Generalized Model Checking: Reasoning about Partial State Spaces*

Glenn Bruns Patrice Godefroid

Bell Laboratories, Lucent Technologies
263 Shuman Boulevard, Naperville, IL 60566, U.S.A.
{grb,god}@bell-labs.com

Abstract. We discuss the problem of model checking temporal properties on partial Kripke structures, which were used in [BG99] to represent incomplete state spaces. We first extend the results of [BG99] by showing that the model-checking problem for any 3-valued temporal logic can be reduced to two model-checking problems for the corresponding 2-valued temporal logic. We then introduce a new semantics for 3-valued temporal logics that can give more definite answers than the previous one. With this semantics, the evaluation of a formula ϕ on a partial Kripke structure M returns the third truth value $\perp$ (read "unknown") only if there exist Kripke structures M_1 and M_2 that both complete M and such that M_1 satisfies ϕ while M_2 violates ϕ, hence making the value of ϕ on M truly unknown. The partial Kripke structure M can thus be viewed as a partial solution to the satisfiability problem which reduces the solution space to complete Kripke structures that are more complete than M with respect to a completeness preorder. This *generalized model-checking problem* is thus a generalization of both satisfiability (all Kripke structures are potential solutions) and model checking (a single Kripke structure needs to be checked). We present algorithms and complexity bounds for the generalized model-checking problem for various temporal logics.

1 Introduction

Many approaches have been proposed to deal with the problem of model checking large state spaces, among them partial-order methods, symbolic verification, and abstraction techniques. But often these approaches do not suffice. Lacking a means to model check the entire state space of a system, one may settle for considering only part of the state space, hoping errors can be found there. The selection of a part of the state space can be done in various ways: randomly, breadth-first, according to general heuristics (i.e. prefer states in which process queues are filling), or according to what could be called "ad-hoc abstraction", in which one ignores certain states, or certain details of states, believed irrelevant to the property of interest.

* This is an extended abstract, with proofs omitted. For the full version of the paper see www.bell-labs.com/~{grb,god}

The result of applying traditional model checking to such a "partial state space" will show that a property does or does not hold. However, one would like the partiality of the state space to be taken into account. Thus, a search should return *true* if the property definitely holds because of information in the partial state space, *false* if the property definitely fails to hold because of information in the partial state space, and *unknown* (denoted $\perp$) if the truth or falsity of the property depends on information not contained in the partial state space.

In [BG99] *partial Kripke structures* were used to represent partial state spaces. A *completeness preorder* was defined for partial Kripke structures, a 3-valued temporal logic for reasoning about such structures was defined, and this logic was shown to characterize the completeness preorder. A corollary of the characterization result is that interpreting a formula on a more complete Kripke structure will give a more definite result.

In this paper, we first extend the results of [BG99] by showing that the model-checking problem for 3-valued temporal logic can be reduced to *two* model-checking problems for the corresponding 2-valued temporal logic. Specifically, we show that any partial Kripke structure can be completed into two "extreme" complete Kripke structures, called the *optimistic* and *pessimistic* completions, and that model-checking a partial Kripke structure can be reduced to model-checking its optimistic and pessimistic completions. This implies that the problem of model-checking partial state spaces can be solved using existing tools.

The 3-valued semantics of [BG99] does not behave exactly as one might expect. Informally, one would like the model checking algorithm to return value $\perp$ for a partial Kripke structure and a formula only if there exist a more complete structure for which the formula is *true* and another more complete structure for which the formula is *false*. However, by the 3-valued semantics of [BG99], it is possible to get result $\perp$ even though only results $\perp$ and *true* can be obtained when making the given structure more complete.

We introduce in this paper a new semantics that gives value $\perp$ only when there exist a more complete structure for which the formula holds *and* a more complete structure for which the formula does not hold. The main question we consider is whether model checking under this semantics is more expensive than model checking under the semantics given in [BG99]. We give algorithms and complexity bounds for several temporal logics showing that it is indeed more expensive. The problem of model checking under the new semantics is interesting on its own because it generalizes the problems of model checking and of satisfiability checking. Solving the problem means determining if some structure more complete than a given structure satisfies a formula. If the given structure is fully incomplete, the problem reduces to satisfiability checking. If the given structure is fully complete, the problem reduces to model checking.

2 Two-valued Modal Logics

In this section we briefly review Kripke structures and propositional modal logic (PML). Let P be a nonempty finite set of *atomic propositions*.

Definition 1. A *Kripke structure* M is a tuple $(S, L, \mathcal{R})$, where S is a set of *states*, $L : S \times P \to \{true, false\}$ is an *interpretation* that associates a truth value in $\{true, false\}$ with each atomic proposition in P for each state in S, and $\mathcal{R} \subseteq S \times S$ is a transition relation on S.

For technical convenience we require of every Kripke structure that its transition relation be *total*, i.e., that every state has an outgoing transition. We also assume that the number of outgoing transitions from a state is finite. We write (M, s) to refer to state s of Kripke structure M, or just s if the structure to which s belongs is clear. Also, we write $s \to s'$ as shorthand for $(s, s') \in \mathcal{R}$.

PML (e.g, see [Var97b]) is propositional logic extended with the modal operator $\Box$. Intuitively, formula $\Box\,\phi$ holds at a state s if ϕ holds at all states that can be reached from s in a single transition. Formulas of PML have the following abstract syntax, where p ranges over P:

$$\phi ::= p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \Box\,\phi$$

Definition 2. The *satisfaction* of a formula ϕ of PML in a state s of a Kripke structure $M = (S, L, \mathcal{R})$, written $(M, s) \models \phi$, is defined inductively as follows:

$$\begin{array}{ll} (M,s) \models p & \text{if } L(s,p) = \mathit{true} \\ (M,s) \models \neg\phi & \text{if } (M,s) \not\models \phi \\ (M,s) \models \phi_1 \wedge \phi_2 & \text{if } (M,s) \models \phi_1 \text{ and } (M,s) \models \phi_2 \\ (M,s) \models \Box\,\phi & \text{if } (M,s') \models \phi \text{ for all } s' \text{ such that } s \to s' \end{array}$$

We define $\phi_1 \vee \phi_2$ as $\neg(\neg\phi_1 \wedge \neg\phi_2)$ and $\Diamond\,\phi$ as $\neg(\Box\,\neg\phi)$.

PML can be used to define an equivalence relation on states of Kripke structures: two states are equivalent if they satisfy the same set of formulas of the logic. It is well known [HM85] that the equivalence relation induced in this way by PML coincides with the notion of bisimulation relation [Mil89, Par81].

Definition 3. Let $M_1 = (S_1, L_1, \mathcal{R}_1)$ and $M_2 = (S_2, L_2, \mathcal{R}_2)$ be Kripke structures. The *bisimilarity* relation $\sim$ is the greatest relation $\mathcal{B} \subseteq S_1 \times S_2$ such that $(s_1, s_2) \in \mathcal{B}$ implies the following:

- $\forall p \in P : L(s_1, p) = L(s_2, p)$,
- if $s_1 \to s_1'$ then there is some $s_2' \in S_2$ such that $s_2 \to s_2'$ and $(s_1', s_2') \in \mathcal{B}$, and
- if $s_2 \to s_2'$ then there is some $s_1' \in S_1$ such that $s_1 \to s_1'$ and $(s_1', s_2') \in \mathcal{B}$.

The following result (from [HM85]) shows that PML *logical characterizes* the bisimulation preorder: two states are bisimilar just if they satisfy the same set of PML formulas.

Theorem 4. *[HM85] Let $M_1 = (S_1, L_1, \mathcal{R}_1)$ and $M_2 = (S_2, L_2, \mathcal{R}_2)$ be Kripke structures such that $s_1 \in S_1$ and $s_2 \in S_2$, and let Φ denote the set of all PML formulas. Then*

$$s_1 \sim s_2 \textit{ iff } (\forall \phi \in \Phi : [(M_1, s_1) \models \phi] = [(M_2, s_2) \models \phi]).$$

3 Partial Kripke Structures and Three-valued Modal Logic

3.1 Background

In this section we present background information on partial Kripke structures and a 3-valued modal logic interpreted over them. The definitions and results of this section come from [BG99].

We model partial state spaces as partial Kripke structures, in which propositions can take a third truth value $\perp$. We then define a 3-valued modal logic whose semantics is defined with respect to partial Kripke structures. We proceed by presenting an equivalence relation and preorder implicitly defined by this logic. As before, let P be a nonempty finite set of atomic propositions.

Definition 5. A *partial Kripke structure* M is a tuple $(S, L, \mathcal{R})$, where S is a set of *states*, $L : S \times P \to \{true, \perp, false\}$ is an *interpretation* that associates a truth value in $\{true, \perp, false\}$ with each atomic proposition in P for each state in S, and $\mathcal{R} \subseteq S \times S$ is a transition relation on S.

A standard Kripke structure is a special case of partial Kripke structure. We sometimes refer to standard Kripke structures as *complete* Kripke structures to emphasize that no propositions within them take value $\perp$.

In interpreting propositional operators on partial Kripke structures we use Kleene's strong 3-valued propositional logic [Kle87]. In this logic $\perp$ is understood as "unknown whether true or false". A simple way to define conjunction (resp. disjunction) in this logic is as the minimum (resp. maximum) of its arguments under the *truth ordering* on truth values, in which *false* is less than $\perp$ and $\perp$ is less than *true*. We write 'min' and 'max' for these functions, and extend them to sets in the obvious way, with $\min(\emptyset) = true$ and $\max(\emptyset) = false$. We define negation using the function 'comp' that maps *true* to *false*, *false* to *true*, and $\perp$ to $\perp$. These functions give the usual meaning of the propositional operators when applied to values *true* and *false*.

Definition 6. The value of a formula ϕ of 3-valued PML in a state s of a partial Kripke structure $M = (S, L, \mathcal{R})$, written $[(M, s) \models \phi]$, is defined inductively as follows:

$$\begin{aligned}
[(M,s) \models p] &= L(s,p) \\
[(M,s) \models \neg\phi] &= \mathrm{comp}([(M,s) \models \phi]) \\
[(M,s) \models \phi_1 \wedge \phi_2] &= \min([(M,s) \models \phi_1], [(M,s) \models \phi_2]) \\
[(M,s) \models \Box\,\phi] &= \min(\{[(M,s') \models \phi] \mid s \to s'\})
\end{aligned}$$

This semantics generalizes the 2-valued semantics for PML.

This 3-valued logic can be used to define a preorder on partial Kripke structures that reflects their degree of completeness. Let $\leq$ be the *information ordering* on truth values, in which $\perp \leq true$, $\perp \leq false$, $x \leq x$ (for all $x \in \{true, \perp$

, *false*}), and $x \not\leq y$ otherwise. The operators comp, min and max preserve information: if $x \leq x'$ and $y \leq y'$, we have $\mathrm{comp}(x) \leq \mathrm{comp}(x')$, $\min(x, y) \leq \min(x', y')$, and $\max(x, y) \leq \max(x', y')$.

Definition 7. Let $M_1 = (S_1, L_1, \mathcal{R}_1)$ and $M_2 = (S_2, L_2, \mathcal{R}_2)$ be partial Kripke structures. The *completeness preorder* $\preceq$ is the greatest relation $\mathcal{B} \subseteq S_1 \times S_2$ such that $(s_1, s_2) \in \mathcal{B}$ implies the following:

- $\forall p \in P : L(s_1, p) \leq L(s_2, p)$,
- if $s_1 \rightarrow s_1'$ then there is some $s_2' \in S_2$ such that $s_2 \rightarrow s_2'$ and $(s_1', s_2') \in \mathcal{B}$, and
- if $s_2 \rightarrow s_2'$ then there is some $s_1' \in S_1$ such that $s_1 \rightarrow s_1'$ and $(s_1', s_2') \in \mathcal{B}$.

Intuitively, $s_1 \preceq s_2$ means that s_1 and s_2 are "nearly bisimilar" except that the atomic propositions in state s_1 may be less defined than in state s_2. Obviously, $s_1 \sim s_2$ implies $s_1 \preceq s_2$. Also, any partial Kripke structure can be completed to obtain a complete Kripke structure.

The following result shows that 3-valued PML logically characterizes the completeness preorder.

Theorem 8. *[BG99] Let $M_1 = (S_1, L_1, \mathcal{R}_1)$ and $M_2 = (S_2, L_2, \mathcal{R}_2)$ be partial Kripke structures such that $s_1 \in S_1$ and $s_2 \in S_2$, and let Φ be the set of all formulas of 3-valued PML. Then*

$$s_1 \preceq s_2 \text{ iff } (\forall \phi \in \Phi : [(M_1, s_1) \models \phi] \leq [(M_2, s_2) \models \phi]).$$

In other words, partial Kripke structures that are "more complete" with respect to $\preceq$ have more definite properties with respect to $\leq$, i.e., have more properties that are either *true* or *false*. Moreover, any formula ϕ of 3-valued PML that evaluates to *true* or *false* on a partial Kripke structure has the same truth value when evaluated on any more complete structure.

Results similar to those of this section were presented also for extended transition systems in [BG99]. Extended transition systems are labelled transition systems with a divergence predicate on states (e.g., see [Wal88, Sti87]). A connection made in [BG99] between 3-valued and 2-valued modal logics on extended transition systems partly inspired the following new results.

3.2 Positive PML

In the following sections we shall use a positive form of PML, which we refer to as PML$^+$. Here we define 2 and 3-valued semantics for PML$^+$ and observe that every formula of PML can be expressed in PML$^+$. The abstract syntax of PML$^+$ is as follows, where p ranges over P:

$$\phi ::= p \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \Box \phi \mid \Diamond \phi$$

Definition 9. The value of a formula ϕ of 3-valued PML$^+$ in a state s of a partial Kripke structure $M = (S, L, \mathcal{R})$, written $[(M,s) \models^+ \phi]$, is defined inductively as follows:

$$\begin{aligned}
[(M,s) \models^+ p] &= L(s,p) \\
[(M,s) \models^+ \phi_1 \wedge \phi_2] &= \min([(M,s) \models^+ \phi_1], [(M,s) \models^+ \phi_2]) \\
[(M,s) \models^+ \phi_1 \vee \phi_2] &= \max([(M,s) \models^+ \phi_1], [(M,s) \models^+ \phi_2]) \\
[(M,s) \models^+ \Box \phi] &= \min(\{[(M,s') \models^+ \phi] \mid s \rightarrow s'\}) \\
[(M,s) \models^+ \Diamond \phi] &= \max(\{[(M,s') \models^+ \phi] \mid s \rightarrow s'\})
\end{aligned}$$

We define 2-valued PML$^+$ using this 3-valued interpretation as follows: a PML$^+$ formula ϕ holds at a state s of a complete Kripke structure M, written $(M,s) \models^+ \phi$, just if $[(M,s) \models^+ \phi] = \mathit{true}$.

We can translate every formula of PML to an equivalent formula of PML$^+$ if we consider only the partial Kripke structures $M = (S, L, \mathcal{R})$ in which, for every $p \in P$ there exists a $q \in P$ such that $L(s,p) = \mathrm{comp}(L(s,q))$ for all s in S. In such structures we refer to a proposition that is complementary to a proposition p as $\overline{p}$. We refer to such a partial Kripke structure as a *complement-closed* structure. Our translation T from PML to PML$^+$ is then as follows: $T(p) = p$, $T(\neg p) = \overline{p}$, $T(\neg(\phi_1 \wedge \phi_2)) = T(\neg\phi_1) \vee T(\neg\phi_2)$, $T(\neg(\Box \phi)) = \Diamond(T(\neg\phi))$, $T(\neg\neg\phi) = T(\phi)$, $T(\phi_1 \wedge \phi_2) = T(\phi_1) \wedge T(\phi_2)$, and $T(\Box \phi) = \Box(T(\phi))$.

Proposition 10. *Let M be a partial Kripke structure that is complement-closed and ϕ be a PML formula. Then*

$$[(M,s) \models \phi] = [(M,s) \models^+ T(\phi)].$$

3.3 Model Checking 3-Valued Modal Logics

In this section we show that model checking 3-valued modal logic is no more expensive than model checking standard modal logic, and can be performed using existing model checkers.

From a 3-valued labelling L of a partial Kripke structure we can derive a pair of 2-valued labellings, one of which treats $\perp$ as *true*, while the other treats $\perp$ as *false*.

Definition 11. Given a 3-valued labelling function L, we define the derived *optimistic* labelling function L_o and *pessimistic* labelling function L_p as follows:

$$L_o(s,p) \stackrel{\text{def}}{=} \begin{cases} \mathit{true} & \text{if } L(s,p) = \perp \\ L(s,p) & \text{otherwise} \end{cases}$$

$$L_p(s,p) \stackrel{\text{def}}{=} \begin{cases} \mathit{false} & \text{if } L(s,p) = \perp \\ L(s,p) & \text{otherwise} \end{cases}$$

Given a partial Kripke structure $M = (S, L, \mathcal{R})$ we write $M_p = (S, L_p, \mathcal{R})$ for the derived pessimistic structure and $M_o = (S, L_o, \mathcal{R})$ for the derived optimistic structure.

The 3-valued interpretation of a PML$^+$ formula at a state s in a partial Kripke structure can be computed from the classical 2-valued interpretations of the formula using the optimistic and pessimistic structures. The formula is *true* at s if it is *true* under the pessimistic interpretation, is *false* at s if it is *false* under the optimistic interpretation, and is $\perp$ otherwise.

Theorem 12. *Let $M = (S, L, \mathcal{R})$ be a partial Kripke structure with s in S, let M_p and M_o be the derived pessimistic and optimistic structures, and let ϕ be a formula of PML$^+$. Then*

$$[(M,s) \models^+ \phi] \stackrel{\text{def}}{=} \begin{cases} \textit{true} \;\; \textit{if}\ (M_p, s) \models^+ \phi \\ \textit{false}\ \textit{if}\ (M_o, s) \not\models^+ \phi \\ \perp \qquad \textit{otherwise} \end{cases}$$

Thus, one can do 3-valued model checking by running a standard 2-valued model checker at most twice, once with the partial Kripke structure transformed to a complete, optimistic Kripke structure, and once with the partial Kripke structure transformed to a complete, pessimistic Kripke structure. These transformations are linear with respect to the size of the structure, so 3-valued model checking for PML has the same time and space complexity as the 2-valued case.

3.4 Adding Fixed-Point Operators

PML can be extended with a fixed-point operator to form a modal fixpoint logic, also referred to as the propositional μ-calculus [Koz83]. This very expressive logic includes as fragments linear-time temporal logic (LTL) [MP92] and computation-tree logic (CTL) [CE81]. In this section we extend PML$^+$ with fixed-point operators and show that model checking for this extended logic can also be reduced to standard 2-valued model checking.

PML$^+$ extended with fixed-point operators has the following abstract syntax, where p ranges over the set P of atomic propositions and X ranges over a set Var of fixed-point variables:

$$\phi ::= p \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \Box\,\phi \mid \Diamond\,\phi \mid X \mid \nu X.\phi \mid \mu X.\phi$$

In fixed-point formulas $\nu X.\phi$ and $\mu X.\phi$ the operators ν and μ bind free occurrences of X in ϕ. We refer to this version of the modal mu-calculus [Koz83] as μL.

We now define a 2-valued semantics of μL. For the case of the fixed-point operator it makes sense to use a semantics that interprets a formula as the set of states for which the formula holds. We can derive such a set-valued semantics for PML$^+$ from the semantics given in Section 3.2 as follows:

$$\|M, \phi\| \stackrel{\text{def}}{=} \{s \in S \mid (M,s) \models^+ \phi\}$$

To interpret μL formulas we need not only a labelling to interpret atomic propositions but also a *valuation* to interpret fixed-point variables. A valuation $\mathcal{V}$

maps a fixed-point variable to a set $\mathcal{V}(X)$ of states. Thus μL formulas are interpreted relative to a Kripke structure M and a valuation $\mathcal{V}$. The new semantic clauses for fixed-point variables and formulas are as follows: $\|M, X\|_{\mathcal{V}} \stackrel{\text{def}}{=} \mathcal{V}(X)$, $\|M, \nu X.\phi\|_{\mathcal{V}} \stackrel{\text{def}}{=} \nu f$, and $\|M, \mu X.\phi\|_{\mathcal{V}} \stackrel{\text{def}}{=} \mu f$, where νf (μf) denotes the greatest (least) fixed-point of function $f : S \rightarrow S$, defined as

$$f(A) \stackrel{\text{def}}{=} \|M, \phi\|_{\mathcal{V}[X:=A]}$$

Here $\mathcal{V}[X := A]$ stands for the valuation that is like $\mathcal{V}$ except that X is mapped to set A. Function f is a monotonic, set-valued function, so by the Knaster-Tarski theorem [Tar55] we know it has a greatest fixed point, namely $\bigcup\{A \subseteq S \mid A \subseteq f(A)\}$, where S is the set of states in Kripke structure M. Similarly, its least fixed point is $\bigcap\{A \subseteq S \mid f(A) \subseteq A\}$.

Now consider a 3-valued interpretation of μL. Here a formula ϕ is interpreted as a pair (S_1, S_2) of disjoint sets of states, where S_1 is the set for which ϕ is known to be true and S_2 is the set for which ϕ is known to be false. A valuation function $\mathcal{V}$ now maps a fixed-point variable to a pair of disjoint sets of states. Given a pair (S_1, S_2) of sets we write $\pi_1(S_1, S_2)$ for the first set and $\pi_2(S_1, S_2)$ for the second.

Definition 13. The 3-valued interpretation $[M, \phi]_{\mathcal{V}}$ of a μL formula relative to a partial Kripke structure $M = (S, L, \mathcal{R})$ is defined as follows:

$$\begin{aligned}
[M, p]_{\mathcal{V}} &\stackrel{\text{def}}{=} (\{s \in S \mid L(s,p) = \mathit{true}\}, \{s \in S \mid L(s,p) = \mathit{false}\}) \\
[M, \phi_1 \wedge \phi_2]_{\mathcal{V}} &\stackrel{\text{def}}{=} (\pi_1([M,\phi_1]_{\mathcal{V}}) \cap \pi_1([M,\phi_2]_{\mathcal{V}}), \pi_2([M,\phi_1]_{\mathcal{V}}) \cup \pi_2([M,\phi_2]_{\mathcal{V}})) \\
[M, \phi_1 \vee \phi_2]_{\mathcal{V}} &\stackrel{\text{def}}{=} (\pi_1([M,\phi_1]_{\mathcal{V}}) \cup \pi_1([M,\phi_2]_{\mathcal{V}}), \pi_2([M,\phi_1]_{\mathcal{V}}) \cap \pi_2([M,\phi_2]_{\mathcal{V}})) \\
[M, \Box\, \phi]_{\mathcal{V}} &\stackrel{\text{def}}{=} (\{s \mid \forall s'.s \rightarrow s' \Rightarrow s' \in \pi_1([M,\phi]_{\mathcal{V}})\}, \\
&\qquad \{s \mid \exists s'.s \rightarrow s' \wedge s' \in \pi_2([M,\phi]_{\mathcal{V}})\}) \\
[M, \Diamond\, \phi]_{\mathcal{V}} &\stackrel{\text{def}}{=} (\{s \mid \exists s'.s \rightarrow s' \wedge s' \in \pi_1([M,\phi]_{\mathcal{V}})\}, \\
&\qquad \{s \mid \forall s'.s \rightarrow s' \Rightarrow s' \in \pi_2([M,\phi]_{\mathcal{V}})\}) \\
[M, X]_{\mathcal{V}} &\stackrel{\text{def}}{=} \mathcal{V}(X) \\
[M, \nu X.\phi]_{\mathcal{V}} &\stackrel{\text{def}}{=} \nu f \\
[M, \mu X.\phi]_{\mathcal{V}} &\stackrel{\text{def}}{=} \mu f
\end{aligned}$$

In the fixed-point clauses $f(S_1, S_2) \stackrel{\text{def}}{=} [M, \phi]_{\mathcal{V}[X:=(S_1,S_2)]}$. We know f has greatest and least fixed-points by the Knaster-Tarski Theorem [Tar55] because pairs of sets ordered by

$$(S_1, S_2) \sqsubseteq (S_1', S_2') \stackrel{\text{def}}{=} S_1 \subseteq S_1' \text{ and } S_2 \supseteq S_2'$$

form a complete lattice with meet and join operators defined as $\bigvee\{(S_i, T_i) \mid i \in I\} \stackrel{\text{def}}{=} (\bigcup S_i, \bigcap T_i)$ and $\bigwedge\{(S_i, T_i) \mid i \in I\} \stackrel{\text{def}}{=} (\bigcap S_i, \bigcup T_i)$. The function $[M, \phi]_{\mathcal{V}}$ is order-preserving according to this order on pairs of sets.

For the PML$^+$ fragment of μL, this semantics is equivalent to the semantics given earlier for 3-valued PML$^+$, in the sense that $[(M,s) \models^+ \phi] = \mathit{true}$ just if s is in $\pi_1([M,\phi]_{\mathcal{V}})$, $[(M,s) \models^+ \phi] = \mathit{false}$ just if s is in $\pi_2([M,\phi]_{\mathcal{V}})$, and $[(M,s) \models^+ \phi] = \perp$ just if s is in neither $\pi_1([M,\phi]_{\mathcal{V}})$ nor $\pi_2([M,\phi]_{\mathcal{V}})$.

Now we show a result for μL analogous to that of the previous section for PML$^+$. Given a valuation $\mathcal{V}$ over a partial Kripke structure with state set S, we write $\mathcal{V}_p$ for the valuation that maps X to $\pi_1(\mathcal{V}(X))$, and $\mathcal{V}_o$ for the valuation that maps X to $S - \pi_2(\mathcal{V}(X))$.

Theorem 14. *Let $M = (S, L, \mathcal{R})$ be a partial Kripke structure, $\mathcal{V}$ be a valuation, M_p and M_o be the derived pessimistic and optimistic structures of M, and ϕ be a formula of μL. Then*

$$[M,\phi]_{\mathcal{V}} = (\|M_p,\phi\|_{\mathcal{V}_p}, S - \|M_o,\phi\|_{\mathcal{V}_o}).$$

4 The Generalized Model-Checking Problem

4.1 Problem Statement

We have said that our three-valued semantics gives a value of $\perp$ for some formula and some state in a partial Kripke structure just if the partial Kripke structure does not contain enough information to give answer *true* or *false*. However, it could be argued that our semantics returns $\perp$ more often than it should. Consider a partial Kripke structure M consisting of a single state s such that $s \rightarrow s$ and the value of proposition p at s is $\perp$. If we interpret formula $p \vee \neg p$ at s we get $\perp$, although in all complete Kripke structures more complete than M the formula is interpreted as *true*.

This problem is not confined to formulas that are tautological or unsatisfiable. Consider the partial Kripke structure like M above but for which the value of q at s is *true*. The formula $q \wedge (p \vee \neg p)$, which is neither a tautology nor unsatisfiable, is $\perp$ at s, yet again in all complete structures the formula is *true*.

Thus, our three-valued semantics does not have the desirable property that the value of a formula ϕ at a state is $\perp$ just if there exists an s' such that $s \preceq s'$ and the value of ϕ at s' is *true* and there also exists an s'' such that $s \preceq s''$ and the value of ϕ at s'' is *false*. However, we can use this property to define an alternative three-valued semantics for modal logics. We call this the *thorough* semantics because it does does more than our other semantics to discover whether enough information is present in a partial Kripke structure to give a definite answer. Let the *completions* $\mathcal{C}(M,s)$ of a state s of a partial Kripke structure M be the set of all states s' of complete Kripke structures M' such that $s \preceq s'$.

Definition 15 Thorough three-valued semantics. Let ϕ be a formula of any two-valued logic for which a satisfaction relation $\models$ is defined on complete

Kripke structures. The truth value of ϕ in a state s of a partial Kripke structure M under the *thorough* interpretation, written $[(M,s) \models \phi]_t$, is defined as follows:

$$[(M,s) \models \phi]_t = \begin{cases} \textit{true} \text{ if } (M',s') \models \phi \text{ for all } (M',s') \text{ in } \mathcal{C}(M,s) \\ \textit{false} \text{ if } (M',s') \not\models \phi \text{ for all } (M',s') \text{ in } \mathcal{C}(M,s) \\ \perp \quad \text{otherwise} \end{cases}$$

In this section we focus on the following related problem.

Definition 16 Generalized Model-Checking Problem. Given a state s of a partial Kripke structure M and a formula ϕ of a (two-valued) temporal logic L, does there exist a state s' of a complete Kripke structure M' such that $s \preceq s'$ and $(M',s') \models \phi$?

We call this problem the *generalized model-checking problem*. It should be clear that interpreting a formula according to the thorough three-valued semantics is equivalent to solving two instances of the generalized model checking problem. We have phrased the problem this way to emphasize its tie to the satisfiability problem.

The generalized model-checking problem generalizes both model checking and satisfiability checking. At one extreme, where M is $(\{s_0\}, L, \{(s_0,s_0)\})$ with $L(s_0,p) = \perp$ for all $p \in P$, all Kripke structures are more complete than M and the problem reduces to the satisfiability problem for the corresponding logic. At the other extreme, where M is complete, only a single structure needs to be checked and the problem reduces to model checking. Therefore, the worst-case complexity for the generalized model-checking problem will never be better than the worst-case complexities for the model-checking and satisfiability problems for the corresponding logic. The following theorem formally states that the generalized model-checking problem is at least as hard as the satisfiability problem.

Theorem 17. *Let L denote the modal mu-calculus or any of its fragments (propositional logic, propositional modal logic, LTL, CTL, CTL*, etc.). Then the satisfiability problem for L is reducible (in linear-time and logarithmic space) to the generalized model-checking problem for L.*

In the following sections, we present algorithms and complexity bounds for the generalized model-checking problem for various temporal logics. Our algorithms are based on automata-theoretic techniques (e.g., see [BVW94]). For basic notions of automata theory (including definitions of nondeterministic and alternating Büchi automata on words and trees), please refer to [Var97a].

4.2 Branching-Time Temporal Logics

We consider first the case of computation tree logic (CTL) [CES86]. The next theorem presents a decision procedure for the generalized model-checking problem for CTL.

Theorem 18. *Given a state s_0 of partial Kripke structure $M = (S, L, \mathcal{R})$ and a CTL formula ϕ, one can construct an alternating Büchi word automaton $A_{(M,s_0),\phi}$ over a 1-letter alphabet with at most $O(|S| \cdot 2^{O(|\phi|)})$ states such that*

$$(\exists (M', s_0') : s_0 \preceq s_0' \text{ and } (M', s_0') \models \phi) \text{ iff } \mathcal{L}(A_{(M,s_0),\phi}) \neq \emptyset.$$

Proof. (Sketch) $A_{(M,s_0),\phi}$ is constructed from the partial Kripke structure M and a nondeterministic Büchi tree automaton A_ϕ that accepts exactly the infinite trees satisfying the formula ϕ, in such a way that $A_{(M,s_0),\phi}$ accepts exactly the computation trees of complete Kripke structures that satisfy the property ϕ and that are more complete than (M, s_0).

A corollary of the above construction is that, if a state s_0' of a complete Kripke structure as defined in Theorem 18 exists, there also exists a state of a complete Kripke structure M' satisfying the property ϕ such that M' contains at most $|S| \cdot 2^{O(|\phi|)}$ states.

Since the emptiness problem for alternating Büchi word automata over a 1-letter alphabet can be reduced in linear time and logarithmic space to the emptiness problem for nondeterministic Büchi tree automata [BVW94], which is itself decidable in quadratic time [VW86], we obtain the following.

Theorem 19. *The generalized model-checking problem for a state s_0 of a partial Kripke structure $M = (S, L, \mathcal{R})$ and a CTL formula ϕ can be decided in time $O(|S|^2 \cdot 2^{O(|\phi|)})$.*

Note that, in the extreme case where M is complete, the upper bound given by the previous algorithm is not optimal since a traditional CTL model-checking algorithm [CES86] can decide whether $(M, s_0) \models \phi$ in time $O(|S| \cdot |\phi|)$.

In the general case, however, we can prove that the time complexity of the previous algorithm in the size of the formula is essentially optimal.

Theorem 20. *The generalized model-checking problem for CTL is EXPTIME-complete.*

Let us now discuss briefly the case of PML. Since PML is included in CTL, the above algorithm can also be used to solve the generalized model-checking problem for PML. However, the problem can now be solved using polynomial space.

Theorem 21. *The generalized model-checking problem for PML is PSPACE-complete.*

If we restrict the logic one step further and reduce it to propositional logic, it is easy to prove that the generalized model-checking problem has again the same complexity as the satisfiability problem.

Theorem 22. *The generalized model-checking problem for propositional logic is NP-complete.*

Let us now consider the case of branching-time logics more expressive than CTL, such as CTL* and the modal mu-calculus. Since formulas in these logics also have translations to nondeterministic Büchi tree automata (e.g., see [BVW94]), the general algorithm presented in the proof of Theorem 18 also provides a decision procedure for the generalized model-checking problem for these logics, with a quadratic time complexity in the size of the partial Kripke structure. As far as the complexity in the formula is concerned, we can prove the following.

Theorem 23. *For any branching-time temporal logic L containing CTL, the generalized model-checking problem for L is polynomial-time reducible to the satisfiability problem for L.*

Putting it all together, we obtain the following theorem which summarizes all the results presented in this section concerning the complexity of the generalized model-checking problem for branching-time temporal logics and a fixed partial Kripke structure.

Theorem 24. *Let L denote propositional logic, propositional modal logic, CTL, or any branching-time logic including CTL (such as CTL* or the modal mu-calculus). The generalized model-checking problem for the logic L has the same complexity as the satisfiability problem for L.*

4.3 Linear-Time Temporal Logics

Let us now turn to linear-time temporal logic (LTL) [MP92]. In this case, we can again reduce the generalized model-checking problem to checking emptiness of an alternating Büchi word automaton over a 1-letter alphabet.

Theorem 25. *Given a state s_0 of partial Kripke structure $M = (S, L, \mathcal{R})$ and an LTL formula ϕ, one can construct an alternating Büchi word automaton $A_{(M,s_0),\phi}$ over a 1-letter alphabet with at most $O(|S| \cdot 2^{|\phi|})$ states such that*

$$(\exists (M', s_0') : s_0 \preceq s_0' \text{ and } (M', s_0') \models \phi) \text{ iff } \mathcal{L}(A_{(M,s_0),\phi}) \neq \emptyset.$$

Since the proof of the above theorem is based only on the fact the property ϕ of interest can be represented by a nondeterministic Büchi word automaton, it also holds for properties directly represented by such automata (i.e., ω-regular languages) or formulas of logics which can be translated into such automata, like extended temporal logic [Wol83] and the linear-time fragment of the mu-calculus [SW90] for instance. We then obtain the following result.

Theorem 26. *The generalized model-checking problem for a state s_0 of a partial Kripke structure $M = (S, L, R)$ and an LTL formula ϕ can be decided in time $O(|S|^2 \cdot 2^{2|\phi|})$.*

So far, the above results for LTL are very similar to those of the previous section on the branching-time case. However, a major difference is that the generalized model-checking problem for LTL is harder than the satisfiability problem and the model-checking problem for LTL, which are both known to be PSPACE-complete [Eme90].

Theorem 27. *The generalized model-checking problem for linear-time temporal logic is EXPTIME-complete.*

In summary, in contrast with the results obtained for branching-time in the previous section, the generalized model checking problem is harder than the satisfiability problem in the LTL case. This is due to the need for alternating/tree automata to solve the problem. Other problems of that flavor include the *realizability* [ALW89] and *synthesis* [PR89a, PR89b] problems for linear-time temporal logic specifications.

At first sight, one could think that the need for tree automata could come from the mismatch between LTL and the completeness preorder $\preceq$. Indeed, the completeness preorder is not logically characterized by the 3-valued extension of LTL that can be obtained following the work of [BG99]. To see this, notice that the completeness preorder reduces to a bisimulation relation in the case of complete Kripke structures [BG99]. It is well-known that, while Kripke structures that are bisimilar satisfy the same LTL formulas, Kripke structures that satisfy the same LTL formulas are not necessarily bisimilar. The completeness preorder is thus stronger than necessary for reasoning only about the linear behaviors of partial Kripke structures. However, replacing the completeness preorder $\preceq$ by the weaker "linear" preorder induced by 3-valued LTL in the definition of the generalized model-checking problem does not make this problem easier: a constructive solution of the modified problem still requires the construction of an alternating automaton $A_{(M,s_0),\phi}$ as in the proof of Theorem 25.

5 Related Work

Most of the existing work on 3-valued modal logic focuses on its proof theory. For example, see [Seg67], [Mor89], and [Fit92a]. Our work in Section 3 is closest to [Fit92b]. Here Fitting presents two interpretations of modal logic: one a many-valued version and the other based on obtaining 2-valued interpretations from each of a set of experts. Fitting shows that such a multi-expert interpretation corresponds in a precise way to a multi-valued interpretation, similarly to how we show that a 3-valued interpretation can be obtained by separate optimistic and pessimistic interpretations. However, in Fitting's case the multi-expert interpretation is not obtained by separate, 2-valued interpretations of each expert. Also, Fitting does not define a completeness preorder over his models, or characterization results.

In [SRW99] a 3-valued logic is used for program analysis. The state of program store is represented as a 3-valued structure of first-order logic. The possible values of program store are conservatively computed by an abstract interpretation of the program on such a structure. The main technical result is an embedding theorem showing that, for a certain class of abstraction functions on the domain of such structures, the interpretation of a first-order formula on the abstract structure is less definite than its interpretation on the structure itself.

A semantics like our thorough semantics could be defined for other preorders on processes, such as refinement preorder of modal process logic [LT88] or the divergence preorder [Wal88]. In [BG99] we defined a 3-valued modal logic that characterizes the divergence preorder, but did not define a thorough semantics based on it. In [ALW89] an implementation preorder is defined on process specifications consisting of a finite labelled transition system and a nondeterministic Buchi word automaton. A process specification P is said to be *realizable* in [ALW89] if a P' lower in the implementation preorder exists for which the infinite behavior of the transition system of P' is contained in the language of the Buchi automaton of P'. The realizability problem and the generalized model checking problem for LTL clearly differ, but their relationship deserves further study.

References

[ALW89] Martín Abadi, Leslie Lamport, and Pierre Wolper. Realizable and unrealizable concurrent program specifications. In *Proc. 16th Int. Colloquium on Automata, Languages and Programming*, volume 372 of *Lecture Notes in Computer Science*, pages 1–17. Springer-Verlag, July 1989.

[BG99] Glenn Bruns and Patrice Godefroid. Model checking partial state spaces with 3-valued temporal logics. In N. Halbwachs and D. Peled, editors, *Proceedings of CAV '99, LNCS 1633*, pages 274–287, 1999.

[BVW94] Orna Bernholtz, Moshe Y. Vardi, and Pierre Wolper. An automata-theoretic approach to branching-time model checking. In *Computer Aided Verification, Proc. 6th Int. Workshop*, volume 818 of *Lecture Notes in Computer Science*, pages 142–155, Stanford, California, June 1994. Springer-Verlag.

[CE81] E. M. Clarke and E. A. Emerson. Design and Synthesis of Synchronization Skeletons using Branching-Time Temporal Logic. In D. Kozen, editor, *Proceedings of the Workshop on Logic of Programs,* Yorktown Heights, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer-Verlag, 1981.

[CES86] E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, January 1986.

[Eme90] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*. Elsevier/MIT Press, Amsterdam/Cambridge, 1990.

[Fit92a] Melvin Fitting. Many-valued modal logics I. *Fundamenta Informaticae*, 15:235–254, 1992.

[Fit92b] Melvin Fitting. Many-valued modal logics II. *Fundamenta Informaticae*, 17:55–73, 1992.

[HM85] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, 1985.

[Kle87] Stephen Cole Kleene. *Introduction to Metamathematics*. North Holland, 1987.

[Koz83] D. Kozen. Results on the Propositional Mu-Calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[LT88] Kim G. Larsen and Bent Thomsen. A modal process logic. In *Proceedings of the 3rd Annual Symposium on Logic in Computer Science*, pages 203–210. IEEE Computer Society Press, 1988.

[Mil89] R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.

[Mor89] Osamu Morikawa. Some modal logics based on a three-valued logic. *Notre Dame Journal of Formal Logic*, 30(1):130–137, 1989.

[MP92] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification.* Springer-Verlag, 1992.

[Par81] D. M. R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, 5^{th} *GI Conference*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, 1981.

[PR89a] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proc. of the Sixteenth Symposim on Principles of Programming Languages*, Austin, January 1989.

[PR89b] A. Pnueli and R. Rosner. On the synthesis of an asynchronous reactive module. In *Proceedings of ICALP'89*, Stresa, July 1989.

[Seg67] Krister Segerberg. Some modal logics based on a three-valued logic. *Theoria*, 33:53–71, 1967.

[SRW99] Mooly Sagiv, Thomas Reps, and Reinhard Wilhelm. Parametric shape analysis via 3-valued logic. In *Proceedings of the 26th Annual ACM Symposium on Principles of Programming Languages*, 1999.

[Sti87] Colin Stirling. Modal logics for communicating systems. *Theoretical Computer Science*, 49:331–347, 1987.

[SW90] C. Stirling and D. Walker. CCS, liveness and local model checking in the linear-time mu-calculus. In *Proc. First International Workshop on Automatic Verification Methods for Finite State Systems*, volume 407 of *Lecture Notes in Computer Science*, pages 166–178. Springer-Verlag, 1990.

[Tar55] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific J. of Maths*, 5:285–309, 1955.

[Var97a] M.Y. Vardi. Alternating automata: Checking truth and validity for temporal logics. In *Proceedings of CADE'97*, 1997.

[Var97b] M.Y. Vardi. Why is modal logic so robustly decidable? In *Proceedings of DIMACS Workshop on Descriptive Complexity and Finite Models.* AMS, 1997.

[VW86] M.Y. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *Journal of Computer and System Science*, 32(2):183–221, April 1986.

[Wal88] D. J. Walker. Bisimulations and divergence. In *Proceedings of the 3rd Annual Symposium on Logic in Computer Science.* IEEE Computer Society Press, 1988.

[Wol83] Pierre Wolper. Temporal logic can be more expressive. *Information and Control*, 56(1–2):72–99, 1983.