' \prec c) && (9)
\end{aligned}$$

3.2 Reduction of PNDFSM's

The following equations compute the transition relation $T_{red}(i,c,c',o)$ and the reset state $r_{red}(c)$ of a reduced FSM, starting from a minimum closed cover of compatibles $S(c)$ computed by PND_REDUCE for a PNDFSM represented by the transition relation $\mathcal{T}(i,p,n,o)$ with initial state $r(p)$.

$$\begin{aligned}
T^{red}(i,c,n,o) &= \exists p\,[S(c)(c \supseteq p) \cdot \mathcal{T}(i,p,n,o)]\} && (10)\\
T^{red}(i,c,d,o) &= Union_{n \to d}\{T^{red}(i,c,n,o) - \exists p\,[Tuple_1(p)(c \supseteq p) \cdot \overline{(\exists n\, \mathcal{T}(i,p,n,o))}\,]\} && (11)\\
T^{red}(i,c,c',o) &= \exists d\{T^{red}(i,c,d,o)S(c')(c' \supseteq d) - \exists c''\,[S(c'')(c'' \prec c')(c'' \supseteq d)]\} && (12)\\
T^{red}(i,c,c',o) &= T^{red}(i,c,c',o) - \exists \tilde{c'}\tilde{o}[T^{red}(i,c,\tilde{c'},\tilde{o})(\tilde{c'} \neq c')(\tilde{c'}\tilde{o} \prec c'o)] && (13)\\
r_a^{red}(c) &= \exists p\,[(r(p)S(c)(c \supseteq p)] && (14)\\
r^{red}(c) &= r_a^{red}(c) - \exists c'[c \to c']r_a^{red}(c)(c' \prec c) && (15)
\end{aligned}$$

The third equation is necessary to enforce that in the reduced FSM, given an input and present state, there is a unique specified next state. Indeed if transitions $i\ c\ d'\ o$ and $i\ c\ \tilde{d'}\ \tilde{o}$ are in relation $T^{red}(i,c,d,o)$, then transitions $i\ c\ c'\ o$ and $i\ c\ \tilde{c'}\ \tilde{o}$ would be in transition $T^{red}(i,c,c',o)$ (the second equation only makes sure that for a given i, c and o there is at most one c'). The third equation chooses one of the two transitions. The term $(\tilde{c'} \neq c')$ ensures that we make unique only the next state and not the output; therefore we may obtain an ISFSM (when for the same input, present state and next state there are all the outputs). If we omit $(\tilde{c'} \neq c')$ we obtain a DFSM.

4 Conversion from Implicit Relation to Compact Table

Once the transition relation of the reduced FSM has been obtained, it is important to convert it into a compact tabular representation, in order to inspect the solution and generate a file in *kiss* format, which is an input to subsequent optimization steps.

The obvious way to perform the conversion is to enumerate the minterms of the BDD of the transition relation and create a tabular line for each of them. For instance, from the following relation $T^{red}(i,c,c',o)$ (with variable order $o, i, i, c, c', c, c', c, c', c, c', c, c', c, c'$):

```
001010010011001
001100100100110
010100001100110
011010010011001
011011000011001
011110000110011
101001001001100
101010010011001
```

110010010011001
110011000011001
111010010011001
111110000110011

the direct method produces the following tabular representation:

01	c_{54}	c_{126}	0
01	c_{126}	c_{34}	0
10	c_{125}	c_{54}	0
11	c_{54}	c_{126}	0
11	c_{34}	c_{126}	0
11	c_{126}	c_{126}	0
01	c_{34}	c_{54}	1
01	c_{54}	c_{126}	1
10	c_{54}	c_{126}	1
10	c_{34}	c_{126}	1
11	c_{54}	c_{126}	1
11	c_{126}	c_{126}	1

The table so obtained is not most compact one (fewest number of symbolic cubes), as the following observations show:

1. The two symbolic cubes
   01 c_{54} c_{126} 0
   11 c_{54} c_{126} 0
   could be merged into one
   -1 c_{54} c_{126} 0.

2. The two symbolic cubes
   11 c_{54} c_{126} 0
   11 c_{54} c_{126} 1
   could be merged into one
   11 c_{54} c_{126} $-$.

These two examples show that enumerating the minterms of the BDD representation, where the variables have the order i, o, c, c' does not yield symbolic cubes that are maximally expanded. Our objective is to find a two-level representation with a minimum number of symbolic cubes; a secondary objective is to maximize the incomplete specification of the tabular representation.

Notice that part of the problem has to do with the chosen variable ordering. For instance, in case 1., if the inputs would be ordered as the last set of variables, the BDD representation would yield the cube -1 c_{54} c_{126} 0. In the same fashion, in case 2., if the outputs would be ordered as the last set of variables, the BDD representation would yield the cube 11 c_{54} c_{126} $-$. We propose an approximate solution to the problem of producing a compact tabular representation, based on the following facts.

Consider the following transitions with the same present state and next state c_{54} c_{126}:

01	c_{54}	c_{126}	0
11	c_{54}	c_{126}	0
01	c_{54}	c_{126}	1
10	c_{54}	c_{126}	1

11 c_{54} c_{126} 1
and restrict them to the input and output fields:
01 0
11 0
01 1
10 1
11 1.
The relation of the inputs and outputs, given that outputs precede inputs in the chosen BDD ordering, has the following three cubes:
-1 0
$1-1$
01 1.
A minimization with *espresso* (*.type fd* by default) returns the cover:
-1 1
$1-$ 1.
The best solution (i.e. having the minimum number of cubes with the maximum of incomplete specification) is:
$-1-$
$1-$ 1.
As another example consider the transitions:
11 c_{126} c_{126} 0
11 c_{126} c_{126} 1
The related input-output relation has one cube, as desired:
11 $-$.

A procedure that builds the input-relation for each available pair (present state, next state) to exploit the simplifications allowed by the enumeration of BDD cubes is shown in Fig. 1.

```
$T^{red}\_to\_kiss$ {
    $T^{red}(c,c') = \exists i\, o\, T^{red}(i,c,c',o)$
    for each minterm $M(c,c') \in T^{red}(c,c')$ {
        $T^{red}_{c,c'}(i,o) = T^{red}(i,c,c',o) M(c,c')$
        for each cube $i,o \in T^{red}_{c,c'}(i,o)$ {
            create symbolic cube $i,c,c',o$
        }
    }
}
```

Figure 1: Pseudo-code to produce a tabular representation.

The following figures shows pairs of initial PNDFSM's and reduced machines obtained by the computations described in the last two sections.

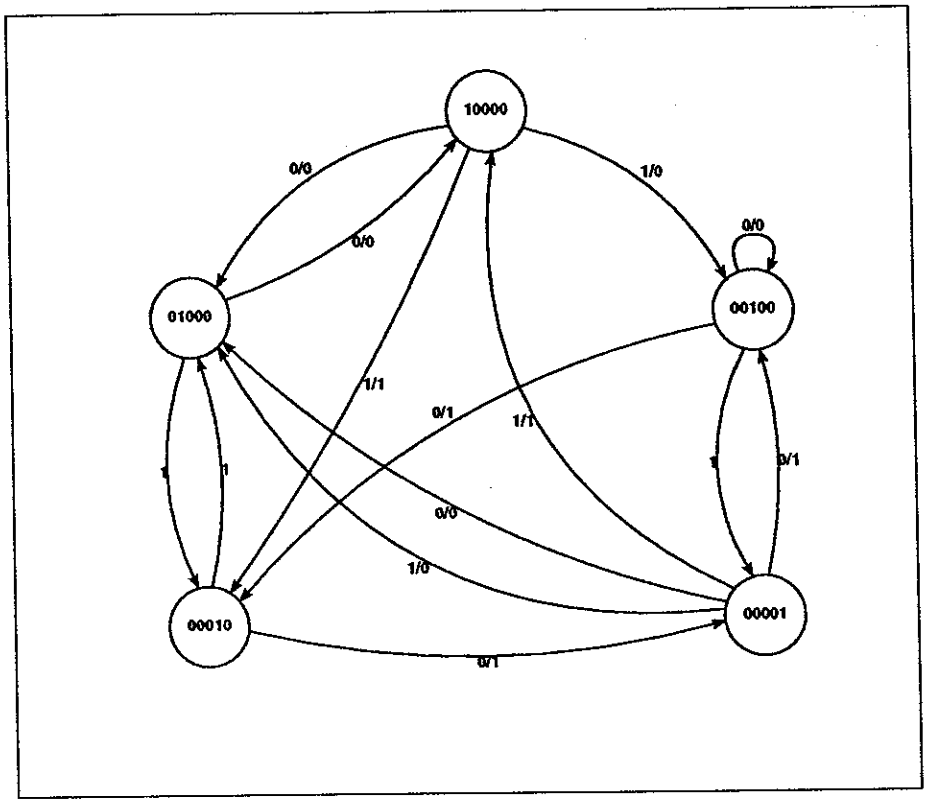

Figure 2: The STG of the PNDFSM *damiani.pnd.*

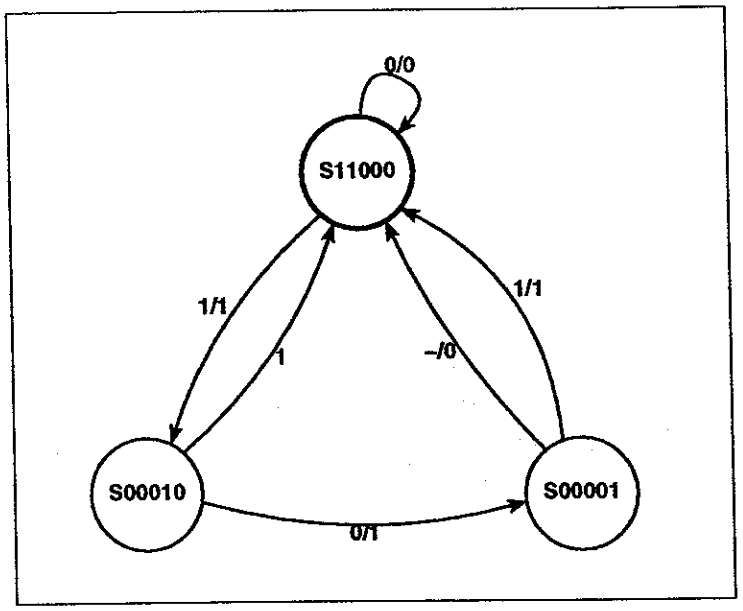

Figure 3: The STG of the reduced FSM obtained with PND_REDUCE from the PNDFSM *damiani.pnd.*

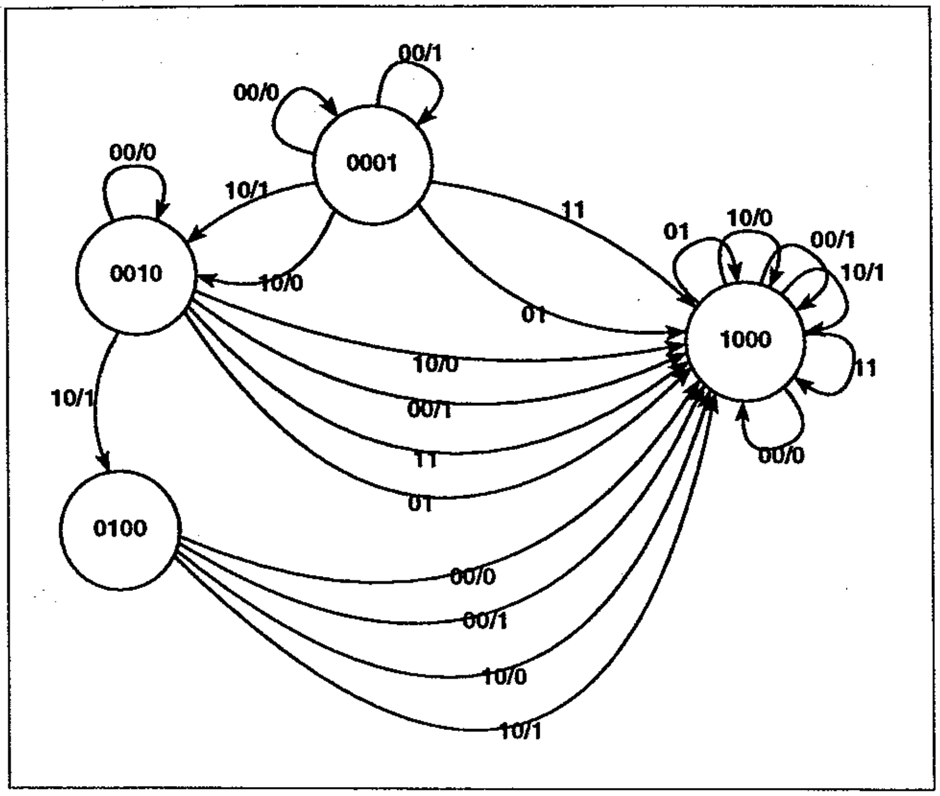

Figure 4: The STG for the PNDFSM *mc9.pnd*.

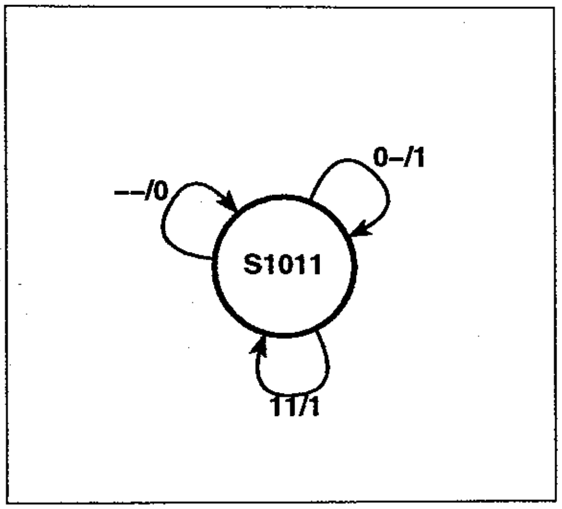

Figure 5: The STG of the reduced FSM obtained with PND_REDUCE from the PNDFSM *mc9.pnd*.

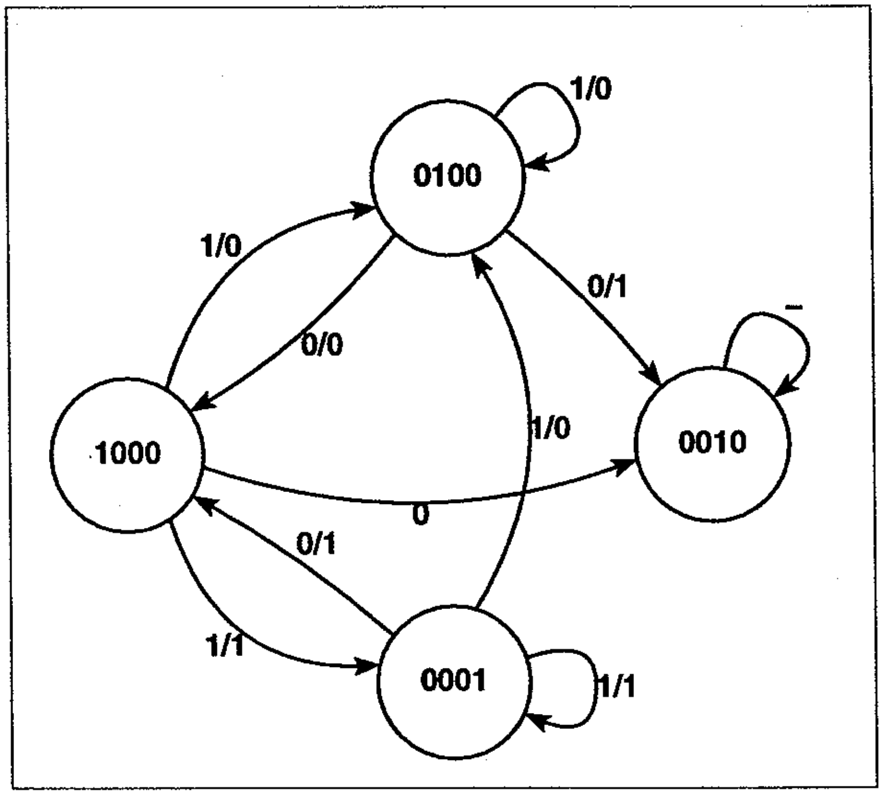

Figure 6: The STG for the PNDFSM *yoshi.pnd*.

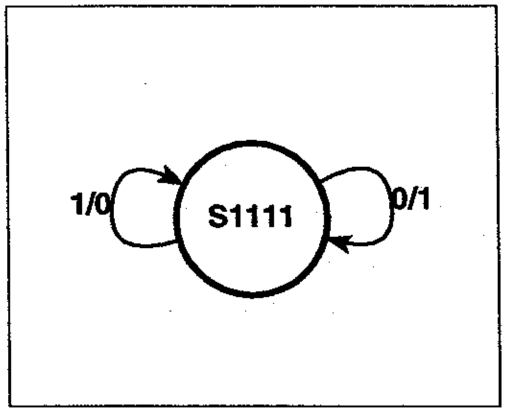

Figure 7: The STG of the reduced FSM obtained with PND_REDUCE from the PNDFSM *yoshi.pnd*.

5 Results

We have augmented the programs ISM and PND_REDUCE (*aliter* ISM2 in [3]) with procedures to produce the relation of the reduced FSM and generate a tabular representation, according to the computations described in this report.

We report the experiments described in [3] for PNDFSM state minimization. Of each example we list the initial and final number of states, and the initial and final number of transitions (symbolic cubes). The last parameter was not known from [3] and it is the size of the tabular representation of the reduced FSM produced by our procedure. For each example a file in *kiss* format is obtained. Note that all reduced FSM's happen to be of Mealy type.

		original PNDFSM		*reduced FSM*			
example	I/O	states	trans.	states	trans.	FSMtype	CPU time
L3	2/3	17	204	2	5	Mealy	6.24
am9	6/6	13	489	1	94	Mealy	1.49
ax4	5/6	11	340	1	32	Mealy	0.41
ax7	3/5	20	422	2	111	Mealy	3.94
bx7	3/5	23	418	2	43	Mealy	4.62
damiani	1/1	5	14	3	6	Mealy	0.55
e4at2	5/4	14	324	1	17	Mealy	0.66
e4bp1	5/5	11	317	1	12	Mealy	0.55
e4t1	7/4	6	125	1	8	Mealy	0.14
e69	2/1	8	43	1	2	Mealy	0.18
e6tm	4/4	21	993	1	58	Mealy	1.9
ex10	3/4	13	239	1	17	Mealy	0.36
ex12	3/4	13	116	1	5	Mealy	0.47
mc9	2/1	4	22	1	3	Mealy	0.04
mt51	5/6	16	490	1	106	Mealy	2.25
mt52	5/6	9	464	1	180	Mealy	0.76
pm03	2/4	15	87	1	4	Mealy	0.44
pm04	2/4	79	513	1	4	Mealy	48.46
pm11	8/8	9	586	1	88	Mealy	2.47
pm12	8/8	7	231	1	40	Mealy	0.29
pm31	6/6	22	1109	1	274	Mealy	4.59
pm33	6/6	21	1220	1	132	Mealy	4.2
s3p1	5/5	38	2857	1	34	Mealy	335.82
s3t2	5/4	36	1663	1	52	Mealy	15.95
tm01	4/4	10	603	1	39	Mealy	0.54
tm02	4/4	7	177	1	4	Mealy	0.24
tm31	3/4	9	95	1	3	Mealy	0.16
tm32	3/4	9	132	2	21	Mealy	1.52

Table 1: **Results** on problems from PNDFSM benchmark suite

References

[1] R. Bryant. Graph based algorithm for Boolean function manipulation. In *IEEE Transactions on Computers*, pages C–35(8):667–691, 1986.

[2] T. Kam, T. Villa, R. Brayton, and A. Sangiovanni-Vincentelli. A fully implicit algorithm for exact state minimization. In *The Proceedings of the Design Automation Conference*, pages 684–690, June 1994.

[3] T. Kam, T. Villa, R. Brayton, and A. Sangiovanni-Vincentelli. Implicit state minimization of non-deterministic FSM's. In *The Proceedings of the International Conference on Computer Design*, pages 250–257, October 1995.