


# On the Complexity of Semantic Self-minimization

Adam Antonik<sup>1,2</sup>

View metadata, citation and similar papers at [core.ac.uk](http://core.ac.uk)

brought to you by  CORE

provided by Elsevier - Publisher Connector

and

Michael Huth<sup>1,3</sup>

*Department of Computing  
Imperial College London  
London, United Kingdom*

---

## Abstract

Partial Kripke structures model only parts of a state space and so enable aggressive abstraction of systems prior to verifying them with respect to a formula of temporal logic. This partiality of models means that verifications may reply with *true* (all refinements satisfy the formula under check), *false* (no refinement satisfies the formula under check) or *don't know*. Generalized model checking is the most precise verification for such models (all *don't know* answers imply that some refinements satisfy the formula, some don't), but computationally expensive. A compositional model-checking algorithm for partial Kripke structures is efficient, sound (all answers *true* and *false* are truthful), but may lose precision by answering *don't know* instead of a factual *true* or *false*. Recent work has shown that such a loss of precision does not occur for this compositional algorithm for most practically relevant patterns of temporal logic formulas. Formulas that never lose precision in this manner are called semantically self-minimizing. In this paper we provide a systematic study of the complexity of deciding whether a formula of propositional logic, propositional modal logic or the propositional modal mu-calculus is semantically self-minimizing.

*Keywords:* 3-valued model checking, partial state spaces, computational complexity, superevaluations.

---

## 1 Introduction

Partial state spaces abstract an actual state space so that the resulting smaller state space allows for a feasible verification of such partial models. Such a check

---

<sup>1</sup> This research was, in part, sponsored by the UK EPSRC grants EP/D50595X/1 *Efficient Specification Pattern Library for Model Validation* and EP/E028985/1 *Complete and Efficient Checks for Branching-Time Abstractions*

<sup>2</sup> Email: [antonik@lsv.ens-cachan.fr](mailto:antonik@lsv.ens-cachan.fr)

<sup>3</sup> Email: [M.Huth@doc.imperial.ac.uk](mailto:M.Huth@doc.imperial.ac.uk)

then has to be conservative: if the partial model satisfies the property under check, then *all* concrete systems consistent with that partial model should satisfy that property. This is required since we otherwise cannot be certain whether the actual state space satisfies this property, unless we check it on that actual system – the task we intend to avoid by using this abstraction technique in the first place – or have some other means of connecting properties of that abstraction to properties of the actual system.

If  $\phi$  is the property we want to check, and  $\phi_M$  a property that characterizes all concrete systems that are consistent with the partial model  $M$ , we therefore want to check whether the property  $\phi_M \rightarrow \phi$ , a logical implication, holds in all concrete systems. Identifying the set of concrete systems with some class of models, and assuming that  $\phi_M$  and  $\phi$  belong to some logic that is interpretable over that class of models, we recognize this problem as a validity check of some logic. Since the size of  $\phi_M$  may be exponential in the size of  $M$ , this conceptually useful insight is perhaps less useful as the basis for efficient algorithms. Indeed, for the propositional modal mu-calculus [15] this would render a method for property verification in 2EXPTIME whereas the most precise conservative check of partial models is EXPTIME-complete [8].

A less precise *compositional* – but much cheaper and still conservative – algorithm [7] (a 3-valued generalization of the familiar labelling algorithm) is available as an alternative that often can verify that all concrete models consistent with a partial one satisfy the property in question. Given that alternative, it is tempting to ask whether a property  $\phi$  is such that the cheaper algorithm always produces a correct result, regardless of the choice of partial model. In fact, it turns out [5] that this is the case for most practically relevant patterns of temporal logic specifications, as documented in [11] and at [patterns.projects.cis.ksu.edu](http://patterns.projects.cis.ksu.edu).

We can express this desirable feature of a property  $\phi$  – written in the propositional modal mu-calculus – via a computation of some normal form, followed by a “quantified” model check. For such a  $\phi$ , one can compute a formula  $\phi^p$  in the propositional modal mu-calculus such that running the cheap model-checking algorithm on  $\phi^p$  gives us the same result as running the expensive but most precise algorithm on  $\phi$ , for *all* partial models. We will refer to  $\phi^p$  as a *pessimistic semantic minimization* of  $\phi$ . Unfortunately,  $\phi^p$  may be exponentially larger than  $\phi$ .

This blow-up in the size of the formula suggests that one cannot decide efficiently whether properties have an efficient, compositional, *and* precise verification of partial models. The results in this paper corroborate this suspicion since we secure hardness results matching those for the validity problem of the respective logics. Alas, our hardness results only come with an upper bound of an exponential gap. To add to this frustration, we cannot show any hardness results for deciding whether a formula *and* its negation do not lose precision in the sense discussed above.

From a practitioner’s point of view, the results of this paper may not matter much. For one, and as already mentioned, popular specification patterns were shown to either not lose precision or to have minor syntactic variants that don’t

lose precision [5]. For another, temporal logic formulas used in practice tend to be rather short, and so an exponential or double exponential worst-case inflation in their size may sometimes be feasible. But we still think that the results reported here are of interest. They pose the question of the exact computational complexity of the decision problems associated with semantic minimizations. And they also suggest connections with recent work on the complexity of decision problems for modal and mixed specifications [3,2,4]. Such specifications are related to the partial models we study in this paper.

### Outline of paper.

In Section 2 we provide background on the verification of partial systems and on the concepts presented informally in the introduction. Our technical results, upper and lower bounds on the computational complexity of decision problems for semantic minimizations in temporal logics, are featured in Section 3. Related work is discussed in Section 4, and we conclude the paper in Section 5.

## 2 Background

In this paper we work with a countable set of atomic propositions,  $\mathbb{A}\mathbb{P}$ . Atomic propositions are the observations one can make at states of partial models.

We now define our models of partial state spaces. Since all temporal logics we study enjoy a finite-model property (and since we wish to verify abstractions that have finite state space), we will – without loss of generality – work with finite-state models throughout this paper.

**Definition 2.1** A partial Kripke structure [7]  $M$  is a tuple  $(S, R, L)$  where  $S$  is a finite set of states,  $R \subseteq S \times S$  is a state transition relation, and  $L: S \times \mathbb{A}\mathbb{P} \rightarrow \{0, 1/2, 1\}$  a total (labeling) function such that  $L(s, q)$  specifies the truth value of atomic proposition  $q$  at state  $s$ .

We identify 0 with *false*, 1 with *true*, and 1/2 with *don't know* and make  $\{0, 1/2, 1\}$  into a poset with respect to the information ordering  $\leq_i$  [14,7] specified as partial order by  $1/2 \leq_i 0$  and  $1/2 \leq_i 1$  as the only non-reflexive instances of  $\leq_i$ . Figure 1 shows two partial Kripke structures. We note that Kripke structures  $M = (S, R, L)$  are those partial Kripke structures that don't have 1/2 in the image of  $L$ . The meaning of a partial Kripke structure is that it describes a set of Kripke structures, those refinements that resolve all partiality of the state space. Such resolution means that labels  $L(s, q)$  have no longer value 1/2 but that the state space may well be larger or smaller. Refinement is defined as in [7].

**Definition 2.2** Let  $M = (S_M, R_M, L_M)$  and  $N = (S_N, R_N, L_N)$  be two partial Kripke structures.

- (i) A binary relation  $\preceq \subseteq S_M \times S_N$  is a refinement iff  $s \preceq t$  implies
  - (a)  $L(s, q) \leq_i L(t, q)$  for all  $q \in \mathbb{A}\mathbb{P}$ ,
  - (b) for all  $(s, s') \in R_M$  there is  $(t, t') \in R_N$  with  $s' \preceq t'$ , and

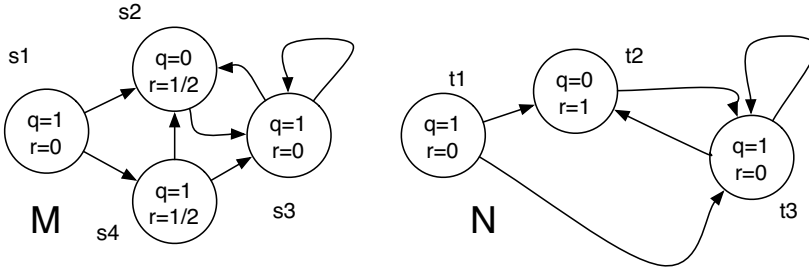


Fig. 1. Two partial Kripke structures  $M$  and  $N$  such that  $(N, t_1)$  refines  $(M, s_1)$  and  $N$  is a Kripke structure.

- (c) for all  $(t, t') \in R_N$  there is  $(s, s') \in R_M$  with  $s' \preceq t'$ .
- (ii) Given  $s \in S_M$  we call  $(M, s)$  a pointed model, which represents the partial Kripke structure  $M$  with initial state  $s$ .
- (iii) We say that  $(M, s)$  has  $(N, t)$  as refinement whenever there is a refinement  $\preceq$  as above such that  $s \preceq t$ .

**Example 2.3** Two partial Kripke structures are depicted in Figure 1. The one on the right is a pointed Kripke structure  $(N, t_1)$  and refines the pointed model  $(M, s_1)$  on the left where  $\preceq = \{(s_1, t_1), (s_2, t_2), (s_3, t_3), (s_4, t_3)\}$ .

We assume the usual satisfaction semantics between pointed Kripke structures and formulas of the propositional modal mu-calculus ( $\mathcal{MC}$ ), both of which we define below. Refinement and this standard satisfaction notion let us define two judgments, one for generalized model checking [8] ( $\text{SAT}$ , called GMC in loc. cit.), and one for its logical dual ( $\text{VAL}$ ).

**Definition 2.4** Let  $(M, s)$  be a pointed model and  $\phi$  a sentence of the propositional modal mu-calculus. Then  $\text{SAT}(M, s, \phi)$  holds iff there is a pointed Kripke structure that refines  $(M, s)$  and satisfies  $\phi$ . Dually,  $\text{VAL}(M, s, \phi)$  holds iff all pointed Kripke structures that refine  $(M, s)$  satisfy  $\phi$ .

Whenever convenient, as in the next example, we make liberal use of CTL\* connectives as syntactic sugar expressible in  $\mathcal{MC}$ .

**Example 2.5** For the partial Kripke structure  $M$  in Figure 1, the judgments  $\text{VAL}(M, s_1, \text{AF}(q \wedge \neg r))$  and  $\text{VAL}(M, s_1, \text{AF EG } \neg r)$  hold. But we don't have  $\text{VAL}(M, s_1, \text{AF AG } \neg r)$ , where the pointed Kripke structure  $(N, t_1)$  serves as a counterexample.

We hasten to point out that every partial Kripke structure has some pointed Kripke structure as refinement, so there are no vacuities [16] – here at the level of models – introduced into these two judgments  $\text{SAT}(M, s, \phi)$  and  $\text{VAL}(M, s, \phi)$ . When  $\phi$  ranges over the propositional modal mu-calculus, both judgments are EXPTIME-complete in the size of  $\phi$  and quadratic in the size of  $M$  [8].

The compositional semantics of [7] trades off the precision of these judgments with their computational complexity, and gives rise to the aforementioned cheap and

sound, compositional algorithm. We present this compositional semantics through two judgments, a “pessimistic” one  $(M, s) \models^p \phi$ , and an “optimistic” one  $(M, s) \models^o \phi$  between pointed models and formulas of the propositional modal mu-calculus.

The syntax of this propositional modal mu-calculus  $\mathcal{MC}$  is given by

$$(1) \quad \phi ::= q \mid Z \mid \phi \wedge \phi \mid \neg\phi \mid \text{EX } \phi \mid \mu Z. \phi$$

where  $q$  ranges over a set of propositional atoms,  $Z$  ranges over a set of recursion variables,  $\mu Z. \phi$  binds occurrences of  $Z$  in its body  $\phi$ , and all free occurrences of  $Z$  in that body are under an even scope of negations. We use  $\phi \vee \psi$  as syntactic sugar for  $\neg(\neg\phi \wedge \neg\psi)$ , and use  $\rightarrow$  and  $\leftrightarrow$  as abbreviations with their usual encodings in terms of  $\neg$  and  $\vee$ . A formula  $\phi$  is a sentence if it has no free variables.

We let  $\mathcal{PML}$ , (basic) propositional modal logic, be those formulas of  $\mathcal{MC}$  that contain neither recursion variables  $Z$  nor fixed-point operators  $\mu Z$ . Propositional logic is the set  $\mathcal{PL}$  of those formulas of  $\mathcal{PML}$  that don’t contain any EX operator.

For each partial Kripke structure  $M = (S, R, L)$  and each formula  $\phi \in \mathcal{MC}$  we define in Figure 2 a set  $\llbracket \phi \rrbracket_\rho^m$  (for  $m \in \{o, p\}$ ) of those states in  $S$  that satisfy  $\phi$ , where  $\rho$  maps each recursion variable  $Z$  to a set of states,  $\rho(Z) \subseteq S$ .

We explain the salient features of this semantics:

- For atomic propositions, the “optimistic” semantics  $\llbracket q \rrbracket_\rho^o$  interprets the labels 1 and 1/2 for  $q$  at states as denoting truth of  $q$  at those states. The “pessimistic” semantics  $\llbracket q \rrbracket_\rho^p$ , in contrast, interprets only the label 1 for  $q$  at states as denoting truth of  $q$  at those states.
- The meaning of conjunction is *compositional*, interpreted as intersection as usual in Kripke structures.
- The meaning of negation is that it swaps the evaluation mode of the semantics: to evaluate  $\neg\phi$  in the pessimistic mode, say, we evaluate  $\phi$  in the optimistic mode and then form the set complement of the resulting set of states. This reflects the duality between satisfiability and validity checking inherent in this reasoning about sets of refinements of models.

For a sentence  $\phi \in \mathcal{MC}$  we define

$$(2) \quad (M, s) \models^m \phi \stackrel{\text{def}}{=} s \in \llbracket \phi \rrbracket_\rho^m \text{ for some } \rho$$

This is well defined as  $\llbracket \phi \rrbracket_\rho^m$  is independent of the choice of  $\rho$  for sentence  $\phi$ . The cost of computing  $\llbracket \phi \rrbracket_\rho^m$  is, up to a constant, essentially that of computing the standard satisfaction relation on Kripke structures [8]. So the judgments in (2) can be computed as efficiently as satisfaction for Kripke structures. We note that both  $(K, t) \models^o \phi$  and  $(K, t) \models^p \phi$  render the standard satisfaction relation [15] on pointed Kripke structures  $(K, t)$ .

The results and proofs in this paper will only apply to sentences. We therefore abuse notation by referring to  $\mathcal{MC}$ ,  $\mathcal{PML}$ , and  $\mathcal{PL}$  also as the subsets of sentences of these respective logics. Context will always determine the proper interpretation of these symbols.

The next theorem, whose result is implicit in [7,8], connects the compositional semantics with that of generalized model checking.

$$\begin{array}{ll}
\llbracket q \rrbracket_{\rho}^o = \{s \mid L(s, q) \neq 0\} & \llbracket q \rrbracket_{\rho}^p = \{s \mid L(s, q) = 1\} \\
\llbracket Z \rrbracket_{\rho}^o = \rho(Z) & \llbracket Z \rrbracket_{\rho}^p = \rho(Z) \\
\llbracket \phi \wedge \psi \rrbracket_{\rho}^o = \llbracket \phi \rrbracket_{\rho}^o \cap \llbracket \psi \rrbracket_{\rho}^o & \llbracket \phi \wedge \psi \rrbracket_{\rho}^p = \llbracket \phi \rrbracket_{\rho}^p \cap \llbracket \psi \rrbracket_{\rho}^p \\
\llbracket \neg \phi \rrbracket_{\rho}^o = S \setminus \llbracket \phi \rrbracket_{\rho}^p & \llbracket \neg \phi \rrbracket_{\rho}^p = S \setminus \llbracket \phi \rrbracket_{\rho}^o \\
\llbracket \text{EX } \phi \rrbracket_{\rho}^o = \text{pre}(\llbracket \phi \rrbracket_{\rho}^o) & \llbracket \text{EX } \phi \rrbracket_{\rho}^p = \text{pre}(\llbracket \phi \rrbracket_{\rho}^p) \\
\llbracket \mu Z. \phi \rrbracket_{\rho}^o = \text{lfp } F_{\phi, \rho}^o & \llbracket \mu Z. \phi \rrbracket_{\rho}^p = \text{lfp } F_{\phi, \rho}^p
\end{array}$$

Fig. 2. Compositional semantics of propositional modal mu-calculus formulas  $\phi$  over a partial Kripke structure  $M = (S, R, L)$  for environment  $\rho$  mapping recursion variables to subsets of  $S$ , where  $\text{pre}: \mathbb{P}(S) \rightarrow \mathbb{P}(S)$  is  $\text{pre}(X) = \{s \in S \mid \exists (s, s') \in R: s' \in X\}$ ,  $F_{\phi, \rho}^m: (\mathbb{P}(S), \subseteq) \rightarrow (\mathbb{P}(S), \subseteq)$  is  $F_{\phi, \rho}^m(X) = \llbracket \phi \rrbracket_{\rho[Z \mapsto X]}^m$  for  $m \in \{o, p\}$ , and  $\text{lfp } F$  is the least fixed point of a monotone function  $F$  on the complete lattice  $(\mathbb{P}(S), \subseteq)$ .

**Theorem 2.6** *Let  $(M, s)$  be a pointed model and  $\phi \in \mathcal{MC}$  a sentence. Then:*

- (i)  $(M, s) \models^p \phi$  implies  $\text{VAL}(M, s, \phi)$  and
- (ii)  $\text{SAT}(M, s, \phi)$  implies  $(M, s) \models^o \phi$ .

Item (i) states that the compositional semantics is sound. We can verify the expensive  $\text{VAL}(M, s, \phi)$  by trying to establish the much cheaper  $(M, s) \models^p \phi$ . Item (ii) in that theorem may not be of direct interest in verification but is useful for the proof of the first implication.

Showing  $\text{VAL}(M, s, \phi)$  through the cheaper judgment  $(M, s) \models^p \phi$  won't always succeed.

**Example 2.7** [12] Consider the formula  $\phi = \text{EX } q \wedge (\text{EX } r \vee \text{EX } \neg r)$  and any model  $M = (\{s\}, \{(s, s)\}, L)$  with  $L(s, q) = 1$  and  $L(s, r) = 1/2$ . Then  $(M, s) \not\models^p \phi$  since  $(M, s) \not\models^p \text{EX } r \vee \text{EX } \neg r$ . But all pointed Kripke structures that refine  $(M, s)$  satisfy  $\phi$ , i.e.  $\text{VAL}(M, s, \phi)$  holds.

This example illustrates the necessary trade-off between the precision of  $\text{VAL}(M, s, \phi)$  and the lower computational cost of  $(M, s) \models^p \phi$ . In this paper we are interested in those  $\phi \in \mathcal{MC}$  for which the implications in Theorem 2.6 are reversible for all pointed models. This leads to the concept of semantic minimization [6, 21, 20, 12].

**Definition 2.8** [12] A sentence  $\phi \in \mathcal{MC}$  is

- (i) *pessimistically self-minimizing* iff for all pointed models  $(M, s)$  we have  $(M, s) \models^p \phi \Leftrightarrow \text{VAL}(M, s, \phi)$ ,
- (ii) *optimistically self-minimizing* iff for all pointed models  $(M, s)$  we have  $(M, s) \models^o \phi \Leftrightarrow \text{SAT}(M, s, \phi)$ , and
- (iii) *semantically self-minimizing* iff it is optimistically and pessimistically self-minimizing.

Given these three concepts, we write PSM, OSM, and  $\text{PSM} \cap \text{OSM}$  for the sets of sentences of  $\mathcal{MC}$  that satisfy the respective concept in items (i), (ii), and (iii).

**Example 2.9** • The formula  $q \vee \neg q$  is optimistically self-minimizing but not pessimistically so – consider the case when  $L(s, q) = 1/2$ .

- The pattern “Precedence Chain: 2 stimuli, 1 response; Globally  $q$  and  $s$  precede  $r$ ” [11], as documented at [patterns.projects.cis.ksu.edu](http://patterns.projects.cis.ksu.edu), written in  $\mathcal{MC}$  as

$$\neg E[\neg q U r] \wedge E[\neg r U (q \wedge \neg r \wedge EX (E[\neg s U (r \wedge \neg s)])])]$$

is pessimistically self-minimizing [5].

- The pattern “Absence of  $q$ , Before  $r$ ”, written in  $\mathcal{MC}$  as

$$\phi = A[\neg q \vee AG(\neg r)W r]$$

is not pessimistically self-minimizing but

$$\phi^p = A[\neg q \vee AX (AG(\neg r)W r)]$$

is, and is logically equivalent to  $\phi$  over Kripke structures [5].

### 3 Decision Problems

We write VAL for the valid sentences of  $\mathcal{MC}$  and, dually, UNSAT for the unsatisfiable sentences of  $\mathcal{MC}$ . The set  $\mathcal{MC}$  is partitioned into VAL, UNSAT, and  $\mathcal{MC} \setminus (\text{VAL} \cup \text{UNSAT})$ . If we refine this partition with the sets PSM and OSM under union, intersection, and complement we arrive at six equivalence classes. Throughout, it will be clear from the context whether these sets are meant to be subsets of  $\mathcal{MC}$ ,  $\mathcal{PML}$  or  $\mathcal{PL}$ .

The proposition stated next would not hold, as stated, in the presence of constants for *true* and *false* in  $\mathcal{MC}$ . But their absence from  $\mathcal{MC}$  merely simplifies the presentation of our results.

**Proposition 3.1** (i) VAL is contained in OSM and disjoint from PSM. Dually, UNSAT is contained in PSM and disjoint from OSM.

- (ii) For each  $\mathcal{L} \in \{\mathcal{MC}, \mathcal{PML}, \mathcal{PL}\}$ , set  $\mathcal{L}$  is partitioned into the six sets

- I VAL
- II UNSAT
- III  $\text{OSM} \setminus (\text{VAL} \cup \text{PSM})$
- IV  $\text{PSM} \setminus (\text{UNSAT} \cup \text{OSM})$
- V  $\mathcal{L} \setminus (\text{PSM} \cup \text{OSM})$
- VI  $\text{PSM} \cap \text{OSM}$

as illustrated in Figure 3.

#### Proof (Sketch)

- (i) Consider the partial Kripke structure  $M_\perp = (\{s_\perp\}, \{(s_\perp, s_\perp)\}, L_\perp)$  where  $L_\perp(s_\perp, q) = 1/2$  for all  $q \in \mathbb{AP}$ . Then every pointed Kripke structure (over  $\mathbb{AP}$ ) is a refinement of  $(M_\perp, s_\perp)$ . One shows that for all sentences  $\phi \in \mathcal{MC}$  we have

$$(3) \quad (M_\perp, s_\perp) \models^o \phi \quad \text{and} \quad (M_\perp, s_\perp) \not\models^p \phi$$

The first statement in (3) is used to prove that UNSAT and OSM are disjoint.

The second statement in (3) implies that VAL is disjoint from PSM.

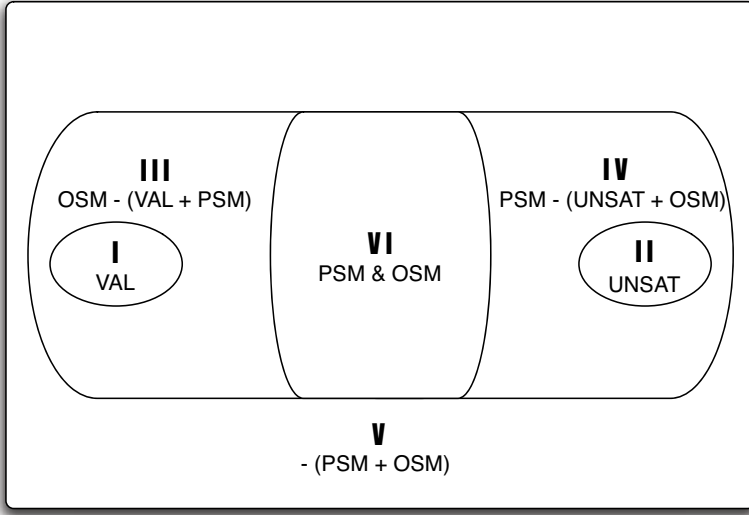


Fig. 3. A partition of  $\mathcal{MC}$ ,  $\mathcal{PML}$ , and  $\mathcal{PL}$  into six equivalence classes, generated by Boolean combinations of  $\text{VAL}$  and  $\text{PSM}$ .

To see that  $\text{VAL} \subseteq \text{OSM}$ , let  $\phi$  be valid and  $(M, s)$  a pointed model. We have that  $\text{SAT}(M, s, \phi)$  holds since there are refining pointed Kripke structures of  $(M, s)$  and  $\phi$  is valid. But then  $(M, s) \models^o \phi$  holds by Theorem 2.6. Since  $(M, s)$  was arbitrary,  $\phi \in \text{OSM}$  follows.

The dual statement, that  $\text{UNSAT}$  is contained in  $\text{PSM}$ , is proved in the dual fashion.

- (ii) From the first item it follows that  $\text{VAL}$ ,  $\text{PSM}$ , and  $\text{OSM}$  alone generate only the six sets **I** - **VI** by repeated applications of union, intersection, and complement; and that these six sets are mutually disjoint. We list sentences that show all six sets are inhabited:  $q \vee \neg q \in \text{VAL}$ ,  $q \wedge \neg q \in \text{UNSAT}$ ,  $(q \vee \neg q) \wedge r \in \text{OSM} \setminus (\text{VAL} \cup \text{PSM})$ ,  $(q \wedge \neg q) \vee r \in \text{PSM} \setminus (\text{UNSAT} \cup \text{OSM})$ ,  $[q \wedge (r \vee \neg r)] \vee [\neg q \wedge (r \wedge \neg r)] \in \mathcal{MC} \setminus (\text{PSM} \cup \text{OSM})$ , and  $q \in \text{PSM} \cap \text{OSM}$ . Since these sentences are in  $\mathcal{PL}$ , we thus have a partition for all three cases of  $\mathcal{L} \in \{\mathcal{MC}, \mathcal{PML}, \mathcal{PL}\}$ .  $\square$

For each of these six sets we now study the complexity of deciding membership of that set. We first observe how negation acts on these sets.

**Lemma 3.2** *Negation  $\phi \mapsto \neg\phi$  maps the following pairs of sets into each other:  $\text{OSM}$  and  $\text{PSM}$ , **I** and **II**, **III** and **IV**, **V** and itself, and **VI** and itself.*

**Proof (Sketch)** A sentence is in  $\text{VAL}$  iff its negation is in  $\text{UNSAT}$ , and it is in  $\text{OSM}$  iff its negation is in  $\text{PSM}$ . Therefore a sentence is in set **I** iff its negation is in set **II**, and it is in  $\text{OSM} \setminus (\text{VAL} \cup \text{PSM})$  iff its negation is in  $\text{PSM} \setminus (\text{UNSAT} \cup \text{OSM})$ . This shows the claim for **I** and **II**, and for **III** and **IV**. Sets **V** and **VI** are closed under negation, since  $\phi \mapsto \neg\phi$  maps set  $\text{OSM}$  into  $\text{PSM}$ , and vice versa.  $\square$



### 3.1 Sets **I** and **II**.

Deciding membership of sets **I** (VAL) and **II** (UNSAT) has, of course, the same complexity as that of the validity of the underlying logic – EXPTIME-complete for  $\mathcal{MC}$ , PSPACE-complete for  $\mathcal{PML}$ , and coNP-complete for  $\mathcal{PL}$ .

### 3.2 Set OSM.

By Lemma 3.2 we have  $\phi \in \text{OSM} \Leftrightarrow \neg\phi \in \text{PSM}$ . Thus the complexity of deciding OSM is the same as that of deciding PSM. But deciding OSM is at least as hard as deciding validity of the underlying logic. This can be seen by considering the function

$$(4) \quad E(\phi) = \phi \vee (x \wedge \neg x)$$

where  $x$  is a propositional atom not in  $\phi$ , and so not contained in  $\phi$ . If  $\phi$  is valid, then so is  $E(\phi)$ , and this implies that  $E(\phi)$  is in OSM. If  $\phi$  is not valid, then there is a pointed Kripke structure  $(K, t)$  in which  $\phi$  is false. We extend the labeling function  $L$  of  $K$  so that  $L(s, x) = 1/2$  for all states  $s$  of  $K$  – making  $K$  into a partial Kripke structure. Then  $(K, t) \models^o E(\phi)$  follows for this extended  $K$  but there is no refinement of  $(K, t)$  that satisfies  $E(\phi)$  as this formula is semantically equivalent to  $\phi$  over Kripke structures. Thus,  $\phi$  is valid iff  $E(\phi)$  is in OSM and so we can reduce validity checks to checks of membership of OSM.

To summarize, deciding OSM is EXPTIME-hard, PSPACE-hard, and coNP-hard for  $\mathcal{MC}$ ,  $\mathcal{PML}$ , and  $\mathcal{PL}$  respectively.

### $\mathcal{MC}$ .

We can decide in 2EXPTIME whether a sentence  $\phi \in \mathcal{MC}$  is in OSM. This is implicit in [12], where from  $\phi$  two alternating tree automata are being constructed – with exponential blowup in the worst case – and membership of  $\phi$  in OSM is then being reduced to a language inclusion check for these automata, again, in EXPTIME but now in the size of these automata.

### $\mathcal{PML}$ .

Consider a sentence  $\phi \in \mathcal{PML}$ . As just said for  $\mathcal{MC}$ , in [12], two tree automata  $A_\phi^3$  and  $A_{\models^o \phi}^3$  were constructed such that  $\phi$  is optimistically self-minimizing iff  $\mathcal{L}(A_{\models^o \phi}^3) \subseteq \mathcal{L}(A_\phi^3)$ , i.e. the language accepted by  $A_{\models^o \phi}^3$  is contained in the language accepted by  $A_\phi^3$ . Since  $\mathcal{PML} \subseteq \mathcal{MC}$ , such a language inclusion check is in EXPTIME in the size of these automata. However, since both automata cannot distinguish trees at depths greater than the size of  $\phi$ , reflecting the shallow model property of  $\mathcal{PML}$ , such a language inclusion check can be performed already in PSPACE. Since the underlying automata has size at most exponential in the size of  $\phi$  we conclude that the language inclusion check can be done in EXPSPACE in the size of  $\phi$ .

```

boolean NotInOSM(phi) {
  **choose** model M such that M(x) = 1/2 for some x in AP(phi);
  if (M |=o phi) {
    for (all x in AP(phi) with M(x) = 1/2) {
      if (!(M[x --> 0] |=o phi) && !(M[x --> 1] |=o phi)) {
        ACCEPT;
      }
    }
  }
  REJECT;
}

```

Fig. 4. NP algorithm that decides membership of  $\mathcal{PL} \setminus \text{OSM}$ . If at least one choice of model leads to **ACCEPT**, the algorithm returns *true*; otherwise it returns *false*.

## $\mathcal{PL}$ .

We now show that, for  $\mathcal{PL}$ , deciding membership of OSM is in coNP, and so the above hardness result is indeed exact. Let  $\mathbb{AP}(\phi)$  be the set of atomic propositions that occur in  $\phi$ . The evaluation of  $(M, s) \models^o \phi$  for  $\phi \in \mathcal{PL}$  depends only on the values  $\{L_M(s, q) \mid \mathbb{AP}(\phi)\}$ . Therefore, we can think of the pointed model  $(M, s)$  as a function from atomic propositions to values in  $\{0, 1/2, 1\}$  and so we will write  $M(q)$  etc. below with that interpretation in mind. We write  $M[q \mapsto v]$  for the model that is as  $M$ , except that it maps  $q$  to value  $v \in \{0, 1\}$ . We will also use that  $M \models^o \phi$  will only depend on the behavior of  $M$  on set  $\mathbb{AP}(\phi)$ .

If there are  $k > 0$  atomic propositions in  $\phi \in \mathcal{PL}$ , we have just seen that we can decide semantic self-minimization of  $\phi$  by inspecting, for  $3^k$  models, whether the compositional model-checking algorithm loses any precision. This observation leads to a non-deterministic algorithm, depicted in Figure 4, for showing that  $\mathcal{PL} \setminus \text{OSM}$  is in NP, and so OSM is in coNP.

The idea behind the algorithm is the following: suppose we have that  $M \models^o \phi$  but neither  $M[q \mapsto 0] \models^o \phi$  nor  $M[q \mapsto 1] \models^o \phi$  hold. Then  $\phi$  cannot be in OSM: otherwise, there is some function  $K$  from atomic propositions to  $\{0, 1\}$  that refines  $M$  and satisfies  $\phi$ . But any refinement of  $M$  has to be either a refinement of  $M[q \mapsto 0]$  or  $M[q \mapsto 1]$ . So at least one  $M[q \mapsto 0] \models^o \phi$  and  $M[q \mapsto 1] \models^o \phi$  would have to hold by Theorem 2.6, a contradiction. Thus this is a sound test for membership in OSM. But if we apply this method to all partial models  $M$ , it becomes complete as such a membership test.

**Proposition 3.3** *The NP algorithm in Figure 4 correctly decides membership of  $\mathcal{PL} \setminus \text{OSM}$ .*

## Proof (Sketch)

- Let  $\phi \in \mathcal{PL}$  be such that the algorithm accepts it. Then there is some model  $M$  and some  $x \in \mathbb{AP}(\phi)$  such that
  - $M(x) = 1/2$ ,
  - $M \models^o \phi$ , and
  - $M[x \mapsto v] \not\models^o \phi$  for all  $v \in \{0, 1\}$ .

The last item and Theorem 2.6 imply that  $\text{SAT}(M[x \mapsto v], \cdot, \phi)$  is false for both  $v = 0, 1$ . Then it must also be that  $\text{SAT}(M, \cdot, \phi)$  is false. But then the second item implies that  $\phi$  is not in OSM.

- Conversely, let  $\phi \notin \text{OSM}$ . Then, by Theorem 2.6, there is a model  $M$  such that  
 (5)  $M \models^o \phi$  holds and  $\text{SAT}(M, \cdot, \phi)$  does not hold

This can only be if there is some  $x \in \text{AP}(\phi)$  with  $M(x) = 1/2$ . Let  $M$  be a model satisfying (5) but where the set

$$\{x \in \text{AP}(\phi) \mid M(x) = 1/2\}$$

is minimal amongst all models satisfying (5). Since there is some  $x$  with  $M(x) = 1/2$ , the algorithm will encounter the first if-statement and its guard will be true. Therefore, its for-statement will be executed and, for the first (indeed all)  $x$  it executes we can now reason that the algorithm will accept.

Since  $\text{SAT}(M, \cdot, \phi)$  is false, we know that  $\text{SAT}(M[x \mapsto v], \cdot, \phi)$  is false for all  $v \in \{0, 1\}$ . By the minimality of model  $M$  with respect to (5), we infer that  $M[x \mapsto v] \models^o \phi$  has to be false for  $v = 0$  and for  $v = 1$ . Thus the algorithm reaches ACCEPT.

□

### 3.3 Set III.

Deciding set **III** is at least as hard as deciding OSM, and therefore at least as hard as deciding validity of the underlying logic. To see this, consider the function

$$(6) \quad F(\phi) = (\phi \vee x) \wedge (y \wedge (z \vee \neg z))$$

where  $x$ ,  $y$ , and  $z$  are propositional atoms not contained in  $\phi$ . The reduction is shown through the composition  $F \circ E$  if

$$(7) \quad F(\phi) \in \text{III} \quad \Leftrightarrow \quad \phi \in \text{OSM}$$

holds. We show (7):

- For no  $\phi$  is  $F(\phi)$  in PSM: consider a pointed model  $(M, s)$  with  $L(s, x) = L(s, y) = 1$  and  $L(s, z) = 1/2$ . Then  $(M, s) \not\models^p F(\phi)$  holds but all pointed Kripke structures that refine  $(M, s)$  satisfy  $\phi$ .
- For no  $\phi$  is  $F(\phi)$  in VAL since there are, e.g., pointed models  $(M, s)$  for which  $L(s, y) = 0$ .

So  $F(\phi)$  is in set **III** iff  $F(\phi)$  is in  $\text{OSM} \setminus \text{VAL}$  iff  $F(\phi)$  is in OSM. Thus it suffices to show  $F(\phi) \in \text{OSM} \Leftrightarrow \phi \in \text{OSM}$ . In doing so, we appeal to the fact that  $\phi, \psi \in \text{OSM}$  imply that  $\phi \wedge \psi \in \text{OSM}$  whenever  $\phi$  and  $\psi$  share no atomic propositions, and that OSM is closed under disjunctions [12].

- Let  $\phi \in \text{OSM}$ . Since  $y$  and  $z \vee \neg z \in \text{OSM}$  we get  $y \wedge (z \vee \neg z) \in \text{OSM}$  as both conjuncts share no atomic propositions. Since  $\phi, x \in \text{OSM}$ , their disjunction  $\phi \vee x$  is in OSM as well. Since  $\phi \wedge x$  and  $y \wedge (z \vee \neg z)$  share no atomic proposition and both are in OSM, we get  $F(\phi) \in \text{OSM}$ .
- Let  $\phi \notin \text{OSM}$ . Then there is a pointed model  $(M, s)$  such that  $(M, s) \models^o \phi$  and  $\text{SAT}(M, s, \phi)$  is false. Extend the labeling function  $L_M$  of  $(M, s)$  such that  $L_M(s, y) = L_M(s, z) = 1$  and  $L_M(s, x) = 0$ . Then  $(M, s) \models^o F(\phi)$  holds for this extension but  $\text{SAT}(M, s, F(\phi))$  is false, since  $\text{SAT}(M, s, \phi)$  is false and all pointed

Kripke structures  $(K, t)$  that refine the extended  $(M, s)$  must satisfy  $L_K(t, x) = 0$  and  $L_K(t, y) = 1$ . So  $F(\phi) \notin \text{OSM}$ .

Combining (7) with the reduction of OSM to validity checks, we infer that deciding set **III** is EXPTIME-hard, PSPACE-hard, and coNP-hard for  $\mathcal{MC}$ ,  $\mathcal{PML}$  and  $\mathcal{PL}$  (respectively).

$\mathcal{MC}$ .

We can decide PSM and OSM in 2EXPTIME, and decide VAL in EXPTIME. So we can decide set **III** in 2EXPTIME.

$\mathcal{PML}$ .

We already have seen that OSM can be decided in EXPSPACE, so this applies to PSM as well. Since VAL can be decided in PSPACE we can decide set **III** also in EXPSPACE.

$\mathcal{PL}$ .

We have shown that OSM is in coNP. By Lemma 3.2 this implies that PSM is in coNP as well. Since VAL is in coNP, the language  $\mathcal{PL} \setminus (\text{VAL} \cup \text{PSM})$  is in NP. Set **III** equals  $\text{OSM} \cap (\mathcal{PL} \setminus (\text{VAL} \cup \text{PSM}))$  and so is in DP [18,19] as the intersection of a language in coNP with one in NP. We are presently unable to show DP-hardness of set **III**, despite having made a considerable effort to that end.

### 3.4 Set **V**.

Sentences in set **V** lose precision in the pessimistic and in the optimistic compositional semantics. Since unsatisfiable sentences are in PSM, sentences in set **V** must be satisfiable. Deciding membership of set **V** is also at least as hard as the satisfiability check of the relevant logic. To see this, consider

$$(8) \quad G(\phi) = (\phi \wedge (x \vee \neg x) \wedge y) \vee (z \wedge \neg z)$$

where  $x$ ,  $y$ , and  $z$  are in  $\mathbb{AP}$  and again not appearing in  $\phi$ .

- For no  $\phi$  is  $G(\phi)$  in OSM, and so  $G(\phi)$  is in set **V** iff  $G(\phi)$  is not in PSM: consider a pointed model  $(M, s)$  with  $L_M(s, z) = 1/2$  and  $L_M(s, y) = 0$ . Then  $(M, s) \models^o G(\phi)$  holds but no pointed Kripke structure that refines  $(M, s)$  satisfies  $\phi$ .
- Now if  $\phi$  is unsatisfiable, then  $G(\phi)$  is also unsatisfiable, so  $G(\phi)$  will be in PSM. Conversely, if  $\phi$  is satisfiable (on some pointed Kripke structure  $(K, t)$ ) we claim that  $G(\phi)$  is not in PSM. To see this we make  $K$  into a partial Kripke structure by extending its labeling function  $L$  with  $L(t, x) = L(t, y) = 1$  and  $L(t, z) = 1/2$ . Then all pointed Kripke structures that refine this expanded  $(K, t)$  satisfy  $\phi$ , yet  $(K, t) \not\models^p G(\phi)$ .

The combination of these two items shows

$$(9) \quad \phi \text{ satisfiable} \quad \Leftrightarrow \quad G(\phi) \in \mathbf{V}$$

To summarize, deciding set  $\mathbf{V}$  is EXPTIME-hard, PSPACE-hard, and NP-hard for  $\mathcal{MC}$ ,  $\mathcal{PML}$ , and  $\mathcal{PL}$  (respectively).

$\mathcal{MC}$ .

It is easily seen that deciding set  $\mathbf{V}$  is in 2EXPTIME as that complexity class is closed under finite unions and complements.

$\mathcal{PML}$ .

We can decide membership of set  $\mathbf{V}$  by two checks, one for OSM and one for PSM – both were shown to be in EXPSPACE. We therefore conclude that set  $\mathbf{V}$  can be decided in EXPSPACE as well.

$\mathcal{PL}$ .

Since OSM and PSM are in coNP so is their union. But then set  $\mathbf{V}$  is in NP as the complement of a language in coNP. Since we already showed that set  $\mathbf{V}$  is NP-hard, we get that set  $\mathbf{V}$  is NP-complete.

### 3.5 Set $\mathbf{VI}$ .

Sentences in  $\mathbf{VI}$  are well behaved in that they lose precision neither for the pessimistic nor for the optimistic compositional semantics. So satisfiability and validity checks for *all* partial state spaces are reducible to a single, simple verification for such sentences. The exact complexity of deciding this set remains to be frustratingly unknown. Of course, deciding set  $\mathbf{VI}$  is no harder than deciding two instances of OSM:

$$(\phi \in \text{PSM} \cap \text{OSM}) \Leftrightarrow (\phi \in \text{OSM} \ \& \ \neg\phi \in \text{OSM})$$

So deciding set  $\mathbf{VI}$  is no harder than deciding validity of the respective logic. Alas, we are unable to produce any hardness results for this class for any of the logics considered.

$\mathcal{MC}$ .

Since OSM and PSM are in 2EXPTIME and the latter is closed under finite intersections, set  $\mathbf{VI}$  is in 2EXPTIME.

$\mathcal{PML}$ .

We argue as for set  $\mathbf{V}$  to see that deciding set  $\mathbf{VI}$  is in EXPSPACE.

$\mathcal{PL}$ .

Since OSM and PSM are in coNP and coNP is closed under finite intersections, set  $\mathbf{VI}$  is in coNP.

### 3.6 Experimental data

With the decision problems at hand, experimental data are probably not obtainable with ease. Still, we wanted to get a feel for how many formulas of a given size are in OSM and in the sets **V** and **VI**. We used Perl scripts to randomly generate “all” formulas of  $\mathcal{PL}$  in sizes ranging from 1 to 5 where “size” refers to the number of occurrences of logical connectives in the formula. These scripts then performed a brute-force check for membership of sets of interest. This showed that about 75% of those formulas are in OSM and about the same percentage are in PSM, whereas about 50% of formulas were in  $\text{PSM} \cap \text{OSM}$ . Of the formulas generated, only about 2.45% were in the NP-complete set  $\mathcal{PL} \setminus (\text{PSM} \cup \text{OSM})$ . Our results indicate that less formulas are in the latter set as the number of occurrences of logical operators in these formulas increases.

### 3.7 Summary of results

The complexity results shown in this paper are summarized in Figure 5. These results illustrate that we cannot issue any exact complexity results, apart from those for propositional logic for sets PSM and OSM (coNP-complete) and set  $\mathcal{PL} \setminus (\text{PSM} \cup \text{OSM})$  (NP-complete). Our hardness results either exhibit an exponential gap for upper bounds (for  $\mathcal{MC}$  and  $\mathcal{PML}$ ) or a believed gap in the Boolean hierarchy over NP (for  $\mathcal{PL}$ ). One can also see that semantic self-minimization, the question of whether a formula and its negation are in PSM, lacks any hardness results at present.

## 4 Related work

The partial models, their refinement notion, and the compositional semantics for partial models presented in this paper were introduced (for CTL, a fragment of  $\mathcal{MC}$ ) in [7]. Generalized model checking, its complexity analysis, and a model-checking algorithm for it were then presented in [8] for linear-time and branching-time temporal logics. Partial models, their refinement, and temporal logic semantics were already developed for labelled transition systems in [17] and for Kripke structures in [9,10]. Partial versions of models that have labels on transitions as well as on states were discussed in [13]. The notion of semantic minimization, as presented in this paper, was proposed and shown to exist for propositional logic, propositional modal logic, and the propositional modal mu-calculus in [12]. The demonstration that practically relevant temporal logic specifications are by and large pessimistically self-minimizing was given in [5].

Blamey [6] studied partial-valued logics and their applications to linguistics and model theory and proved the existence of semantic minimizations (in our terminology) for propositional logic. The notion of supervaluational meaning was defined and studied by van Fraassen [21]; it is the definitional template for the generalized model checking judgements for temporal logics in this paper. Reps et al. [20] use BDD-based prime-implicant algorithms for a more efficient implementation of the

Results for  $\mathcal{MC}$ :

2EXPTIME, EXPTIME-hard	EXPTIME-complete	2EXPTIME
OSM	VAL	PSM $\cap$ OSM
PSM	UNSAT	
OSM $\setminus$ (VAL $\cup$ PSM)		
PSM $\setminus$ (UNSAT $\cup$ OSM)		
$\mathcal{MC} \setminus$ (PSM $\cup$ OSM)		

Results for  $\mathcal{PML}$ :

EXPSPACE, PSPACE-hard	PSPACE-complete	EXPSPACE
OSM	VAL	PSM $\cap$ OSM
PSM	UNSAT	
OSM $\setminus$ (VAL $\cup$ PSM)		
PSM $\setminus$ (UNSAT $\cup$ OSM)		
$\mathcal{PML} \setminus$ (PSM $\cup$ OSM)		

Results for  $\mathcal{PL}$ :

DP, coNP-hard	NP-complete	coNP-complete	coNP
OSM $\setminus$ (VAL $\cup$ PSM)	$\mathcal{PL} \setminus$ (PSM $\cup$ OSM)	VAL	PSM $\cap$ OSM
PSM $\setminus$ (UNSAT $\cup$ OSM)		UNSAT	
	OSM		
	PSM		

Fig. 5. Complexity results for PSM, OSM, and for the partition induced by VAL and PSM; the three tables present these results for  $\mathcal{MC}$ ,  $\mathcal{PML}$ , and  $\mathcal{PL}$  (respectively).

computation of semantic minimizations in propositional logic.

In [3,2,4], modal and mixed transition systems serve as partial models and the following decision problems are studied:

- (Common implementation) Do finitely many models have a common transition system as refinement? This is EXPTIME-complete for both mixed and modal transition systems [4].
- (Consistency) Does a model have a transition system as a refinement? This is trivial for modal transition systems, and EXPTIME-complete for mixed transition systems [4].
- (Thorough refinement) Are all transition systems that refine one model also refinements of another model? This is EXPTIME-complete for mixed transition systems [4], and only known to be PSPACE-hard for modal transition systems [2,3] at the time of writing.

## 5 Conclusions

We presented two notions of satisfaction for partial state spaces – a precise but expensive one, and a cheap but imprecise one. We then asked how complex it is to decide whether a given property yields the same satisfaction result, for all partial state spaces, in both notions. We showed that this problem is connected to the validity problem of the respective temporal logic but that the actual picture is

more complex. For the propositional modal mu-calculus and propositional modal logic we showed that deciding optimistic and pessimistic self-minimization is at least as hard as the respective validity problems, but that we can show membership of this decision problem only for a complexity class exponentially higher than that. For semantic self-minimization, we could not show a hardness result for any logic considered. For propositional logic we could show that optimistic and pessimistic self-minimization both match the complexity of validity and that the set of formulas that are neither optimistically nor pessimistically self-minimizing matches the complexity of satisfiability. We also discovered that two sets, for which we had exponential gaps for the propositional modal mu-calculus and for propositional modal logic, are coNP-hard sets in DP in the case of propositional logic.

## Acknowledgments

We expressly thank the anonymous referees for their thoughtful comments, which helped to improve the presentation and clarity of this paper. A preliminary version of this paper, with all results reported in this final version, appeared in the workshop proceedings of the Seventh International Workshop on Automated Verification of Critical Systems, held in Oxford, England, from 10-12 September 2007. The work reported here is already published as Chapter 4 of the first author's PhD thesis [1].

## References

- [1] A. Antonik. *Decision problems for partial specifications: empirical and worst-case complexity*. PhD thesis, Imperial College London, Department of Computing, London, United Kingdom, October 2008.
- [2] A. Antonik, M. Huth, K. G. Larsen, U. Nyman, and A. Wasowski. 20 Years of Mixed and Modal Specifications. *Bulletin of the European Association for Theoretical Computer Science* 95:94–129, July 2008.
- [3] A. Antonik, M. Huth, K. G. Larsen, U. Nyman, and A. Wasowski. Complexity of decision problems for mixed and modal specifications. In *Proc. of the 11th International Conference on Foundations of Software Science and Computation Structures*, volume 4962 of *Lecture Notes in Computer Science* pages 112-126. Springer Verlag, March 2008.
- [4] A. Antonik, M. Huth, K. G. Larsen, U. Nyman, and A. Wasowski. EXPTIME-complete Decision Problems for Mixed and Modal Specifications. In *Proc. of the 15th International Workshop On Expressiveness in Concurrency (2008)*, to appear in ENTCS.
- [5] A. Antonik and M. Huth. Efficient Patterns for Model Checking Partial State Spaces in CTL & LTL. *ENTCS* 158:41–57, Elsevier and Science Direct, 2006.
- [6] S. Blamey. *Partial-Valued Logic*. PhD thesis, University of Oxford, Oxford, England, 1980.
- [7] G. Bruns and P. Godefroid. Model Checking Partial State Spaces with 3-Valued Temporal Logics. In *Proc. of the 11th Conference on Computer Aided Verification*, volume 1633 of *Lecture Notes in Computer Science*, pages 274–287. Springer Verlag, July 1999.
- [8] G. Bruns and P. Godefroid. Generalized Model Checking: Reasoning about Partial State Spaces. In *Proc. of the 11th International Conference on Concurrency Theory*, volume 1877 of *Lecture Notes in Computer Science*, pages 168–182. Springer Verlag, August 2000.
- [9] D. Dams. *Abstract interpretation and partition refinement for model checking*. PhD thesis, Technische Universiteit Eindhoven, The Netherlands, 1996.
- [10] D. Dams, R. Gerth, and O. Grumberg. *Abstract interpretation of reactive systems*. ACM TOPLAS, 19:253–291, 1997.
- [11] M. B. Dwyer, G. S. Avrunin, and J. C. Corbett. Patterns in Property Specifications for Finite-state Verification. In *Proc. of the 1999 International Conference on Software Engineering*, pages 411-420, IEEE Computer Society Press, May 1999.



- [12] P. Godefroid and M. Huth. Model Checking Vs. Generalized Model Checking: Semantic Minimizations for Temporal Logics. In *Proc. of LICS'05*, pages 158–167, Chicago, Illinois, 26–29 June 2005. IEEE Computer Society Press.
- [13] M. Huth, R. Jagadeesan, and D. A. Schmidt. Modal transition systems: a foundation for three-valued program analysis. In D. Sands, editor, *Proc. of the 10th European Symposium on Programming*, pages 155–169. Springer Verlag, April 2001.
- [14] S. C. Kleene. *Introduction to Metamathematics*. Van Nostrand, 1952.
- [15] D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.
- [16] O. Kupferman and M. Y. Vardi. Vacuity Detection in Temporal Model Checking. In *Proc. of the 10th IFIP WG 10.5 Advanced Research Working Conference on Correct Hardware Design and Verification*, volume 1703 of *Lecture Notes in Computer Science*, pages 82–96, Springer Verlag, 1999.
- [17] K. G. Larsen and B. Thomsen. A modal process logic. In *Proc. of the 13th Annual Symposium on Logic in Computer Science*, pages 203–210, IEEE Computer Society Press, 1989.
- [18] C. H. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). In *Proc. of the fourteenth annual ACM symposium on Theory of Computing*, pages 255–260, San Francisco, California, ACM Press, 1982.
- [19] C. H. Papadimitriou and D. Wolfe. The Complexity of Facets Resolved. *Journal of Computer and System Sciences* 37:2–13 (1998).
- [20] T. Reps, A. Loginov, and M. Sagiv. Semantic Minimization of 3-Valued Propositional Formulae. In *Proc. of the 17th Annual IEEE Symposium on Logic in Computer Science*, pages 40–51, Copenhagen, Denmark, 22–25 July 2002. IEEE Computer Society Press.
- [21] B. van Fraassen. Singular terms, truth-value gaps, and free logic. *J. Phil.*, 63(17):481–495, September 1966.