ISCAS-LCS-96-12 Dec., 1996

中国科学院软件研究所
计算机科学实验室报告

On the Relational Translation Method for Propositional Modal Logics

by

Jian Zhang

Laboratory for Computer Science
Institute of Software
Chinese Academy of Sciences
Beijing 100080. China

On the Relational Translation Method for Propositional Modal Logics*

Jian Zhang
Laboratory for Computer Science
Institute of Software
Chinese Academy of Sciences
P.O.Box 8718, Beijing 100080
P.R. China
Email: zj@ox1.ios.ac.cn

December 1996

Abstract

One way of proving theorems in modal logics is translating them into the predicate calculus and then using conventional resolution-style theorem provers. This approach has been regarded as inappropriate in practice, because the resulting formulas are too lengthy and it is impossible to show the non-theoremhood of modal formulas. In this paper, we demonstrate the practical feasibility of the (relational) translation method. Using a state-of-the-art theorem prover for first-order predicate logic, we proved many benchmark theorems available from the modal logic literature. We show the invalidity of propositional modal and temporal logic formulas, using model generators or satisfiability testers for the classical logic. Many satisfiable formulas are found to have very small models. Finally, several different approaches are compared.

*Supported in part by the National High Technology Program under grant No. 863-306-05-03-3.

1 Introduction

Modal logics have been studied by many researchers in computer science and artificial intelligence. There are roughly two classes of methods for automated reasoning in these logics. With the *translational* approach, we transform modal formulas to first-order predicate logic (FOPL) formulas and then use existing tools for the classical logic. Alternatively, we can design tools specifically for modal logics. Such methods are classified as the *direct* approach. They include, among others, semantic tableaux, modal resolution, and matrix-based procedures. See, for example, [1, 5, 22].

The translational approach can be further divided into syntactic and semantic methods [16]. To use the *syntactic* method, we introduce a special unary predicate P which means its argument is provable, and translate the axioms and rules into classical logic. This method can only be applied to propositional modal logics. On the other hand, the *semantic* methods are based on Kripke semantics. Such a method is applicable to first-order modal logics, but properties of the accessibility relation should be defined by a finite set of sentences in predicate logic. This can be done in a number of ways. In the standard way [16], we introduce a special binary predicate R to describe the accessibility relation between the possible worlds. This is now called *relational* translation. In addition, there are so-called *functional* translation and *semi-functional* translation [18, 17].

The relational translation approach is very simple and quite general. It can deal with many modal logics, and it benefits from the power of existing tools for the classical logic. But there are also some problems with it. To quote Goré [8] (page 2), "by translating into first order logic, the translational methods immediately surrender the decidability of the propositional modal logic they translate." Another problem is that, the result of translation is usually a complicated formula, which makes its proof difficult to find. Ohlbach and Weidenbach [19] gave the following example:

> Prove that $(\Diamond\Box p \leftrightarrow \Diamond\Box\Diamond\Box p)$ is valid in **KD45.**

Using the standard translation method and the popular resolution-style prover OTTER 3.0 [13], they failed in finding a proof.

For these reasons, the (relational) translation method has been neglected by many researchers. Is it really that bad? Our experiments show that it is quite competitive. In this paper, we shall describe some of the results. We suggest using both theorem provers and counterexample finders to decide the

validity of modal formulas. In addition, we study the problem of satisfying propositional temporal logic formulas.

2 Translating Normal Modal Logics

2.1 Propositional Normal Modal Logics

We first review some relevant definitions and facts. For more details, see [2, 10]. There are many modal logic systems. But in this paper, we shall consider only normal modal logics.

The smallest normal system is called **K**. It can be extended by defining additional axioms. The most well-known axiom schemas include *D, T, B, 4, 5*. There are only 15 distinct normal systems (including **K** itself) produced by taking the schemas (as axioms) in all combinations. They are named **K**, **KD**, **KT4**, and so on. Among them, **KT4** and **KT5** are also known as **S4** and **S5**, respectively.

The semantics of modal logics is often defined in terms of the possible worlds structures. Many propositional modal systems have the *finite model property*, which means, every non-theorem is false in some finite model. This property implies decidability, if the logic is finite axiomatizable. It is well known that each of the above 15 modal logics has the finite model property. Moreover, in some systems (e.g. **S5**), any satisfiable formula can be satisfied in a small finite model (whose size does not exceed the length of the formula).

2.2 The Relational Translation Method

Roughly speaking, the modal operators $\Box$ and $\Diamond$ correspond to universal and existential quantifiers in FOPL, respectively. There is a standard way of translating modal formulas into FOPL formulas. To do so, we add a 'world' argument to each proposition, and introduce a new binary predicate R to describe the accessibility relation between worlds. If the current world is w, then the formula φ will be translated to $Tr(\varphi, w)$, which means φ is true at w. The basic translation rules are defined inductively as follows:

$$\begin{aligned}
Tr(p, w) &\equiv p(w) \\
Tr(\neg\psi, w) &\equiv \neg Tr(\psi, w) \\
Tr(\psi_1 \wedge \psi_2, w) &\equiv Tr(\psi_1, w) \wedge Tr(\psi_2, w) \\
Tr(\Box\psi, w) &\equiv \forall v\,(R(w, v) \rightarrow Tr(\psi, v)) \\
Tr(\Diamond\psi, w) &\equiv \exists v\,(R(w, v) \ \wedge \ Tr(\psi, v))
\end{aligned}$$

where v is a new 'world' variable. The proposition p in the original formula becomes a unary predicate. For example, $Tr(\Diamond\Box p, o)$ is the formula

$$\exists w\,(R(o,w) \wedge \forall v\,(R(w,v) \rightarrow p(v)))$$

The following theorem [16, 15] forms the basis of the relational translation method.

Theorem. A formula φ is valid in a modal logic system S if and only if $Ax(S) \rightarrow \forall w\, Tr(\varphi, w)$ is valid in first-order predicate logic, where $Ax(S)$ is a set of axioms describing properties of the accessibility relation.

For example, $(\Box p \rightarrow \Diamond p)$ is valid in **KT**, because the sentence

$$\begin{array}{l}\forall x\, R(x,x) \;\rightarrow \\ \forall w\,(\forall v\,(R(w,v) \rightarrow p(v)) \;\rightarrow\; \exists v\,(R(w,v) \wedge p(v)))\end{array}$$

is a theorem in the predicate calculus.

2.3 A Concurrent Program as the Decision Procedure

The concurrent program in Fig. 1 consists of a theorem proving process (P_1) and a model finding process (P_2). The theorem proving process can be based on any refutationally complete proof procedure. To find a model of some fixed size, one can either use a decision procedure for the classical propositional calculus (such as the Davis-Putnam algorithm) or use a finite model search program for the first-order predicate logic. The input parameter φ is a propositional modal logic formula and S is a normal modal logic system.

We assert that the concurrent program can serve as a decision procedure for many propositional modal logics like **K**, **KT**, **KD**, **S4**, ... Such a modal system should possess the finite model property, and the accessibility relation should be characterized by a finite set of sentences in first-order predicate logic. The program terminates under the fairness assumption. (It is unfair if one process, e.g., the theorem prover, never has a chance to be executed.)

3 Experimental Results

For non-classical logics, there are not so many automated reasoning tools and test problems. Here we describe some experimental results on the benchmarks used by other authors [1, 3, 9, 6]. We shall not elaborate on the

```
program CDP(φ,S)
cobegin
    φ := Ax(S) → ∀w Tr(φ,w);
    P1::
      repeat
        apply a suitable set of inference rules to φ;
      until (contradiction is deduced);
      kill(P2);
    ||
    P2::
      var n = 0;
      repeat
        n := n+1;
        find an n-element model of φ;
      until (a model is found);
      kill(P1);
coend
```

Figure 1: The decision procedure

programs' performances, because they are affected by several factors such as the data structures and the programming languages. Moreover, not all timing information are available in the related papers.

It is very easy to implement a tool for translating modal logic formulas. (The translation time will be neglected.) To show the satisfiability of the formulas, we may use various tools, such as FINDER, MGTP, LDPP, SATO, MACE and SEM [20, 21, 23, 14, 25]. In the following, we only report the size of the smallest model satisfying each formula. In many cases, the models are very small, and can be easily found by any of the tools.

To prove modal theorems, we use the resolution-style theorem prover OTTER 3.0.4 [13], running on a SPARCstation 20 with 32 MB memory. We did not take much advantage of OTTER's special features to achieve high performances. It was instructed to run in autonomous mode. However, since the prover is not so good at proving "if-and-only-if" theorems [12], we broke each theorem of the form $A \leftrightarrow B$ into two cases $A \rightarrow B$ and $B \rightarrow A$. In such a way, Ohlbach and Weidenbach's example mentioned earlier can be proved within 2 seconds.

Problem Set 1[1]

In [1] Catach describes a tableaux-based program called TABLEAUX, and gives the validity status of 31 formulas in 16 modal logic systems. The formulas are very simple, and TABLEAUX completed all 496 tests in less than 1 minute. For simplicity, we consider only the logics **K**, **KD45**, **S4** and **S5**, which are very important in knowledge representation and reasoning.

There are totally $31 \times 4 = 124$ tests to be performed. Among these, 67 cases are validity proofs. With only a few exceptions, OTTER finds each proof within 1 second. In the remaining 57 cases, we found the smallest structures which show the invalidity of the formulas. Of these 57 counter-models, only 8 are of size 3. The others are of size 1 or 2.

For those formulas containing the equivalence connective, if we give them directly to the translator and then to OTTER, it will be much more difficult to find the proofs. For example, the last formula $\Box\Box p \leftrightarrow \Diamond\Box p$ is valid in **KD45**. It takes OTTER about 2 minutes to find a proof. But each of the formulas $\Box\Box p \rightarrow \Diamond\Box p$ and $\Diamond\Box p \rightarrow \Box\Box p$ can be proved to be valid in less than 0.25 second.

[1] Some early results on this set of problems were described in [24].

Problem Set 2

Demri [3] analyses several different strategies in a tableau-based **S4** prover, and compares them on a set of 9 valid formulas. Here we list 4 of them:

4. $\Box_c(\neg PC \rightarrow \Box_b \neg PC)$
   $\wedge\ \Box_c\Box_b\Box_a(PC \vee PB \vee PA)$
   $\wedge\ \Box_c\Box_b(\neg PB \rightarrow \Box_a \neg PB)$
   $\wedge\ \Box_c\Box_b(\neg PC \rightarrow \Box_a \neg PC)$
   $\wedge\ \Box_c \neg \Box_b PB$
   $\wedge\ \Box_c\Box_b \neg \Box_a PA$
   $\rightarrow \Box_c PC$
5. $\Diamond\Box(\Box(p \vee \Box q) \leftrightarrow (\Box p \vee \Box q))$
6. $\Diamond\Box((p \rightarrow q) \leftrightarrow F(q, F(p, q)))$
   where $F(A, B)$ stands for $(\neg A \vee \neg\Diamond(A \wedge B) \vee (B \wedge \Diamond(A \wedge \neg B))$
9. $\Box(\Box(\Box p \rightarrow \Box(\Box q \rightarrow \Box r)) \rightarrow \Box(\Box(\Box p \rightarrow \Box q) \rightarrow \Box r))$

The first one is a multimodal formula, encoding McCarthy's 3 Wise Men puzzle. (C is the wisest man.) There are typos in formulas (6) and (9). The correct versions are as follows [4]:

6a. $\Diamond\Box((p \rightarrow q) \leftrightarrow F(p, F(p, q)))$
9a. $\Box(\Box(\Box p \rightarrow \Box(\Box q \rightarrow \Box r)) \rightarrow \Box(\Box(\Box p \wedge \Box q) \rightarrow \Box r))$

Formulas (5) and (6a) are obtained from **S5**-valid formulas. A formula ϕ is satisfiable in **S5** iff $\Diamond\Box\phi$ is satisfiable in **S4**. Instead of proving the validity of $\Diamond\Box\phi$ in **S4**, we prove directly ϕ is valid in **S5**. As previously, we divide each "if-and-only-if" theorem into two cases. In this way, 8 of the 9 theorems are easily proved, each requiring less than 2 seconds. The exception is formula (6a) whose proof is found in about 1 minute.

It is interesting to note that some non-theorems are falsified in very small models. For example, the negation of (6) and (9) are satisfied in 1-world models. And, if we substitute $\Box_b PB$ for $\Box_c PC$ in formula (4), then there is also a countermodel of size 1, which can be easily found by exhaustive search.

Problem Set 3

Recently Goré, Heinle and Heuerding [9] studied the relations between some propositional normal modal logics, using the Logics Work Bench (LWB). As

a side-effect of this work, they collected a database of theorems which can be used to test modal theorem provers.

LWB has automated proof procedures for only a few modal logics, namely, **K**, **KT**, **KT4**, **KT45** and **KW**. To deal with extensions of these logics, one can add some modalised instances of the new axioms to the premises. The user has to provide the appropriate instances. For more details about this technique, see [9].

In the appendix of [9], 72 **K**-theorems are listed. Most of them are easy for OTTER to prove, but there are also some difficult ones. The results are summarized in the following table.

OTTER time	number of theorems
< 10 sec.	51
10–100 sec.	4
100–1000 sec.	5
> 1000 sec.	12

So about 3 quarters of the theorems can be proved within 2 minutes. The 12 difficult formulas are very complicated. Some of them can be easily proved if we work in a different modal system rather than in **K**. For example, using our translation tool and OTTER, we can prove that $M \rightarrow Pt$ is valid in **K4** within 10 seconds. Here M and Pt stand for the following two formulas:

$$M \equiv \Box\Diamond p \rightarrow \Diamond\Box p \qquad Pt \equiv \Box(p \vee \Diamond p) \rightarrow \Diamond(p \wedge \Box p)$$

Instead of proving the theorem directly, Goré, Heinle and Heuerding prove the following **K**-theorem:

$$\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge M \rightarrow Pt$$

where ϕ_1 is an instance of *4*, ϕ_2 and ϕ_3 are instances of $\Box$*4*. The resulting formula is quite lengthy, and OTTER has difficulty finding a proof for it. I am not indicating that the "modalised instance" technique is not useful. But the generality and power of the translation approach should be emphasized. Of course, a powerful prover for the predicate calculus is very important.

Based on the formulas given in [9], we also examined some conjectures which are not valid in **K**. They are usually falsified in very small Kripke structures. The following are some examples:

formula	size of structure
$M \to Pt$	2
$H \to L$	3
$H^+ \to L^+$	2
$L \to L^+$	2
$L^{++} \to L^+$	2
$Dum4 \to Dum$	2

The formulas H, H^+, L, L^+ and L^{++} are defined as follows:

$$\begin{aligned}
H &\equiv (\Box(p \vee q) \wedge \Box(\Box p \vee q) \wedge \Box(p \vee \Box q)) \to (\Box p \vee \Box q) \\
H^+ &\equiv (\Box(\Box p \vee q) \wedge \Box(p \vee \Box q)) \to (\Box p \vee \Box q) \\
L &\equiv \Box((p \wedge \Box p) \to q) \vee \Box((q \wedge \Box q) \to p) \\
L^+ &\equiv \Box(\Box p \to q) \vee \Box(\Box q \to p) \\
L^{++} &\equiv \Box(\Box p \to \Box q) \vee \Box(\Box q \to \Box p) \\
Dum &\equiv \Box(\Box(p \to \Box p) \to p) \to (\Diamond \Box p \to p) \\
Dum4 &\equiv \Box(\Box(p \to \Box p) \to p) \to (\Diamond \Box p \to (p \vee \Box p))
\end{aligned}$$

To show that Dum holds in **KTDum2**, Goré, Heinle and Heuerding proved the **K**-theorem:

$$\varphi_1 \wedge \varphi_2 \wedge \Box Dum2 \to Dum$$

where φ_1 and φ_2 are instances of T and $\Box T$, respectively. $Dum2$ is defined by:

$$Dum2 \equiv \Box(\Box(p \to \Box p) \to \Box p) \to (\Diamond \Box p \to p)$$

We tried to find a small reflexive structure falsifying $Dum2 \to Dum$, but failed. The formula turned out to be **KT**-valid, which can be proved in about 1 second. So Dum is implied by either $Dum2$ or $\Box Dum2$ in **KT**.

Problem Set 4

Giunchiglia and Sebastiani [6] developed a SAT-based decision procedure for **K**, called KSAT. It has been tested against a large number of randomly generated 3CNF modal formulas. We implemented a similar method for generating random formulas, and found that quite some satisfiable formulas have very small models. But the number of formulas that have been tested is not enough for us to make any conclusion.

We did not spend much time on this set of problems. There are two reasons for this. Firstly, the generated formulas usually contain repetitive or complementary literals in the same clause, e.g. $\Box(p_1 \vee p_2 \vee \neg p_1)$. This problem is quite serious when there are only a few variables (say, fewer than 5). Secondly, we are more interested in structured formulas. But when only the average timings are recorded, some hard formulas (e.g., formulas of the form $A \leftrightarrow B$) can be hided in many other easy formulas.

4 Satisfying Propositional Temporal Logic Formulas

Temporal logic was introduced into computer science by Amir Pnueli about 20 years ago. It is a convenient tool for specifying and verifying concurrent programs [11]. There are many versions of temporal logic. For example, time may be discrete or dense, a time point can have only one successor or several successors. Here we consider the linear time temporal logic (LTL), where time is modeled by the set of natural numbers. In addition to the 'box' and 'diamond' operators, LTL has such operators as 'next-time' and 'until'.

Since LTL is based on the natural numbers, induction is needed to prove some theorems. A resolution-style automatic theorem prover is not enough if we translate LTL formulas into FOPL formulas. However, propositional LTL also has the finite model property, i.e., a formula is satisfiable iff it has a model consisting of a finite number of states. And we can still use model generation tools in FOPL to find models of satisfiable propositional temporal logic formulas. But we have to check the generated model can be translated into some LTL model.

For simplicity, let us consider only the next-time operator. In this case, we need a new unary successor operator S. The binary relation R is reflexive and transitive. In addition, we have to add such axioms as $R(x, S(x))$.

Let us give a simple example to illustrate the difference between models found in this way and models obtained by the tableau method. Consider the formula $\bigcirc\bigcirc\Box p$, where $\bigcirc$ denotes next-time. With the tableau method, we get a 3-state model: $\{s_0, s_1, s_2\}$, where $S(s_0) = s_1, S(s_1) = s_2, S(s_2) = s_2$, p can take any value in the first two states, but it is true in s_2. In general, if there are k next-time operators, we shall get a $(k+1)$-state model. However, with the translation method, a 1-state model is produced: $\{s_0\}$, $S(s_0) = s_0$, p holds at s_0.

5 Related Work

As we mentioned in the introduction, there are quite some methods for the automated reasoning in modal logics. But relatively few implementations are available. The matrix method and the functional translation method require specialized unification algorithms. In contrast, it is very easy to write a relational translator.

Another advantage of the translational approach is its generality. It can be applied to many modal logic systems. Changing from one system to another, we need only modify the axioms. On the other hand, many other methods and tools are designed for a few specific systems. For example, semi-functional translation [17] and the technique described in [19] are most suitable for *serial* modal logics like **KD** and **KD45**. Modal resolution rules are discussed in [5], within 5 systems, i.e., **K, Q, T, S4** and **S5**. The program LWB has built-in proof procedures for 5 selected logics, and KSAT [6, 7] works only in **K**. TABLEAUX can deal with many modal systems. But if one was to extend it with new systems, new procedures (encoding the semantic properties) would have to be added.

Tableau-based decision procedures can be used both to prove theorems and to show the satisfiability of formulas. As we point out in the previous section, the models found by our method are usually different from those generated by tableau procedures. An example in linear time temporal logic has been given to show the difference. For another example, the formula *4*. $\Box p \rightarrow \Box\Box p$ is not valid in **K**. Catach [1] gives a 3-world countermodel produced by the tableau-based procedure. But with our method, we shall first find the smallest model having 2 worlds:

R	w_0	w_1
w_0	$False$	$True$
w_1	$True$	$False$

p	w_0	w_1
	$False$	$True$

where w_0 is the real world.

According to our experiences, many satisfiable formulas have very small models. Thus it is practical to use model generation tools in the classical logic to show the satisfiabiliy of propositional modal formulas. However, for some modal logics (like **K**), there are some specially constructed formulas [10] which are satisfiable only in exponential-size models. They present much difficulty for methods based on the Kripke semantics. The program KSAT can deal with such formulas efficiently [7].

6 Concluding Remarks

The (relational) translation technique has been known for a long time. But few reports on its practical performances are available, and previously no one has studied the generation of countermodels with this method. Our experiments show that, the method is quite competitive. Without spending much effort, we can prove a lot of theorems in many modal logics, using existing tools in the classical logic. Moreover, by combining a theorem prover with a model generator, we can decide the satisfiability as well as the validity of propositional formulas. It is interesting that many non-theorems turn out to have very small countermodels.

Compared with other methods, relational translation has some appealing features. (See the previous section.) Theoretically, it provides decision procedures for various important logics. Practically, we expect that it can deal with a large number of formulas of reasonable sizes. It should be very useful in those applications where first-order reasoning is inevitable, and in the applications which have several modal operators belonging to different systems.

It is certainly not true that the translational approach is better than other approaches in all respects. The method has some drawbacks. But it is not so weak as thought previously. Rather than proposing a entirely new method, we try to identify and overcome some difficulties. The power of the translation approach is largely dependent on the tools available in predicate calculus. In our experiments, only OTTER (in its autonomous mode) was used to prove theorems. This may not be the best choice. One can use other types of provers (e.g., a nonclausal one) or exploit more features of OTTER to achieve better performances. One can also develop special techniques for handling the transitivity axiom or the seriality axiom (similar to that of [19]).

References

[1] L. Catach, "TABLEAUX: A general theorem prover for modal logics," *J. of Automated Reasoning* 7 (1991) 489–510.

[2] B.F. Chellas, *Modal Logic: An Introduction*, Cambridge University Press (1980).

[3] S. Demri, "Efficient strategies for automated reasoning in modal logics," *JELIA'94* (*Logics in AI*), *LNAI* 838 (1994) 182–197.

[4] S. Demri, private communication, Nov. 1996.

[5] P. Enjalbert and L. Farinas del Cerro, "Modal resolution in clausal form," *Theor. Comp. Sci.* 65 (1989) 1–33.

[6] F. Giunchiglia and R. Sebastiani, "Building decision procedures for modal logics from propositional decision procedures — the case study of modal K," *CADE-13* (1996).

[7] F. Giunchiglia and R. Sebastiani, "A SAT-based decision procedure for ALC," *KR'96*.

[8] R. Goré, *Cut-free sequent and tableau systems for propositional normal modal logics.* PhD thesis, Computer Laboratory, University of Cambridge, England (1992).

[9] R. Goré, W. Heinle and A. Heuerding, "Relations Between Propositional Normal Modal Logics: An Overview," Technical Report TR-ARP-16-95, Automated Reasoning Project, Australian National University (1995).

[10] J.Y. Halpern and Y. Moses, "A guide to completeness and complexity for modal logics of knowledge and belief," *Artif. Intel.* 54 (1992) 319–379.

[11] Z. Manna and A. Pnueli, *The Temporal Logic of Reactive and Concurrent Systems: Specification*, Springer-Verlag (1991).

[12] W. McCune, "OTTER 2.0", *CADE-10*, *LNAI* 449, 663–664.

[13] W. McCune, "OTTER 3.0 Reference manual and guide," Technical Report ANL-94/6, Argonne National Laboratory (1994).

[14] W. McCune, "A Davis-Putnam program and its application to finite first-order model search: Quasigroup existence problems," Technical Report ANL/MCS-TM-194, Argonne National Laboratory (1994).

[15] Moore, R., *Reasoning about Knowledge and Action*, PhD Thesis, MIT, Cambridge, USA (1980).

[16] C.G. Morgan, "Methods for automated theorem proving in nonclassical logics," *IEEE Trans. Computers* 25 (1976) 852–862.

[17] A. Nonnengart, "First-order modal logic theorem proving and functional simulation," *IJCAI-93*, 80–85.

[18] H.J. Ohlbach, "Translation methods for non-classical logics – an overview," *Bulletin of IGPL* 1 (1993) 69–91. Also: Technical Report MPI-I-93-225, Max-Planck-Institut für Informatik, Saarbrücken, Germany.

[19] H.J. Ohlbach and C. Weidenbach, "A note on assumptions about Skolem functions," *J. of Automated Reasoning* 15 (1995) 267–275.

[20] J.K. Slaney, "FINDER: Finite domain enumerator – system description," *Proc. CADE-12* (1994) 798–801.

[21] J. Slaney, M. Fujita and M. Stickel, "Automated reasoning and exhaustive search: Quasigroup existence problems," *Computers and Mathematics with Applications* 29 (1995) 115–132.

[22] L.A. Wallen, *Automated Proof Search in Non-Classical Logics*, The MIT Press (1990).

[23] H. Zhang, and M. Stickel, "Implementing the Davis-Putnam algorithm by tries," Technical Report, University of Iowa (1994).

[24] J. Zhang, *Model Generation for Many-sorted Equations and Automated Reasoning in Modal Logics*, Technical Report ISCAS-LCS-94-04, Institute of Software, Chinese Academy of Sciences, Beijing, China (1994).

[25] J. Zhang, and H. Zhang, "Generating Models by SEM," *Proc. CADE-13* (1996).