INFORMATIK

Proving a conjecture of Andreka on temporal logic

Jean-François Raskin
Pierre-Yves Schobbens

MPI–I–99–3–004 August 1999

FORSCHUNGSBERICHT RESEARCH REPORT

MAX-PLANCK-INSTITUT
FÜR
INFORMATIK

Im Stadtwald 66123 Saarbrücken Germany

Authors' Addresses

Jean-François Raskin
Max-Planck-Institut für Informatik
Im Stadtwald
66123 Saarbrücken
`jfr@mpi-sb.mpg.de`

Pierre-Yves Schobbens
Computer Science Department
University of Namur
Namur
Begium `pys@info.fundp.ac.be`

Publication Notes

A revised version of this report has been accepted for publication in *Information Processing Letters*.

Acknowledgements

This work was supported by the Belgian National Fund for Scientific Research (FNRS).

Keywords

Temporal Logic, Completness, Decidability, Complexity.

1 Introduction

The temporal logic has been created by Prior [13, 14]. Its importance for program construction and verification has been noted by [12], and has since then been a topic of intensive research [9, 1, 4, 6, 7]. The creation of complete inference systems, as required for practical proofs of programs, is an important sub-topic. A rich propositional discrete-time linear temporal logic is treated in [9]: we shall use the same technique here. Treating first-order [17, 2], second-order, branching, or dense-time [5, 15] temporal logics is more difficult and often only relative completeness results are available.

In this article we deal with a simple propositional temporal logic restricted to two modal operators: *initially*, noted $\odot$, and *next*, noted $\bigcirc$. This logic is very useful since sequential and concurrent programs are described by their initial values and by their transitions [8, 10, 11]. The logic thus allows to prove invariance properties of programs, of the form $\gamma \vdash \beta$, where γ is a description of the program, and β is an invariance property to be checked.

Section 2 recalls the problem [3], Section 3 will give simple lemmas for use in the completeness proof, which form Section 4. Section 5 sketches a simple tableau algorithm for deciding the logic. Section 6 shows that this algorithm is theoretically optimal, and thus gives the complexity deciding entailment and satisfiability, which are PSPACE-complete. In contrast, validity is only co-NP-complete. This might be surprising, since all temporal logics studied by [16] have the same complexity for both problems. In modal logics, this needs not be the case, since satisfiability is not the complement of validity.

2 The logic of initially and next

For programs, time is modelled as a sequence of execution steps, beginning when the program is launched. Time is thus *discrete*: we are not interested in what happens inside a step, and *linear*: we observe the execution sequences of the program, but do not want to look into the decisions open to the program. Therefore, we model time by the natural numbers. Only the ordering of time points is relevant: thus we do not need operations like multiplication, and we use the structure of time $\mathbb{T} = \langle \mathbb{N}, 0, succ, \leq \rangle$.

2.1 Syntax

Given a set of proposition symbols $p \in P$, the syntax of the the logic is defined by:

$$\phi ::= \bigcirc\phi \mid \ \odot\,\phi \mid p \mid \phi_1 \to \phi_2 \mid \neg\phi$$

We will also use the other propositional connectives as shorthands.

2.2 Semantics

Since the structure of time is fixed, we only need to know the evolution of propositions; we record the subset of propositions that are true.

$$Mod = \mathbb{N} \to 2^P$$

Given a model and a point of time, we are now ready to define the satisfaction of formulae:

$$\begin{array}{lll}
M, t \Vdash p & \equiv & p \in M(t) \\
M, t \Vdash \phi_1 \to \phi_2 & \equiv & M, t \Vdash \phi_2 \text{ if } M, t \Vdash \phi_1 \\
M, t \Vdash \neg\phi & \equiv & M, t \Vdash \phi \text{ does not hold.} \\
M, t \Vdash \bigcirc\phi & \equiv & M, succ(t) \Vdash \phi \\
M, t \Vdash \odot\phi & \equiv & M, 0 \Vdash \phi
\end{array}$$

A model *satisfies* are formula, noted $M \Vdash \phi$ iff $M, t \Vdash \phi$ at all times $t \in \mathbb{N}$. A formula ϕ is *satisfiable* if we can find M satisfying it: $M \Vdash \phi$. A formula *entails* another, noted $\gamma \models \beta$ iff $M \Vdash \gamma$ implies $M \Vdash \beta$ for any model M. We are interested by this entailment relation between formulae. Axiomatizing it is called *medium completeness* in [3], to contrast it with *strong completeness* where we allow an infinite set of formulas on the left instead of γ, and *weak completeness* where no γ is allowed. No effective proof system can be strongly complete for this logic, since the logic is not compact.

2.3 Proof theory

To perform proofs of this type, we use a Hilbert system: the formula γ will be used as a supplementary axiom, and we also use the following axioms and inference rules, on top of the well-known boolean rules:

$$\odot(\phi \to \psi) \to (\odot\phi \to \odot\psi) \tag{K$_\odot$}$$

$$\bigcirc(\phi \to \psi) \to (\bigcirc\phi \to \bigcirc\psi) \tag{K$_\bigcirc$}$$

$$\odot\neg\phi \leftrightarrow \neg \odot \phi \tag{fun$_\odot$}$$

$$\bigcirc\neg\phi \leftrightarrow \neg \bigcirc \phi \tag{fun$_\bigcirc$}$$

$$\odot\phi \leftrightarrow \odot \odot \phi \tag{con'$_\odot$}$$

$$\odot\phi \leftrightarrow \bigcirc \odot \phi \tag{con''$_\odot$}$$

$$\frac{\phi, \quad \phi \to \psi}{\psi} \tag{MP}$$

$$\frac{\phi}{\odot\phi} \tag{NEC$_\odot$}$$

$$\frac{\phi}{\bigcirc\phi} \tag{NEC$_\bigcirc$}$$

$$\frac{\psi \to \bigcirc\psi, \quad \odot\psi}{\psi} \tag{IND}$$

If we can provide a Hilbert proof of a formula β without using the supplementary axiom γ, we say that β is a theorem and we write this as $\vdash \beta$. If we used γ, we write $\gamma \vdash \beta$. In [3], $\vdash$ is noted $\vdash^0_{\odot\bigcirc}$.

3 Preparation

By looking at the proposed inference system, one can make simple but useful observations that will be used in the next section:

Lemma 1 *The replacement rule is derivable (two equivalent formulae can be replaced in any context):*

$$\frac{\phi \leftrightarrow \psi}{\chi(\phi) \leftrightarrow \chi(\psi)} \tag{RE}$$

Proof. By induction on χ: for boolean connectives this is a basic property, for temporal connectives we deal with each direction of the equivalence separately, use necessitation, then modus ponens.

□

Lemma 2 *(AND) $\odot(\phi \wedge \psi) \to \odot\phi$ is derivable.*

Proof. We use NEC$_\odot$ on $(\phi \wedge \psi) \to \phi$, then K$_\odot$ and MP.

□

Lemma 3 *Any boolean operator commutes with* $\odot$ *and* $\bigcirc$, *i.e.:*

$$\odot(\phi \wedge \psi) \leftrightarrow \odot\phi \wedge \odot\psi \qquad (\mathrm{COM}_{\wedge\odot})$$
$$\odot(\phi \to \psi) \leftrightarrow \odot\phi \to \odot\psi \qquad (\mathrm{COM}_{\to\odot})$$
$$\odot(\phi \vee \psi) \leftrightarrow \odot\phi \vee \odot\psi \qquad (\mathrm{COM}_{\vee\odot})$$
$$\bigcirc(\phi \wedge \psi) \leftrightarrow \bigcirc\phi \wedge \bigcirc\psi \qquad (\mathrm{COM}_{\wedge\bigcirc})$$
$$\bigcirc(\phi \to \psi) \leftrightarrow \bigcirc\phi \to \bigcirc\psi \qquad (\mathrm{COM}_{\to\bigcirc})$$
$$\bigcirc(\phi \vee \psi) \leftrightarrow \bigcirc\phi \vee \bigcirc\psi \qquad (\mathrm{COM}_{\vee\bigcirc})$$

Proof. We treat just the implication since the other operators are defined from it. $\odot(\phi \to \psi) \leftrightarrow \odot\phi \to \odot\psi$. The first direction $\to$ is just the axiom $\mathrm{K}_\odot$. For the other direction, we prove the contraposite $\neg \odot (\phi \to \psi) \to \neg(\odot\phi \to \odot\psi)$. Using $\mathrm{fun}_\odot$, it gives $\odot(\phi \wedge \neg\psi) \to (\odot\phi \wedge \neg \odot \psi)$, which is provable by two uses of (AND). All other boolean connectives are definable for $\neg, \to$. The proof for $\bigcirc$ is similar.

□

Lemma 4 *(NXT)* $\neg \bigcirc \bot$ *is a theorem.*

Proof. By $\mathrm{NEC}_\bigcirc$, $\mathrm{fun}_\bigcirc$.

□

Lemma 5 *The rule (IND')* $\frac{\phi\to\bigcirc\phi}{\odot\phi\to\phi}$ *is derivable.*

Proof. We instantiate the induction rule $\frac{\psi\to\bigcirc\psi, \quad \odot\psi}{\psi}$ by $\psi := (\odot\phi \to \phi)$. Using (COM), (con'$_\odot$), (con"$_\odot$) the rule simplifies to $\frac{\odot\phi\to\odot\phi, \quad \odot\phi\to(\phi\to\bigcirc\phi)}{\odot\phi\to\phi}$. The first antecedent is trivial, and the second can be proved from the antecedent of the rule to be proved.

□

4 Completeness

In this section, we prove the medium completeness of the proof system proposed in [3]. So we must prove that if $\gamma \vDash \beta$ then $\gamma \vdash \beta$. We construct a model by a variant of the filtration of the canonical model, as in [9].

Definition 1 *The* closure C *is the smallest set:*

1. *containing all subformulae of* γ, β,
2. *closed under* $\odot, \neg$.

3. such that $\odot\phi \in C \rightarrow \bigcirc \odot \phi \in C$

This set is used to obtain a finite number of relevant formulae, to enable the later induction step. This is called filtration.

We preserve finiteness by using the simplification rules $\neg\neg\phi \leftrightarrow \neg\phi$, $\odot \odot \phi \leftrightarrow \odot\phi$ (con'$_\odot$), $\odot\neg\phi \leftrightarrow \neg \odot \phi$, $\bigcirc \odot \phi \leftrightarrow \odot\phi$. So, for instance, we will note that $\odot \odot p \in a$ even though we only keep $\odot p \in a$. Now, we define a sequence of structures $S^0, S^1, \ldots, S^k$ and study their properties:

Definition 2 *The structure* $S^0 = (A^0, R^0_\bigcirc, R^0_\odot)$ *where:*

- A^0 *is the set of* atoms, *i.e. maximally propositionally consistent subsets of* C, *i.e.:*
  - *for all* $a \in A^0$, *for all* $\phi \in C$, $\phi \in a$ *iff* $\neg\phi \notin a$,
  - *for all* $a \in A^0$, *for all* $\phi_1 \vee \phi_2 \in C$, $\phi_1 \vee \phi_2 \in a$ *iff* $\phi_1 \in a$ *or* $\phi_2 \in a$.
- *for all* $a, b \in A^0$, $aR^0_\bigcirc b$ *iff* $\forall \bigcirc \phi \in C : \bigcirc\phi \in a \leftrightarrow \phi \in b$;
- *for all* $a, b \in A^0$, $aR^0_\odot b$ *iff* $\forall \odot \phi \in C : \odot\phi \in a \leftrightarrow \phi \in b$;

Note that $R^0_\bigcirc$ is not functional, since C is not closed under $\bigcirc$, whereas $R^0_\odot$ is a partial function since C is closed under $\odot$. Had we added (COM) in our simplification rules, it would have been total.

Lemma 6 $R^0_\odot(a) = \{\phi | \odot \phi \in a\}$

Proof. $R^0_\odot(a) \supset \{\phi | \odot \phi \in a\}$ by definition of $R^0_\odot$. This set is maximal, since the closure is closed under $\odot$, so that $\forall \phi \in C, \odot\phi \in a$ or $\neg \odot \phi \in a$. In this second case, $\neg \odot \phi$ is the same as $\odot\neg\phi$.

□

Lemma 7 $\phi \in R^0_\odot(a) \leftrightarrow \odot\phi \in R^0_\odot(a)$

Proof. Since $\odot\phi = \odot \odot \phi \in a$, by simplification.

□

Lemma 8 $aR^0_\bigcirc b$ *implies* $R^0_\odot(a) = R^0_\odot(b)$

Proof. $\odot\phi$ is the same as $\bigcirc \odot \phi$, thus b will contain the same initial formulae.

□

In those structures, we are interested by particular paths:

Definition 3 *A path* $\pi = a_0 a_1 \ldots a_n \ldots$ *fulfills* β *under hypothesis* γ *in structure* S^0 *iff*

- *there exists* $n \geq 0$ *such that* $\beta \in a_n$*;*
- *for all* $i \geq 0$, $a_i R^0_{\odot} a_0$*;*
- *for all* $i \geq 0$, $a_i R^0_{\bigcirc} a_{i+1}$*;*
- *for all* $i \geq 0$, $\gamma \in a_i$*;*

Lemma 9 $\gamma \vDash \neg\beta$ *iff there is no path* π *in* S^0 *that fulfills* β *under the hypothesis* γ

Proof.(sketch) First, every model M such that there exists $i \geq 0$ with $M, i \vDash \beta$ and for all $j \geq 0$, $M, j \vDash \gamma$, can be transformed into a fulfilling path as follows: take $\pi = a_0 a_1 \ldots a_n \ldots$ with $a_i = \{\phi | \phi \in C \text{ and } M, i \vDash \phi\}$. It is trivial to verify that π is a fulfilling path for β under γ in S^0. For the other direction, we prove that every fulfilling path has the Hintikka property, i.e. for all $i \geq 0$, for every $\phi \in a_i$, $M_\pi, i \vDash \phi$, where $M_\pi = p_0 p_1 \ldots p_n \ldots$ is the sequence of states obtained by projecting the atoms on the propositions, i.e $p_i = a_i \cap P$.

□

We now define a series of structure S^i from S^0 by deleting atoms that can not take part into a model for β under the hypothesis γ. Formally:

Definition 4 *The atom* a *is not useful in* S^i *iff either:*

1. $\gamma \notin a$*: under hypothesis* γ*,* γ *must be true in every atom;*
2. $a \notin (R^i_{\bigcirc})^*(R^i_{\odot}(a))$ [1]*: every atom in a fulfilling path must be reachable from its initial atom;*
3. $\neg\exists b : a R^i_{\bigcirc} b$*, as fulfilling paths are infinite sequences, every atom participating to a fulfilling path must have a successor;*

So we pass from S^i to S^{i+1} by deleting an a as above. More formally, $S^{i+1} = (A^{i+1}, R^{i+1}_{\bigcirc}, R^{i+1}_{\odot})$ where $A^{i+1} = A^i \setminus \{a\}$, $R^{i+1}_{\bigcirc} = R^i_{\bigcirc} \cap (A^{i+1} \times A^{i+1})$ and $R^{i+1}_{\odot} = R^i_{\odot} \cap (A^{i+1} \times A^{i+1})$. We have the following lemma:

Lemma 10 *There exists a fulfilling path* π *for* β *under* γ *in* S^{i+1} *iff there exists one in* S^i

Proof.(sketch) We must show that no atom that can participate in a fulfilling path is deleted. If we make the hypothesis that an atom a from a fulfilling path is deleted for one of the four reasons above, we can derive a contradiction.

[1] $(R^i_{\bigcirc})^*$ denotes the reflexo-transitive closure of R^i.

□

The procedure described above stops when there are no more atoms to delete. Let us note S^k the final structure. We have the following lemma:

Lemma 11 *$\gamma \models \neg\beta$ iff there is no $a \in A^k$ such that $\beta \in A^k$.*

Proof. If there is a fulfilling path then by definition it contains an atom a_i such that $\beta \in a_i$, take this a_i. For the other direction, we proceed as follows to construct a fulfilling path:

1. take one a such that $\beta \in a$, $a_i = a$;
2. take $a_0 = R^k_{\odot}(a)$, this atoms belongs to S^k and from a_0 there exists a path in S^k that reaches a otherwise a would have been deleted;
3. from a we can construct an infinite suffix as $R^k_{\bigcirc}$ is total in S^k (otherwise other deletions would have been possible).

It is easy to show that the constructed path fulfills β under γ and thus can be used to construct a model.

□

We are now equipped to prove the completeness of the proposed proof system. Assume that $\gamma \models \beta$. Applying the procedure above gives us S^k with for all $a \in A^k$, $\neg\beta \in a$. We establish the completeness of the proposed proof system by proving the following lemmas:

Lemma 12 $\vdash \vee_{a \in A^0} \hat{a}$ [2]

Proof. A^0 is the set of maximally propositionally consistent subsets of C. So the lemma is obtained by propositional completeness.

□

Lemma 13 *For all $a \in A^0$, $\vdash \hat{a} \rightarrow \bigcirc \vee_{b \in R^0_{\bigcirc}(a)} \hat{b}$.*

Proof. The successors of a contain all possible combinations of formulae ψ where $\bigcirc\psi \notin C$. They can thus be eliminated, using distribution of $\vee, \wedge$, giving $\vee_{b \in R^0_{\bigcirc}(a)} \hat{b} = \bigwedge_{\bigcirc\phi \in a} \phi$

□

Lemma 14 *For all $a \in A^0$, $\vdash \hat{a} \rightarrow \odot \hat{R}^0_{\odot}(a)$*

Proof. By propositional completeness.

[2] $\hat{a}$ denotes the conjunction of the formulas that belong to the atom a.

$\square$

As a consequence of the three previous lemmas and monotonicity of $\vdash$, we have:

Corollary 1 *$\gamma \vdash \vee_{a \in A^0} \hat{a}$ and $\gamma \vdash \bigcirc \vee_{b \in R^0_{\bigcirc}(a)} \hat{b}$, $\gamma \vdash \hat{a} \to \odot \hat{R}^0_{\odot}(a)$.*

Now we prove that the above lemmas are still valid in the structures S^i by an inductive reasoning:

Lemma 15 *For every $a \notin A^i$, $\gamma \vdash \neg\hat{a}$.*

Proof. We prove that if a is not useful in S^i, we have that: $\gamma \vdash \hat{a} \leftrightarrow \bot$.

1. $\gamma \notin a$: in this case, we have $\gamma \vdash \hat{a} \to \neg\gamma$, by a propositional reasoning and the fact that γ is a hypothesis, we derive $\gamma \vdash \neg\hat{a}$;

2. $a \notin (R^{i-1}_{\bigcirc})^*(R^{i-1}_{\odot}(a))$: Let R^* be the reflexive and transitive closure of $R^{i-1}_{\bigcirc}$. Let $R = R^*(R^{i-1}_{\odot}(a))$. We can prove $\gamma \vdash \check{R} \to \bigcirc\check{R}$, where $\check{R} = \bigvee_{a \in R} \hat{a}$. Indeed, $\gamma \vdash b \to \bigcirc\check{R}^{i-1}_{\bigcirc}(b)$ by inductive hypothesis (generalizing lemma 13). Thus $\gamma \vdash b \to \bigcirc\check{R}^*(b)$, since the closure contains the base relation, and using $NEC_{\bigcirc}, K_{\bigcirc}, MP$. We can do this for any $b \in R$, giving $\gamma \vdash \check{R} \to \bigcirc\check{R}$ using the propositional disjunction rule and (COM). Now we can use the (IND') rule to derive $\gamma \vdash \odot\check{R} \to \check{R}$. Now $\vdash \hat{a} \to \odot\hat{R}_{\odot}(a)$ by lemma 6; $\vdash \hat{R}_{\odot}(a) \to \check{R}$ since $R_{\odot}(a) \in R$; by NEC$_{\odot}$, K$_{\odot}$, MP, $\vdash \odot\hat{R}_{\odot}(a) \to \odot\check{R}$. Finally, using the induction result, we obtain $\gamma \vdash \hat{a} \to \check{R}$. Assume a is not reachable, then R is a disjunction of atoms incompatible with a, or said propositionally $\vdash R \to \neg\hat{a}$. Thus $\gamma \vdash \hat{a} \to \neg\hat{a}$, which simplifies to $\gamma \vdash \neg\hat{a}$.

3. $\neg\exists b : aR^{i-1}_{\bigcirc}b$: in this case, we have $\gamma \vdash \hat{a} \to \bigcirc\bot$ by lemma 13. Using theorem $\neg \bigcirc \bot$ and a propositional reasoning, we derive $\gamma \vdash \neg\hat{a}$.

$\square$

Using the previous lemma, we update the following lemmas:

Lemma 16 *For every $a \in A^i$, $\gamma \vdash \bigcirc \vee_{b \in R^i_{\bigcirc}(a)} \hat{b}$, $\gamma \vdash \hat{a} \to \odot\hat{R}^i_{\odot}(a)$ and $\gamma \vdash \vee_{a \in A^i} \hat{a}$.*

Corollary 2 *$\gamma \vdash \vee_{a \in A^k} \hat{a}$.*

Theorem 1 *If $\gamma \vDash \beta$ then $\gamma \vdash \beta$.*

Proof. Recall that by hypothesis there is no fulfilling path for $\neg\beta$ under γ, by lemma 11, we have that $\neg\beta \notin a$ for every $a \in A^k$. So, propositionally, we obtain $\gamma \vdash \wedge_{a \in A^k}(\hat{a} \to \beta)$ and by corollary 2, that $\gamma \vdash \beta$.

$\square$

5 A tableau system

The proof above strongly suggests a tableau procedure to decide whether $\gamma \vdash \beta$, that only constructs the part of the structure $\mathcal{A}^k$ which is really used. As usual in modal logics, we construct several tableaux, each representing a instant of time. To each tableau, we add γ, since γ has to hold at each instant.

We construct the tableaux:

1. T_0 (modelling a_n, the instant at which $\neg\beta$ is satisfied) containing initially $\neg\beta, \gamma$,

2. the initial tableau I_0 (modelling I) collecting all formulae $\{\odot\phi, \phi | \odot\phi \in T_i, I_i\}$ and γ,

3. the successor tableaux $I_{i+1} = \{\gamma, \odot\phi, \psi | \odot\phi \in I_0, \bigcirc\psi \in I_i\}$,

4. T_{i+1} similarly ($i \in 0..N-1$)

We stop after N successors, where N is 2^c, and c is the size of $\{\bigcirc\phi | \bigcirc\phi \in C\}$. We unite T_0 with some I_u. As a consequence T_j also unites with $I_{u+j}, u+j \leq N$.

We complete the tableaux propositionally while maintaining these constraints. If we fail, we know by the previous proof that $\gamma \vdash \beta$. If we succeed, we will have built a model of $\neg\beta, \gamma$.

To build a theoretically optimal procedure for entailment, we start by guessing $u \leq N$, which takes a nondeterministic time and space $logN = c$. Then we guess the initial tableau, in nondeterministic time and space n. We check propositional consistency with γ. Now we compute $I_i (1 \leq i < u)$ by transferring formulae of the form $\bigcirc\phi$ to the next tableau, guessing the missing formulae until we reach the tableau u. There we also check consistency with $\neg\beta$. We go on transferring formulae and checking consistency until we reach tableau T_N. Then a loop has been created, that allows to build an infinite model. The procedure below uses exponential nondeterministic time and linear space. It is thus in PSPACE.

To build a theoretically optimal procedure for expression entailment, that is when γ is constant, we construct a structure B^k similar to S^k above, but built only on the subformulae of γ. It has a constant size bounded by $2^{|\gamma|}$. We compute its transition relation, and its reflexo-transitive closure. Again, we first conjecture about the time u at which $\neg\beta$ will hold. Either:

- $u \leq n$: We guess u. We guess the tableaux $I_0, I_1, \ldots I_u$ and check that their projection in the structure B^k exists. For I_u we check also consistency with $\neg\beta$. We proceed constructing $I_u = T_0, T_1, \ldots, T_n$ similarly. At this point we know that $\neg\beta$ indeed holds. We also know that the path can be continued infinitely, since each atom of B^k has a successor.

- $u > n$ Then the value of u is of no interest. We start guessing the tableaux $I_0, I_1, \ldots I_n$ and check that their projection in the structure B^k exists. Then we choose a follower of the projection of I_n by the transitive closure. We extend this atom to ensure that $\neg\beta$ holds, giving T_0. We build $T_1, \ldots, T_n$ as above. Again we do not have to check further because we are in B^k.

This procedure has to conjecture $2n$ tableaux, each of size n, and thus proceeds in quadratic nondeterministic time. It is thus in NP.

In practice:

1. It is more efficient to move propositional connectives outside using (COM). This also allows to use known optimisations on propositional tableaux, such as clausal tableaux.

2. It is more efficient to simplify β, γ using the (con) axioms before starting. These axioms will no longer be needed during the course of the procedure.

3. As all $\odot\phi$ must be the same everywhere, and $\odot\phi \leftrightarrow \phi$ in the initial tableau I_0, it is more efficient to keep only ϕ in I_0 and import $\odot I_0$ implicitly everywhere.

4. Similarly, modal formulae can be put directly in the right tableau: e.g. instead of putting $\odot \bigcirc^i p$ in a tableau, we can put p into I_i; similarly instead of putting $\bigcirc^j p$ into T_i, we can put p into T_{i+j}. In this case we have to use m supplementary successors, where n is the nesting level of $\bigcirc$ in β, γ, but we only need to check consistency on proposition symbols for these last m tableaux (there is no need to check that they satisfy γ).

5. It is often useful to check for loops in the tableaux constructed. This sometimes allows to use less than N tableaux.

6. It is often useful to develop T_0, then the I_i in increasing order to choose the right I_u to unite with T_0, then develop the T_i.

6 Complexity

6.1 Complexity of validity

First note that the complexity of the validity problem, deciding whether $\vdash \beta$, is co-NP-hard, because our logic contains propositional logic whose validity problem is co-NP-complete.

However, we decide to solve the more general problem of *expression entailment* defined by [18]: here, we assume a constant antecedent γ, which needs not be $\top$.

6.2 Complexity of entailment

The problem of entailment is more complex, as expected from the fact that it can represent interesting programming problems:

Lemma 17 *Deciding the satisfiability is PSPACE-hard.*

Proof. We can encode a deterministic PSPACE Turing machine in this satisfiability problem. Let $M = (Q, \delta, q_0, F)$ be a Turing machine that only uses $P_1(n)$ cells of tape, where n is the length of its inputs, and P_1 is some polynomial. The components of the machine are:

1. Q is the set of control states;
2. $\delta : Q \times \{0,1\} \to Q \times \{0,1\} \times \{L, R\}$ is the transition function: given the current state and the current reading of the tape, it gives the new state, the new contents of the tape, and the move of the head.
3. q_0 is the initial state;
4. F is the set of final states (the answer is then on the tape).

We construct a formula γ describing the rule of evolution of the machine. We encode each control state $q \in Q$ by a proposition q; the contents of each cell of the tape by a proposition $c_i (i \leq P_1(n))$; and h_i tells whether the head is in front of cell i. Clearly, we have the property that there is a single control: $\neg(q_i \wedge q_j)(q_i \neq q_j, q_i, q_j \in Q)$ and a single head position: $\neg(h_i \wedge h_j)(h_i \neq h_j, i, j \leq S(n))$ The evolution rule states exactly the evolution of a Turing machine: Given current control at q and content v, the control goes to a new state given by the first component of δ, the new content of the cell is given by the second component, the head moves according to the third component of delta, for which we define: $s(q, v, i) = i + 1 if \pi_3(\delta(q, v)) = R; s(q, v, i) = i - 1 if \pi_3(\delta(q, v)) = L$. The other cells, which are not under the head, are unchanged.

$$\bigwedge_{i \leq P_1(n)} \bigwedge_{q \in Q} \bigwedge_{v \in \{0,1\}} (q \wedge h_i \wedge (c_i \leftrightarrow v)) \to \begin{array}{l} \wedge \bigcirc \pi_1 \delta(q, v) \\ \wedge \bigcirc c_i \leftrightarrow \pi_2(\delta(q, v)) \\ \wedge \bigcirc h_{s(q,v,i)} \\ \wedge \bigwedge_{j \neq i} (\bigcirc c_j \leftrightarrow c_j) \end{array}$$

The size of this formula is dominated by the last part, of size $P_1^2(n)$. The initial content of the tape, X, is coded by a formula $\odot \bigwedge_{j \leq n} c_i \leftrightarrow X_i$.

The machine is deterministic, and thus the formula above has at most one model. It accepts X iff it terminates, that is if $\bigvee_{q \in F} q$ is true at some instant. Thus, we define γ as the conjunction of all formulae above and $\bigwedge_{q \in F} \neg q$. M accepts X iff γ is not satisfiable, thus the problem is co-PSPACE-hard, which is the same as PSPACE-hard.

□

A similar problem is the *data entailment* defined by [18]: here, we assume a constant consequent β. This problem thus generalises the problem of co-satisfiability.

Theorem 2 *The problems of entailment, data entailment and satisfiability are PSPACE-complete.*

Proof.

1. They are PSPACE-easy: we can use the tableau procedure above.
2. They are PSPACE-hard: even the easiest problem, satisfiability, is already PSPACE-hard.

□

References

[1] M. Abadi. *Temporal-Logic Theorem Proving.* PhD thesis, Stanford University, 1987.

[2] Martín Abadi. Corrigendum: The power of temporal proofs. *Theoretical Computer Science*, 70(2):275, 26 January 1990.

[3] H. Andréka, V. Goranko, S. Mikulás, I. Németi, and I. Sain. Effective temporal logics of programs. In L. Bolc and A. Szałas, editors, *Time and logic: a computational approach*, chapter 2, pages 51–130. UCL Press, 1995.

[4] H. Barringer. The use of temporal logic in the compositional specification of concurrent systems. In A. Galton, editor, *Temporal Logics and their Applications*, pages 53–90. Academic Press, 1987.

[5] H. Barringer, R. Kuiper, and A. Pnueli. A really abstract concurrent model and its temporal logic. In *Proceedings of the 13th Annual Symposium on Principles of Programming Languages*, pages 173–183. ACM Press, 1986.

[6] E.A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. Elsevier Science Publishers, 1990.

[7] D. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic.* Clarendon Press, Oxford, 1994.

[8] E. Pascal Gribomont. A programming logic for formal concurrent systems. In J. C. M. Baeten and J. W. Klop, editors, *CONCUR '90: Theories of Concurrency: Unification and Extension*, volume 458 of *Lecture Notes in Computer Science*, pages 298–313, Amsterdam, The Netherlands, 27–30August 1990. Springer.

[9] O. Lichtenstein, A. Pnueli, and L.D. Zuck. The glory of the past. In R. Parikh, editor, *Logics of Programs*, Lecture Notes in Computer Science 193, pages 196–218. Springer-Verlag, 1985.

[10] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems-Specification*. Springer, 1992.

[11] Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems: Safety*. Springer, New York, 1995.

[12] A. Pnueli. The temporal logic of programs. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, pages 46–57. IEEE Computer Society Press, 1977.

[13] A. N. Prior. *Time and Modality*. Oxford University Press, Oxford, UK, 1957.

[14] A. N. Prior. *Papers on time and tense*. Oxford UP, 1968.

[15] J.-F. Raskin, P.-Y. Schobbens, and T. A. Henzinger. Axioms for real-time logics. In *Proceedings of CONCUR'98, LNCS*. Springer Verlag, 1999.

[16] A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logic. *Journal of the ACM*, 32(3):733–749, July 1985.

[17] A. Szalas. A complete axiomatic characterization of first-order temporal logic of linear time. *Theoretical Computer Science*, 54(2-3):199–214, October 1987.

[18] Moshe Y. Vardi. The complexity of relational query languages. In *ACM STOC'82*, pages 137–146, Baltimore, USA, May 1982. ACM Press.

INFORMATIK

Below you find a list of the most recent technical reports of the research group *Logic of Programming* at the Max-Planck-Institut für Informatik. They are available by anonymous ftp from our ftp server `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Birgit Hofmann
Im Stadtwald
D-66123 Saarbrücken
GERMANY
e-mail: `library@mpi-sb.mpg.de`

MPI-I-98-2-011	A. Degtyarev, A. Voronkov	Equality Reasoning in Sequent-Based Calculi
MPI-I-98-2-010	S. Ramangalahy	Strategies for Conformance Testing
MPI-I-98-2-009	S. Vorobyov	The Undecidability of the First-Order Theories of One Step Rewriting in Linear Canonical Systems
MPI-I-98-2-008	S. Vorobyov	AE-Equational theory of context unification is Co-RE-Hard
MPI-I-98-2-007	S. Vorobyov	The Most Nonelementary Theory (A Direct Lower Bound Proof)
MPI-I-98-2-006	P. Blackburn, M. Tzakova	Hybrid Languages and Temporal Logic
MPI-I-98-2-005	M. Veanes	The Relation Between Second-Order Unification and Simultaneous Rigid E-Unification
MPI-I-98-2-004	S. Vorobyov	Satisfiability of Functional+Record Subtype Constraints is NP-Hard
MPI-I-98-2-003	R.A. Schmidt	E-Unification for Subsystems of S4
MPI-I-97-2-012	L. Bachmair, H. Ganzinger, A. Voronkov	Elimination of Equality via Transformation with Ordering Constraints
MPI-I-97-2-011	L. Bachmair, H. Ganzinger	Strict Basic Superposition and Chaining
MPI-I-97-2-010	S. Vorobyov, A. Voronkov	Complexity of Nonrecursive Logic Programs with Complex Values
MPI-I-97-2-009	A. Bockmayr, F. Eisenbrand	On the Chvátal Rank of Polytopes in the 0/1 Cube
MPI-I-97-2-008	A. Bockmayr, T. Kasper	A Unifying Framework for Integer and Finite Domain Constraint Programming
MPI-I-97-2-007	P. Blackburn, M. Tzakova	Two Hybrid Logics
MPI-I-97-2-006	S. Vorobyov	Third-order matching in $\lambda \rightarrow$-Curry is undecidable
MPI-I-97-2-005	L. Bachmair, H. Ganzinger	A Theory of Resolution
MPI-I-97-2-004	W. Charatonik, A. Podelski	Solving set constraints for greatest models
MPI-I-97-2-003	U. Hustadt, R.A. Schmidt	On evaluating decision procedures for modal logic
MPI-I-97-2-002	R.A. Schmidt	Resolution is a decision procedure for many propositional modal logics