FI MU

Faculty of Informatics
Masaryk University Brno

Parametric Modal Transition Systems

by

Nikola Beneš
Jan Křetínský
Kim G. Larsen
Mikael H. Møller
Jiří Srba

FI MU Report Series FIMU-RS-2011-03

July 2011

Publications in the FI MU Report Series are in general accessible via WWW:

`http://www.fi.muni.cz/reports/`

Further information can be obtained by contacting:

Faculty of Informatics
Masaryk University
Botanická 68a
602 00 Brno
Czech Republic

Parametric Modal Transition Systems

Nikola Beneš[*]
Faculty of Informatics, Masaryk University
xbenes3@fi.muni.cz

Jan Křetínský[†]
Faculty of Informatics, Masaryk University
Technische Universität München
jan.kretinsky@fi.muni.cz

Kim G. Larsen
Aalborg Universitet
kgl@cs.aau.dk

Mikael H. Møller
Aalborg Universitet
mikaelhm@cs.aau.dk

Jiří Srba
Aalborg Universitet
srba@cs.aau.dk

July 14, 2011

Abstract

Modal transition systems (MTS) is a well-studied specification formalism of reactive systems supporting a step-wise refinement methodology. Despite its many advantages, the formalism as well as its currently known extensions are incapable of expressing some practically needed aspects in the refinement process like exclusive, conditional and persistent choices. We introduce a new model called parametric modal transition systems (PMTS) together with a general modal refinement notion that overcome many of the limitations and we investigate the computational complexity of modal refinement checking.

[*] The author has been supported by Czech Grant Agency, grant no. GAP202/11/0312.

[†] The author is a holder of Brno PhD Talent Financial Aid and is supported by the Czech Science Foundation, grant No. P202/10/1469.

1 Introduction

The specification formalisms of Modal Transition Systems (MTS) [LT88, AHL$^+$08] grew out of a series of attempts to achieve a flexible and easy-to-use compositional development methodology for reactive systems. In fact the formalism of MTS may be seen as a fragment of a temporal logic [BL92], while having a behavioural semantics allowing for an easy composition with respect to process constructs.

In short, MTS are labelled transition systems equipped with two types of transitions: *must* transitions which are mandatory for any implementation, and *may* transitions which are optional for an implementation. Refinement of an MTS now essentially consists of iteratively resolving the unsettled status of may transitions: either by removing them or by turning them into must transitions.

It is well admitted (see e.g. [RBB$^+$09]) that MTS and their extensions like disjunctive MTS (DMTS) [LX90], 1-selecting MTS (1MTS) [FS08] and transition systems with obligations (OTS) [BK10] provide strong support for a specification formalism allowing for step-wise refinement process. Moreover, the MTS formalisms have applications in other contexts, which include verification of product lines [GLS08, LNW07], interface theories [UC04, RBB$^+$09] and modal abstractions in program analysis [GHJ01, HJS01, NNN08].

Unfortunately, all of these formalisms lack the capability to express some intuitive specification requirements like exclusive, conditional and persistent choices. In this paper we extend considerably the expressiveness of MTS and its variants so that it can model arbitrary Boolean conditions on transitions and also allows to instantiate persistent transitions. Our model, called *parametric modal transition systems* (PMTS), is equipped with a finite set of parameters that are fixed prior to the instantiation of the transitions in the specification. The generalized notion of modal refinement is designed to handle the parametric extension and it specializes to the well-studied modal refinements on all the subclasses of our model like MTS, disjunctive MTS and MTS with obligations.

To the best of our knowledge, this is the first sound attempt to introduce persistence into a specification formalism based on modal transition systems. The most related work is by Fecher and Schmidt on 1-selecting MTS [FS08] where the authors allow to model exclusive-or and briefly mention the desire to extend the formalism with persistence. However, as in detail explained in Appendix A, their definition does not capture the notion of persistence. Our formalism is in several aspects semantically more general and handles persistence in a complete and uniform manner.

The main technical contribution, apart from the formalism itself, is a comprehensive complexity characterization of modal refinement checking on all of the practically relevant subclasses of PMTS. We show that the complexity ranges from P-completeness to Π_4^p-completeness, depending on the requested generality of the PMTS specifications on the left-hand and right-hand sides.

2 Parametric Modal Transition Systems

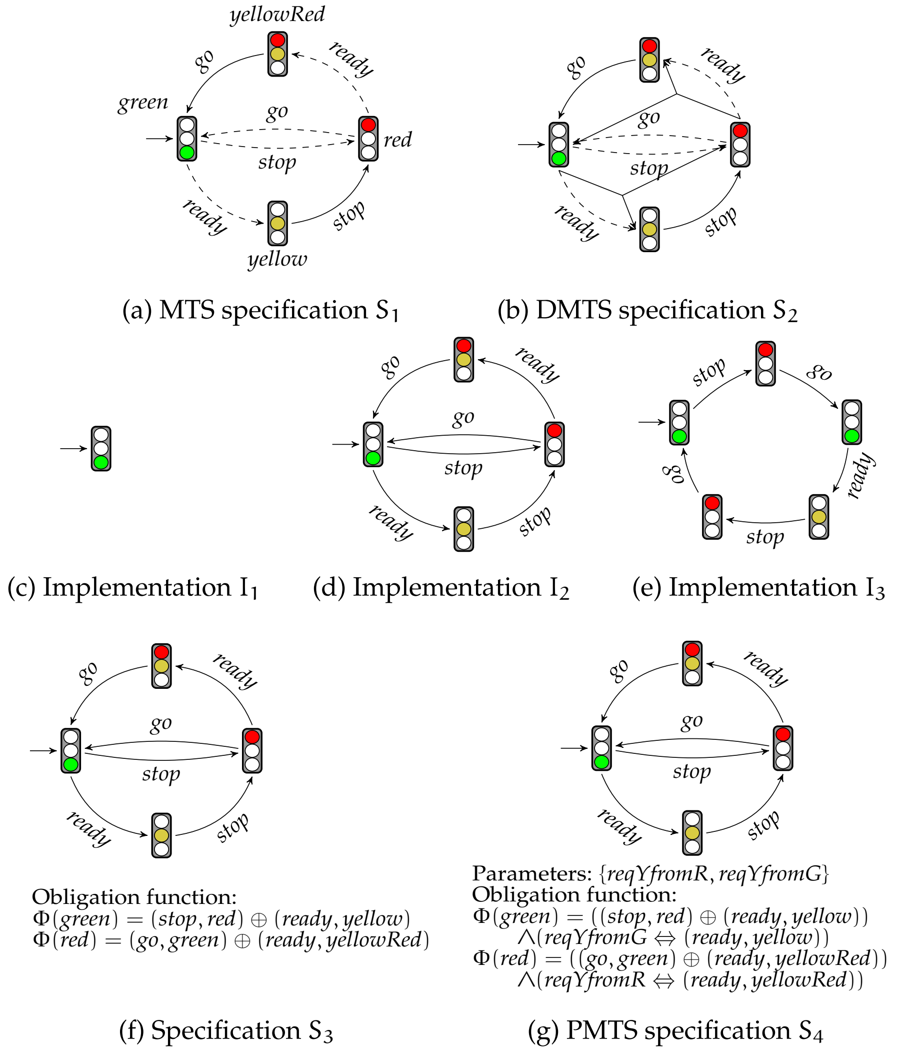
In this section we present the formalism of parametric modal transition systems (PMTS), starting with a motivating example and continuing with the formal definitions, followed by the general notion of modal refinement.

2.1 Motivation

Modal transition systems and their extensions described in the literature are lacking the capability to express several specification requirements like exclusive, conditional and persistent choices. We shall now discuss these limitations on an example as a motivation for the introduction of parametric MTS formalism with general Boolean conditions in specification requirements.

Consider a simple specification of a traffic light controller that can be at any moment in one of the four predefined states: *red*, *green*, *yellow* or *yellowRed*. The requirements of the specification are: when *green* is on the traffic light may either change to *red* or *yellow* and if it turned *yellow* it must go to *red* afterward; when *red* is on it may either turn to *green* or *yellowRed*, and if it turns *yellowRed* (as it is the case in some countries) it must go to *green* afterwords.

Figure 1a shows an obvious MTS specification (defined formally later on) of the proposed specification. The transitions in the standard MTS formalism are either of type may (optional transitions depicted as dashed lines) or must (required transitions depicted as solid lines). In Figure 1c, Figure 1d and Figure 1e we present three different implementations of the MTS specification where there are no more optional transitions. The implementation I_1 does not implement any may transition as it is a valid possibility to satisfy the specification S_1. Of course, in our concrete example, this means that the light is constantly *green* and it is clearly an undesirable behaviour that cannot be, how-



(a) MTS specification S_1

(b) DMTS specification S_2

(c) Implementation I_1

(d) Implementation I_2

(e) Implementation I_3

Obligation function:
$\Phi(green) = (stop, red) \oplus (ready, yellow)$
$\Phi(red) = (go, green) \oplus (ready, yellowRed)$

(f) Specification S_3

Parameters: $\{reqYfromR, reqYfromG\}$
Obligation function:
$\Phi(green) = ((stop, red) \oplus (ready, yellow))$
$\quad \wedge (reqYfromG \Leftrightarrow (ready, yellow))$
$\Phi(red) = ((go, green) \oplus (ready, yellowRed))$
$\quad \wedge (reqYfromR \Leftrightarrow (ready, yellowRed))$

(g) PMTS specification S_4

Figure 1: Specifications and implementations of a traffic light controller

ever, easily avoided. The second implementation I_2 on the other hand implements all may transitions, again a legal implementation in the MTS methodology but not a desirable implementation of a traffic light as the next action is not always deterministically given. Finally, the implementation I_3 of S_1 illustrates the third problem with the MTS specifications, namely that the choices made in each turn are not persistent and the implementation alternates between entering *yellow* or not. None of these problems can be avoided when using the MTS formalism.

A more expressive formalism of disjunctive modal transition systems (DMTS) can overcome some of the above mentioned problems. A possible DMTS specification S_2 is depicted in Figure 1b. Here the *ready* and *stop* transitions, as well as *ready* and *go* ones, are disjunctive, meaning that it is still optional which one is implemented but at least one of them must be present. Now the system I_1 in Figure 1c is not a valid implementation of S_2 any more. Nevertheless, the undesirable implementations I_2 and I_3 are still possible and the modelling power of DMTS is insufficient to eliminate them.

Inspired by the recent notion of transition systems with obligations [BK10], we can model the traffic light using specification as a transition system with arbitrary[1] obligation formulae. These formulae are Boolean propositions over the outgoing transitions from each state, whose satisfying assignments yield the allowed combinations of outgoing transitions. A possible specification called S_3 is given in Figure 1f and it uses the operation of exclusive-or. We will follow an agreement that whenever the obligation function for some node is not listed in the system description then it is implicitly understood as requiring all the available outgoing transitions to be present. Due to the use of exclusive-or in the obligation function, the transition systems I_1 and I_2 are not valid implementation any more. Nevertheless, the implementation I_3 in Figure 1e cannot be avoided in this formalism either.

Finally, the problem with the alternating implementation I_3 is that we cannot enforce in any of the above mentioned formalisms a uniform (persistent) implementation of the same transitions in all its states. In order to overcome this problem, we propose the so-called parametric MTS where we can, moreover, choose persistently whether the transition to *yellow* is present or not via the use of parameters. The PMTS specification with two parameters *reqYfromR* and *reqYfromG* is shown in Figure 1g. Fixing a priori the (Boolean) values of the parameters makes the choices permanent in the whole implementation, hence we eliminate also the last problematic implementation I_3.

2.2 Definition of Parametric Modal Transition System

We shall now formally capture the intuition behind parametric MTS introduced above. First, we recall the standard propositional logic.

A Boolean formula over a set X of atomic propositions is given by the following abstract syntax

$$\varphi ::= \mathbf{tt} \mid x \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi$$

where x ranges over X. The set of all Boolean formulae over the set X is denoted by $\mathcal{B}(X)$. Let $\nu \subseteq X$ be a truth assignment, i.e. a set of variables with value true, then the satisfaction relation $\nu \models \varphi$ is given by $\nu \models \mathbf{tt}$, $\nu \models x$ iff $x \in \nu$, and the satisfaction of the remaining Boolean connectives is defined in the standard way. We also use the

[1] In the transition systems with obligations only positive Boolean formulae are allowed.

standard derived operators like exclusive-or $\varphi \oplus \psi = (\varphi \wedge \neg\psi) \vee (\neg\varphi \wedge \psi)$, implication $\varphi \Rightarrow \psi = \neg\varphi \vee \psi$ and equivalence $\varphi \Leftrightarrow \psi = (\neg\varphi \vee \psi) \wedge (\varphi \vee \neg\psi)$.

We can now proceed with the definition of parametric MTS.

Definition 2.1. *A* parametric MTS *(PMTS) over an action alphabet* Σ *is a tuple* (S, T, P, Φ) *where* S *is a set of* states, $T \subseteq S \times \Sigma \times S$ *is a* transition relation, P *is a finite set of* parameters, *and* $\Phi : S \to \mathcal{B}((\Sigma \times S) \cup P)$ *is an* obligation function *over the atomic propositions containing outgoing transitions and parameters. We implicitly assume that whenever* $(a, t) \in \Phi(s)$ *then* $(s, a, t) \in T$. *By* $T(s) = \{(a, t) \mid (s, a, t) \in T\}$ *we denote the set of all outgoing transitions of* s.

We recall the agreement that whenever the obligation function for some node is not listed in the system description then it is implicitly understood as $\Phi(s) = \bigwedge T(s)$, with the empty conjunction being **tt**.

We call a PMTS *positive* if, for all $s \in S$, any negation occurring in $\Phi(s)$ is applied only to a parameter. A PMTS is called *parameter-free* if $P = \emptyset$. We can now instantiate the previously studied specification formalisms as subclasses of PMTS.

Definition 2.2. *A PMTS is called*

- transition system with obligation *(OTS) if it is parameter-free and positive,*
- disjunctive modal transition system *(DMTS) if it is an OTS and* $\Phi(s)$ *is in the conjunctive normal form for all* $s \in S$,
- modal transition system *(MTS) if it is a DMTS and* $\Phi(s)$ *is a conjunction of positive literals (transitions) for all* $s \in S$, *and*
- implementation *(or simply a labelled transition system) if it is an MTS and* $\Phi(s) = \bigwedge T(s)$ *for all* $s \in S$.

Note that positive PMTS, despite the absence of a general negation and the impossibility to define for example exclusive-or, can still express useful requirements like $\Phi(s) = p \Rightarrow (a, t) \wedge \neg p \Rightarrow (b, u)$ requiring in a state s a conditional presence of certain transitions. Even more interestingly, we can enforce binding of actions in different states, thus ensuring certain functionality. Take a simple two state-example: $\Phi(s) = p \Rightarrow (\textit{request}, t)$ and $\Phi(t) = p \Rightarrow (\textit{response}, s)$. We shall further study OTS with formulae in the disjunctive normal form that are dual to DMTS and whose complexity of parallel composition is lower [BK10] while still being as expressive as DMTS.

2.3 Modal Refinement

A fundamental advantage of MTS-based formalisms is the presence of *modal refinement* that allows for a step-wise system design (see e.g. [AHL+08]). We shall now provide such a refinement notion for our general PMTS model so that it will specialize to the well-studied refinement notions on its subclasses. In the definition, the parameters are fixed first (persistence) followed by all valid choices modulo the fixed parameters that now behave as constants.

First we set the following notation. Let (S, T, P, Φ) be a PMTS and $\nu \subseteq P$ be a truth assignment. For $s \in S$, we denote by $\mathrm{Tran}_\nu(s) = \{E \subseteq T(s) \mid E \cup \nu \models \Phi(s)\}$ the set of all admissible sets of transitions from s under the fixed truth values of the parameters.

We can now define the notion of modal refinement between PMTS.

Definition 2.3 (Modal Refinement). *Let* (S_1, T_1, P_1, Φ_1) *and* (S_2, T_2, P_2, Φ_2) *be two PMTSs. A binary relation* $R \subseteq S_1 \times S_2$ *is a* modal refinement *if for each* $\mu \subseteq P_1$ *there exists* $\nu \subseteq P_2$ *such that for every* $(s, t) \in R$ *holds*

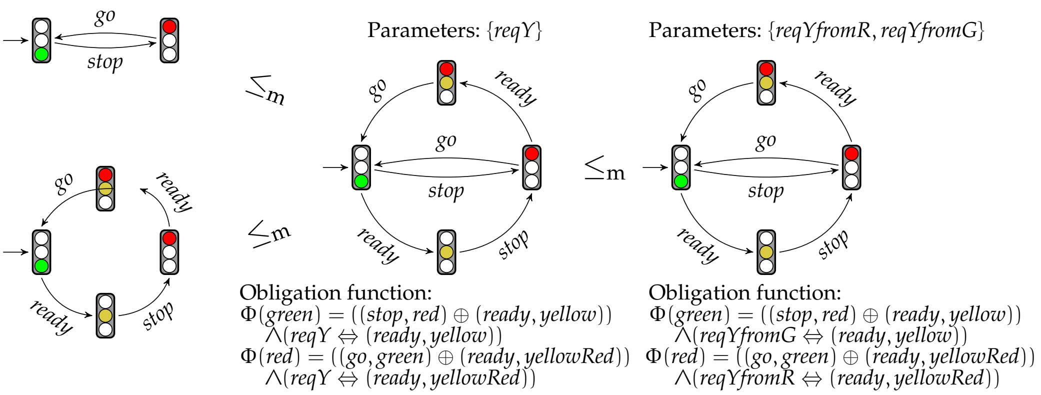
$$\begin{aligned}\forall M \in \mathrm{Tran}_\mu(s) : \exists N \in \mathrm{Tran}_\nu(t) : \ & \forall(a, s') \in M : \exists(a, t') \in N : (s', t') \in R \ \wedge \\ & \forall(a, t') \in N : \exists(a, s') \in M : (s', t') \in R\,.\end{aligned}$$

We say that s *modally refines* t*, denoted by* $s \leq_m t$*, if there exists a modal refinement* R *such that* $(s, t) \in R$*.*

Example 2.4. *Consider the rightmost PMTS in Figure 2. It has two parameters reqYfromG and reqYfromR whose values can be set independently and it can be refined by the system in the middle of the figure having only one parameter reqY. This single parameter simply binds the two original parameters to the same value. The PMTS in the middle can be further refined into the implementations where either yellow is always used in both cases, or never at all. Notice that there are in principle infinitely many implementations of the system in the middle, however, they are all bisimilar to either of the two implementations depicted in the left of Figure 2.*

In the next section, we shall investigate the complexity of positive subclasses of PMTS. For this reason we prove the following lemma showing how the definition of modal refinement can be simplified in this particular case.

We shall first realize that in positive PMTS and for any truth assignment ν, $\mathrm{Tran}_\nu(s)$ is *upward closed*, meaning that if $M \in \mathrm{Tran}_\nu(s)$ and $M \subseteq M' \subseteq T(s)$ then $M' \in \mathrm{Tran}_\nu(s)$.



Parameters: $\{reqY\}$

Obligation function:
$\Phi(green) = ((stop, red) \oplus (ready, yellow))$
$\wedge(reqY \Leftrightarrow (ready, yellow))$
$\Phi(red) = ((go, green) \oplus (ready, yellowRed))$
$\wedge(reqY \Leftrightarrow (ready, yellowRed))$

Parameters: $\{reqYfromR, reqYfromG\}$

Obligation function:
$\Phi(green) = ((stop, red) \oplus (ready, yellow))$
$\wedge(reqYfromG \Leftrightarrow (ready, yellow))$
$\Phi(red) = ((go, green) \oplus (ready, yellowRed))$
$\wedge(reqYfromR \Leftrightarrow (ready, yellowRed))$

Figure 2: Example of modal refinement

Lemma 2.5. *Consider Definition 2.3 where the right-hand side PMTS is positive. Now the condition in Definition 2.3 can be equivalently rewritten as a conjunction of conditions* (1) *and* (2)

$$\forall M \in \mathrm{Tran}_\mu(s) : \forall (a, s') \in M : \exists (a, t') \in T(t) : (s', t') \in R \tag{1}$$

$$\forall M \in \mathrm{Tran}_\mu(s) : \mathrm{match}_t(M) \in \mathrm{Tran}_\nu(t) \tag{2}$$

where $\mathrm{match}_t(M)$ *denotes the set* $\{(a, t') \in T(t) \mid \exists (a, s') \in M : (s', t') \in R\}$*. If the left-hand side PMTS is moreover positive too, Condition* (1) *is equivalent to*

$$\forall (a, s') \in T(s) : \exists (a, t') \in T(t) : (s', t') \in R\,. \tag{3}$$

Proof. We shall first argue that the condition of modal refinement is equivalent to the conjunction of Conditions (4) and (5).

$$\forall M \in \mathrm{Tran}_\mu(s) : \exists N \in \mathrm{Tran}_\nu(t) : \forall (a, s') \in M : \exists (a, t') \in N : (s', t') \in R \tag{4}$$

$$\forall M \in \mathrm{Tran}_\mu(s) : \exists N \in \mathrm{Tran}_\nu(t) : \forall (a, t') \in N : \exists (a, s') \in M : (s', t') \in R \tag{5}$$

Let μ, ν, R, s and t be fixed. Definition 2.3 trivially implies both Conditions (4) and (5). We now prove that (4) and (5) imply the condition in Definition 2.3.

Let $M \in \mathrm{Tran}_\mu(s)$ be arbitrary. There is some $N_1 \in \mathrm{Tran}_\nu(t)$ satisfying (4) and some $N_2 \in \mathrm{Tran}_\nu(t)$ satisfying (5). Let now $N_1' = \{(a, t') \in N_1 \mid \exists (a, s') \in M : (s', t') \in R\}$. Consider $N = N_1' \cup N_2$. Clearly, as $\mathrm{Tran}_\nu(t)$ is upward closed, $N \in \mathrm{Tran}_\nu(t)$. Moreover, due to Condition (4) we have some $(a, t') \in N_1$ such that $(s', t') \in R$. Clearly, $(a, t') \in N_1'$ and thus also in N.

Now let $(a, t') \in N$ be arbitrary. If $(a, t') \in N_2$, due to Condition (5) we have some $(a, s') \in M$ such that $(s', t') \in R$. If $(a, t') \notin N_2$ then $(a, t') \in N_1'$. The existence of $(a, s') \in M$ such that $(s', t') \in R$ is then guaranteed by the definition of N_1'.

Let us now proceed with proving the claims of the lemma. Condition (4) is trivially equivalent to (1) since $\mathrm{Tran}_\nu(t)$ is upward closed. Condition (5) is equivalent to (2). Indeed, (2) clearly implies (5) and we show that also (5) implies (2). Let M be arbitrary. We then have some N satisfying (5). Clearly, $N \subseteq \mathrm{match}_t(M)$. Since $\mathrm{Tran}_\nu(t)$ is upward closed, $N \in \mathrm{Tran}_\nu(t)$ implies $\mathrm{match}_t(M) \in \mathrm{Tran}_\nu(t)$. Due to the upward closeness of both $\mathrm{Tran}_\mu(s)$ and $\mathrm{Tran}_\nu(t)$ in the case of a positive left-hand side, the equivalence of (1) and (3) follows. □

Theorem 2.6. *Modal refinement as defined on PMTS coincides with the standard modal refinement notions on MTS, DMTS and OTS. On implementations it coincides with bisimulation.*

Proof. The fact that Definition 2.3 coincides with modal refinement on OTS as defined in [BK10] is a straightforward corollary of Lemma 2.5 and its proof. Indeed, the two conditions given in [BK10] are exactly conditions (3) and (5). As the definition of modal refinement on OTS coincides with modal refinement on DMTS (as shown in [BK10]) and thus also on MTS, the proof is done.

However, for the reader's convenience, we present a direct proof that Definition 2.3 coincides with modal refinement on MTS. Assume a parameter-free PMTS (S, T, P, Φ) where $\Phi(s)$ is a conjunction of transitions for all $s \in S$, in other words it is a standard MTS where the must transitions are listed in the conjunction and the may transitions are simply present in the underlying transition system but not a part of the conjunction. Observe that every transition $(s, a, t) \in T$ is contained in some $M \in \mathrm{Tran}_\emptyset(s)$. Further, each must transition $(s, a, t) \in T$ is contained in all $M \in \mathrm{Tran}_\emptyset(s)$. Therefore, the first conjunct in Definition 2.3 requires that for all may transition from s there be a corresponding one from t with the successors in the refinement relation. Similarly, the second conjunct now requires that for all must transitions from t there be a corresponding must transition from s. This is exactly the standard notion of modal refinement as introduced in [LT88]. □

3 Complexity of Modal Refinement Checking

We shall now investigate the complexity of refinement checking on PMTS and its relevant subclasses. Without explicitly mentioning it, we assume that all considered PMTS are now finite and the decision problems are hence well defined. The complexity bounds include classes from the polynomial hierarchy (see e.g. [Pap94]) where for example $\Sigma_0^p = \Pi_0^p = \mathrm{P}$, $\Pi_1^p = \mathrm{coNP}$ and $\Sigma_1^p = \mathrm{NP}$.

Table 1: Complexity of modal refinement checking of parameter-free systems

	Boolean	Positive	pCNF	pDNF	MTS
Boolean	Π_2^p-complete	coNP-complete	$\in$ coNP P-hard	coNP-complete	$\in$ coNP P-hard
Positive	Π_2^p-complete	coNP-complete	P-complete	coNP-complete	P-complete
pCNF	Π_2^p-complete	coNP-complete	P-complete	coNP-complete	P-complete
pDNF	Π_2^p-complete	P-complete	P-complete	P-complete	P-complete
MTS	Π_2^p-complete	P-complete	P-complete	P-complete	P-complete
Impl	NP-complete	P-complete	P-complete	P-complete	P-complete

3.1 Parameter-Free Systems

Since even the parameter-free systems have interesting expressive power and the complexity of refinement on OTS has not been studied before, we first focus on parameter-free systems. Moreover, the results of this subsection are then applied to parametric systems in the next subsection. The results are summarized in Table 1. The rows in the table correspond to the restrictions on the left-hand side PMTS while the columns correspond to the restrictions on the right-hand side PMTS. Boolean denotes the general system with arbitrary negation. Positive denotes the positive systems, in this case exactly OTS. We use pCNF and pDNF to denote positive systems with formulae in conjunctive and disjunctive normal forms, respectively. In this case, pCNF coincides with DMTS. The special case of satisfaction relation, where the refining system is an implementation is denoted by Impl. We do not include Impl to the columns as it makes sense that an implementation is refined only to an implementation and here modal refinement corresponds to bisimilarity that is P-complete [BGS92] (see also [SJ05]). The P-hardness is hence the obvious lower bound for all the problems mentioned in the table.

We start with the simplest NP-completeness result.

Proposition 3.1. *Modal refinement between an implementation and a parameter-free PMTS is NP-complete.*

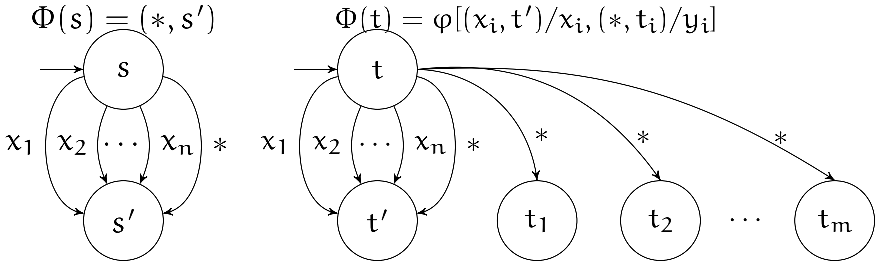
Proof. The containment part is straightforward. First we guess the relation R. As s is an implementation then the set $\mathrm{Tran}_\emptyset(s)$ is a singleton. We thus only need to further guess $N \in \mathrm{Tran}_\nu(t)$ and then in polynomial time verify the two conjuncts in Definition 2.3.

The hardness part is by a simple reduction from SAT. Let $\varphi(x_1, \ldots, x_n)$ be an given Boolean formula (instance of SAT). We construct two PMTSs (S, T, P, Φ) and (S', T', P', Φ') such that (i) $S = \{s, s'\}, T = (s, a, s'), P = \emptyset$, $\Phi(s) = (a, s')$ and $\Phi(s') = \mathbf{tt}$ and (ii) $S' = \{t, t_1, \ldots, t_n\}$, $T = \{(t, a, t_i) \mid 1 \le i \le n.\}$, $P' = \emptyset$, $\Phi(t) = \varphi[(a, t_i)/x_i]$ and $\Phi(t_i) = \mathbf{tt}$ for all i, $1 \le i \le n$. Clearly, φ is satisfiable if and only if $s \le_m t$. □

Next we show that modal refinement is Π_2^p-complete. The following lemma introduces a gadget used also later on in other hardness results. We will refer to it as the *$*$-construction.*

Proposition 3.2. *Modal refinement between two parameter-free PMTS is Π_2^p-hard even if the left-hand side is an MTS.*

Proof. The proof is by polynomial time reduction from the validity of the quantified Boolean formula $\psi \equiv \forall x_1 \ldots \forall x_n \exists y_1 \ldots \exists y_m : \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$ to the refinement checking problem $s \le_m t$ where s and t are given as follows.



Assume that ψ is true. Let $M \in \mathrm{Tran}_\emptyset(s)$ (clearly $(*, s') \in M$) and we want to argue that there is $N \in \mathrm{Tran}_\emptyset(t)$ with $(*, t') \in N$ such that for all $(x_i, s') \in M$ there is $(x_i, t') \in N$ (clearly the states s', t' and t_i are in modal refinement) and for all $(x_i, t') \in N$ there is $(x_i, s') \in M$. Such an N can be found by simply including (x_i, t') whenever $(x_i, s') \in M$ and by adding also $(*, t')$ into N. As ψ is true, we include into N also all $(*, t_i)$ whenever y_i is set to true in ψ. Hence we get $s \le_m t$.

On the other hand if ψ is false then we pick $M \in \mathrm{Tran}_\emptyset(s)$ such that M corresponds to the values of x_i's such that there are no values of $y_1, \ldots, y_m$ that make ψ true. This means that from t there will be no transitions as $\mathrm{Tran}_\emptyset(t) = \emptyset$ assuming that (x_i, t') have to be set to true whenever $(x_i, s') \in M$, otherwise the refinement between s and t will fail. However, now $(*, s') \in M$ cannot be matched from t and hence $s \not\le_m t$. □

Proposition 3.3. *Modal refinement between two parameter-free PMTS is in Π_2^p.*

Proof. The containment follows directly from Definition 2.3 (note that the parameters are empty) and the fact that the last conjunction in Definition 2.3 is polynomially verifiable once the sets M and N were fixed. The relation R could be in principle guessed before it is verified, however, this would increase the complexity bound to Σ_3^p. Instead, we will initially include all pairs (polynomially many) into R and for each pair ask whether for every M there is N such that the two conjuncts are satisfied. If it fails, we remove the pair and continue until we reach (after polynomially many steps) the greatest fixed point. The complexity in this way remains in Π_2^p. We shall use this standard method also in further proofs and refer to it as a co-inductive computation of R. □

3.1.1 Positive Right-Hand Side.

We have now solved all the cases where the right-hand side is arbitrary. We now look at the cases where the right-hand side is positive. In the proofs that follow we shall use the alternative characterization of refinement from Lemma 2.5. The following proposition determines the subclasses on which modal refinement can be decided in polynomial time.

Proposition 3.4. *Modal refinement on parameter-free PMTS is in P, provided that both sides are positive and either the left-hand side is in pDNF or the right-hand side is in pCNF.*

Proof. Due to Lemma 2.5, the refinement is equivalent to the conjunction of (3) and (2). Clearly, (3) can be checked in P. We show that Condition (2) can be verified in P too. Recall that (2) says that

$$\forall M \in \mathrm{Tran}_\mu(s) : \mathrm{match}_t(M) \in \mathrm{Tran}_\nu(t)$$

where $\mathrm{match}_t(M) = \{(a, t') \in T(t) \mid \exists (a, s') \in M : (s', t') \in R\}$.

First assume that the left-hand side is in pDNF. If for some M the Condition (2) is satisfied then it is also satisfied for all $M' \supseteq M$, as $\mathrm{Tran}_\mu(s)$ is upwards closed. It it thus sufficient to verify the condition for all minimal elements (wrt. inclusion) of $\mathrm{Tran}_\mu(s)$. In this case it correspond to the clauses of $\Phi(s)$. Thus we get a polynomial time algorithm as shown in Algorithm 1.

Second, assume that the right-hand side is in pCNF. Note that Condition (2) can be equivalently stated as

$$\forall M : \mathrm{match}_t(M) \notin \mathrm{Tran}_\nu(t) \Rightarrow M \notin \mathrm{Tran}_\mu(s) \tag{6}$$

Algorithm 1: Test for Condition (2) of modal refinement (pDNF)

```
Input  : states $s$ and $t$ such that $\Phi(s)$ is in positive DNF and $\Phi(t)$ is positive,
         relation $R$
Output: true if $s$, $t$ satisfy the refinement condition, false otherwise
foreach clause $(a_1, s_1) \land \cdots \land (a_k, s_k)$ in $\Phi(s)$ do
    $N \leftarrow \{(a, t') \in T(t) \mid \exists i : a_i = a \land (s_i, t') \in R\}$;
    if $N \notin \mathrm{Tran}_\nu(t)$ then return false;
return true;
```

As $\Phi(t)$ is in conjunctive normal form then $N \in \mathrm{Tran}_\nu(t)$ is equivalent to saying that N has nonempty intersection with each clause of $\Phi(t)$. We may thus enumerate all maximal $N \notin \mathrm{Tran}_\nu(t)$. Having a maximal $N \notin \mathrm{Tran}_\nu(t)$, we can easily construct M such that $N = \mathrm{match}_t(M)$. This leads to the polynomial time Algorithm 2.

Algorithm 2: Test for Condition (2) of modal refinement (pCNF)

```
Input  : states $s$ and $t$ such that $\Phi(s)$ is positive and $\Phi(t)$ is in positive CNF,
         relation $R$
Output: true if $s$, $t$ satisfy the refinement condition, false otherwise
foreach clause $(a_1, t_1) \lor \cdots \lor (a_k, t_k)$ in $\Phi(t)$ do
    $M \leftarrow T(s) \setminus \{(a, s') \in T(s) \mid \exists i : a_i = a \land (s', t_i) \in R\}$;
    if $M \in \mathrm{Tran}_\mu(s)$ then return false;
return true;
```

The statement of the proposition thus follows. □

Proposition 3.5. *Modal refinement on parameter-free PMTS is in coNP, if the right-hand side is positive.*

Proof. Due to Lemma 2.5 we can solve the two refinement conditions separately. Furthermore, both Condition (1) an (2) of Lemma 2.5 can be checked in coNP. The guessing of R is done co-inductively as described in the proof of Proposition 3.3. □

Proposition 3.6. *Modal refinement on parameter-free systems is coNP-hard, even if the left-hand side is in positive CNF and the right-hand side is in positive DNF.*

Proof. We reduce SAT into non-refinement. Let $\varphi(x_1, \ldots, x_n)$ be a formula in CNF. We modify φ into an equivalent formula φ' as follows: add new variables $\tilde{x}_1, \ldots, \tilde{x}_n$ and for all i change all occurrences of $\neg x_i$ into $\tilde{x}_i$ and add new clauses $(x_i \lor \tilde{x}_i)$ and $(\neg x_i \lor \neg \tilde{x}_i)$.

Table 2: Complexity of modal refinement checking with parameters

	Boolean	positive	pCNF	pDNF
Boolean	Π_4^p-complete	Π_3^p-complete	$\in \Pi_3^p$ Π_2^p-hard	Π_3^p-complete
positive	Π_4^p-complete	Π_3^p-complete	Π_2^p-complete	Π_3^p-complete
pCNF	Π_4^p-complete	Π_3^p-complete	Π_2^p-complete	Π_3^p-complete
pDNF	Π_4^p-complete	Π_2^p-complete	Π_2^p-complete	Π_2^p-complete
MTS	Σ_3^p-complete	NP-complete	NP-complete	NP-complete
Impl	NP-complete	NP-complete	NP-complete	NP-complete

Observe now that all clauses contain either all positive literals or all negative literals. Let ψ^+ denote a CNF formula that contains all positive clauses of φ' and ψ^- denote a CNF formula that contains all negative clauses of φ'. As $\varphi' = \psi^+ \land \psi^-$ it is clear that φ' is satisfiable if and only if $(\psi^+ \Rightarrow \neg\psi^-)$ is not valid.

Now we construct two PMTSs (S, T, P, Φ) and (S', T', P', Φ') over $\Sigma = \{x_1, \dots, x_n, \tilde{x}_1, \dots, \tilde{x}_n\}$ as follows: (i) $S = \{s, s'\}$, $T = \{(s, x_i, s'), (s, \tilde{x}_i, s') \mid 1 \leq i \leq n\}$, $P = \emptyset$, $\Phi(s) = \psi^+[(x_i, s')/x_i, (\tilde{x}_i, s')/\tilde{x}_i]$ and $\Phi(s') = \mathbf{tt}$, and (ii) $S' = \{t, t'\}$, $T' = \{(t, x_i, t'), (t, \tilde{x}_i, t) \mid 1 \leq i \leq n\}$, $P' = \emptyset$, $\Phi(t) = \neg\psi^-[(x_i, t')/x_i, (\tilde{x}_i, t')/\tilde{x}_i]$ and $\Phi(t') = \mathbf{tt}$. Note that by pushing the negation of ψ^- inside, this formula can be written as pDNF. It is easy to see that now $s \leq_m t$ if and only if $(\psi^+ \Rightarrow \neg\psi^-)$ is valid. Therefore, $s \not\leq_m t$ if and only if φ is satisfiable. □

Note that the exact complexity of modal refinement with the right-hand side being in positive CNF or MTS and the left-hand side Boolean remains open.

3.2 Systems with Parameters

In the sequel we investigate the complexity of refinement checking in the general case of PMTS with parameters. The complexities are summarized in Table 2. We start with an observation of how the results on parameter-free systems can be applied to the parametric case.

Proposition 3.7. *The complexity upper bounds from Table 1 carry over to Table 2, as follows. If the modal refinement in the parameter-free case is in* NP, coNP *or* Π_2^p*, then the modal refinement* *with parameters is in Π_2^p, Π_3^p and Π_4^p, respectively. Moreover, if the left-hand side is an MTS, the complexity upper bounds shift from* NP *and* Π_2^p *to* NP *and* Σ_3^p*, respectively.*

Proof. In the first case, we first universally choose μ, we then existentially choose ν and modify the formulae $\Phi(s)$ and $\Phi(t)$ by evaluating the parameters. This does not change the normal form/positiveness of the formulae. We then perform the algorithm for the parameter-free refinement. For the second case note that implementations and MTS have no parameters and we may simply choose (existentially) ν and run the algorithm for the parameter-free refinement. □

We now focus on the respective lower bounds (proof of Proposition 3.9 can be found in Appendix B).

Proposition 3.8. *Modal refinement between an implementation and a right-hand side in positive CNF or in DNF is NP-hard.*

Proof. The proof is by reduction from SAT. Let $\varphi(x_1, \dots, x_n)$ be a formula in CNF and let $\varphi_1, \varphi_2, \dots, \varphi_k$ be the clauses of φ. We construct two PMTSs (S, T, P, Φ) and (S', T', P', Φ') over the action alphabet $\Sigma = \{a_1, \dots, a_k\}$ as follows: (i) $S = \{s, s'\}$, $T = \{(s, a_i, s') \mid 1 \leq i \leq k\}$, $P = \emptyset$, $\Phi(s) = \bigwedge_{1 \leq i \leq k}(a_i, s')$ and $\Phi(s') = \mathbf{tt}$ and (ii) $S' = \{t\} \cup \{t_i \mid 1 \leq i \leq k\}$, $T' = \{(t, a_i, t_i) \mid 1 \leq i \leq k\}$, $P' = \{x_1, \dots, x_n\}$, $\Phi'(t) = \bigwedge_{1 \leq i \leq k}(a_i, t_i)$ and $\Phi'(t_i) = \varphi_i$ for all $1 \leq i \leq k$. Notice that each φ_i in $\Phi'(t_i)$ is in positive form as we negate only the parameters x_i and every clause φ_i is trivially in DNF. Now we easily get that $s \leq_m t$ if and only if φ is satisfiable. □

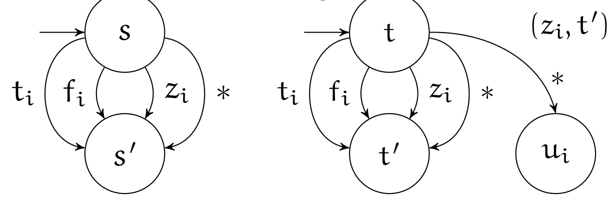
Proposition 3.9. *Modal refinement is Σ_3^p-hard even if the left-hand side is MTS.*

The following proof introduces a gadget used also later on in other hardness results. We refer to it as *CNF-binding*. Further, we use the $*$-construction here.

Proposition 3.10. *Modal refinement is Π_4^p-hard even if the left-hand side is in positive CNF.*

Sketch. Consider a Π_4^p-hard QSAT instance, a formula $\psi = \forall x \exists y \forall z \exists w : \varphi(x, y, z, w)$ with φ is in CNF and x, y, z, w vectors of length n. We construct two system s and t and use the variables $\{x_1, \dots, x_n\}$ as parameters for the left-hand side system s, and $\{y_1, \dots, y_n\}$ as parameters for the right-hand side system t.

$\Phi(s) = (*, s') \wedge$ CNF-binding $\Phi(t) = (*, s') \wedge \varphi[(t_i, t')/x_i, (f_i, t')/\neg x_i,$
$(z_i, t')/z_i, (*, u_i)/w_i]$



for all $1 \leq i \leq n$

On the left we require $\Phi(s) = (*, s') \wedge \bigwedge_{1\leq i\leq n} \Big((x_i \Rightarrow (t_i, s')) \wedge (\neg x_i \Rightarrow (f_i, s')\Big)$ and call the latter conjunct *CNF-binding*. Thus the value of each parameter x_i is "saved" into transitions of the system. Note that although both t_i and f_i may be present, a "minimal" implementation contains exactly one of them. On the right-hand side the transitions look similar but we require $\Phi(t) = (*, t) \wedge \varphi'$ where φ' is created from φ by changing every positive literal x_i into (t_i, t'), every negative literal $\neg x_i$ into (f_i, t'), every z_i into (z_i, t'), and every w_i into $(*, u_i)$.

We show that ψ is true iff $s \leq_m t$. Assume first that ψ is true. Therefore, for every choice of parameters x_i there is a choice of parameters y_i so that $\forall z \exists w : \varphi(x, y, z, w)$ is true and, moreover, t_i or f_i is present on the left whenever x_i or $\neg x_i$ is true, respectively (and possibly even if it is false). We set exactly all these transitions t_i and f_i on the right, too. Further, for every choice of transitions z_i on the left there are w_i's so that $\varphi(x, y, z, w)$ holds. On the right, we implement a transition (z_i, t') for each z_i set to true and $(*, u_i)$ for each w_i set to true. Now φ' is satisfied as it has only positive occurrences of (t_i, t') and (f_i, t') and hence the extra t_i's and f_i's do not matter. Now for every implementation of s we obtained an implementation of t. Moreover, their transitions match. Indeed, t_i's and f_i's were set the same as on the left, similarly for z_i's. As for the $*$-transition, we use the same argumentation as in the original $*$-construction. On the left, there is always one. On the right, there can be more of them due to w_i's but at least one is also guaranteed by $\Phi(t)$.

Let now $s \leq_m t$. Then for every choice of x_i's—and thus also for every choice of *exactly* one transition of t_i, f_i for each i—there are y_i's so that every choice of transitions z_i can be matched on the right so that φ' is true with some transitions $(*, u_i)$. Since choices of t_i/f_i correspond exactly to choices of x_i it only remains to set w_i true for each transition $(*, u_i)$ on the right, thus making φ true. □

Based on the idea of CNF-binding, the following propositions are proved in Appendix B.

Proposition 3.11. *Modal refinement is Π_3^p-hard for the left-hand side in positive CNF and the right-hand side in positive DNF.*

Proposition 3.12. *Modal refinement is Π_2^p-hard even if both sides are in positive CNF.*

The last three propositions use a modification of the CNF-binding idea called *DNF-binding*. Instead of $(x_i \Rightarrow (t_i, s')) \wedge (\neg x_i \Rightarrow (f_i, s'))$ we use $(x_i \wedge (t_i, s')) \vee (\neg x_i \wedge (f_i, s'))$

to bind parameters of the left-hand side system with transitions of the right-hand side system. Details are in Appendix B.

Proposition 3.13. *Modal refinement is Π_2^{p}-hard even if left-hand side is in positive DNF and right-hand side is in positive CNF.*

Proposition 3.14. *Modal refinement is Π_2^{p}-hard even if left-hand side is in positive DNF and right-hand side is in positive DNF.*

Proposition 3.15. *Modal refinement is Π_4^{p}-hard even if the left-hand side is in positive DNF.*

Although the complexity may seem discouraging in many cases, there is an important remark to make. The refinement checking may be exponential, but only in the outdegree of each state and the number of parameters, while it is polynomial in the number of states. As one may expect the outdegree and the number of parameters to be much smaller than the number of states, this means that the refinement checking may still be done in a rather efficient way. This claim is furthermore supported by the existence of efficient SAT solvers that may be employed to check the inner conditions in the modal refinement.

4 Conclusion and Future Work

We have introduced an extension of modal transition systems called PMTS for parametric systems. The formalism is general enough to capture several features missing in the other extensions, while at the same time it offers an easy to understand semantics and a natural notion of modal refinement that specializes to the well-known refinements already studied on the subclasses of PMTS. Finally, we provided a comprehensive overview of complexity of refinement checking on PMTS and its subclasses.

We believe that our formalism is a step towards a more applicable notion of specification theories based on MTS. In the future work we will study logical characterizations of the refinement relation, investigate compositional properties and focus on introducing quantitative aspects into the model in order to further increase its applicability.

Acknowledgments. We would like to thank to Sebastian Bauer for suggesting the traffic light example and for allowing us to use his figure environments.

References

[AHL$^+$08] A. Antonik, M. Huth, K. G. Larsen, U. Nyman, and A. Wasowski. 20 years of modal and mixed specifications. *Bulletin of the EATCS no. 95*, pages 94–129, 2008.

[BGS92] J. L. Balcazar, J. Gabarró, and M. Santha. Deciding bisimilarity is P-complete. *Formal aspects of computing*, 4(6 A):638–648, 1992.

[BK10] N. Beneš and J. Křetínský. Process algebra for modal transition systemses. In Ludek Matyska, Michal Kozubek, Tomás Vojnar, Pavel Zemcík, and David Antos, editors, *MEMICS*, volume 16 of *OASICS*, pages 9–18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2010.

[BL92] G. Boudol and K. G. Larsen. Graphical versus logical specifications. *Theor. Comput. Sci.*, 106(1):3–20, 1992.

[FS08] H. Fecher and H. Schmidt. Comparing disjunctive modal transition systems with an one-selecting variant. *J. of Logic and Alg. Program.*, 77(1-2):20–39, 2008.

[GHJ01] P. Godefroid, M. Huth, and R. Jagadeesan. Abstraction-based model checking using modal transition systems. In *Proc. CONCUR'01*, volume 2154 of *LNCS*, pages 426–440. Springer, 2001.

[GLS08] A. Gruler, M. Leucker, and K. D. Scheidemann. Modeling and model checking software product lines. In Gilles Barthe and Frank S. de Boer, editors, *FMOODS*, volume 5051 of *Lecture Notes in Computer Science*, pages 113–131. Springer, 2008.

[HJS01] M. Huth, R. Jagadeesan, and D. A. Schmidt. Modal transition systems: A foundation for three-valued program analysis. In *Proc. of ESOP'01*, volume 2028 of *LNCS*, pages 155–169. Springer, 2001.

[LNW07] K. G. Larsen, U. Nyman, and A. Wasowski. On modal refinement and consistency. In *Proc. of CONCUR'07*, volume 4703 of *LNCS*, pages 105–119. Springer, 2007.

[LT88] K. G. Larsen and B. Thomsen. A modal process logic. In *LICS*, pages 203–210. IEEE Computer Society, 1988.

[LX90] K. G. Larsen and L. Xinxin. Equation solving using modal transition systems. In *LICS*, pages 108–117. IEEE Computer Society, 1990.

[NNN08] S. Nanz, F. Nielson, and H. R. Nielson. Modal abstractions of concurrent behaviour. In *Proc. of SAS'08*, volume 5079 of *LNCS*, pages 159–173. Springer, 2008.

[Pap94] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley Publishing Co., Inc., Reading, MA, USA, 1994.

[RBB$^+$09] J.-B. Raclet, E. Badouel, A. Benveniste, B. Caillaud, and R. Passerone. Why are modalities good for interface theories? In *ACSD*, pages 119–127. IEEE, 2009.

[SJ05] Z. Sawa and P. Jančar. Behavioural equivalences on finite-state systems are PTIME-hard. *Computing and informatics*, 24(5):513–528, 2005.

[UC04] S. Uchitel and M. Chechik. Merging partial behavioural models. In *Proc. of FSE'04*, pages 43–52. ACM, 2004.

A Note on Expressiveness of Other MTS Variants

DMTS and OTS are capable of expressing positive Boolean conditions, but not exclusive or, neither persistence. 1MTS are moreover capable of expressing exclusive or. Nevertheless, 1MTS do not support persistence. The reason is that Definition 25 [FS08] of refinement relation R has he form of:

$$\begin{aligned} \forall(s,t) \in R : &\forall\gamma \in \mathsf{choice}(s \dashrightarrow) : \exists\delta \in \mathsf{choice}(t \dashrightarrow) \\ &\qquad\qquad \forall M \in s \dashrightarrow : \exists N \in t \dashrightarrow : \gamma(M) R \gamma(N) \wedge \\ &\qquad\qquad \forall N \in t \longrightarrow : \exists M \in s \longrightarrow : \gamma(M) R \gamma(N) \end{aligned}$$

where the choice functions are chosen dependent on the states s, t. Therefore, in different states refining the same state, different decisions can be made. As an example consider a specification S with a hypermust transition under a and b to itself and the corresponding may hypertransition. This systems has an implementation I with transitions (I, a, J) and (J, b, I). Indeed, both I and J refine S as in the former case we choose δ so that $\delta(\{(a, S), (b, S)\}) = a$ while in the latter case $\delta(\{(a, S), (b, S)\}) = b$. Thus, the alternation problem is still present in 1MTS.

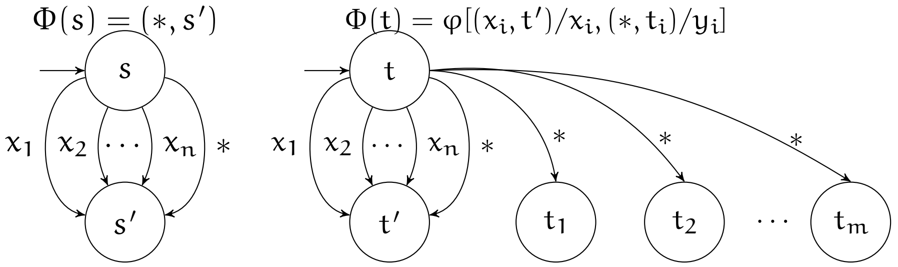
B Proofs from Section 3

In this section, we prove Propositions 3.9, 3.11, 3.12, 3.13, 3.14 and 3.15 giving the lower bounds for the cases not proved in the main text. We start with Proposition 3.9, as its setting is the simplest.

Proposition 3.9. Modal refinement is Σ_3^p-hard even if the left-hand side is MTS.

Proof. The proof is done using the construction of the proof of Proposition 3.2 with parameters added on the right-hand side.

We will make a reduction from the validity of the quantified Boolean formula $\psi \equiv \exists z_1, \ldots, z_k : \forall x_1 \ldots \forall x_n \exists y_1 \ldots \exists y_m : \varphi(z_1, \ldots, z_k, x_1, \ldots, x_n, y_1, \ldots, y_m)$ to the refinement checking problem $s \leq_m t$ where s and t are given as follows. Moreover, the right-hand side system has $\{z_1, \ldots, z_k\}$ as its set of parameters.



Assume that ψ is true. Then there exists a valuation ν on $\{z_1, \dots, z_k\}$ such that $\forall x : \exists y : \varphi(x, y)$ is true. Using now the same argument as that in the proof of Proposition 3.2, we get that $s \leq_m t$.

On the other hand let ψ be false and let ν be an arbitrary valuation on $\{z_1, \dots, z_k\}$. We then may again use the reasoning in the proof of Proposition 3.2 to get that $s \not\leq_m t$. □

We now prove Proposition 3.12, followed by Proposition 3.11. The reason for this ordering is that the setting of Proposition 3.12 is simpler and the proof involves just one method, namely that of the CNF-binding, no $*$-construction is used and no additional actions are needed.

Proposition 3.12. Modal refinement is Π_2^p-hard even if both sides are in positive CNF.

Proof. Recall that positive means that there may be negations, but only limited to parameter literals. The proof is done by reduction from the validity of $\forall x_1, \dots, x_n \exists y_1, \dots, y_m : \varphi(x_1, \dots, x_n, y_1, \dots, y_m)$, where φ is in CNF. The idea is that the left-hand side has only x_i as parameters while the right-hand side has y_i as parameters. To ensure that the valuation of x_i is the same on both sides, we bind them through transitions.

Let $\Sigma = \{t_1, \dots, t_n, f_1, \dots, f_n\}$ be the set of actions. The systems (S, T, P, Φ) and (S', T', P', Φ') are built as follows: $S = \{s, s'\}$, $T = \{(s, t_i, s'), (s, f_i, s') \mid 1 \leq i \leq n\}$, $P = \{x_1, \dots, x_n\}$, $\Phi(s) = \bigwedge_{1 \leq i \leq n}((x_i \Rightarrow (t_i, s')) \wedge (\neg x_i \Rightarrow (f_i, s'))$ (note that this may be written in positive CNF), $\Phi(s') = \mathbf{tt}$; $S' = \{t, t'\}$, $T' = \{(t, t_i, t'), (t, f_i, t') \mid 1 \leq i \leq n\}$, $P' = \{y_1, \dots, y_m\}$, $\Phi'(t) = \varphi[(t_i, t')/x_i, (f_i, t')/\neg x_i]$, $\Phi'(t') = \mathbf{tt}$. We now claim that $\forall x \exists y : \varphi$ holds if and only if $s \leq_m t$. We show the two implications separately.

Let first $\forall x \exists y : \varphi$ hold. Let $\mu \subseteq P_1$ be arbitrary. As this is a truth valuation on the x_i variables, we know that there exists a valuation on the y_i variables such that φ holds. Let $\nu \subseteq P_2$ be such a valuation. Let further $M \in \mathrm{Tran}_\mu(s)$ be arbitrary. Clearly, if $x_i \in \mu$ then $(t_i, s') \in M$ and if $x_i \notin \mu$ then $(f_i, s') \in M$.

We set $N = \{(x, t') \mid (x, s') \in M\}$. Clearly, such N satisfies both conjuncts of the refinement definition. We need to show that $N \in \mathrm{Tran}_\nu(t)$. We thus need to show that N satisfies all the clauses in $\Phi'(t) = \varphi[(t_i, t')/x_i, (f_i, t')/\neg x_i]$.

We use the fact that φ holds, given the current valuation μ on x_i and ν on y_i. Let $(\ell_1 \vee \ell_2 \vee \cdots \vee \ell_k)$ be an arbitrary clause of φ. Clearly, at least one literal was satisfied. If that literal was y_i or $\neg y_i$ then the same literal appears in the modified clause of $\Phi'(t)$ and we are done. If that literal was x_i then it has been changed into (t_i, t'), but as $x_i \in \mu$, we have that $(t_i, t') \in N$. Similarly, if that literal was $\neg x_i$ then it has been changed into (f_i, t'), but as $x_i \notin \mu$, we have that $(f_i, t') \in N$. Thus $s \leq_m t$.

For the other implication let $\exists x \forall y : \neg\varphi$ hold. We show that $s \not\leq_m t$. Let μ be the valuation of x_i such that $\exists x \forall y : \neg\varphi$ holds. Let ν be arbitrary. This corresponds to a valuation on y_i.

We now set $M = \{(t_i, s') \mid x_i \in \mu\} \cup \{(f_i, s') \mid x_i \notin \mu\}$. Clearly, $M \in \mathrm{Tran}_\mu(s)$. Let further $N \in \mathrm{Tran}_\nu(t)$. (If $\mathrm{Tran}_\nu(t) = \emptyset$, we are done.)

We know that given the current x and y valuation, φ does not hold. This means that there exists at least one clause of φ that is false. Let $(\ell_1 \vee \ell_2 \vee \cdots \vee \ell_k)$ be such clause. All ℓ_j are false, given current valuation μ and ν. However, the modified clause of $\Phi'(t)$ corresponding to this one is satisfied by N (valuation of (t_i, t') and (f_i, f')) as $N \in \mathrm{Tran}_\nu(t)$.

Therefore, for some i, either $(t_i, t') \in N$ while $x_i \notin \mu$ or $(f_i, t') \in N$ while $x_i \in \mu$. In both cases N does not satisfy the second conjunct part of the modal refinement definition. Therefore $s \not\leq_m t$. □

The next proposition again reuses the idea of CNF-binding in the very same fashion as above. Moreover, it handles more actions, more precisely those that appear as z_i's in Proposition 3.10. Thus, the proof is the same, omitting the $*$-construction. Therefore, we only provide the reduction without repeating the formal arguments that it indeed works.

Proposition 3.11. Modal refinement is Π_3^p-hard for the left-hand side in positive CNF and the right-hand side in positive DNF.

Proof. The proof is done by reduction from the validity of the quantified Boolean formula $\forall x_1, \ldots, x_k \exists y_1, \ldots, y_l \forall z_1, \ldots, z_m : \varphi$ with φ in DNF.

Let the action alphabet be $\Sigma = \{t_1, \ldots, t_k, f_1, \ldots, f_k, z_1, \ldots, z_m\}$.

The two systems (S, T, P, Φ) and (S', T', P', Φ') are built as follows: $S = \{s, s'\}$, $T = \{(s, t_i, s'), (s, f_i, s') \mid 1 \leq i \leq k\} \cup \{(s, z_j, s') \mid 1 \leq j \leq m\}$, $P = \{x_1, \ldots, x_k\}$, $\Phi(s) = \bigwedge_{1 \leq i \leq n}((x_i \Rightarrow (t_i, s')) \wedge (\neg x_i \Rightarrow (f_i, s'))$, $\Phi(s') = \mathbf{tt}$; $S' = \{t, t'\}$, $T' = \{(t, t_i, t'), (t, f_i, t') \mid 1 \leq i \leq k\} \cup \{(t, z_j, t') \mid 1 \leq j \leq m\}$, $P' = \{y_1, \ldots, y_k\}$, $\Phi'(t) = \varphi[(t_i, t')/x_i, (f_i, t')/\neg x_i, (z_i, t')/z_i]$, $\Phi'(t') = \mathbf{tt}$.

Now $\forall x \exists y \forall z : \varphi(x, y, z)$ holds if and only if $s \leq_m t$. □

We now modify the idea of CNF-binding to **DNF-binding** where instead of $(x_i \Rightarrow (t_i, s')) \wedge (\neg x_i \Rightarrow (f_i, s'))$ we use $(x_i \wedge (t_i, s')) \vee (\neg x_i \wedge (f_i, s'))$ to bind parameters of left-hand side with transitions of right-hand side. The binding works slightly differently, as with DNF we are unable to make a conjuction of such formulae for all i. We thus employ a new special action $\bullet$. The left-hand side then first requires a $\bullet$-transition into n different states s_i, each requiring the above formula for its respective i.

Proposition 3.13. Modal refinement is Π_2^p-hard even if left-hand side is in positive DNF and right-hand side is in positive CNF.

Proof. The proof is done by reduction from the validity of the quantified Boolean formula $\forall x_1, \ldots, x_n \exists y_1, \ldots, y_m : \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$, where φ is in CNF.

Let the action alphabet be $\Sigma = \{t_1, \ldots, t_n, f_1, \ldots, f_n, \bullet\}$. The two systems (S, T, P, Φ) and (S', T', P', Φ') are built as follows: $S = \{s, s'\} \cup \{s_i \mid 1 \leq i \leq n\}$, $T = \{(s, \bullet, s_i), (s_i, t_i, s'), (s_i, f_i, s') \mid 1 \leq i \leq n\}$, $P = \{x_1, \ldots, x_n\}$, $\Phi(s) = \bigwedge_i (\bullet, s_i)$, $\Phi(s_i) = (x_i \wedge (t_i, s')) \vee (\neg x_i \wedge (f_i, s'))$, $\Phi(s') = \mathbf{tt}$; $S' = \{t, t'\} \cup \{u_i, v_i \mid 1 \leq i \leq n\}$, $T' = \{(t, \bullet, u_i), (t, \bullet, v_i), (u_i, t_i, t'), (u_i, f_i, t'), (v_i, f_i, t'), (v_i, t_i, t') \mid 1 \leq i \leq n\}$, $P' = \{y_1, \ldots, y_n\}$, $\Phi'(t) = \varphi[(\bullet, u_i)/x_i, (\bullet, v_i)/\neg x_i]$, $\Phi'(u_i) = (t_i, t')$, $\Phi'(v_i) = (f_i, t')$, $\Phi'(t') = \mathbf{tt}$.

Now $\forall x \exists y : \varphi(x, y)$ holds if and only if $s \leq_m t$. The reasoning behind this fact is similar to the proof of Proposition 3.12. □

The proof of the next proposition is only a slight alteration of previous proof where the $\bullet$-construction is performed in two steps.

Proposition 3.14. Modal refinement is Π_2^p-hard even if left-hand side is in positive DNF and right-hand side is in positive DNF.

Proof. The proof is done by reduction from the validity of the quantified Boolean formula $\forall x_1, \ldots, x_n \exists y_1, \ldots, y_m : \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$, where φ is in CNF. Let $\varphi_1, \ldots, \varphi_k$ denote the clauses of φ.

Let the action alphabet be $\Sigma = \{t_1, \dots, t_n, f_1, \dots, f_n, \bullet\}$. The two systems (S, T, P, Φ) and (S', T', P', Φ') are built as follows: $S = \{s, s', s''\} \cup \{s_i \mid 1 \leq i \leq n\}$, $T = \{(s, \bullet, s')\} \cup \{(s', \bullet, s_i), (s_i, t_i, s''), (s_i, f_i, s'') \mid 1 \leq i \leq n\}$, $P = \{x_1, \dots, x_n\}$, $\Phi(s) = (\bullet, s')$, $\Phi(s') = \bigwedge_i (\bullet, s_i)$, $\Phi(s_i) = (x_i \wedge (t_i, s'')) \vee (\neg x_i \wedge (f_i, s''))$, $\Phi(s'') = \mathbf{tt}$; $S' = \{t, t'\} \cup \{u_i, v_i \mid 1 \leq i \leq n\} \cup \{w_j \mid 1 \leq j \leq k\}$, $T' = \{(t, \bullet, w_j) \mid 1 \leq j \leq k\} \cup \{(w_j, \bullet, u_i), (w_j, \bullet, v_i) \mid 1 \leq i \leq n, 1 \leq j \leq k\} \cup \{(u_i, t_i, t'), (u_i, f_i, t'), (v_i, f_i, t'), (v_i, t_i, t') \mid 1 \leq i \leq n\}$, $P' = \{y_1, \dots, y_n\}$, $\Phi'(t) = \bigwedge_j w_j$, $\Phi'(w_j) = \varphi'_j$ where φ'_j is created from φ_j by changing all positive literals x_i into $(\bullet, u_i)$ and all negative literals $\neg x_i$ into $(\bullet, v_i)$. $\Phi'(u_i) = (t_i, t')$, $\Phi'(v_i) = (f_i, t')$, $\Phi'(t') = \mathbf{tt}$.

Now $\forall x \exists y : \varphi(x, y)$ holds if and only if $s \leq_m t$. □

The proof of the last proposition is a combination of DNF-binding (including the $\bullet$-construction) with the previously used $*$-construction.

Proposition 3.15. Modal refinement is Π_4^p-hard even if the left-hand side is in positive DNF.

Proof. The proof is done by reduction from the validity of the quantified Boolean formula $\forall x \exists y \forall z \exists w : \varphi(x, y, z, w)$ where x, y, z, w are all n-dimensional binary vectors and φ is in CNF.

We let $\Sigma = \{t_1, \dots, t_n, f_1, \dots, f_n, z_1, \dots, z_n, *, \bullet\}$ and we create the two systems (S, T, P, Φ), (S', T', P', Φ') over the action alphabet Σ as follows:

$S = \{s, s'\} \cup \{s_i \mid 1 \leq i \leq n\}$, $T = \{(s, \bullet, s_i), (s_i, t_i, s'), (s_i, f_i, s'), (s, z_i, s') \mid 1 \leq i \leq n\} \cup \{(s, *, s')\}$, $P = \{x_1, \dots, x_n\}$, $\Phi(s) = (*, s') \wedge \bigwedge_i (\bullet, s_i)$, $\Phi(s_i) = (x_i \wedge (t_i, s')) \vee (\neg x_i \wedge (f_i, s'))$ for all $1 \leq i \leq n$, $\Phi(s') = \mathbf{tt}$;

$S' = \{t, t'\} \cup \{u_i, v_i, w_i \mid 1 \leq i \leq n\}$, $T' = \{(t, z_i, t'), (t, \bullet, u_i), (t, \bullet, v_i), (t, *, w_i), (u_i, t_i, t'), (u_i, f_i, t'), (v_i, t_i, t'), (v_i, f_i, t') \mid 1 \leq i \leq n\} \cup \{(t, *, t')\}$, $P' = \{y_1, \dots, y_n\}$, $\Phi'(t) = (*, t') \wedge \varphi[(\bullet, u_i)/x_i, (\bullet, v_i)/\neg x_i, (z_i, t')/z_i, (*, w_i)/w_i]$, for all $1 \leq i \leq n$: $\Phi'(u_i) = (t_i, t')$, $\Phi'(v_i) = (f_i, t')$, $\Phi'(w_i) = \Phi'(t') = \mathbf{tt}$.

It can be verified, using similar arguments as before, that $s \leq_m t$ if and only if $\forall x \exists y \forall z \exists w : \varphi(x, y, z, w)$. □