DOI 10.1007/s00165-005-0063-z
BCS © 2005
Formal Aspects of Computing (2005) 17: 113–137

Formal Aspects of Computing

Refinement is complete for implementations

Michael Huth

Department of Computing, Imperial College London, London, SW7 2AZ, UK

Abstract. Modal transition systems specify sets of implementations, their refining labelled transition systems, through Larsen & Thomsen's co-inductive notion of refinement. We demonstrate that refinement precisely captures the identification of a modal transition system with its set of implementations: refinement is reverse containment of sets of implementations. This result extends to models that combine state and event observables and is drawn from a *SFP*-domain whose elements are equivalence classes of modal transition systems under refinement [HJS04], and abstraction-based finite-model properties proved in this paper. As a corollary, validity checking is model checking for Hennessy-Milner formulas that characterize modal transition systems with bounded computation paths. We finally sketch how techniques developed in this paper can be used to detect inconsistencies between multiple modal transition systems and, if consistent, to verify properties of all common implementations.

Keywords: Model checking; Refinement; Modal transition systems; Implementation relation

1. Introduction

Formal modelling of computing systems and analysis of such models have always been key techniques of the formal-methods communities. We mention requirement elicitation, program analysis, many software-validation techniques, and protocol verification as beneficiaries of modelling and analysis of models. One such technique, model-checking [CE81, QS81], has been extremely successful as a formal method. In model-checking, a model M captures relevant aspects of a computer artifact, a formula ϕ denotes static or dynamic goals that the artifact should meet, and $M \models \phi$ holds iff goal ϕ is met by model M. Using conjunction we may assume in this discussion that only one goal ϕ needs to be checked. Since M is an abstraction of some concrete artifact C, i.e. C is a refinement of M, we desire that goals verified on the abstraction hold in the concrete artifact: for all ϕ, $M \models \phi$ should imply $C \models \phi$. If the judgment $M \models \phi$ is undecidable or has too high complexity, we want to find an abstraction A of M such that $A \models \phi$ implies $M \models \phi$ for all ϕ. In either case we are concerned with defining an efficiently decidable notion of refinement, whose relational inverse is abstraction, such that certifications of goals are preserved under refinement. Once such a framework for abstraction-based model checking is in place, we may also use it to check loose specifications M that model a potentially infinite set of concrete artifacts C.

The vast majority of model-checking frameworks utilize labelled transition systems as models, perhaps allowing for state-based or quantitative variations thereof. Properties ϕ are then typically expressed in a temporal logic such as computation tree logic [CE81, QS81] or linear-time temporal logic [GPVW95]. The tremendous success of this approach to formal verification is tarnished by two related short-comings:

Correspondence and offprint requests to: Michael Huth, Department of Computing, Imperial College London, London, SW7 2AZ, UK. Email: M.Huth@doc.imperial.ac.uk

1. models may be *specifications* and the *under-specification* of certain aspects may not be expressible; or
2. models may be abstractions of models or artifacts but *sound* reasoning for goals that *mix path quantifiers* is required; or the transfer of verifications *and* refutations of goals from an abstract to the abstracted model is desired – none of these are catered for by the standard approach of abstraction for model checking in [CGL94].

Example 1.1 To illustrate the first point, consider the specification of a global assembly cache within the Microsoft .NET framework [EJS03]. Such a cache C consists of a set of components cs. Each component $c \in cs$ offers a set of services $export$ to other components and requires a set of services $import$ from other components. A service could be a type, a field, a method, etc. A component may have a service called $main$ that would start the execution of software. But a component does not necessarily have such a service, e.g. the component could provide "plug-ins" only.

This optional specification of structure is not possible in labelled transition systems or their state-based versions. Similarly, labelled transition systems cannot specify that "Event α could possibly lead from state s to state s' but is not guaranteed to have that ability." Either the triple (s, α, s') is in the transition relation of the labelled transition system in question, and then such a behavior is always possible, or it isn't, and then such a behavior is always impossible.

To appreciate the second point, let M' be an abstraction of M and let us say that goal ϕ is verified in M' iff $M' \models \phi$ holds; otherwise, ϕ is refuted in M'. It is then elementary to see that the sound transfer of refutations *and* verifications from M' to M requires that for all goals ϕ we have ($M \models \phi$ *iff* $M' \models \phi$): Suppose that $M' \models \phi$ holds, then ϕ is verified at M' and so the sound transfer of verifications yields that $M \models \phi$ holds; conversely if $M' \not\models \phi$ holds, then ϕ is refuted at M' and the sound transfer of refutations renders $M \not\models \phi$. If goals subsume Hennessy-Milner logic [HM85], this forces M and M' to be bisimilar which, as noted by Larsen & Thomsen in [LT88], is too restrictive for aggressive abstraction techniques of state-space compression needed in model checking.

One can show that this limitation is closely related to the problems encountered with the mix of path quantifiers [HJS01]. A labelled transition system M may have another labelled transition system M_{safe} as a *safe simulation* that can match all transitions from M co-inductively. Then every trace of events observable for M is also a possible trace in M_{safe}. So if we restrict reasoning to universal path properties such as "on all paths, if an alarm is triggered, the monitor will inevitable receive this alarm," then verifications on M_{safe} are sound for M [CC77]. Dually, one may develop a notion of *live simulation* for which verifications of existential path properties such as "there is always a way to reach a stable state again" is sound. Mixing universal and existential path properties, e.g. "at all reachable states there is a path to some stable state," naturally leads to the formulation of 3-valued models and their checks [CC00].

Such a mix may even occur if it is not discernible from the structure of the formula ϕ as can be seen by the problem of checking reachability in linear-time temporal logic or computation-tree logic, a universal path quantifier, in the presence of simple fairness constraints, an existential path quantifier. This mix of quantifiers and the need for abstraction-based model checking are even harder to escape in modelling applications that require path constraints not expressible in temporal logics. We mention the extension of model checking to hybrid logics in the context of modelling mobility and agents in distributed systems [FdR03], where existential constraints are needed to express the unique location of agents and universal constraints are needed to specify safety properties.

By now it is well understood that both of the two short-comings above can be addressed within a *3-valued* model-checking framework [Lar89, Dam96, BG99, BG00]. In 3-valued models, an atomic observable, e.g. "in state s an event α can lead to state s'," can not only be "true" or "false" but may be under-specified in that it could be optional, "true or false," empowering specifiers and implementors alike with more degrees of freedom. In Example 1.1 the signature declaration "a component may have a *main* service" is optional in this sense. In identifying scalars with singletons we may write $tt = \{tt\}, ff = \{ff\}$, and $\bot = \{tt, ff\}$ for the values of these observables, respectively. Note that we can determine the value of observables by asking and answering the questions "Is the observed value tt? And if not, is it $\bot$?" as two "no" replies determine that the value of the observable is ff. Therefore any 3-valued model M can be presented as a pair of two two-valued models (M^a, M^c). The acronym "a" stands for **a**sserted whereas "c" denotes **c**onsistent model observables. Accordingly, the truth values tt and ff are interpreted as tt and ff in M^a and in M^c, respectively; and $\bot$ is interpreted as ff in M^a and as tt in M^c; see Fig. 1.

Fig. 1. On the right: a 3-valued transition system with two states and a set $Act = \{\alpha\}$ of possible events. On the left: the corresponding interpretation which shows only *may*-transitions $R^c \setminus R^a$ (with value $\perp$ on the right, *dashed lines* in figures) and solid *must*-transitions R^a (with value tt, solid lines in figures); "transitions" with value ff are omitted

1.1. Completeness of refinement for models of propositional logic

For sake of illustration and in order to gently stage the development of our technical material, we first sketch a 3-valued model-checking framework for propositional logic. The grammar for formulas is

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \phi \tag{1}$$

where p ranges over a countable set of propositional variables Var. We write $\phi \vee \psi$ for $\neg(\neg\phi \wedge \neg\psi)$. A model M is a total function $M\colon Var \to \{tt, ff, \perp\}$. Model M is a standard 2-valued model if $\perp$ is not in the range of M, whereas a value $M(p) = \perp$ expresses that model M is uncertain about the truth value of p. This uncertainty is best expressed non-deterministically by identifying tt with $\{tt\}$, ff with $\{ff\}$, and $\perp$ with $\{tt, ff\}$. This identification immediately gives us the compositional *weak* semantics, Kleene's strong interpretation of 3-valued propositional logic [Kle52], by applying logical connectives point-wise and collecting results as a set. For example, $\perp \wedge tt = \perp$ since

$$\{tt, ff\} \wedge \{tt\} = \{tt \wedge tt, ff \wedge tt\} = \{tt, ff\}\,. \tag{2}$$

Similarly, we obtain $\neg\perp = \perp$, $tt \vee \perp = tt$ etc.

Example 1.2 For a model M with $M(p) = \perp$ the tautology $p \vee \neg p$ evaluates to $\perp \vee \neg\perp = \perp$. This loss of precision suggests that there is a more precise, *strong* semantics for 3-valued propositional logic.

Given two models $M, M'\colon Var \to \{tt, ff, \perp\}$ we say that M refines, is abstracted by, M' iff for all $p \in Var$ we have $M'(p) \leqslant M(p)$ where $\perp$ is the least element and tt and ff are maximal, incomparable elements with respect to $\leqslant$. One can quickly see that a model M' has no refinements other than itself iff M' contains no uncertainty, i.e. iff $\perp$ is not in the range of M'. Otherwise, if M' maps exactly $k < \infty$ many variables to $\perp$, then M' has 2^k many maximal refinements and one could define a strong semantics by appealing to those 2^k models [Bla80]. If we call maximal refinements "implementations" we can show that implementations completely determine refinement.

Proposition 1.1 For any models $M, M'\colon Var \to \{tt, ff, \perp\}$ we have that M refines M' iff all implementations of M are also implementations of M'.

Proof. Since refinement is a transitive relation the "only if" part is shown. Conversely let all implementations of M be implementations of M'. We need to show that M refines M'. For $p \in Var$ it suffices to show $M'(p) \leqslant M(p)$. Proof by contradiction: If $M'(p) \not\leqslant M(p)$, then $M'(p) \neq \perp$. Without loss of generality $M'(p) = tt$. If $M(p) = tt$ we are done. Otherwise, let $M''\colon Var \to \{tt, ff, \perp\}$ be defined by $M''(q) = ff$ if $M(q) = \perp$; and $M''(q) = M(q)$ if $M(q) \neq \perp$. Then M'' is an implementation of M and therefore an implementation of M' by assumption. So $tt = M'(p) \leqslant M''(p) = ff$ is a contradiction. □

The proof above is very simply and would hardly be worth mentioning. But this paper sets out to generalize this fact from propositional models to behavioral models in which we have finitely many events and not only one but possibly infinitely many states.

1.2. Refinement for behavioral models

Three-valued labelled transition systems can be represented as Larsen & Thomsen's modal transition systems [LT88]. Figure 2 depicts two modal transition systems. A modal transition system $M = (M^a, M^c)$ is such that, for each mode $m \in \{a, c\}$, M^m is a labelled transition system (Σ, R^m) where Σ is a set of states and $R^m \subseteq \Sigma \times Act \times \Sigma$ is a transition relation over a set of events Act; furthermore, R^a is contained in R^c. If we think of a modal transition system as a specification, we must ask what are its implementations. Intuitively, a refinement M' of M

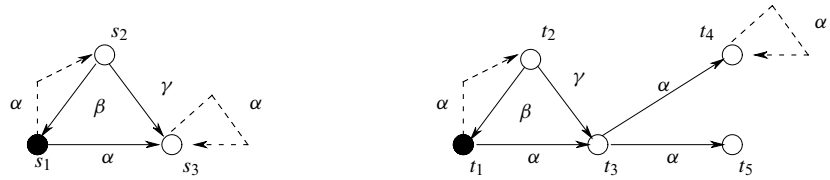

Fig. 2. Two modal transition systems with distinguished filled, initial states. The one to the right refines the one to the left

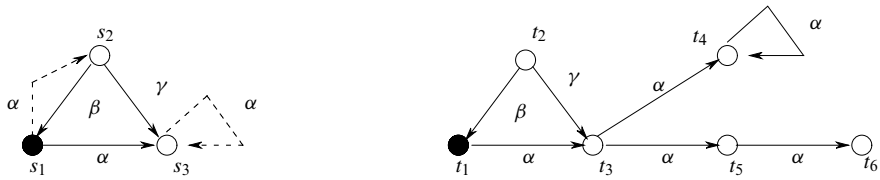

Fig. 3. Two modal transition systems with filled initial states. The one to the right implements the one to the left

resolves some may-transitions ($\perp$ values, elements of $R^c \setminus R^a$, dashed lines in figures) to must-transitions (*tt* values, elements of R^a, solid lines in figures) or to absent transitions (*ff* values, elements of the complement of R^c) and honors all *tt* and *ff* values of M. The resolution of $\perp$ to *tt* constitutes a choice to implement, whereas a resolution of $\perp$ to *ff* reflects a choice not to implement optional behavior. This naive intuition is being formalized in Larsen's co-inductive definition of refinement $M \prec M'$ [LT88], M' refines/is abstracted by M, generalizing bisimulation to the 3-valued setting; see Fig. 2. Definition 2.2 re-states the notion of refinement formally. An implementation I of M should accordingly resolve *all* may-transitions to must-transitions or absent transitions, forcing $I^a = I^c$, and honor all *tt* and *ff* values of M, as can be seen in Fig. 3. One could summarize this discussion of refinement into a slogan: "Must stays Must, May may be Must or Absent, and only Must and May may cause a Must." In this way, we recognize implementations as refining *labelled* transition systems and may define the class of implementations of M as

$$\mathcal{I}[M] = \{I \text{ modal transition system } \mid I^a \text{ equals } I^c \ \&\ M \prec I\}\,. \tag{3}$$

Since refinement $\prec$ is transitive [Lar89], it is immediate that $M \prec M'$ implies $\mathcal{I}[M'] \subseteq \mathcal{I}[M]$; all implementations of a refinement are implementations of what is being refined. So refinement of modal transition systems is sound if modal transition systems are interpreted as their respective classes of implementations since subsequent refinements cannot introduce new implementations.

In this paper we ask, and answer affirmatively, whether refinement is *complete* for this interpretation:

"For all modal transition systems M and M':
If all implementations of M' are implementations of M, does M' refine M?" (4)

Therefore, refinement does not lose any precision if interpreted as reverse inclusion of sets of implementations, the scenario on the left of Fig. 4 is the norm whereas the scenario on the right is impossible. Answering the question in (4) is non-trivial for two reasons. First, refinement can be captured as a winning strategy in a two-person game

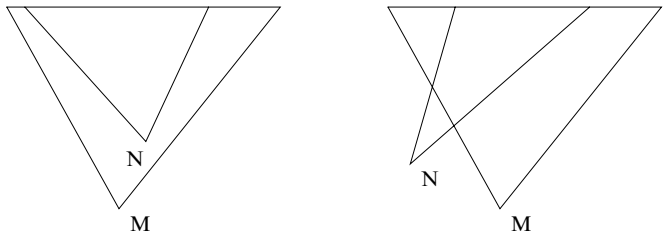

Fig. 4. *Triangles* denote the class of refinements of their lower endpoints. *Top horizontal lines* indicate the classes of implementations. *To the left:* Model N refines M so all implementations of N are implementations of M. *To the right:* All implementations of N are implementations of M, but N does not refine M; this paper shows that this is impossible for modal transition systems

and our proofs depend on the ability to dynamically synthesize winning strategies out of implementations. Second, such a synthesis of winning strategies from implementations may be difficult to obtain for infinite-state or cyclic models and we seem to require a tool that allows us to restrict attention to certain finite-state models. We address the latter problem by expressing the model-checking framework for modal transition systems in a mathematical universe $\mathbb{D}$ that plays a double role [HJS04] as a *SFP*-domain [AJ94] and, up to refinement equivalence, as a modal transition system. A side effect of our affirmative answer is that $\mathcal{I}[M'] \subseteq \mathcal{I}[M]$ can be verified or refuted by always winning a game that involves M and M' only. We develop more consequences of this completeness result toward the end of this paper.

1.3. Weak and strong semantics of Hennessy-Milner logic

For the technical development of this paper we need to define $M \models^a \phi$, the weak semantics of Larsen [Lar89] denoted by $\models$ in loc. cit., and defined over formulas ϕ of Hennessy-Milner logic [HM85]

$$\phi ::= tt \mid \neg\phi \mid \langle\alpha\rangle\phi \mid \phi \wedge \phi \tag{5}$$

where α ranges over a finite set of events Act. Larsen's weak semantics for a pointed modal transition system $(N, i) = ((\Sigma, R^a \subseteq \Sigma \times Act \times \Sigma, R^c \subseteq \Sigma \times Act \times \Sigma), i)$, a modal transition system N with a designated initial state i, is given by two judgments

$$\begin{array}{ll} (N, i) \models^a \phi & \phi \text{ is } \textbf{a}\text{sserted at state } i \text{ in } N \\ (N, i) \models^c \phi & \phi \text{ may be } \textbf{c}\text{onsistent at state } i \text{ in } N \end{array} \tag{6}$$

which are defined by structural induction on ϕ:

$$\begin{array}{ll} (N, i) \models^a tt & (N, i) \models^c tt \\ (N, i) \models^a \neg\phi \text{ iff } (N, i) \not\models^c \phi & (N, i) \models^c \neg\phi \text{ iff } (N, i) \not\models^a \phi \\ (N, i) \models^m \langle\alpha\rangle\phi \text{ iff (for some } (i, \alpha, i') \in R^m, (N, i') \models^m \phi) & (N, i) \models^m \phi \wedge \psi \text{ iff } ((N, i) \models^m \phi \text{ and } (N, i) \models^m \psi) \end{array}$$

where $m \in \{a, c\}$. Note that the definitions of the judgments $\models^a$ and $\models^c$ are mutually recursive via negation, reflecting the duality between validity and satisfiability. The strong semantics, the two judgments $(N, i) \models^{a+} \phi$ and $(N, i) \models^{c-} \phi$ are Bruns & Godefroid's strong/thorough semantics of generalized model checking [BG00] and defined as

$$\begin{array}{ll} (N, i) \models^{a+} \phi & \text{iff (for all } (I, j) \in \mathcal{I}[N, i], (I, j) \,\mathtt{sat}\, \phi) \\ (N, i) \models^{c-} \phi & \text{iff (for some } (I, j) \in \mathcal{I}[N, i], (I, j) \,\mathtt{sat}\, \phi) \end{array} \tag{7}$$

where `sat` is the usual satisfaction relation for Hennessy-Milner logic over labelled transition systems and $\mathcal{I}[N, i]$ is the class of pointed labelled transition systems that refine (N, i). Subsequently we write $\phi \vee \psi$ for $\neg(\neg\phi \wedge \neg\psi)$, $\bigvee$ for its $n$ary version, and abbreviate $\neg\langle\alpha\rangle\neg$ by $[\alpha]$.

Example 1.3 We verify or refute some concrete judgments $\models^a$, $\models^c$, $\models^{a+}$, and $\models^{c-}$. Let N be the modal transition system on the left of Fig. 2.

1. Since $(N, s_3) \not\models^a \langle\gamma\rangle tt$ and $(s_1, \alpha, s_3) \in R^c$ we have $(N, s_1) \not\models^a [\alpha]\langle\gamma\rangle tt$.
2. By the semantics of negation and the previous item, we have $(N, s_1) \models^c \neg[\alpha]\langle\gamma\rangle tt$.
3. But we also have $(N, s_1) \models^{c-} \neg[\alpha]\langle\gamma\rangle tt$ since t_1 in the modal transition system M on the right of Fig. 3 implements s_1 and $(M, t_1) \,\mathtt{sat}\, \neg[\alpha]\langle\gamma\rangle tt$.
4. We have $(N, s_3) \not\models^a \langle\alpha\rangle tt \vee \neg\langle\alpha\rangle tt$ since $(s_3, \alpha, s_3) \in R^c$ but $(s_3, \alpha, x) \in R^a$ for no x.
5. But we have $(N, s_3) \models^{a+} \langle\alpha\rangle tt \vee \neg\langle\alpha\rangle tt$ as the latter formula is a tautology over labelled transition systems.

Modal transition systems are more expressive than labelled transition systems as they can model classes of implementations that are more general than equivalence classes of labelled transition systems with respect to bisimulation. Consequently, $\models^a$ and $\models^{a+}$ are more powerful than `sat` in their ability to verify goals for *all* implementations of a modal transition system.

The weak semantics is sound in that

$$\begin{array}{ll} \models^a \subseteq \models^{a+} & \text{if } \phi \text{ is asserted at } (N, i) \text{ by } \models^a, \text{ then } \phi \text{ holds in all implementations of } (N, i) \\ \models^{c-} \subseteq \models^c & \text{if some implementation of } (N, i) \text{ satisfies } \phi, \text{ then } (N, i) \models^c \phi \text{ holds} \end{array} \tag{8}$$

but loses precision [BG00]. For example, for the modal transition system N on the left of Fig. 2 we saw that $(N, s_3) \not\models^a \langle\alpha\rangle tt \vee \neg\langle\alpha\rangle tt$ but $(N, s_3) \models^{a+} \langle\alpha\rangle tt \vee \neg\langle\alpha\rangle tt$. This incompleteness of $\models^a$ with respect to $\models^{a+}$ is to be expected as $(M, s) \models^a \phi \vee \psi$ attempts to reason that all implementations of (M, s) satisfy $\phi \vee \psi$ by trying to reason that all implementations satisfy ϕ or all implementations satisfy ψ. On the other hand, the complexity of $\models^a$ in the size of the model or formula is the one for the standard semantics `sat` [BG00] so practitioners can get the best of both worlds, the generality of validity checking *and* efficiency of model checking, if they can verify $(M, s) \models^a \phi$.

1.4. Outline of paper

In this first section we attempted to give an informal overview of key concepts and contributions in this paper without stressing the, admittedly, very technical nature of this work. In Sect. 2, we define modal transition systems and refinement formally and present a process algebra *MPA* whose terms have partial modal transition trees as meanings. A game semantics for modal transition systems and their refinement is presented in Sect. 3, characterizing refinements as winning strategies for a verifier. The universal domain $\mathbb{D}$ and its key properties are featured in Sect. 4 and, in Sect. 5, its topological structure is exploited to prove a finite-model property for abstractions in the weak and in the strong semantics. Sect. 6 uses the material developed in Sects. 3–5 to prove the completeness of refinement. As an immediate corollary, we obtain that Hennessy-Milner logic characterizes refinement in the strong semantics as well. In Sect. 7 we explain that completeness of refinement means that validity checking is model checking for characteristic formulas of partial modal transition trees. Sect. 8 demonstrates an application of techniques developed in this paper: checking the consistency and collectively validating multiple models of the same product or design. Finally, Sect. 9 states related work and Sect. 10 concludes. In order to facilitate the progression of the narrative in this paper, we moved proofs of auxiliary or secondary results, as well as definitions and expositions of existing results from domain theory and topology, to an appendix.

2. A process algebra for partial modal transition trees

Throughout we assume a finite set of events Act. We define modal transition systems formally.

Definition 2.1 (Mixed [Dam96, DGG97] and modal transition systems [LT88]) A *mixed transition system* [Dam96, DGG97] is a triple $M = (\Sigma, R^a, R^c)$ such that, for every *mode* $m \in \{a, c\}$, the pair (Σ, R^m) is a labelled transition system, i.e. $R^m \subseteq \Sigma \times Act \times \Sigma$. If $R^a \subseteq R^c$, we call M a *modal* transition system [LT88]. A mixed transition system (M, i) with a designated initial state i is *pointed*. We call elements of R^a *must-transitions* and elements of $R^c \setminus R^a$ *may-transitions*.

In this paper we assume that mixed transition systems are image-finite, unless indicated otherwise, in that for all $s \in \Sigma$, $\alpha \in Act$, and $m \in \{a, c\}$ the set $\{s' \in \Sigma \mid (s, \alpha, s') \in R^m\}$ is finite. In the definition of refinement, we work with the relational inverse of the Q in [LT88, Dam96, HJS04], as done in [GJ02].

Definition 2.2 (Refinement [LT88, Dam96] and mix condition [HJS04]) Let $M = (\Sigma, R^a, R^c)$ be a mixed transition system.

1. A relation $Q \subseteq \Sigma \times \Sigma$ is a *refinement within M* [LT88, Dam96] iff $(s, t) \in Q$ implies for all $\alpha \in Act$
   (a) if $(s, \alpha, s') \in R^a$, there exists some $(t, \alpha, t') \in R^a$ such that $(s', t') \in Q$;
   (b) if $(t, \alpha, t') \in R^c$, there exists some $(s, \alpha, s') \in R^c$ such that $(s', t') \in Q$.

   We write $s \prec_M t$ or $s \prec t$ if there is some refinement Q with $(s, t) \in Q$. In that case t *refines*, *is abstracted by*, s. States s and t are *refinement-equivalent* iff ($s \prec t$ and $t \prec s$). We write $(M, i) \prec (N, j)$ if j refines i in the mixed transition system that is the disjoint union of M and N. Let $\mathcal{I}[M, i]$ be the class of *implementations* of (M, i), those mixed transition systems (N, j) without any may-transitions that refine (M, i).
2. We say that M satisfies the *mix condition (MC)* iff for all $(s, \alpha, s') \in R^a$ there is some $(s, \alpha, s'') \in R^a \cap R^c$ such that $s' \prec s''$.

Example 2.1 Figure 5 [Hut05] reveals that mixed transition systems (Σ, R^a, R^c) that satisfy the mix condition (MC) are refinement-equivalent to modal transition systems $(\Sigma, R^a \cap R^c, R^c)$ and so merely modal transition systems in disguise [HJS04].

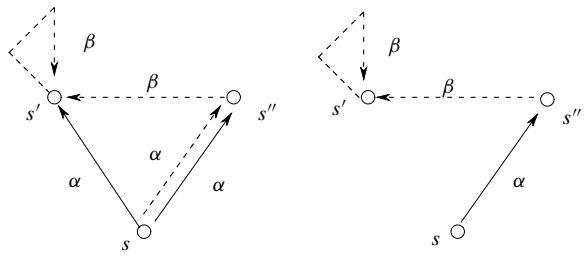

Fig. 5. *On the left:* a mixed transition system (Σ, R^a, R^c) satisfying the mix condition (MC). In this case *dashed lines* denote elements of R^c and *solid lines* denote elements of R^a. For $(s, \alpha, s') \in R^a$ there is $(s, \alpha, s'') \in R^a \cap R^c$ with $s' \prec s''$. The other tuple in R^a is matched by itself as it is in $R^a \cap R^c$. *On the right:* a modal transition system that is refinement-equivalent to the mixed transition system on the left. Its set of must-transitions is $R^a \cap R^c$ (*solid lines*) and its set of may-transitions is R^c (*solid* or *dashed lines*)

$$\frac{}{\bot \longrightarrow^{\gamma}_{\bot} \bot}\ \text{MayStub}$$

$$\frac{}{\alpha_{tt}.p \longrightarrow^{\alpha}_{tt} p}\ \text{MustPrefix} \qquad \frac{}{\alpha_{\bot}.p \longrightarrow^{\alpha}_{\bot} p}\ \text{MayPrefix}$$

$$\frac{p \longrightarrow^{\alpha}_{v} p'}{p + q \longrightarrow^{\alpha}_{v} p'}\ \text{LeftChoice} \qquad \frac{q \longrightarrow^{\alpha}_{v} q'}{p + q \longrightarrow^{\alpha}_{v} q'}\ \text{RightChoice}$$

Fig. 6. Structural operational semantics for terms of the process algebra *MPA*. An expression $p \longrightarrow^{\alpha}_{\bot} p'$ denotes a may-transition from p to p' whereas $p \longrightarrow^{\alpha}_{tt} p'$ denotes a must-transition from p to p', each labelled with some $\alpha \in Act$. A value v stands for either $\bot$ or tt. There are no transitions out of $\mathbf{0}$ and the free occurrence of γ ranges over all events in *Act*

Since any element of $\mathcal{I}[M, i]$ can be "colored" with any set, $\mathcal{I}[M, i]$ is not a set but a class. This has neither consequences for the results of this paper nor for practical aspects of model checking. Since the union of refinements within M is a refinement within M, $\prec_M$ is the greatest refinement relation within M. In all respects, we identify labelled transition systems with modal transition systems that have no may-transitions, i.e. for which $R^a = R^c$. In that case, both $\models^a$ and $\models^c$ are equal to the standard satisfaction relation `sat` of Hennessy-Milner logic over labelled transition systems.

Example 2.2 The relation $Q = \{(s_1, t_1), (s_2, t_2), (s_3, t_3), (s_3, t_4), (s_3, t_5)\}$ between states of the pointed modal transition systems from Fig. 2 is a refinement within their union so $s_1 \prec t_1$ etc.

Definition 2.3 ([Hut04]) The *process algebra MPA* is a fragment of the modal process logic in [Lar89]:

$$p ::= \mathbf{0} \mid \bot \mid \alpha_{tt}.p \mid \alpha_{\bot}.p \mid p + p \tag{9}$$

where α denotes any event in *Act* and all p in $p + p$ are different from $\mathbf{0}$ and $\bot$. The *structural operational semantics for terms of MPA* is given in Fig. 6. We write $[\![\ p\ ]\!]$ for the modal transition system derived from term $p \in MPA$ and these rules.

The prefix $\alpha_{tt}.$ is the sole direct source of must-transitions whereas $\bot$ and $\alpha_{\bot}.$ are the sole direct sources of may-transitions. Non-determinism + is the indirect source of transitions.

Example 2.3 Figure 7 illustrates a process term $p = \alpha_{\bot}.(\beta_{tt}.\bot + \alpha_{\bot}.\beta_{tt}.\bot) + \alpha_{tt}.(\alpha_{tt}.\mathbf{0} + \beta_{\bot}.\bot) \in MPA$ and its operational meaning $([\![\ p\ ]\!], p)$ as a partial modal transition tree.

Such meanings $([\![\ p\ ]\!], p)$ are trees in that leaves either deadlock or turn into *may-stubs* that have may-transition loops for each $\gamma \in Act$, so may-stubs model partiality. Partial modal transition trees are finite trees that approximate a set of infinite modal transition systems by abstracting some states and all their reachable states with $\bot$. This idea of approximating infinite process terms or behaviors with finite ones can already be seen in the algebraic semantics à la Nivat-Courcelle-Guessarian [CN76] or à la Goguen-Thatcher-Wagner-Wright [GTWW77]. As this is elementary, we don't show formally that every partial modal transition tree is the meaning of some

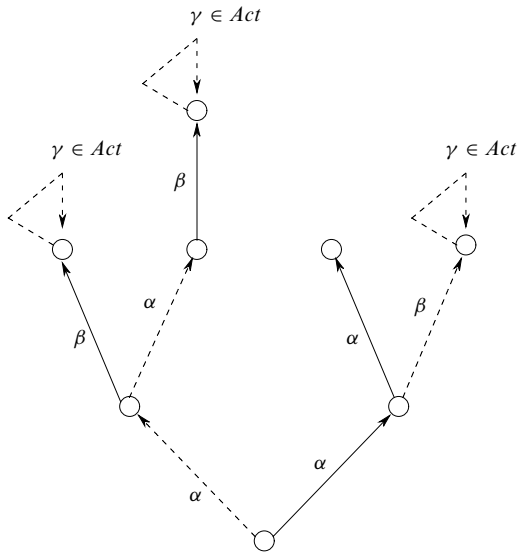

Fig. 7. The operational meaning of the term $\alpha_{\perp}.(\beta_{tt}.\perp + \alpha_{\perp}.\beta_{tt}.\perp) + \alpha_{tt}.(\alpha_{tt}.\mathbf{0} + \beta_{\perp}.\perp)$ is a *partial* modal transition *tree* in that its leaves either deadlock ($\mathbf{0}$) or model a may-stub ($\perp$) that loops for all events $\gamma \in Act$

term of MPA and that every such meaning is a partial modal transition tree as all summands are guarded. For example, the term $\alpha_{\perp}.\mathbf{0} + \beta_{tt}.\mathbf{0} + \perp$ is not in MPA as its rightmost summand $\perp$ is not guarded by any prefix.

3. Refinements as winning strategies of a two-person game

Refinements of pointed modal transition systems can be characterized in terms of winning strategies of Ehrenfeucht-Fraïssé games, as worked out for labelled transition systems and bisimulation by Stirling in [Sti96]. The idea is that checking whether (M, i) is refined by (N, j) can be reduced to showing that the verifier has a winning strategy in a two-person game $\mathcal{G}[(M, i), (N, j)]$ played between a refuter, who tries to show that (M, i) is not refined by (N, j), and a verifier, who wants to establish that (N, j) refines (M, i). Since modal transition systems and refinement are generalizations of labelled transition systems and bisimulation, our adaptation of Stirling's concepts and results to refinement of modal transition systems is straightforward.

Definition 3.1 Let (M, i) and (N, j) be two pointed modal transition systems.

1. We define a *two-person game* $\mathcal{G}[(M, i), (N, j)]$ as follows.
   - *Game positions* are all pairs (s, t) where s and t are states of M and N, respectively;
   - there are two players, a *refuter* and a *verifier*;
   - each move consists of a question by the refuter followed, if possible, by an answer of the verifier:
     - if in position (s, t) the refuter asks as question a R^a-transition (s, α, s') in M, the verifier has to answer with a R^a-transition (t, α, t') in N resulting in the new game position (s', t');
     - if in position (s, t) the refuter asks as question a R^c-transition (t, α, t') in N, then the verifier has to answer with a R^c-transition (s, α, s') in M resulting in the new game position (s', t'); since $R^a \subseteq R^c$, the question or answer may well be a must-transition here;
   - no other kinds of questions can be asked and only the refuter can ask questions; and
   - a *run* is a possibly infinite sequence of moves beginning in the initial game position (i, j); the refuter wins only those runs on which the verifier eventually cannot answer; therefore, the verifier wins all infinite runs and those runs with a position in which the refuter cannot ask a question.
2. A *strategy for the refuter* is a partial function that maps each game position to at most one legitimate question for that position.

3. A *strategy for the verifier* is a partial function that maps each game position and question for that position to at most one legitimate answer to that question.
4. A strategy is *winning* if all runs played according to that strategy are won by the player who obeys it.

Therefore, strategies are history-free and winning strategies are total functions on positions reachable from the initial position (i, j).

Example 3.1 1. For the relation $Q = \{(s_1, t_1), (s_2, t_2), (s_3, t_3), (s_3, t_4), (s_3, t_5)\}$, which witnesses a refinement $s_1 \prec t_1$ of the pointed modal transition systems from Fig. 2, one can easily synthesize a winning strategy for the verifier. For example, any question raised by the refuter in game position (s_3, t_3) has to be a R^c-transition from t_3 labelled with α and the verifier responds with the R^c-transition from s_3 back to itself. The refuter cannot ask a question that involves a transition out of s_3 in position (s_3, t_3) since there are no R^a-transitions out of s_3.

2. To see a winning strategy for the refuter, consider the game $\mathcal{G}[(N, s_1), (M, r)]$ where N is the modal transition system on the left of Fig. 2 and (M, r) is the pointed modal transition system from Fig. 7, r being the root node. The refuter begins the run by asking the R^a-transition (s_1, α, s_3) in N. The verifier can only reply with the R^a-transition labelled with α to the node n of the right subtree since the transition to the left subtree is not in R^a. At game position (s_3, n) the refuter can now ask the R^c-transition labelled with β and source n, but the verifier has no matching answer in N out of s_3.

As claimed, refinement between pointed modal transition systems is characterized by the existence of a winning strategy for the verifier in the respective refinement game.

Theorem 3.1 Let (M, i) and (N, j) be pointed modal transition systems. Then (M, i) is refined by (N, j) iff the verifier has a winning strategy in the game $\mathcal{G}[(M, i), (N, j)]$. In that game, exactly one player has a winning strategy.

Proof. The proof in [Sti96] can be generalized to be aware of the presence of two kinds of transitions and to make use of the logical characterization of refinement through Hennessy-Milner logic in [Lar89, HJS04]. □

4. A universal domain and modal transition system

One can show completeness of refinement by assuming that (N, j) does not refine (M, i) and constructing an implementation of (N, j) that is not an implementation of (M, i). Using Theorem 3.1, one could do this by synthesizing a suitable implementation from a winning strategy for the refuter in the game $\mathcal{G}[(M, i), (N, j)]$. We do not know how to do this unless (M, i) and (N, j) are *partial* modal transition *trees*. Therefore, we require a tool that ensures it is sufficient to consider the case of partial modal transition trees. Before we present this tool we recall concepts from domain theory [AJ94]. We refer to Appendix A for standard definitions, notation, and results from topology and domain theory used subsequently.

The mixed powerdomain $\mathcal{M}[D]$ [Hec90, Gun92] of a *SFP*-domain D has as elements all pairs (L, U) where L is Scott-closed and U is Scott-compact saturated such that L and U satisfy the mix condition

$$L = \downarrow(L \cap U)\,. \tag{10}$$

The order on $\mathcal{M}[D]$ is defined by

$$(L, U) \leqslant (L', U') \qquad \text{iff} \qquad (L \subseteq L' \;\&\; U' \subseteq U)\,. \tag{11}$$

The mix condition (10) turns out to precisely express the mix condition (MC) for mixed transition systems if, for some event α, we interpret elements of L as R^a-successors and elements of U as R^c-successors of some state s and see the order as refinement [HJS04]: if $s' \in L$, i.e. if $(s, \alpha, s') \in R^a$, then there is some $s'' \in L \cap U$ with $s' \leqslant s''$ by (10) and so $(s, \alpha, s'') \in R^a \cap R^c$. Definition 4.1 below makes all of this formal. For the sake of illustration, we state a prominent example of the mixed power domain.

Example 4.1 1. [Hec90, Hec91] For $D = \{*\}$, $\mathcal{L}[D] = \mathcal{U}[D] = \{\{\}, \{*\}\}$ as sets. The mixed powerdomain $\mathcal{M}[D]$ contains exactly $(\{\}, \{\})$, $(\{*\}, \{*\})$, and $(\{\}, \{*\})$ as all three pairs satisfy the mix condition (10), whereas the pair $(\{*\}, \{\})$ does not. If we write $\bot = (\{\}, \{*\})$, $tt = (\{*\}, \{*\})$, and $ff = (\{\}, \{\})$, the order on $\mathcal{M}[D]$ is the one used earlier on for refinement of models for propositional logic: $\bot < tt, ff$, the information ordering of [BG99].

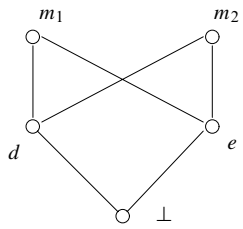

Fig. 8. A domain in which d and e have the same set of maximal elements $\{m_1, m_2\}$ above them although d and e are incomparable in the order

2. Note that in $\mathcal{M}[\{*\}]$ it is the case that $d \leqslant e$ iff all maximal elements above e are also above d. This is not true in every domain. For example, it is false in the domain in Fig. 8 and known to be false in $\mathcal{M}[\mathcal{M}[\{*\}]]$ [ABH97], but it is true in a domain D iff (for all $d \in D$, the element d is the infimum in D of all maximal elements above d). We can show completeness of refinement only since our domain model is the fixed point of a domain equation, which allows us to unfold the mixed powerdomain construction in an unbounded way.

Definition 4.1 ([HJS04])

1. Since $\mathcal{M}[D]$ is a *SFP*-domain if D is one [Hec90], we can solve the domain equation
$$D = \prod_{\alpha \in Act} \mathcal{M}[D] \tag{12}$$
where $\prod_{\alpha \in Act}$ denotes the product over all events in Act, and write $\mathbb{D}$ for the *SFP*-domain and *initial solution* of that equation [HJS04].
2. Every element $d \in \mathbb{D}$ may be interpreted as a pointed mixed transition system $(\mathcal{D}, d)$ where d is the unique initial state and the recursion $d = ((d^a_\alpha, d^c_\alpha))_{\alpha \in Act}$ of (12) specifies that all d' in the set d^a_α (d^c_α) are exactly the R^a_α-successors (R^c_α-successors) of d in $(\mathcal{D}, d)$ (respectively). This makes $\mathbb{D}$ into a mixed transition system, which we denote subsequently by $\mathcal{D} = (\mathbb{D}, \mathbb{R}^a, \mathbb{R}^c)$.
3. As noted in Proposition 1 in [HJS04] and as seen in Example 2.1, the mix condition (10) guarantees that the mixed transition system $\mathcal{D}$ is refinement-equivalent to the modal transition system $(\mathbb{D}, \mathbb{R}^a \cap \mathbb{R}^c, \mathbb{R}^c)$. Thus, all reasoning that is invariant under refinement equivalence — as is the case in this paper — may be done with the latter modal transition system instead of $\mathcal{D}$ and we abuse notation to refer to that modal transition system as $\mathcal{D}$ as well.

We recall some facts from [HJS04] needed as basic tools in most proofs of this paper.

Facts 4.1 1. The order $\leqslant$ on $\mathbb{D}$ is the greatest refinement in $\mathcal{D}$, i.e. the union of all refinements within $\mathcal{D}$: $d \leqslant e$ in $\mathbb{D}$ iff $(\mathcal{D}, d) \prec (\mathcal{D}, e)$ by Theorem 5 in [HJS04].

2. For all pointed modal transition systems (M, i) and ϕ of Hennessy-Milner logic, $(M, i) \models^a \phi$ implies $(M, i) \models^c \phi$ by item 2 of Theorem 3 in [HJS04].
3. For all pointed modal transition systems without any may-transitions $\models^a$ and $\models^c$ equal the standard satisfaction relation `sat` over labelled transition systems by Theorem 2 in [HJS04].
4. Refinement between modal transition systems without any may-transitions coincides with bisimulation [Lar89].
5. For all pointed modal transition systems (M, i) and (N, j) we have $(M, i) \prec (N, j)$ iff (for all ϕ of Hennessy-Milner logic, the relation $(M, i) \models^a \phi$ implies $(N, j) \models^a \phi$) iff (for all ϕ of Hennessy-Milner logic, the relation $(N, j) \models^c \phi$ implies $(M, i) \models^c \phi$) by [Lar89] and Theorem 5 in [HJS04].
6. Every pointed modal transition system (M, i) has an embedding $\llparenthesis M, i \rrparenthesis \in \mathbb{D}$ such that $(\mathcal{D}, \llparenthesis M, i \rrparenthesis)$ is refinement-equivalent to (M, i) by item 1 of Theorem 6 in [HJS04]. In particular, the compact elements of $\mathbb{D}$ are precisely the embeddings of partial modal transition trees. (We discuss details of this embedding on page 124.)

Example 4.2 1. Item 1 of Fact 4.1 means that we can identify the order on $\mathbb{D}$ with refinement on $\mathcal{D}$. For example, $\perp_{\mathbb{D}} \leqslant \llparenthesis \mathbf{0} \rrparenthesis$ for the denotational semantics defined below and so $(\mathcal{D}, \llparenthesis \mathbf{0} \rrparenthesis)$ refines $(\mathcal{D}, \llparenthesis \perp \rrparenthesis)$.

2. Item 2 allows us to weaken $\models^a$ judgments to $\models^c$ judgments. Let N be the model on the left of Fig. 2. Then $(N, s_2) \models^a [\beta]\langle\alpha\rangle tt$ and so $(N, s_2) \models^c [\beta]\langle\alpha\rangle tt$ follows.

3. Item 3 can be appreciated by checking judgments $\models^m$ for the model on the right of Fig. 3.
4. Item 4 follows since then $R^c = R^a$ so the definition of refinement reads as the familiar definition of bisimulation.
5. The soundness part of item 5 is often used in this paper. Revisiting item 3 of Example 1.3 we learn that $(M, t_1) \models^c \neg[\alpha]\langle\gamma\rangle tt$ by item 3 of Fact 4.1 and so $s_1 \prec t_1$ implies $(M, s_1) \models^c \neg[\alpha]\langle\gamma\rangle tt$.
6. Let $Act = \{\alpha, \beta\}$. If we extend MPA with recursion we can specify $p = \alpha_\perp.(\beta_{tt}.p + \alpha_{tt}.\mathbf{0})$ which is embedded into $\mathbb{D}$ as $p_\mathbb{D}$ via a system of recursive equations. We order $\alpha < \beta$ in tuples: $\mathbf{0}_\mathbb{D} = ((\{\}, \{\}), (\{\}, \{\}))$, $p_\mathbb{D} = ((\{\}, \uparrow\{p'_\mathbb{D}\}), (\{\}, \{\}))$, and $p'_\mathbb{D} = ((\downarrow\mathbf{0}_\mathbb{D}, \uparrow\mathbf{0}_\mathbb{D}), (\downarrow\{p_\mathbb{D}\}, \uparrow\{p_\mathbb{D}\}))$.

In [Lar89] we find an alternative way of checking refinement by checking a system of greatest fixed-point equations with a semantics as in $\models^a$. This system of equations simply expresses the refinement game as a system of formulas such that its $\models^a$ check captures the existence of a winning strategy.

Definition 4.2 [Lar89] Let $(M, i) = ((\Sigma, R^a, R^c), i)$ be a pointed modal transition system, not necessarily image-finite. For each $s \in \Sigma$ we define a formula $X_{(M,s)}$ via the greatest fixed point of the recursive equations

$$X_{(M,s)} = \Big(\bigwedge_{(s,\alpha,s') \in R^a} \langle\alpha\rangle X_{(M,s')} \Big) \wedge \bigwedge_{\alpha \in Act} [\alpha]\Big(\bigvee_{(s,\alpha,s') \in R^c} X_{(M,s')} \Big) \tag{13}$$

for all $s \in \Sigma$, as specified in equation (3) in [Lar89]. Intuitively, each $X_{(M,s)}$ denotes the set of states t of M that are refinements of s within M. Each equation (13) is a transducer that computes the set on the left-hand side from the sets on the right-hand side such that the logical connectives are interpreted with respect to $\models^a$ as already defined. To solve this system of equations, we initially set each $X_{(M,s)}$ to be the entire state space and then update the values of these sets simultaneously through their transducers for all s until all sets stabilize.

Please note that the conjunctions and disjunctions in (13) may well be infinite. Even for an image-finite pointed modal transition system, the formula $X_{(M,i)}$ is not in general expressible in the modal mu-calculus [Koz83] via its explicit operator for greatest fixed points but it is expressible in this way if only finitely many states are R^c-reachable from i in M [Hut04].

Example 4.3 We write $\nu Z.\phi$ for greatest fixed-point formulas with recursion variable Z and recursion body ϕ and express $X_{(N,s_1)}$ for the s_1 in the left of Fig. 3 as a closed formula of the modal mu-calculus using greatest fixed points only. Let $X_{(N,s_3)}$ be represented by the closed formula $\nu Z_{s_3}.[\alpha]Z_{s_3}$ of the modal mu-calculus. Let $X_{(N,s_2)}$ be represented by the formula $\nu Z_{s_2}.\langle\beta\rangle Z_{s_1} \wedge \langle\gamma\rangle X_{(N,s_3)} \wedge [\beta]Z_{s_1} \wedge [\gamma]X_{(N,s_3)}$ which contains Z_{s_1} as only free variable. Finally, let $X_{(N,s_1)}$ be represented by the closed formula $\nu Z_{s_1}.\langle\alpha\rangle X_{(N,s_3)} \wedge [\alpha](X_{(N,s_3)} \vee X_{(N,s_2)})$ of the modal mu-calculus. Extending the semantics of $\models^a$ to greatest fixed points via greatest semantic fixed points [HJS01] therefore captures the meaning of $(M, i) \models^a X_{(N,s_1)}$ stated in Definition 4.2.

At the end of this section, we see that $X_{(M,i)}$ is expressible in Hennessy-Milner logic for all pointed partial modal transition trees (M, i). We now secure that $\models^a$ checks of $X_{(M,i)}$ capture refinement checks.

Lemma 4.1 Let (M, i) and (N, j) be pointed modal transition systems, not necessarily image-finite. Then $(M, i) \prec (N, j)$ iff $(N, j) \models^a X_{(M,i)}$.

Proof. Both statements are shown in the proof for Theorem 4.1 in [Lar89] for *image-finite* modal transition systems but this property is never needed in that proof, apart from the finiteness of disjunctions and conjunctions on which we do not rely here. □

Definition 4.3 ([Hut04]) Figure 9 shows a *denotational semantics* $\{\!|\ p\ |\!\}$ for terms p of MPA in $\mathbb{D}$.

Example 4.4 For $p = \alpha_\perp.\perp + \beta_{tt}.\mathbf{0}$ and $Act = \{\alpha, \beta\}$, the pair $\{\!|\ p\ |\!\}$ contains two pairs: $(\{\}, \uparrow\{\!|\ \perp\ |\!\})$ for α, and $(\downarrow\{\!|\ \mathbf{0}\ |\!\}, \uparrow\{\!|\ \mathbf{0}\ |\!\})$ for β.

We need to establish that the denotational semantics $\{\!|\ p\ |\!\}$ captures the operational semantics $[\!|\ p\ |\!]$.

Lemma 4.2 For all p of MPA, the pointed mixed transition systems $([\!|\ p\ |\!], p)$ and $(\mathcal{D}, \{\!|\ p\ |\!\})$ are refinement-equivalent.

Proof. We use structural induction on p and present only proofs for the two clauses for prefixes for sake of illustration. Assume that $([\!|\ p\ |\!], p)$ and $(\mathcal{D}, \{\!|\ p\ |\!\})$ are refinement-equivalent.

$$\{\!|\ \mathbf{0}\ |\!\} = ((\{\}, \{\}))_{\alpha \in Act} \qquad \{\!|\ \bot\ |\!\} = ((\{\}, \mathbb{D}))_{\alpha \in Act}$$

$$(\{\!|\ \alpha_{tt}.p\ |\!\}^a_\alpha, \{\!|\ \alpha_{tt}.p\ |\!\}^c_\alpha) = (\downarrow\!\{\!|\ p\ |\!\}, \uparrow\!\{\!|\ p\ |\!\}) \qquad (\{\!|\ \alpha_{tt}.p\ |\!\}^a_\beta, \{\!|\ \alpha_{tt}.p\ |\!\}^c_\beta) = (\{\}, \{\}),\ \alpha \neq \beta$$

$$(\{\!|\ \alpha_{\bot}.p\ |\!\}^a_\alpha, \{\!|\ \alpha_{\bot}.p\ |\!\}^c_\alpha) = (\{\}, \uparrow\!\{\!|\ p\ |\!\}) \qquad (\{\!|\ \alpha_{\bot}.p\ |\!\}^a_\beta, \{\!|\ \alpha_{\bot}.p\ |\!\}^c_\beta) = (\{\}, \{\}),\ \alpha \neq \beta$$

$$\{\!|\ p + q\ |\!\}^a_\gamma = \{\!|\ p\ |\!\}^a_\gamma \cup \{\!|\ q\ |\!\}^a_\gamma \qquad \{\!|\ p + q\ |\!\}^c_\gamma = \{\!|\ p\ |\!\}^c_\gamma \cup \{\!|\ q\ |\!\}^c_\gamma,\ \gamma \in Act\ .$$

Fig. 9. Denotational semantics $\{\!|\ p\ |\!\}$ for terms p of *MPA* in $\mathbb{D}$; it interprets $\mathbf{0}$ as deadlock in $\mathbb{D}$, $\bot$ as the least element of $\mathbb{D}$, + as the mix union of [Hec90], and prefixes as expected except for saturations with $\downarrow$ and $\uparrow$ to ensure that the meaning is in $\mathbb{D}$

1. We show $(\llbracket\ \alpha_{tt}.p\ \rrbracket, p) \prec (\mathcal{D}, \{\!|\ \alpha_{tt}.p\ |\!\})$:
   - A R^a-transition out of $(\llbracket\ \alpha_{tt}.p\ \rrbracket, \alpha_{tt}.p)$ can only arise through $\alpha_{tt}.p \longrightarrow^\alpha_{tt} p$ but $(\{\!|\ \alpha_{tt}.p\ |\!\}, \alpha, \{\!|\ p\ |\!\})$ is a $\mathbb{R}^a$-transition in $\mathcal{D}$ and $(\llbracket\ p\ \rrbracket, p) \prec (\mathcal{D}, \{\!|\ p\ |\!\})$ by induction.
   - Let $(\{\!|\ \alpha_{tt}.p\ |\!\}, \beta, d)$ be a $\mathbb{R}^c$-transition in $\mathcal{D}$. Then $\beta = \alpha$ and $d \in \uparrow\!\{\!|\ p\ |\!\}$. By induction, $(\llbracket\ p\ \rrbracket, p)$ refines $(\mathcal{D}, \{\!|\ p\ |\!\})$ which, together with $\{\!|\ p\ |\!\} \leqslant d$, item 1 of Fact 4.1, and the transitivity of refinement implies $(\llbracket\ p\ \rrbracket, p) \prec (\mathcal{D}, d)$; and $(\alpha_{tt}.p, \alpha, p)$ is also a R^c-transition in $\llbracket\ \alpha_{tt}.p\ \rrbracket$.
2. Next we show $(\mathcal{D}, \{\!|\ \alpha_{tt}.p\ |\!\}) \prec (\llbracket\ \alpha_{tt}.p\ \rrbracket, \alpha_{tt}.p)$:
   - Let $(\{\!|\ \alpha_{tt}.p\ |\!\}, \beta, d)$ be a $\mathbb{R}^a$-transition. Then $\beta = \alpha$ and $d \in \downarrow\!\{\!|\ p\ |\!\}$. Induction renders that $(\mathcal{D}, \{\!|\ p\ |\!\})$ is refined by $(\llbracket\ p\ \rrbracket, p)$ and so item 1 of Fact 4.1 and the transitivity of refinement imply $(\mathcal{D}, d) \prec (\llbracket\ p\ \rrbracket, p)$; and $(\alpha_{tt}.p, \alpha, p)$ is a R^a-transition.
   - A R^c-transition out of $(\llbracket\ \alpha_{tt}.p\ \rrbracket, \alpha_{tt}.p)$ has to arise from $\alpha_{tt}.p \longrightarrow^\alpha_{tt} p$ but $(\{\!|\ \alpha_{tt}.p\ |\!\}, \alpha, \{\!|\ p\ |\!\})$ is a $\mathbb{R}^c$-transition and $(\llbracket\ p\ \rrbracket, p) \prec (\mathcal{D}, \{\!|\ p\ |\!\})$ by induction.
3. We show $(\llbracket\ \alpha_{\bot}.p\ \rrbracket, \alpha_{\bot}.p) \prec (\mathcal{D}, \{\!|\ \alpha_{\bot}.p\ |\!\})$:
   - There are no initial R^a-transitions out of $(\llbracket\ a_{\bot}.p\ \rrbracket, \alpha_{\bot}.p)$.
   - Let $(\{\!|\ \alpha_{\bot}.p\ |\!\}, \beta, d)$ be a $\mathbb{R}^c$-transition. Then $\beta = \alpha$ and $d \in \uparrow\!\{\!|\ p\ |\!\}$. By induction, $(\llbracket\ p\ \rrbracket, p) \prec (\mathcal{D}, \{\!|\ p\ |\!\})$ which, as reasoned before, implies $(\llbracket\ p\ \rrbracket, p) \prec (\mathcal{D}, d)$; and $(\alpha_{\bot}.p, \alpha, p)$ is also a R^c-transition.
4. Finally, we show $(\mathcal{D}, \{\!|\ \alpha_{\bot}.p\ |\!\}) \prec (\llbracket\ \alpha_{\bot}.p\ \rrbracket, \alpha_{\bot}.p)$:
   - There are no initial $\mathbb{R}^a$-transitions out of $(\mathcal{D}, \{\!|\ \alpha_{\bot}.p\ |\!\})$.
   - A R^c-transition out of $(\llbracket\ \alpha_{\bot}.p\ \rrbracket, \alpha_{\bot}.p)$ can only arise through $\alpha_{\bot}.p \longrightarrow^\alpha_{\bot} p$ but $(\{\!|\ \alpha_{\bot}.p\ |\!\}, \alpha, \{\!|\ p\ |\!\})$ is a $\mathbb{R}^c$-transition and $(\llbracket\ p\ \rrbracket, p) \prec (\mathcal{D}, \{\!|\ p\ |\!\})$ by induction. □

One can use that lemma to embed any pointed modal transition system (N, i) faithfully into $\mathcal{D}$. For each $m \geqslant 0$ let $(N[m], i)$ be the partial modal transition tree that unwinds the transitions beginning from i in N as a modal transition tree of depth m [HJS04]. If a leaf in that tree has a R^c-successor state in N, that leaf turns into a may-stub; otherwise, said leaf deadlocks. Figure 10 depicts two of these approximations for the pointed modal transition system (N, t_1) of Fig. 2. For each $m \geqslant 0$ there is some $p_m \in MPA$ such that $(N[m], i)$ is refinement-equivalent to $(\llbracket\ p_m\ \rrbracket, p_m)$ as all $(N[m], i)$ are pointed modal transition trees. For example, for $(N[2], t_1)$ of Fig. 10 we may choose $p_2 \in MPA$ as $\alpha_{\bot}.(\beta_{tt}.\bot + \gamma_{tt}.\bot) + \alpha_{tt}.(\alpha_{tt}.\bot + \alpha_{tt}.\mathbf{0})$. Furthermore, the set $\{\{\!|\ p_m\ |\!\} \mid m \geqslant 0\}$ is directed in $\mathbb{D}$ as $m \leqslant m'$ implies $(N[m], i) \prec (N[m'], i)$. For example, we have $(N[1], t_1) \prec (N[2], t_1)$ in Fig. 10 where refinement is name identity on the common temporal layer except that may-leaves, e.g. t_2 on the left, get also refined by all successor states of their name-identical version in $(N[2], t_1)$, e.g. t_1 and t_3 as successor states of t_2 on the right. So

$$\langle\!|\ N, i\ |\!\rangle = \bigvee_{m \geqslant 0} \{\!|\ p_m\ |\!\} \tag{14}$$

exists as a directed supremum in $\mathbb{D}$ and $(\mathcal{D}, \langle\!|\ N, i\ |\!\rangle)$ and (N, i) are refinement-equivalent by item 6 of Fact 4.1.

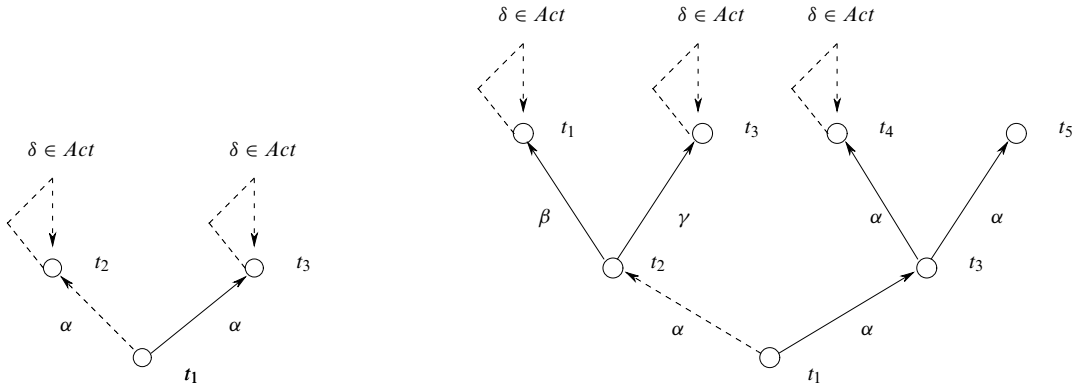

Fig. 10. *To the left:* The approximation $(N[1], t_1)$ of the pointed modal transition system (N, t_1) *on the right* of Fig. 2; note how t_2 and t_3 turn into may-stubs since (t_2, γ, t_3) and (t_3, α, t_4) are in R^c in N. *To the right:* A more precise approximation $(N[2], t_1)$ of that same pointed modal transition system, which recognizes that t_5 deadlocks

$$\psi_{\mathbf{0}} = \bigwedge_{\alpha \in Act} \neg\langle\alpha\rangle tt$$

$$\psi_{\perp} = tt$$

$$\psi_{\alpha_{tt}.p} = \langle\alpha\rangle\psi_p \wedge [\alpha]\psi_p \wedge \bigwedge_{\beta \neq \alpha} \neg\langle\beta\rangle tt$$

$$\psi_{\alpha_{\perp}.p} = [\alpha]\psi_p \wedge \bigwedge_{\beta \neq \alpha} \neg\langle\beta\rangle tt$$

$$\psi_{p+q} = \bigwedge\{\langle\alpha\rangle\psi_{r'} \mid \alpha \in Act,\ p+q \longrightarrow^{\alpha}_{tt} r'\} \wedge \bigwedge_{\alpha \in Act} [\alpha] \bigvee\{\psi_{r'} \mid \exists v \in \{\perp, tt\}\colon p+q \longrightarrow^{\alpha}_{v} r'\}$$

Fig. 11. The customizations of (13) for terms of *MPA*, where the shape of partial modal transition trees ensures that these formulas are inductively definable within Hennessy-Milner logic: For all pointed modal transition system (N, i) we have $(N, i) \models^a \psi_p$ iff $(\llbracket\ p\ \rrbracket, p) \prec (N, i)$. The expressions $\neg\langle\beta\rangle tt$ above result from the expression $\ldots[\beta](\bigvee\ldots)$ in (13) where the disjunction ranges over the empty set and therefore denotes *ff*

From Theorem 6.4 in [Hec90] we can infer that the compact elements of $\mathbb{D}$ are all denotations of *MPA*:

$$\mathbf{K}(\mathbb{D}) = \{\{\!|\ p\ |\!\} \mid p \in MPA\}\ . \tag{15}$$

In Example 4.3 we already saw that formulas $X_{(M,s)}$ are expressible in the modal mu-calculus for finite-state models. For *partial* modal transition *trees* these formulas are even expressible in Hennessy-Milner logic.

Definition 4.4 For every $p \in MPA$ let ψ_p of Hennessy-Milner logic be defined inductively as in Fig. 11.

We ensure that ψ_p captures $X_{(\llbracket p \rrbracket, p)}$.

Lemma 4.3 For all $p \in MPA$ and all pointed modal transition systems (N, i), not necessarily image-finite, we have $(N, i) \models^a \psi_p$ iff $(\llbracket\ p\ \rrbracket, p) \prec (N, i)$

Proof. We prove this by structural induction on p.

- We have $(N, i) \models^a \psi_{\mathbf{0}}$ iff (there are no R^c-transitions whatsoever out of i in N) iff $(\llbracket\ \mathbf{0}\ \rrbracket, \mathbf{0}) \prec (N, i)$.
- We have $(N, i) \models^a tt$ and $(\llbracket\ \perp\ \rrbracket, \perp) \prec (N, i)$ for all pointed modal transition system (N, i).
- Using induction on p, we have $(N, i) \models^a \psi_{\alpha_{tt}.p}$ iff (there is a R^a-transition (i, α, i') in N with $(\llbracket\ p\ \rrbracket, p) \prec (N, i')$; all R^c-transitions (i, α, i'') in N satisfy $(\llbracket\ p\ \rrbracket, p) \prec (N, i'')$; and there are no R^c-transitions out of i in N for other events). This exactly captures $(\llbracket\ \alpha_{tt}.p\ \rrbracket, \alpha_{tt}.p) \prec (N, i)$.

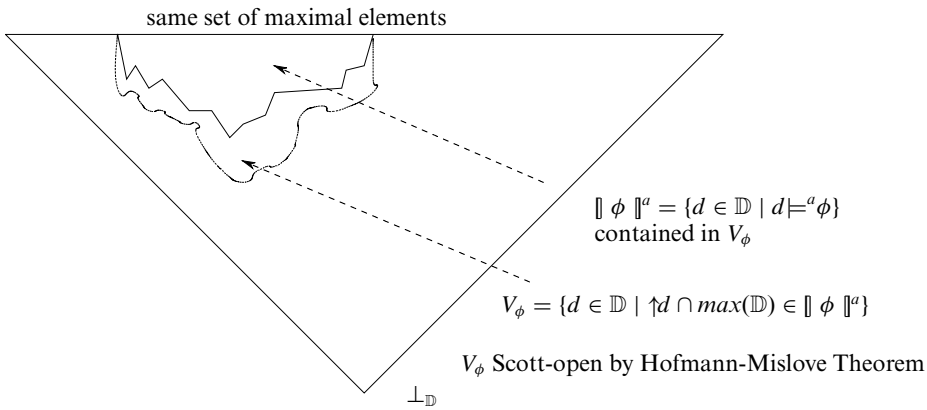

Fig. 12. A schematic description of the potential loss of precision of $\models^a$ over $\models^{a+}$ represented by $V_\phi \setminus [\![\phi]\!]^a$. The properties of $[\![\phi]\!]^a$ and V_ϕ stated in this figure are shown subsequently, except for $V_\phi \cap max(\mathbb{D}) = [\![\phi]\!]^a \cap max(\mathbb{D})$ which is obvious

- By induction on p, we have $(N, i) \models^a \psi_{\alpha_\perp . p}$ iff (there are no R^c-transitions out of i for events other than α, and all R^c-transitions (i, α, i') satisfy $([\![p]\!], p) \prec (N, i')$). But this captures $([\![\alpha_\perp . p]\!], \alpha_\perp . p) \prec (N, i)$.
- Let $(N, i) \models^a \psi_{p+q}$. Then $(N, i) \models^a \bigwedge_{\alpha \in Act} \bigwedge_{p+q \longrightarrow^\alpha_a r'} \langle \alpha \rangle \psi_{r'}$ expresses that all R^a-transitions out of $p+q$ can be answered in the game $\mathcal{G}[([\![p+q]\!], p+q), (N, i)]$ such that a "winning" position for the verifier is reached by induction; whereas $(N, i) \models^a \bigwedge_{\alpha \in Act} [\alpha] \big(\bigvee \{ \psi_{r'} \mid \exists v \in \{\perp, tt\} \colon p+q \longrightarrow^\alpha_v r' \} \big)$ states that all initial questions asked as R^c-transitions from (N, i) can be answered in $([\![p+q]\!], p+q)$ to reach a "winning" position for the verifier as well, by induction. So (N, i) refines $([\![p+q]\!], p+q)$.

□

5. Abstract witnesses

Before we can prove completeness of refinement we need to understand the topology of abstraction better. In abstraction-based model checking for modal transition systems we would like a finite-model property for abstractions: if ϕ holds in some pointed modal transition system (M, i), there should be some finite-state (N, j) such that $(N, j) \prec (M, i)$ and ϕ holds in (N, j) as well. If this were not the case, model checking ϕ for an infinite-state M through finite-state abstractions would be futile. This cannot always be secured. As Dams & Namjoshi [DN04] point out, abstraction-based model checking of the modal mu-calculus is incomplete for finite-state modal transition systems as abstractions. In this section, we prove that such a finite-model property holds for partial modal transition trees as finite abstractions of modal transition systems and Hennessy-Milner logic in the weak *and* in the strong semantics. For the weak semantics this is perhaps rather obvious but we strengthen it to securing that there are finitely many maximally abstract modal transition systems satisfying ϕ. This stronger result for the weak semantics is then used to prove the result for the stronger semantics. We define the concepts of interest via subsets of $\mathbb{D}$.

Definition 5.1 Given ϕ of Hennessy-Milner logic, we define

$$\begin{array}{lcl} [\![\phi]\!]^a & = & \{d \in \mathbb{D} \mid (\mathcal{D}, d) \models^a \phi\} \\ [\![\phi]\!]^c & = & \{d \in \mathbb{D} \mid (\mathcal{D}, d) \models^c \phi\} \\ V_\phi & = & \{d \in \mathbb{D} \mid \uparrow d \cap max(\mathbb{D}) \subseteq [\![\phi]\!]^a\} \,. \end{array} \tag{16}$$

The set V_ϕ has as elements those $d \in \mathbb{D}$ for which all "implementations" satisfy ϕ where "implementations" refers to the interpretation of that notion *in* the model $\mathbb{D}$ and $\mathcal{D}$. Soundness of $\models^a$ with respect to $\models^{a+}$ therefore recognizes V_ϕ as a superset of $[\![\phi]\!]^a$. See Fig. 12. For all pointed modal transition system (N, i) and ϕ of Hennessy-Milner logic we have

$$(N, i) \models^m \phi \;\text{ iff }\; \langle\!\langle N, i \rangle\!\rangle \in [\![\phi]\!]^m \quad (m \in \{a, c\}) \tag{17}$$

by item 6 of Fact 4.1. Below we show this correspondence for the strong semantics and so V_ϕ faithfully models implementations and the strong semantics of ϕ. The finite-model property for the weak semantics can surely

be shown more directly than through arguments based on the model $\mathbb{D}$. But we show a stronger property: the desired finite-state model is an abstraction; and, for each formula of Hennessy-Milner logic, there are finitely many maximally abstract models, which turn out to be partial modal transition trees. To do this, we need to recognize labelled transition systems as elements of $max(\mathbb{D})$, the set of maximal elements of $\mathbb{D}$.

Proposition 5.1 ([Hut04]) For every labelled transition system (L, i), the element $\langle\!| L, i |\!\rangle$ is in $max(\mathbb{D})$.

Now we can establish structural properties of the sets $[\![\phi]\!]^c$, $[\![\phi]\!]^a$, and V_ϕ.

Proposition 5.2 Let ϕ be any formula of Hennessy-Milner logic.

1. The sets $[\![\phi]\!]^a$ and $[\![\phi]\!]^c$ are Lawson-clopen in $\mathbb{D}$.
2. There is a finite set $P_\phi \subseteq MPA$ such that
   - $[\![\phi]\!]^a = \uparrow\{\{\!| p |\!\} \mid p \in P_\phi\}$; and
   - for all pointed modal transition systems (N, i) with $(N, i) \models^a \phi$ there is some p in P_ϕ with $([\![p]\!], p) \prec (N, i)$ and $([\![p]\!], p) \models^a \phi$.
3. The sets $[\![\phi]\!]^a$ and V_ϕ are Scott-open in $\mathbb{D}$ and $[\![\phi]\!]^a \subseteq V_\phi$.
4. For all pointed modal transition systems (N, i), we have

$$(N, i) \models^{a+} \phi \quad \text{iff} \quad \langle\!| N, i |\!\rangle \in V_\phi \,. \tag{18}$$

The proof for item 1 above was already given in Lemma 2 of [Hut04]. We obtain a first finite-model property.

Corollary 5.1 For every ϕ of Hennessy-Milner logic and pointed modal transition system (N, i): If we have $(N, i) \models^{a+} \phi$, there is some $p \in MPA$ such that $([\![p]\!], p) \prec (N, i)$ and $([\![p]\!], p) \models^{a+} \phi$.

Proof. If $(N, i) \models^{a+} \phi$, then $\langle\!| N, i |\!\rangle$ is contained in V_ϕ by item 4 of Proposition 5.2. The claim then follows from (15) and item 3 of Proposition 5.2 since $\mathbb{D}$ is algebraic. □

Example 5.1 1. Unfortunately,

$$\downarrow\mathbf{K}(\mathbb{D}) \not\subseteq \mathbf{K}(\mathbb{D}) \tag{19}$$

so we can't offer an easy proof that the $p \in MPA$ in Corollary 5.1 may be chosen to be *maximally* abstract: consider the pointed modal transition system $(M, i) = (\{i\}, \{\}, \{(i, \alpha, i)\})$ with $Act = \{\alpha, \beta\}$. In M there are no must-transitions and one may-transition $(i, \alpha, i) \in R^c$ only. Then $\{\!| \mathbf{0} |\!\}$ is in $max(\mathbb{D}) \cap \mathbf{K}(\mathbb{D})$ and $\langle\!| M, i |\!\rangle \leqslant \{\!| \mathbf{0} |\!\}$ but $\langle\!| M, i |\!\rangle$ is not compact as it does not equal any of its finite approximations $\langle\!| M[m], i |\!\rangle$ with $m \geqslant 1$, for may-leaves in each $M[m]$ contain R^c-transitions for event β.

2. There are even infinite strictly descending chains $(l_n)_{n \geqslant 1}$ in $\mathbf{K}(\mathbb{D})$. For each $n \geqslant 1$ let $p_n \in MPA$ be the term $\alpha_\perp.\alpha_\perp. \ldots .\alpha_\perp.\mathbf{0}$ which nests n may-prefixes for the same event α and then stops. Each $l_n = \{\!| p_n |\!\}$ is compact in $\mathbf{K}(\mathbb{D})$ and for all $n < m$ we have $l_m \leqslant l_n$ and $l_n \neq l_m$.

6. Proving completeness of refinement

Now we can use the game-theoretic interpretation of refinement and our results from Sect. 5 to show that refinement is complete. As the proof of Theorem 6.1 is rather involved we first give an informal outline of strategy for this proof. In the next proposition we show that refinement is complete for partial modal transition trees as models. Then we show that completeness of refinement holds iff the domain $\mathbb{D}$ satisfies that its order $d \leqslant e$ is equivalent to $\uparrow e \cap max(\mathbb{D}) \subseteq \uparrow d \cap max(\mathbb{D})$. Next we use topology, notably the Scott-openness of V_ϕ, to argue that the equivalence of $d \leqslant e$ and $\uparrow e \cap max(\mathbb{D}) \subseteq \uparrow d \cap max(\mathbb{D})$ is true for all $d, e \in \mathbb{D}$ iff it is true for d and e ranging over $\mathbf{K}(\mathbb{D})$ only. Finally, we note that this property restricted to elements of $\mathbf{K}(\mathbb{D})$ is ensured by the completeness of refinement for partial modal transition trees.

Proposition 6.1 For all partial modal transition trees (M, i) and (M', i') we have $(M, i) \prec (M', i')$ iff $\mathcal{I}[M', i'] \subseteq \mathcal{I}[M, i]$.

Proof. Since pointed modal transition trees are denotations of terms in MPA and since $(M, i) \prec (N, j)$ implies $\mathcal{I}[N, j] \subseteq \mathcal{I}[M, i]$, it suffices to show

$$\text{“For all } p, q \in MPA \text{ the relation } \mathcal{I}[[\![q]\!], q] \subseteq \mathcal{I}[[\![p]\!], p] \text{ implies } ([\![p]\!], p) \prec ([\![q]\!], q)\text{.”} \tag{20}$$

which we do by induction on $deg(q)$, the number of occurrences of prefixes $\gamma_\perp.$ in q summed up over all $\gamma \in Act$. For example for the q from Fig. 7 we have $deg(q) = 3$, stemming from the two occurrences of $\alpha_\perp.$ and one occurrence of $\beta_\perp$.

Base case: Let $deg(q) = 0$. Then the only may-transitions, if any, in $(\!| q |\!|, q)$ stem from may-stubs $\perp$. It is sufficient to show that the verifier has a winning strategy in the game $\mathcal{G}[(\!| p |\!|, p), (\!| q |\!|, q)]$. The argument is complex in that we use $\mathcal{I}[\!| q |\!|, q] \subseteq \mathcal{I}[\!| p |\!|, p]$ to synthesize implementations on demand from which we then synthesize the promised winning strategy for the verifier. This is done by a case analysis of the players' behavior in runs. (If there are no may-stubs $\perp$ in $\!| q |\!|$, the arguments below reduce to stating that $(\!| q |\!|, q)$ is its own implementation and therefore an implementation of $(\!| p |\!|, p)$ as well.) Consider any run in the game $\mathcal{G}[(\!| p |\!|, p), (\!| q |\!|, q)]$.

- Suppose that the refuter never chooses a may-transition as a question in said run. Then all her questions are must-transitions in $(\!| p |\!|, p)$ or $(\!| q |\!|, q)$. But then she cannot win this run. For let q_1 be obtained from q by replacing all occurrences of $\perp$ in q with $\mathbf{0}$. Then $(\!| q_1 |\!|, q_1) \in \mathcal{I}[\!| q |\!|, q]$, since $deg(q) = 0$, and $\mathcal{I}[\!| q |\!|, q]$ is contained in $\mathcal{I}[\!| p |\!|, p]$ by assumption. Thus, $(\!| q_1 |\!|, q_1) \in \mathcal{I}[\!| p |\!|, p]$ and so $(\!| p |\!|, p) \prec (\!| q_1 |\!|, q_1)$. Said run can be interpreted as a run in the game $\mathcal{G}[(\!| p |\!|, p), (\!| q_1 |\!|, q_1)]$ and is therefore won by the verifier.

- Suppose that the refuter does choose a may-transition as a question in said run. Let (p', q') be the first game position in that run in which the refuter asks such a question, which has to be a transition in $\!| q |\!|$ derived from $q' \longrightarrow_\perp^\alpha q''$ for some sub-terms q' and q'' of q and some $\alpha \in Act$ as the refuter cannot ask a may-transition in $(\!| p |\!|, p)$. Since $deg(q) = 0$ we infer that q' and q'' are the same sub-term $\perp^1$, where 1 signifies this occurrence of $\perp$.
  From position $(p', \perp^1)$ in game $\mathcal{G}[(\!| p |\!|, p), (\!| q |\!|, q)]$ the refuter can keep asking any R^c-transition questions on the may-stub $\perp^1$ which have to be answered by the verifier with an R^c-transition from p' onwards in $\!| p |\!|$. Therefore, we can construct a winning strategy for the verifier by showing that *all* such answers can be given and can be chosen to reach a position $(p'', \perp^1)$ such that there are no R^a-transitions out of p''. This argument is inductive in the length of the path from p' to p'', including the path of length zero, will trap the refuter in the situation of asking R^c-transitions as questions in $\!| q |\!|$, and results in an infinite run won by the verifier. We write $\sum$ for the $n$ary version of non-deterministic choice + in MPA.

  - Let the length of the path from p' to p'' be zero. We need to show that there are no R^a-transitions out of p' in $\!| p |\!|$. Let $\alpha \in Act$. Consider the implementation $(\!| q_2 |\!|, q_2)$ of $(\!| q |\!|, q)$ which replaces the occurrence $\perp^1$ in q with $(\Sigma_{\beta \neq \alpha} \beta_{\mathit{tt}}.\mathbf{0})^1$, where 1 still indicates the unique occurrence within the parse tree of q_2, and replaces all other occurrences of $\perp$ in q with $\mathbf{0}$. Note that our run in the game $\mathcal{G}[(\!| p |\!|, p), (\!| q |\!|, q)]$ up to position $(p', \perp^1)$ can be interpreted as a run in the game $\mathcal{G}[(\!| p |\!|, p), (\!| q_2 |\!|, q_2)]$ as no may-transition has been asked prior to that position. But all implementations of $(\!| q |\!|, q)$ are also implementations of $(\!| p |\!|, p)$ by assumption. So $(\!| p |\!|, p) \prec (\!| q_2 |\!|, q_2)$ follows and the verifier has a winning strategy in that game. This means that in position $(p', (\Sigma_{\beta \neq \alpha} \beta_{\mathit{tt}}.\mathbf{0})^1)$ the refuter cannot ask a R^a-transition labelled with α out of p' as there are no such transitions out of $(\Sigma_{\beta \neq \alpha} \beta_{\mathit{tt}}.\mathbf{0})^1$. Since $\alpha \in Act$ was arbitrary we conclude that there are no R^a-transitions out of p' in $\!| p |\!|$.
  - Let $\alpha^1 \ldots \alpha^n$ be the entire sequence of events corresponding to the sequence of R^c-transition questions asked by the refuter from position $(p', \perp^1)$ towards position $(p'', \perp^1)$ in said run of the game $\mathcal{G}[(\!| p |\!|, p), (\!| q |\!|, q)]$. Let $\alpha \in Act$. Consider the implementation $(\!| q_3 |\!|, q_3)$ of $(\!| q |\!|, q)$ which replaces the occurrence $\perp^1$ in q with $(\alpha^1_{\mathit{tt}}.\alpha^2_{\mathit{tt}}. \ldots \alpha^n_{\mathit{tt}}.\Sigma_{\beta \neq \alpha} \beta_{\mathit{tt}}.\mathbf{0})^1$ and replaces all other occurrences of $\perp$ in q with $\mathbf{0}$. Our run in the game $\mathcal{G}[(\!| p |\!|, p), (\!| q |\!|, q)]$ up to the position $(p'', \perp^1)$ is interpretable as a run in the game $\mathcal{G}[(\!| p |\!|, p), (\!| q_3 |\!|, q_3)]$ up to position $(p'', (\Sigma_{\beta \neq \alpha} \beta_{\mathit{tt}}.\mathbf{0})^1)$ since $R^a \subseteq R^c$. We may now reason in the same manner as done for the path of length zero.

Inductive step: Let the statement (20) be true for all $p, q \in MPA$ with $deg(q) < n$. Now let $0 < deg(q) = n$ with $\mathcal{I}[(\!| q |\!|, q)] \subseteq \mathcal{I}[(\!| p |\!|, p)]$. Since $deg(q) > 0$ there is some $\alpha \in Act$ with some sub-term $\alpha_\perp.q'$ in q. Define q^+ by replacing that sub-term in q with $\alpha_{\mathit{tt}}.q'$ and let q^- be obtained by replacing $\alpha_\perp.q'$ in q with $\mathbf{0}$. These are *linear* replacements, only that one occurrence of $\alpha_\perp.q'$ is being replaced if $\alpha_\perp.q'$ occurs more than once in q. It is immediate that $(\!| q^+ |\!|, q^+)$ and $(\!| q^- |\!|, q^-)$ refine $(\!| q |\!|, q)$ and that $deg(q^+)$ and $deg(q^-)$ are less than n. But $(\!| q |\!|, q) \prec (\!| q^+ |\!|, q^+)$ and $\mathcal{I}[\!| q |\!|, q] \subseteq \mathcal{I}[\!| p |\!|, p]$ imply $\mathcal{I}[\!| q^+ |\!|, q^+] \subseteq \mathcal{I}[\!| p |\!|, p]$ as refinement is transitive. Similarly, we establish that $\mathcal{I}[\!| q^- |\!|, q^-] \subseteq \mathcal{I}[\!| p |\!|, p]$. By induction on $deg(q^+)$ and $deg(q^-)$ we infer $(\!| p |\!|, p) \prec (\!| q^+ |\!|, q^+)$ and $(\!| p |\!|, p) \prec (\!| q^- |\!|, q^-)$, respectively.

But then there exist some w^+ and w^-, respective winning strategies for the verifier in the two refinement games $\mathcal{G}[(\llbracket\ p\ \rrbracket, p), (\llbracket\ q^+\ \rrbracket, q^+)]$ and $\mathcal{G}[(\llbracket\ p\ \rrbracket, p), (\llbracket\ q^-\ \rrbracket, q^-)]$. We synthesize from w^+ and w^- a winning strategy for the verifier in the game $\mathcal{G}[(\llbracket\ p\ \rrbracket, p), (\llbracket\ q\ \rrbracket, q)]$ as follows.

- As long as questions in the latter game are also questions in the game $\mathcal{G}[(\llbracket\ p\ \rrbracket, p), (\llbracket\ q^-\ \rrbracket, q^-)]$ the verifier answers according to w^- and these questions and answers will be legitimate questions and answers in the game $\mathcal{G}[(\llbracket\ p\ \rrbracket, p), (\llbracket\ q\ \rrbracket, q)]$ since all transitions in $\llbracket\ q^-\ \rrbracket$ are also in $\llbracket\ q\ \rrbracket$ and in the same mode.
- As soon as the refuter asks a question that is not in the game $\mathcal{G}[(\llbracket\ p\ \rrbracket, p), (\llbracket\ q^-\ \rrbracket, q^-)]$, that question has to be the may-transition stemming from the prefix $\alpha_\perp$. which is implemented in q^+ as a must-transition. In particular, the position at which this question is raised is reachable from (p, q^+) in the game $\mathcal{G}[(\llbracket\ p\ \rrbracket, p), (\llbracket\ q^+\ \rrbracket, q^+)]$. Hence the verifier uses $R^a \subseteq R^c$ to interpret that question as an R^c-transition in the game $\mathcal{G}[(\llbracket\ p\ \rrbracket, p), (\llbracket\ q^+\ \rrbracket, q^+)]$ and replies with the answer according to w^+. But this is a legitimate answer in the game $\mathcal{G}[(\llbracket\ p\ \rrbracket, p), (\llbracket\ q\ \rrbracket, q)]$, resulting in a position of that game from which onwards the refuter can only reach positions and ask questions that are also positions and questions in the game $\mathcal{G}[(\llbracket\ p\ \rrbracket, p), (\llbracket\ q^+\ \rrbracket, q^+)]$, respectively.

So regardless of whether the shift from w^- to w^+ ever happens, the verifier wins every run and therefore $(\llbracket\ p\ \rrbracket, p) \prec (\llbracket\ q\ \rrbracket, q)$ follows. □

As stated in the outline of our proof strategy, we need to show that $\uparrow\!\!\langle M,', i' \rangle \cap max(\mathbb{D}) \subseteq \uparrow\!\!\langle M, i \rangle \cap max(\mathbb{D})$ captures the relation $\mathcal{I}[M', i'] \subseteq \mathcal{I}[M, i]$ for all pointed modal transition systems (M, i) and (M', i').

Lemma 6.1 For all pointed modal transition systems (M, i) and (M', i') we have $\mathcal{I}[M', i'] \subseteq \mathcal{I}[M, i]$ iff $\uparrow\!\!\langle M,', i' \rangle \cap max(\mathbb{D}) \subseteq \uparrow\!\!\langle M, i \rangle \cap max(\mathbb{D})$.

In accordance with our overall strategy we now reveal that the domain is complete for refinements iff this is true for its compact elements already.

Lemma 6.2 Suppose that for all compact elements $k, l \in \mathbf{K}(\mathbb{D})$ we have that $\uparrow l \cap max(\mathbb{D}) \subseteq \uparrow k \cap max(\mathbb{D})$ implies $k \leqslant l$. Then this implication holds for all elements of $\mathbb{D}$: for all $d, e \in \mathbb{D}$ the relation $\uparrow e \cap max(\mathbb{D}) \subseteq \uparrow d \cap max(\mathbb{D})$ implies $d \leqslant e$.

Proof. Let $d, e \in \mathbb{D}$ with $\uparrow e \cap max(\mathbb{D}) \subseteq \uparrow d \cap max(\mathbb{D})$. Choose any $k \in \mathbf{K}(\mathbb{D})$ below d. As $\mathbb{D}$ is algebraic, it suffices to show $k \leqslant e$. By (15) there is $p \in MPA$ with $k = \{\!|\ p\ |\!\}$ and so $e \in V_{\psi_p}$ by Lemma 4.3 since $\uparrow e \cap max(\mathbb{D}) \subseteq \uparrow d \subseteq \uparrow k$, where $\uparrow d \subseteq \uparrow k$ since $k \leqslant d$. But $\mathbb{D}$ is algebraic and the set V_{ψ_p} is Scott-open by item 3 of Proposition 5.2. So there is some $l \in \mathbf{K}(\mathbb{D})$ below e with $l \in V_{\psi_p}$. Thus, $\uparrow l \cap max(\mathbb{D}) \subseteq \llbracket\ \psi_p\ \rrbracket^a \cap max(\mathbb{D}) \subseteq \uparrow k \cap max(\mathbb{D})$ which implies $k \leqslant l$ by the assumption of the lemma. So $k \leqslant l \leqslant e$ renders $k \leqslant e$. □

Finally, we can tie together all arguments to secure the completeness of refinement.

Theorem 6.1 Refinement for pointed modal transition systems is complete with respect to the interpretation of pointed modal transition systems as their respective classes of implementations: for all pointed modal transition systems (M, i) and (N, j) we have $(M, i) \prec (N, j)$ iff $\mathcal{I}[N, j] \subseteq \mathcal{I}[M, i]$.

Proof. From Lemma 6.1 we know that the completeness of refinement holds iff for all $d, e \in \mathbb{D}$ the relation $\uparrow e \cap max(\mathbb{D}) \subseteq \uparrow d \cap max(\mathbb{D})$ implies $d \leqslant e$. By Proposition 6.1, Lemma 6.1 applied to partial modal transition trees, and (15) this holds for all compact elements $d, e \in \mathbf{K}(\mathbb{D})$. By Lemma 6.2, this is sufficient. □

Using the completeness of refinement, we show that refinement is characterized by Hennessy-Milner logic not only under the weak, but also under the *strong* semantics.

Corollary 6.1 For all pointed modal transition systems (M, i) and (N, j) we have $(M, i) \prec (N, j)$ iff (for all ϕ of Hennessy-Milner logic, $(M, i) \models^{a+} \phi$ implies $(N, j) \models^{a+} \phi$) iff (for all ϕ of Hennessy-Milner logic, $(N, j) \models^{c-} \phi$ implies $(M, i) \models^{c-} \phi$).

Using the results of Godefroid & Jagadeesan in [GJ03], we infer that refinement is complete for a host of 3-valued models and their notion of refinement, including those that combine event and state information.

Corollary 6.2 Refinement is complete for implementations for the model-checking frameworks of Kripke modal transition systems [HJS01] and partial Kripke structures [BG99].

Proof. Godefroid & Jagadeesan show [GJ03] that there are linear-time and log-space translations between any of these models and modal transition systems, and between Hennessy-Milner logic and the corresponding temporal logics for the other models such that refinement and the meaning of model checks is preserved and reflected. Thus our arguments are invariant under such a change of representation. □

7. Completeness of refinement as semantic minimization

This paper asks as second question whether a semantic minimization $\phi \mapsto \phi^+$ exists for modal transition systems and Hennessy-Milner logic:

$$\text{“For all } \phi \text{ of Hennessy-Milner logic, is there some } \phi^+ \text{ of Hennessy-Milner logic such that for all pointed modal transition systems } (M,i)\text{: } ((M,i)\models^{a+}\phi \text{ iff } (M,i)\models^{a}\phi^+)\text{?”} \tag{21}$$

As Blamey has shown [Bla80], the answer is affirmative in the setting of 3-valued propositional logic where $\models^a$ is Kleene's strong 3-valued interpretation of propositional logic, our weak semantics, and $\models^{a+}$ the super-valuational meaning [vF66], our strong semantics. As one would expect for propositional logic, the length of ϕ^+ is exponential in the length of ϕ in the worst case.

Example 7.1 Let the formula ϕ be $p \leftrightarrow q$, which states that p and q have the same truth value. Then ϕ is semantically self-minimizing in that we may choose ϕ^+ to be ϕ. For if M is any 3-valued model for which ϕ evaluates to $\perp$ in the weak semantics, there will always be two refining 2-valued models for which ϕ evaluates to different truth values.

The connection between semantic minimization and completeness of refinement can now be explained. In Lemma 4.3 we showed that for all formulas ψ_p of Fig. 11 and for the operational meaning $(\llbracket\ p\ \rrbracket, p)$ of the process term p defined in Fig. 6 we have

$$(\llbracket\ p\ \rrbracket, p) \prec (M,i) \text{ iff } (M,i)\models^{a}\psi_p \qquad ((M,i) \text{ pointed modal transition system}). \tag{22}$$

Moreover, the set of all such ψ_p logically characterizes refinement by (15) and items 1 and 5 of Fact 4.1 in the weak semantics. We introduce terminology for the case in which $\llbracket\ \phi\ \rrbracket^a$ in Fig. 12 is a proper subset of V_ϕ.

Definition 7.1 A formula ϕ of Hennessy-Milner logic *loses precision* iff (for some pointed modal transition system (M,i), $(M,i)\models^{a+}\phi$ but $(M,i)\not\models^{a}\phi$).

Remark 7.1 The completeness of refinement is equivalent to the fact that no formula ψ_p loses precision. To see this we repeatedly use (22), Lemma 4.3, and the fact that $\models^a$ equals `sat` on implementations:

- Assume that some ψ_p loses precision. Then there is some (M,i) with $(M,i)\models^{a+}\psi_p$ and $(M,i)\not\models^{a}\psi_p$. The former guarantees $\mathcal{I}[M,i] \subseteq \mathcal{I}[\llbracket\ p\ \rrbracket, p]$, the latter means $(\llbracket\ p\ \rrbracket, p) \not\prec (M,i)$, and so refinement is incomplete.
- Conversely, assume that no ψ_p loses precision. Proof by contradiction: Assume that $\mathcal{I}[M,i] \subseteq \mathcal{I}[N,j]$ and $(N,j) \not\prec (M,i)$. By items 1 and 6 of Fact 4.1 and (15) there is some p with $(\llbracket\ p\ \rrbracket, p) \prec (N,j)$ such that $(\llbracket\ p\ \rrbracket, p) \not\prec (M,i)$. From $\mathcal{I}[M,i] \subseteq \mathcal{I}[N,j)]$ and $(\llbracket\ p\ \rrbracket, p) \prec (N,j)$ we get $(M,i)\models^{a+}\psi_p$ as valid $\models^a$ checks are preserved under refinement by item 5 of Fact 4.1. Thus, $(M,i)\models^{a}\psi_p$ as well as ψ_p does not lose precision. But $(M,i)\models^{a}\psi_p$ means $(\llbracket\ p\ \rrbracket, p) \prec (M,i)$, a contradiction.

That is, completeness of refinement is equivalent to the fact that ψ_p^+ of (21) exists for all formulas ψ_p of Fig. 11 and that ψ_p^+ may be chosen to be ψ_p; these formulas are *semantically self-minimizing* and so validity checks $(M,i)\models^{a+}\psi_p$ are reducible to model checks $(M,i)\models^{a}\psi_p$. This insight corroborates that answering question (4) is non-trivial since the semantics of ψ_{p+q} under $\models^a$ depend heavily on the evaluation of disjunctions, potentially causing a loss of precision.

8. Checking consistency and verifying goals of multiple models

Before we conclude, we point out a genuine application of our results in the realm of requirements engineering. Basic functional requirements may be captured as a term p of a process algebra, say, *MPA* for sake of simplicity. In fact the process algebra *MPA*, enriched with parallel composition, is sufficient for activities such as state-space exploration, simulation, and bounded model checking [BCCZ99] where one puts bounds on the depth of computation paths.

Fig. 13. Two modal transition systems with initial states s_1 and s_2, respectively, that have a common refinement

As requirements are likely to be under-specified, prefixes come in two modes, $\alpha_{\mathbb{T}}.$ and $\alpha_{\perp}.$, as in the process algebra *MPA*. A considerable problem in practice is that there are often finitely many such terms $p_1, p_2, \ldots, p_n$ each describing requirements on, or aspects of, the *same* design or product. This raises fundamental problems:

1. Events known in one term may be foreign in another.
2. Does the finite set of terms $\{p_k \mid 1 \leqslant k \leqslant n\}$ have a *common* implementation?
3. If so, do *all* (respectively *some*) common implementations of that set satisfy a goal?

Example 8.1 Figure 13 depicts two modal transition systems that have a common refinement. The reader is invited to determine whether all such common refinements satisfy that there is an event path on which β events happen until one gets to a state at which not all α events lead to states at which β can happen.

The first problem could be solved by identifying and encoding all foreign events as may-transitions that can lead from states to subsets of states with that imported foreign label, creating a common set of events for all terms p_k. Two obvious choices of such subsets are the entire state space, modelling state-wide divergence, and singleton self-loops, modelling that local state is unaffected by foreign events.

The second problem may now be solved as follows. For each $1 \leqslant k \leqslant n$ let $(\llbracket\ p_k\ \rrbracket, p_k)$ be the pointed modal transition system that arises as the operational meaning of the term p_k. We assume that each term p has a formula ψ_p of some logic, not necessarily Hennessy-Milner logic or the modal mu-calculus, satisfying (22) for a judgment $\models^a$ that interprets conjunction in a compositional manner. A common implementation is a pointed labelled transition system that refines all of these pointed modal transition systems, therefore being a witness to the satisfiability of

$$\bigwedge_{k=1}^{n} \psi_{p_k}\ . \tag{23}$$

Conversely, any labelled transition system that is a satisfiability witness of that formula has to be a common implementation of all of the $(\llbracket\ p_k\ \rrbracket, p_k)$ by (22). For *MPA*, the formulas in (23) can be computed inductively as in Fig. 11 and satisfiability for Hennessy-Milner logic and labelled transition systems is PSPACE-complete, for this is the case for the basic modal logic K and Kripke structures and there are linear-time and log-space translations between these frameworks [GJ03]. These results are modest since practical specifications often require recursion and so the ψ_{p_k} in (23–27) need to be replaced with the $X_{(\llbracket p_k \rrbracket, p_k)}$ in (13) which are then expressible in the modal mu-calculus if $\llbracket\ p_k\ \rrbracket$ is finite-state.

For *MPA*, the third problem above can be specified in terms of $\models^a$ and $\models^{a+}$ by exploiting the topological structure of the *SFP*-domain $\mathbb{D}$. Each $\llbracket\ \psi_{p_k}\ \rrbracket^a$ is of the form $\uparrow C_k$ for a finite set $C_k \subseteq \mathbf{K}(\mathbb{D})$ by Proposition 5.2. Since the finite intersection of Scott-open, Scott-compact sets is again Scott-compact and Scott-open in a *SFP*-domain [AJ94] we conclude that

$$\bigcap_{k=1}^{n} \llbracket\ \psi_{p_k}\ \rrbracket^a = \uparrow\{\{\!|\ q\ |\!\}\} \mid q \in Q\} \tag{24}$$

for some finite set $Q \subseteq MPA$. The set Q is non-empty iff the first problem has a positive solution. In that case, each $(\llbracket\ q\ \rrbracket, q)$ with $q \in Q$ is a maximally abstract common refinement of all $(\llbracket\ p_k\ \rrbracket, p_k)$, up to refinement equivalence, and every common refinement of all $(\llbracket\ p_k\ \rrbracket, p_k)$ is a refinement of some $(\llbracket\ q\ \rrbracket, q)$ with $q \in Q$. Therefore, we can answer whether all common refinements satisfy ϕ by answering

$$\text{“Does } (\llbracket\, q\, \rrbracket, q) \models^{a+} \phi \text{ hold for all } q \in Q\text{?”} \tag{25}$$

A computationally less expensive approximation uses $\models^{a}$ in (25) instead of $\models^{a+}$.

No doubt has the reader realized that this constitutes a solution only in as much as one is able to compute all $q \in Q$ from the set of all $(\llbracket\, p_k\, \rrbracket, p_k)$. In the general case, we may have to rely on the validity of (22) to decide whether *all* common implementations of all p_k satisfy ϕ by checking the satisfiability of

$$\neg\phi \wedge \bigwedge_{k=1}^{n} \psi_{p_k} \tag{26}$$

over labelled transition systems and negating the answer of that check. Dually, in asking whether *some* common implementation of all p_k satisfies ϕ, we check whether

$$\phi \wedge \bigwedge_{k=1}^{n} \psi_{p_k} \tag{27}$$

is satisfiable by some labelled transition system. The check for consistency in (23) is a special case of (27) where ϕ equals tt and consistency checks should, intuitively, be in PTIME and not be PSPACE-complete or EXPTIME-complete as is the case for checks of general Hennessy-Milner logic or modal mu-calculus formulas, respectively.

All this is modest but promising progress in a longstanding open problem in formal software engineering.[1]

9. Related work

Dams & Namjoshi [DN04] show that finite-state modal transition systems are incomplete as abstractions of infinite-state modal transition systems for modal mu-calculus checking. They propose focused transition systems as a generalization of modal transition systems, show completeness for this class of models, and define game semantics for refinement of such systems and model checks of alternating tree automata on such systems.

In [Hut04] further structural properties of refinement are shown, notably, that $max(\mathbb{D})$ is Lawson-closed in $\mathbb{D}$, that $\mathbb{X} = max(\mathbb{D})$ is a Stone space where the topology has basis $\{\uparrow k \cap max(\mathbb{D}) \mid k \in \mathbf{K}(\mathbb{D})\}$, and that the space $\mathbb{X}$ is a topological model of all labelled transition systems over a finite set of events Act up to bisimulation such that the embeddings of image-finite labelled transition systems are dense in $\mathbb{X}$. Consistency measures for modal transition systems are introduced and discussed. The journal version of that paper [Hut05] also presents a Galois adjunction between compact sets of implementations and Scott-closed sets of modal transition systems.

The paper [HJS04] presents the *SFP*-domain $\mathbb{D}$ and its underlying modal transition system and provides most of the basic facts on which the work in this paper relies. Although we have strived to make this paper self-contained, we recommend reading [HJS04].

Uchitel & Chechik [UC04] merge modal transition systems with overlapping but different sets of events to obtain a minimal common refinement and suggest user participation to explore common behavior if no minimal common refinement exists.

The work proposed in Sect. 8 was continued in [HH04] which determined a polynomial-time algorithm for checking whether multiple models have a common implementation. In loc. cit. summary models for diagnostic purposes similar to those of [UC04] were also being defined.

Bruns & Godefroid develop 3-valued model checking in [BG99]; their generalized model checking in [BG00] eradicates any loss of precision through automata-theoretic means that blow up the model to be checked. They offer model-checking algorithms and complexity bounds for most practically relevant temporal logics. The abstraction-based approach to 3-valued model checking is described in [GHJ01] and, by Godefroid & Jagadeesan, in [GJ02].

Dams [Dam96] develops an abstract-interpretation and partition-refinement framework for model checking based on mixed transition systems which are often not modal transition systems since optimality considerations suggest to construct as few may-transitions and as many must-transitions as possible. It is unknown whether the models in [Dam96] always satisfy the mix condition (MC) if they abstract concrete models.

The proof that semantic minimization, in the sense of (21), is possible for all formulas of propositional logic is contained in Blamey's thesis [Bla80], Theorem I.3.3. An implementation of such minimizations, using prime implicants and binary decision diagrams, was given by Reps et al. in [RLS02].

[1] The ideas proposed in this section lead subsequently to the work in [HH04], to which we refer to interested reader.

The definition of modal transition systems and their refinement is given by Larsen & Thomsen in [LT88]. Larsen defines a semantics for Hennessy-Milner logic over modal transition systems and shows that it logically characterizes refinement in [Lar89]; in loc. cit. he also proposes modal specifications as a way of combining modal transition systems declaratively.

Cousot & Cousot invent abstract interpretation [CC77] as a formal framework in which one can express abstractions of concrete data and transformations, and formulate soundness and optimality principles for abstract interpretations of concrete transformations. In that context, non-distributive flow analyzes [NNH99] lose precision in a way similar to the loss encountered by the weak semantics of Hennessy-Milner logic.

Van Fraassen defines the super-valuational meaning of propositional logic formulas in [vF66].

10. Conclusions

We presented the 3-valued model-checking framework for modal transition systems, their refinement, and a weak and a strong semantics for Hennessy-Milner logic. The weak semantics, a bottom-up labelling algorithm for model checks, is well known to lose precision. The strong semantics does not lose precision but known algorithms require the transformation of the model under check. We asked whether one model refines another if *and only if* its implementations are also implementations of the system it is refining. We answered this affirmatively and thus showed that Hennessy-Milner logic characterizes refinement under the strong semantics. This also means that the characteristic formulas of partial modal transition trees do not lose precision when checked under the weak semantics.

The proofs of these results relied in part on a topological model developed with Jagadeesan & Schmidt in [HJS04], a *SFP*-domain which is also a universal modal transition system. Using this model, we furthermore secured that all model checks can be decided by model checks on partial modal transition trees that abstract the model under check; and that, in the weak semantics, there are finitely many models – which happen to be partial modal transition trees – that are maximally abstract with respect to satisfying a given formula of Hennessy-Milner logic. For the strong semantics we proved that each model that satisfies a formula of Hennessy-Milner logic is abstracted by a partial modal transition tree that satisfies that formula. These results constitute an abstraction-based finite-model property for the weak and the strong semantics.

We presented some preliminary results on applications of this work in the context of determining whether finitely many pointed modal transition systems have a common refinement and, if so, whether all common refinements satisfy a goal.

Finally, we remarked that the results of this paper are stable under a change of representation as they apply to 3-valued models that are state-based or combine state and event information.

Acknowledgements

We wish to thank Samson Abramsky, Michael Goldsmith, Chris Hankin, Achim Jung, Bill Roscoe, and especially Glenn Bruns, Patrice Godefroid, Radha Jagadeesan, and David Schmidt for their insight and feedback. Discussions with Sebastian Uchitel triggered the development in Sect. 8. We also thank the anonymous referees for making many suggestions on improving the organization of this paper and the clarity of its presentation.

References

[ABH97] Alessi F, Baldan P, Honsell F (1997) Partializing Stone spaces using SFP domains. In: Bidoit M, Dauchet M, (eds), TAPSOFT'97 conference proceedings, Lille, France, 14–18. lecture notes in computer science, vol. 1214. Springer, Berlin Heidelberg New York, pp 478–489

[AJ94] Abramsky S, Jung A (1994) Domain theory. In: Abramsky S, Gabbay DM, Maibaum TSE, (eds), Handbook of Logic in Computer Science, vol. 3. Oxford University Press, pp 1–168

[BCCZ99] Biere A, Cimatti A, Clarke E, Zhu Y (1999) Symbolic model checking without BDDs. In: Proceedings of tools and algorithms for the analysis and construction of systems. lecture notes in computer science, vol. 1579 pp 193–207

[BG99] Bruns G, Godefroid P (1999) Model checking partial state spaces with 3-valued temporal logics. In: Proceedings of the 11th conference on computer aided verification. lecture notes in computer science, vol. 1633. Springer, Berlin Heidelberg New York, pp 274–287

[BG00] Bruns G, Godefroid P (2000) Generalized model checking: reasoning about partial state spaces. In: Proceedings of the 11th international conference on concurrency theory. lecture notes in computer science, vol. 1877. Springer, Berlin Heidelberg New York, pp 168–182

[Bla80] Blamey S (1980) Partial-valued logic. PhD Thesis, University of Oxford, Oxford

[CC77] Cousot P, Cousot R (1977) Abstract interpretation: a unified lattice model for static analysis of programs. In: Proceedings 4th ACM symposium on principles of programming languages, Los Angeles

[CC00] Cousot P, Cousot R (2000) Temporal abstract interpretation. In: Conference record of the 27th annual ACM SIGPLAN-SIGACT symposium on principles of programming languages. Boston, MA, January 2000. ACM Press, New York, pp 12–25

[CE81] Clarke EM, Emerson EA (1981) Synthesis of synchronization skeletons for branching time temporal logic. In: Kozen D, (ed), Logic of programs workshop, Yorktown Heights, New York, May 1981. lecture notes in computer science, No. 131. Springer Berlin Heidelberg, New York

[CGL94] Clarke EM, Grumberg O, Long DE (1994) Model checking and abstraction. ACM Trans Program Lang Syst, 16(5):1512–1542

[CN76] Courcelle B, Nivat M (1976) Algebraic families of interpretations. In: Proceedings of the 17th IEEE symposium on foundations of computer science, pp 137–146

[Dam96] Dams D (1996) Abstract interpretation and partition refinement for model checking. PhD Thesis, Technische Universiteit Eindhoven, The Netherlands

[DGG97] Dams D, Gerth R, Grumberg O (1997) Abstract interpretation of reactive systems. ACM Trans Program Lang Syst, 19:253–291

[DN04] Dams D, Namjoshi K (2004) The existence of finite abstractions for branching time model checking. In: Proceedings of the 19th annual IEEE symposium on logic in computer science, Turku, Finland, 13–17 July 2004. IEEE Computer Society Press, pp 335–344

[EJS03] Eisenbach S, Jurisic V, Sadler C (2003) Modeling the evolution of NET programs. In: IFIP international conference on formal methods for open distributed systems, Lecture notes in computer science. Springer, Berlin Heidelberg New York

[FdR03] Franceschet M, de Rijke M (2003) Model checking for hybrid logics. In: Proceedings of the workshop on methods for modalities. INRIA Lorraine, Nancy

[GHJ01] Godefroid P, Huth M, Jagadeesan R (2001) Abstraction-based model checking using modal transition systems. In: Proceedings of the 12th international conference on theory and practice of concurrency. lecture notes in computer science, vol 2154. Springer Berlin Heidelberg New York, pp 426–440

[GJ02] Godefroid P, Jagadeesan R (2002) Automatic abstraction using generalized model checking. In: Brinksma E, Larsen KG, (eds), Proceedings of the 14th international conference on computer aided verification, Copenhagen, Denmark, July 2002. lecture notes in computer science, Springer, Berlin Heidelberg New York, pp 137–150

[GJ03] Godefroid P, Jagadeesan R (2003) On the expressiveness of 3-valued models. In: Zuck LD, Attie PC, Cortesi A, Mukhopadhyay S, (eds), Proceedings of the 4th conference on verification, model checking and abstract interpretation. lecture notes in computer science, vol 2575. Springer, Berlin Heidelberg New York, pp 206–222

[GPVW95] Gerth R, Peled D, Vardi MY, Wolper P (1995) Simple on-the-fly verification of linear temporal logic. In: Protocol specification, testing and verification. Chapman & Hall, London, pp 3–18

[GTWW77] Goguen JA, Thatcher JW, Wagner EG, Wright JB (1977) Initial algebra semantics and continuous algebras. J ACM 24(1):44–67

[Gun92] Gunter C (1992) The mixed power domain. Theoret Comp Sci 103:311–334

[Hec90] Heckmann R (1990) Set domains. In: Proceedings of the 3rd European symposium on programming, Copenhagen, Denmark, 1990. Springer, Berlin Heidelberg New York, pp 177–196

[Hec91] Heckmann R (1991) Power domain constructions. Sci Comp Program, 17(1–3):77–117

[HH04] Hussain A, Huth M (2004) On model checking multiple hybrid views. In: Preliminary proceedings of the 1st international symposium on leveraging applications of formal method, Paphos, Cyprus, 30 October – 2 November 2004. Technical report TR-2004-6 department of computer science, University of Cyprus, pp 235–242

[HJS01] Huth M, Jagadeesan R, Schmidt DA (2001) Modal transition systems: a foundation for three-valued program analysis. In: Sands D. (ed), Proceedings of the European symposium on programming. lecture notes in computer science, vol 2028. Springer, Berlin Heidelberg New York, pp 155–169

[HJS04] Huth M, Jagadeesan R, Schmidt D (2004) A domain equation for refinement of partial systems. Math Struct Comp Sci 14(4):469–505

[HM81] Hofmann KH, Mislove M (1981) Local compactness and continuous lattices. In: Banaschewski B, Hoffmann RE, (eds), Continuous lattices, Bremen, Germany, 1981. lecture notes in mathematics, vol 871. Springer, Berlin Heidelberg New York, pp 209–248

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

[Hut04] Huth M (2004) Beyond image-finiteness: labelled transition systems as a Stone space. In: Proceedings of the 19th annual IEEE symposium on logic in computer science, Turku, Finland, 13–17 July 2004. IEEE Computer Society Press, pp 222–231

[Hut05] Huth M (2005) Labelled transition systems as a Stone space. Log Methods Comp Sci 1(1):1–28. www.lmcs-online.org.

[Kle52] Kleene SC (1952) Introduction to metamathematics. Van Nostrand, Crystal city

[Koz83] Kozen D (1983) Results on the propositional mu-calculus. Theor Comp Sci 27:333–354

[Lar89] Larsen KG (1989) Modal specifications. In: Sifakis J, (ed), Automatic verification methods for finite state systems, Grenoble, France, 12–14 June 1989. No. 407 lecture notes in computer science, Springer, Berlin Heidelberg New York, pp 232–246

[LT88] Larsen KG, Thomsen B (1988) A modal process logic. In: 3rd annual IEEE symposium on logic in computer science. IEEE Computer Society Press, Los Angeles, pp 203–210

[NNH99] Nielson F, Nielson HR, Hankin C (1999) Principles of program analysis. Springer, Berlin Heidelberg New York

[QS81] Quielle JP, Sifakis J (1981) Specification and verification of concurrent systems in CESAR. In: Proceedings of the 5th international symposium on programming

[RLS02] Reps T, Loginov A, Sagiv M (2002) Semantic minimization of 3-valued propositional formulae. In: Proceedings of the 17th annual IEEE symposium on logic in computer science, Copenhagen, Denmark, 22–25 July 2002. IEEE Computer Society Press, pp 40–51

[Sti96] Stirling C (1996) Games and Modal Mu-Calculus. In: Margaria T, Steffen B, (eds), Proceedings of the 2nd international workshop in tools and algorithms for construction and analysis of systems, Passau, Germany, 27–29 March 1996. lecture notes in computer science, vol 1055. Springer, Berlin Heidelberg New York, pp 298–312

[UC04] Uchitel S, Chechik M (2004) Merging partial behavioural models. ACM SIGSOFT Notes 29(6):43–52
[vF66] van Fraassen BC (1966) Singular terms, truth-value gaps and free logic. J Philos 63:481–495

A. Definitions and established results from topology and domain theory

(We recommend [AJ94] for a thorough reference on these issues.) A partial order $(D, \leqslant)$ is a set D with a binary relation $\leqslant$ on D that is reflexive, transitive, and antisymmetric. An upper bound for a subset A of a partial order D is an element $u \in D$ such that $a \leqslant u$ for all $a \in A$; we write $ub(A)$ for the set of upper bounds of A. The set $mub(A) = \{u \in ub(A) \mid \forall d \in D\colon d \leqslant u \;\&\; d \in ub(A) \Rightarrow d = u\}$ consists of all minimal upper bounds of A in D.

A subset A of D is directed iff all finite subsets of A have an upper bound in A. Given $X \subseteq D$ we write $\downarrow_D X$ for $\{d \in D \mid \exists x \in X\colon d \leqslant x\}$, $\uparrow_D X$ for $\{d \in D \mid \exists x \in X\colon x \leqslant d\}$, and elide the subscript D if it is determined by context. We use $\downarrow x$ and $\uparrow x$ if $X = \{x\}$. Subsets U of D with $U = \uparrow U$ are upper sets, and subsets L of D with $L = \downarrow L$ are lower sets.

A partial order $(D, \leqslant)$ is a dcpo iff all its directed subsets have a least upper bound $\bigvee A$, i.e. iff there is some $\bigvee A \in D$ with $ub(A) = \uparrow\bigvee A$. An element $k \in D$ is compact in a dcpo D iff for all directed sets A of D with $k \leqslant \bigvee A$ there is some $a \in A$ with $k \leqslant a$; we write $\mathbf{K}(D)$ for the set of compact elements. A dcpo D is algebraic iff for all $d \in D$ the set $\{k \in \mathbf{K}(D) \mid k \leqslant d\}$ is directed with least upper bound d. For a finite subset F of D define $mub^1(F) = mub(F)$, $mub^{n+1}(F) = mub(mub^n(F))$ for all $n \geqslant 1$, and $mub^\infty(F) = \bigcup_{n \geqslant 1} mub^n(F)$. A *SFP*-domain, also known as a *bifinite* domain, is an algebraic dcpo D such that for every finite subset $F \subseteq \mathbf{K}(D)$ the set $mub^\infty(F)$ is finite and contained in $\mathbf{K}(D)$ with $ub(F) = \uparrow mub(F)$.

A topological space (X, τ) consists of a set X and a family τ of subsets of X such that $\{\}$ and X are in τ, and τ is closed under finite intersections and arbitrary unions. Elements $O \in \tau$ are τ-open, complements $X \setminus O$ with $O \in \tau$ are τ-closed, and sets that are τ-open and τ-closed are τ-clopen. A topological space (X, τ) is τ-compact iff for all $\mathcal{U} \subseteq \tau$ with $X \subseteq \bigcup \mathcal{U}$ there is a finite subset $\mathcal{F} \subseteq \mathcal{U}$ with $X \subseteq \bigcup \mathcal{F}$. A subset A of X is dense in (X, τ) iff $A \cap O$ is non-empty for all non-empty $O \in \tau$.

Given a topological space (X, τ) and a subset $Y \subseteq X$, the subspace topology on Y consists of the set $\{O \cap Y \mid O \in \tau\}$. A subset Y of X is τ-compact iff Y is compact in its subspace topology. A subset Y is τ-saturated in X iff Y is the intersection of τ-open sets. The upper powerdomain $\mathcal{U}[X]$ is defined as the set of all τ-compact τ-saturated subsets of X, ordered by reverse inclusion.

The definitions and characterizations below assume that D is a *SFP*-domain. The Scott-topology on D consists of all subsets U of D satisfying $U = \uparrow(U \cap \mathbf{K}(D))$; such elements are Scott-open. The Lawson-topology on D consists of all subsets V of D such that $x \in V$ implies the existence of some $k, l \in \mathbf{K}(D)$ with $x \in \uparrow k \setminus \uparrow l \subseteq V$. Note that every Scott-open is Lawson-open and every Scott-closed is therefore Lawson-closed. For all $d \in D$, the set $\uparrow d$ is Lawson-closed upper. A subset C of D is Scott-compact (Scott-)saturated in D iff C is Lawson-closed upper in D. A subset U of D is Scott-open and Scott-compact iff U is of the form $\uparrow F$ for a finite set $F \subseteq \mathbf{K}(D)$.

A collection $(F_i)_{i \in I}$ of subsets of D, indexed by a directed set $(I, \leqslant)$, is filtered iff (for all $i, j \in I$ there is some $k \in I$ with $k \in ub(\{i, j\})$ such that $F_k \subseteq F_i \cap F_j$). The Hofmann-Mislove Theorem [HM81] states that if the intersection $\bigcap_{i \in I} C_i$ of a filtered collection of Scott-compact saturated sets $(C_i)_{i \in I}$ in D is contained in a Scott-open set $U \subseteq D$, then there is some $i_0 \in I$ with $C_{i_0} \subseteq U$ already.

B. Proofs of auxiliary or secondary results

Proof of Proposition 5.1 Let (L, i) be a labelled transition system, a modal transition system with must-transitions only, and set $d = \llparenthesis L, i \rrparenthesis$. Since (L, i) and $(\mathcal{D}, d)$ are refinement-equivalent by item 6 of Fact 4.1 and since $\models^a$ equals $\models^c$ on modal transition systems with must-transitions only by item 3 of Fact 4.1, we infer that, for all ϕ of Hennessy-Milner logic, $((\mathcal{D}, d) \models^a \phi$ iff $(\mathcal{D}, d) \models^c \phi)$. Proof by contradiction: If $d \notin max(\mathbb{D})$, there is some $e \in \mathbb{D}$ with $d \leqslant e$ and $e \not\leqslant d$. Since $\mathbb{D}$ is algebraic, the latter implies that there is some $k \in \mathbf{K}(D)$ with $k \leqslant e$ and $k \not\leqslant d$. By (15) there is some $p \in MPA$ with $k = \llparenthesis\, p \,\rrparenthesis$. But $k \leqslant e$ means $(\mathcal{D}, e) \models^a \psi_p$ and $k \not\leqslant d$ means $(\mathcal{D}, d) \not\models^a \psi_p$ by Lemma 4.3 which, as inferred above, implies $(\mathcal{D}, d) \not\models^c \psi_p$. But $(\mathcal{D}, e) \models^a \psi_p$ implies $(\mathcal{D}, e) \models^c \psi_p$ by item 2 of Fact 4.1 and so $d \leqslant e$ implies $(\mathcal{D}, d) \models^c \psi_p$ by item 5 of Fact 4.1, a contradiction. □

Proof of Proposition 5.2

1. We proceed by structural induction on ϕ of Hennessy-Milner logic. This is evident for the clauses tt, negation, and conjunction since $\mathbb{D}$ is Lawson-clopen and clopens are closed under set complement

($[\![\neg\phi]\!]^a = \mathbb{D} \setminus [\![\phi]\!]^c$ and $[\![\neg\phi]\!]^c = \mathbb{D} \setminus [\![\phi]\!]^a$) and finite intersections. We require mode-dependent proofs for $\langle\alpha\rangle\phi$, where $[\![\langle\alpha\rangle\phi]\!]^m = \{d \in \mathbb{D} \mid d^m_\alpha \cap [\![\phi]\!]^m \neq \{\}\}$ for $m \in \{a, c\}$.

- Let $m = a$. By Theorem 4.2 in [HJS04] all $[\![\psi]\!]^a$, ψ formula of Hennessy-Milner logic, are Scott-open, so $[\![\langle\alpha\rangle\phi]\!]^a$ is Scott-open and therefore Lawson-open and it suffices to show that $[\![\langle\alpha\rangle\phi]\!]^a$ is Lawson-closed, i.e. Scott-compact as an upper set. By induction, $[\![\phi]\!]^a$ is Lawson-clopen; it is also Scott-open so $[\![\phi]\!]^a = \uparrow F_\phi$ for a finite subset $F_\phi \subseteq \mathbf{K}(\mathbb{D})$ as $\mathbb{D}$ is algebraic. By the definition of $[\![\langle\alpha\rangle\phi]\!]^a$, we have $e \in [\![\langle\alpha\rangle\phi]\!]^a$ iff $e^a_\alpha \cap \uparrow F_\phi \neq \{\}$ iff $e^a_\alpha \cap F_\phi \neq \{\}$ (as e^a_α is a lower set). For each $y \in F_\phi$ define $c(y) = (c(y)_\gamma)_{\gamma \in Act} \in \mathbb{D}$ by $c(y)_\beta = (\{\}, \mathbb{D})$ for all $\beta \neq \alpha$; and $c(y)_\alpha = (\downarrow y, \mathbb{D})$. Then $C = \{c(y) \mid y \in F_\phi\}$ is finite and $C \subseteq \mathbf{K}(\mathbb{D})$. Since $y \in c(y)^a_\alpha \cap F_\phi$ for all $y \in C$, we get $\uparrow C \subseteq [\![\langle\alpha\rangle\phi]\!]^a$ as the latter set is upper. Note that for each $y \in F_\phi$ we have $c(y) \leqslant e$ in $\mathbb{D}$ iff $y \in e^a_\alpha$. Therefore, $e \in [\![\langle\alpha\rangle\phi]\!]^a$ implies $e \in \uparrow C$. Thus, $[\![\langle\alpha\rangle\phi]\!]^a$ equals $\uparrow C$ for the finite subset C of $\mathbf{K}(\mathbb{D})$.
- Let $m = c$. From Theorem 4.2 in [HJS04] we already know that $[\![\langle\alpha\rangle\phi]\!]^c$ is Scott-closed and therefore Lawson-closed. Thus, it suffices to show that $[\![\langle\alpha\rangle\phi]\!]^c$ is Lawson-open. By induction, $[\![\phi]\!]^c$ is Lawson-open and therefore $D \setminus [\![\phi]\!]^c = [\![\neg\phi]\!]^a$ is Lawson-closed (and Scott-open), i.e. Scott-compact upper. Since $\mathbb{D}$ is algebraic, $[\![\neg\phi]\!]^a = \uparrow F_{\neg\phi}$ for a finite subset $F_{\neg\phi}$ of $\mathbf{K}(\mathbb{D})$. Thus, $[\![\phi]\!]^c = \mathbb{D} \setminus \uparrow F_{\neg\phi}$. Inspecting the definition of $[\![\langle\alpha\rangle\phi]\!]^c$, we infer $e \in [\![\langle\alpha\rangle\phi]\!]^c$ iff there is some $x \in e^c_\alpha$ such that $x \notin \uparrow F_{\neg\phi}$. Now let $d \in [\![\langle\alpha\rangle\phi]\!]^c$. We claim that there are compact elements k and l with $d \in \uparrow k \setminus \uparrow l \subseteq [\![\langle\alpha\rangle\phi]\!]^c$, which concludes the proof since $\uparrow k \setminus \uparrow l$ is Lawson-open. Choose any $k \in \downarrow d \cap \mathbf{K}(\mathbb{D})$. As for $l = (l_\gamma)_{\gamma \in Act}$, set $l_\beta = (\{\}, \mathbb{D})$ for all $\beta \neq \alpha$; and $l_\alpha = (\{\}, \uparrow F_{\neg\phi})$; in particular, $l \in \mathbf{K}(\mathbb{D})$. Note that $l \not\leqslant e$ in $\mathbb{D}$ iff $e^c_\alpha \not\subseteq \uparrow F_{\neg\phi}$ iff (for some $x \in e^c_\alpha$, $x \notin \uparrow F_{\neg\phi}$). Therefore, $d \in \uparrow k \setminus \uparrow l \subseteq [\![\langle\alpha\rangle\phi]\!]^c$.

2. The first claim follows from item 1 and the fact that all $[\![\phi]\!]^a$ are Scott-open, as shown in [HJS04]. The second claim follows from this, Lemma 4.2, and item 6 of Fact 4.1.
3. By item 2 of Theorem 4 in [HJS04] each $[\![\phi]\!]^a$ is Scott-open. The inclusion $[\![\phi]\!]^a \subseteq V_\phi$ follows from item 5 of Fact 4.1. The set V_ϕ is an upper set in $\mathbb{D}$ since $d \in V_\phi$ and $d \leqslant e$ imply $\uparrow e \cap max(\mathbb{D}) \subseteq \uparrow d \cap max(\mathbb{D}) \subseteq [\![\phi]\!]^a$ and so $e \in V_\phi$. Let $d \in V_\phi$. Since $\mathbb{D}$ is algebraic, it suffices to show that $l \in V_\phi$ for some $l \in \downarrow d \cap \mathbf{K}(\mathbb{D})$ as then V_ϕ is Scott-open. Let M be the intersection of all Lawson-closed upper subsets of $\mathbb{D}$ that contain $max(\mathbb{D})$. Then for every $x \in \mathbb{D}$ the set $\uparrow x \cap M$ is Scott-compact and saturated in $\mathbb{D}$ as the intersection of two Lawson-closed upper sets. Since D is algebraic, we infer that $\{\uparrow l \cap M \mid l \in \downarrow d \cap \mathbf{K}(\mathbb{D})\}$ is a filtered collection of Scott-compact saturated subsets of $\mathbb{D}$ whose intersection equals $\uparrow d \cap M$. But the latter set is contained in $[\![\phi]\!]^a$. This is so since $[\![\phi]\!]^a$ is Lawson-closed upper; and $d \in V_\phi$ implies $\uparrow d \cap max(\mathbb{D}) \subseteq [\![\phi]\!]^a$ and so $[\![\phi]\!]^a$ also contains the intersection of all Lawson-closed upper subsets containing $\uparrow d \cap max(\mathbb{D})$ — which is $\uparrow d \cap M$. Since $\mathbb{D}$ is sober as an algebraic domain and since the filtered intersection of Scott-compact saturated sets $\bigcap\{\uparrow l \cap M \mid l \in \downarrow d \cap \mathbf{K}(\mathbb{D})\}$ is contained in the Scott-open $[\![\phi]\!]^a$, the Hofmann-Mislove Theorem [HM81] implies the existence of some $l \in \downarrow d \cap \mathbf{K}(\mathbb{D})$ such that $\uparrow l \cap M \subseteq [\![\phi]\!]^a$. The latter implies $\uparrow l \cap max(\mathbb{D}) \subseteq [\![\phi]\!]^a$ as $max(\mathbb{D}) \subseteq M$, so $l \in V_\phi$.
4. First, let $\langle\!| N, i |\!\rangle$ be in V_ϕ and consider any $(M, j) \in \mathcal{I}[N, i]$. By Proposition 5.1, $\langle\!| M, j |\!\rangle$ is an element of $\uparrow\langle\!| N, i |\!\rangle \cap max(\mathbb{D})$ and so $\langle\!| N, i |\!\rangle \in V_\phi$ implies $(\mathcal{D}, \langle\!| M, j |\!\rangle) \models^a \phi$ whence $(M, j) \models^a \phi$ by item 6 of Fact 4.1. Thus, $(N, i) \models^{a+} \phi$ follows.
   Second, assume that $(N, i) \models^{a+} \phi$. Proof by contradiction: If $\langle\!| N, i |\!\rangle \notin V_\phi$, then there has to be some $m \in \uparrow\langle\!| N, i |\!\rangle \cap max(\mathbb{D})$ satisfying $\neg\phi$. Let $X_{(N,i)}$ be the formula defined in (13) which satisfies, for all pointed modal transition systems (M, j), that $(M, j) \models^a X_{(N,i)}$ iff $(N, i) \prec (M, j)$. In particular, $(\mathcal{D}, m) \models^a X_{(N,i)}$. Therefore, $(\mathcal{D}, m)$ is refinement equivalent to a labelled transition system by Theorem 2.2 of [Hut04], not necessarily image-finite, that satisfies $\neg\phi \wedge X_{(N,i)}$. Since $\neg\phi$ is in Hennessy-Milner logic and since all bodies in any $X_{(N,i')}$ with i' being R^c-reachable from i in N involve finite disjunctions or conjunctions only — as (N, i) is image-finite — there is some image-finite labelled transition system (L, l) that satisfies $\neg\phi \wedge X_{(N,i)}$ as well. But then $(L, l) \models^a X_{(N,i)}$ implies $(N, i) \prec (L, l)$ and so (L, l) is an implementation of (N, i) which does not satisfy ϕ, a contradiction.

Proof of Lemma 6.1 Given $\uparrow\langle\!| M,', i' |\!\rangle \cap max(\mathbb{D}) \subseteq \uparrow\langle\!| M, i |\!\rangle \cap max(\mathbb{D})$, item 6 of Fact 4.1 and Proposition 5.1 imply $\mathcal{I}[M', i'] \subseteq \mathcal{I}[M, i]$. Conversely, we use proof by contradiction. Suppose that $\uparrow\langle\!| M,', i' |\!\rangle \cap max(\mathbb{D}) \not\subseteq \uparrow\langle\!| M, i |\!\rangle \cap max(\mathbb{D})$ and $\mathcal{I}[M', i'] \subseteq \mathcal{I}[M, i]$. The former means that there is some $m \in \uparrow\langle\!| M', i' |\!\rangle \cap max(\mathbb{D})$ that is not in $\uparrow\langle\!| M, i |\!\rangle \cap max(\mathbb{D})$, i.e. $\langle\!| M, i |\!\rangle \not\leqslant m$. Since $\mathbb{D}$ is algebraic, there is some $p \in MPA$ with $\{\!| p |\!\} \leqslant \langle\!| M, i |\!\rangle$ and $\{\!| p |\!\} \not\leqslant m$ by (15).

Since $m \in \uparrow\!\langle\!\langle M', i' \rangle\!\rangle$, we obtain $(\mathcal{D}, m) \models^a X_{(M',i')}$ by Lemma 4.1. But $\langle\!\langle p \rangle\!\rangle \not\leqslant m$ implies $(\mathcal{D}, m) \not\models^a \psi_p$ by Lemma 4.3 and so $(\mathcal{D}, m) \models^a \neg\psi_p$ as $m \in max(\mathbb{D})$ so $(\mathcal{D}, m)$ is refinement-equivalent to a pointed labelled transition system by Theorem 2.2 of [Hut04] and $\models^a$ equals $\models^c$ for $(\mathcal{D}, m)$. This implies $(\mathcal{D}, m) \models^{a+} \neg\psi_p \wedge X_{(M',i')}$ as $\models^a$ is sound and so there is some labelled transition system that satisfies $\neg\psi_p \wedge X_{(M',i')}$. But for all states s of (M', i') the bodies in $X_{(M',s)}$ are finite conjunctions and disjunctions — as (M', i') is image-finite — and $\neg\psi_p$ is a formula of Hennessy-Milner logic, so there has be to some image-finite labelled transition system (L, l) satisfying $\neg\psi_p \wedge X_{(M',i')}$ as well. In particular, (L, l) satisfies $X_{(M',i')}$ and so $(L, l) \in \mathcal{I}[M', i']$ which is contained in $\mathcal{I}[M, i]$ by assumption. Therefore, $\langle\!\langle p \rangle\!\rangle \leqslant \langle\!\langle M, i \rangle\!\rangle \leqslant \langle\!\langle L, l \rangle\!\rangle$ implies that (L, l) satisfies ψ_p, a contradiction. □

Proof of Corollary 6.1 We write (1), (2), and (3) for the first, second, and third "iff" statement (respectively). Then (1) $\Rightarrow$ (2) holds as $(M, i) \prec (N, j)$ implies $\mathcal{I}[N, j] \subseteq \mathcal{I}[M, i]$. The implication (2) $\Rightarrow$ (3) follows from the duality of $\models^{a+}$ and $\models^{c-}$ as the statements quantify over all ϕ of Hennessy-Milner logic. To see (3) $\Rightarrow$ (1), we use proof by contradiction. Assume (3) and let $(M, i) \not\prec (N, j)$. By Theorem 6.1, there is some $(L, l) \in \mathcal{I}[N, j] \setminus \mathcal{I}[M, i]$. Then $(M, j) \not\prec (L, l)$ implies that there is some $p \in MPA$ with $\langle\!\langle p \rangle\!\rangle \leqslant \langle\!\langle M, i \rangle\!\rangle$ and $\langle\!\langle p \rangle\!\rangle \not\leqslant \langle\!\langle L, l \rangle\!\rangle$ by (15). But then $\langle\!\langle p \rangle\!\rangle \leqslant \langle\!\langle M, i \rangle\!\rangle$ implies $(\mathcal{D}, \langle\!\langle M, i \rangle\!\rangle) \models^{a+} \psi_p$ since $\uparrow\!\langle\!\langle M, i \rangle\!\rangle \cap max(\mathbb{D}) \subseteq \uparrow\!\langle\!\langle p \rangle\!\rangle = [\![\psi_p]\!]^a$, whereas $\langle\!\langle p \rangle\!\rangle \not\leqslant \langle\!\langle L, l \rangle\!\rangle$ implies $(\mathcal{D}, \langle\!\langle L, l \rangle\!\rangle) \not\models^a \psi_p$ and so $(\mathcal{D}, \langle\!\langle N, j \rangle\!\rangle) \not\models^{a+} \psi_p$ as $(L, l) \in \mathcal{I}[N, j]$. But then $(\mathcal{D}, \langle\!\langle N, j \rangle\!\rangle) \models^{c-} \neg\psi_p$ follows by duality and so (3) implies $(\mathcal{D}, \langle\!\langle M, i \rangle\!\rangle) \models^{c-} \neg\psi_p$, a contradiction to $(\mathcal{D}, \langle\!\langle M, i \rangle\!\rangle) \models^{a+} \psi_p$. □

Received January 2004
Revised August 2004
Accepted December 2004 by M. Leuschel and D. J. Cooke
Published online 25 May 2005