

# A domain equation for refinement of partial systems

Michael R. A. Huth,<sup>1</sup> Radha Jagadeesan,<sup>2</sup> and David A. Schmidt<sup>3</sup>

<sup>1</sup> *Department of Computing,*

*Imperial College London, London, England*

<sup>2</sup> *School of Computer Science, Telecommunications, and Information Sciences,*

*DePaul University, Chicago, Illinois*

<sup>3</sup> *Department of Computing and Information Sciences,*

*Kansas State University, Manhattan, Kansas*

*Received 23 January 2003*

A reactive system can be specified by a labeled transition system, which indicates static structure, along with temporal-logic formulas, which assert dynamic behavior. But refining the former while preserving the latter can be difficult, because (i) labeled transition systems are “total” — characterized up to bisimulation — meaning that no new transition structure can appear in a refinement; (ii) alternatively, a refinement criterion not based on bisimulation might generate a refined transition system that violates the temporal properties.

In response, Larsen and Thomson proposed *modal transition systems*, which are “partial,” and defined a refinement criterion that preserved formulas in Hennessy-Milner logic. We show that modal transition systems are, up to a saturation condition, exactly the mixed transition systems of Dams that have consistent model checks, and we extend such systems to non-flat state sets. We then solve a domain equation over the mixed powerdomain whose solution is a bifinite domain that is universal for all saturated modal transition systems and is itself fully abstract when considered as a modal transition system. We demonstrate that many frameworks of partial systems can be translated into the domain: partial Kripke structures, partial bisimulation structures, Kripke modal transition systems, and pointer-shape-analysis graphs.

## Contents

1	Introduction	2
2	Modal transition systems	4
2.1	Background	4
2.2	Refinement	5
2.3	Property logic	7
2.4	DeMorgan duals	9
2.5	Totality	10
2.6	Maximal consistency	10
2.7	An extension: non-flat data	13

3	A domain equation for modal transition systems	14
3.1	The universal domain as a fully abstract mixed transition system	15
3.2	Embedding modal transition systems into the universal domain	19
3.3	Complementary processes	24
4	Expressiveness of modal transition systems	24
4.1	Kripke modal transition systems	25
4.2	Shape analysis with Kripke modal transition systems	27
4.3	Extended transition systems	29
4.4	Partial Kripke structures	32
5	Related work	34
	References	35

## 1. Introduction

A specification of a computing system typically consists of a segment that specifies static structure and a segment that describes dynamic behavior. For example, a sequential program can be specified by a class diagram, which displays the structure of components to be written, and by sequence diagrams, which display behaviors that must be fulfilled by the executing program; the latter assert behaviors that must be satisfied by any implementation of the former. Refinements of the specification should lead to an implementation that has the structure in the class diagram and preserves the behaviors stated by the sequence diagrams.

Reactive systems should also be specified and implemented with the assistance of structural and behavioral specifications, and indeed, it is common to employ labeled transition systems to specify the communication structure of a reactive system and to use temporal logic to assert the system’s desired behaviors. Then, a model check can verify the consistency of structure with behavior, a consistency that must be maintained in the refinements that lead to the implementation.

But what does it mean to refine a labeled transition system while preserving its desired behaviors? Labeled transition systems are “total” entities — a transition from one state to another can happen or it cannot. This may seem innocuous, but the consequences are far-reaching, because labeled transition systems can be distinguished only up to bisimulation (Par89; Mil89), so it is impractical to use bisimulation to guide the refinement of a labeled-transition system into an implementation (Lar89).

Alternatively, one might define refinement as a simulation (one-half of bisimulation) (Mil81) and “refine” one labeled transition system into a second, such that all transitions in the second system are simulated by transitions in the first. But such a simulation does *not* preserve all the temporal-logic behaviors one might specify — behaviors that are existentially quantified can hold true in the specification transition system but fail in the refinement transition system. (And if we dualize the definition of simulation, then the universally quantified properties can be lost.)

Similar problems arise when attempting to synthesize, from an implementation of a reactive system, its abstraction (*abstract interpretation* (CC77)), which might be statically analyzed for its temporal-logic properties.

The difficulty with employing a labeled transition system as a specification has its root in the manner that negative capabilities are portrayed. A labeled transition system identifies a set of states  $\Sigma$ , a set of actions  $\text{Act}$ , and a state transition relation,  $R \subseteq \Sigma \times \text{Act} \times \Sigma$ , such that  $(s, \alpha, s') \in R$  states the system is capable of performing action  $\alpha$  in state  $s$ , producing  $s'$  as its successor. By force,  $(t, \beta, t') \in (\Sigma \times \text{Act} \times \Sigma) \setminus R$  implies that, at state  $t$ , action  $\beta$  either cannot be taken or cannot result in  $t'$ . But the human who specifies  $R$  might wish to express that some instances of  $(\Sigma \times \text{Act} \times \Sigma) \setminus R$  are *still possible* (but not required) in a correct implementation. Labeled transition systems do not provide this flexibility.

Larsen and Thomsen understood well the limitations of labeled transition systems and temporal logic as a specification methodology and proposed *modal transition systems* (LT88; Lar89) as a solution. Simply stated, a modal transition system is a “partial” variant of a labeled transition system that can express the *possibility* as well as the *necessity* of a state transition. Larsen and Thomsen revised the definition of bisimulation to accommodate refinement of modal transition systems into implementations and showed that temporal properties written in Hennessy-Milner logic (HM85) are preserved by refinement (Lar89).

Larsen and Thomsen’s work applied to transition systems whose state set,  $\Sigma$ , was an unordered set. In this paper, we extend their results to state sets that are domains (AJ94), which are crucial to higher-order programming and abstract interpretation: We show that both modal transition systems and Dams’s consistent mixed transition systems (Dam96; DGG97) are instances of “saturated” transition systems. Inspired by Abramsky’s result, which characterized labeled transition systems up to bisimulation as elements within a recursively defined convex powerdomain (Abr91), we characterize domain-based, saturated transition systems up to bi-refinement as elements within a reflexively defined product of *mixed powerdomains* (Hec90; Gun92). This yields a sound treatment of refinement for a temporal logic with universal and existential quantification and negation. As a corollary, the reflexive product of mixed powerdomains generalizes Kleene’s strong semantics for propositional logic (Kle52) to non-flat settings.

**Outline of paper.** Section 2 presents modal transition systems for “loosely” specifying reactive systems; such specifications may have many non-bisimilar implementations. Refinement and a property semantics (temporal logic) are defined; the latter is shown sound with respect to the former. Consistent mixed transition systems are related to modal transition systems by means of saturation, and we extend both to non-flat data domains by means of the mixed powerdomain. In Section 3, we solve a mixed powerdomain equation to obtain a saturated transition system that is universal (all saturated transition systems can be embedded into it) and fully abstract (its greatest abstraction relation coincides with the domain order). As a by-product, refinement of mixed transition systems is logically characterized by Hennessy-Milner logic. Section 4 testifies to the expressiveness of our framework and universal domain, by showing that various frameworks for modeling and analyzing partial systems — used in concurrency theory, partial state-space model checking, and shape analysis — have linear translations into the domain. Finally, Section 5 states related work.

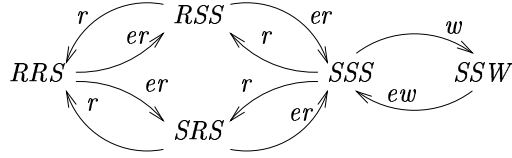


Fig. 1. A labeled transition system for two readers and one writer.

## 2. Modal transition systems

### 2.1. Background

Labeled transition systems play a prominent role in the specification, explication, and analysis of programs, as seen in structural operational semantics (Plo81), process algebras (Hoa85; Mil89), and model-checking (Hol97).

**Definition 1.** A *labeled transition system with signature*  $\text{Act}$  is a pair  $\mathcal{L} = (\Sigma, R)$ , where  $\Sigma$  is a set of *states* and  $R \subseteq \Sigma \times \text{Act} \times \Sigma$  is a *transition relation*. A labeled transition system is *pointed* if some  $s_0 \in \Sigma$  is distinguished as the starting state.

Figure 1 presents the graphical representation of a labeled transition system, which specifies the structure of a system of two readers and one writer that share a file resource. Read a state such as  $RSW$  as “first reader reads, second reader sleeps, writer writes”; the actions are  $r$  (“start read”),  $er$  (“end read”),  $w$  (“start write”), and  $ew$  (“end write”). We designate state  $SSS$  as the system’s starting state.

If a transition system  $\mathcal{L}$  is pointed, we can use its starting state and transition relation to generate a derivation tree of the process defined by  $\mathcal{L}$  (Mil89).

Throughout this paper, we assume that  $\{s' \in \Sigma \mid (s, \alpha, s') \in R\}$  is finite for all  $s \in \Sigma$  and  $\alpha \in \text{Act}$ . The intuitive meaning of  $R_{s,\alpha} = \{s' \in \Sigma \mid (s, \alpha, s') \in R\} \neq \{\}$  is that

“in state  $s$ , model  $\mathcal{L}$  has the reactive capability to engage in action  $\alpha$  which, if chosen and executed, results in a successor state  $s' \in R_{s,\alpha}$ .”

Despite the non-determinism present in labeled transition systems, the reactive capabilities in  $R_{s,\alpha}$  are *firm guarantees*: although  $\mathcal{L}$  cannot promise that action  $\alpha$  will be chosen and executed — the resolution of such choices is accomplished by mechanisms external to the model, e.g. a deterministic scheduler or a communication handshake — it does promise that an  $\alpha$ -action *is executable* from state  $s$  and that the resulting state *can be chosen* from  $R_{s,\alpha}$ . Thus, labeled transition systems are *total* specifications in the information-theoretic sense: reactive capabilities are either present or absent and such capabilities cannot, up to bisimulation equivalence (Par89; Mil89), be modified by a correct implementation.

Larsen and Thomsen (LT88) noted that labeled transition systems have limited utility as specifications of computational systems, because a correct implementation of a labeled-transition-system specification must have bisimilar behavior. This rules out use of under-determined specifications and limits the flexibility needed for stepwise implementation. Dually, the analysis of legacy software typically faces the state-explosion problem and

usually must resort to aggressive abstraction techniques. However, state-space reduction is severely constrained if conducted within a fixed bisimulation-equivalence class.

Consequently, Larsen and Thomson proposed *modal transition systems* (LT88; Lar89) as specification models that overcome these shortcomings. Their solution has a pleasant and free side effect in that it allows an extension of existing abstract-interpretation techniques (CC77) to temporal logics that combine universal and existential path quantifiers (Lar89).

## 2.2. Refinement

Dams developed, independently, mixed transition systems (Dam96) which can be seen as a more general notion of modal transition systems, so we define the two systems together:

### Definition 2 (Mixed and modal transition systems).

- 1 A *mixed transition system* (Dam96) with signature  $\text{Act}$  is a triple  $\mathcal{M} = (\Sigma, R^a, R^c)$  such that  $(\Sigma, R^m)$  is a labeled transition system with signature  $\text{Act}$ , for every *mode*  $m \in \{a, c\}$ .
- 2 A *modal transition system* (LT88) with signature  $\text{Act}$  is a mixed transition system  $\mathcal{M} = (\Sigma, R^a, R^c)$  with signature  $\text{Act}$  such that  $R^a \subseteq R^c$ .
- 3 A mixed transition system is *pointed* if there is some  $s_0 \in \Sigma$  distinguished as the starting state.
- 4 A modal transition system is *concrete* or *total* when  $R^a = R^c$ .

A mixed (modal) transition system is intended to be a “loose” specification or an abstraction of concrete-system behavior, and each of its labeled transition systems expresses a distinct “aspect” or “modality” of reactive capability:

- $R^a$  lists firm guarantees of non-deterministic reactive capabilities — as familiar from labeled transition systems;
- $R^c \setminus R^a$  lists capabilities that are possible but not guaranteed; the implementation of these reactive capabilities is optional; and
- in the case of a modal transition system, elements  $(s, \alpha, s') \in (\Sigma \times \text{Act} \times \Sigma) \setminus R^c$  represent firm guarantees that, in state  $s$ , action  $\alpha$  — if possible at all — cannot result in the successor state  $s'$ .

In Larsen’s notation (LT88), elements of  $R^a$  are denoted by  $s \xrightarrow{\alpha}^{\square} s'$  and elements of  $R^c$  by  $s \xrightarrow{\alpha}^{\diamond} s'$ , where  $\square$  denotes “for all implementations” and  $\diamond$  “for some implementation.” A modal transition system follows the philosophy that every firmly guaranteed transition is also possible to implement.

**Example 1.** Figure 2 (left) shows a mixed transition system that abstracts just the read/write-acquisition structure of a one-or-more-readers/one-writer system. For brevity,  $R^a$ -transitions are drawn as solid arcs, and those from  $R^c$  are drawn as dashed arcs. State *Reads* represents the situation when one or more readers are engaged in reading; *Write* denotes that the writer is active; and *Sleep* asserts that no process uses the shared file.

The  $R^a$ -transitions from the *Sleep* state assert that read- and write-acquisition transitions are guaranteed in any correct implementation of the mixed-transition-system spec-

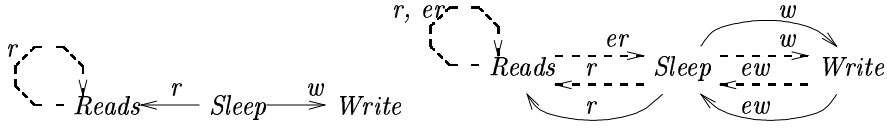


Fig. 2. A mixed transition system that is not a modal transition system (left) and a modal transition system (right).

ification. The self-transition at *Reads* is possible but not guaranteed because, if all the readers are already engaged in reading, then yet another read acquisition is impossible. Since the transitions from the *Sleep* state are guaranteed and are not shadowed by any  $R^c$ -transitions, the transition system is not modal.

In contrast, Figure 2 (right) shows a modal transition system that shows the structure of both acquisition and release transitions for a system of readers and writer. Here, every firmly guaranteed transition is shadowed by one that is possible. The self-arcs on *Reads* admit the possibility of multiple readers. Note that the transition from *Reads* to *Sleep* is in  $R^c \setminus R^a$ , because *Reads* represents the state where one or many readers are reading the shared file—it is not guaranteed that the release of merely one reader will make the system return to *Sleep*.

Larsen’s interpretation of  $\square$  and  $\diamond$  in mixed transition systems suggests that the refinement of one mixed transition system into another must refine the two forms of transitions in dual fashion:

**Definition 3 (Refinement).** Let  $\mathcal{M} = (\Sigma, R^a, R^c)$  be a mixed transition system with signature  $\text{Act}$ . A relation  $Q \subseteq \Sigma \times \Sigma$  is a *refinement within  $\mathcal{M}$*  (LT88; Dam96) iff  $(s, t) \in Q$  implies for all  $\alpha \in \text{Act}$

- 1 if  $(t, \alpha, t') \in R^a$ , there exists some  $s' \in \Sigma$  such that  $(s, \alpha, s') \in R^a$  and  $(s', t') \in Q$ ;
- 2 if  $(s, \alpha, s') \in R^c$ , there exists some  $t' \in \Sigma$  such that  $(t, \alpha, t') \in R^c$  and  $(s', t') \in Q$ .

We write  $s \prec_{\mathcal{M}} t$  or  $s \prec t$  if there is some refinement  $Q$  with  $(s, t) \in Q$ . In that case,  $s$  *refines (is abstracted by)*  $t$ .

The union  $\prec_{\mathcal{M}}$  of all refinements within a mixed transition system  $\mathcal{M}$  is the greatest such refinement and a preorder.

The above definition is applied to the case of showing that one mixed transition system  $\mathcal{M} = (\Sigma_1, R_1^a, R_1^c)$  refines another system  $\mathcal{N} = (\Sigma_2, R_2^a, R_2^c)$  by constructing the disjoint union  $\mathcal{M} + \mathcal{N} = (\Sigma_1 + \Sigma_2, R_1^a + R_2^a, R_1^c + R_2^c)$ . If  $\mathcal{M}$  and  $\mathcal{N}$  are pointed with start states  $i$  and  $j$ , respectively, we say that  $(\mathcal{M}, j)$  *refines (is abstracted by)*  $(\mathcal{N}, i)$  and write  $(\mathcal{M}, j) \prec (\mathcal{N}, i)$  iff  $(j, i) \in Q$  for some refinement  $Q$  within  $\mathcal{M} + \mathcal{N}$ .

The intuition behind  $\mathcal{M}$  refining  $\mathcal{N}$  is that all guaranteed reactive capabilities,  $R^a$ -transitions in  $\mathcal{N}$ , are preserved (up to simulation) within the more-concrete system  $\mathcal{M}$ ; and  $\mathcal{M}$  contains only those possible reactive capabilities,  $R^c$ -transitions in  $\mathcal{M}$ , (up to simulation) that were originally specified within  $\mathcal{N}$ .

**Example 2 (Refinement of mixed transition systems).** If we read the labeled transition system in Figure 1 as a total modal transition system (that is, each arc in

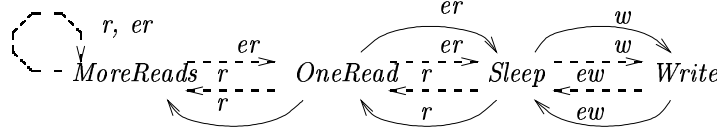


Fig. 3. A refinement of the modal transition system in Figure 2 (right).

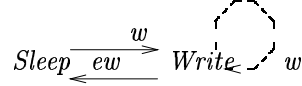


Fig. 4. A system that is not a refinement of any system in Figure 2.

the Figure denotes an  $R^c$ - as well as an  $R^a$ -transition), then the system is a refinement of the modal transition system in Figure 2 (right) — given explicitly by  $Q = \{(SSS, Sleep), (SSW, Write), (RSS, Reads), (SRS, Reads), (RRS, Reads)\}$  — but *not* of the mixed transition system, Figure 2 (left).

Figure 3 shows a modal transition system that is not total yet is still a refinement of Figure 2 (right): The refinement depicts a specification of a system of at least two readers and a writer.

Finally, Figure 4 shows a system that does not refine either of the systems in Figure 2, because not all  $R^a$ -transitions are preserved and a new  $R^c$ -transition is added.

### 2.3. Property logic

We equip mixed and modal transition systems with a property logic  $\mathbb{L}$ , the modal mu-calculus (Koz83), parametric in signature  $\text{Act}$ :

$$\phi ::= \text{tt} \mid Z \mid \neg\phi \mid \phi \wedge \phi \mid (\exists\alpha)\phi \mid \mu Z.\phi, \quad (1)$$

where  $\alpha \in \text{Act}$ ,  $Z \in \text{var}$  for a countable set of recursion variables  $\text{var}$ , and all free occurrences of  $Z$  in  $\phi$  for  $\mu Z.\phi$  are under an even scope of negations. We assume the standard embedding of  $\text{Act-CTL}$  (see e.g. (Bra91)) into  $\mathbb{L}$ , e.g.  $\text{EF}_\alpha \neg(\exists\beta)\text{tt}$  (“there is an  $\alpha$ -path on which, eventually, there is no  $\beta$ -successor state”) translates into  $\mu Z.(\neg(\exists\beta)\text{tt}) \vee (\exists\alpha)Z$ , and make liberal use of  $\text{Act-CTL}$  connectives as abbreviations of their corresponding syntactic equivalents in  $\mathbb{L}$ .

The logic’s denotational semantics  $\llbracket \cdot \rrbracket^m$  maps formulas  $\phi$  and environments  $\rho$  into sets of states for a mode of analysis  $m \in \{a, c\}$ ; its definition, in Figure 5, utilizes a variable environment  $\rho = (\rho^a, \rho^c)$  such that  $\rho^m: \text{var} \rightarrow \mathcal{P}(\Sigma)$  for  $m \in \{a, c\}$ . Note that  $\neg a \stackrel{\text{def}}{=} c$ ,  $\neg c \stackrel{\text{def}}{=} a$ , and  $\text{pre}_\alpha^m(A) \stackrel{\text{def}}{=} \{s \in \Sigma \mid \exists s' \in \Sigma, (s, \alpha, s') \in R^m, s' \in A\}$ .

We write  $s \models_\rho^a \phi$  iff  $s \in \llbracket \phi \rrbracket_\rho^a$  and say that  $\phi$  is a  $(\rho)$ -valid assertion at  $s$ ; similarly  $s \models_\rho^c \phi$  iff  $s \in \llbracket \phi \rrbracket_\rho^c$  and say that  $\phi$  is  $(\rho)$ -consistent at  $s$ . (If  $\phi$  is closed, we elide  $\rho$ .)

The semantics in Figure 5 is the standard one for labeled transition systems with signature  $\text{Act}$ , *except* for the treatment of negation: to evaluate  $\neg\phi$  in mode  $m$ , first evaluate  $\phi$  in mode  $\neg m$  and then negate that result (Kel94). Least fixed points  $\text{lfp } F^m$  are

$$\begin{aligned}
\llbracket \text{tt} \rrbracket_\rho^m &\stackrel{\text{def}}{=} \Sigma \\
\llbracket Z \rrbracket_\rho^m &\stackrel{\text{def}}{=} \rho^m(Z) \\
\llbracket \neg\phi \rrbracket_\rho^m &\stackrel{\text{def}}{=} \Sigma \setminus \llbracket \phi \rrbracket_\rho^m \\
\llbracket \phi_1 \wedge \phi_2 \rrbracket_\rho^m &\stackrel{\text{def}}{=} \llbracket \phi_1 \rrbracket_\rho^m \cap \llbracket \phi_2 \rrbracket_\rho^m \\
\llbracket (\exists\alpha)\phi \rrbracket_\rho^m &\stackrel{\text{def}}{=} \text{pre}_\alpha^m(\llbracket \phi \rrbracket_\rho^m) \\
\llbracket \mu Z.\phi \rrbracket_\rho^m &\stackrel{\text{def}}{=} \text{lfp } F^m; \text{ where } F^m(A) \stackrel{\text{def}}{=} \llbracket \phi \rrbracket_{\rho[Z \mapsto A]}^m.
\end{aligned}$$

Fig. 5. Property semantics over mixed transition systems (HJS01) for mode  $m \in \{a, c\}$ .

computed in the complete lattice  $(\mathcal{P}(\Sigma), \subseteq)$ . For the standard syntactic approximations of fixed-point formulas  $\mu Z.\phi$

$$\mu_0 Z.\phi \stackrel{\text{def}}{=} \neg\text{tt} \qquad \mu_{l+1} Z.\phi \stackrel{\text{def}}{=} \phi[Z \mapsto \mu_l Z.\phi] \quad (l \geq 0) \quad (2)$$

we have  $s \models_\rho^m \mu Z.\phi$  in a mixed transition system iff  $s \models_\rho^m \mu_l Z.\phi$  for some  $l \geq 0$ ; *provided* that  $\Sigma$  is finite or  $\mu Z.\phi$  is unnested (Lar90) — no fixed-point subformulas depend on an outer fixed point. (As customary,  $\phi[Z \mapsto \psi]$  denotes the formula obtained by replacing all free occurrences of  $Z$  in  $\phi$  with  $\psi$ .)

**Example 3 (Valid and consistent assertions).** Consider the modal transition system in Figure 2 (right). The assertion  $\neg(\exists w)(\exists r)\text{tt}$  is valid at all states, because we fail to prove that  $(\exists w)(\exists r)\text{tt}$  is consistent — there is no sequence of two  $R^c$ -transitions labeled by  $w$  and then  $r$ .

The property  $(\exists r)(\exists r)(\exists er)(\exists w)\text{tt}$  is consistent at *Sleep*, because we can find (at *Sleep*) a sequence of four  $R^c$ -transitions labeled by these actions in that order. (The assertion is valid, however, at no state.)

Finally, we see that  $\neg\mu Z.\neg(\exists w)(\exists ew)\neg Z$  is valid at *Sleep*, because the system allows arbitrarily many cycles of write acquisitions and releases along guaranteed arcs before a read is performed.

Environments  $\rho$  are *sound for  $\mathcal{M}$*  iff for all  $s \prec_{\mathcal{M}} t$  and  $Z \in \text{var}$ ,  $t \in \rho^a(Z)$  implies  $s \in \rho^a(Z)$  and  $s \in \rho^c(Z)$  implies  $t \in \rho^c(Z)$ .

**Theorem 1 (Soundness of semantics with respect to refinement (HJS01)).** For mixed transition system,  $\mathcal{M}$ , with signature,  $\text{Act}$ , let  $s, t \in \Sigma_{\mathcal{M}}$  and  $s \prec_{\mathcal{M}} t$ . For every  $\phi \in \mathbb{L}$  with signature  $\text{Act}$  and every sound environment  $\rho$

- 1 if  $t \models_\rho^a \phi$ , then  $s \models_\rho^a \phi$ ;
- 2 if  $s \models_\rho^c \phi$ , then  $t \models_\rho^c \phi$ .

*Proof of Theorem 1* The two items are proved by a nested induction: the outermost induction is on the fixed-point depth and the innermost induction on the structure of a formula  $\phi$ .

- There is nothing to show for the clause  $\text{tt}$ ; for clause  $Z$ , we use the soundness of  $\rho$ ;
- for  $\neg\phi$ ,



- let  $t \models_{\rho}^a \neg\phi$ . Then  $t \not\models_{\rho}^c \phi$ . By induction on item 2, we infer  $s \not\models_{\rho}^c \phi$ , i.e.  $s \models_{\rho}^a \neg\phi$ ;
- let  $s \models_{\rho}^c \neg\phi$ . Then  $s \not\models_{\rho}^a \phi$ . By induction on item 1, we infer  $t \not\models_{\rho}^a \phi$ , i.e.  $t \models_{\rho}^c \neg\phi$ ;
- for  $\phi_1 \wedge \phi_2$ ,
  - let  $t \models_{\rho}^a \phi_1 \wedge \phi_2$ . Then  $t \models_{\rho}^a \phi_i$  for  $i = 1, 2$ . By induction on item 1, we infer  $s \models_{\rho}^a \phi_i$  for  $i = 1, 2$ ; i.e.  $s \models_{\rho}^a \phi_1 \wedge \phi_2$ ;
  - let  $s \models_{\rho}^c \phi_1 \wedge \phi_2$ . Then  $s \models_{\rho}^c \phi_i$  for  $i = 1, 2$ . By induction on item 2, we infer  $t \models_{\rho}^c \phi_i$  for  $i = 1, 2$ ; i.e.  $t \models_{\rho}^c \phi_1 \wedge \phi_2$ ;
- for  $(\exists\alpha)\phi$ ,
  - let  $t \models_{\rho}^a (\exists\alpha)\phi$ . Then there exists some  $(t, \alpha, t') \in R^a$  such that  $t' \models_{\rho}^a \phi$ . Since  $s \prec_{\mathcal{M}} t$ , there exists some  $s' \in \Sigma$  such that  $(s, \alpha, s') \in R^a$  and  $s' \prec_{\mathcal{M}} t'$ . By induction on item 1,  $t' \models_{\rho}^a \phi$  implies  $s' \models_{\rho}^a \phi$ . But then  $(s, \alpha, s') \in R^a$  secures  $s \models_{\rho}^a (\exists\alpha)\phi$ ;
  - let  $s \models_{\rho}^c (\exists\alpha)\phi$ . Then there exists some  $(s, \alpha, s') \in R^c$  such that  $s' \models_{\rho}^c \phi$ . Since  $s \prec_{\mathcal{M}} t$ , there exists some  $t' \in \Sigma$  such that  $(t, \alpha, t') \in R^c$  and  $s' \prec_{\mathcal{M}} t'$ . By induction on item 2,  $s' \models_{\rho}^c \phi$  implies  $t' \models_{\rho}^c \phi$ . But then  $(t, \alpha, t') \in R^c$  secures  $t \models_{\rho}^c (\exists\alpha)\phi$ ;
- for  $\mu Z.\phi$ , let  $\mathbb{L}[\Sigma, \prec]$  be the collection of lower subsets  $L$  of  $\Sigma$  with respect to  $\prec$ :  $t \in L$  and  $s \prec t$  imply  $s \in L$ . Dually,  $\mathbb{U}[\Sigma, \prec]$  is the collection of upper subsets  $U$  of  $\Sigma$  with respect to  $\prec$ :  $s \in U$  and  $s \prec t$  imply  $t \in U$ . We set  $F_0^m \stackrel{\text{def}}{=} \{\}$ ,  $F_{\gamma+1}^m \stackrel{\text{def}}{=} F^m(F_{\gamma}^m)$ , and  $F_{\lambda}^m \stackrel{\text{def}}{=} \bigcup_{\gamma < \lambda} F_{\gamma}^m$  for limit ordinals  $\lambda$ .
  - By induction on  $\phi$  and the fact that lower sets are closed under arbitrary unions,  $F_{\gamma}^a \in \mathbb{L}[\Sigma, \prec]$  for all ordinals  $\gamma$ . Since  $\text{lfp}F^a$  is of that form, we have shown item 1;
  - Similarly, we infer  $F_{\gamma}^c \in \mathbb{U}[\Sigma, \prec]$  for all ordinals  $\gamma$ . Since  $\text{lfp}F^c$  is of that form, this shows item 2.

□

Theorem 1 is central to the utility of model-checking partial systems: item 1 says that all assertions that are valid at state  $t$  remain valid at all states that refine  $t$ . Dually, item 2 states that properties consistent at state  $s$  remain consistent for states that abstract  $s$ . Note that item 2 is required even if there is no need for explicit consistency checks — validating  $\neg\phi$  at state  $t$  amounts to checking whether  $\phi$  is consistent at  $t$ .

#### 2.4. DeMorgan duals

The propositional operators falsity (**ff**), disjunction ( $\phi \vee \phi$ ), implication ( $\phi \rightarrow \phi$ ), universal branching ( $(\forall\alpha)\phi$ ), and greatest fixed points ( $\nu Z.\phi$ ) are expressed in  $\mathbb{L}$  in the expected way:

$$\begin{array}{ll}
\mathbf{ff} \stackrel{\text{def}}{=} \neg\mathbf{tt} & \phi_1 \vee \phi_2 \stackrel{\text{def}}{=} \neg(\neg\phi_1 \wedge \neg\phi_2) \\
\phi_1 \rightarrow \phi_2 \stackrel{\text{def}}{=} \neg(\phi_1 \wedge \neg\phi_2) & (\forall\alpha)\phi \stackrel{\text{def}}{=} \neg(\exists\alpha)\neg\phi \\
\nu Z.\phi \stackrel{\text{def}}{=} \neg\mu Z.\neg\phi[Z \mapsto \neg Z] & 
\end{array}$$

**Remark 1 (Semantics of DeMorgan duals).** In every mixed transition system with state set  $\Sigma$  for every mode  $m \in \{a, c\}$ , state  $s \in \Sigma$ , and environment  $\rho$ , we have

- 1  $s \not\models^m \text{ff}$ ;
- 2  $s \models^m \phi_1 \vee \phi_2$  iff  $s \models^m \phi_1$  or  $s \models^m \phi_2$ ;
- 3  $s \models^m \phi_1 \rightarrow \phi_2$  iff  $s \not\models^m \phi_1$  or  $s \models^m \phi_2$ ;
- 4  $s \models^m (\forall \alpha) \phi$  iff for all  $s' \in \Sigma$ ,  $(s, \alpha, s') \in R^{-m}$  implies  $s' \models^m \phi$ ;
- 5 If  $\Sigma$  is finite or if  $\nu Z.\phi$  is unnested (Lar90), then  $s \models^m \nu Z.\phi$  iff there is some  $l \geq 0$  such that  $s \not\models^m \mu_l Z.\neg\phi[Z \mapsto \neg Z]$ .

The last three items of Remark 1 highlight the treatment of negation in mixed transition systems: In mode  $m$ , we can (3) verify an implication by refuting its premise in the dual mode or by verifying its conclusion in the original mode; (4) verify a universally branching formula  $(\forall \alpha) \phi$  by showing that all  $R^{-m}$ -successor states satisfy  $\phi$  in mode  $m$ ; and (5) verify a greatest fixed point  $\nu Z.\phi$  by refuting some syntactic approximation of a dual least fixed point in the dual mode. The derivation of (4) is instructive:  $s \models^m (\forall \alpha) \phi$  iff  $s \models^m \neg(\exists \alpha) \neg\phi$  iff not  $(s \models^m (\exists \alpha) \neg\phi)$  iff not (for some  $s'$ ,  $(s, \alpha, s') \in R^{-m}$  and  $s' \models^m \neg\phi$ ) iff not (for some  $s'$ ,  $(s, \alpha, s') \in R^{-m}$  and not  $(s' \models^m \phi)$ ) iff for all  $s'$ ,  $(s, \alpha, s') \in R^{-m}$  implies  $s' \models^m \phi$ .

## 2.5. Totality

Modal transition systems are partial systems whose total versions render an established model-checking framework.

**Theorem 2 (Totality for modal transition systems).** Let  $\mathcal{M} = (\Sigma, R^a, R^c)$  be a modal transition system with signature  $\text{Act}$  that is total:  $R^a = R^c$ . Then, for all  $\phi \in \mathbb{L}$  and  $\rho$  with  $\rho^a = \rho^c$ :  $\models \phi \models_\rho^a$  equals  $\models \phi \models_\rho^c$ ; the semantics  $\models \phi \models_\rho^m$  is the usual one for the labeled transition system  $(\Sigma, R^m)$ ; and every refinement in  $\mathcal{M}$  — in particular  $\prec_{\mathcal{M}}$  — is a bisimulation.

*Proof of Theorem 2*

- 1 The proof that  $\models \phi \models_\rho^a = \models \phi \models_\rho^c$  is a straightforward induction which uses  $\rho^a = \rho^c$  for clause  $Z$  and  $R^a = R^c$  for clause  $(\exists \alpha) \phi$ .
- 2 If  $\models \phi \models_\rho^a = \models \phi \models_\rho^c$  for all  $\phi$ , then the semantics in Figure 5 is the standard one for labeled transition systems with signature  $\text{Act}$ .
- 3 If  $R^a = R^c$ , then Definition 3 is the one of a bisimulation.

□

Theorem 2 justifies our liberal use of  $\models$  for  $\models^a$  and  $\models^c$  over total models.

## 2.6. Maximal consistency

For a pointed mixed transition system  $(\mathcal{M}, i)$  with  $i \models^a \phi$  it is desirable that  $\phi$  be satisfiable in a refining total model:  $j \models \phi$  for a total model  $(\mathcal{N}, j)$  with  $(\mathcal{N}, j) \prec (\mathcal{M}, i)$ .

**Definition 4 (Consistent mixed transition systems).** A mixed transition system  $\mathcal{M}$  is *consistent* iff for all its states  $s$  and for all  $\phi \in \mathbb{L}$ ,  $s \models^a \phi$  implies that  $\phi$  is satisfiable

over some total refining model: there is some state  $t$  in some labeled transition system  $\mathcal{L}$  such that  $t \models \phi$  and  $(\mathcal{L}, t) \prec (\mathcal{M}, s)$ .

**Example 4 (An inconsistent mixed transition system).** Consider the mixed transition system consisting of one state  $s$  and one  $R^a \setminus R^c$  self-transition labeled by  $\alpha$ :  $\mathcal{M} = (\{s\}, \{(s, \alpha, s)\}, \emptyset)$ .

Because of the guaranteed transition, we have that  $s \models^a (\exists \alpha) \text{tt}$ , but the lack of  $R^c$ -transitions implies that  $s \not\models^c (\exists \alpha) \text{tt}$ . By the semantics of negation and conjunction, we infer  $s \models^a (\exists \alpha) \text{tt} \wedge \neg(\exists \alpha) \text{tt}$ , but the latter formula clearly is not satisfiable in labeled transition systems. In particular, it cannot be satisfied in some refining total model.

We now present a condition on mixed transition systems that guarantees their consistency and is met in all modal transition systems. This condition has an established domain-theoretic analogue (Hec90; Gun92), used in the next section to build a universal domain of consistent mixed transition systems.

**Definition 5 (The mix condition (MC)).** A mixed transition system  $\mathcal{M} = (\Sigma, R^a, R^c)$  satisfies the *mix condition* (MC) iff for all  $(s, \alpha, s') \in R^a$ , there is some  $s'' \in \Sigma$  such that  $(s, \alpha, s'') \in R^a \cap R^c$  and  $s'' \prec s'$ .

Condition (MC) is satisfied for all modal transition systems, since we may choose  $s''$  to be  $s'$  whenever  $R^a \subseteq R^c$ . Conversely, any mixed transition system that satisfies (MC) has a modal transition system as a *saturated version*:

**Definition 6 (Saturated mixed transition system).** For every mixed transition system  $\mathcal{M} = (\Sigma, R^a, R^c)$ , we define the *saturated mixed transition system*  $\tilde{\mathcal{M}} = (\Sigma, \tilde{R}^a, \tilde{R}^c)$ , where  $\tilde{R}^a \stackrel{\text{def}}{=} R^a \cap R^c$  and  $\tilde{R}^c \stackrel{\text{def}}{=} R^c$ .

Unlike the definitions above, the alternative one of letting  $\tilde{R}^a$  be  $R^a$  and  $\tilde{R}^c$  be the union of  $R^a$  and  $R^c$  renders a modal transition system that is *not* equivalent to  $\mathcal{M}$ .

**Proposition 1.** Let  $\mathcal{M} = (\Sigma, R^a, R^c)$  be a mixed transition system with start state  $i$  satisfying condition (MC). Then  $(\tilde{\mathcal{M}}, i)$  is a pointed modal transition system such that  $(\tilde{\mathcal{M}}, i) \prec_{\mathcal{M}} (\mathcal{M}, i)$  and  $(\mathcal{M}, i) \prec_{\mathcal{M}} (\tilde{\mathcal{M}}, i)$ .

*Proof of Proposition 1* Let  $s \prec_{\mathcal{M}} t$ .

— We show that  $(\tilde{\mathcal{M}}, i) \prec_{\mathcal{M}} (\mathcal{M}, i)$ :

- If  $(t, \alpha, t') \in R^a$ , then  $s \prec_{\mathcal{M}} t$  implies the existence of some  $(s, \alpha, s') \in R^a$  such that  $s' \prec_{\mathcal{M}} t'$ . Using the condition (MC), there exists some  $(s, \alpha, s'') \in R^a \cap R^c = \tilde{R}^a$  such that  $s'' \prec_{\mathcal{M}} s'$ . By transitivity of  $\prec_{\mathcal{M}}$ , we get  $s'' \prec_{\mathcal{M}} t'$  and have  $(s, \alpha, s'') \in \tilde{R}^a$ .
- If  $(s, \alpha, s') \in \tilde{R}^c = R^c$ , then  $s \prec_{\mathcal{M}} t$  implies  $(t, \alpha, t') \in R^c$  for some  $t' \in \Sigma$  such that  $s' \prec_{\mathcal{M}} t'$ .

— We show that  $(\mathcal{M}, i) \prec_{\mathcal{M}} (\tilde{\mathcal{M}}, i)$ :

- If  $(t, \alpha, t') \in \tilde{R}^a = R^a \cap R^c$ , then  $s \prec_{\mathcal{M}} t$  implies  $(s, \alpha, s') \in R^a$  for some  $s' \in \Sigma$  such that  $s' \prec_{\mathcal{M}} t'$ .

- If  $(s, \alpha, s') \in R^c$ , then  $s \prec_{\mathcal{M}} t$  implies the existence of some  $(t, \alpha, t') \in R^c = \tilde{R}^c$  such that  $s' \prec_{\mathcal{M}} t'$ .

□

This result informs us that any mixed transition system that meets condition (MC) is merely an unsaturated version of a modal transition system and that these two systems cannot be distinguished via observations through  $\models^a$  or  $\models^c$  (by Theorem 1). In that sense, modal transition systems and mixed transition systems satisfying condition (MC) are equally expressive for purposes of design and analysis; these systems are all consistent.

**Theorem 3.** Let  $\mathcal{M}$  be a mixed transition system with signature  $\text{Act}$  that satisfies the mix condition (MC).

- 1  $\mathcal{M}$  is consistent.
- 2 For every  $\rho$  with  $\rho^a(Z) \subseteq \rho^c(Z)$ ,  $Z \in \text{var}$ , and every  $\phi \in \mathbb{L}$ :  $\|\phi\|_{\rho}^a \subseteq \|\phi\|_{\rho}^c$ .
- 3 for every  $\rho$  with  $\rho^a(Z) \subseteq \rho^c(Z)$ ,  $Z \in \text{var}$ , and every  $\phi \in \mathbb{L}$ :  $\|\phi \wedge \neg\phi\|_{\rho}^a = \{\}$ .

*Proof of Theorem 3*

- As for (1), let  $\phi \in \mathbb{L}$  such that  $(\mathcal{M}, s) \models_{\rho}^a \phi$ . Define  $\mathcal{N} = (\Sigma, R^a \cap R^c, R^a \cap R^c)$ . An analogous reasoning as for the first part of Proposition 1 then renders  $(\mathcal{N}, s) \prec_{\mathcal{M}} (\mathcal{M}, s)$ . Thus,  $(\mathcal{M}, s) \models_{\rho}^a \phi$  implies  $(\mathcal{N}, s) \models_{\rho}^a \phi$  by Theorem 1. Applying Theorem 2 to  $\mathcal{N}$ , we infer that  $s \models_{\rho}^a \phi$  in the labeled transition system  $\mathcal{L} \stackrel{\text{def}}{=} (\Sigma, R^a \cap R^c)$ , so  $\mathcal{M}$  is consistent.
- We prove (2) by structural induction on  $\phi$ ; the clauses for  $\text{tt}$ ,  $\neg$ ,  $\wedge$ , and  $\mu Z.\phi$  are routine. For clause  $Z$ , we use the assumption that  $\rho^a(Z) \subseteq \rho^c(Z)$  for every  $Z \in \text{var}$ . As for  $(\exists\alpha)$ , let  $s \in \|\exists\alpha\phi\|_{\rho}^a$ , i.e.  $s \models_{\rho}^a (\exists\alpha)\phi$ . Then there exists some  $(s, \alpha, s') \in R^a$  such that  $s' \models_{\rho}^a \phi$ . From (MC), we then infer the existence of some  $s'' \in \Sigma$  such that  $(s, \alpha, s'') \in R^a \cap R^c$  and  $s'' \prec s'$ . So  $s' \models_{\rho}^a \phi$  and  $s'' \prec s'$  imply  $s'' \models_{\rho}^a \phi$  by Theorem 1. By induction, this gives us  $s'' \models_{\rho}^c \phi$ . But then  $(s, \alpha, s'') \in R^c$  implies  $s \models_{\rho}^c (\exists\alpha)\phi$ .
- (2) and (3) are equivalent: the set  $\|\phi \wedge \neg\phi\|_{\rho}^a$  is non-empty iff there is some  $s \in \Sigma$  such that  $s \models_{\rho}^a \phi$  and  $s \not\models_{\rho}^c \phi$  iff there is an element in  $\|\phi\|_{\rho}^a \setminus \|\phi\|_{\rho}^c$  iff  $\|\phi\|_{\rho}^a$  is not a subset of  $\|\phi\|_{\rho}^c$ .

□

**Example 5 (More precise property semantics).** Our property semantics loses precision in two places: the interpretation of disjunction in the assertion mode a, and the interpretation of conjunction in the consistency checking mode c. For example, any formula  $\phi$  with  $s \models_{\rho}^c \phi$  and  $s \not\models_{\rho}^a \phi$  renders  $s \models_{\rho}^c \phi \wedge \neg\phi$  and  $s \not\models_{\rho}^a \phi \vee \neg\phi$ . Such loss of precision may severely impact the quality of an analysis. Various techniques exist for obtaining more precise interpretations, although at a significant increase in complexity: we mention the focus operation of Ball et al. (BPR01) for program analysis and Bruns and Godefroid’s generalized model checking (BG00).

2.7. An extension: non-flat data

To this point, modal transition systems have flat data sets: together,  $R^a$  and  $R^c$  partially specify a binary relation  $R$  over a *discrete* set  $\Sigma$ . We use the mixed powerdomain to generalize modal transition systems to non-flat sets, modeled as domains (AJ94).

**Definition 7 (Mixed powerdomain (Gun92; Hec90)).** Let  $(D, \leq)$  be a bifinite domain (Jun88) — a domain  $D$  such that its identity function  $\text{id}_D$  be the directed image of Scott-continuous functions  $d: D \rightarrow D$  with finite image such that  $d = d \circ d \leq \text{id}_D$ . The *mixed powerdomain*  $\mathbf{M}[D]$  of  $(D, \leq)$  consists of the set of all pairs  $(L, U)$ , where  $L$  is Scott-closed in  $(D, \leq)$  and  $U$  is a Scott-compact upper set in  $(D, \leq)$  such that  $L$  and  $U$  satisfy the consistency condition

$$L = \downarrow(L \cap U). \tag{3}$$

The order in  $\mathbf{M}[D]$  is given by

$$(L, U) \leq (L', U') \stackrel{\text{def}}{=} L \subseteq L' \text{ and } U' \subseteq U. \tag{4}$$

Here is why elements of a mixed powerdomain might be used as states in a mixed transition system: a state should be characterized by both the assertions  $L$  that are guaranteed to hold true for it and by the assertions  $U$  that are possibly true for it; the pair is consistent if  $L = \downarrow(L \cap U)$ . A state  $(L, U)$  should refine state  $(L', U')$  when  $(L, U) \leq (L', U')$ . These intuitions are formalized in the next section, but a small example is in order:

**Example 6 (Mixed powerdomains).**

- 1 For the domain  $D = \{*\}$ , each subset is Scott-closed and a Scott-compact upper set. However, the pair  $(L, U) = (\{*\}, \{\})$  does not satisfy the consistency condition (3) as the righthand side of (3) is then empty. The three remaining pairs  $\text{false} \stackrel{\text{def}}{=} (\{\}, \{\})$ ,  $\perp \stackrel{\text{def}}{=} (\{\}, \{*\})$ , and  $\text{true} \stackrel{\text{def}}{=} (\{*\}, \{*\})$  satisfy (3) and comprise all elements of  $\mathbf{M}[D]$ . As for the ordering, (4) informs us that  $\perp \leq \text{false}$  and  $\perp \leq \text{true}$  are the only non-reflexive instances of  $\leq$  in  $\mathbf{M}[D]$  (Hec90).
- 2 Let  $D$  be a finite set with a preorder  $\leq$ . Elements of  $\mathbf{M}[D]$  are pairs  $(L, U)$ , where  $L$  is a lower and  $U$  is an upper set with respect to  $\leq$ . If the ordering is flat, the consistency conditions reads as  $L \subseteq U$ .

As an element of a powerdomain, every  $(L, U) \in \mathbf{M}[D]$  models a “set”  $A$ . However, claims of the form, “Element  $d$  is contained in the ‘set’  $A$ ,” have *three*, instead of the conventional two, possible outcomes:  $\text{false}$  if  $d \notin U$ ;  $\text{true}$  if  $d \in L$ ; and  $\perp$  otherwise, i.e. if  $d \in U \setminus L$ . The Scott-closed set  $L$  specifies firm guarantees of membership, whereas the Scott-compact upper set  $U$  specifies the possibility of membership. Naturally, this three-valued interpretation of membership determines a three-valued interpretation of existential quantification, as worked out in (Hec90). Non-flat data routinely arises in the framework of abstract interpretation (CC77).

**Example 7 (Multiple viewpoints).** Non-flat applications of modal transition systems also occur in software engineering in the context of requirements analysis and consistency

checking (NKF94). In a simplified scenario, each element of a finite domain  $(D, \leq)$  is a pointed modal transition system  $d$  and  $d \leq e$  expresses that  $e$  has higher or equal priority than  $d$ . Each  $d$  is a different view of a software artifact. Assertions validated at a viewpoint are obliged to hold at viewpoints of lesser priority. In light of Theorem 1, this means that properties consistent at a viewpoint are obliged to be consistent in viewpoints of higher priorities. A semantics collects these obligations of validity  $\{\mathcal{M}:\phi\}^a$  and consistency  $\{\mathcal{M}:\phi\}^c$  (HP02)

$$\begin{aligned} \{\mathcal{M}:\phi\}^a &\stackrel{\text{def}}{=} \{d \in D \mid \exists e \in D: d \leq e, e \models^a \phi\} \\ \{\mathcal{M}:\phi\}^c &\stackrel{\text{def}}{=} \{d \in D \mid \exists e \in D: e \leq d, e \models^c \phi\} \\ \{\mathcal{M}:\phi\} &\stackrel{\text{def}}{=} (\{\mathcal{M}:\phi\}^a, \{\mathcal{M}:\phi\}^c). \end{aligned} \quad (5)$$

In general,  $\{\mathcal{M}:\phi\}^a$  won't be a subset of  $\{\mathcal{M}:\phi\}^c$ , but  $\{\mathcal{M}:\phi\}$  is an element of  $\mathbf{M}[D]$ , since  $e \models^a \phi$  implies  $e \models^c \phi$  for pointed modal transition systems  $e$  by Theorem 3.2. Given an inconsistent set  $\Phi$  of properties,  $\bigcap_{\phi \in \Phi} \{\mathcal{M}:\phi\}^a$  identifies viewpoints that are impacted by this inconsistency. For a full exposition of this semantics and its usage in the detection, location, and mitigation of inconsistencies we refer to (HP02).

### 3. A domain equation for modal transition systems

Powerdomains (Plo76; Smy78; AJ94) are recognized and widely used as spaces of meaning for denotational semantics of systems that exhibit non-determinism. Powerdomains that are the initial solution to a domain equation have also been used as internally fully abstract models of systems that specify concurrent systems and their abstraction order. For example, Abramsky (Abr91) used an adaptation of the convex powerdomain (Plo76) to model labeled transition systems and partial bisimulations. In this section, we apply the machinery of powerdomains and domain equations to provide a domain-theoretic model for mixed transition systems that meet condition (MC) and their refinement. It is a pleasant surprise that the mixed powerdomain, discovered independently by Gunter (Gun92) and Heckmann (Hec90), serves as a ready-to-use meaning space for this task. Throughout this section, we assume a fixed finite signature  $\text{Act}$  and use the well known topological representations of powerdomains. For the purpose at hand, we work with countably based bifinite domains (Jun88).

For simplicity, the items of the remark below represent Scott-closed subsets as sets of lower sets of compact elements. For  $x \in (D, \leq)$  we write  $\uparrow x = \{y \in D \mid x \leq y\}$ .

**Remark 2 (Universal property of the mixed powerdomain (Hec90)).** For every countably based bifinite domains  $D$  and  $E$ ,

- 1  $\mathbf{M}[D]$  is a countably based bifinite domain;
- 2 the map  $d \mapsto \{d\} : D \rightarrow \mathbf{M}[D]$ , defined by  $\{d\} \stackrel{\text{def}}{=} (\{k \in D \mid k \leq d, k \text{ compact}\}, \uparrow d)$ , is continuous;
- 3 the formal union operator  $\bar{\cup} : \mathbf{M}[D] \times \mathbf{M}[D] \rightarrow \mathbf{M}[D]$ , defined by

$$(L, U) \bar{\cup} (L', U') \stackrel{\text{def}}{=} (L \cup L', U \cup U') \quad (6)$$

is continuous;

- 4 for any continuous function  $f: D \rightarrow M[E]$ , there exists a unique continuous map  $\bar{f}: M[D] \rightarrow M[E]$  such that  $\bar{f} \circ \bar{\cup} = \cup \circ \bar{f} \times \bar{f}$  and  $\bar{f} \circ \{\cdot\} = f$ ;
- 5 all compact elements of  $M[D]$  are obtained by a finite application of the constant  $\bar{\emptyset} \stackrel{\text{def}}{=} (\emptyset, \emptyset)$  and the operations  $\bar{\cup}$ ,  $\{\cdot\}$ , and  $\{\cdot?\}$  to compact elements of  $D$ ; where  $\{d?\} \stackrel{\text{def}}{=} (\{\}, \uparrow d)$ .

### 3.1. The universal domain as a fully abstract mixed transition system

For mixed transition systems, our discussion of membership in  $M[D]$  suggests to use Scott-closed sets as a model of the set of  $R^a$ -successors of a state and to use Scott-compact upper sets as a model of the set of  $R^c$ -successors of a state. Fortunately, there is an intimate connection between the condition (MC), which guarantees consistency, and the domain-theoretic consistency condition (3).

**Definition 8 (Universal domain).** In the style of Abramsky's domain equation for partial bisimulation (Abr91), we let  $\mathcal{D}_{\text{Act}}$  be the initial solution to the domain equation

$$D = \prod_{\alpha \in \text{Act}} M[D] \quad (7)$$

over bifinite domains and Scott-continuous maps, where  $\prod_{\alpha \in \text{Act}} D_\alpha$  denotes the categorical product of the domains  $D_\alpha$  whose elements are tuples  $(d_\alpha)_{\alpha \in \text{Act}}$  with  $d_\alpha \in D_\alpha$  for all  $\alpha \in \text{Act}$ . We write  $\mathcal{D}$  for  $\mathcal{D}_{\text{Act}}$  if  $\text{Act}$  is determined by context or irrelevant. We write  $\perp_{\mathcal{D}}$  for the bottom element  $((\emptyset, \mathcal{D}))_{\alpha \in \text{Act}}$  of  $\mathcal{D}$ .

We note that  $\mathcal{D}$  is well defined since  $M[\cdot]$  and  $\prod$  are locally continuous functors in the category of countably based bifinite domains and Scott-continuous maps (Hec90; AJ94). According to (7), any element  $d$  of  $\mathcal{D}$  corresponds to a tuple of pairs  $((L_\alpha, U_\alpha))_{\alpha \in \text{Act}}$ , where  $(L_\alpha, U_\alpha) \in M[D]$  for each  $\alpha \in \text{Act}$ .

**Definition 9 (Universal domain as a mixed transition system).**

- 1 For every  $d = ((L_\beta, U_\beta))_{\beta \in \text{Act}} \in \mathcal{D}$  and  $\alpha \in \text{Act}$ , we define

$$d_\alpha \stackrel{\text{def}}{=} (L_\alpha, U_\alpha), \quad d_\alpha^a \stackrel{\text{def}}{=} L_\alpha, \quad d_\alpha^c \stackrel{\text{def}}{=} U_\alpha. \quad (8)$$

- 2 We define state transition relations  $\mathcal{R}^m \subseteq \mathcal{D} \times \text{Act} \times \mathcal{D}$ :

$$\begin{aligned} \mathcal{R}^a &\stackrel{\text{def}}{=} \{(d, \alpha, d') \mid \alpha \in \text{Act}, d' \in d_\alpha^a\} \\ \mathcal{R}^c &\stackrel{\text{def}}{=} \{(d, \alpha, d') \mid \alpha \in \text{Act}, d' \in d_\alpha^c\}. \end{aligned} \quad (9)$$

We note that suprema in  $\mathcal{D}$  are computed component-wise. Therefore,  $d \in \mathcal{D}$  is compact in  $\mathcal{D}$  iff for all  $\alpha \in \text{Act}$ ,  $d_\alpha$  is compact in  $D$ .

**Remark 3 (Elements of  $\mathcal{D}$  as pointed systems).** Each element  $d$  of  $\mathcal{D}$  represents a pointed mixed transition system. The start state is  $d$ , and its sets of  $\mathcal{R}^a$ -reachable and  $\mathcal{R}^c$ -reachable states are defined inductively in the standard manner via (9). Note that

$$\begin{array}{ll}
\neg_a \neg_c \phi = \phi & \neg_c \neg_a \phi = \phi \\
\phi \leq_a \psi \Rightarrow \neg_a \psi \leq_c \neg_a \phi & \phi \leq_c \psi \Rightarrow \neg_c \psi \leq_a \neg_c \phi
\end{array}$$

Fig. 6. Axioms for AC-lattices.

the operation  $\prod_{\alpha \in \text{Act}} \bar{\cup}$  has type  $\mathcal{D} \times \mathcal{D} \rightarrow \mathcal{D}$ , using the isomorphism implicit in the “=” of (7) as a casting and the general distributivity of products; it elegantly models the sum of pointed mixed transition systems.

As a mixed transition system, the domain  $\mathcal{D}$  has a greatest refinement  $\prec_{\mathcal{D}}$ . We can already prove one half of the statement that the relational inverse of  $\prec_{\mathcal{D}}$  is the order on the domain  $\mathcal{D}$ .

**Proposition 2 (Order of universal domain as abstraction).** The relational inverse of the ordering on  $\mathcal{D}$  is a refinement in the mixed transition system  $(\mathcal{D}, \mathcal{R}^a, \mathcal{R}^c)$ .

*Proof of Proposition 2* Let  $e \leq^{\text{op}} d$ , i.e.  $d \leq e$  in  $\mathcal{D}$ . Let  $\alpha \in \text{Act}$ .

- 1 If  $(d, \alpha, d') \in \mathcal{R}^a$ , then  $d' \in d'_\alpha$ . But then  $d \leq e$  implies  $d' \in d'_\alpha \subseteq e'_\alpha$ , i.e.  $(e, \alpha, d') \in \mathcal{R}^a$ ; clearly,  $d' \leq^{\text{op}} d'$ ;
- 2 if  $(e, \alpha, e') \in \mathcal{R}^c$ , then  $e' \in e'_\alpha$  follows. But then  $d \leq e$  implies  $e' \in e'_\alpha \subseteq d'_\alpha$ , i.e.  $(d, \alpha, e') \in \mathcal{R}^c$ ; clearly,  $e' \leq^{\text{op}} e'$ .

□

The mixed transition system  $(\mathcal{D}, \mathcal{R}^a, \mathcal{R}^c)$  is not a modal transition system since the inclusion  $\mathcal{R}^a \subseteq \mathcal{R}^c$  is a stronger condition than (3) for non-flat data. But condition (3) is simply the topological version of the mix condition (MC).

**Proposition 3 (Universal domain satisfies mix condition).** The mixed transition system  $(\mathcal{D}, \mathcal{R}^a, \mathcal{R}^c)$  satisfies the mix condition (MC).

*Proof of Proposition 3* Given  $(d, \alpha, d')$  in  $\mathcal{R}^a$ , we have  $d' \in d'_\alpha$ . By (3) and (7), there has to exist some  $d'' \in d'_\alpha \cap d''_\alpha$  (i.e.  $(d, \alpha, d'') \in \mathcal{R}^a \cap \mathcal{R}^c$ ) such that  $d' \leq d''$ , i.e.  $d'' \prec d'$ . □

That the relational inverse of the order in  $\mathcal{D}$  equals  $\prec_{\mathcal{D}}$  can be shown by a logical characterization of refinement for a Hennessy-Milner logic (HM85)  $\mathbb{L}_{\text{HM}}$ , defined by the grammar

$$\phi ::= \text{tt} \mid \neg \phi \mid \phi \wedge \phi \mid (\exists \alpha) \phi \quad (10)$$

where  $\alpha \in \text{Act}$ . Since  $\mathbb{L}_{\text{HM}}$  is a sublogic of  $\mathbb{L}$  without free variables and since  $(\mathcal{D}, \mathcal{R}^a, \mathcal{R}^c)$  is a mixed transition system, we infer that the subsets  $\llbracket \phi \rrbracket^a$  and  $\llbracket \phi \rrbracket^c$  of  $\mathcal{D}$  are well defined for all  $\phi \in \mathbb{L}_{\text{HM}}$ , as specified in Figure 5. These meanings are elements of an assertion-consistency lattice (AC-lattice).

**Definition 10 (AC-lattices (HP02)).** An AC-lattice is a tuple  $(\mathcal{L}_a, \leq_a, \neg_a, \mathcal{L}_c, \leq_c, \neg_c)$ , where  $(\mathcal{L}_a, \leq_a)$  and  $(\mathcal{L}_c, \leq_c)$  are partial orders that induce lattices, and  $\neg_a: \mathcal{L}_a \rightarrow \mathcal{L}_c$  and  $\neg_c: \mathcal{L}_c \rightarrow \mathcal{L}_a$  are functions that meet the axioms of Figure 6.



A canonical example of AC-lattices are topological spaces  $X$  where  $(\mathcal{L}^a, \leq^a)$  and  $(\mathcal{L}^c, \leq^c)$  are the lattice of all closed and open subsets of  $X$  (respectively) — ordered by set inclusion; and  $\neg^a$  and  $\neg^c$  are set complementation. In (HP02), it is shown that — up to an order-isomorphism — this example is exhaustive for finite, distributive AC-lattices. The sets  $\llbracket \phi \rrbracket^a$  and  $\llbracket \phi \rrbracket^c$  of  $\mathcal{D}$  form the elements of an AC-lattice within the canonical AC-lattice of the topological space  $(\mathcal{D}, \sigma(\mathcal{D}))$ , where  $\sigma(\mathcal{D})$  denotes the Scott-topology of  $\mathcal{D}$ .

**Definition 11 (AC-lattice operations in  $\mathcal{D}$ ).** For each  $m \in \{a, c\}$ , we define the partial order  $\mathbf{M}_m \stackrel{\text{def}}{=} \{\llbracket \phi \rrbracket^m \mid \phi \in \mathbb{L}_{\text{HM}}\}$ , ordered by inclusion, and a negation operation  $\neg^m : \mathbf{M}_m \rightarrow \mathbf{M}_{\neg m}$ :

$$\neg^m \llbracket \phi \rrbracket^m \stackrel{\text{def}}{=} \llbracket \neg \phi \rrbracket^{\neg m}. \quad (11)$$

**Theorem 4 (AC-lattice of  $\mathcal{D}$ ).**

- 1 The structure  $(\mathbf{M}_a, \subseteq, \neg^a, \mathbf{M}_c, \subseteq, \neg^c)$  is a distributive bounded AC-lattice, where  $\neg^a$  and  $\neg^c$  equal set complementation in the domain  $\mathcal{D}$ .
- 2 Each element of  $\mathbf{M}_a$  is Scott-open in  $\mathcal{D}$  and each element of  $\mathbf{M}_c$  is Scott-closed in  $\mathcal{D}$ .

*Proof of Theorem 4*

- 1 Since  $\llbracket \phi_1 \rrbracket^m \cap \llbracket \phi_2 \rrbracket^m = \llbracket \phi_1 \wedge \phi_2 \rrbracket^m$  and  $\llbracket \phi_1 \rrbracket^m \cup \llbracket \phi_2 \rrbracket^m = \llbracket \neg(\neg \phi_1 \wedge \neg \phi_2) \rrbracket^m$ , the partial orders  $(\mathbf{M}_m, \subseteq)$  determine lattices with bottom  $\llbracket \neg \text{tt} \rrbracket^m$  and top  $\llbracket \text{tt} \rrbracket^m$ . Since  $\neg \neg m = m$  and  $\llbracket \neg \neg \phi \rrbracket^m = \llbracket \phi \rrbracket^m$ , the first two axioms of Figure 6 are met. As for the remaining two axioms, let  $d \in \neg^m \llbracket \psi \rrbracket^m$ . Then  $d \models \neg^m \neg \psi$  implies  $d \not\models^m \psi$  which — assuming  $\llbracket \phi \rrbracket^m \subseteq \llbracket \psi \rrbracket^m$  — implies  $d \not\models^m \phi$ , i.e.  $d \in \neg^m \llbracket \phi \rrbracket^m$ . The last claim about  $\neg^m$  follows since  $d \models \neg^m \neg \phi$  iff  $d \not\models^m \neg \phi$  iff  $d \not\models^m \phi$  iff  $d \in \neg^m \llbracket \phi \rrbracket^m$ .
- 2a To see that  $\llbracket \phi \rrbracket^a$  is an upper set in  $\mathcal{D}$ , let  $d \in \llbracket \phi \rrbracket^a$  and  $d \leq e$  in  $\mathcal{D}$ . By Proposition 2,  $e \prec d$ . Since  $d \models^a \phi$ , Theorem 1 implies  $e \models^a \phi$ , i.e.  $e \in \llbracket \phi \rrbracket^a$ . An analogous proof shows that  $\llbracket \phi \rrbracket^c$  is a lower set in  $\mathcal{D}$ .
- 2b We show the remaining claims by simultaneous structural induction on (10):
  - (a) For  $\text{tt}$ , this is clear as  $\mathcal{D}$  is a Scott-closed and Scott-open subset of  $\mathcal{D}$ .
  - (b) For negation, this follows by induction from  $\llbracket \neg \phi \rrbracket^m = \mathcal{D} \setminus \llbracket \phi \rrbracket^{\neg m}$ .
  - (c) For conjunction, this follows by induction since  $\llbracket \phi_1 \wedge \phi_2 \rrbracket^m = \llbracket \phi_1 \rrbracket^m \cap \llbracket \phi_2 \rrbracket^m$ .
  - (d) For  $(\exists \alpha) \phi$ , the proof for each mode is different and mode  $c$  makes use of the Hofmann-Mislove Theorem (HM81).
    - i Let  $d \in \llbracket (\exists \alpha) \phi \rrbracket^a$  which we know to be an upper set. Let  $D$  be the set of compact elements  $k$  in  $\mathcal{D}$  such that  $k \leq d$ . For every  $k \in D$  and  $\alpha \in \text{Act}$ ,  $k_\alpha$  is compact and  $k_\alpha \leq d_\alpha$  in  $\mathbf{M}[D]$ . Then  $d_\alpha^a \cap \llbracket \phi \rrbracket^a$  equals the directed union  $\bigcup_{k \in D} k_\alpha^a \cap \llbracket \phi \rrbracket^a$ , using the fact that  $\mathcal{D}$  is algebraic and that  $\llbracket \phi \rrbracket^a$  is Scott-open by induction. Since the former set is non-empty, there has to be some  $k \in D$  for which  $k_\alpha^a \cap \llbracket \phi \rrbracket^a$  is non-empty as well. But then  $k \in \llbracket (\exists \alpha) \phi \rrbracket^a$  for that  $k$ .
    - ii We already know that  $\llbracket (\exists \alpha) \phi \rrbracket^c$  is a lower set. For  $D \subseteq \llbracket (\exists \alpha) \phi \rrbracket^c$ , where

$D$  is directed, let  $e$  be the supremum of  $D$  in  $\mathcal{D}$ . Proof by contradiction: if  $e$  is not in  $\llbracket (\exists\alpha) \phi \rrbracket^c$ , then  $e_\alpha^c$  is contained in  $\mathcal{D} \setminus \llbracket \phi \rrbracket^c$  which is Scott-open by induction. Since  $e_\alpha^c = \bigcap_{d \in D} d_\alpha^c$  is the filtered intersection of a family of Scott-compact upper (i.e. saturated) sets in the bifinite domain  $\mathcal{D}$ , we may invoke the Hofmann-Mislove Theorem (HM81) as bifinite domains are sober spaces (AJ94). Therefore, there is some  $d \in D$  for which  $d_\alpha^c$  is contained in  $\mathcal{D} \setminus \llbracket \phi \rrbracket^c$  already. But then  $d \in D \setminus \llbracket (\exists\alpha) \phi \rrbracket^c$  is a contradiction.  $\square$

Sets of the form  $\llbracket \phi \rrbracket^a$  are model-based versions of valid assertions: the collection of elements in  $\mathcal{D}$  for which property  $\phi$  can be successfully verified. Sets of the form  $\mathcal{D} \setminus \llbracket \phi \rrbracket^c$ , which equals  $\llbracket \neg\phi \rrbracket^a$ , are model-based versions of inconsistent assertions: the set of elements in  $\mathcal{D}$  for which property  $\phi$  is not consistent. All of these sets are Scott-open observables.

To show internal full abstraction — that the ordering of the domain equals the greatest abstraction relation of the domain viewed as a mixed transition system that meets condition (MC) — we need to prove that each upper set generated by a compact element of  $\mathcal{D}$  is a denotation of a formula of Hennessy-Milner logic in assertion mode a.

**Lemma 1 (Compact elements as denotations).** For every compact element  $k$  in  $\mathcal{D}$ , there exists some  $\phi_k \in \mathbb{L}_{\text{HM}}$  such that  $\llbracket \phi_k \rrbracket^a$  equals the upper set generated by  $k$  in  $\mathcal{D}$ .

*Proof of Lemma 1* For a domain  $E$ , we write  $\mathbf{K}(E)$  for the partial order of compact elements of  $E$ . For  $i \geq 0$ , let  $\mathcal{D}_i$  be the  $i$ th approximation of  $\mathcal{D}$  via its defining domain equation in (7). We prove the lemma by induction on  $i \geq 0$  for  $\mathbf{K}(\mathcal{D}_i)$ . This is sound since  $\mathbf{K}(\mathcal{D}) = \bigcup_{i \geq 0} \mathbf{K}(\mathcal{D}_i)$  by (7).

- For  $i = 0$ ,  $\mathbf{K}(\mathcal{D}_0)$  is a singleton set  $\{*\}$ , so we may choose  $\phi_* \stackrel{\text{def}}{=} \text{tt}$ .
- Let  $k \in \mathbf{K}(\mathcal{D}_{i+1})$ . Then each  $k_\alpha$  is compact in  $\mathbf{M}[\mathcal{D}_i]$ . We invoke Theorem 6.4 of (Hec90) to the approximating domain  $\mathcal{D}_i$ : each compact element of  $\mathbf{M}[\mathcal{D}_i]$  is obtained by a finite application of the constant  $\bar{\emptyset}$  and the operations  $\bar{\cup}$ ,  $\{\cdot\}$ , and  $\{\cdot?\}$  to compact elements of  $\mathcal{D}_i$ . Since  $\{\downarrow l\} = (\downarrow, \uparrow l)$ , and  $\{\uparrow l?\} = (\emptyset, \uparrow l)$ , and  $k_\alpha \in \mathbf{K}(\mathbf{M}[\mathcal{D}_i])$ ; we infer for all  $\alpha \in \text{Act}$  that  $(k_\alpha^a, k_\alpha^c) = (\downarrow F_\alpha, \uparrow G_\alpha)$  for some finite sets  $F_\alpha, G_\alpha \subseteq \mathbf{K}(\mathcal{D}_i)$ . By induction, for each  $x \in F_\alpha \cup G_\alpha$  there is some  $\phi_x \in \mathbb{L}_{\text{HM}}$  that satisfies the claim of the lemma for  $x$ . We use the abbreviations  $(\forall\alpha)$  and  $\bigvee$  to define

$$\begin{aligned} \psi_\alpha &\stackrel{\text{def}}{=} \bigwedge_{l \in F_\alpha} (\exists\alpha) \phi_l \\ \eta_\alpha &\stackrel{\text{def}}{=} (\forall\alpha) \bigvee_{m \in G_\alpha} \phi_m \\ \phi_k &\stackrel{\text{def}}{=} \bigwedge_{\alpha \in \text{Act}} \psi_\alpha \wedge \eta_\alpha. \end{aligned}$$

Note that  $\phi_k \in \mathbb{L}_{\text{HM}}$  since  $\text{Act}$  is finite.

- 1 We show  $k \in \llbracket \phi_k \rrbracket^a$ , i.e.  $\uparrow k \subseteq \llbracket \phi_k \rrbracket^a$ . Let  $\alpha \in \text{Act}$ .
  - (a) For  $l \in F_\alpha$  we have  $\uparrow l = \llbracket \phi_l \rrbracket^a$  by induction, so  $l \in k_\alpha^a \cap \llbracket \phi_l \rrbracket^a$  since  $l \in \downarrow F_\alpha = k_\alpha^a$ . But then  $k \models^a (\exists \alpha) \phi_l$ . Therefore,  $k \models^a \psi_\alpha$ .
  - (b) We have  $k \models^a (\forall \alpha) \bigvee \phi_m$  iff  $k \not\models^c (\exists \alpha) \neg \bigvee \phi_m$  iff  $k_\alpha^c \cap \llbracket \neg \bigvee \phi_m \rrbracket^c = \{\}$  iff  $k_\alpha^c \subseteq \mathcal{D} \setminus \llbracket \neg \bigvee \phi_m \rrbracket^c = \llbracket \bigvee \phi_m \rrbracket^a$ . Since the latter is an upper set and since  $k_\alpha^c$  equals  $\uparrow G_\alpha$ , it suffices to show  $G_\alpha \subseteq \llbracket \bigvee_{m \in G_\alpha} \phi_m \rrbracket^a$ . But

$$\uparrow G_\alpha = \llbracket \bigvee_{m \in G_\alpha} \phi_m \rrbracket^a \quad (12)$$

follows from induction and Remark 1.2. Thus,  $k \models^a \eta_\alpha$ .

- 2 Let  $d \in \mathcal{D}$  such that  $d \models^a \phi_k$ . We have to show  $k \leq d$  in  $\mathcal{D}$ , i.e.  $F_\alpha \subseteq d_\alpha^a$  and  $d_\alpha^c \subseteq \uparrow G_\alpha$  for all  $\alpha \in \text{Act}$ . For every  $\alpha \in \text{Act}$ ,  $d \models^a \phi_k$  implies
  - (a)  $d \models^a \psi_\alpha$ , so for all  $l \in F_\alpha$  there is some  $l' \in d_\alpha^a$  such that  $l' \models^a \phi_l$ . By induction,  $l \leq l'$  follows. Thus,  $F_\alpha \subseteq \downarrow d_\alpha^a = d_\alpha^a$ .
  - (b)  $d \models^a \eta_\alpha$  which is equivalent to  $d_\alpha^c \subseteq \llbracket \bigvee_{m \in G_\alpha} \phi_m \rrbracket^a$ . By (12), we get  $d_\alpha^c \subseteq \uparrow G_\alpha$ .  $\square$

**Theorem 5 (Internal full abstraction and logical characterization).** The following are equivalent:

- 1  $d \leq e$  in the domain  $\mathcal{D}$ ;
- 2  $e \prec_{\mathcal{D}} d$  in the mixed transition system  $(\mathcal{D}, \mathcal{R}^a, \mathcal{R}^c)$ ;
- 3  $\{\phi \in \mathbb{L}_{\text{HM}} \mid d \models^a \phi\} \subseteq \{\phi \in \mathbb{L}_{\text{HM}} \mid e \models^a \phi\}$ ; and
- 4  $\{\phi \in \mathbb{L}_{\text{HM}} \mid e \models^c \phi\} \subseteq \{\phi \in \mathbb{L}_{\text{HM}} \mid d \models^c \phi\}$ .

*Proof of Theorem 5* We show (1)  $\Rightarrow$  (2)  $\Rightarrow$  (4)  $\Rightarrow$  (3)  $\Rightarrow$  (1). The first two implications follow from Proposition 2 and Theorem 1 (respectively). To show (4)  $\Rightarrow$  (3), let  $d \models^a \phi$ . Then we have  $d \not\models^c \neg \phi$  which implies  $e \not\models^c \neg \phi$ , by (4), i.e.  $e \models^a \phi$ . But (3)  $\Rightarrow$  (1) follows directly from Lemma 1, noting that  $\mathcal{D}$  is algebraic.  $\square$

### 3.2. Embedding modal transition systems into the universal domain

We already argued that the domain  $\mathcal{D}$  is an internally fully abstract model of a mixed transition system that meets condition (MC). We now demonstrate its universality by embedding every mixed transition system that satisfies condition (MC) into  $\mathcal{D}$  such that one system refines another iff this is the case for their corresponding embeddings in  $\mathcal{D}$ : the embedding preserves and reflects refinements. As a by-product, we obtain that the assertion check semantics  $\models^a$  for Hennessy-Milner logic characterizes refinement of mixed transition systems that meet condition (MC). Since elements of  $\mathcal{D}$  correspond to pointed mixed transition systems, we work with pointed mixed transition systems  $(\mathcal{M}, i)$ .

We approximate pointed mixed transition systems  $(\mathcal{M}, i)$  by a family of finite-state pointed mixed transition systems  $(\mathcal{M}[n], i)$ ,  $n \geq 0$ . Our intent is to define the embedding  $\llbracket (\mathcal{M}, i) \rrbracket$  as the directed supremum of the embeddings  $\llbracket (\mathcal{M}[n], i) \rrbracket$  ( $n \geq 0$ ) in  $\mathcal{D}$ .

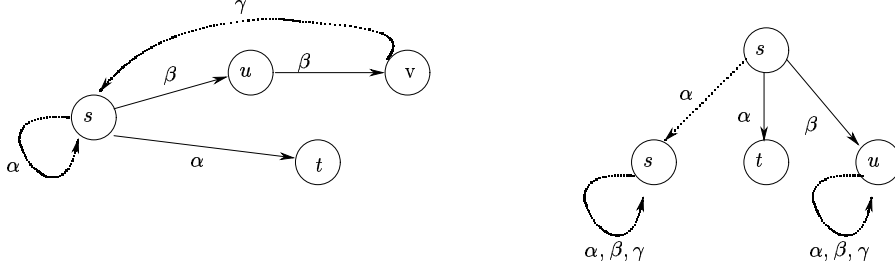


Fig. 7. To the left: a pointed modal transition system  $(\mathcal{M}, s)$ . To the right: its finite approximation  $(\mathcal{M}[1], s)$ , where  $s$  is unfolded once. ( $R^c$ -transitions that shadow  $R^a$ -transitions are omitted.)

**Definition 12 (Finite approximation systems).** Let  $(\mathcal{M}, i) = ((\Sigma, R^a, R^c), i)$  be a pointed mixed transition system. For each  $n \geq 0$ , we define a finite pointed mixed transition system  $(\mathcal{M}[n], i) = ((\Sigma[n], R[n]^a, R[n]^c), i)$  by induction on  $n$ .

- 1 The mixed transition system  $(\mathcal{M}[0], i)$  has no  $R^a$ -transitions and state set  $\Sigma[0] = \{i\}$ ; its set of  $R^c$ -transitions equals  $\{(i, \alpha, i) \mid \alpha \in \text{Act}\}$ .
- 2 Assume that  $(\mathcal{M}[n], i)$  is defined for all mixed transition systems  $(M, i)$ . Let  $\alpha \in \text{Act}$  and  $A_\alpha$  and  $C_\alpha$  be the set of  $R_\alpha^a$ -successors and  $R_\alpha^c$ -successors of  $i$  in  $(M, i)$  respectively. In  $(\mathcal{M}[n+1], i)$ , state  $i$  has as  $R_\alpha^a$ -successors all pointed mixed transition systems  $(\mathcal{M}[n], a)$  with  $a \in A_\alpha$ , where transitions are interpreted between pointed systems. Similarly, in  $(\mathcal{M}[n+1], i)$  state  $i$  has all pointed mixed transition systems  $(\mathcal{M}[n], c)$  with  $c \in C_\alpha$  as  $R_\alpha^c$ -successors. The state set  $\Sigma[n+1]$  is the disjoint sum of  $\{i\}$  and the state sets of  $(\mathcal{M}[n], l)$  for all  $l \in \bigcup_\alpha A_\alpha \cup C_\alpha$ .

Note that  $(\mathcal{M}[n+1], i)$  has no transitions whatsoever for all  $n \geq 0$  if  $i$  has no successor state in  $(M, i)$ .

**Example 8 (The first two approximations).**

- 1 Our embedding will map the initial approximation  $(\mathcal{M}[0], i)$  to the least element  $((\{\}, \mathcal{D})_{\alpha \in \text{Act}})$  of  $\mathcal{D}$ , where  $\{\}$  models the absence of  $R^a$ -transitions and  $\mathcal{D}$  models the set of  $R^c$ -transitions  $\{(i, \alpha, i) \mid \alpha \in \text{Act}\}$ .
- 2 Figure 7 depicts a pointed modal transition system  $(\mathcal{M}, s)$  and its approximation  $(\mathcal{M}[1], s)$ . In  $\mathcal{M}[1]$ , there are no transitions out of  $t$ . State  $v$  is not present/reachable.

Our finite approximations have the expected and desired properties.

**Proposition 4 (Finite approximations are monotone).** Let  $(\mathcal{N}, j)$  and  $(\mathcal{M}, i)$  be mixed transition systems.

- 1 For all  $n \geq 0$ ,  $(\mathcal{M}, i) \prec (\mathcal{M}[n], i)$ .
- 2 For all  $n \geq 0$ ,  $(\mathcal{M}[n+1], i) \prec (\mathcal{M}[n], i)$ .
- 3  $(\mathcal{N}, j) \prec (\mathcal{M}, i)$  iff for all  $n \geq 0$ ,  $(\mathcal{N}[n], j) \prec (\mathcal{M}[n], i)$  iff for all  $n \geq 0$ ,  $(\mathcal{N}, j) \prec (\mathcal{M}[n], i)$ .

*Proof of Proposition 4*

- 1 Let  $(s, t) \in \Sigma \times \Sigma[n]$  be in  $Q$  iff

- $t \in \Sigma[n]$  is an unfolded version of  $s \in \Sigma$  and  $t$  does not occur as the  $n$ th state on any path in  $\mathcal{M}[n]$  beginning in  $i$ ; or
- $t$  is the  $n$ th state for some path in  $\mathcal{M}[n]$  beginning in  $i$ ,  $\bar{s}$  is the folded version of  $t$ , and either  $\bar{s} = s$  or there is a path in  $\mathcal{M}$  beginning in  $i$  on which  $s$  occurs after  $\bar{s}$ .

We claim that  $Q$  is a refinement. Let  $(s, t) \in Q$ .

(a) Given  $(t, \alpha, t') \in R[n]^a$ ,

- let  $t$  be an unfolded version of  $s$  such that  $t$  does not occur as the  $n$ th state on any path in  $\mathcal{M}[n]$  beginning in  $i$ ; then  $(s, \alpha, s') \in R^a$ , where  $s'$  is the folded version of  $t'$ . Regardless of whether  $t'$  is “an  $n$ th state” or not, we infer  $(s', t') \in Q$  by the definition of  $Q$ ;
- then  $t$  cannot be the  $n$ th state for some path in  $\mathcal{M}[n]$  beginning in  $i$ .

(b) Given  $(s, \alpha, s') \in R^c$ ,

- let  $t$  be an unfolded version of  $s$  such that  $t$  does not occur as the  $n$ th state on any path in  $\mathcal{M}[n]$  beginning in  $i$ ; then  $(t, \alpha, t') \in R[n]^c$ , where  $t'$  is the unfolded version of  $s'$ . Regardless of whether  $t'$  is “an  $n$ th state” or not, we infer  $(s', t') \in Q$  by the definition of  $Q$ ;
- if  $t$  is the  $n$ th state for some path in  $\mathcal{M}[n]$  beginning in  $i$ ;  $\bar{s}$  is the folded version of  $t$ , and either  $\bar{s} = s$  or there is a path in  $\mathcal{M}$  on which  $s$  occurs after  $\bar{s}$ , then  $(t, \alpha, t) \in R[n]^c$  by the definition of  $\mathcal{M}[n]$ . We readily infer  $(s', t) \in Q$  since  $s'$  occurs after  $\bar{s}$  on some path in  $\mathcal{M}$ .

- 2 This proof is identical to the one of the previous item, except that one replaces  $(\mathcal{M}, i)$  with  $(\mathcal{M}[n+1], i)$ .
- 3 Since all mixed transition systems in this paper are finitely-branching, the greatest refinement  $\prec$  between  $(\mathcal{N}, j)$  and  $(\mathcal{M}, i)$  is the intersection of its finite approximants  $\prec_n$  ( $n \geq 0$ ) of the greatest fixed-point iterations. Thus, it suffices to show, for all  $n \geq 0$ , that  $(\mathcal{N}, j) \prec_n (\mathcal{M}, i)$  iff  $(\mathcal{N}[n], j) \prec_n (\mathcal{M}[n], i)$  iff  $(\mathcal{N}, j) \prec_n (\mathcal{M}[n], i)$ ; this is routine.

□

Next, we need to define the embeddings  $\langle (\mathcal{M}[n], i) \rangle$  for all  $n \geq 0$ .

**Proposition 5 (Embedding approximants into  $\mathcal{D}$ ).** Let  $(\mathcal{M}, i)$  be a pointed mixed transition system that satisfies condition (MC) and has state set  $\Sigma$ . For every  $s \in \Sigma$  and  $n \geq 0$ , we can construct a compact element  $\langle (\mathcal{M}[n], s) \rangle$  in  $\mathcal{D}$  such that

$$(\mathcal{M}[n], s) \prec (\mathcal{D}, \langle (\mathcal{M}[n], s) \rangle) \quad \text{and} \quad (\mathcal{D}, \langle (\mathcal{M}[n], s) \rangle) \prec (\mathcal{M}[n], s). \quad (13)$$

*Proof of Proposition 5* We proceed by induction on  $n$  for all approximants of the form  $(\mathcal{M}[n], i)$ . In each inductive step, we construct concrete refinements  $Q_1^i \subseteq \Sigma[n] \times \mathcal{D}$  and  $Q_2^i \subseteq \mathcal{D} \times \Sigma[n]$  that verify (13).

- Let  $n = 0$ . We set  $\langle (\mathcal{M}[0], i) \rangle \stackrel{\text{def}}{=} \perp_{\mathcal{D}}$ ,  $Q_1^i \stackrel{\text{def}}{=} \{(i, \perp_{\mathcal{D}})\}$ , and  $Q_2^i \stackrel{\text{def}}{=} \{(\perp_{\mathcal{D}}, i)\}$ . The element  $\langle (\mathcal{M}[0], i) \rangle$  is compact in  $\mathcal{D}$ .

— Let such embeddings be well defined for approximants of pointed systems for all  $k < n$ . To define  $\Downarrow (\mathcal{M}[n], i) \Downarrow$ , we need to define  $i_\alpha \in \mathbf{M}[\mathcal{D}]$  for each  $\alpha \in \mathbf{Act}$  and set  $\Downarrow (\mathcal{M}[n], i) \Downarrow \stackrel{\text{def}}{=} (i_\alpha)_{\alpha \in \mathbf{Act}}$ . Consider the approximant  $(\mathcal{M}[n], i)$ . Define

$$\begin{aligned} F_\alpha &\stackrel{\text{def}}{=} \{s' \in \Sigma[n] \mid (i, \alpha, s') \in R[n]^a\} \\ G_\alpha &\stackrel{\text{def}}{=} \{s' \in \Sigma[n] \mid (i, \alpha, s') \in R[n]^c\}. \end{aligned} \quad (14)$$

For every  $s' \in F_\alpha \cup G_\alpha$ ,  $\Downarrow (\mathcal{M}[n-1], s') \Downarrow$  is already defined by induction. We set

$$\begin{aligned} i_\alpha^a &\stackrel{\text{def}}{=} \downarrow \{ \Downarrow (\mathcal{M}[n-1], l) \Downarrow \mid l \in F_\alpha \} \\ i_\alpha^c &\stackrel{\text{def}}{=} \uparrow \{ \Downarrow (\mathcal{M}[n-1], m) \Downarrow \mid m \in G_\alpha \} \end{aligned} \quad (15)$$

for each  $\alpha \in \mathbf{Act}$ . (Note that  $i_\alpha^a$  is empty if  $i$  has no  $R_\alpha^a$ -successors in  $(\mathcal{M}[n], i)$ . Similarly,  $i_\alpha^c$  is empty without any  $R_\alpha^c$ -successors in  $(\mathcal{M}[n], i)$ . Thus,  $i_\alpha = \emptyset$  if there are no transitions out of  $i$  in  $(\mathcal{M}[n], i)$ .) It suffices to show that  $i_\alpha^a \subseteq \downarrow (i_\alpha^a \cap i_\alpha^c)$  for each  $\alpha \in \mathbf{Act}$ . Given  $d \in i_\alpha^a$ , there is some  $l \in F_\alpha$  such that  $d \leq \Downarrow (\mathcal{M}[n-1], l) \Downarrow$ . But  $l \in F_\alpha$  means  $(i, \alpha, l) \in R^a$ . By condition (MC), there exists some  $m \in \Sigma$  such that  $(i, \alpha, m) \in R^a \cap R^c$  and  $(\mathcal{M}, m) \prec (\mathcal{M}, l)$ . Using induction, the latter gives us

$$(\mathcal{D}, \Downarrow (\mathcal{M}[n-1], m) \Downarrow) \prec (\mathcal{M}[n-1], m) \prec (\mathcal{M}[n-1], l) \prec (\mathcal{D}, \Downarrow (\mathcal{M}[n-1], l) \Downarrow) \quad (16)$$

which implies  $\Downarrow (\mathcal{M}[n-1], l) \Downarrow \leq \Downarrow (\mathcal{M}[n-1], m) \Downarrow$  by Theorem 5 and so, by transitivity,  $d \leq \Downarrow (\mathcal{M}[n-1], m) \Downarrow$  follows. From  $(i, \alpha, m) \in R^a \cap R^c$  we infer  $m \in F_\alpha \cap G_\alpha$  and therefore  $\Downarrow (\mathcal{M}[n-1], m) \Downarrow \in i_\alpha^a \cap i_\alpha^c$  shows the claim. Since  $i_\alpha^a$  and  $i_\alpha^c$  in (15) are order-generated by finitely many elements — which are compact by induction —, we infer that  $i_\alpha$  is compact in  $\mathbf{M}[\mathcal{D}]$  for each  $\alpha \in \mathbf{Act}$ . Therefore,  $\Downarrow (\mathcal{M}[n], i) \Downarrow$  is compact in  $\mathcal{D}$ . As for the refinement relations  $Q_1^i$  and  $Q_2^i$ , we define

$$\begin{aligned} Q_1^i &\stackrel{\text{def}}{=} \{(i, \Downarrow (\mathcal{M}[n], i) \Downarrow)\} \\ &\quad \cup \left( \bigcup_{x \in F_\alpha \cup G_\alpha} Q_1^x \right) \\ &\quad \cup \{(s, d) \in \Sigma[n] \times \mathcal{D} \mid d \leq \Downarrow (\mathcal{M}[n-1], s) \Downarrow, \Downarrow (\mathcal{M}[n-1], s) \Downarrow \in i_\alpha^c\} \\ Q_2^i &\stackrel{\text{def}}{=} \{(\Downarrow (\mathcal{M}[n-1], i) \Downarrow, i)\} \\ &\quad \cup \left( \bigcup_{x \in F_\alpha \cup G_\alpha} Q_2^x \right) \\ &\quad \cup \{(e, t) \in \mathcal{D} \times \Sigma[n] \mid \Downarrow (\mathcal{M}[n-1], t) \Downarrow \leq e, \Downarrow (\mathcal{M}[n-1], t) \Downarrow \in i_\alpha^a\}. \end{aligned}$$

□

**Definition 13 (General embedding).** Let  $(\mathcal{M}, i)$  be a pointed mixed transition system that meets condition (MC). By Proposition 5,  $\Downarrow (\mathcal{M}[n], i) \Downarrow \in \mathcal{D}$  is defined for all  $n \geq 0$ . By Proposition 4.2 and 5 and Theorem 5, these elements form an ascending chain in  $\mathcal{D}$  and therefore

$$\llbracket (\mathcal{M}, i) \rrbracket \stackrel{\text{def}}{=} \bigvee_{n \geq 0} \llbracket (\mathcal{M}[n], i) \rrbracket \quad (17)$$

exists.

The properties of this embedding allow us to prove important facts about mixed transition systems that meet condition (MC).

**Theorem 6 (Logical and domain-theoretic characterization of refinement).** Let  $(\mathcal{N}, j)$  and  $(\mathcal{M}, i)$  be mixed transition systems that satisfy condition (MC).

- 1  $(\mathcal{N}, j) \prec (\mathcal{M}, i)$  iff  $\llbracket (\mathcal{M}, i) \rrbracket \leq \llbracket (\mathcal{N}, j) \rrbracket$  in  $\mathcal{D}$ .
- 2  $(\mathcal{N}, j) \prec (\mathcal{M}, i)$  iff  $\{\phi \in \mathbb{L}_{\text{HM}} \mid (\mathcal{M}, i) \models^a \phi\} \subseteq \{\phi \in \mathbb{L}_{\text{HM}} \mid (\mathcal{N}, j) \models^a \phi\}$ .

*Proof of Theorem 6*

- 1 By Proposition 4, we have  $(\mathcal{N}, j) \prec (\mathcal{M}, i)$  iff for all  $n \geq 0$ ,  $(\mathcal{N}[n], j) \prec (\mathcal{M}[n], i)$  iff (by Proposition 5) for all  $n \geq 0$ ,  $(\mathcal{D}, \llbracket (\mathcal{N}[n], j) \rrbracket) \prec (\mathcal{D}, \llbracket (\mathcal{M}[n], i) \rrbracket)$  iff (by internal full abstraction) for all  $n \geq 0$ ,  $\llbracket (\mathcal{M}[n], i) \rrbracket \leq \llbracket (\mathcal{N}[n], j) \rrbracket$  in  $\mathcal{D}$ . Thus,  $(\mathcal{N}, j) \prec (\mathcal{M}, i)$  implies

$$\llbracket (\mathcal{M}, i) \rrbracket = \bigvee_{n \geq 0} \llbracket (\mathcal{M}[n], i) \rrbracket \leq \bigvee_{n \geq 0} \llbracket (\mathcal{N}[n], j) \rrbracket = \llbracket (\mathcal{N}, j) \rrbracket. \quad (18)$$

Conversely, let  $\llbracket (\mathcal{M}, i) \rrbracket \leq \llbracket (\mathcal{N}, j) \rrbracket$ . Proof by contradiction: if  $(\mathcal{N}, j) \not\prec (\mathcal{M}, i)$ , then Proposition 4.3 implies

$$(\mathcal{N}, j) \not\prec (\mathcal{M}[n], i) \quad (19)$$

for some  $n \geq 0$ . We claim that

$$\forall m \geq 0: (\mathcal{N}[m], j) \not\prec (\mathcal{M}[n], i). \quad (20)$$

Proof by contradiction: if there is some  $m \geq 0$  with  $(\mathcal{N}[m], j) \prec (\mathcal{M}[n], i)$ , then  $(\mathcal{N}, j) \prec (\mathcal{N}[m], j)$  holds by Proposition 4.1, for  $(\mathcal{N}, j)$ , and implies  $(\mathcal{N}, j) \prec (\mathcal{M}[n], i)$  since  $\prec$  is transitive — a contradiction to (19).

For every  $m \geq 0$ ,  $(\mathcal{N}[m], j) \prec (\mathcal{D}, \llbracket (\mathcal{N}[m], j) \rrbracket)$  and  $(\mathcal{D}, \llbracket (\mathcal{M}[n], i) \rrbracket) \prec (\mathcal{M}[n], i)$  follow from Proposition 5. But then (20) and the transitivity of  $\prec$  imply that the pointed mixed transition system  $(\mathcal{D}, \llbracket (\mathcal{N}[m], j) \rrbracket)$  does not refine  $(\mathcal{D}, \llbracket (\mathcal{M}[n], i) \rrbracket)$ . Since  $m \geq 0$  was arbitrary, Proposition 2 renders

$$\forall m \geq 0: \llbracket (\mathcal{M}[n], i) \rrbracket \not\leq \llbracket (\mathcal{N}[m], j) \rrbracket \quad (21)$$

which contradicts (18) since  $\llbracket (\mathcal{M}[n], i) \rrbracket$  is compact in  $\mathcal{D}$  by Proposition 5 and  $\llbracket (\mathcal{M}[n], i) \rrbracket \leq \llbracket (\mathcal{M}, i) \rrbracket \leq \llbracket (\mathcal{N}, j) \rrbracket = \bigvee_{m \geq 0} \llbracket (\mathcal{N}[m], j) \rrbracket$ , where the supremum is directed.

- 2 One implication follows from Theorem 1. Conversely, the relation  $(\mathcal{N}, j) \not\prec (\mathcal{M}, i)$  implies  $\llbracket (\mathcal{M}, i) \rrbracket \not\leq \llbracket (\mathcal{N}, j) \rrbracket$  in  $\mathcal{D}$  by the previous item. Since  $\mathcal{D}$  is algebraic, there exists some compact element  $k$  in  $\mathcal{D}$  such that  $k \leq \llbracket (\mathcal{M}, i) \rrbracket$  and  $k \not\leq \llbracket (\mathcal{N}, j) \rrbracket$ .

By Lemma 1, there exists some  $\phi_k \in \mathbb{L}_{\text{HM}}$  with  $\uparrow k = \llbracket \phi_k \rrbracket^a$ . Thus,  $\langle (\mathcal{M}, i) \rangle \models^a \phi_k$  and  $\langle (\mathcal{N}, j) \rangle \not\models^a \phi_k$  follow. By Proposition 5 and Theorem 1, we get  $(\mathcal{M}, i) \models^a \phi_k$  and  $(\mathcal{N}, j) \not\models^a \phi_k$ .

□

### 3.3. Complementary processes

The universal domain  $\mathcal{D}$  models processes whose reactive capabilities are either firmly guaranteed, possible, or firmly disallowed (i.e. impossible). One may wonder whether such processes have a *complement* whose reactive capabilities are the logical negations of those of the original process. Given a modal transition system  $\mathcal{M} = (\Sigma, R^a, R^c)$ , a complementary process is evidently defined by  $\bar{\mathcal{M}} = (\Sigma, \bar{R}^a, \bar{R}^c)$ , where

$$(s, \alpha, s') \in \bar{R}^m \text{ iff } (s, \alpha, s') \notin R^{-m} \quad (m \in \{a, c\}). \quad (22)$$

Specification (22) can be modeled in our universal domain  $\mathcal{D}$ . Since we can embed modal transition systems into  $\mathcal{D}$ , this follows readily from the fact that  $\bar{\mathcal{M}}$  is a modal transition system if  $\mathcal{M}$  is one. For  $\bar{R}^a$  equals  $(\Sigma \times \Sigma) \setminus R^c$  which is contained in  $(\Sigma \times \Sigma) \setminus R^a$ , since  $\mathcal{M}$  is a modal transition system. But  $(\Sigma \times \Sigma) \setminus R^a = \bar{R}^c$ .

**Remark 4 (Complementary process in  $\mathcal{D}$ ).** For every pointed modal transition system  $(\mathcal{M}, i)$ , the complementary process  $\langle (\bar{\mathcal{M}}, i) \rangle \in \mathcal{D}$  is well defined.

## 4. Expressiveness of modal transition systems

Domain equation (7) for refinement in partial systems chooses modal transition systems to represent partial, under-determined aspects of a system in its state transition capabilities. However, systems may also be under-determined in state observables — atomic propositions such as “the network cable is plugged in,” or “pointer  $x$  may point to location 1 in the heap.” Therefore, we formulate a notion of partial systems (Kripke modal transition systems), refinement, and property semantics that allows for under-determined aspects in state transitions and state observables and prove that Kripke modal transition systems can be translated into modal transition systems such that refinements and property semantics are preserved and reflected. In particular, the results obtained for our universal domain in (7) apply to Kripke modal transition systems as well. Since this translation is linear in the size of models and formulas, no real overhead is involved in this representational shift.

The ability of modal transition systems to faithfully represent partial systems, their operational refinement, and property semantics is not limited to Kripke MTSs. In this section, we also show that labeled transition systems with a divergence predicate — the extended transition systems in (BG99) — and partial Kripke structures (BG99), as well as their operational abstraction preorders and three-valued semantics of modal logic, have such faithful embeddings into the model checking framework for modal transition systems.



## 4.1. Kripke modal transition systems

A *doubly labeled transition system* (dNV95) with signature  $(\mathbf{Act}, \mathbf{AP})$  is comprised of a non-empty set of states  $\Sigma$ , a set  $\mathbf{Act}$  of action labels, a set  $\mathbf{AP}$  of (atomic) state propositions, a state transition relation  $R \subseteq \Sigma \times \mathbf{Act} \times \Sigma$ , and a labeling function  $L: \Sigma \rightarrow \mathcal{P}(\mathbf{AP})$ . (Throughout, we assume that, for every  $s \in \Sigma$  and  $\alpha \in \mathbf{Act}$ , the sets  $L(s)$  and  $\{s' \mid (s, \alpha, s') \in R\}$  are finite.) Such structures are expressive and flexible models since they allow for state ( $\mathbf{AP}$ ) and state transition ( $\mathbf{Act}$ ) observables. Kripke modal transition systems are partial versions of doubly labeled transition systems in the same manner in which modal transition systems are partial versions of labeled transition systems.

**Definition 14 (Kripke modal transition systems (HJS01)).** A *Kripke modal transition system* (Kripke MTS)  $\mathcal{K}$  with signature  $(\mathbf{Act}, \mathbf{AP})$  is a tuple  $(\Sigma, R^a, R^c, L^a, L^c)$  such that  $(\Sigma, R^a, L^a)$  and  $(\Sigma, R^c, L^c)$  form *doubly labeled transition systems* with the same signature;  $R^a \subseteq R^c$ ; and  $L^a(s) \subseteq L^c(s)$  for all  $s \in \Sigma$ .

Of course, one may define Kripke mixed transition systems and their version of the consistency condition (MC). However, in practical applications modelers will want to rely on a consistency condition that is enforced by the underlying specification language, and in a transparent manner: Kripke modal transition systems are such a specification language. Refinements of Kripke modal transition systems are generalizations of refinements of modal transition systems in that state proposition observables are preserved (for mode a) and reflected (for mode c).

**Definition 15 (Refinement of Kripke modal transition systems (HJS01)).** A *refinement* within a Kripke MTS  $\mathcal{K} = (\Sigma, R^a, R^c, L^a, L^c)$  with signature  $(\mathbf{Act}, \mathbf{AP})$  is a relation  $Q \subseteq \Sigma \times \Sigma$  such that  $(s, t) \in Q$  implies for all  $\alpha \in \mathbf{Act}$

- 1 for all  $(t, \alpha, t') \in R^a$ , there is some  $s' \in \Sigma$  with  $(s, \alpha, s') \in R^a$  and  $(s', t') \in Q$ ;
- 2 for all  $(s, \alpha, s') \in R^c$ , there is some  $t' \in \Sigma$  with  $(t, \alpha, t') \in R^c$  and  $(s', t') \in Q$ ;
- 3  $L^a(t) \subseteq L^a(s)$ ; and
- 4  $L^c(s) \subseteq L^c(t)$ .

We write  $s \prec_{\mathcal{M}} t$  or  $s \prec t$  if there is some refinement  $Q$  with  $(s, t) \in Q$ . In that case,  $s$  *refines (is abstracted by) t*.

**Remark 5 (Refinement for pointed models).** Let  $\mathcal{K}_1$  and  $\mathcal{K}_2$  be two Kripke MTSs with start states  $i_1$  and  $i_2$  (respectively). Since the set-theoretic sum of these two Kripke MTSs is a Kripke MTS with the sum of their respective signatures, we say that  $(\mathcal{K}_1, i_1)$  *refines (is abstracted by)  $(\mathcal{K}_2, i_2)$*  iff there is a refinement  $Q$  on their sum such that  $(i_1, i_2) \in Q$ .

The logic for Kripke MTSs,  $\mathbb{L}_{\mathcal{K}}$ , is the modal mu-calculus as in (1), except that one replaces the clause for  $\mathbf{tt}$  with a clause for atomic propositions ( $p \in \mathbf{AP}$ ). (We may re-express  $\mathbf{tt}$  as  $\neg(p \wedge \neg p)$  since Kripke MTSs are consistent.) The semantics of this logic over Kripke MTSs is as the one in Figure 5, expect that clause  $\mathbf{tt}$  is replaced with (23).

$$\llbracket p \rrbracket_\rho^m \stackrel{\text{def}}{=} \{s \in \Sigma \mid p \in L^m(s)\} \quad (23)$$

The additional capability of Kripke MTSs to express state observables may be encoded in state transition observables. We translate Kripke MTSs into modal transition systems over an extended signature and show that this translation preserves and reflects refinement and the property semantics. In particular, Kripke MTSs can be embedded into our universal domain (for the appropriately extended signature).

**Definition 16 (Translating Kripke MTSs).** Let  $\mathcal{K} = (\Sigma, R^a, R^c, L^a, L^c)$  be a Kripke MTS with signature  $(\text{Act}, \text{AP})$ . This determines a mixed transition system  $\mathbf{M}[\mathcal{K}]$  with signature  $\text{AP} + \text{Act}$ , state space  $\Sigma$ , and transition relations  $\bar{R}^m \subseteq \Sigma \times (\text{AP} + \text{Act}) \times \Sigma$ , where

$$\bar{R}^m \stackrel{\text{def}}{=} \{(s, \beta, s') \mid \beta \in L^m(s) \text{ and } s = s'; \text{ or } (s, \beta, s') \in R^m\} \quad (m \in \{a, c\}). \quad (24)$$

Note that the resulting mixed transition system is finitely branching. We define a translation from  $\mathbb{L}_{\mathcal{K}}$  to  $\mathbb{L}$ :

$$\begin{aligned} T(p) &\stackrel{\text{def}}{=} (\exists p) \neg(p \wedge \neg p) & T(Z) &\stackrel{\text{def}}{=} Z \\ T(\neg\phi) &\stackrel{\text{def}}{=} \neg T(\phi) & T(\phi_1 \wedge \phi_2) &\stackrel{\text{def}}{=} T(\phi_1) \wedge T(\phi_2) \\ T((\exists\alpha)\phi) &\stackrel{\text{def}}{=} (\exists\alpha) T(\phi) & T(\mu Z.\phi) &\stackrel{\text{def}}{=} \mu Z.T(\phi). \end{aligned}$$

The transformations of models ( $\mathcal{K} \mapsto \mathbf{M}[\mathcal{K}]$ ) and properties ( $\phi \mapsto T(\phi)$ ) preserve and reflect refinement, abstraction, and model checks.

**Theorem 7 (Soundness and completeness of translation).** Let  $\mathcal{K}$  be a Kripke MTS  $(\Sigma, R^a, R^c, L^a, L^c)$  with signature  $(\text{Act}, \text{AP})$ .

- 1  $\mathbf{M}[\mathcal{K}]$  is a modal transition system with signature  $\text{AP} + \text{Act}$ .
- 2 For  $s, t \in \Sigma$ , we have  $s \prec t$  in  $\mathcal{K}$  iff  $s \prec t$  in  $\mathbf{M}[\mathcal{K}]$ .
- 3 For all  $\phi \in \mathbb{L}_{\mathcal{K}}$ ,  $\rho$ , and  $m$ , we have  $\llbracket \phi \rrbracket_\rho^m = \llbracket T(\phi) \rrbracket_\rho^m$ .

*Proof of Theorem 7*

- 1 Let  $(s, \beta, s') \in \bar{R}^a$ . If  $\beta \in L^a(s)$ , then  $L^a(s) \subseteq L^c(s)$  implies  $s' = s$  and  $(s, \beta, s) \in \bar{R}^c$ . Otherwise,  $(s, \beta, s') \in R^a \subseteq R^c$ , so  $(s, \beta, s') \in \bar{R}^c$ .
- 2 Let  $s, t \in \Sigma$ .
  - (a) Let  $s \prec t$  in  $\mathcal{K}$ . We show  $s \prec t$  in  $\mathbf{M}[\mathcal{K}]$ :
    - i Let  $(t, \beta, t') \in \bar{R}^a$ . If  $(t, \beta, t') \in R^a$ , then  $s \prec t$  in  $\mathcal{K}$  implies the existence of some  $s' \in \Sigma$  with  $s' \prec t'$  in  $\mathcal{K}$  and  $(s, \beta, s') \in R^a$ , i.e.  $(s, \beta, s') \in \bar{R}^a$ . Otherwise,  $\beta \in L^a(t)$  and  $s' = s$ , so  $s \prec t$  in  $\mathcal{K}$  implies  $\beta \in L^a(s)$  and  $s' = s$ , i.e.  $(s, \beta, s) \in \bar{R}^a$ .
    - ii Let  $(s, \beta, s') \in \bar{R}^c$ . If  $(s, \beta, s') \in R^c$ , then  $s \prec t$  in  $\mathcal{K}$  implies the existence of some  $t' \in \Sigma$  with  $s' \prec t'$  and  $(t, \beta, t') \in R^c$ , i.e.  $(t, \beta, t') \in \bar{R}^c$ . Otherwise,  $\beta \in L^c(s)$  and  $t' = t$ , so  $s \prec t$  in  $\mathcal{K}$  implies  $\beta \in L^c(t)$  and  $t' = t$ , i.e.  $(t, \beta, t) \in \bar{R}^c$ .
  - (b) Let  $s \prec t$  in  $\mathbf{M}[\mathcal{K}]$ . We show  $s \prec t$  in  $\mathcal{K}$ :

- i Given  $(t, \alpha, t') \in R^a$ , we have  $(t, \alpha, t') \in \bar{R}^a$ , so  $s \prec t$  in  $\mathbf{M}[\mathcal{K}]$  implies the existence of some  $s' \in \Sigma$  with  $s' \prec t'$  in  $\mathbf{M}[\mathcal{K}]$  and  $(s, \alpha, s') \in \bar{R}^a$ , i.e.  $(s, \alpha, s') \in R^a$  since **AP** and **Act** are disjoint in **AP + Act**.
  - ii Given  $(s, \alpha, s') \in R^c$ , we have  $(s, \alpha, s') \in \bar{R}^c$ , so  $s \prec t$  in  $\mathbf{M}[\mathcal{K}]$  implies the existence of some  $t' \in \Sigma$  with  $s' \prec t'$  in  $\mathbf{M}[\mathcal{K}]$  and  $(t, \alpha, t') \in \bar{R}^c$ , i.e.  $(t, \alpha, t') \in R^c$  since **AP** and **Act** are disjoint in **AP + Act**.
  - iii Given  $p \in L^a(t)$ , we have  $(t, p, t) \in \bar{R}^a$ , so  $s \prec t$  in  $\mathbf{M}[\mathcal{K}]$  implies the existence of some  $s' \in \Sigma$  such that  $(s, p, s') \in \bar{R}^a$ . Since **AP** and **Act** are disjoint in **AP + Act**, we infer  $s' = s$  and  $p \in L^a(s)$ .
  - iv Given  $p \in L^c(s)$ , we have  $(s, p, s) \in \bar{R}^c$ , so  $s \prec t$  in  $\mathbf{M}[\mathcal{K}]$  implies the existence of some  $t' \in \Sigma$  such that  $(t, p, t') \in \bar{R}^c$ . Since **AP** and **Act** are disjoint in **AP + Act**, we have  $t' = t$  and  $p \in L^c(t)$ .
- 3 This is proved by the same induction as in the proof of Theorem 1:
- (a) For variables  $Z$ ,  $\llbracket \phi \rrbracket_\rho^m = \rho^m(Z) = \llbracket T(Z) \rrbracket_\rho^m$ .
  - (b) For  $p$ , we have  $\llbracket \phi \rrbracket_\rho^m = \{s \in \Sigma \mid p \in L^m(s)\} = \{s \in \Sigma \mid \exists s' \in \Sigma, (s, p, s') \in \bar{R}^m\} = \llbracket (\exists p) \neg(p \wedge \neg p) \rrbracket_\rho^m$  since **AP** and **Act** are disjoint in **AP + Act**.
  - (c) For  $\neg\phi$ ,  $\llbracket \neg\phi \rrbracket_\rho^m = \Sigma \setminus \llbracket \phi \rrbracket_\rho^m = \Sigma \setminus \llbracket T(\phi) \rrbracket_\rho^m = \llbracket \neg T(\phi) \rrbracket_\rho^m = \llbracket T(\neg\phi) \rrbracket_\rho^m$ .
  - (d) For  $\phi_1 \wedge \phi_2$ ,  $\llbracket \phi_1 \wedge \phi_2 \rrbracket_\rho^m = \llbracket \phi_1 \rrbracket_\rho^m \cap \llbracket \phi_2 \rrbracket_\rho^m = \llbracket T(\phi_1) \rrbracket_\rho^m \cap \llbracket T(\phi_2) \rrbracket_\rho^m$  which equals  $\llbracket T(\phi_1 \wedge \phi_2) \rrbracket_\rho^m$ .
  - (e) For  $(\exists\alpha)\phi$ ,  $\llbracket (\exists\alpha)\phi \rrbracket_\rho^m = \{s \in \Sigma \mid \exists s' (s, \alpha, s') \in R^m, s' \in \llbracket \phi \rrbracket_\rho^m\} = \{s \in \Sigma \mid \exists s' (s, \alpha, s') \in \bar{R}^m, s' \in \llbracket T(\phi) \rrbracket_\rho^m\} = \llbracket T((\exists\alpha)\phi) \rrbracket_\rho^m$  since **AP** and **Act** are disjoint in **AP + Act**.
  - (f) For  $\mu Z.\phi$ ,  $\llbracket T(\mu Z.\phi) \rrbracket_\rho^m$  is defined to be  $\llbracket \mu Z.T(\phi) \rrbracket_\rho^m$  which is the least fixed point of the function  $A \mapsto \llbracket T(\phi) \rrbracket_{\rho[Z \mapsto A]}^m$ . By induction, this function equals  $A \mapsto \llbracket \phi \rrbracket_{\rho[Z \mapsto A]}^m$  and so its least fixed point is  $\llbracket \mu Z.\phi \rrbracket_\rho^m$ .

□

#### 4.2. Shape analysis with Kripke modal transition systems

An important form of pointer analysis is *shape analysis* (CWZ90; GH96; JM79; SRW99; WR99), where the contents of heap storage are approximated by a graph whose nodes denote objects and whose arcs denote the values of the objects' fields. Local ("stack") variables that point into the heap are drawn as arcs pointing to the nodes.

Figure 8 displays the syntax of such *shape graphs*. The example in the Figure depicts an approximation to a singly linked list of length at least two: objects are circles; a double-circled object is a "summary node," meaning that it possibly represents more than one concrete object. Since the objects were constructed from a class/struct that owns a `next` field, objects have `next`-labeled arcs. For discussion, the objects are named  $u_0$  and  $u_1$ , and local variables  $x$  and  $y$  point to the objects. A solid arc denotes that a field definitely points to an object; a dotted arc means the field possibly points to it. Thus, the self-arc on  $u_1$  must be dotted because  $u_1$  possibly denotes multiple nodes, meaning that a `next` dereference possibly points to one of the concrete objects denoted by the node.

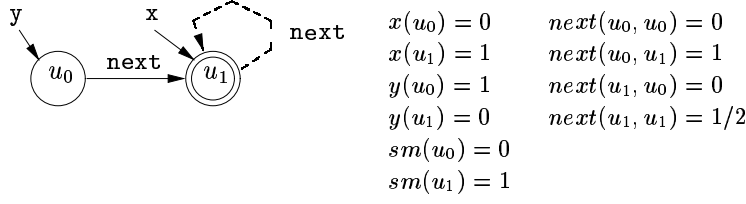


Fig. 8. A shape graph and its representation through predicates.

$$\begin{aligned}
T[x = y] &: x'(v) = y(v); \text{ all other predicates } p' = p \\
T[x.\text{next} = y] &: next'(v_1, v_2) = (next(v_1, v_2) \wedge (sm(v_1) \vee \neg x(v_1))) \vee (x(v_1) \wedge y(v_2)); \\
&\text{all other } p' = p \\
T[x = y.\text{next}] &: x'(v) = \exists v_1. y(v_1) \wedge next(v_1, v); \text{ all other } p' = p \\
T[x = \text{new Node}()] &: \text{let } v_{new} \text{ be a fresh node, in } x'(v) = (v = v_{new}); \\
&\text{all other } p'(v) = (p(v) \wedge (v \neq v_{new}))
\end{aligned}$$

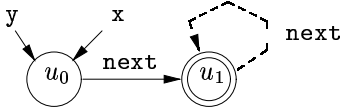
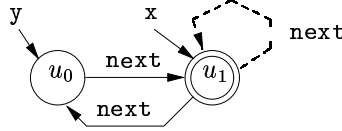
Effect of  $x = y$  on Figure 8:Effect of  $x.\text{next} = y$  on Figure 8:

Fig. 9. Transfer functions on shape graphs.

Shape graphs can be encoded in various ways; in Figure 8, we display a coding due to Sagiv, Reps, and Wilhelm (SRW99), who define local-variable points-to information with unary predicates and field points-to information with binary ones. The predicates produce the answers “guaranteed to point to” (1), “possibly points to” (1/2), and “not points to” (0), where the values are ordered  $0 \leq 1/2 \leq 1$ . Similarly, the predicate  $sm$  notes which nodes are summary nodes; those  $s$  for which  $sm(s) = 1$ .

Shape graphs can be used as data values for a data-flow analysis, where a program’s transfer functions transform an input shape graph to an output one. The transfer functions for assignment and object construction appear in Figure 9, where  $p'$  denotes predicate  $p$  updated by the transfer function  $T[C]$  for command  $C$ . The transfer functions are written as predicate-logic formulas and interpreted on top of Kleene’s strong semantics for propositional logic.

A shape graph is in fact a Kripke MTS  $(\Sigma, R^a, R^c, L^a, L^c)$ , where

- $\Sigma$  is the shape graph’s nodes;
- $\text{Act} = \{\text{next}\}$ ;
- $\text{AP}$  contains the symbol  $sm$  and all identifiers of the program’s pointer variables;
- $R^a$  contains the solid labeled arcs between nodes;
- $R^c \setminus R^a$  contains the dashed labeled arcs between nodes;
- $x \in L^a(s)$  when a solid arc shows that  $x$  points to object  $s$ ;
- $x \in L^c(s) \setminus L^a(s)$  when a dashed arc shows that  $x$  points to object  $s$ ;
- When  $s$  is a summary node, then  $sm \in L^a(s)$ .

Given a shape graph/Kripke MTS, we check the graph for correctness properties that

are expressible in the CTL-subset (BCKLMH90; Dam94) of the modal mu-calculus. In (SRW99), such properties are encoded in predicate logic augmented with a transitive closure operator.

Here are examples: The direction relationship (GH96), stating that an access path exists from the object named by  $x$  to an object named by  $y$ , is written  $D(x, y) \stackrel{\text{def}}{=} x \wedge \text{EF}_{\text{next}} y$  — an object  $s$  has atomic property  $x$  iff  $x$  points to  $s$ . Recall that  $\text{EF}_{\alpha} \phi$  states, “there exists a path of  $\alpha$ -labeled transitions such that, at some state in the future,  $\phi$  holds true.” To validate that there is a *guaranteed (possible)* path from  $s$ , we check whether  $s \models^a D(x, y)$  ( $s \models^c D(x, y)$ ); to refute existence of a path, we check  $s \models^a \neg D(x, y)$ .

The interference relationship (GH96), saying that pointers  $x$  and  $y$  have access paths to a common heap node, is written with *inverse* transition relationships of  $R^a$ :  $I(x, y) \stackrel{\text{def}}{=} (\text{EF}_{\text{next}-1} x) \wedge (\text{EF}_{\text{next}-1} y)$ . We check  $s \models^c I(x, y)$  to see if aliasing of object  $s$  by  $x$  and  $y$  is possible.

Aliasing of pointers can be expressed: for **aliasing**  $\stackrel{\text{def}}{=} \text{EF}_{\text{next}} (\bigvee_{x \neq y} x \wedge y)$ , the formulas (a)  $\text{AG}_{\text{next}} \neg \text{aliasing}$ , (b)  $\text{AG}_{\text{next}} \neg (x \wedge \bigvee_{x \neq y} y)$ , and (c)  $\text{AG}_{\text{next}} \neg (x \wedge y)$  can then be used to check: (a) the absence of any kind of aliasing; (b) that  $x$  has no alias; and that (c)  $x$  and  $y$  never point to the same heap node. (Recall that  $\text{AG}_{\alpha} \phi$  states, “for all  $\alpha$ -paths, it is globally true for all states along the path that  $\phi$  holds.”)

We can check for possibly cyclic data structures. The predicate **cyclic**  $\stackrel{\text{def}}{=} \bigvee_{x \in \text{AP}} x \wedge \text{EX}_{\text{next}} \text{EF}_{\text{next}} x$  states that a heap node is pointed to by some  $x$  that has an access path to, presumably the same, heap node pointed to by  $x$ . (Recall that  $\text{EX}_{\alpha} \phi$  says, “there exists an  $\alpha$ -transition to a next state where  $\phi$  holds.”)

A full exposition of shape analysis based on shape graphs is beyond the scope of this paper. However, we note that the modal transition systems for shape graphs may have total refinements that have no correspondence to shapes that may occur at run-time. For example, a variable cannot point to two distinct locations in the heap at the same time. Thus, one may need to use techniques for restricting the set of total refinements of a graph in order to conclude that properties are valid or consistent. One such technique is assume-guarantee reasoning for branching-time logics (KV98).

### 4.3. Extended transition systems

In Section 2, we already discussed how labeled transition systems give rise to partial systems whose under-determined aspects are represented explicitly. Modeling under-determinacy in systems through a pair of labeled transition systems, connected with a consistency constraint, is not the only way of enriching labeled transition systems with explicit under-determined aspects. Bruns and Godefroid define an extended transition system  $\mathcal{E}$  (BG99) with signature  $\text{Act}$  as a labeled transition system  $(\Sigma, R)$  with the same signature, endowed with a divergence predicate  $\uparrow \subseteq \Sigma \times \text{Act}$  (Mil81; Wal90). The intuitive meaning of  $s \uparrow \alpha$  is that “some of the  $\alpha$ -transitions from  $s$  in the full model may be missing at  $s$  in the ETS  $\mathcal{E}$ ” (BG99). As customary, we write  $s \downarrow \alpha$  when  $s \uparrow \alpha$  fails to hold, meaning that all  $\alpha$ -transitions from  $s$  in the full model (possibly none at all) are present in the ETS  $\mathcal{E}$ . Partial bisimulations (Mil81; Wal90) are the operational abstraction preorder for extended transition systems.

$$\begin{aligned}
\llbracket \mathbf{tt} \rrbracket^\downarrow s &\stackrel{\text{def}}{=} 1 \\
\llbracket \neg\phi \rrbracket^\downarrow s &\stackrel{\text{def}}{=} 1 - (\llbracket \phi \rrbracket^\downarrow s) \\
\llbracket \phi_1 \wedge \phi_2 \rrbracket^\downarrow s &\stackrel{\text{def}}{=} (\llbracket \phi_1 \rrbracket^\downarrow s) \wedge (\llbracket \phi_2 \rrbracket^\downarrow s) \\
\llbracket (\exists\alpha)\phi \rrbracket^\downarrow s &\stackrel{\text{def}}{=} \bigvee (\{1/2 \mid s \uparrow \alpha\} \cup \{\llbracket \phi \rrbracket^\downarrow s' \mid (s, \alpha, s') \in R\}).
\end{aligned}$$

Fig. 10. Property semantics for Hennessy-Milner logic  $\mathbb{L}_{\text{HM}}$  over extended transition systems (BG99), where  $s \in \Sigma$  and  $\wedge$  and  $\bigvee$  are defined for  $0 < 1/2 < 1$ . The set  $\{1/2 \mid s \uparrow \alpha\}$  is empty iff  $s \downarrow \alpha$ .

**Definition 17 (Partial bisimulation (Mil81)).** A *partial bisimulation* in an extended transition system  $\mathcal{E} = (\Sigma, R, \uparrow)$  is a relation  $Q \subseteq \Sigma \times \Sigma$  such that  $(t, s) \in Q$  implies for all  $\alpha \in \text{Act}$

- 1 if  $(t, \alpha, t') \in R$ , there exists some  $s' \in \Sigma$  such that  $(s, \alpha, s') \in R$  and  $(t', s') \in Q$ ;
- 2 if  $t \downarrow \alpha$ , then (i)  $s \downarrow \alpha$ , and (ii) whenever  $(s, \alpha, s') \in R$ , there exists  $t' \in \Sigma$  such that  $(t, \alpha, t') \in R$  and  $(t', s') \in Q$ .

One can form the sum of two extended transition systems by forming the sum of their underlying labeled transition systems, the sum of their respective divergence predicates, and the sum of their signatures. In this manner, Definition 17 also defines partial bisimulations between pointed extended transition systems. The intuitive readings of  $s \uparrow \alpha$  and  $s \downarrow \alpha$  suggest that extended transition systems can be represented as modal transition systems, and therefore embed into our universal domain. What is maybe more surprising is that partial bisimulations in an extended transition system turn out to be the relational inverses of refinements of the representing modal transition systems. Moreover, the three-valued semantics for Hennessy-Milner logic in (BG99) corresponds to the assertion checking semantics of the representing modal transition system.

**Definition 18 (Translating extended transition systems (HJS01)).** Let  $\mathcal{E}$  be an extended transition system  $(\Sigma, R, \uparrow)$  with signature  $\text{Act}$ . We define a modal transition system  $\mathbf{E}[\mathcal{E}] = (\Sigma, R, R^c)$  with the same signature  $\text{Act}$ , where

$$R^c \stackrel{\text{def}}{=} R \cup \{(s, \alpha, s') \in \Sigma \times \text{Act} \times \Sigma \mid s \uparrow \alpha\}. \quad (25)$$

Note that the state variable  $s'$  is free in (25), meaning that the modal transition system represents each instance of  $s \uparrow \alpha$  conservatively in that it adds  $R^c$ -transitions of type  $\alpha$  from  $s$  to *all states in*  $\Sigma$ . In (BG99), formulas of  $\mathbb{L}_{\text{HM}}$  are interpreted over extended transition systems with signature  $\text{Act}$ ; the semantics is given in Figure 10. We use the truth ordering  $0 < 1/2 < 1$  as a representation instead of the false  $< \perp < \text{true}$  of (BG99).

**Theorem 8 (Soundness and completeness of translation).** Let  $\mathcal{E} = (\Sigma, R, \uparrow)$  be an extended transition system with signature  $\text{Act}$ .

- 1 The structure  $\mathbf{E}[\mathcal{E}]$  is a modal transition system with signature  $\text{Act}$ .

- 2 For all  $\phi \in \mathbb{L}_{\text{HM}}$ , we have  $\llbracket \phi \rrbracket^\downarrow s = 1$  iff  $s \in \llbracket \phi \rrbracket^{\text{a}}$ ;  $\llbracket \phi \rrbracket^\downarrow s = 0$  iff  $s \notin \llbracket \phi \rrbracket^{\text{c}}$ . In particular,  $\llbracket \phi \rrbracket^\downarrow s = 1/2$  iff  $s \in \llbracket \phi \rrbracket^{\text{c}} \setminus \llbracket \phi \rrbracket^{\text{a}}$ .
- 3 The relational inverse of a reflexive partial bisimulation in  $\mathcal{E}$  is a refinement in the modal transition system  $\mathbf{E}[\mathcal{E}]$ . Conversely, if  $\mathbf{E}[\mathcal{E}]$  is such that

$$s \prec_{\mathcal{M}} t \text{ and } t \downarrow \alpha \Rightarrow s \downarrow \alpha \quad (26)$$

then the relational inverse of every refinement in  $\mathbf{E}[\mathcal{E}]$  is a partial bisimulation in  $\mathcal{E}$ . In that case, the relational inverse of the greatest partial bisimulation equals the greatest refinement in  $\mathbf{E}[\mathcal{E}]$ .

*Proof of Theorem 8*

- 1 Since  $R^{\text{a}}$  equals  $R$ , equation (25) enforces the consistency condition  $R^{\text{a}} \subseteq R^{\text{c}}$ .
- 2 By the previous item and Theorem 3.2, it suffices to show the claims about 0 and 1 which we prove by structural induction:
  - (a) We have  $\llbracket \text{tt} \rrbracket^\downarrow s = 1$  and  $s \in \llbracket \text{tt} \rrbracket^{\text{m}}$  for  $\text{m} \in \{\text{a}, \text{c}\}$ .
  - (b) For negation,  $x \stackrel{\text{def}}{=} \llbracket \neg\phi \rrbracket^\downarrow s = 1 - \llbracket \phi \rrbracket^\downarrow s$ . By induction,  $x$  equals 1 iff  $s \notin \llbracket \phi \rrbracket^{\text{c}}$  iff  $s \in \llbracket \neg\phi \rrbracket^{\text{a}}$ . By induction,  $x$  equals 0 iff  $s \in \llbracket \phi \rrbracket^{\text{a}}$  iff  $s \notin \llbracket \neg\phi \rrbracket^{\text{c}}$ .
  - (c) For conjunction,  $y \stackrel{\text{def}}{=} \llbracket \phi_1 \wedge \phi_2 \rrbracket^\downarrow s = (\llbracket \phi_1 \rrbracket^\downarrow s) \wedge (\llbracket \phi_2 \rrbracket^\downarrow s)$ . By induction,  $y$  equals 1 iff  $s \in \llbracket \phi_i \rrbracket^{\text{a}}$  for  $i = 1, 2$  iff  $s \in \llbracket \phi_1 \wedge \phi_2 \rrbracket^{\text{a}}$ . By induction,  $y$  equals 0 iff  $s \notin \llbracket \phi_i \rrbracket^{\text{c}}$  for some  $i = 1, 2$  iff  $s \notin \llbracket \phi_1 \wedge \phi_2 \rrbracket^{\text{c}}$ .
  - (d) For the modalities,  $z \stackrel{\text{def}}{=} \llbracket (\exists\alpha)\phi \rrbracket^\downarrow s = \bigvee(\{1/2 \mid s \uparrow \alpha\} \cup \{\llbracket \phi \rrbracket^\downarrow s' \mid (s, \alpha, s') \in R\})$ . By induction,  $z$  equals 1 iff there is some  $s'$  with  $(s, \alpha, s') \in R$  and  $s' \in \llbracket \phi \rrbracket^{\text{a}}$  iff  $s \in \llbracket (\exists\alpha)\phi \rrbracket^{\text{a}}$ . By induction,  $z$  equals 0 iff  $s \downarrow \alpha$  and  $s' \notin \llbracket \phi \rrbracket^{\text{c}}$  for all  $s'$  with  $(s, \alpha, s') \in R$  iff  $s \notin \llbracket (\exists\alpha)\phi \rrbracket^{\text{c}}$  by (25).
- 3 (a) Let  $\sqsubseteq$  be a reflexive, partial bisimulation in  $\mathcal{E}$ . We show that  $Q$ , the relational inverse of  $\sqsubseteq$ , is a refinement in  $\mathbf{E}[\mathcal{E}]$ . Let  $(s, t) \in Q$ , i.e.  $t \sqsubseteq s$ .
  - i If  $(t, \alpha, t') \in R^{\text{a}}$ , i.e.  $(t, \alpha, t') \in R$ , then  $t \sqsubseteq s$  implies the existence of some  $s' \in \Sigma$  such that  $(s, \alpha, s') \in R = R^{\text{a}}$  and  $t' \sqsubseteq s'$ , i.e.  $(s', t') \in Q$ .
  - ii If  $(s, \alpha, s') \in R^{\text{c}}$ , there are two cases to consider:
    - A If  $t \uparrow \alpha$ , then  $(t, \alpha, s') \in R^{\text{c}}$  by (25). But  $s' \sqsubseteq s'$ , i.e.  $(s', s') \in Q$ , as  $\sqsubseteq$  is reflexive.
    - B If  $t \downarrow \alpha$ , then  $t \sqsubseteq s$  implies  $s \downarrow \alpha$  which, in turn, implies  $(s, \alpha, s') \in R$  by (25) since  $(s, \alpha, s') \in R^{\text{c}}$ . But then  $t \sqsubseteq s$  implies the existence of some  $t' \in \Sigma$  such that  $(t, \alpha, t') \in R \subseteq R^{\text{c}}$  and  $t' \sqsubseteq s'$ , i.e.  $(s', t') \in Q$ .
- (b) Let  $Q$  be a refinement in  $\mathbf{E}[\mathcal{E}]$  and  $(t, s) \in Q^{-1}$ , i.e.  $(s, t) \in Q$ .
  - i If  $(t, \alpha, t') \in R$ , then  $(t, \alpha, t') \in R^{\text{a}}$  and  $(s, t) \in Q$  imply the existence of some  $s' \in \Sigma$  such that  $(s, \alpha, s') \in R^{\text{a}} = R$  and  $(s', t') \in Q$ , i.e.  $(t', s') \in Q^{-1}$ .
  - ii If  $t \downarrow \alpha$ , then
    - A If (26) holds, then  $s \downarrow \alpha$  as  $(s, t) \in Q$  which is contained in  $\prec$  as a refinement.

- B If  $(s, \alpha, s') \in R \subseteq R^c$ , then  $(s, t) \in Q$  implies the existence of some  $t' \in \Sigma$  such that  $(t, \alpha, t') \in R^c$  and  $(s', t') \in Q$ . But then  $(t, \alpha, t') \in R^c$  and  $t \downarrow \alpha$  imply  $(t, \alpha, t') \in R$ .

□

This result not only states that extended transition systems and their partial bisimulation can be seen as modal transition systems with their abstraction order. Since the latter models can be embedded into our universal domain, the former models are themselves embeddable into the same domain by the composition of these transformations. Inspecting the work in (Abr91), this suggests that there is an embedding of Abramsky’s universal domain (Abr91), which is based on an extended Plotkin powerdomain, into our universal domain  $\mathcal{D}$ . Since the divergence predicate in (Abr91) is state-wide, this is false. However, an embedding can be given for the action-dependent divergence predicate of this section by modifying the domain in (Abr91) to a recursive solution of products of lifted Plotkin powerdomains. This embedding is based on the embedding of the Plotkin powerdomain  $\mathcal{C}_D$  of a domain  $D$  into the mixed powerdomain  $\mathbf{M}[D]$ : every compact, convex set  $C$ , even the empty set used in (Abr91), is mapped to the pair  $(L, U)$ , where  $L$  and  $U$  are the lower and upper closure of  $C$  (respectively) (Hec90).

#### 4.4. Partial Kripke structures

Bruns and Godefroid (BG99) also devise partial Kripke structures as under-determined models for partial-state-space model checking. In loc. cit. they specify an abstraction preorder between such models, give a three-valued semantics over such models for the branching-time temporal logic CTL (CE81), and present a model-checking algorithm for that semantics (BG99). Since partial Kripke structures are special Kripke MTSs, we may use the translation of Section 4.1 to represent these models as modal transition systems. This translation preserves and reflects the abstraction preorder and the three-valued semantics of propositional modal logic.

#### Definition 19 (Partial Kripke structures (BG99)).

- 1 Let  $\mathbf{K}$  be the *partial information order*  $\{0, 1/2, 1\}$  with  $1/2 \sqsubseteq 0$  and  $1/2 \sqsubseteq 1$  — an isomorphic copy of  $\mathbf{M}[*]$ .
- 2 A *partial Kripke structure*  $\mathcal{P}$  (BG99) with signature  $\mathbf{AP}$  is a triple  $(\Sigma, R, L)$ , where  $\Sigma$  is a set of states,  $R \subseteq \Sigma \times \Sigma$  a state transition relation, and  $L: \Sigma \times \mathbf{AP} \rightarrow \mathbf{K}$  is a labeling function.
- 3 A *completeness order* (BG99) in a partial Kripke structure  $\mathcal{P}$  with signature  $\mathbf{AP}$  is a relation  $Q \subseteq \Sigma \times \Sigma$  such that  $(s, t) \in Q$  implies
  - (a) for all  $p \in \mathbf{AP}$ ,  $L(s, p) \sqsubseteq L(t, p)$  in the information order of  $\mathbf{K}$ ;
  - (b) if  $(s, s') \in R$ , then there exists some  $t' \in \Sigma$  with  $(t, t') \in R$  and  $(s', t') \in Q$ ;
  - (c) if  $(t, t') \in R$ , then there exists some  $s' \in \Sigma$  with  $(s, s') \in R$  and  $(s', t') \in Q$ .

Intuitively,  $L(s, p) = 1/2$  expresses that “ $p$  is true at state  $s$ ” is a consistent statement; whereas  $L(s, p) = 1$  ( $L(s, p) = 0$ ) expresses that “ $p$  is true at state  $s$ ” (“ $p$  is false



$$\begin{aligned}
\llbracket p \rrbracket^{\mathbf{K}} s &\stackrel{\text{def}}{=} L(s, p) \\
\llbracket \neg\phi \rrbracket^{\mathbf{K}} s &\stackrel{\text{def}}{=} 1 - (\llbracket \phi \rrbracket^{\mathbf{K}} s) \\
\llbracket \phi_1 \wedge \phi_2 \rrbracket^{\mathbf{K}} s &\stackrel{\text{def}}{=} (\llbracket \phi_1 \rrbracket^{\mathbf{K}} s) \wedge (\llbracket \phi_2 \rrbracket^{\mathbf{K}} s) \\
\llbracket \diamond\phi \rrbracket^{\mathbf{K}} s &\stackrel{\text{def}}{=} \bigvee \{ \llbracket \phi \rrbracket^{\mathbf{K}} s' \mid (s, s') \in R \}.
\end{aligned}$$

Fig. 11. Property semantics for propositional modal logic over partial Kripke structures (BG99), where  $s \in \Sigma$  and  $\wedge$  and  $\bigvee$  are defined for  $0 < 1/2 < 1$  — the *truth ordering* of  $\mathbf{K}$ .

at state  $s$ ) is a valid assertion (respectively). For a completeness order  $Q$ ,  $(s, t) \in Q$  implies that valid assertions for  $s$  are also valid for  $t$ , and consistent statements for  $s$  are consistent for  $t$  as well; this correspondence is preserved in a co-inductive manner, familiar from bisimulations (Par89; Mil89). In (BG99), a three-valued semantics for propositional modal logic is given over partial Kripke structures; see Figure 11.

**Lemma 2 (Correspondence to Kripke MTSs (HJS01)).** Partial Kripke structures  $\mathcal{P} = (\Sigma, R, L)$  with signature  $\text{AP}$  are in one-to-one correspondence to Kripke MTSs  $\mathcal{K} = (\Sigma, R^a, R^c)$  with signature  $(\{*\}, \text{AP})$  such that  $R^a = R^c = \{(s, *, s') \mid (s, s') \in R\}$ ,  $L^a(s) = \{p \in \text{AP} \mid L(s, p) = 1\}$ , and  $L^c(s) = \{p \in \text{AP} \mid L(s, p) \neq 0\}$ .

*Proof of Lemma 2* Relations of type  $\Sigma \times \Sigma$  are in one-to-one correspondence to relations of type  $\Sigma \times \{*\} \times \Sigma$ . As for the labeling functions,  $L^a(s) \subseteq L^c(s)$  follows since  $0 \neq 1$ . Conversely, any pair  $(L^a, L^c)$  with  $L^a(s) \subseteq L^c(s)$  for all  $s \in \Sigma$  determines a function  $L: \Sigma \times \text{Act} \rightarrow \mathbf{K}$  such that  $L(s, p) = 1$  iff  $p \in L^a(s)$ ; and  $L(s, p) = 0$  iff  $p \notin L^c(s)$ . These transformations are clearly inverses of each other.  $\square$

**Definition 20 (Translating partial Kripke structures).** Let  $\mathcal{P} = (\Sigma, R, L)$  be a partial Kripke structure with signature  $\text{AP}$  and let  $\mathcal{K}$  be its corresponding Kripke MTSs as in Lemma 2. We then define  $\mathbf{P}[\mathcal{P}] \stackrel{\text{def}}{=} \mathbf{M}[\mathcal{K}]$ . Given a formula  $\phi$  of propositional modal logic, let  $K(\phi)$  be the formula obtained by replacing each occurrence of  $\diamond$  in  $\phi$  with  $(\exists*)$ .

**Theorem 9 (Soundness and completeness of translation).** Let  $\mathcal{P} = (\Sigma, R, L)$  be a partial Kripke structure with signature  $\text{AP}$ .

- 1 The modal transition system  $\mathbf{P}[\mathcal{P}]$  has signature  $\text{AP} + \{*\}$ .
- 2 The relational inverse of the greatest completeness order in  $\mathcal{P}$ , which is the union of all completeness orders in  $\mathcal{P}$  (BG99), is the greatest refinement in  $\mathbf{P}[\mathcal{P}]$ .
- 3 For all  $\phi$  of propositional modal logic,  $\llbracket \phi \rrbracket^{\mathbf{K}} s = 1$  iff  $s \in \llbracket T(K(\phi)) \rrbracket^a$ ;  $\llbracket \phi \rrbracket^{\mathbf{K}} s = 0$  iff  $s \notin \llbracket T(K(\phi)) \rrbracket^c$ .

*Proof of Theorem 9*

- 1 This is an immediate consequence of Theorem 7.
- 2 By Theorem 7, it suffices to show that statement for the corresponding Kripke MTS  $\mathcal{K}$  instead of for  $\mathbf{P}[\mathcal{P}]$ . Inspecting Definition 15 and the third part of Definition 19, this is now clear.

- 3 By Theorem 7, it suffices to show that statement for  $\llbracket K(\phi) \rrbracket^m$  over  $\mathcal{K}$  instead of  $\llbracket T(K(\phi)) \rrbracket^m$  over  $\mathbf{P}[\mathcal{P}]$ , which we prove by structural induction:
- (a) We have  $\llbracket p \rrbracket^{\mathbf{K}} s = 1$  iff  $L(s, p) = 1$  iff  $p \in L^a(s)$  iff  $s \in \llbracket K(p) \rrbracket^a$ ; dually,  $\llbracket p \rrbracket^{\mathbf{K}} s = 0$  iff  $L(s, p) = 0$  iff  $p \notin L^c(s)$  iff  $s \notin \llbracket K(p) \rrbracket^c$ .
  - (b) For negation,  $x \stackrel{\text{def}}{=} \llbracket \neg\phi \rrbracket^{\mathbf{K}} s = 1 - \llbracket \phi \rrbracket^{\mathbf{K}} s$ . By induction,  $x$  equals 1 iff  $s \notin \llbracket K(\phi) \rrbracket^c$  iff  $s \in \llbracket K(\neg\phi) \rrbracket^a$ . By induction,  $x$  equals 0 iff  $s \in \llbracket K(\phi) \rrbracket^a$  iff  $s \notin \llbracket K(\neg\phi) \rrbracket^c$ .
  - (c) For conjunction,  $y \stackrel{\text{def}}{=} \llbracket \phi_1 \wedge \phi_2 \rrbracket^{\mathbf{K}} s = (\llbracket \phi_1 \rrbracket^{\mathbf{K}} s) \wedge (\llbracket \phi_2 \rrbracket^{\mathbf{K}} s)$ . By induction,  $y$  equals 1 iff  $s \in \llbracket K(\phi_i) \rrbracket^a$  for  $i = 1, 2$  iff  $s \in \llbracket K(\phi_1 \wedge \phi_2) \rrbracket^a$ . By induction,  $y$  equals 0 iff  $s \notin \llbracket K(\phi_i) \rrbracket^c$  for some  $i = 1, 2$  iff  $s \notin \llbracket K(\phi_1 \wedge \phi_2) \rrbracket^c$ .
  - (d) For modalities,  $z \stackrel{\text{def}}{=} \llbracket \diamond\phi \rrbracket^{\mathbf{K}} s = \bigvee \{ \llbracket \phi \rrbracket^{\mathbf{K}} s' \mid (s, s') \in R \}$ . By induction,  $z$  equals 1 iff there is some  $s'$  with  $(s, s') \in R$  (i.e.  $(s, *, s) \in R^a$ ) and  $s' \in \llbracket K(\phi) \rrbracket^a$  iff  $s \in \llbracket K(\diamond\phi) \rrbracket^a$ . By induction,  $z$  equals 0 iff  $s' \notin \llbracket K(\phi) \rrbracket^c$  for all  $s'$  with  $(s, *, s') \in R$  iff  $s \notin \llbracket K(\diamond\phi) \rrbracket^c$ .

□

## 5. Related work

*Models and abstraction.* Modal transition systems were introduced by Larsen and Thomsen in (LT88). A logical characterization of refinement can be found in (Lar89). The models developed in Dams' thesis (Dam96) and in (DGG97) correspond to the “mixed” Kripke MTSs of Section 4.1. (We presented a more special class of mixed transition systems, informed by our choice of domain equation in Section 3.) Partial Kripke structures (Mor89) were studied by Bruns and Godefroid in (BG99). They showed that their three-valued property semantics can be computed by conventional model checks over two Kripke structures (BG00) — this is also possible for modal transition systems (GHJ01) and Kripke modal transition systems (Hut02a). In (BG00) generalized model checking specifies a more precise semantics for such models and reduces such property verification to the non-emptiness problem of alternating Büchi word automata over a one-letter alphabet. The account of extended transition systems and partial bisimulations (Mil81; Wal90) was based on (BG99). In (Sch01), it is shown how a concrete and naive trace-set semantics is transformed, by stepwise abstract interpretation (CC77), into a modal transition system that is then subject to property checks for branching-time logics. This transformation of models makes use of the existential and universal abstractions presented by Cousot and Cousot in (CC00). In (Hut99; Hut01; Hut02b), the modalities of modal transition systems are generalized to a wider class of models and sound abstractions are developed. The paper (HJS01) is the original rendition of portions of Sections 2 and 4. In (GHJ01), a calculus for the computation and representation of incremental abstractions is presented for modal transition systems. Loose specifications are also considered for variations of first-order logic; we mention the semantics of Alloy (JSS00; JSS01), the use of the Smyth powerdomain in (HP01), and the Kleene semantics of an extended first-order logic in (SRW99) used for shape analysis.

*Domains and logic.* In (Abr91), Abramsky studies a domain of synchronization trees and describes its logical counterpart, using Stone duality. This logic serves as a “rational reconstruction” (Abr91) of Hennessy-Milner logic. In this domain, a fully abstract semantics for terms of the process algebra SCCS is given. The mixed powerdomain was discovered independently by Heckmann (Hec90) and Gunter (Gun92). The former contains a concise axiomatization of the mix algebras. Three-valued logic historically emphasized the development of proof theory; see e.g. (Seg67; Mor89). The three-valued interpretation of set-theory used in this paper is Kleene’s strong interpretation of propositional logic (Kle52). It appears that the mixed powerdomain generalizes this semantics to non-flat data settings.

### Acknowledgments

We gratefully thank the anonymous referees for their detailed corrections and most helpful suggestions on improving the presentation of this material. Part of this research was carried out with the generous support of the U.S. National Science Foundation under grants CCR-99010171, 9970679, 02030716, ITR-0085949, 0086154, and INT-9981558. We would like to dedicate this paper to Dana Scott on the occasion of his 70th birthday.

### References

- S. Abramsky. A domain equation for bisimulation. *Information and Computation*, 92(2):161–218, June 1991.
- 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.
- T. Ball, A. Podelski, and S. K. Rajamani. Boolean and Cartesian Abstraction for Model Checking C Programs. In T. Margaria and W. Yi, editors, *Proceedings of TACAS’2001*, volume 2031 of *LNCS*, pages 268–283, Genova, Italy, April 2001. Springer Verlag.
- J. C. Bradfield. *Verifying Temporal Properties of Systems*. Birkhäuser, Boston, Mass., 1991.
- 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.
- 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.
- J. R. Burch, E. M. Clarke, D. L. Dill, K. L. McMillan, and J. Hwang. Symbolic model checking:  $10^{20}$  states and beyond. *Proceedings of the Fifth Annual Symposium on Logic in Computer Science*, June 1990.
- D. Chase, M. Wegman, and F. Zadeck. Analysis of pointers and structures. In *SIGPLAN Conf. on Prog. Lang. Design and Implementation*, pages 296–310. ACM Press, 1990.
- E. M. Clarke and E. A. Emerson. Synthesis of synchronization skeletons for branching time temporal logic. In D. Kozen, editor, *Logic of Programs Workshop*, number 131 in *LNCS*, pages 52–71. Springer Verlag, 1981.

- 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.
- 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.
- M. Dam. CTL\* and ECTL\* as Fragments of the Modal  $\mu$ -Calculus. *Theoretical Computer Science*, 126:77–96, 1994.
- D. Dams. *Abstract interpretation and partition refinement for model checking*. PhD thesis, Technische Universiteit Eindhoven, The Netherlands, 1996.
- D. Dams, R. Gerth, and O. Grumberg. Abstract interpretation of reactive systems. *ACM Transactions on Programming Languages and Systems*, 19(2):253–291, 1997.
- R. de Nicola and F. Vaandrager. Three Logics for Branching Bisimulation. *Journal of the Association of Computing Machinery*, 42(2):458–487, March 1995.
- R. Ghiya and L. J. Hendren. Is it a Tree, a DAG, or a Cyclic Graph? A Shape Analysis for Heap-Directed Pointers in C. In *Proceedings of the 23rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 1–15, 1996.
- 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*, volume 2154 of *Lecture Notes in Computer Science*, pages 426–440. Springer Verlag, August 2001.
- C. Gunter. The mixed power domain. *Theoretical Computer Science*, 103:311–334, 1992.
- R. Heckmann. Set Domains. In Neil D. Jones, editor, *European Symposium on Programming*, volume 432 of *Lecture Notes in Computer Science*, pages 177–196, Springer Verlag, May 1990.
- M. C. B. Hennessy and Robin Milner. Algebraic laws for non-determinism and concurrency. *JACM*, 32:137–161, 1985.
- C. A. R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.
- K. H. Hofmann and M. Mislove. Local compactness and continuous lattices. In B. Banaschewski and R.-E. Hoffmann, editors, *Continuous Lattices*, volume 871 of *Lecture Notes in Mathematics*, pages 209–248, Bremen, Germany, 1981. Springer Verlag.
- G. Holzmann. The model checker SPIN. *IEEE Transactions on Software Engineering*, 23:279–295, 1997.
- M. Huth. A Unifying Framework for Model Checking Labeled Kripke Structures, Modal Transition Systems, and Interval Transition Systems. In *Proceedings of the 19th International Conference on the Foundations of Software Technology & Theoretical Computer Science*, volume 1738 of *Lecture Notes in Computer Science*, pages 369–380, IIT Chennai, India, December 1999. Springer Verlag.
- M. Huth. *Domains and Processes*, chapter *Domains of view: a foundation for specification and analysis*, pages 183–218. Kluwer Academic Press, December 2001.
- M. Huth. Model checking modal transition systems using Kripke structures. In *Third International Workshop on Verification, Model Checking and Abstract Interpretation*, volume 2294 of *Lecture Notes in Computer Science*, pages 302–316, Venice, Italy, January 21–22 2002. Springer Verlag.
- M. Huth. Possibilistic and Probabilistic Abstraction-Based Model Checking. In H. Hermanns and R. Segala, editors, *Process Algebra and Probabilistic Methods, Performance Modeling and Verification, Second Joint International Workshop PAPM-PROBMIV 2002*, volume 2399 of *Lecture Notes in Computer Science*, pages 115–134, Copenhagen, Denmark, July 25–26 2002. Springer Verlag.

- M. Huth and S. Pradhan. Model-Checking View-Based Partial Specifications. In S. Brookes and M. Mislove, editors, *Electronic Notes in Theoretical Computer Science*, volume 45. Elsevier Science Publishers, 2001.
- M. Huth and S. Pradhan. Lifting assertion and consistency checkers from single to multiple viewpoints. Technical Report TR 2002/11, Imperial College London, Department of Computing, London, UK, March 2002.
- M. Huth, R. Jagadeesan, and D. Schmidt. Modal transition systems: a foundation for three-valued program analysis. In Sands D., editor, *Proceedings of the European Symposium on Programming (ESOP'2001)*, pages 155–169. Springer Verlag, April 2001.
- D. Jackson, I. Schechter, and I. Shlyakhter. Alcoa: the alloy constraint analyzer. In *Proc. International Conference on Software Engineering*, Limerick, Ireland, 2000.
- D. Jackson, I. Shlyakhter, and M. Sridharan. A Micromodularity Mechanism. In *Proceedings of the ACM SIGSOFT Conference on the Foundations of Software Engineering/European Software Engineering Conference (FSE/ESEC'01)*, September 2001.
- N.D. Jones and S. Muchnick. Flow analysis and optimization of LISP-like structures. In *Proc. 6th ACM Symp. Principles of Programming Languages*, pages 244–256, 1979.
- A. Jung. *Cartesian Closed Categories of Domains*. PhD thesis, Fachbereich Mathematik, Technische Hochschule Darmstadt, July 1988.
- P. Kelb. Model checking and abstraction: a framework preserving both truth and failure information. Technical Report OFFIS, University of Oldenburg, Germany, 1994.
- S. C. Kleene. *Introduction to Metamathematics*. Van Nostrand, 1952.
- D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.
- O. Kupferman and M.Y. Vardi. Modular model checking. In *Proc. Compositionality Workshop*, volume 1536 of *Lecture Notes in Computer Science*, pages 381–401. Springer-Verlag, 1998.
- 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.
- K. Larsen. Proof systems for satisfiability in Hennessy-Milner logic with recursion. *Theoretical Computer Science*, 72:265–288, 1990.
- 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.
- R. Milner. A modal characterisation of observable machine behaviours. In G. Astesiano and C. Böhm, editors, *CAAP '81*, volume 112 of *Lecture Notes in Computer Science*, pages 25–34. Springer Verlag, 1981.
- R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.
- O. Morikawa. Some modal logics based on a three-valued logic. *Notre Dame J. of Formal Logic*, 30:130–137, 1989.
- D. M. R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *In Proc. of the 5th GI Conference*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer Verlag, 1989.
- 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.
- G. D. Plotkin. A powerdomain construction. *SIAM Journal on Computing*, 5:452–487, 1976.
- G. D. Plotkin. A Structural Approach to Operational Semantics. Technical Report FN-19, DAIMI, Computer Science Department, Aarhus University, Ny Munkegade, Building 540, DK-8000 Aarhus, Denmark, September 1981. Reprinted April 1991.

- 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.
- D. A. Schmidt. From Trace Sets to Modal-Transition Systems by Stepwise Abstract Interpretation. Submitted for publication, 27 July 2001.
- K. Segerberg. Some modal logics based on a three-valued logic. *Theoria*, 33:53–71, 1967.
- M. B. Smyth. Powerdomains. *Journal of Computer and Systems Sciences*, 16:23–36, 1978.
- D. J. Walker. Bisimulation and divergence. *Information and Computation*, 85(2):202–241, 1990.
- J. Whaley and M. Rinard. Compositional pointer and escape analysis for Java programs. In *Proc. OOPSLA '99*, pages 187–206. ACM, 1999.