9-1990

Unification in Modal Theorem Proving

Xiaolin Zhang

Chilukuri K. Mohan
Syracuse University, ckmohan@syr.edu

SU-CIS-90-30

Unification in Modal Theorem Proving

Xiaolin Zhang and Chilukuri K. Mohan

September 1990

School of Computer and Information Science
Syracuse University
Suite 4-116
Center for Science and Technology
Syracuse, NY 13244-4100

(315) 443-2368

Unification in Modal Theorem Proving

Xiaolin Zhang Chilukuri K. Mohan

School of Computer and Information Science
4-116 Center for Science and Technology
Syracuse University, NY 13244-4100

315-443-2368, xzhang/mohan@top.cis.syr.edu

ABSTRACT: Modal formulas can be proved by translating them into a three-typed logic and then using unification and resolution, with axioms describing properties of the reachability relation among possible worlds. In this paper, we improve on the algorithms in [1], showing that "strong skolemisation" and occurrence checks are not needed for proving theorems of Q, T, Q4, and S4. We also extend the 'path logic' approach to S5, give the appropriate unification algorithm, and prove its correctness.

1 Introduction

Modal logics extend classical logic, with formulas containing additional symbols '$\Box$' (necessity) and '$\Diamond$' (possibility) [4]. One common technique for modal theorem proving is to translate modal formulas into first order logic, along with axioms representing the reachability relation defined by Kripke's possible worlds semantics for the modal logic [5, 6, 7, 11]. We can then apply the widely available, well studied techniques for classical first order theorem proving. But a major disadvantage of this method is that much of the structure of the original modal formulas is lost in the process of translation. This prevents the development of efficient theorem provers for specific modal logics. In most cases, though provability in a propositional modal logic is decidable, this approach yields only a semi-decision procedure.

To overcome this shortcoming, Auffray and Enjalbert introduced three typed first order logics, called *path logics*, tailored to fit the structure of modal logics [1, 2]. When the translated path logic formulas have the *Unique Prefix Property* (U.P.P.), the commonly used modal logics allow formulas to have a finite complete set of unifiers under certain conditions [12]. Since path logic formulas do not always have the U.P.P., Auffray and

Enjalbert introduced a special method called *Strong Skolemisation*, and showed how it can be used for unification in the modal logics T, Q, Q4, and S4.

We show that strong skolemisation is not necessary for unification in Q, T, Q4, and S4. In fact, the translated formulas without strong skolemisation already have U.P.P.! This allows us to substantially simplify the relevant unification algorithms. Another important consequence is that we have been able to derive a unification algorithm even for the modal logic S5, which could not be handled by the strong skolemisation approach in [1].

At first, the existence of a most general unifier for S5 may seem counter-intuitive. [Notation: "t/x" in a substitution denotes that the variable x is mapped to term t. Intuitively, "$\alpha!\beta$" denotes a path of possible worlds where β follows α. Greek letters $\alpha, \beta, \gamma, \delta$ with subscripts/superscripts stand for variables, and "1" is the identity element.] For instance, in the theory of S5, unifiers of $\alpha!\beta$ and $\gamma!\delta$ should include $\theta_1 = \{1/\alpha,\ \gamma!\delta/\beta\}$ as well as $\theta_2 = \{\gamma!\delta/\alpha,\ 1/\beta\}$. But these substitutions are instances of the most general unifier $\{(\alpha!\beta!\alpha'')/\gamma,\ [\alpha'']^{-1}/\delta\}$ generated by our algorithm, where α'' is a new variable. For instance, the substitution which composes with this m.g.u. to generate the less general unifier θ_1 is $\{1/\alpha,\ \gamma!\delta/\beta,\ \delta^{-1}/\alpha''\}$.

In the next section, we introduce path logics, and show how modal formulas are translated into path logics. In section 3, we define the unique prefix property, and show that translating modal formulas into path logic preserves this property. Section 4 describes our unification algorithm for S5, with examples. Section 5 contains the proof of correctness for this algorithm.

2 Path Logic

Let S be any of the modal logics Q, T, Q4, S4 or S5 (cf. [4] for details). The Path Logic $L(S)$ consists of three types (sorts) $\mathcal{A}$, $\mathcal{W}$, $\mathcal{D}$. The language of $L(S)$ is the classical typed first order language built using one of the following signatures Σ_S, depending on S. We denote by "$t : \tau$" that a term "t" is of type "τ". Note that function and predicate symbols now have an extra (first) argument of type $\mathcal{W}$.

Σ_Q: $\varepsilon : \mathcal{W}$, constant

$! : \mathcal{W} \times \mathcal{A} \to \mathcal{W}$

Function symbols, $f : \mathcal{W} \times \mathcal{D}^n \to \mathcal{D}$

Predicate symbols, $P : \mathcal{W} \times \mathcal{D}^n \to \{\mathsf{true}, \mathsf{false}\}$

Σ_T: $\Sigma_Q \cup \{1 : \mathcal{A}\}$

Σ_{Q4}: $\Sigma_Q \cup \{* : \mathcal{A} \times \mathcal{A} \to \mathcal{A}\}$

Σ_{S4}: $\Sigma_{Q4} \cup \Sigma_T$

Σ_{S5}: $\Sigma_{S4} \cup \{()^{-1} : \mathcal{A} \to \mathcal{A}\}$

Intuitively, $\mathcal{D}$ is the usual domain of individuals associated with a first order language, $\mathcal{W}$ is the set of possible worlds, following Kripke's semantics [10], and $\mathcal{A}$ is a set of operators on $\mathcal{W}$, such that *"The world w' is accessible from w"* is interpreted as *"There is some α of type $\mathcal{A}$ such that $w' = w!\alpha$"*. The constant "ε" stands for the distinguished "present world". [Notation: Binary operators associate to the left].

An interpretation $\mathbf{K}$ for $L(S)$ is a classical interpretation subject to typing constraints which imply that terms are mapped to objects of the correct type, and modulo the equational theory $E(S)$ which depends on the choice of S. $E(S)$ is chosen to be faithful to Kripke's semantics, according to which the reachability relation among possible worlds is reflexive for the logic T; transitive for Q4; reflexive and transitive for S4; and is an equivalence relation for S5.

$$E(Q) = \emptyset$$
$$E(T) = \{w!1 = w\}$$
$$E(Q4) = \{w!(a * a') = (w!a)!a', (a * a') * a'' = a * (a' * a'')\}$$
$$E(S4) = E(Q4) \cup E(T) \cup \{a * 1 = a, 1 * a = a\}$$
$$E(S5) = E(S4) \cup \{a * a^{-1} = 1\}$$

The following rules [1] show how formulas in commonly studied modal logics can be uniformly transformed into path logic formulas. We assume that implication symbols have already been eliminated from the formulas, with "$\phi \to \psi$" replaced by "$\neg\phi \vee \psi$" everywhere. The transformation is defined as the normal form obtained by repeated reduction using the following (canonical) rewrite rules, where the first step in the translation of a formula "B" is "$t(\varepsilon, B)$".

$t(\pi, \neg B) \rightarrow \neg t(\pi, B)$

$t(\pi, B_1 \vee B_2) \rightarrow t(\pi, B_1) \vee t(\pi, B_2)$

$t(\pi, B_1 \wedge B_2) \rightarrow t(\pi, B_1) \wedge t(\pi, B_2)$

$t(\pi, \forall x B) \rightarrow \forall x t(\pi, B)$

$t(\pi, \exists x B) \rightarrow \exists x t(\pi, B)$

$t(\pi, \Box B) \rightarrow \forall \alpha t(\pi ! \alpha, B)$, where α is a new variable of sort $\mathcal{A}$

$t(\pi, \Diamond B) \rightarrow \exists \alpha t(\pi ! \alpha, B)$, where α is a new variable of sort $\mathcal{A}$

$t(\pi, p(t_1, \ldots, t_n)) \rightarrow p(\pi, t_1, \ldots, t_n)$ for each predicate/proposition symbol p.

Example 1 The reduction sequence translating $\Box([\neg \forall x. \Diamond A(x)] \vee \Diamond \exists y. A(y))$ is

$$
\begin{aligned}
& t(\varepsilon, \Box([\neg \forall x. \Diamond A(x)] \vee \Diamond \exists y. A(y))) \\
& \rightarrow \forall \alpha. t(\varepsilon ! \alpha, ([\neg \forall x. \Diamond A(x)] \vee \Diamond \exists y. A(y))) \\
& \rightarrow \forall \alpha. ([t(\varepsilon ! \alpha, \neg \forall x. \Diamond A(x))] \vee [t(\varepsilon ! \alpha, \Diamond \exists y. A(y))]) \\
& \rightarrow \forall \alpha. ([\neg t(\varepsilon ! \alpha, \forall x. \Diamond A(x))] \vee [\exists \beta. t(\varepsilon ! \alpha ! \beta, \exists y. A(y))]) \\
& \rightarrow \forall \alpha. ([\neg \forall x. t(\varepsilon ! \alpha, \Diamond A(x))] \vee [\exists \beta \exists y. t(\varepsilon ! \alpha ! \beta, A(y))]) \\
& \rightarrow \forall \alpha. ([\neg \forall x \exists \gamma. t(\varepsilon ! \alpha ! \gamma, A(x))] \vee [\exists \beta \exists y. t(\varepsilon ! \alpha ! \beta, A(y))]) \\
& \rightarrow \forall \alpha. ([\neg \forall x \exists \gamma. A(\varepsilon ! \alpha ! \gamma, x)] \vee [\exists \beta \exists y. A(\varepsilon ! \alpha ! \beta, y)])
\end{aligned}
$$

Elimination of quantifiers with skolemisation then yields the clause $\neg A(\varepsilon ! \alpha ! \gamma,\ f_1(\alpha)) \vee A(\varepsilon ! \alpha ! \mathcal{F}(\alpha),\ f_2(\alpha))$, where $f_1, f_2, \mathcal{F}$ are skolem functions.

3 The Unique Prefix Property

Unification is an important part of the resolution proof procedure [14], generalized to unification (and resolution) modulo equational theories [8], [9], [13]. For resolution-based theorem-proving, it may be necessary to have a procedure to generate a *"complete set of unifiers" (CSU)* for pairs of terms such that

$$\forall \theta. (t_1 \theta \equiv t_2 \theta) \Rightarrow \exists \sigma [\sigma \in CSU(t_1, t_2) \wedge (\exists \rho). \theta \equiv \sigma \circ \rho],$$

where σ, θ, ρ are substitutions, and t_1, t_2 are terms. In general, non-trivial equational theories (*e.g.*, with an associativity axiom) have no finite complete set of unifiers, and this makes it difficult to use the generalized resolution proof procedure. Note that the modal

reachability relation is transitive for Q4, S4 and S5, hence the "$*$" operator in the corresponding path logics is associative. In some such cases (but not always[1]), the existence of a finite complete set of $E(S)$-unifiers is guaranteed by the following property.

Definition [12], [1]: A set L of path logic formulas have the unique prefix property (U.P.P.) iff for every subterm $\pi!\alpha : \mathcal{W}$ that occurs in L, π uniquely depends on α, *i.e.*, $\pi \equiv \pi'$ whenever L contains the subterms $\pi!\alpha$ and $\pi'!\alpha$.

Since formulas in path logic need not have the U.P.P., a complicated technique was used in [1], motivated by the following alleged counterexample.

Example 2 Let $E = E(Q4)$. The terms $\{\sigma!\alpha!a, \sigma!a!\alpha\}$ (where α is a variable, and a is a constant) do not have a finite complete set of E-unifiers.

In addition to added complexity, a disadvantage of the suggested strong skolemisation technique was that [1] could not include an algorithm for computing the CSU for S5. But we observe that strong skolemisation is really not necessary, for a simple reason. Although arbitrary formulas do not have a finite CSU, such formulas never arise as a result of the translation procedure described earlier! In other words, the set of translated formulas and the clauses obtained from them (using previously known skolemisation techniques) already have U.P.P., making the strong skolemisation procedure redundant. With the removal of this obstacle, we are able to give an algorithm to generate the CSU even in the case of S5 (see section 4).

In the following, we assume w.l.o.g. that variables in different modal formulas are distinct, and that only new variables or skolem-terms are introduced when each modal operator is eliminated. By "skolem-term", we refer to either a skolem constant or a term "$g(\cdots)$" whose outermost symbol g is a skolem function, introduced during the skolemisation process. In the rest of this section, "$\mathcal{C}$" denotes a set of clauses obtained by translating a set of modal logic formulas into path logic and applying the usual classical transformations to conjunctive normal form and variable-elimination (including skolemisation).

Fact 1 In the translation of any modal formula to its path logic representation, elimination of each modal operator introduces occurrences of only one new variable of type $\mathcal{A}$.

[1] *For example, U.P.P. does not ensure a finite C.S.U. for an equational theory with $a * b = 1$ where "$*$" is an associative operator with identity "1". For instance, terms $\alpha * \beta$ and $\gamma * \delta$ have no finite C.S.U.*

Fact 2 In clauses in $\mathcal{C}$, variables of type $\mathcal{A}$ occur only by themselves (as in "$\cdots!\alpha!\cdots$"), or as arguments in skolem terms. We refer to these as non-skolem and skolem occurrences respectively.

Fact 3 Every non-skolem occurrence of a type $\mathcal{A}$ variable in any clause of $\mathcal{C}$ is due to the same modal ("$\Box$") operator in the original modal formulas.

Fact 4 Each occurrence of a type $\mathcal{A}$ skolem-term in a clause in $\mathcal{C}$ is due to the same modal ("$\Diamond$") operator in the original modal formula.

Lemma 1 For any distinct variables α, β of type $\mathcal{A}$, an occurrence of α precedes an occurrence of β in any $\mathcal{W}$-term, obtained on translating a modal formula (before skolemisation), iff the corresponding original modal operator that generated α precedes the modal operator that generated β in the original modal formula.

Proof: [Note: "Δ_1 precedes Δ_2" denotes that Δ_2 is within the scope of Δ_1].

The "if"-part of the proof is a straightforward consequence of the translation procedure. For the "only-if"-part of the proof, let the modal operators associated with α and β be Δ and Δ' respectively (known to be unique, by the above mentioned facts). If α precedes β but Δ' precedes Δ in the modal formula, then it has a subformula of the form $\Delta'(\ldots\Delta B\ldots)$ such that $t(\pi, \Delta'(\ldots\Delta B\ldots)) = \nabla t(\pi!\beta, (\ldots B\ldots))$, where '$\nabla$' is $\forall\beta$ or $\exists\beta$. From this we can verify that every rule of translation will either not change $\pi!\beta$ or add some type $\mathcal{A}$ variable or skolem term after β. So, α would have to follow β, which contradicts the assumption that α precedes β.

Corollary 1 If an occurrence of α (a type $\mathcal{A}$ variable) precedes an occurrence of β in one term in a path logic formula (obtained by translating a modal formula), then an occurrence of α precedes every occurrence of β in every subterm of the same formula.

Proof: If α precedes β in a term, then the modal operator which generates α precedes the operator which generates β, by lemma 1. Then (by lemma 1 again), α precedes β in every other term in which these variables occur.

Example 3 The path logic translation of $\Box\Box(\Diamond A \vee \Box B)$ is

$$\forall\alpha\forall\beta[\exists\gamma\, A(\varepsilon!\alpha!\beta!\gamma) \vee \forall\delta\, B(\varepsilon!\alpha!\beta!\delta)].$$

Note that α precedes β in every subterm, since the modal operator contributing α to the translated form precedes that contributing β. Both these precede γ as well as δ, whereas the scoping of modal operators is such that neither γ precedes δ nor vice versa, in any term. For instance, even if the formula contained other subterms, we would expect every occurrence of γ to be preceded by the unique prefix "$\varepsilon!\alpha!\beta!$".

Corollary 2 No type $\mathcal{A}$ variable occurs more than once in one term in a formula obtained by translation of a modal logic formula.

To prove this, we observe that if α occurs more than once in a $\mathcal{W}$-term, then (by facts 3,4), α originates from the same modal operator. By Lemma 1, this operator will have to precede itself, a contradiction.

Theorem 1 Every path logic formula obtained as the translation of a modal formula (using the rules in section 2) has the unique prefix property.
Proof: If $\pi!\alpha$, $\pi'!\alpha$ are two $\mathcal{W}$-subterms of a formula in the subset such that $\pi \neq \pi'$, suppose that $\pi = \beta_1!\cdots!\beta_n$ and $\pi' = \beta'_1!\cdots!\beta'_n$. Let β_i and β'_i be the first pair such that $\beta_i \neq \beta'_i$, then since β'_i precedes α, by corollary 2, $\beta'_i \in \pi$. Similarly $\beta_i \in \pi'$. Since β_i, β'_i are the first distinct pair and there is no more than one occurrence of β_i in π and β'_i in π' (by corollary 2), it must be the case that β_i precedes β'_i in π and β'_i precedes β_i in π'. This contradicts corollary 1.

With minor modification, the above result continues to hold after the process of converting path logic formulas into clauses using skolemisation.

Theorem 2 :

- Every non-skolem occurrence of an $\mathcal{A}$-type variable in $\mathcal{C}$ has a unique prefix;
- every occurrence of an $\mathcal{A}$-type skolem term in $\mathcal{C}$ has a unique prefix;
- a variable of type $\mathcal{A}$ may occur at most once in a skolem term in $\mathcal{C}$; and
- every skolem occurrence of an $\mathcal{A}$-type variable in $\mathcal{C}$ is preceded by its non-skolem occurrence.

We note that the terms in example 2 do not satisfy corollary 1. Hence these can never arise as subterms on translating modal logic formulas to path logic. Therefore, even though there is no finite complete set of unifiers, this pair of terms is not a relevant counterexample. We can hence proceed to apply classical E-resolution-based theorem-proving techniques to the path logic formulas obtained by translating modal formulas. Neither factoring nor the application of resolving substitutions disturbs the U.P.P., hence resolution-based proof procedures can be readily applied.

4 Unification Algorithms

The algorithms we suggest for computing the complete set of unifiers for the cases of Q, Q4, T and S4 are essentially the same as those given in [1], except that we safely omit the "occurrence-check" present in the original algorithm, thanks to the unique prefix property. Following preliminary discussion, we present an E-unification algorithm for the modal logic S5. In what follows, we abuse notation a little, treating "$*$" and "!" as the same.

As in the previous section, let "$\mathcal{C}$" denote a set of clauses obtained from translations of a set of modal logic formulas into path logic $L(S5)$. To E(S5)-unify any two distinct $\mathcal{W}$-terms t and t' in $\mathcal{C}$ (or in clauses obtained by resolution from $\mathcal{C}$), we have to consider only the following non-trivial cases:

1. $t = t'!\pi$ or $t' = t!\pi$, *i.e.*, one term is a subterm of the other.
2. t' does not contain non-skolem occurrences of any variable

The only other possible case is when $t = \pi!\pi_1$ and $t' = \pi!\pi_2$, where π_1 and π_2 start with different symbols. In this case, to unify $t = \pi!\pi_1$ and $t' = \pi!\pi_2$ is the same as to unify π_1 and π_2. This is covered by case 2 above: if β is a variable with a non-skolem occurrence in π_1, then U.P.P. implies that β cannot occur in π or in π_2, hence β cannot occur in t'.

Note that every occurrence of a skolem-term must have the same prefix, hence no skolem-term can be common to π_1 and π_2. Skolem-terms whose leading symbols are different function symbols cannot be unified with each other. If both π_1 and π_2 end with skolem-terms with distinct outermost symbols, they cannot be unified.

ALGORITHM **Unifier**$(S5, t, t')$

(1)• IF $t = t'$, RETURN the identity substitution.
(2)• IF $t = 1$ or $t' = t!\alpha_i! \ldots !\alpha_n$, RETURN Unifier$(S5, t', t)$.
(3)• IF $t = t'!\alpha_1! \ldots !\alpha_n$,and $t' \neq 1$ RETURN Unifier$(S5, \alpha_1! \ldots !\alpha_n, 1)$.
(4)• IF $t = \pi!t_1$ and $t' = \pi!t'_1$, where the first symbols of t_1 and t'_1 are different,
(5) RETURN Unifier$(S5, t_1, t'_1)$.
(6)• IF the first symbols of t and t' are different, and $t' \neq 1$:
(7) Let t be $t_1! \ldots !t_N$, and t' be $t'_1! \ldots !t'_M$ respectively.
(8) CASE 1. If both t_N and t'_M are skolem-terms:
(9) Consider the longest skolem-term-sequence suffixes of t, t',
(10) *i.e.*, select the largest k, k' such that
(11) $t_{N-k}, \cdots, t_N$, and $t'_{M-k'}, \cdots, t'_M$ are skolem terms.
(12) IF $k \neq k'$, or $\exists i \leq k$ such that t_{N-i} and t'_{M-i} have different
(13) outermost (skolem) function symbols, THEN FAILURE
(14) ELSE If Unifier$(S5, t_1! \ldots !t_{N-k-1}, t'_1! \ldots !t'_{M-k-1})$ succeeds returning σ,
(15) Then [for i decreasing from k to 0 do
(16) with each pair $t_{N-i} \equiv \phi_i(s_1, \cdots, s_{n_i})$ and $t'_{M-i} \equiv \phi_i(s'_1, \cdots, s'_{n_i})$:
(17) for j increasing from 1 to n_i do:
(18) if Unifier(S5, $s_j\sigma$, $s'_j\sigma$) returns a substitution θ,
(19) then $\sigma := \sigma\theta$
(20) else FAILURE];
(21) RETURN σ
(22) Else FAILURE.
(23) CASE 2. If t'_M is not a skolem-term, RETURN Unifier$(S5, t^{-1}!t', 1)$.
(24) CASE 3. If t'_M (but not t_N) is a skolem-term, RETURN Unifier$(S5, t', t)$.
(25)• OTHERWISE *i.e.*, ($t' = 1$ and $t = t_1! \ldots !t_n$):
(26) IF t_n is not a variable THEN FAILURE,
(27) ELSE if t is itself a variable then RETURN $\{t'/t\}$
(28) else
(29) Let t be written w.l.o.g. as $C_1!\alpha_2! \ldots !\alpha_{2k}!C_{2k+1}!\alpha_{2k+2} \ldots !\alpha_{2m}$,
(30) where each C_i is either 1 or a sequence $s_1! \ldots !s_K$ of skolem terms;
(31) Let σ_0 be the identity substitution.
(32) For i increasing from 1 to m, do:
(33) $\sigma_i := \sigma_{i-1} \cup \{C_{2i-1}^{-1}\sigma_{i-1} * \alpha'_{2i}/\alpha_{2i}\}$, where each α'_{2i} is a new variable;
(34) Let $\sigma' = \{\alpha''_2/\alpha'_2, \ldots, [\alpha''_{i-2}]^{-1} * \alpha''_i/\ \alpha'_i \ldots, [\alpha''_{2m-2}]^{-1}/\alpha'_{2m}\}$
(35) where each α''_{2i} is a new variable.
(36) RETURN $\sigma_m \circ \sigma'$.

Since every term has an inverse, if π_2 ends with a variable (*i.e.*, $\pi_2 \equiv t_1!\cdots!\alpha$), then we attempt to unify $(\pi_1)^{-1}!\pi_2$ with "1" (the identity). In this process, since variables with non-skolem occurrences in π_1 do not occur in π_2, the entire inverted term $(\pi_1)^{-1}$ may be treated as equivalent to a new term "$f(\beta_1,\ldots,\beta_n)$" where each β_i is a variable occurring in π_1. Implicitly, it is assumed that β_i are "independent" variables, *i.e.*, the unifier does not assign any terms to variables β_i, but the variables in π_2 are assigned terms that depend on $\beta_1,\ldots,\beta_n$.

We assume in the algorithm that at every stage terms may be reduced to their "inverse-normal-form", reducing subterms by applying the following rewrite rules:

$$\{(\alpha^{-1})^{-1} \rightarrow \alpha, \quad (\alpha!\beta)^{-1} \rightarrow (\beta^{-1}!\alpha^{-1}), \quad \alpha!1 \rightarrow \alpha, \quad 1!\alpha \rightarrow \alpha\}$$

Example 4 Consider the task of unifying $\alpha_1!\alpha_2!g(\alpha_1,\alpha_2)!\alpha_3$ with $\beta_1!h(\beta_1)!\beta_2$, where each α_i and each β_j is a variable, g and h are skolem functions, This is converted into the task of unifying 1 with $[\alpha_1!\alpha_2!g(\alpha_1,\alpha_2)!\alpha_3]^{-1}!\beta_1!h(\beta_1)!\beta_2$. The result of E(S5)-unification using our algorithm is

$$\{\alpha_1!\alpha_2!g(\alpha_1,\alpha_2)!\alpha_3!\beta_3/\beta_1,\ [h(\alpha_1!\alpha_2!g(\alpha_1,\alpha_2)!\alpha_3!\beta_3)]^{-1}!\beta_3^{-1}!\beta_4/\beta_2\ \}.$$

Example 5 Consider the formula $\Box(\Diamond A \wedge \Diamond\neg A \wedge \Box A)$, whose path logic translation is $\forall\theta[\exists\beta A(\varepsilon!\theta!\beta) \wedge \exists\gamma\neg A(\varepsilon!\theta!\gamma) \wedge \forall\delta A(\varepsilon!\theta!\delta)]$. Three clauses are obtained from this after variable renaming: $A(\varepsilon!\theta_1!f(\theta_1))$, $\neg A(\varepsilon!\theta_2!g(\theta_2))$, and $A(\varepsilon!\theta_3!\delta)$, where f and g are skolem functions.

In an attempt to resolve the first two clauses, we would first try to unify $\varepsilon!\theta_1!f(\theta_1)$ and $\varepsilon!\theta_2!g(\theta_2)$. Eliminating the common prefix (line 4 in the algorithm), we then call Unifier($S5, \theta_1!f(\theta_1),\ \theta_2!g(\theta_2)$). Unification fails by the IF-THEN part of CASE 1 (line 13) in the above algorithm, since the terminal skolem terms have different outermost function symbols. We hence cannot resolve the first two clauses to obtain a contradiction, even in the theory of E(S5).

We now attempt to resolve the last two clauses, calling Unifier($S5, \theta_2!g(\theta_2), \theta_3!\delta$), after eliminating the common prefix as before. By CASE 2 in the algorithm, we then call Unifier($S5, [\theta_2!g(\theta_2)]^{-1}!\theta_3!\delta,\ 1$). Now the last ('OTHERWISE') clause of the algorithm applies, and the 'ELSE' and 'else' branches (line 28) are taken. $[\theta_2!g(\theta_2)]^{-1}$ corresponds to

C_1 in the description of the algorithm, and C_3 is just 1 since there is nothing between θ_3 and δ. σ_1 is $\{[[\theta_2!g(\theta_2)]^{-1}]^{-1} * \alpha_2'/\theta_3\}$, and σ_2 is $\sigma_1 \cup \{\alpha_4'/\delta\}$. σ' is $\{\alpha_2''/\alpha_2',\ [\alpha_2'']^{-1}/\alpha_4'\}$. Finally, the unifier $\sigma_2\sigma' = \{[\theta_2!g(\theta_2)] * \alpha_2''/\theta_3,\ [\alpha_2'']^{-1}/\delta\}$ is returned. Unification succeeds, and hence so does resolution to the empty clause.

5 Proof of Correctness

In the following theorem and the lemmas that follow, we are concerned only with terms in path logic formulas obtained as a result of translating modal formulas using the transformation rules given in section 2.

Theorem 3 The algorithm Unifier$(S5, t, t')$ returns the most general $E(S5)$-unifier whenever t and t' are $E(S5)$- unifiable, and returns FAILURE otherwise.

The proof follows from the lemmas that follow. For every pair of terms t, t', since there is an 'OTHERWISE' clause (line 25) in the algorithm, a call to Unifier$(S5, t, t')$ always results in further recursive calls, or in failure, or in successfully returning some substitution. Termination (lemma 2) ensures that either failure or a substitution is always returned. By lemma 5, failure implies the absence of an E(S5)-unifier. By lemma 4, the substitution returned is indeed a most general E(S5)-unifier.

Lemma 2 Unifier$(S5, t, t')$ always terminates.

Proof: The only possible sources of non-termination are the recursive calls in lines 3, 23, 24, 2, 5, 14 and 18.

- If line 3 is invoked, then there is only a recursive call Unifier$(S5, t', 1)$, which terminates because it is evaluated by the OTHERWISE clause (line 25) from which there is no further recursion.
- The invocation of Unifier$(S5, t^{-1}!t', 1)$ in line 23 terminates for the same reason.
- In line 24, there is a call to Unifier$(S5, t', t)$ which can be evaluated only by line 23, which leads to termination.

- If line 2 is invoked, then $t = 1$ or $t' = t!\alpha_1 \ldots \alpha_n$. If $t = 1$, the recursive call Unifier$(S5, t', 1)$ terminates as in the case of line 3. If $t' = t!\alpha_1 \ldots \alpha_n$, the recursive call Unifier$(S5, t!\alpha_1 \ldots \alpha_n, t)$ is evaluated on line 3 of the algorithm, leading to termination (as mentioned above).

- Line 5 is invoked when $t = \pi!t_1$ and $t' = \pi!t'_1$ have a common prefix π. The recursive call Unifier$(S5, t_1, t'_1)$ has arguments that are strictly smaller (in size) than the original arguments, hence this cannot lead to non-termination.

- The recursive call Unifier$(S5, t_1!\cdots!t_{n-k-1}, t'_1!\cdots!t'_{n-k-1})$ in line 14 also calls the algorithm on prefixes of t, t', *i.e.*, arguments that are strictly smaller in size. Hence this call cannot lead to non-termination.

- Termination in the case of the recursive call Unifier$(S5, s_j\sigma, s'_j\sigma)$ in line 18 is proved as follows. From the method of translating formulas into path logic, s_j, s'_j do not contain skolem function symbols. We also observe (from the substitutions in lines 33-34 of the algorithm) that no term of the form '$\cdots!\phi(\cdots)$' is ever substituted for a variable, where ϕ is a skolem function. Hence $s_j\sigma$ and $s'_j\sigma$ do not end with skolem terms, *i.e.*, are not of the form '$\cdots!\phi(\cdots)$' where ϕ is a skolem function. Hence evaluation of the recursive call Unifier$(S5, s_j\sigma, s'_j\sigma)$ will never satisfy the conditions for CASE 1 (line 8), hence this call is not repeated, and cannot lead to non-termination.

□

Lemma 3 Let φ be a skolem function symbol. If σ is the m.g.u. of t and t', and σ' is the m.g.u. of $\sigma(\overline{s})$ and $\sigma(\overline{s'})$, then $\sigma \circ \sigma'$ is the m.g.u. of $t!\varphi(\overline{s})$ and $t'!\varphi(\overline{s'})$.

Proof: Let σ_1 be any unifier of $t!\varphi(\overline{s})$ and $t'!\varphi(\overline{s'})$. Then σ_1 is also a unifier of t and t', so σ_1 is an instance of σ, the m.g.u. of t, t'. Let $\sigma_1 = \sigma \circ \sigma_2$. Since σ_1 must also be a unifier of $\overline{s}$ and $\overline{s'}$, so σ_2 is a unifier of $\sigma(\overline{s})$ and $\sigma(\overline{s'})$. Hence σ_2 is an instance of σ', the m.g.u. of $\sigma(\overline{s})$ and $\sigma(\overline{s'})$. If $\sigma_2 = \sigma' \circ \sigma_3$, we have $\sigma_1 = \sigma \circ (\sigma' \circ \sigma_3)$. By the associativity of composition, σ_1 is an instance of $\sigma \circ \sigma'$, *i.e.*, $\sigma \circ \sigma'$ is the m.g.u. of $t!\varphi(\overline{s})$ and $t'!\varphi(\overline{s'})$.

□

Lemma 4 When Unifier$(S5, t, t')$ terminates with success, the substitution it computes is the most general unifier of t and t' in the theory of E(S5).

Proof: The success cases in lines 1, 2, 3, 5, 23, 24 and 27 are straightforward: unification is commutative; the identity substitution is more general than any other substitution; and unifiers of t_1 with t_2 are exactly the unifiers of $t_1^{-1}!t_2$ with 1, the identity for "!" and "$*$" operators. The preceding lemma covers CASE 1 (line 21) in the algorithm. The core of the algorithm is in the OTHERWISE case, and we accordingly analyze the remaining success case (line 36). We first note that from the definition of "$*$", it follows that unification of $\alpha_1!\ldots!\alpha_n$ with $\beta_1!\ldots!\beta_n$ is the same as unification of $\alpha_1 * \ldots * \alpha_n$ with $\beta_1 * \ldots * \beta_n$. From the construction of the algorithm, it is a straightforward consequence that the substitution returned by the algorithm is indeed an E(S5)-unifier of the argument terms; we now need to show that it is the most general unifier.

Let $\sigma_m \circ \sigma'$ be the substitution returned by the algorithm (cf. lines 25-36). We now prove that, for any unifier $\sigma_1 = \{\pi_2/\ \alpha_2 \ldots \pi_{2m}/\alpha_{2m}\}$ of $C_1!\alpha_2!\ldots!C_{2m-1}!\alpha_{2m}$ with 1, we have $\sigma_1 = \sigma_m \circ \sigma' \circ \theta$, where θ is defined as the following, where α_i' and α_i'' are new variables not occurring in t, t'. There are three cases to be considered.

Case

1. if α_i'/α_i is in σ_m and α_i''/α_i' is in σ'
   then π_i/α_i'' is in θ.
2. if $C_{2i-1}^{-1}\sigma_{i-1} * \alpha_{2i}'/\alpha_{2i}$ is in σ_m and $\alpha_{2i}''/\alpha_{2i}'$ is in σ'
   then $C_{2i-1}\sigma_{i-1} * \pi_{2i}/\alpha_{2i}''$ is in θ.
3. if $C_{2i-1}^{-1}\sigma_{i-1} * \alpha_{2i}'/\alpha_{2i}$ is in σ_m and $\alpha_{2i-2}^{-1}{}'' * \alpha_{2i}''/\alpha_{2i}'$ is in σ'
   then $C_{2i-1}\sigma_{i-1} * \alpha_{2i-2}'' * \pi_{2i}/\alpha_{2i}''$ is in θ.

Since an appropriate θ can be constructed showing that any unifier in an instance of the substitution returned by the algorithm, the latter is indeed the most general unifier.

□

Lemma 5 If Unifier$(S5, t, t')$ returns FAILURE, then t and t' are not E(S5)-unifiable.

Proof: FAILURE is returned at lines 13, 20, 22, 26 in the algorithm: these cases are analyzed below in reverse order.

- In line 26, $t' = 1$ and $t = t_1!\ldots!t_n$ where t_n is a skolem term in which occur all the variables in t_1 through t_{n-1}. By the 'occurs-check' criterion, no substitution for these variables can make $t_1!\ldots!t_{n-1}$ E(S5)-equivalent to t_n^{-1}. Hence t cannot E(S5)-unify with 1.

- Line 22 is invoked if prefixes of the given terms are found to be not E(S5)-unifiable. If π and π' are not E(S5)-unifiable, and if all variables of π occur in a skolem term p and if all variables of π' occur in a skolem term p', then $\pi!p$ cannot E(S5)-unify with $\pi'!p'$. Applying this argument $k+1$ times, we conclude that $(t_1!\ldots!t_{N-k-1})!t_{N-k}!\ldots!t_N$ cannot E(S5)-unify with $(t'_1!\ldots!t'_{M-k-1})!t_{M-k}!\ldots!t_M$, if $t_1!\ldots!t_{N-k-1}$, $t'_1!\ldots!t'_{M-k-1}$ are not E(S5)-unifiable, and the terms following them are skolem-terms (which contain the variables of the relevant prefix). Hence t, t' are not E(S5)-unifiable in this case.

- We now consider the case of line 20, which results when Unifier$(S5, s_j\sigma, s'_j\sigma)$ fails, where σ was the result of composing the unifier σ_{prefix} obtained for $t_1!\ldots!t_{N-i-1}$, $t'_1!\ldots!t'_{M-i-1}$ with the substitutions successively E(S5)-unifying the terms $s_k\sigma_{prefix}$, $s'_k\sigma_{prefix}$ where $1 \leq k < j$, where t is $t_1!\ldots!t_{N-i-1}!\phi_i(s_1,\ldots,s_{n_i})!\ldots!\phi_0(\ldots)$ and t' is $t'_1!\ldots!t'_{M-i-1}!\phi_i(s'_1,\ldots,s'_{n_i})!\ldots!\phi_0(\ldots)$. By lemma 4, we may assume that σ is indeed the E(S5)-m.g.u. of these terms. Under these conditions, if some pair $s_j\sigma, s'_j\sigma$ are not E(S5)-unifiable, then $\phi_i(s_1,\ldots,s_{n_i})\sigma$ is not E(S5)-unifiable with $\phi_i(s'_1,\ldots,s'_{n_i})\sigma$, and $\phi_i(s_1,\ldots,s_{n_i})$ is not E(S5)-unifiable with $\phi_i(s'_1,\ldots,s'_{n_i})$. This in turn implies that $t_1!\ldots!t_{N-i-1}!\phi_i(s_1,\ldots,s_{n_i})$ and $t'_1!\ldots!t'_{M-i-1}!\phi_i(s'_1,\ldots,s'_{n_i})$ are not E(S5)-unifiable. Finally, by the same argument as given above for the case of line 22, this implies that $t_1!\ldots!t_{N-i-1}!\phi_i(s_1,\ldots,s_{n_i})!\ldots!\phi_0(\ldots)$ and $t'_1!\ldots!t'_{M-i-1}!\phi_i(s'_1,\ldots,s'_{n_i})!\ldots!\phi_0(\ldots)$ are not E(S5)-unifiable.

- For the case of line 13, let $t \equiv t_1!\ldots!t_N$ and $t' \equiv t'_1!\ldots!t_M$, where $t_{N-k},\ldots,t_N$ and $t'_{M-k'},\ldots,t'_M$ are the skolem terms in the suffixes of t, t' respectively. Suppose t and t' are E(S5)-unifiable by a substitution σ. Then $t\sigma$ and $t'\sigma$ must be identical, modulo the associativity, inverse and identity properties. We assume w.l.o.g. that the variables in t and t' are distinct, and let $\sigma \equiv \sigma_1 \uplus \sigma_2$, where σ_1 is the restriction of σ to the variables of t and σ_2 is the restriction of σ to variables of t'. Let $t\sigma_1 \equiv \pi!t_{N-k}\sigma_1!\ldots!t_N\sigma_1$ and $t'\sigma_2 \equiv \pi'!t'_{M-k'}\sigma_2!\ldots!t'_M\sigma_2$ respectively, where π and π' are obtained by applying the

unifier σ to the prefixes of t, t'. When maximally simplified, $t\sigma_1$ and $t'\sigma_2$ have this form, since π cannot contain occurrences of the skolem symbols of $t_{N-k} \ldots t_N$, and similarly with π'; the skolem symbols of each t_{N-i} and t'_{M-j} cannot disappear from these expressions. Unifiability implies that $t\sigma_1 \equiv t'\sigma_2$.

- If $\pi \equiv \pi'$ then we must have $t_{N-k}\sigma_1! \ldots !t_N\sigma_1 \equiv t'_{M-k'}\sigma_2! \ldots !t'_M\sigma_2$, which is only possible if $k = k'$ and each t_{N-i}, t'_{M-i} have the same outermost skolem function symbol. This justifies line 13: if $k \neq k'$, or corresponding skolem terms have different function symbols, then the terms are not E(S5)-unifiable.
- If π and π' are not identical, the requirement that $t\sigma_1 \equiv t'\sigma_2$ may be satisfied if π is a prefix of π' or vice versa. Assume w.l.o.g. that $\pi \equiv \pi'!\pi_1$ for some nonempty (sequence) π_1, and there is a non-empty suffix of $t'\sigma_2$ which is identical to $t_{N-k}\sigma_1! \ldots !t_N\sigma_1$, implying that $t_N\sigma_1 \equiv t'_M\sigma_2$ which implies that t_N, t'_M must have the same outermost skolem function symbol. In this case $\pi'!\pi_1!t_{N-k}\sigma_1! \ldots !t_N\sigma_1 \equiv \pi'!t'_{M-k'}\sigma_2! \ldots !t'_M\sigma_2$, which implies that the skolem function symbol of $t'_{M-k'}$ occurs in π_1. Hence some variable x in t must have been assigned (by σ_1) a term which contains $t'_{M-k'}\sigma_2$. Since each variable in t occurs in t_N, we observe that $t_N\sigma_1$ contains $t'_{M-k}\sigma_2$ as a subterm of one of its arguments, say the i^{th} argument. Let the i^{th} argument of t'_M be y_i. The terms $t_N\sigma_1$ and $t'_M\sigma_2$ cannot be the same (even if they have the same outermost skolem function symbol), because $y_i\sigma_2$ is contained as a proper subterm of the i^{th} argument of $t_N\sigma_1$. By reductio ad absurdum, since we have assumed the terms to be $E(S5)$-unifiable by σ, it is not possible for π_1 to be non-empty, hence it is sufficient to consider only the preceding case ($\pi \equiv \pi'$).

□

6 Conclusions

We have simplified algorithms for modal theorem-proving based on path logic, improving on the work in [1]. We have given a unification algorithm for S5, not obtained earlier, which generates the most general unifier of path logic terms obtained from modal formulas, in the theory of S5. The main advantage of the path logic-based technique is that the target logic preserves the distinction between individuals and worlds, exploiting the properties of the reachability relation among possible worlds. Our results vindicate the approach suggested by [1]; the overall result is now cleaner and simpler. This approach seems promising, and likely to yield other efficient procedures for modal theorem-proving.

Acknowledgements: We thank Profs. Howard Blair, Allen Brown, and Patrice Enjalbert for comments and helpful suggestions.

References

[1] Y.Auffray and P.Enjalbert, *Modal Theorem Proving: An Equational Viewpoint*, Proc. Intl. Joint Conf. on A.I., 1989.

[2] Y.Auffray and P.Enjalbert, *Modal Theorem Proving Using Equational Methods*, Tech.Rep. 88-11, Laboratoire d'Informatique de l'Universite de Caen, France, November 1988.

[3] C.L.Chang and R.C.Lee, *Symbolic Logic and Mechanical Theorem-Proving*, Academic Press, 1973.

[4] B.F. Chellas, *Modal Logic: An Introduction*, Cambridge Press, 1980.

[5] P.Enjalbert and L.Farinas del Cerro, *Modal Resolution in Clausal Form*, Theoretical Computer Science, Vol.65, pp1-33, 1989.

[6] L.Farinas del Cerro, *Resolution Modal Logic*, Logique et Analyse 110/111, pp152-172, 1985.

[7] L.Farinas del Cerro, *A simple deduction method for modal logic*, Information Processing Letters, Vol.14, No.2, pp49-51, 1982.

[8] F.Fages, *Formes canoniques dans les algebres booleennes et application a la demonstration automatique en logique du premier ordre*, These de 3eme cycle, Univ. Paris (France), 1983.

[9] M.Fay, *First Order Unification in Equational Theories*, Fourth workshop on Automated Deduction, Austin (Texas), 1979.

[10] S.Kripke, *Semantical Considerations on Modal Logic*, Acta Philosophica Fennica 16, pp83-94, 1963.

[11] C.G. Morgan: *Methods for Automated Theorem Proving in Nonclassical Logics*, IEEE Trans. Comp. Vol. c25. No.8, Aug.1976.

[12] H.J.Ohlbach: *A Resolution Calculus for Modal Logic*, Proc. 9th Int.Conf. Auto. Deduc., (Eds: Lusk and Overbeek), Springer-Verlag LNCS 310, 1988.

[13] G.Plotkin, *Building in Equational Theories*, Machine Intelligence 7, pp73-90, 1972.

[14] J.A. Robinson, *A Machine-Oriented Logic based on the Resolution Principle*, J. ACM 20(1), pp23-41, Jan. 1965.