Electronic Notes in Theoretical Computer Science 45 (2001)
URL: http://www.elsevier.nl/locate/entcs/volume45.html 23 pages

Model-Checking View-Based Partial Specifications

Michael Huth [1]

Department of Computing
Imperial College of Science, Technology and Medicine
London, United Kingdom

Shekhar Pradhan [2]

Department of Computer Science
University of Maryland
College Park, USA

Department of English & Philosophy
Central Missourri State University
Warrensburg, USA

Abstract

We develop foundations for the view-based specification of software artifacts in first-order logic. Standard notions of models and semantics of first-order logic are generalized to partial models that do not have access to the entire global signature. At the model-theoretic level, this is achieved via the Smyth powerdomain over the semantic universe of elements. At the logical level, we accomplish this via the Smyth powerdomain over the standard two-valued booleans. A refinement notion is developed and its soundness is proved for a fixed set of semantic elements. Standard models and semantics of first-order logic are subsumed by our framework as models in which all signature information is present ("complete models"). We decompose our semantics into a consensus (the union of the Smyth powerdomain) of an optimistic and a pessimistic semantics. That way, we may compute this semantics as a standard model check in first-order logic over a model lifted by the Smyth powerdomain.

[1] Email: M.Huth@doc.ic.ac.uk
[2] Email: pradhan@cs.umd.edu

1 Introduction

The work presented in this paper makes a first step at laying the semantic and algorithmic foundations for property verification over multiple and potentially conflicting models — be they specification descriptions or other abstractions of software artifacts. We mention two important applications for such foundations. In requirement engineering, different stake holders formulate their individual view of the requirements that a software artifact should meet. The requirement elicitation process can therefore be greatly improved if (i) inconsistencies between these views can be detected and if (ii) specified properties can be "verified" at this early stage in the software life cycle [25,24]. These kinds of analyses are also needed in the evaluation of software artifacts in a very late stage of design, implementation, or maintenance. For example, `xlinkit` [23] is a tool that generates rule-based smart links and then analyzes a collection of XML files for their consistency according to a set of rules.

These practical problems provide the context of our theoretical work. This work can be broken down into two steps. In the first step, semantic and algorithmic machinery needs to be developed for verifying in a partial view global properties expressed using predicate and function symbols that the partial view may have no knowledge of. In the second step, semantic and algorithmic machinery needs to be developed to enable reasoning over multiple viewpoints. Although global property verification is done at each partial view, various forms of communication between some of the partial views should provide the means for a more integrated global check of system properties. In this paper we undertake the first step only; the second step will be tackled in future work.

We model views as conventional finite[3] models of first-order logic, except that individual views may not have access to the global signature of all views involved. The question is then how to define a semantics of first-order logic in these views. We use part of the machinery of domain theory [1], notably the Smyth powerdomain, to lift the standard first-order-logic semantics to this partial setting. The Smyth powerdomain turns classical two-valued propositional logic into Kleene's strong interpretation of three-valued logic [19]. Inspired by [2], this three-valued semantics is decomposed into two semantics whose consensus (the union of the Smyth powerdomain) recovers the original semantics. This consensus, together with a translation of formulas into negation normal form, is used to reduce our three-valued semantics to two standard semantics of first-order logic. Thus, three-valued model checking of global properties in a partial model can be reduced to two-valued model checking of standard models of first-order logic.

[3] Our results generalize to infinite models by moving from finite partial orders to dcpos.

Outline of paper

In Section 2, we define partial models that have only limited access to a global signature. Section 3 uses these models to interpret global terms and formulas. This is done from first principles by applying the Smyth powerdomain to the set of semantic elements (for terms) and to the discrete set of ordinary truth values $\{\mathtt{T}, \mathtt{F}\}$ (for formulas). A refinement preorder of partial models is proposed in Section 4 and proved to be sound, provided that the set of semantic elements remains the same. In Section 5, we decompose our semantics into two semantics whose consensus (the union of the Smyth powerdomain) recovers the original semantics. Section 6 uses this consensus, together with a translation of formulas into negation normal form, to reduce our three-valued semantics to two standard semantics of first-order logic. In Section 7, we describe related work and Section 8 concludes.

2 Models of views

First-order logic signature and syntax

We consider a first-order logic with terms t and formulas ϕ defined by the grammar

$$\begin{aligned} &t ::= x \mid f(t, t, \ldots, t) \\ (1)\quad &\phi ::= \top \mid P(t, t, \ldots, t) \mid \neg\phi \mid \phi \wedge \phi \mid \exists x\, \phi, \end{aligned}$$

where x, f, and P range over sets of variables $\mathcal{X}$, function symbols $\mathcal{F}$, and predicate symbols $\mathcal{P}$ (respectively), resulting in the signature

$$\Sigma \stackrel{\text{def}}{=} (\mathcal{X}, \mathcal{F}, \mathcal{P}). \tag{2}$$

In this paper, we write "n" as a generic name for the arities $\mathrm{ar}(f)$ and $\mathrm{ar}(P)$, the number of arguments required for f and P (respectively). To simplify our exposition, we omit sorts, types, and other structuring mechanisms. As customary, we write $\phi_1 \vee \phi_2$ for $\neg(\neg\phi_1 \wedge \neg\phi_2)$, $\phi_1 \rightarrow \phi_2$ for $\neg(\phi_1 \wedge \neg\phi_2)$, and $\forall x\, \phi$ for $\neg\exists x\, \neg\phi$.

Partial models

Fixing a nonempty, finite set of elements $\mathcal{U}$ — the semantic universe for all functions and predicates of signature Σ — we define models that are partial in that they have access to a part of Σ and $\mathcal{U}$ only.

Definition 2.1 [Partial models] A *partial model* $\mathcal{M}_v$ *for* $(\Sigma, \mathcal{U})$ consists of

(i) a signature $\Sigma_v = (\mathcal{X}_v, \mathcal{F}_v, \mathcal{P}_v)$ such that $\mathcal{X}_v \subseteq \mathcal{X}$, $\mathcal{F}_v \subseteq \mathcal{F}$, and $\mathcal{P}_v \subseteq \mathcal{P}$;

(ii) a nonempty set of elements $\mathcal{U}_v$ with $\mathcal{U}_v \subseteq \mathcal{U}$;

(iii) for all $f \in \mathcal{X}_v$, a (total) function $f^v : \mathcal{U}_v^n \rightarrow \mathcal{U}_v$; and

(iv) for all $P \in \mathcal{P}_v$, a relation $P^v \subseteq \mathcal{U}_v^n$.

The parameter v in $\mathcal{M}_v$ indicates that this model is a particular "view" of models of sort Σ. Such views may be partial, for some of the inclusions for

sorts (e.g. $\mathcal{X}_v \subseteq \mathcal{X}$) can be strict. We point out that different views may well have the same signature, making inconsistencies a definite possibility. Also observe that $\mathcal{U}_v$ may be a strict subset of $\mathcal{U}$, e.g. the set of all even natural numbers below 100, where $\mathcal{U}$ is the set of all natural numbers below 100.

3 Semantics of views

Smyth powerdomain

We define a semantics of partial models that conservatively extends the usual semantics for first-order logic. In doing so, we employ the Smyth power domain [1], thereby rediscovering Kleene's strong three-valued interpretation of (propositional) logic [19].

Definition 3.1 [Smyth powerdomain functor [28,1]]

(i) For a finite partial order $(P, \leq)$, the *Smyth powerdomain of* P, $\mathbb{P}_s(P)$, is the collection of all nonempty upper sets[4] of P, ordered by *reverse* inclusion:

$$U_1 \sqsubseteq U_2 \text{ in } \mathbb{P}_s(P) \qquad \text{iff} \qquad U_2 \subseteq U_1. \tag{3}$$

(ii) For a monotone function $f\colon P \to Q$ between finite partial orders, we define the (total) function $\mathbb{P}_s(f)\colon \mathbb{P}_s(P) \to \mathbb{P}_s(Q)$ as

$$\mathbb{P}_s(f)(U) \stackrel{\text{def}}{=} \{q \in Q \mid \text{ for some } u \in U,\ f(u) \leq q\}. \tag{4}$$

Remark 3.2 [Universal property of $\mathbb{P}_s(\cdot)$] $\mathbb{P}_s(\cdot)$ is a functor on the category of finite partial orders and monotone functions. For any finite partial order P, the finite partial order $\mathbb{P}_s(P)$ is universal in the following sense: Let Q be any finite partial order, $f\colon P \to Q$ any monotone function, and $\epsilon_P\colon P \to \mathbb{P}_s(P)$ defined by $\epsilon_P(p) \stackrel{\text{def}}{=} \{p' \in P \mid p \leq p'\}$. Then there exists a unique monotone function $\bar{f}\colon \mathbb{P}_s(P) \to Q$ such that

$$f = \bar{f} \circ \epsilon_P \text{ and } \bar{f}(U_1 \cup U_2) = \bar{f}(U_1) \cup \bar{f}(U_2) \text{ for all } U_1, U_2 \in \mathbb{P}_s(P). \tag{5}$$

In particular, $\mathbb{P}_s(f)$ equals $\overline{\epsilon_Q \circ f}$.

Example 3.3 (i) For the discrete partial order $\mathbb{B} \stackrel{\text{def}}{=} \{\mathtt{F}, \mathtt{T}\}$, the Smyth power domain $\mathbb{P}_s(\mathbb{B})$ has three elements

$$\{\mathtt{F}, \mathtt{T}\},\ \{\mathtt{F}\},\ \text{and } \{\mathtt{T}\}, \tag{6}$$

where the first one is the least element and the other two are maximal elements. Note that $\mathbb{P}_s(\mathbb{B})$ is isomorphic to $\mathbb{P}_s(\mathbb{P}_s(\mathbb{B}))$.

(ii) For a finite set $\mathcal{U}_v$, the Smyth power domain $\mathbb{P}_s(\mathcal{U}_v)$ consists of all nonempty subsets of $\mathcal{U}_v$ ordered under reverse inclusion. In particular, the least element of $\mathbb{P}_s(\mathcal{U}_v)$ is $\mathcal{U}_v$, and its maximal elements are all singleton sets $\{u\}$ $(u \in \mathcal{U}_v)$.

[4] An upper set of P is a subset U of P such that $x \in U$ and $x \leq y$ imply $y \in U$.

(iii) If the function $=^{\mathbb{B}}: \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ encodes classical equality of truth values, then $\mathbb{P}_{\mathrm{s}}(=^{\mathbb{B}})$ is a conservative extension of $=^{\mathbb{B}}$ from $\mathbb{B}$ to $\mathbb{P}_{\mathrm{s}}(\mathbb{B})$ which returns $\mathtt{U}$ iff at least one of its arguments equals $\mathtt{U}$.

Definition 3.4 [Propositional logic on $\mathbb{P}_{\mathrm{s}}(\mathbb{B})$] Let $\neg^{\mathbb{B}}$, $\wedge^{\mathbb{B}}$, and $\bigvee^{\mathbb{B}}$ be the standard interpretations of negation, conjunction, and $n$ary disjunction on $\mathbb{B}$. We define the monotone functions [5]

$$\neg^s \stackrel{\text{def}}{=} \mathbb{P}_{\mathrm{s}}(\neg^{\mathbb{B}}) \qquad \wedge^s \stackrel{\text{def}}{=} \mathbb{P}_{\mathrm{s}}(\wedge^{\mathbb{B}}) \qquad \bigvee^s \stackrel{\text{def}}{=} \mathbb{P}_{\mathrm{s}}(\bigvee^{\mathbb{B}}) \tag{7}$$

as our interpretation of propositional logic on $\mathbb{P}_{\mathrm{s}}(\mathbb{B})$.

This categorical semantics turns out to be Kleene's strong interpretation of three-valued propositional logic [19]. We write $\mathtt{U}$ ("unknown") for the set $\{\mathtt{F}, \mathtt{T}\}$ and identify $\mathtt{T}$ with $\{\mathtt{T}\}$ and $\mathtt{F}$ with $\{\mathtt{F}\}$ whenever this causes no confusion.

Proposition 3.5 (Characterization of propositional logic in $\mathbb{P}_{\mathrm{s}}(\mathbb{B})$) (i) *The function $\neg^s$ maps $\mathtt{T}$ to $\mathtt{F}$, $\mathtt{F}$ to $\mathtt{T}$, and $\mathtt{U}$ to $\mathtt{U}$.*

(ii) *The function $x \wedge^s y$ returns $\mathtt{F}$ if x or y equals $\mathtt{F}$; otherwise, if x or y equals $\mathtt{U}$, it returns $\mathtt{U}$; otherwise, it returns $\mathtt{T}$.*

(iii) *The expression $\bigvee_j^s x_j$ returns $\mathtt{T}$ if x_j equals $\mathtt{T}$ for some j with $1 \leq j \leq n$; otherwise, it returns $\mathtt{U}$ if $x_{j'}$ equals $\mathtt{U}$ for some j' with $1 \leq j' \leq n$; otherwise, it returns $\mathtt{F}$.*

Proof. Most of this follows from the universal property of the Smyth powerdomain. We show some cases for sake of illustration.

(i) The set $\neg^s(\{\mathtt{F}, \mathtt{T}\})$ equals all the elements in $\mathbb{B}$ that are greater than or equal to $\neg^{\mathbb{B}}(\mathtt{F})$ or $\neg^{\mathbb{B}}(\mathtt{T})$. Thus, this set is $\{\mathtt{F}, \mathtt{T}\}$.

(ii) The set $\{\mathtt{F}, \mathtt{T}\} \wedge^s \{\mathtt{T}\}$ equals $\{\mathtt{F} \wedge^{\mathbb{B}} \mathtt{T}, \mathtt{T} \wedge^{\mathbb{B}} \mathtt{T}\}$, which is $\{\mathtt{F}, \mathtt{T}\}$.

(iii) Let $x_{i_0} = \{\mathtt{F}, \mathtt{T}\}$ and $x_i \neq \{\mathtt{T}\}$ for all $i \neq i_0$. Then $\bigvee^s x_i$ equals $\{\mathtt{F}, \mathtt{T}\}$: we obtain $\mathtt{F}$ by choosing $\mathtt{F}$ from each x_i, including x_{i_0}; we get $\mathtt{T}$ by choosing $\mathtt{T}$ from x_{i_0}.

□

Semantics of global terms

For a partial model $\mathcal{M}_v$ as in Definition 2.1, we define a semantics $[\![t]\!]^v_\rho$ as an element of $\mathbb{P}_{\mathrm{s}}(\mathcal{U}_v)$ for terms t of the *global* signature Σ, where ρ is a *local* environment — a partial function from $\mathcal{X}_v$ to $\mathcal{U}_v$ that binds variables to

[5] These functions are well defined since any function on the discrete set $\mathbb{B}$ is monotone.

concrete elements. For variables $x \in \mathcal{X}$, we define [6]

$$\llbracket\ x\ \rrbracket_\rho^v \stackrel{\text{def}}{=} \begin{cases} \{\rho(x)\} \text{ if } x \in \mathcal{X}_v \\ \mathcal{U}_v \qquad \text{otherwise.} \end{cases} \tag{8}$$

Thus, we model the fact that the name x is not known in this view by evaluating it to "possibly any element in this view". For function symbols, we proceed similarly:

$$\llbracket\ f(t_1,\ldots,t_n)\ \rrbracket_\rho^v \stackrel{\text{def}}{=} \begin{cases} \{f^v(u_1,\ldots,u_n) \mid u_i \in \llbracket\ t_i\ \rrbracket_\rho^v\} \text{ if } f \in \mathcal{F}_v \\ \mathcal{U}_v \qquad\qquad\qquad\qquad\qquad \text{otherwise.} \end{cases} \tag{9}$$

If $f \in \mathcal{F}_v$, then the semantics of $f(t_1,\ldots,t_n)$ in (9) is obtained by applying $\mathbb{P}_{\mathrm{s}}(\cdot)$ to the interpretation f^v of type $\mathcal{U}_v^n \to \mathcal{U}_v$ in the partial model. Otherwise, we interpret f as a constant function whose image is the least element of $\mathbb{P}_{\mathrm{s}}(\mathcal{U}_v)$.

Example 3.6 Let $\mathcal{U}_v$ be the set of even natural numbers below 100, including zero. Suppose that $+^v$ and $*^v$ denote the usual operations of addition and multiplication (respectively), except that they are executed "modulo 100" (e.g. $*^v(46,78) = 24$ and $*^v(78,0) = 0$). Let $x \notin \mathcal{X}_v$. Then $\llbracket\ +(6,x)\ \rrbracket_\rho^v$ equals $\{6 + u \,(\mathrm{mod}\, 100) \mid u \in \mathcal{U}_v\}$, which is $\mathcal{U}_v$; whereas $\llbracket\ *(0,x)\ \rrbracket_\rho^v$ equals $\{0\}$.

Semantics of global formulas

The meaning of a global formula is an element of the Smyth power domain $\mathbb{P}_{\mathrm{s}}(\mathbb{B})$ of $\mathbb{B}$: it is `T`, if the formula is definitely true; `F` if the formula is definitely false; and `U` if the truth value of the formula cannot be determined, due to the partiality of the model. Clearly,

$$\llbracket\ \top\ \rrbracket_\rho^v \stackrel{\text{def}}{=} \mathtt{T}. \tag{10}$$

Since the meaning of a partial term is a nonempty subset of the local universe of semantic elements, we need to lift the relations that interpret predicate symbols to subsets:

$$\llbracket\ P(t_1,\ldots,t_n)\ \rrbracket_\rho^v \stackrel{\text{def}}{=} \begin{cases} \mathtt{T} \text{ if } P \in \mathcal{P}_v \ \&\ [\prod_i \llbracket\ t_i\ \rrbracket_\rho^v \subseteq P^v] \\ \mathtt{F} \text{ if } P \in \mathcal{P}_v \ \&\ [\prod_i \llbracket\ t_i\ \rrbracket_\rho^v \cap P^v = \emptyset] \\ \mathtt{U} \text{ otherwise.} \end{cases} \tag{11}$$

Note that the side conditions in (11) can be checked by automated theorem provers, e.g. SVC or PVS, if the sets $\llbracket\ t_i\ \rrbracket_\rho^v$ and P^v can be expressed as quantifier-free formulas of first-order logic. In particular, such checks may allow the evaluation of properties over *infinite* partial models. If $P \in \mathcal{P}_v$,

[6] In computing $\llbracket\ t\ \rrbracket_\rho^v$ and $\llbracket\ \phi\ \rrbracket_\rho^v$, we make the standard assumption that ρ is defined for all free variables of t and ϕ that are contained in $\mathcal{X}_v$.

then (11) computes $\mathbb{P}_{\mathrm{s}}(P^v)$, where we identify the relation P^v with its characteristic function of type $\mathcal{U}_v^n \to \mathbb{B}$. The interpretation of the remaining connectives uses our propositional logic on $\mathbb{P}_{\mathrm{s}}(\mathbb{B})$:

$$(12) \qquad [\![\neg\phi]\!]^v_\rho \stackrel{\text{def}}{=} \neg^s [\![\phi]\!]^v_\rho$$

$$(13) \quad [\![\phi_1 \wedge \phi_2]\!]^v_\rho \stackrel{\text{def}}{=} [\![\phi_1]\!]^v_\rho \wedge^s [\![\phi_2]\!]^v_\rho$$

$$(14) \qquad [\![\exists x\, \phi]\!]^v_\rho \stackrel{\text{def}}{=} \bigvee_{u \in \mathcal{U}_v}^{s} [\![\phi]\!]^v_{\rho[x \mapsto u]},$$

where $\rho[x \mapsto u]$ contains the same bindings as ρ, except that it binds x to u. In (14), it does not matter whether x is in $\mathcal{X}_v$ since $\exists x$ binds the name x and thereby makes it "locally known" by extending the current environment ρ.

Example 3.7 Revisiting Example 3.6, let $>^v$ be the standard interpretation "strictly greater than" of $>$ on $\mathcal{U}_v$. The formula $(6 + x) > 4$ evaluates to $\mathtt{U}$ and the formula $(0 * x) > 2$ evaluates to $\mathtt{F}$. The reason for the former is that in (11) some instances of $P^v(u_1, \ldots, u_n)$ hold (e.g. $(6 + 0) >^v 4$), but not all of them (e.g. not $(6 + 98) >^v 4$).

We show that our semantics for partial models coincides with the standard semantics of first-order logic for all "complete" partial models, which are understood to be those partial models $\mathcal{M}_v$, where Σ_v equals Σ. That is, these are just the ordinary models of first-order logic for the signature Σ.

Proposition 3.8 (Conservative extension of first-order logic) *Let $\mathcal{M}_v$ be a partial model for $(\Sigma, \mathcal{U})$ such that $\Sigma_v = \Sigma$. For all ϕ and ρ, the set $[\![\phi]\!]^v_\rho$ is a singleton $\{d\}$, where $d \in \{\mathtt{T}, \mathtt{F}\}$. Moreover, the truth value d coincides with the usual first-order logic semantics of ϕ in the model $\mathcal{M}_v$.*[7]

Proof.

- Since $\Sigma_v = \Sigma$, a structural induction on t shows that $[\![t]\!]^v_\rho$ is a singleton and that its sole element is the standard denotation of term t in environment ρ of model $\mathcal{M}_v$.
- Since all sets $[\![t]\!]^v_\rho$ are singletons, (11) entails that $[\![P(t_1, \ldots, t_n)]\!]^v_\rho$ cannot be $\mathtt{U}$. From (11) and the fact that the truth value it contains matches the standard first-order semantics, this match holds for all atomic formulas.
- Finally, the functions $\neg^s$, $\wedge^s$, and $\bigvee^s$ all restrict to the set

  $$(15) \qquad \{\{\mathtt{T}\}, \{\mathtt{F}\}\}$$

  of maximal elements of $\mathbb{P}_{\mathrm{s}}(\mathbb{B})$; on that set they coincide with the standard semantics of negation, conjunction, and $n$ary disjunction: $\neg^{\mathbb{B}}$, $\wedge^{\mathbb{B}}$, and $\bigvee^{\mathbb{B}}$ (respectively).

□

[7] I.e. $d = \mathtt{T}$ iff $\mathcal{M}_v \models \phi$ holds for the standard satisfaction relation $\models$ of first-order logic.

4 Refinement of views

One partial view can be a refinement of another partial view. Intuitively, this means that the refining view encodes more precise knowledge of a software system without contradicting the knowledge of the refined view. We say that a semantics of partial views is sound with respect to refinement if it supports this property of the refinement relation. In Definition 4.1 we formalize the idea that the refining view encodes more precise knowledge of the software system in question. In Proposition 4.5 below we show that the refining view does not contradict the knowledge of the refined view if the domains of semantic elements of the two views are the same and, thus, establish that our semantics is sound with respect to the refinement relation under the specified circumstances.

Definition 4.1 [Refinement of partial models] Let $\mathcal{M}_{v_1}$ and $\mathcal{M}_{v_2}$ be two partial models for $(\Sigma,\mathcal{U})$ with signatures $\Sigma_{v_1} = (\mathcal{X}_{v_1},\mathcal{F}_{v_1},\mathcal{P}_{v_1})$ and $\Sigma_{v_2} = (\mathcal{X}_{v_2},\mathcal{F}_{v_2},\mathcal{P}_{v_2})$ (respectively). Then $\mathcal{M}_{v_1}$ *refines* $\mathcal{M}_{v_2}$, denoted by $\mathcal{M}_{v_1} \sqsubseteq \mathcal{M}_{v_2}$, iff

(i) the signature and universe of $\mathcal{M}_{v_1}$ extend those of $\mathcal{M}_{v_2}$: $\mathcal{X}_{v_2} \subseteq \mathcal{X}_{v_1}$, $\mathcal{F}_{v_2} \subseteq \mathcal{F}_{v_1}$, $\mathcal{P}_{v_2} \subseteq \mathcal{P}_{v_1}$, and $\mathcal{U}_{v_2} \subseteq \mathcal{U}_{v_1}$; and

(ii) $\mathcal{M}_{v_1}$ conservatively extends the semantics of functions and predicates that are in the signature of $\mathcal{M}_{v_2}$:

(16) $f \in \mathcal{F}_{v_2}, u_i \in \mathcal{U}_{v_2} \Rightarrow f^{v_2}(u_1,\ldots,u_n) = f^{v_1}(u_1,\ldots,u_n)$
(17) $P \in \mathcal{P}_{v_2}, u_i \in \mathcal{U}_{v_2} \Rightarrow (u_1,\ldots,u_n) \in P^{v_2}$ iff $(u_1,\ldots,u_n) \in P^{v_1}$.

Example 4.2 The partial model $\mathcal{M}_v$ from Example 3.6 is refined by the partial model $\mathcal{M}_{v'}$, where $\Sigma_{v'}$ equals Σ_v, $\mathcal{U}_{v'}$ is the set of *all* natural numbers below 100, and $+^{v'}$ and $*^{v'}$ are the interpretations of addition and multiplication "modulo 100" (respectively). We can further refine $\mathcal{M}_{v'}$ to $\mathcal{M}_{v''}$ by making x an element of $\mathcal{X}_{v''}$.

Lemma 4.3 *The relation $\sqsubseteq$ is a preorder on the set of all partial models for Σ.*

Proof. For sake of illustration, we only show the transitivity of condition (16). Let $\mathcal{M}_{v_1} \sqsubseteq \mathcal{M}_{v_2}$ and $\mathcal{M}_{v_2} \sqsubseteq \mathcal{M}_{v_3}$. Given $f \in \mathcal{F}_{v_3}$ and $u_i \in \mathcal{U}_{v_3}$, we infer (i) $u_i \in \mathcal{U}_{v_2}$, (ii) $f \in \mathcal{F}_{v_2}$, and (iii) $f^{v_3}(u_1,\ldots,u_n) = f^{v_2}(u_1,\ldots,u_n)$ from $\mathcal{M}_{v_2} \sqsubseteq \mathcal{M}_{v_3}$. But then $\mathcal{M}_{v_1} \sqsubseteq \mathcal{M}_{v_2}$ and (i)–(iii) imply $f^{v_2}(u_1,\ldots,u_n) = f^{v_1}(u_1,\ldots,u_n)$, and so $f^{v_3}(u_1,\ldots,u_n) = f^{v_1}(u_1,\ldots,u_n)$ follows. □

Example 4.4 [Unsound semantics] Our semantics is *not* sound in the sense that $\mathcal{M}_{v_1} \sqsubseteq \mathcal{M}_{v_2}$ does not imply $[\![\,\phi\,]\!]^{v_2}_{\rho} \sqsubseteq [\![\,\phi\,]\!]^{v_1}_{\rho}$ for all ϕ. The latter is a way of expressing the requirement that v_1 has more precise knowledge of the software system than v_2, but that this knowledge is consistent with that of v_2.

In Example 4.2, we saw that $\mathcal{M}_v$ is refined by the model $\mathcal{M}_{v''}$. The formula [8]

$$\exists x\,(98 < x) \tag{18}$$

evaluates to F in $\mathcal{M}_v$ since 98 is the maximal element of $\mathcal{U}_v$, but it evaluates to T in the refining model $\mathcal{M}_{v''}$ since it has 99 as a witness. Note that this could only happen because $\mathcal{M}_v$ had a somewhat limited view of the natural numbers contained in the interval $[0, 99]$.

Naturally, this unsoundness is a good thing as it allows us to detect inconsistencies between these views even without having any policies regarding the priorities of these views. However, one can prove the soundness of our semantics if all views share the same set of elements $\mathcal{U}$. Then the sets $[\![\, t \,]\!]^v_\rho$ and $[\![\, \phi \,]\!]^v_\rho$ computed by a partial model $\mathcal{M}_v$ are sound approximations (supersets) of the corresponding sets computed by partial models that refine $\mathcal{M}_v$.

Proposition 4.5 (Soundness of semantics) *Let $\mathcal{M}_{v_i}$ be partial models for $(\Sigma, \mathcal{U})$ with signatures Σ_{v_i} $(i = 1, 2)$ such that $\mathcal{M}_{v_1} \sqsubseteq \mathcal{M}_{v_2}$ and $\mathcal{U}_{v_1} = \mathcal{U}_{v_2}$. Let ρ be an environment for $\mathcal{M}_{v_2}$.*

(i) *For all terms t over Σ, $[\![\, t \,]\!]^{v_2}_\rho \sqsubseteq [\![\, t \,]\!]^{v_1}_\rho$ in $\mathbb{P}_{\mathrm{s}}(\mathcal{U}_{v_2})$.*

(ii) *For all formulas ϕ over Σ, $[\![\, \phi \,]\!]^{v_2}_\rho \sqsubseteq [\![\, \phi \,]\!]^{v_1}_\rho$ in $\mathbb{P}_{\mathrm{s}}(\mathbb{B})$.*

Proof. Since $\mathcal{X}_{v_2} \subseteq \mathcal{X}_{v_1}$ and $\mathcal{U}_{v_2} \subseteq \mathcal{U}_{v_1}$, we conclude that ρ can be cast to an environment for $\mathcal{M}_{v_1}$ as well.

(i) We consider variables x:
- If $x \notin \mathcal{X}_{v_1}$, then $x \notin \mathcal{X}_{v_2}$ as well, and so $[\![\, x \,]\!]^{v_2}_\rho = \mathcal{U}_{v_2} = [\![\, x \,]\!]^{v_1}_\rho$.
- If $x \in \mathcal{X}_{v_1}$, then $[\![\, x \,]\!]^{v_1}_\rho$ equals $\{\rho(x)\}$ which is contained in $\mathcal{U}_{v_2}$. But $[\![\, x \,]\!]^{v_2}_\rho$ can only equal $\{\rho(x)\}$ or $\mathcal{U}_{v_2}$.

(ii) As for general terms t:
- If $f \notin \mathcal{F}_{v_1}$, then $f \notin \mathcal{F}_{v_2}$ as well, and so $[\![\, f(t_1, \ldots, t_n \,]\!]^{v_2}_\rho = \mathcal{U}_{v_2} = [\![\, f(t_1, \ldots, t_n) \,]\!]^{v_1}_\rho$, since $\mathcal{U}_{v_1} = \mathcal{U}_{v_2}$.
- If $f \in \mathcal{F}_{v_1}$, we have two cases. First, if $f \in \mathcal{F}_{v_2}$, then (16) and $\mathcal{U}_{v_1} = \mathcal{U}_{v_2}$ imply that the functions f^{v_1} and f^{v_1} are identical. Thus,

$$\begin{aligned} [\![\, f(t_1, \ldots, t_n) \,]\!]^{v_2}_\rho &= \{f^{v_2}(u_1, \ldots, u_n) \mid u_i \in [\![\, t_i \,]\!]^{v_2}_\rho\} \\ &\supseteq \{f^{v_2}(u_1, \ldots, u_n) \mid u_i \in [\![\, t_i \,]\!]^{v_1}_\rho\} && \text{by ind.} \\ &= \{f^{v_1}(u_1, \ldots, u_n) \mid u_i \in [\![\, t_i \,]\!]^{v_1}_\rho\} && \text{as } \mathcal{M}_{v_1} \sqsubseteq \mathcal{M}_{v_2} \\ &= [\![\, f(t_1, \ldots, t_n) \,]\!]^{v_1}_\rho. \end{aligned} \tag{19}$$

  Second, if $f \notin \mathcal{F}_{v_2}$, then $[\![\, f(t_1, \ldots, t_n) \,]\!]^{v_2}_\rho = \mathcal{U}_{v_2}$ is the least element of $\mathbb{P}_{\mathrm{s}}(\mathcal{U}_{v_2})$, so there is nothing to show.

(iii) For formulas, the statement is clear for $\top$.

(iv) Consider $P(t_1, \ldots, t_n)$.

[8] As customary, we identify natural numbers with their corresponding constants in the logic.

- If $P \notin \mathcal{P}_{v_1}$, then $P \notin \mathcal{P}_{v_2}$ as well, and so $[\![P(t_1, \ldots, t_n]\!]_\rho^{v_2} = \mathtt{U} = [\![P(t_1, \ldots, t_n)]\!]_\rho^{v_1}$.
- If $P \in \mathcal{P}_{v_1}$, we have two cases.
  - If $P \in \mathcal{P}_{v_2}$, then (17) and $\mathcal{U}_{v_1} = \mathcal{U}_{v_2}$ imply that the relations P^{v_1} and P^{v_1} are identical. Assume that $[\![P(t_1, \ldots, t_n)]\!]_\rho^{v_2} = \mathtt{T}$. Inspecting (11) and noting that the $[\![t_i]\!]_\rho^{v_2}$ are supersets of $[\![t_i]\!]_\rho^{v_1}$, this implies $[\![P(t_1, \ldots, t_n)]\!]_\rho^{v_1} = \mathtt{T}$ as well. Similarly, from these inclusions and (11) we infer that $[\![P(t_1, \ldots, t_n)]\!]_\rho^{v_2}$ equals $\mathtt{F}$ whenever $[\![P(t_1, \ldots, t_n)]\!]_\rho^{v_2} = \mathtt{F}$. Finally, if $[\![P(t_1, \ldots, t_n)]\!]_\rho^{v_2} = \mathtt{U}$, there is nothing to show.
  - If $P \notin \mathcal{P}_{v_2}$, then $[\![P(t_1, \ldots, t_n)]\!]_\rho^{v_2} = \mathtt{U}$, so there is nothing to show.

(v) For the remaining connectives $\neg$, $\wedge$, and $\exists x$ we use induction and the fact that the corresponding operations $\neg^s$, $\wedge^s$, and $\bigvee^s$ are monotone on $\mathbb{P}_{\mathrm{s}}(\mathbb{B})$.

□

5 View semantics as consensus

As the semantics $[\![\phi]\!]_\rho^v$ is three-valued it might seem as if it cannot be computed using the techniques of standard first-order logic. However, we show below how the semantics $[\![\phi]\!]_\rho^v$ can be computed in such a standard manner. This is accomplished by splitting this semantics into two two-valued ones whose consensus recovers the original semantics. This method has already been applied successfully by Bruns & Godefroid in [2] for partial Kripke structures and in [8] for modal transition systems in the context of model checking partial state spaces. [9] Specifically, we define two functions Opt^s and Pess^s of type $\mathbb{P}_{\mathrm{s}}(\mathbb{B}) \to \mathbb{P}_{\mathrm{s}}(\mathbb{B})$ by

$$\mathrm{Pess}^s(x) \stackrel{\mathrm{def}}{=} \bigwedge^{\mathbb{B}} x \qquad \mathrm{Opt}^s(x) \stackrel{\mathrm{def}}{=} \bigvee^{\mathbb{B}} x. \tag{20}$$

Note that these functions leave $\mathtt{T}$ and $\mathtt{F}$ fixed, but promote $\mathtt{U}$ to a proper truth value:

$$\mathrm{Pess}^s(\mathtt{U}) = \mathtt{F} \qquad \mathrm{Opt}^s(\mathtt{U}) = \mathtt{T}. \tag{21}$$

These functions are used to cast $\mathtt{U}$ values, arising from the evaluation of atomic formulas $P(t_1, \ldots, t_n)$, to proper truth values; their duality is expressed in the equations

$$\mathrm{Pess}^s = \neg^s \circ \mathrm{Opt}^s \circ \neg^s \qquad \mathrm{Opt}^s = \neg^s \circ \mathrm{Pess}^s \circ \neg^s. \tag{22}$$

Thus, we arrive at two semantics defined in Figure 1. Following Kelb [18], the treatment of negation switches the mode of evaluation from o(v) to p(v) or vice versa. We now show that $[\![\cdot]\!]_\rho^v$ can be reconstructed from $[\![\cdot]\!]_\rho^{\mathrm{o}(v)}$ and $[\![\cdot]\!]_\rho^{\mathrm{p}(v)}$ with set-theoretic union as a "consensus operator".

[9] Our results can be redeveloped for a first-order logic extended with a transitive-closure operator, a crucial extension of first-order logic to a realistic specification language.

$$
\begin{aligned}
[\![P(t_1,\ldots,t_n)]\!]^{\mathrm{o}(v)}_{\rho} &\stackrel{\mathrm{def}}{=} \mathrm{Opt}^v([\![P(t_1,\ldots,t_n)]\!]^{v}_{\rho}) \\
[\![\neg\phi]\!]^{\mathrm{o}(v)}_{\rho} &\stackrel{\mathrm{def}}{=} \neg^s [\![\phi]\!]^{\mathrm{p}(v)}_{\rho} \\
[\![\phi_1 \wedge \phi_2]\!]^{\mathrm{o}(v)}_{\rho} &\stackrel{\mathrm{def}}{=} [\![\phi_1]\!]^{\mathrm{o}(v)}_{\rho} \wedge^s [\![\phi_2]\!]^{\mathrm{o}(v)}_{\rho} \\
[\![\exists x\, \phi]\!]^{\mathrm{o}(v)}_{\rho} &\stackrel{\mathrm{def}}{=} \bigvee_{u \in U}^{s} [\![\phi]\!]^{\mathrm{o}(v)}_{\rho[x \mapsto u]}
\end{aligned}
$$

$$
\begin{aligned}
[\![P(t_1,\ldots,t_n)]\!]^{\mathrm{p}(v)}_{\rho} &\stackrel{\mathrm{def}}{=} \mathrm{Pess}^v([\![P(t_1,\ldots,t_n)]\!]^{v}_{\rho}) \\
[\![\neg\phi]\!]^{\mathrm{p}(v)}_{\rho} &\stackrel{\mathrm{def}}{=} \neg^s [\![\phi]\!]^{\mathrm{o}(v)}_{\rho} \\
[\![\phi_1 \wedge \phi_2]\!]^{\mathrm{p}(v)}_{\rho} &\stackrel{\mathrm{def}}{=} [\![\phi_1]\!]^{\mathrm{p}(v)}_{\rho} \wedge^s [\![\phi_2]\!]^{\mathrm{p}(v)}_{\rho} \\
[\![\exists x\, \phi]\!]^{\mathrm{p}(v)}_{\rho} &\stackrel{\mathrm{def}}{=} \bigvee_{u \in U}^{s} [\![\phi]\!]^{\mathrm{p}(v)}_{\rho[x \mapsto u]}.
\end{aligned}
$$

Fig. 1. Optimistic and pessimistic denotational semantics for first-order logic formulas in environment ρ and partial model $\mathcal{M}_v$.

Proposition 5.1 (Kleene's alignment operator [19]) *Kleene's alignment operation may be defined as a function of type* $\mathbb{P}_{\mathrm{s}}(\mathbb{B}) \times \mathbb{P}_{\mathrm{s}}(\mathbb{B}) \to \mathbb{P}_{\mathrm{s}}(\mathbb{B})$*: for an input pair* (x, y) *it returns* x *in case that* x *equals* y*; otherwise, it returns* $\mathtt{U}$*. This function is simply the* binary union operator

$$\cup : \mathbb{P}_{\mathrm{s}}(\mathbb{B}) \times \mathbb{P}_{\mathrm{s}}(\mathbb{B}) \to \mathbb{P}_{\mathrm{s}}(\mathbb{B}). \tag{23}$$

Proof. We have $\{\mathtt{T}\} \cup \{\mathtt{T}\} = \{\mathtt{T}\}$ and $\{\mathtt{F}\} \cup \{\mathtt{F}\} = \{\mathtt{F}\}$. But $x \cup y$ equals $\{\mathtt{F}, \mathtt{T}\}$ whenever the set x is different from the set y in $\mathbb{P}_{\mathrm{s}}(\mathbb{B})$. □

The reconstruction of $[\![\cdot]\!]^{v}_{\rho}$ as a consensus of $[\![\cdot]\!]^{\mathrm{p}(v)}_{\rho}$ and $[\![\cdot]\!]^{\mathrm{o}(v)}_{\rho}$ requires that we prove the *consistency* of $[\![\cdot]\!]^{\mathrm{p}(v)}_{\rho}$. The optimistic semantics $[\![\cdot]\!]^{\mathrm{o}(v)}_{\rho}$, however, cannot offer such a consistency for a partial model $\mathcal{M}_v$ in general, since $[\![\cdot]\!]^{\mathrm{o}(v)}_{\rho}$ records what kind of properties *may* hold for complete models that refine $\mathcal{M}_v$: it is often the case that there exist such refining models $\mathcal{M}_{v'}$ and $\mathcal{M}_{v''}$, where $[\![\phi]\!]^{\mathrm{o}(v')}_{\rho} = \mathtt{T}$ and $[\![\neg\phi]\!]^{\mathrm{o}(v'')}_{\rho} = \mathtt{T}$, resulting in $[\![\phi \wedge \neg\phi]\!]^{\mathrm{o}(v)}_{\rho} = \mathtt{T}$.[10] In this context, it is useful to think of $[\![\phi]\!]^{\mathrm{p}(v)}_{\rho} = \mathtt{T}$ as an *under*approximation of "ϕ holds in all complete models that refine $\mathcal{M}_v$, but retain $\mathcal{U}_v$", whereas $[\![\phi]\!]^{\mathrm{o}(v)}_{\rho} = \mathtt{T}$ is an *over*approximation of "ϕ holds in some complete models that refines $\mathcal{M}_v$, but retains $\mathcal{U}_v$".

Theorem 5.2 (Consistency of $[\![\cdot]\!]^{\mathrm{p}(v)}_{\rho}$) *Let* $\mathcal{M}_v$ *be a partial model,* ϕ *a global formula, and* ρ *a local environment. Then*

$$[\![\phi \wedge \neg\phi]\!]^{\mathrm{p}(v)}_{\rho} = \mathtt{F}. \tag{24}$$

[10] This is the price we pay for defining the meaning of $\wedge$ ($\vee$) via $\wedge^s$ ($\vee^s$) for $\mathrm{p}(v)$ ($\mathrm{o}(v)$). One could conceivably use the ideas of [3] to improve on this.

Proof. By structural induction on ϕ. By definition, it suffices to show that $[\![\phi]\!]^{\mathrm{p}(v)}_{\rho} \wedge^s \neg^s [\![\phi]\!]^{\mathrm{o}(v)}_{\rho} = \mathtt{F}$. Throughout the proof, we use that $[\![\phi]\!]^{\mathrm{p}(v)}_{\rho}$ and $[\![\phi]\!]^{\mathrm{o}(v)}_{\rho}$ always compute a value in $\mathbb{B}$.

- We compute $[\![\top]\!]^{\mathrm{p}(v)}_{\rho} \wedge^s \neg^s [\![\top]\!]^{\mathrm{o}(v)}_{\rho} = \mathtt{T} \wedge^s \neg^s \mathtt{T} = \mathtt{F}$.
- For atomic formulas, $[\![P(t_1, \ldots, t_n)]\!]^{\mathrm{p}(v)}_{\rho} \wedge^s \neg^s [\![P(t_1, \ldots, t_n)]\!]^{\mathrm{o}(v)}_{\rho}$ equals
  $$\mathrm{Pess}^s(x) \wedge^s \neg^s \mathrm{Opt}^s(x) \tag{25}$$
  where x equals $[\![P(t_1, \ldots, t_n)]\!]^{v}_{\rho}$. But the conjunction in (25) cannot return $\mathtt{U}$ since its arguments are different from $\mathtt{U}$. This conjunction can also not evaluate to $\mathtt{T}$, for then both conjuncts evaluate to $\mathtt{T}$, forcing x to be $\mathtt{T}$ and $\mathtt{F}$ at the same time.
- For negated formulas, we compute
  $$\begin{aligned} [\![\neg\phi']\!]^{\mathrm{p}(v)}_{\rho} \wedge^s \neg^s [\![\neg\phi']\!]^{\mathrm{o}(v)}_{\rho} &= \neg^s [\![\neg\phi']\!]^{\mathrm{o}(v)}_{\rho} \wedge^s [\![\neg\phi']\!]^{\mathrm{p}(v)}_{\rho} && \wedge^s \text{ comm.} \\ &= \neg^s \neg^s [\![\phi']\!]^{\mathrm{p}(v)}_{\rho} \wedge^s \neg^s [\![\phi']\!]^{\mathrm{o}(v)}_{\rho} && \text{sem. of } \neg \\ &= [\![\phi']\!]^{\mathrm{p}(v)}_{\rho} \wedge^s \neg^s [\![\phi']\!]^{\mathrm{o}(v)}_{\rho} && \neg^s \text{ idemp.} \\ &= \mathtt{F} && \text{by ind.} \end{aligned} \tag{26}$$
- For conjunctions, the expression
  $$[\![\phi_1 \wedge \phi_2]\!]^{\mathrm{p}(v)}_{\rho} \wedge^s \neg^s [\![\phi_1 \wedge \phi_2]\!]^{\mathrm{o}(v)}_{\rho} \tag{27}$$
  cannot be $\mathtt{U}$. Assuming that it is $\mathtt{T}$, we infer that both conjuncts are $\mathtt{T}$. Thus $[\![\phi_i]\!]^{\mathrm{p}(v)}_{\rho} = \mathtt{T}$ for $i = 1, 2$ and $[\![\phi_1 \wedge \phi_2]\!]^{\mathrm{o}(v)}_{\rho} = \mathtt{F}$. Without loss of generality, $[\![\phi_2]\!]^{\mathrm{o}(v)}_{\rho} = \mathtt{F}$ follows, and so $[\![\neg\phi_2]\!]^{\mathrm{p}(v)}_{\rho} = \mathtt{T}$. Thus, we arrive at $[\![\phi_2 \wedge \neg\phi_2]\!]^{\mathrm{p}(v)}_{\rho} = \mathtt{T}$, contradicting the induction hypothesis.
- For existential formulas, the expression
  $$[\![\exists x\, \phi]\!]^{\mathrm{p}(v)}_{\rho} \wedge^s \neg^s [\![\exists x\, \phi]\!]^{\mathrm{o}(v)}_{\rho} \tag{28}$$
  cannot be $\mathtt{U}$. Assuming that it is $\mathtt{T}$, we infer that both conjuncts are $\mathtt{T}$. Thus
  $$\bigvee^{s}_{u \in \mathcal{U}_v} [\![\phi]\!]^{\mathrm{p}(v)}_{\rho[x \mapsto u]} = \mathtt{T} \tag{29}$$
  $$\bigvee^{s}_{u \in \mathcal{U}_v} [\![\phi]\!]^{\mathrm{o}(v)}_{\rho[x \mapsto u]} = \mathtt{F}. \tag{30}$$
  From the first equation we infer $[\![\phi]\!]^{\mathrm{p}(v)}_{\rho[x \mapsto u']} = \mathtt{T}$ for some $u' \in \mathcal{U}_v$. From the second equation, we conclude that $[\![\phi]\!]^{\mathrm{o}(v)}_{\rho[x \mapsto u']} = \mathtt{F}$, rendering $[\![\neg\phi]\!]^{\mathrm{p}(v)}_{\rho[x \mapsto u']} = \mathtt{T}$, and so $[\![\phi \wedge \neg\phi]\!]^{\mathrm{p}(v)}_{\rho[x \mapsto u']} = \mathtt{T}$, contradicting the induction hypothesis (which is quantified over *all* local environments).

□

Corollary 5.3 (Consistency in entailment form) *For all partial models $\mathcal{M}_v$, global formulas ϕ, and local environments ρ, we have*

$$[\![\phi]\!]^{\mathrm{p}(v)}_{\rho} = \mathtt{T} \quad \Rightarrow \quad [\![\phi]\!]^{\mathrm{o}(v)}_{\rho} = \mathtt{T}. \tag{31}$$

Proof. Proof by contradiction: If $[\![\phi]\!]_\rho^{\mathrm{p}(v)} = \mathtt{T}$ and $[\![\phi]\!]_\rho^{\mathrm{o}(v)} = \mathtt{F}$, then $\neg^s [\![\phi]\!]_\rho^{\mathrm{o}(v)}$ equals $\mathtt{T}$, i.e. $[\![\neg\phi]\!]_\rho^{\mathrm{p}(v)} = \mathtt{T}$, and so $[\![\phi \wedge \neg\phi]\!]_\rho^{\mathrm{p}(v)} = \mathtt{T}$ follows, contradicting Theorem 5.2. □

It is rather obvious that Corollary 5.3 also implies Theorem 5.2, so these assertions are equivalent.

Theorem 5.4 (Semantics of consensus) *Let $\mathcal{M}_v$ be a partial model. For all global formulas ϕ and local environments ρ we have*

$$[\![\phi]\!]_\rho^v = [\![\phi]\!]_\rho^{\mathrm{p}(v)} \cup [\![\phi]\!]_\rho^{\mathrm{o}(v)}. \tag{32}$$

The proof of (32) requires several lemmas.

Lemma 5.5 (Distributivity of negation) *For all $x, y \in \mathbb{P}_{\mathrm{s}}(\mathbb{B})$, we have*

$$\neg^s(x \cup y) = \neg^s x \cup \neg^s y. \tag{33}$$

Proof. Since $\neg^s$ is a bijection, we have $x \neq y$ iff $\neg^s x \neq \neg^s y$, so the LHS of (33) equals $\mathtt{U}$ iff its RHS equals $\mathtt{U}$. Otherwise, both sides evaluate to $\neg^s x$.□

Lemma 5.6 (Consistent distributivity of conjunction) *Let $a, b, x, y \in \mathbb{P}_{\mathrm{s}}(\mathbb{B}) \setminus \{\mathtt{U}\}$. If $a = \mathtt{T}$ implies $b = \mathtt{T}$ and if $x = \mathtt{T}$ implies $y = \mathtt{T}$, then*

$$(a \cup b) \wedge^s (x \cup y) = (a \wedge^s x) \cup (b \wedge^s y). \tag{34}$$

Proof.

(i) Assume that the LHS of (34) is $\mathtt{F}$. Then $a \cup b$ or $x \cup y$ evaluate to $\mathtt{F}$.
   (a) If $a \cup b$ evaluates to $\mathtt{F}$, then a and b equal $\mathtt{F}$, and so $a \wedge^s x$ and $b \wedge^s y$ evaluate to $\mathtt{F}$. Therefore, the RHS of (34) evaluates to $\mathtt{F}$ as well.
   (b) If $x \cup y$ evaluates to $\mathtt{F}$, we reason symmetrically.

(ii) Assume that the LHS is $\mathtt{T}$. Then $a \cup b$ and $x \cup y$ evaluate to $\mathtt{T}$. Thus, a, b, x, and y all evaluate to $\mathtt{T}$. Therefore, the RHS of (34) evaluates to $\mathtt{T}$ as well.

(iii) Assume that the LHS evaluates to $\mathtt{U}$. Then $a \cup b$ or $x \cup y$ evaluates to $\mathtt{U}$.
   (a) If the RHS computes to $\mathtt{F}$, we derive a contradiction:
   First, if $a \cup b$ evaluates to $\mathtt{U}$, then $a \neq b$ and $x \cup y \neq \mathtt{F}$; the latter because the LHS equals $\mathtt{U}$. But $a \neq b$ implies $a \neq \mathtt{T}$ by assumption of the lemma. Thus, a equals $\mathtt{F}$ and so b equals $\mathtt{T}$. Since the RHS equals $\mathtt{F}$, we infer $y = \mathtt{F}$ from $b = \mathtt{T}$. By assumption of the lemma, $y = \mathtt{F}$ implies $x = \mathtt{F}$, contradicting $x \cup y \neq \mathtt{F}$.
   Second, if $x \cup y$ evaluates to $\mathtt{U}$, we argue symmetrically.
   (b) If the RHS computes to $\mathtt{T}$, then $a \wedge^s x$ and $b \wedge^s y$ evaluate to $\mathtt{T}$. But then a, b, x, and y evaluate to $\mathtt{T}$. Thus, the LHS computes to $\mathtt{T}$, a contradiction.

□

Lemma 5.7 (Consistent distributivity of disjunction) *Given two families $(x_i)_{i \in I}$ and $(y_i)_{i \in I}$ in $\mathbb{P}_{\mathrm{s}}(\mathbb{B}) \setminus \{\mathtt{U}\}$ such that $x_i = \mathtt{T}$ implies $y_i = \mathtt{T}$ for all*

$i \in I$, we have

$$\bigvee^{s}(x_i \cup y_i) = \left(\bigvee^{s} x_i\right) \cup \left(\bigvee^{s} y_i\right). \tag{35}$$

Proof.

(i) If the LHS of (35) is $\mathtt{T}$, then $x_{i_0} \cup y_{i_0} = \mathtt{T}$ for some $i_0 \in I$, so $x_{i_0} = y_{i_0} = \mathtt{T}$. The former implies $\bigvee^s x_i = \mathtt{T}$, the latter implies $\bigvee^s y_i = \mathtt{T}$. Therefore, the RHS of (35) computes to $\mathtt{T}$ as well.

(ii) If the LHS of (35) is $\mathtt{F}$, then $x_i \cup y_i = \mathtt{F}$ for all $i \in I$, so $x_i = y_i = \mathtt{F}$ for all $i \in I$. The former implies $\bigvee^s x_i = \mathtt{F}$, the latter implies $\bigvee^s y_i = \mathtt{F}$. Therefore, the RHS of (35) computes to $\mathtt{F}$ as well.

(iii) If the LHS of (35) is $\mathtt{U}$, then there is some $i_0 \in I$ with $x_{i_0} \cup y_{i_0} = \mathtt{U}$, and for all $i \in I$, we have $x_i \cup y_i \neq \mathtt{T}$.
   (a) Assume that the RHS of (34) equals $\mathtt{F}$. Then $\bigvee^s x_i$ and $\bigvee^s y_i$ equals $\mathtt{F}$, meaning that all x_i and y_i are $\mathtt{F}$, contradicting $x_{i_0} \cup y_{i_0} = \mathtt{U}$.
   (b) Assume that the RHS of (34) equals $\mathtt{T}$. Then $\bigvee^s x_i$ and $\bigvee^s y_i$ equals $\mathtt{T}$, meaning that there exist $i_1, i_2 \in I$ with $x_{i_1} = \mathtt{T}$ and $y_{i_2} = \mathtt{T}$. But $x_{i_1} = \mathtt{T}$ implies $y_{i_1} = \mathtt{T}$, and so $x_{i_1} \cup y_{i_1} = \mathtt{T}$ implies that the LHS equals $\mathtt{T}$, a contradiction.

□

Proof of Theorem 5.4

By structural induction on ϕ:

- We have $[\![\top]\!]^v_\rho = \{\mathtt{T}\} = \{\mathtt{T}\} \cup \{\mathtt{T}\} = [\![\top]\!]^{\mathrm{p}(v)}_\rho \cup [\![\top]\!]^{\mathrm{o}(v)}_\rho$.
- For atomic formulas, we compute $[\![P(t_1, \ldots, t_n)]\!]^{\mathrm{p}(v)}_\rho \cup [\![P(t_1, \ldots, t_n)]\!]^{\mathrm{o}(v)}_\rho = \mathrm{Pess}^s(x) \cup \mathrm{Opt}^s(x)$, where x equals $[\![P(t_1, \ldots, t_n)]\!]^v_\rho$. But in $\mathbb{P}_{\mathrm{s}}(\mathbb{B})$, we have the equation
$$\mathrm{Pess}^s(x) \cup \mathrm{Opt}^s(x) = x. \tag{36}$$
- For negated formulas, we use induction, Lemma 5.5, and the commutativity of $\cup$.
- For conjunctions, we use induction, Corollary 5.3, and Lemma 5.6.
- For existential formulas, we use induction, Corollary 5.3, and Lemma 5.7.

As remarked earlier $[\![\cdot]\!]^{\mathrm{p}(v)}_\rho$ is consistent, whereas this is not true for $[\![\cdot]\!]^{\mathrm{o}(v)}_\rho$ in general. For complete models, however, it turns out that these semantics agree.

Proposition 5.8 (Semantics of complete models) *Let $\mathcal{M}_v$ be a partial model for $(\Sigma, \mathcal{U})$ with $\Sigma_v = \Sigma$. Then $[\![\phi]\!]^{\mathrm{p}(v)}_\rho = [\![\phi]\!]^{\mathrm{o}(v)}_\rho$ for all global formulas ϕ and local environments ρ.*

Proof. We already remarked that $[\![\phi]\!]^v_\rho \in \mathbb{P}_{\mathrm{s}}(\mathbb{B}) \setminus \{\mathtt{U}\}$ holds for complete models. But then Theorem 5.4 implies that $[\![\phi]\!]^{\mathrm{p}(v)}_\rho$ is equal to $[\![\phi]\!]^{\mathrm{o}(v)}_\rho$. □

6 Model-checking $[\![\cdot]\!]^v_\rho$

We now show that the computation of $[\![\cdot]\!]^v_\rho$, $[\![\cdot]\!]^{\mathrm{p}(v)}_\rho$, and $[\![\cdot]\!]^{\mathrm{o}(v)}_\rho$ can be reduced to the usual model-checking of models of first-order logic. One may then approximate such standard checks with the techniques of abstract interpretation [5].

6.1 Expressive power of $[\![\cdot]\!]^{\mathrm{p}(v)}_\rho$ and $[\![\cdot]\!]^{\mathrm{o}(v)}_\rho$

The results shown in previous sections entail that we may compute $[\![\phi]\!]^v_\rho$ via $[\![\cdot]\!]^{\mathrm{p}(v)}_\rho$ or $[\![\cdot]\!]^{\mathrm{o}(v)}_\rho$ *only*. Thus, it suffices to implement a model checker for any of these semantics.

Theorem 6.1 (A first reduction of $[\![\cdot]\!]^v_\rho$) *Let $\mathcal{M}_v$ be a partial model for $(\Sigma,\mathcal{U})$, ϕ a global formula, and ρ a local environment. Then*

(37) $[\![\phi]\!]^v_\rho = [\![\phi]\!]^{\mathrm{p}(v)}_\rho \cup \neg^s [\![\neg\phi]\!]^{\mathrm{p}(v)}_\rho$

(38) $[\![\phi]\!]^v_\rho = [\![\phi]\!]^{\mathrm{o}(v)}_\rho \cup \neg^s [\![\neg\phi]\!]^{\mathrm{o}(v)}_\rho$.

Proof. Because of the semantics of $\neg$ and the idempotency of $\neg^s$, these equations are equivalent. For the same reasons the first equation follows from (32). □

6.2 Negation normal forms for model checks

We decomposed the semantics of $[\![\phi]\!]^v_\rho$ into the consensus of a pessimistic and an optimistic semantics in order to avoid having to deploy a three-valued model checker. Our goal is to (re)use conventional first-order model checkers to the extent possible to verify arbitrary first-order formulas over partial views. However, the treatment of negation makes the pessimistic and optimistic model checkers mutually dependent, preventing the use of standard tools for first-order-logic model checking. We can avoid this limitation by transforming ϕ to its negation normal form. Similarly to [2,8], this then moves the optimistic and pessimistic interpretations into the phase of *model construction*.

In presenting first-order logic, we assumed that $\vee$, $\rightarrow$, and $\forall x$ are derived syntactic notions. This assumption was intuitively justified by the familiar semantic DeMorgan laws for first-order logic. For each signature Σ and each ϕ of our logic, we now use these laws to compute the standard negation normal form $T(\phi)$, an element generated by the grammar

$$L ::= P(t,t,\ldots,t) \mid \neg P(t,t,\ldots,t) \qquad // \textbf{ L}\text{iterals}$$

(39) $$\phi ::= \bot \mid \top \mid L \mid \phi \wedge \phi \mid \phi \vee \phi \mid \exists x\, \phi \mid \forall x\, \phi.$$

We specify this translation $\phi \to T(\phi)$, where the domain of this translation is all expressions of (1) and the range is all expressions of (39), via (contextual) rewrite rules:

$$\begin{array}{ll} \top \rightsquigarrow \top & P(t,\ldots,t) \rightsquigarrow P(t,\ldots,t) \\ \phi \wedge \phi \rightsquigarrow \phi \wedge \phi & \exists x\, \phi \rightsquigarrow \exists x\, \phi \\ \neg\top \rightsquigarrow \bot & \neg P(t,\ldots,t) \rightsquigarrow \neg P(t,\ldots,t) \\ \neg\neg\phi \rightsquigarrow \phi & \neg(\phi \wedge \phi) \rightsquigarrow \neg\phi \vee \neg\phi \\ \neg\exists x\, \phi \rightsquigarrow \forall x\, \neg\phi. & \end{array}$$

Example 6.2 For ϕ being $\neg[\exists y\, (\neg(y = 0) \wedge \neg(\exists x\, [x < y + 1]))]$, we compute $T(\phi)$ to be

$$\forall y\, [(y = 0) \vee \exists x\, (x < y + 1)]. \tag{40}$$

The denotational semantics for formulas $T(\phi)$ is given in Figure 2. The function $\vee^s$ is the binary version of $\bigvee^s$. The semantics for $\forall x$, the function $\bigwedge^s$, has the same type as $\bigvee^s$ and is defined as

$$\bigwedge^s \stackrel{\text{def}}{=} \neg^s \circ \bigvee^s \circ \prod \neg^s. \tag{41}$$

Notice that the semantics $[\![\cdot]\!]^{\mathrm{n}(v)}_\rho$ always computes over $\mathbb{P}_{\mathrm{s}}(\mathbb{B}) \setminus \{\mathtt{U}\}$ only, except for the evaluation of $[\![P(t_1,\ldots,t_n)]\!]^v_\rho$ in the clauses for $P(t_1,\ldots,t_n)$ and $\neg P(t_1,\ldots,t_n)$; such values get immediately lifted to proper truth values by means of Pess^s and $\neg^s \mathrm{Opt}^s$ (respectively). [11] Observe that if $[\![\phi]\!]^v_\rho = \mathtt{U}$ then $[\![\neg\phi]\!]^v_\rho$ should also evaluate to $\mathtt{U}$, which is guaranteed by defining it as $\neg^s [\![\phi]\!]^v_\rho = \neg^s \mathtt{U} = \mathtt{U}$. So what we want in the definition of $[\![P(t_1,\ldots,t_n)]\!]^{\mathrm{n}(v)}_\rho$and $[\![\neg P(t_1,\ldots,t_n)]\!]^{\mathrm{n}(v)}_\rho$ is that when $[\![\phi]\!]^v_\rho = \mathtt{U}$,

$$[\![P(t_1,\ldots,t_n)]\!]^{\mathrm{n}(v)}_\rho = [\![\neg P(t_1,\ldots,t_n)]\!]^{\mathrm{n}(v)}_\rho$$

This requires us to define $[\![\neg P(t_1,\ldots,t_n)]\!]^{\mathrm{n}(v)}_\rho$ in the manner given in Fig. 2.

We can show the adequacy of this semantics.

Theorem 6.3 (A second reduction of $[\![\cdot]\!]^v_\rho$) *Let $\mathcal{M}_v$ be a partial model for Σ, ϕ a global formula, and ρ a local environment. Then*

$$[\![\phi]\!]^{\mathrm{p}(v)}_\rho = [\![T(\phi)]\!]^{\mathrm{n}(v)}_\rho. \tag{42}$$

Proof. By structural induction on ϕ. The cases where ϕ is not of the form $\neg\phi'$ follow directly by induction, given the rewrite rules for such formulas. Let ϕ be $\neg\phi'$.

- If ϕ' equals $\top$, then $[\![\neg\top]\!]^{\mathrm{p}(v)}_\rho = \neg^s \mathtt{T} = \mathtt{F} = [\![\bot]\!]^{\mathrm{n}(v)}_\rho = [\![T(\neg\top)]\!]^{\mathrm{n}(v)}_\rho$.

[11] Also notice that $[\![\neg P(t_1,\ldots,t_n)]\!]^{\mathrm{n}(v)}_\rho$ is not defined as $\neg^s [\![P(t_1,\ldots,t_n)]\!]^{\mathrm{n}(v)}_\rho$, in contrast to $[\![\neg\phi]\!]^v_\rho$ which is defined as $\neg^s [\![\phi]\!]^v_\rho$.

$$
\begin{aligned}
[\![\bot]\!]^{\mathrm{n}(v)}_{\rho} &\stackrel{\text{def}}{=} \mathtt{F} \\
[\![\top]\!]^{\mathrm{n}(v)}_{\rho} &\stackrel{\text{def}}{=} \mathtt{T} \\
[\![P(t_1,\ldots,t_n)]\!]^{\mathrm{n}(v)}_{\rho} &\stackrel{\text{def}}{=} \mathrm{Pess}^s([\![P(t_1,\ldots,t_n)]\!]^{v}_{\rho}) \\
[\![\neg P(t_1,\ldots,t_n)]\!]^{\mathrm{n}(v)}_{\rho} &\stackrel{\text{def}}{=} \neg^s \mathrm{Opt}^s([\![P(t_1,\ldots,t_n)]\!]^{v}_{\rho}) \\
[\![\phi_1 \wedge \phi_2]\!]^{\mathrm{n}(v)}_{\rho} &\stackrel{\text{def}}{=} [\![\phi_1]\!]^{\mathrm{n}(v)}_{\rho} \wedge^s [\![\phi_2]\!]^{\mathrm{n}(v)}_{\rho} \\
[\![\phi_1 \vee \phi_2]\!]^{\mathrm{n}(v)}_{\rho} &\stackrel{\text{def}}{=} [\![\phi_1]\!]^{\mathrm{n}(v)}_{\rho} \vee^s [\![\phi_2]\!]^{\mathrm{n}(v)}_{\rho} \\
[\![\exists x\, \phi]\!]^{\mathrm{n}(v)}_{\rho} &\stackrel{\text{def}}{=} \bigvee_{u \in \mathcal{U}_v}^{s} [\![\phi]\!]^{\mathrm{n}(v)}_{\rho[x \mapsto u]} \\
[\![\forall x\, \phi]\!]^{\mathrm{n}(v)}_{\rho} &\stackrel{\text{def}}{=} \bigwedge_{u \in \mathcal{U}_v}^{s} [\![\phi]\!]^{\mathrm{n}(v)}_{\rho[x \mapsto u]}.
\end{aligned}
$$

Fig. 2. Denotational semantics for first-order logic formulas in negation normal form, evaluated in environment ρ and partial model $\mathcal{M}_v$.

- If ϕ' equals $P(t_1,\ldots,t_n)$, then

$$
\begin{aligned}
[\![\neg P(t_1,\ldots,P_n)]\!]^{\mathrm{p}(v)}_{\rho} &= \neg^s [\![P(t_1,\ldots,t_n)]\!]^{\mathrm{o}(v)}_{\rho} \\
&= \neg^s \mathrm{Opt}^s([\![P(t_1,\ldots,t_n)]\!]^{v}_{\rho}) \\
&= [\![T(\neg P(t_1,\ldots,t_n))]\!]^{\mathrm{n}(v)}_{\rho}.
\end{aligned}
$$

- If ϕ' equals $\phi_1 \wedge \phi_2$, then

$$
\begin{aligned}
[\![\neg(\phi_1 \wedge \phi_2)]\!]^{\mathrm{p}(v)}_{\rho} &= \neg^s [\![\phi_1 \wedge \phi_2]\!]^{\mathrm{o}(v)}_{\rho} \\
&= \neg^s([\![\phi_1]\!]^{\mathrm{o}(v)}_{\rho} \wedge^s [\![\phi_2]\!]^{\mathrm{o}(v)}_{\rho}) \\
&= \neg^s [\![\phi_1]\!]^{\mathrm{o}(v)}_{\rho} \vee^s \neg^s [\![\phi_2]\!]^{\mathrm{o}(v)}_{\rho} && \text{sem. law} \\
&= [\![\neg\phi_1]\!]^{\mathrm{p}(v)}_{\rho} \vee^s [\![\neg\phi_2]\!]^{\mathrm{p}(v)}_{\rho} \\
&= [\![T(\neg\phi_1)]\!]^{\mathrm{n}(v)}_{\rho} \vee^s [\![T(\neg\phi_2)]\!]^{\mathrm{n}(v)}_{\rho} && \text{by ind.} \\
&= [\![T(\neg(\phi_1 \wedge \phi_2))]\!]^{\mathrm{n}(v)}_{\rho} && \text{rewrite rule.} \qquad (43)
\end{aligned}
$$

- Finally, if ϕ' is of the form $\exists x\, \phi''$, then

$$
\begin{aligned}
[\![\neg\exists x\, \phi'']\!]^{\mathrm{p}(v)}_{\rho} &= \neg^s [\![\exists x\, \phi'']\!]^{\mathrm{o}(v)}_{\rho} \\
&= \neg^s \bigvee_{u \in \mathcal{U}_v}^{s} [\![\phi'']\!]^{\mathrm{o}(v)}_{\rho[x \mapsto u]} \\
&= \neg^s \bigvee_{u \in \mathcal{U}_v}^{s} \neg^s\neg^s [\![\phi'']\!]^{\mathrm{o}(v)}_{\rho[x \mapsto u]} && \neg^s \text{ idemp.} \\
&= \bigwedge_{u \in \mathcal{U}_v}^{s} \neg^s [\![\phi'']\!]^{\mathrm{o}(v)}_{\rho[x \mapsto u]} && \text{def. of } \textstyle\bigwedge^s
\end{aligned}
$$

$$
\begin{aligned}
&= \bigwedge_{u \in \mathcal{U}_v}{}^{s} \; [\![\neg\phi'']\!]^{\mathrm{p}(v)}_{\rho[x \mapsto u]} \\
&= \bigwedge_{u \in \mathcal{U}_v}{}^{s} \; [\![T(\neg\phi'')]\!]^{\mathrm{n}(v)}_{\rho[x \mapsto u]} \qquad \text{by ind.} \\
&= [\![\forall x\, T(\neg\phi'')]\!]^{\mathrm{n}(v)}_{\rho[x \mapsto u]} \\
&= [\![T(\neg\exists x\, \phi'')]\!]^{\mathrm{n}(v)}_{\rho[x \mapsto u]} \qquad \text{rewrite rule.} \qquad (44)
\end{aligned}
$$

□

6.3 Model construction

We emphasize the significance of Theorem 6.3. In order to compute $[\![\phi]\!]^{\mathrm{p}(v)}_{\rho}$ over a partial model $\mathcal{M}_v$ for Σ, we may instead perform a *conventional model check* for a standard model $\mathcal{M}^n_v$ of first-order logic. This model is obtained from the partial model $\mathcal{M}_v$ by (i) *extending its signature* Σ with complementary predicate symbols $\overline{P}$ for each $P \in \mathcal{P}_v$; and by (ii) letting the interpretations P^v_n and $\overline{P}^v_n$ be $n$ary relations over $\mathbb{P}_{\mathrm{s}}(\mathcal{U}_v)$. For $u_1, \ldots, u_n \in \mathbb{P}_{\mathrm{s}}(\mathcal{U}_v)$, we set

$$P^v_n \stackrel{\mathrm{def}}{=} \{(u_1, \ldots, u_n) \in \prod \mathbb{P}_{\mathrm{s}}(\mathcal{U}_v) \mid \mathrm{Pess}^s([\![P(u_1, \ldots, u_n)]\!]^v_\rho) = \mathtt{T}\} \qquad (45)$$

$$\overline{P}^v_n \stackrel{\mathrm{def}}{=} \{(u_1, \ldots, u_n) \in \prod \mathbb{P}_{\mathrm{s}}(\mathcal{U}_v) \mid \neg^s\mathrm{Opt}^s([\![P(u_1, \ldots, u_n)]\!]^v_\rho) = \mathtt{T}\}. \qquad (46)$$

In (45) and (46), we extended the definition of $[\![\cdot]\!]^v_\rho$ to elements of $\mathbb{P}_{\mathrm{s}}(\mathcal{U}_v)$ by setting $[\![u]\!]^v_\rho \stackrel{\mathrm{def}}{=} u$. The sets P^v_n and $\overline{P}^v_n$ are well defined since the expressions $[\![P(u_1, \ldots, u_n)]\!]^v_\rho$ evaluate to the same truth value, independent of the choice of ρ. Observe that the consistency of this model is represented by the fact that

$$P^v_n \cap \overline{P}^v_n = \emptyset \qquad (P \in \mathcal{P}). \qquad (47)$$

This is guaranteed since $\mathrm{Pess}^s(x) = \mathtt{T}$ and $\mathrm{Opt}^s(x) = \mathtt{T}$ imply $x = \mathtt{T}$ and $x = \mathtt{F}$ (respectively). However,

$$P^v_n \cup \overline{P}^v_n = \{(u_1, \ldots, u_n) \in \prod \mathbb{P}_{\mathrm{s}}(\mathcal{U}_v) \mid [\![P(u_1, \ldots, u_n)]\!]^v_\rho \neq \mathtt{U}\} \qquad (48)$$

is different from $\prod \mathbb{P}_{\mathrm{s}}(\mathcal{U}_v)$ in general. Having constructed the model $\mathcal{M}^n_v$ in this manner, the semantics in Figure 2 can now be seen as a specification of a *conventional* first-order-logic model checker for $[\![\cdot]\!]^{\mathrm{p}(v)}_\rho$. In summary, we have reduced our three-valued model-checking problem to two two-valued model-checking problems of ordinary first-order logic by adding complementary atomic predicates and by lifting their interpretations from the domain of the original partial model to its Smyth powerdomain. The latter domain can then be approximated with the techniques of abstract interpretation [5].

7 Related work

In [2], G. Bruns and P. Godefroid develop a three-valued version of model-checking models of computation tree logic (CTL). The models are similar to conventional Kripke structures, except that state propositions can take on values `T`, `F`, or `U`. They have a notion of refinement and show that their temporal logic semantics is sound and complete with respect to this refinement. With each three-valued model they associate two Kripke structures, an optimistic and a pessimistic one, allowing them to implement their three-valued model-checking problem as two two-valued model-checking problems for CTL (over Kripke structures). In [3], this work is being extended to generalized model checking (GMC), improving the precision (and increasing the complexity) of their initial semantics, and implementing GMC via (finite-state) automata on infinite words. [12]

In [11,8], the contributions of [2] are redeveloped for (Kripke) modal transition systems (MTSs). In MTSs transitions may take on any value `T`, `F`, or `U` — meaning that such transitions are guaranteed, impossible, and possible (respectively) [21,20].

In [12], partial state-machines are modeled as MTSs and a multiple-view semantics and its model checker are derived from the property semantics of MTSs in [11].

In [6], Cousot & Cousot use two abstraction functions $\alpha^{\forall}$ and $\alpha^{\exists}$ for the same concretization function of an abstract interpretation to systematically derive a branching-time semantics from a linear trace semantics. Similar to Pess^s and Opt^s, the abstraction functions $\alpha^{\forall}$ and $\alpha^{\exists}$ are dual with respect to complementation.

In program analysis, three-valued models of first-order logic have been used for the safe abstraction of "shape invariants" [27] and the verification of safety properties of Java programs [29].

Our views $\mathcal{M}_v$ can be partial in that they may not have interpretations for *certain* function or predicate symbols, but apart from that we assumed that views have the same representation of a specification. As D. Jackson points out, this may not always be appropriate [13].

The tool `xlinkit` analyzes distributed XML documents for possible inconsistencies, based on rules written in first-order logic [23]. Our semantic framework could possibly be used as a foundation for such a tool.

D. Jackson wrote the object modelling language Alloy [14] as a lightweight tool for specifying relational models of software artifacts. Primitive types are interpreted as finite sets without any algebraic structure. Models written in Alloy can be automatically analyzed with the Alloy Constraint Analyzer (formerly known as "Alcoa" [15]) in two modes. First, one can check inconsistency

[12] The precision of our semantics can be improved in the same manner as carried out in [3], and it would be of interest to see whether such a semantics of first-order logic could be implemented with suitable (finite-state) automata.

(= overspecification) of multiple specification components and their interaction through the checking of invariants. Second, one can refute properties (= underspecification) by computing a counterexample to an assert statement. Since the full language (first-order relational logic with a transitive closure operator) is undecidable, both modes run in a user-specified scope, the maximal size of primitive types. This approach is partial in that the inability of generating a counterexample within a given scope means that the status of inconsistency and property refutation is unknown (respectively).

S. Guerra [10] develops a framework for specifications of software artifacts, where such specifications have defaults and allow for exceptions stemming from the reuse or evolution of system demands. This framework is cast in the machinery of institutions [9]. Specifications are written in linear-time temporal logic [26] and a non-monotonic semantics for this logic is defined based on default institutions. A distance between interpretation morphisms induces a preferential preorder between models that is used to define that semantics. The signature is split into observable actions (where the distance enforces consistency across models) and attributes (where the distance checks for consistency across models).

M. Chechik and S. Easterbrook [7] merge multiple viewpoints of a software system expressed as partial state-machines. Different viewpoints need not recognize the same vocabulary of observables. Their models are multi-valued Kripke structure over a *quasi-boolean lattice*. Their semantics for the temporal logic CTL evaluates a state and a formula to an element of the merged lattice.

VDM [16] is a model-based specification language for software systems. Unlike specification languages such as Z, VDM allow for the specification of *partial* functions and its semantics is essentially Kleene's strong interpretation of propositional logic. Note that our framework assumes that functions are *total* and that the only source of partiality resides in the "lack of knowledge" of how to interpret certain function or predicate symbols.

In [22], J. M. Morris and A. Bunkenburg use a strong three-valued interpretation of equality which — unlike $\mathbb{P}_{\mathrm{s}}(=^{\mathbb{B}})$ from Example 3.3 — returns $\mathtt{U}$ only if *both* its arguments are $\mathtt{U}$. They combine this equational theory [13] with a typed logic of partial functions (LPF [17]) to obtain a system for equational reasoning in the presence of partial functions — the specifications of programs.

In [4] powerdomains are used to systematically derive the relational semantics of the standard staple of database operations.

8 Conclusions

We generalized the standard notions of models and semantics of first-order logic to partial models that may not have any information about certain function symbols and predicate symbols at their disposal. At the model-theoretic

[13] As they note, its semantics is *not* monotone over $\mathbb{P}_{\mathrm{s}}(\mathbb{B})$.

level, this is achieved via the Smyth powerdomain over the semantic universe of elements. At the logical level, we accomplish this via the Smyth powerdomain over the standard two-valued booleans. A refinement notion is developed and its soundness is proved for a fixed set of semantic elements. Standard models and semantics of first-order logic are subsumed by our framework as "complete" models, where all information is present. We decompose our semantics into a consensus (the union of the Smyth powerdomain) of an optimistic and a pessimistic semantics. That way, one can compute this semantics as a standard model check in first-order logic over a model lifted by the Smyth powerdomain, where the signature is extended with a complementary symbol for each predicate symbol in the original signature.

Acknowledgments

We wish to thank G. Bruns, P. Godefroid, R. Jagadeesan, and D. Schmidt. Their contributions to the semantics of partial models for temporal logics enabled us to carry out the technical work presented in this paper. We thank M. Mislove for his invitation to give a talk at the special session "Model checking" at MFPS'01 and for organizing such a stimulating meeting. S. Guerra made helpful comments on drafts of this paper.

References

[1] S. Abramsky and A. Jung. Domain theory. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 3, pages 1–168. Oxford Univ. Press, 1994.

[2] G. Bruns and P. Godefroid. Model Checking Partial State Spaces with 3-Valued Temporal Logics. In *Proceedings of the 11th Conference on Computer Aided Verification*, volume 1633 of *Lecture Notes in Computer Science*, pages 274–287. Springer Verlag, July 1999.

[3] G. Bruns and P. Godefroid. Generalized Model Checking: Reasoning about Partial State Spaces. In *Proceedings of CONCUR'2000 (11th International Conference on Concurrency Theory)*, volume 1877 of *Lecture Notes in Computer Science*, pages 168–182. Springer Verlag, August 2000.

[4] P. Buneman, A. Jung, and A. Ohori. Using powerdomains to generalize relational databases. *Theoretical Computer Science*, 91(1):23–55, 1991.

[5] P. Cousot and R. Cousot. Abstract interpretation: a unified lattice model for static analysis of programs. In *Proc. 4th ACM Symp. on Principles of Programming Languages*, pages 238–252. ACM Press, 1977.

[6] P. Cousot and R. Cousot. Temporal abstract interpretation. In *Conference Record of the 27th Annual ACM SIGPLAN-SIGACT Symposium on Principles*

of Programming Languages, pages 12–25, Boston, Mass., January 2000. ACM Press, New York, NY.

[7] S. M. Easterbrook and M. Chechik. A Framework for Multi-Valued Reasoning over Inconsistent Viewpoints. In *Proceedings, 23rd International Conference on Software Engineering (ICSE-01)*, Toronto, Canada, May 12-19 2001. IEEE Computer Society Press. To appear.

[8] P. Godefroid, M. Huth, and R. Jagadeesan. Abstraction-based Model Checking using Modal Transition Systems. In *Proceedings of the International Conference on Theory and Practice of Concurrency*, Lecture Notes in Computer Science. Springer Verlag, August 2001. To appear.

[9] J. A. Goguen and R. M. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the ACM*, 39(1):95–146, January 1992.

[10] S. Guerra. Distance Functions for Defaults in Reactive Systems. In T. Rus, editor, *Proc. of the 8th Int. Conf. on Algebraic Methodology and Software Technology (AMAST 2000)*, volume 1816 of *Lecture Notes in Computer Science*, pages 26–40, Iowa City, Iowa, May 2000. Springer Verlag.

[11] M. Huth, R. Jagadeesan, and D. Schmidt. Modal transition systems: a foundation for three-valued program analysis. In D. Sands, editor, *Proceedings of the European Symposium on Programming (ESOP'2001)*, volume 2028 of *LNCS*, pages 155–169, Genova, Italy, April 2001. Springer Verlag.

[12] M. Huth and S. Pradhan. Aspect-driven Property Verification of Inconsistent System Descriptions. Technical Report KSU-CIS-TR-2001-1, Computing & Information Sciences, Kansas State University, Manhattan, Kansas, May 2001. Extended 12-page abstract sunmitted to FSTTCS 2001.

[13] D. Jackson. Structuring Z Specifications With Views. *ACM Trans. on Software Engineering and Methodology*, 4(4):365–389, October 1995.

[14] D. Jackson. Alloy: A Lightweight Object Modelling Language. Technical Report TR-797, Laboratory of Computer Science, Massachusetts Institute of Technology, 28 July 2000.

[15] D. Jackson, I. Schechter, and I. Shlyahter. Alcoa: The Alloy Constraint Analyzer. In *Proc. Int'l Conf. on Software Engineering 2000 (ICSE 2000)*, pages 730–733. IEEE Computer Society Press, 2000.

[16] C. B. Jones. *Systematic Software Development Using VDM*. Prentice-Hall, Upper Saddle River, NJ 07458, USA, 1990.

[17] C. B. Jones and C. A. Middelburg. A typed logic of partial functions reconstructed classically. *Acta Informatica*, 31:399–430, 1994.

[18] P. Kelb. Model checking and abstraction: a framework preserving both truth and failure information. Technical Report OFFIS, University of Oldenburg, Germany, 1994.

[19] S. C. Kleene. *Introduction to Metamathematics*. Van Nostrand, 1952.

[20] K. G. Larsen. Modal Specifications. In J. Sifakis, editor, *Automatic Verification Methods for Finite State Systems*, number 407 in Lecture Notes in Computer Science, pages 232–246. Springer Verlag, June 12–14 1989. International Workshop, Grenoble, France.

[21] K. G. Larsen and B. Thomsen. A Modal Process Logic. In *Third Annual Symposium on Logic in Computer Science*, pages 203–210. IEEE Computer Society Press, 1988.

[22] J. M. Morris and A. Bunkenburg. E3: A Logic for Reasoning Equationally in the Presence of Partiality. *Science of Computer Programming*, 34(2):144–158, 1999.

[23] C. Nentwich, W. Emmerich, and A. Finkelstein. xlinkit: links that make sense. Technical report, Department of Computer Science, University College of London, 2001.

[24] B. Nuseibeh and S. M. Easterbrook. The Process of Inconsistency Management: A framework for understanding. In *Proceedings of the First International Workshop on the Requirements Engineering Process (REP'99)*, 2-3 September 1999.

[25] B. Nuseibeh, J. Kramer, and A. Finkelstein. A Framework for Expressing the Relationships Between Multiple Views in Requirements Specification. *IEEE Transactions on Software Engineering*, 20(10):760–773, October 1994.

[26] A. Pnueli. The temporal logic of programs. In *Proc. of the 18th IEEE Symposium on the Foundations of Computer Science*, pages 46–57, 1977.

[27] M. Sagiv, T. Reps, and R. Wilhelm. Parametric Shape Analysis via 3-Valued Logic. In *Proceedings of the 26th ACM SIGPLAN-SIGACT Symposium on Principles of programming languages*, pages 105–118, January 20-22, San Antonio, Texas 1999.

[28] M. Smyth. Powerdomains. *Journal of Computer and Systems Science*, 16:23–36, 1977.

[29] Eran Yahav. Verifying safety properties of concurrent Java programs using 3-valued logic. *ACM SIGPLAN Notices*, 36(3):27–40, March 2001.