Noname manuscript No.
(will be inserted by the editor)

Refinement Checking on Parametric Modal Transition Systems

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

the date of receipt and acceptance should be inserted later

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 overcomes many of the limitations. We investigate the computational complexity of modal and thorough refinement checking on PMTS and its subclasses and provide a direct encoding of the modal refinement problem into quantified Boolean formulae, allowing us to employ state-of-the-art QBF solvers for modal refinement checking. The experiments we report on show that the feasibility of refinement checking is more influenced by the degree of nondeterminism rather than by the syntactic restrictions on the types of formulae allowed in the description of the PMTS.

1 Introduction

The specification formalism of Modal Transition Systems (MTS) [4, 42] 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 [9, 22], while having a behavioural semantics allowing for an easy composition with respect to process constructs.

Nikola Beneš
Faculty of Informatics, Masaryk University, Brno, Czech Republic

Jan Křetínský
IST, Austria

Kim G. Larsen · Mikael H. Møller · Jiří Srba
Department of Computer Science, Aalborg University, Denmark

Salomon Sickert
Technical University Munich, Germany

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. [52]) that MTS and their extensions like disjunctive MTS (DMTS) [43], 1-selecting MTS (1MTS) [29] and transition systems with obligations (OTS) [12] 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 [33, 41], interface theories [52, 54] and modal abstractions in program analysis [31, 34, 48].

Unfortunately, all of these formalisms lack the capability to express some intuitive specification requirements like exclusive choice (either this option or that option but not both at the same time), conditional choice (under certain conditions an option is mandatory) and persistent choice (once we commit to some option, the same choice must be taken also everywhere else). 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 generalised notion of modal refinement is designed to handle the parametric extension and it specialises 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. A related work by Fecher and Schmidt on 1-selecting MTS [29] allows to model exclusive-or and the authors briefly mention the desire to extend the formalism with persistence. However, as in detail explained in the appendix of [10], their definition does not capture the intended notion of persistence. Our formalism is in several aspects semantically more general and handles persistence in a complete and uniform manner.

Our contribution can be summarised as follows:

- In Section 2, we introduce the formalism of PMTS.
- Section 3 discusses several possible refinement notions and their relationship.
- In Section 4, we provide a comprehensive complexity characterisation of modal refinement checking on a number of 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.
- In Section 5, due to the high complexity of modal refinement in many cases, we also focus on its practical complexity. We reduce modal refinement problems to satisfiability problems solvable directly by QBF solvers and perform preliminary experiments indicating that our solution scales well in the number of parameters.

Related work Over the years, many extensions of MTS have been proposed. While MTS can only specify whether or not a particular transition is required, some

extensions equip MTS with more general abilities to describe what *combinations* of transitions are possible. Disjunctive MTS (DMTS) [19, 43] can specify that at least one of a given set of transitions is present. One selecting MTS [29] allow to choose exactly one of them. Underspecified transition systems (UTS) [30] also allow disjunctions on may transitions. Transition systems with obligations (OTS) [12] can express positive Boolean combinations. The subclass of parameter-free PMTS we study in this paper covers all Boolean combination of transitions. The same holds for acceptance automata [50] and Boolean formulae with states [9], which both express the requirement by listing all possible sets instead of a Boolean formula. Parametric MTS (PMTS) introduced here add parameters on top of it, so that we can also express persistent choices of transitions and relate possible choices in different parts of a system. This way, one can model hardware dependencies of transitions and systems with prices [15].

There are various other approaches to deal with component *refinements*. They range from subtyping [44] over Java modelling language [35] to interface theories close to MTS such as interface automata [2, 45]. Similarly to MTS, interface automata are behavioural interfaces for components. However, their composition works very differently. Furthermore, their notion of refinement is based on alternating simulation [3], which has been proved in [40] strictly less expressive than MTS refinement. The compositionality of the combination of MTS and interface automata based on I/O automata [46] is further investigated in [51].

Further, alternatively to the design of correct software where an abstract verified MTS is transformed into a concrete implementation, one can consider checking correctness of software through *abstracting* a concrete implementation into a coarser system. The use of MTS as abstractions has been advocated e.g. in [31]. While usually overapproximations (or underapproximations) of systems are constructed and thus only purely universal (or existential) properties can be checked, [31] shows that using MTS one can check mixed formulae (arbitrarily combining universal and existential properties) and, moreover, at the same cost as checking universal properties using traditional conservative abstractions. This advantage has been investigated also in the context of systems equivalent or closely related to MTS [24, 26, 27, 32, 34, 47] but parameters have not been used in this context.

The MTS framework has been extended not only with respect to the expressive power in the classical setting of automata, but also lifted to *quantitative settings*. This includes probabilistic systems [23] as well as various timed systems [7, 15, 20, 25, 28, 36] with clear applications in the embedded systems design. The MTS formalism can also be viewed as a fragment of mu-calculus that is "graphically representable" [9, 21]. The graphical representability of a variant of alternating simulation called covariant-contravariant simulation has been recently studied in [1].

This article is an extended version of two conference papers [14, 38] with complete proofs and closing the open problems mentioned in [14].

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.

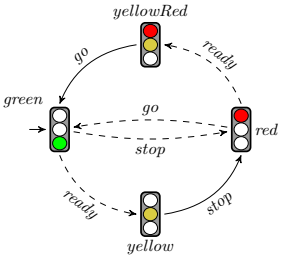
Consider a simple specification of a traffic light controller that can be at any moment in one of the four predefined states: *red*, *green*, *yellow* or *yellowRed*. The requirements of the specification are: when *green* is on the traffic light it may either change to *red* or *yellow* and if it turned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

Figure 2.1a 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 2.1c, Figure 2.1d and Figure 2.1e 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, however, 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.

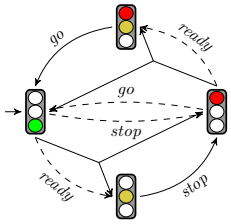
A more expressive formalism of disjunctive modal transition systems (DMTS) can overcome some of the above mentioned problems. A possible DMTS specification S_2 is depicted in Figure 2.1b. Here the *ready* and *stop* transitions, as well as *ready* and *go* ones, are disjunctive, meaning that it is still optional which one is implemented but at least one of them must be present. Now the system I_1 in Figure 2.1c 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 [12], 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 2.1f and it uses the operation of exclusive-or (which is defined via negation). 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 implementations

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

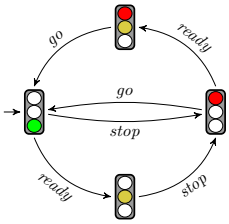


(a) MTS specification S_1

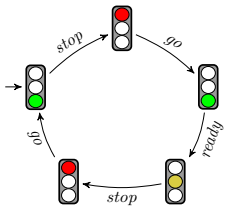


(b) DMTS specification S_2

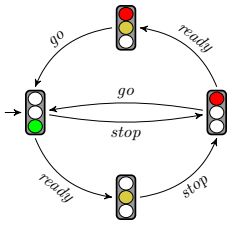
(c) Implementation I_1



(d) Implementation I_2

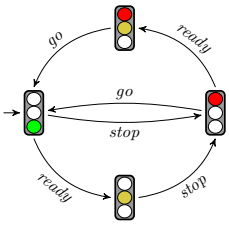


(e) Implementation I_3



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

(g) PMTS specification S_4

Fig. 2.1: Specifications and implementations of a traffic light controller

any more. Nevertheless, the implementation I_3 in Figure 2.1e 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 2.1g. 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 Systems

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 $\mu \subseteq X$ be a truth assignment, i.e. a set of variables with value true, then the satisfaction relation $\mu \models \varphi$ is given by $\mu \models \mathbf{tt}$, $\mu \models x$ iff $x \in \mu$, and the satisfaction of the remaining Boolean connectives is defined in the standard way. We also use the 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) appears 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.

Note that we are not explicitly pointing to any initial state in a given PMTS. In what follows, the initial states will be clear from the context. We also 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 $\mathbf{tt}$.

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

- a *transition system with obligations* (OTS) if it is parameter-free and positive,
- a *disjunctive modal transition system* (DMTS) if it is an OTS and $\Phi(s)$ is in the conjunctive normal form for all $s \in S$,
- a *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
- an *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: $\Phi(s) = p \Rightarrow (\mathit{request}, t)$ and $\Phi(t) = p \Rightarrow (\mathit{response}, s)$.

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. [4]). We shall now provide such a refinement notion for our general PMTS model so that it will specialise 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 $\mu \subseteq P$ be a truth assignment for parameters. For $s \in S$, we denote by

$$\mathrm{Tran}_\mu(s) = \{E \subseteq T(s) \mid E \cup \mu \models \Phi(s)\}$$

the set of all admissible sets of transitions from s under the fixed truth values of the parameters. For example, if $\mu = \{\mathit{reqYfromR}\}$ in the right-most specification in Figure 2.2 then $\mathrm{Tran}_\mu(\mathit{green}) = \{\{(\mathit{stop}, \mathit{red})\}\}$ and $\mathrm{Tran}_\mu(\mathit{red}) = \{\{(\mathit{ready}, \mathit{yellowRed})\}\}$.

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. Let further $\mu \subseteq P_1$ and $\nu \subseteq P_2$. A binary relation $R \subseteq S_1 \times S_2$ is a *modal refinement respecting μ and ν* if 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 $s \leq_{\mathrm{m}} t$, if for each $\mu \subseteq P_1$ there exists $\nu \subseteq P_2$ and a modal refinement R respecting μ and ν such that $(s, t) \in R$.

Example 2.4 Consider the rightmost PMTS in Figure 2.2. 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 on the left of Figure 2.2.

In what follows, 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 realise that in positive PMTS and for any truth assignment μ, $\mathrm{Tran}_\mu(s)$ is *upward closed*, meaning that if $M \in \mathrm{Tran}_\mu(s)$ and $M \subseteq M' \subseteq T(s)$ then $M' \in \mathrm{Tran}_\mu(s)$.

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* (2.1) *and* (2.2)

$$\forall M \in \mathrm{Tran}_\mu(s) : \forall (a, s') \in M : \exists (a, t') \in T(t) : (s', t') \in R \tag{2.1}$$

$$\forall M \in \mathrm{Tran}_\mu(s) : \mathrm{match}_t(M) \in \mathrm{Tran}_\nu(t) \tag{2.2}$$

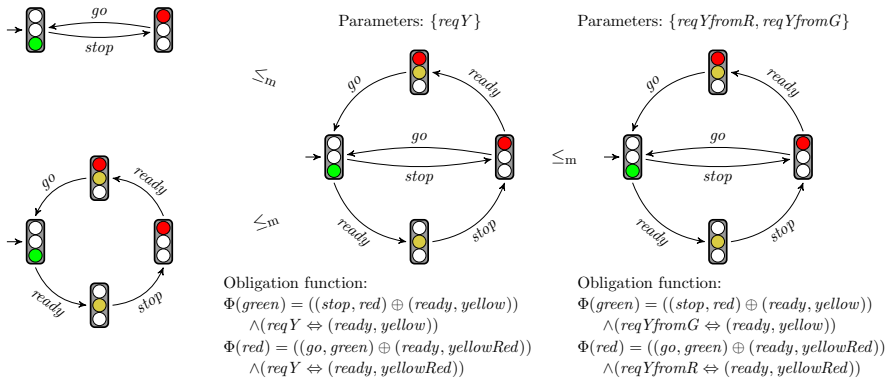


Fig. 2.2: Example of modal refinement

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* (2.1) *is equivalent to*

$$\forall (a,s') \in T(s) : \exists (a,t') \in T(t) : (s',t') \in R \ . \tag{2.3}$$

Proof We shall first argue that the condition of modal refinement is equivalent to the conjunction of Conditions (2.4) and (2.5).

$$\forall M \in \mathrm{Tran}_\mu(s) : \exists N \in \mathrm{Tran}_\nu(t) : \forall (a,s') \in M : \exists (a,t') \in N : (s',t') \in R \tag{2.4}$$

$$\forall M \in \mathrm{Tran}_\mu(s) : \exists N \in \mathrm{Tran}_\nu(t) : \forall (a,t') \in N : \exists (a,s') \in M : (s',t') \in R \tag{2.5}$$

Let μ, ν, R, s and t be fixed. Definition 2.3 trivially implies both Conditions (2.4) and (2.5). We now prove that (2.4) and (2.5) 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 (2.4) and some $N_2 \in \mathrm{Tran}_\nu(t)$ satisfying (2.5). 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 (2.4) for every $(a,s') \in M$ 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 (2.5) 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 (2.4) is trivially equivalent to (2.1) since $\mathrm{Tran}_\nu(t)$ is upward closed. Condition (2.5) is equivalent to (2.2). Indeed, (2.2) clearly implies (2.5) and we show that also (2.5) implies (2.2). Let M be arbitrary. We then have some N satisfying (2.5). 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 (2.1) and (2.3) 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 [12] is a straightforward corollary of Lemma 2.5 and its proof. Indeed, the two conditions given in [12] are exactly conditions (2.3) and (2.5). As the definition of modal refinement on OTS coincides with modal refinement on DMTS (as shown in [12]) 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 $(S, T, \emptyset, \Phi)$ 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 necessarily 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 [42]. □

3 Thorough Refinement

While modal refinement is defined syntactically, there is also a corresponding notion defined semantically, the so-called thorough refinement as studied e.g. in [5, 13, 16, 17]. From a semantical point of view, a state s of a PMTS may be seen to denote the set of its implementations, i.e.

$$[\![s]\!] := \{i \mid i \text{ is an implementation and } i \leq_\mathrm{m} s\} .$$

We can now give the definition of a thorough refinement based on the set inclusion of all possible implementations.

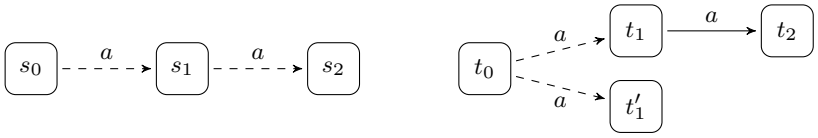
Definition 3.1 (Thorough Refinement) For PMTS states s and t, we say that s *thoroughly refines* t, written $s \leq_\mathrm{t} t$, if $[\![s]\!] \subseteq [\![t]\!]$.

We discuss the relationship of the two refinements. Firstly, the modal refinement is a sound approximation to the thorough refinement.

Proposition 3.2 *Let s and t be PMTS states. If $s \leq_\mathrm{m} t$ then $s \leq_\mathrm{t} t$.*

Proof For any $i \in [\![s]\!]$, we have $i \leq_\mathrm{m} s$ and due to transitivity of $\leq_\mathrm{m}$, $i \leq_\mathrm{m} s \leq_\mathrm{m} t$ implies $i \leq_\mathrm{m} t$, hence $i \in [\![t]\!]$. □

The converse does not hold even for MTS as shown in the following classical example [18] where $s_0 \leq_\mathrm{t} t_0$, but $s_0 \not\leq_\mathrm{m} t_0$.



Let us now consider the subclass of deterministic PMTS. A PMTS (S, T, P, Φ) is *deterministic* if for every $(s, a, t), (s, a, t') \in T$ we have $t = t'$. Provided that the refined MTS is deterministic, the approximation of thorough refinement via a modal refinement is now also complete [18]. This property holds even for parameter-free PMTS, as formulated in the next proposition, and it is a useful observation as deterministic systems often appear in practice [18] and checking modal refinement is—as we shall see later—computationally easier than the thorough refinement.

Proposition 3.3 *Let s_0 be a PMTS state and t_0 a deterministic parameter-free PMTS state. If $s_0 \leq_{\mathrm{t}} t_0$ then $s_0 \leq_{\mathrm{m}} t_0$.*

Proof In this proof, we use the notation $\widehat{T}(s) = \bigcup \mathrm{Tran}_\mu(s)$. We now fix a valuation μ of parameters and define a relation R that satisfies the condition of Definition 2.3. The relation R is taken as the smallest relation such that $(s_0, t_0) \in R$ and whenever $(s, t) \in R$, $(a, s') \in \widehat{T}(s)$ and $(t, a, t') \in T$ then also $(s', t') \in R$. Before we prove that R satisfies the conditions, we make the claim that $(s, t) \in R$ implies $s \leq_{\mathrm{t}} t$. Clearly, this holds for (s_0, t_0). Suppose now that $s \leq_{\mathrm{t}} t$, $(a, s') \in \widehat{T}(s)$, $(t, a, t') \in T$ and i' is an arbitrary implementation of s'. Then there exists an implementation $i \in [\![s]\!]$ such that $i \xrightarrow{a} i'$. But as $s \leq_{\mathrm{t}} t$, i is also an implementation of t. Therefore, as t is deterministic, i' is an implementation of t', thus $s' \leq_{\mathrm{t}} t'$. We can now check that R satisfies the condition of Definition 2.3. Let $(s, t) \in R$ and $M \in \mathrm{Tran}_\mu(s)$. Define $A := \{a \mid \exists s' : (a, s') \in M\}$. There is an implementation i with exactly transitions under A. Moreover, according to the assumption it is also an implementation of t. Hence $N := \{(a, t') \mid (t, a, t') \in T \wedge a \in A\}$ is an element of $\mathrm{Tran}_\emptyset(t)$. The two conjuncts then clearly hold by the construction of R. □

However, the completeness fails if the refined system is deterministic but with parameters.

Example 3.4 Consider a parameter-free PMTS, in fact in this case it is sufficient to use only an MTS, $(\{s_0, s_1\}, \{(s_0, a, s_1)\}, \emptyset, \{s_0 \mapsto \mathbf{tt}, s_1 \mapsto \mathbf{tt}\})$ where we use the standard MTS notation, and a deterministic PMTS $(\{t_0, t_1\}, \{(t_0, a, t_1)\}, \{p\}, \{t_0 \mapsto a \Leftrightarrow p, t \mapsto \mathbf{tt}\})$ below. Obviously $[\![s_0]\!] = [\![t_0]\!]$ contains the implementations with no transitions or one step a-transitions. Although $s_0 \leq_{\mathrm{t}} t_0$, we do not have $s_0 \leq_{\mathrm{m}} t_0$ as we cannot match with any valuation of p.

Corollary 3.5 *There is a PMTS state s and a deterministic PMTS state t such that $s \leq_{\mathrm{t}} t$ but $s \not\leq_{\mathrm{m}} t$.*

In order to overcome the difficulties when modal and thorough refinement do not coincide, the underapproximation of thorough refinement by modal refinement has been complemented by an overapproximation using deterministic hull [18] and parameter-free deterministic hull [38]. The property of the hull $\mathcal{D}$ is that $s \leq_{\mathrm{t}} t$ implies $s \leq_{\mathrm{m}} \mathcal{D}(t)$. The importance of using modal refinement instead of the semantically more precise thorough refinement stems from their complexities. While modal refinement on PMTS can be decided in polynomial space and for

Table 3.1: Thorough refinement complexity and its relation to modal refinement

	MTS	parameter-free PMTS	PMTS
$\leq_t \in$	EXPTIME	NEXPTIME	2-EXPTIME
$s \leq_t t$, t deterministic	$\leq_m = \leq_t$	$\leq_m = \leq_t$	$\leq_m \neq \leq_t$

many subclasses even in polynomial time, deciding the thorough refinement is EXPTIME-hard already for MTS [11]. The upper bounds are summarised in Table 3.1.

Containment of the problem in EXPTIME for MTS was proven in [11] and for DMTS in [19] with the full proof in [8], in NEXPTIME for parameter-free PMTS in [38] with the full proof in [39]. Here we only focus on the general PMTS case. First, we show how to transform PMTS to parameter-free PMTS and DMTS and thus reduce our problems to the already solved one.

For a PMTS, we define a system where we can use any valuation of the parameters.

Definition 3.6 For a PMTS $\mathcal{M} = (S, T, P, \Phi)$ with a given state $s \in S$, we define a parameter-free PMTS called *de-parameterisation* $\mathcal{M}^B = (\{s^B\} \cup S \times 2^P, T', \emptyset, \Phi')$ with a newly added state s^B as

- $T' = \{(s^B, a, (s, \mu)) \mid (s, a, s) \in T, \mu \subseteq P\} \cup \{((s, \mu), a, (s', \mu) \mid (s, a, s') \in T\}$,
- $\Phi'(s^B) = \bigoplus_{\mu \subseteq P} \Phi(s)[\mathbf{tt}/p \text{ for } p \in \mu, \mathbf{ff}/p \text{ for } p \notin \mu, (s, \mu)/s]$, and
- $\Phi'\big((s, \mu)\big) = \Phi(s)[\mathbf{tt}/p \text{ for } p \in \mu, \mathbf{ff}/p \text{ for } p \notin \mu, (s, \mu)/s]$.

The de-parameterisation is a parameter-free PMTS having exactly all the implementations of the PMTS and only one (trivial) valuation.

Proposition 3.7 *Let s be a PMTS state. Then $[\![s]\!] = [\![s^B]\!]$ and $s \leq_m s^B$.*

Proof For any parameter valuation μ we match it with $\emptyset$ and the modal refinement is achieved in the copy with μ fixed in the second component. Clearly, any implementation of s^B corresponds to a particular parameter valuation and thus also to an implementation of s. □

Remark 3.8 Notice that de-parameterisation only preserves the set of implementations but the de-parameterised system does not necessarily modally refine the original system. Moreover, the price we have to pay is a blowup exponential in $|P|$. This is, however, inevitable. Indeed, consider a PMTS $(\{s_0, s_1, s_2\}, \{(s_0, p, s_1), (s_1, p, s_2) \mid p \in P\}, P, \{s_0, s_1 \mapsto \bigwedge_{p \in P}(p, s) \Leftrightarrow p, s_2 \mapsto \mathbf{tt}\})$. Then in every equivalent parameter-free PMTS we need to remember the transitions of the first step so that we can repeat exactly these in the following step. Since there are exponentially many possibilities, the result follows.

Further, similarly to Boolean formulae with states in [9], we can transform every parameter-free PMTS to a DMTS.

Definition 3.9 For a parameter-free PMTS $\mathcal{M} = (S, T, \emptyset, \Phi)$, we define a DMTS called *de-negation* $\mathcal{M}^D = (S', T', \emptyset, \Phi')$

– $S' = \{M \in \mathrm{Tran}_\emptyset(s) \mid s \in S\}$,
– $\Phi'(M) = \bigwedge_{(a,s')\in M} \bigvee_{M'\in \mathrm{Tran}_\emptyset(s')} (a, M')$,

and T' is minimal such that for each $M \in S'$ and each occurrence of (a, M') in $\Phi(M)$, we have $(M, a, M') \in T'$.

However, this DMTS needs to have more initial states in order to be equivalent to the original parameter-free PMTS.

Lemma 3.10 *For a state s of a parameter-free PMTS, $[\![s]\!] = \bigcup_{M\in \mathrm{Tran}_\emptyset(s)} [\![M]\!]$ (where M are taken as states of the de-negation).*

Note that both transformations are exponential. The first one in $|P|$ and the second one in the branching degree. Therefore, their composition is still only singly exponential, yielding a state space where each state has two components: a valuation of original parameters and $\mathrm{Tran}_\emptyset$ of the original state under this valuation.

Theorem 3.11 *Thorough refinement on PMTS is in 2-EXPTIME and EXPTIME-hard.*

Proof Recall that thorough refinement on DMTS is in EXPTIME [5, 13]. Further, note that we have reduced the PMTS and parameter-free PMTS thorough refinement problems to the one on DMTS with more initial states. However, this does not pose a problem. Indeed, let s and t be states of a parameter-free PMTS. We want to check whether $s \leq_\mathrm{t} t$. According to [8] where DMTS only have one initial state, we only need to check whether for each $M \in \mathrm{Tran}_\emptyset(s)$ we have $(M, \mathrm{Tran}_\emptyset(t)) \notin \mathit{Avoid}$ (defined in [8]), which can clearly still be done in exponential time.

The hardness follows from the fact that thorough refinement checking on ordinary MTS is already EXPTIME-hard [13]. □

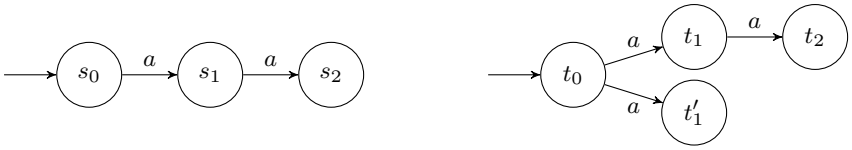
Due to this high complexity of the thorough refinement, we shall only deal with the modal refinement from now on. Note that the original version of modal refinement on PMTS [14] required, for $s_0 \leq_\mathrm{m} t_0$ to hold, that there exists a *fixed* $R \subseteq S_1 \times S_2$ such that for every $\mu \subseteq P_1$ there exists $\nu \subseteq P_2$ satisfying for each $(s, t) \in R$

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

Clearly, the original definition is stronger: For any two PMTS states, if $s_0 \leq_\mathrm{m} t_0$ holds according to [14] it also holds according to Definition 2.3. Indeed, the relation for any sets of parameters can be chosen to be the fixed relation R. On the other hand, the opposite does not hold, as illustrated by the following example.

Example 3.12 Consider the PMTS on the left with parameter set $\{p\}$ and obligation $\Phi(s_0) = (a, s_1),\ \Phi(s_1) = (b, s_2) \Leftrightarrow p,\ \Phi(s_2) = \mathbf{tt}$ and the PMTS on the right with parameter set $\{q\}$ and obligation $\Phi(t_0) = \big((a, t_1) \Leftrightarrow q\big) \wedge \big((a, t_1') \Leftrightarrow \neg q\big),\ \Phi(t_1) = (a, t_2),\ \Phi(t_2) = \Phi(t_1') = \mathbf{tt}$. On the one hand, according to our definition $s_0 \leq_\mathrm{m} t_0$, we intuitively agree this should be the case (and note they also have the same set of implementations). On the other hand, the original definition does not allow to conclude modal refinement between s_0 and t_0. The reason is that depending on

the value of p, s_1 is put in the relation either with t_1 (for p being true and thus choosing q true, too) or with t_1' (for p being false and thus choosing q false, too). In contrast to the original definition, our definition allows us to pick different relations for different parameter valuations.



We propose our modification of the definition since it is not only more intuitive, but it also better approximates the thorough refinement and for all considered fragments of PMTS has the same complexity as the original one. Note that the two definitions coincide on parameter-free PMTS. Further, on MTS they coincide with the classical definition [42] and on labelled transition systems with bisimulation.

4 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. [49]) where for example $\Sigma_0^{\mathrm{P}} = \Pi_0^{\mathrm{P}} = \mathrm{P}$, $\Pi_1^{\mathrm{P}} = \mathrm{coNP}$ and $\Sigma_1^{\mathrm{P}} = \mathrm{NP}$.

Let us also recall the QBF decision problems.

Definition 4.1 (QBF_n^Q) Let Ap be a set of atomic propositions, partitioned into n sets $Ap = \bigcup_{i=0}^{n} X_i$. Let $\phi \in \mathcal{B}(Ap)$ be a Boolean formula over the set of atomic propositions Ap and let $Q \in \{\forall, \exists\}$ be a quantifier where by convention $\overline{\forall} = \exists$ and $\overline{\exists} = \forall$. Then a formula of the form

$$QX_1\overline{Q}X_2QX_3\overline{Q}X_4 \dots \widetilde{Q}X_n\phi \quad \text{where } \widetilde{Q} = \begin{cases} Q & \text{if } n \text{ is odd} \\ \overline{Q} & \text{if } n \text{ is even} \end{cases}$$

is an instance of QBF_n^Q if and only if it is true.

A formula is true means that if e.g. $Q = \exists$ then there is some partial valuation for the atomic propositions in X_1, such that for all partial valuations for the elements of X_2, there is another partial valuation for the propositions of X_3 and so on up to X_n, such that ϕ is satisfied by the union of all partial valuations. It is well known, see e.g. [49], that these problems are complete for the polynomial hierarchy: for each $i \geq 1$, $\mathrm{QBF}_i^{\exists}$ is Σ_i^{P}-complete and $\mathrm{QBF}_i^{\forall}$ is Π_i^{P}-complete.

4.1 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 summarised in

Table 4.1: Complexity of modal refinement checking of parameter-free systems

	Boolean	Positive	pCNF	pDNF	MTS
Boolean	Π_2^{P}-compl.	coNP-compl.	coNP-compl.	coNP-compl.	coNP-compl.
Positive	Π_2^{P}-compl.	coNP-compl.	P-compl.	coNP-compl.	P-compl.
pCNF	Π_2^{P}-compl.	coNP-compl.	P-compl.	coNP-compl.	P-compl.
pDNF	Π_2^{P}-compl.	P-compl.	P-compl.	P-compl.	P-compl.
MTS	Π_2^{P}-compl.	P-compl.	P-compl.	P-compl.	P-compl.
Impl	NP-compl.	P-compl.	P-compl.	P-compl.	P-compl.

Table 4.1. 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 further 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 [6] (see also [53]). 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 4.2 *Modal refinement between an implementation and a parameter-free PMTS is NP-complete.*

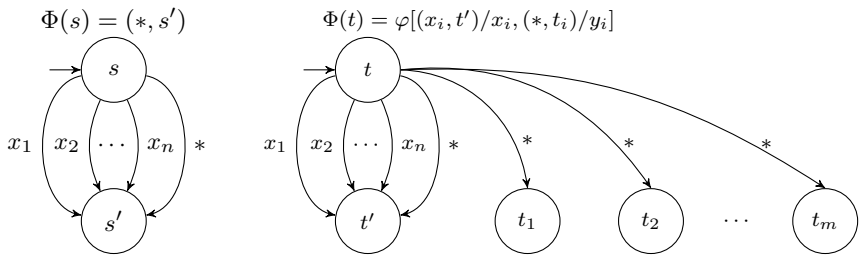
Proof The containment part is straightforward. First we guess the relation R. As s is an implementation then the set $\mathrm{Tran}_{\emptyset}(s)$ is a singleton. We thus only need to further guess $N \in \mathrm{Tran}_{\emptyset}(t)$ and then in polynomial time verify the two conjuncts in Definition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

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

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

Proposition 4.3 *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 $s \leq_{\mathrm{m}} t$ 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 $s \leq_\mathrm{m} t$.

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 there is no $N \in \mathrm{Tran}_\emptyset(t)$ that would contain exactly those (x_i, t') that correspond to (x_i, s') in M. Hence, $s \not\leq_\mathrm{m} t$. □

Proposition 4.4 *Modal refinement between two parameter-free PMTS is in* Π_2^P.

Proof We first note that the condition of Definition 2.3 can itself be verified in Π_2^P. Indeed, if we universally guess M and then existentially guess N, the rest of the condition is verifiable in polynomial time.

We could therefore, in principle, guess the relation R and then verify that the condition holds for all pairs $(s, t) \in R$. However, this would increase the complexity bound to Σ_3^P. Instead, we initially include all (polynomially many) pairs into R and for each pair check the condition. If it fails, we remove the pair and continue until we reach the greatest fixed point. This standard method is usually known as the coinductive computation of R.

However, a naive implementation of this idea would only prove that modal refinement is in Δ_3^P as we have a polynomial algorithm with Π_2^P oracle. To actually show the statement of the proposition, we describe a $\mathrm{coNP}^\mathrm{NP}$ algorithm, i.e. an algorithm with universal branching that uses an NP oracle.

Suppose that we want to check if $s \leq_\mathrm{m} t$. The algorithm works as follows:

1. Set $R := S \times S'$ (all pairs of states).
2. If R does not contain (s, t), reject.
3. Universally choose $(s', t') \in R$ and $M \in \mathrm{Tran}(s')$.
4. Using the NP-oracle, decide whether there exists $N \in \mathrm{Tran}(t')$ such that the refinement condition for (s', t') is satisfied.
5. If the result was false, remove (s', t') from R and go back to step 2. Otherwise, accept.

It may be clearly seen that the algorithm accepts iff $s \leq_\mathrm{m} t$. □

4.1.1 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 characterisation of refinement from Lemma 2.5. The following proposition determines the subclasses on which modal refinement can be decided in polynomial time.

Proposition 4.5 *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 (2.3) and (2.2). Clearly, (2.3) can be checked in P. We show that Condition (2.2) can be verified in P too. Recall that (2.2) 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 Condition (2.2) 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 these correspond to the clauses of $\Phi(s)$. Thus we get a polynomial time algorithm as shown in Algorithm 1.

Algorithm 1: Test for Condition (2.2) 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* $(a_1, s_1) \wedge \cdots \wedge (a_k, s_k)$ *in* $\Phi(s)$ **do**
  $N \leftarrow \{(a,t') \in T(t) \mid \exists i : a_i = a \wedge (s_i, t') \in R\}$;
  if $N \notin \mathrm{Tran}_\nu(t)$ **then return** false
return true;

Second, assume that the right-hand side is in pCNF. Note that Condition (2.2) can be equivalently stated as

$$\forall M : \mathrm{match}_t(M) \notin \mathrm{Tran}_\nu(t) \Rightarrow M \notin \mathrm{Tran}_\mu(s). \tag{4.1}$$

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 (2.2) 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* $(a_1, t_1) \vee \cdots \vee (a_k, t_k)$ *in* $\Phi(t)$ **do**
  $M \leftarrow T(s) \setminus \{(a,s') \in T(s) \mid \exists i : a_i = a \wedge (s', t_i) \in R\}$;
  if $M \in \mathrm{Tran}_\mu(s)$ **then return** false
return true;

The statement of the proposition thus follows. □

Proposition 4.6 *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 (2.1) and (2.2) of Lemma 2.5 can be checked in coNP. The computation of R is then done similarly to the algorithm described in the proof of Proposition 4.4. □

Proposition 4.7 *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 $\tilde{x}_1$, ..., $\tilde{x}_n$, change all occurrences of $\neg x_i$ into $\tilde{x}_i$ for all i, and add new clauses $(x_i \vee \tilde{x}_i)$ and $(\neg x_i \vee \neg \tilde{x}_i)$.

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 $\varphi' = \psi^+ \wedge \psi^-$ it is clear that φ' is satisfiable if and only if $(\psi^+ \Rightarrow \neg\psi^-)$ is not valid.

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

Proposition 4.8 *Modal refinement on parameter-free PMTS is coNP-hard, even if the right-hand side is an MTS.*

Proof We prove the hardness by a reduction from SAT to non-refinement. Let $\varphi(x_1, \ldots, x_n)$ be a Boolean formula (instance of SAT). We construct two parameter-free PMTSs $(S, T, \emptyset, \Phi)$ and $(S', T', \emptyset, \Phi')$ over $\Sigma = \{a, b\}$ as follows:

1. $S = \{s, s_1, \ldots, s_n\}$, $T = \{(s, a, s_i) \mid 1 \leq i \leq n\}$, $\Phi(s) = \varphi[(a, s_i)/x_i]$, $\Phi(s_i) = \mathbf{tt}$ for all i.
2. $S' = \{t\}$, $T' = \{(t, b, t)\}$, $\Phi'(t) = (b, t)$. (Clearly, this is an MTS.)

Let now φ be satisfiable and let $M \in \mathrm{Tran}_\emptyset(s)$ be arbitrary. Clearly, there can be no match $N \in \mathrm{Tran}_\emptyset(t)$ and thus $s \not\leq_{\mathrm{m}} t$.

Let now φ be unsatisfiable. This means that $\mathrm{Tran}_\emptyset(s) = \emptyset$ and the refinement condition trivially holds. Thus $s \leq_{\mathrm{m}} t$. □

4.2 Systems with Parameters

In the sequel we investigate the complexity of refinement checking in the general case of PMTS with parameters. The complexities are summarised in Table 4.2. We start with an observation of how the results on parameter-free systems can be applied to the parametric case.

Table 4.2: Complexity of modal refinement checking with parameters

	Boolean	positive	pCNF	pDNF
Boolean	Π_4^{P}-complete	Π_3^{P}-complete	Π_3^{P}-complete	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

Proposition 4.9 *The complexity upper bounds from Table 4.1 carry over to Table 4.2, 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.

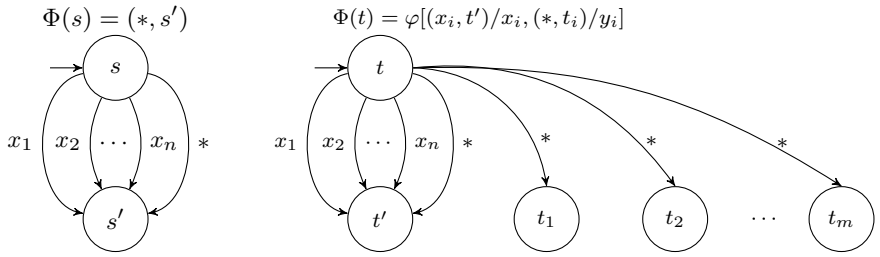
Proposition 4.10 *Modal refinement between an implementation and a right-hand side in positive CNF or in positive DNF is NP-hard.*

Proof The proof is by reduction from SAT. Let $\varphi(x_1, \ldots, x_n)$ be a formula in CNF and let $\varphi_1, \varphi_2, \ldots, \varphi_k$ be the clauses of φ. We construct two PMTSs (S, T, P, Φ) and (S', T', P', Φ') over the action alphabet $\Sigma = \{a_1, \ldots, a_k\}$ 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) $S' = \{t\} \cup \{t_i \mid 1 \leq i \leq k\}$, $T' = \{(t, a_i, t_i) \mid 1 \leq i \leq k\}$, $P' = \{x_1, \ldots, x_n\}$, $\Phi'(t) = \bigwedge_{1 \leq i \leq k}(a_i, t_i)$ 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 $s \leq_{\mathrm{m}} t$ if and only if φ is satisfiable. □

Proposition 4.11 *Modal refinement is* Σ_3^{P}*-hard even if the left-hand side is an MTS.*

Proof The proof is done using the construction of the proof of Proposition 4.3 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 $s \leq_{\mathrm{m}} t$ 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.



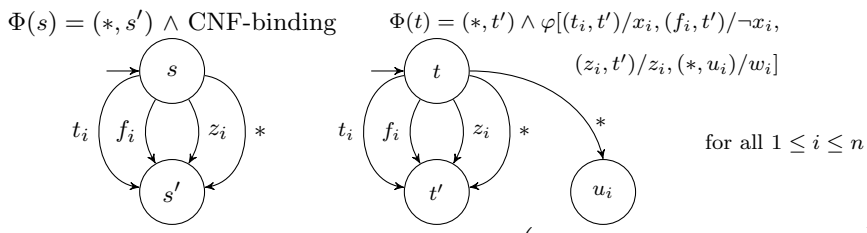
Assume that ψ is true. Then there exists a valuation ν on $\{z_1, \ldots, z_k\}$ such that $\forall x : \exists y : \varphi(x, y)$ is true. Using now the same argument as that in the proof of Proposition 4.3, we get that $s \leq_{\mathrm{m}} t$.

On the other hand assume that ψ is false and let ν be an arbitrary valuation on $\{z_1, \ldots, z_k\}$. We then may again use the reasoning in the proof of Proposition 4.3 to get that $s \not\leq_{\mathrm{m}} t$. □

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 4.12 *Modal refinement is* Π_4^{P}*-hard even if the left-hand side is in positive CNF.*

Proof Consider a $\mathrm{QBF}_4^{\forall}$ instance, a formula $\psi = \forall x \exists y \forall z \exists w : \varphi(x, y, z, w)$ with φ in CNF and x, y, z, w vectors of length n. We construct two systems s and t and use the variables $\{x_1, \ldots, x_n\}$ as parameters for the left-hand side system s, and $\{y_1, \ldots, y_n\}$ as parameters for the right-hand side system t.



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 $s \leq_{\mathrm{m}} t$. 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 $s \leq_{\mathrm{m}} t$. 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, we also prove the following two propositions.

Proposition 4.13 *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 $\forall x_1, \ldots, x_n \exists y_1, \ldots, y_m : \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$, where φ is in CNF. 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 $\Sigma = \{t_1, \ldots, t_n, f_1, \ldots, f_n\}$ be the set of actions. The systems (S, T, P, Φ) and (S', T', P', Φ') are built as follows: $S = \{s, s'\}$, $T = \{(s, t_i, s'), (s, f_i, s') \mid 1 \leq i \leq n\}$, $P = \{x_1, \ldots, x_n\}$, $\Phi(s) = \bigwedge_{1 \leq i \leq n}((x_i \Rightarrow (t_i, s')) \wedge (\neg x_i \Rightarrow (f_i, s'))$ (note that this may be written in positive CNF), $\Phi(s') = \mathbf{tt}$; $S' = \{t, t'\}$, $T' = \{(t, t_i, t'), (t, f_i, t') \mid 1 \leq i \leq n\}$, $P' = \{y_1, \ldots, y_m\}$, $\Phi'(t) = \varphi[(t_i, t')/x_i, (f_i, t')/\neg x_i]$, $\Phi'(t') = \mathbf{tt}$. We now claim that $\forall x \exists y : \varphi$ holds if and only if $s \leq_{\mathrm{m}} t$. We show the two implications separately.

Let first $\forall x \exists y : \varphi$ hold. Let $\mu \subseteq P$ 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 $\nu \subseteq P'$ 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 $s \leq_{\mathrm{m}} t$.

For the other implication let $\exists x \forall y : \neg\varphi$ hold. We show that $s \not\leq_{\mathrm{m}} t$. 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 $s \not\leq_{\mathrm{m}} t$. □

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 4.12. 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 4.14 *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 $s \leq_{\mathrm{m}} t$. □

The following 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. The binding works slightly differently, as with DNF we are unable to make a conjunction 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 4.15 *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 $s \leq_{\mathrm{m}} t$. The reasoning behind this fact is similar to the proof of Proposition 4.13. □

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

Proposition 4.16 *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 $\Sigma = \{t_1, \ldots, t_n, f_1, \ldots, f_n, \bullet\}$. The two systems (S, T, P, Φ) and (S', T', P', Φ') are built as follows: $S = \{s, s', s''\} \cup \{s_i \mid 1 \leq i \leq n\}$, $T = \{(s, \bullet, s')\} \cup \{(s', \bullet, s_i), (s_i, t_i, s''), (s_i, f_i, s'') \mid 1 \leq i \leq n\}$, $P = \{x_1, \ldots, x_n\}$, $\Phi(s) = (\bullet, s')$, $\Phi(s') = \bigwedge_i (\bullet, s_i)$, $\Phi(s_i) = (x_i \wedge (t_i, s'')) \vee (\neg x_i \wedge (f_i, s''))$, $\Phi(s'') = \mathbf{tt}$; $S' = \{t, t'\} \cup \{u_i, v_i \mid 1 \leq i \leq n\} \cup \{w_j \mid 1 \leq j \leq k\}$, $T' = \{(t, \bullet, w_j) \mid 1 \leq j \leq k\} \cup \{(w_j, \bullet, u_i), (w_j, \bullet, v_i) \mid 1 \leq i \leq n, 1 \leq j \leq k\} \cup \{(u_i, t_i, t'), (u_i, f_i, t'), (v_i, f_i, t'), (v_i, t_i, t') \mid 1 \leq i \leq n\}$, $P' = \{y_1, \ldots, y_n\}$, $\Phi'(t) = \bigwedge_j w_j$, $\Phi'(w_j) = \varphi'_j$ 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 $s \leq_{\mathrm{m}} t$. □

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

Proposition 4.17 *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 $\Sigma = \{t_1, \ldots, t_n, f_1, \ldots, f_n, z_1, \ldots, z_n, *, \bullet\}$ and we create the two systems (S, T, P, Φ), (S', T', P', Φ') over the action alphabet Σ as follows:

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

$S' = \{t, t'\} \cup \{u_i, v_i, w_i \mid 1 \leq i \leq n\}$, $T' = \{(t, z_i, t'), (t, \bullet, u_i), (t, \bullet, v_i), (t, *, w_i), (u_i, t_i, t'), (u_i, f_i, t'), (v_i, t_i, t'), (v_i, f_i, t') \mid 1 \leq i \leq n\} \cup \{(t, *, t')\}$, $P' = \{y_1, \ldots, y_n\}$, $\Phi'(t) = (*, t') \wedge \varphi[(\bullet, u_i)/x_i, (\bullet, v_i)/\neg x_i, (z_i, t')/z_i, (*, w_i)/w_i]$, 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 $s \leq_{\mathrm{m}} t$ if and only if $\forall x \exists y \forall z \exists w : \varphi(x, y, z, w)$. □

The following proposition solves the only remaining case.

Proposition 4.18 *Modal refinement is Π_3^{P}-hard even if the right-hand side is in positive CNF.*

Proof We prove the hardness by reduction from the validity of the $\mathrm{QBF}_3^{\forall}$ formula $\forall x \exists y \forall z : \varphi(x, y, z)$ where x, y, z are vectors of size n. We construct two PMTSs (S, T, P, Φ) and (S', T', P', Φ') as follows:

1. $P = \{x_1, \ldots, x_n\}$, $S = \{s, s', s_1, \ldots, s_n\}$,
   $T = \{(s, t_i, s'), (s, f_i, s'), (s, *, s_i) \mid 1 \leq i \leq n\} \cup \{(s, *, s')\}$,
   $\Phi(s) = \neg\varphi[(t_i, s')/y_i, (*, s_i)/z_i] \wedge \bigwedge_i ((t_i, s') \Leftrightarrow \neg(f_i, s')) \wedge (*, s')$,
   $\Phi(s') = \Phi(s_i) = \mathbf{tt}$ for all i.

2. $P' = \{y_1, \ldots, y_n\}$, $S' = \{t, t', u_1, \ldots, u_n, v_1, \ldots, v_n\}$, $T' = \{(t, *, t')\} \cup \{(t, t_i, t'), (t, f_i, t'), (t, t_i, u_i), (t, f_i, v_i), (u_i, b, t'), (v_i, b, t') \mid 1 \leq i \leq n\}$, $\Phi'(t) = \bigvee_i ((t_i, u_i) \vee (f_i, v_i))$, $\Phi'(u_i) = y_i \Rightarrow (b, t')$ for all i, $\Phi'(v_i) = \neg y_i \Rightarrow (b, t')$ for all i, $\Phi'(t') = \mathbf{tt}$.
(Clearly, this is in positive CNF. In fact, we only use disjunctions here.)

To make our first observation, we use the following notation: By $s \leq_{\mathrm{m}}^{\mu,\nu} t$ we denote the fact that there exists a modal refinement respecting μ and ν containing (s, t). We now observe that for every choice of $\mu \subseteq P$ and $\nu \subseteq P'$, $s' \leq_{\mathrm{m}}^{\mu,\nu} u_i$ iff $y_i \notin \nu$. Similarly, $s' \leq_{\mathrm{m}}^{\mu,\nu} v_i$ iff $y_i \in \nu$.

Assume now that $\forall x \exists y \forall z : \varphi(x, y, z)$ is true. We want to show that $s \leq_{\mathrm{m}} t$. Let $\mu \subseteq P$ be arbitrary. Let us fix the valuation $\hat{x}$ of x given by μ. We know that $\exists y \forall z : \varphi(\hat{x}, y, z)$ holds. Let us fix such valuation of y and call it $\hat{y}$. Let us further choose $\nu = \{y_i \mid \hat{y}_i = \mathbf{tt}\}$. Let now $M \in \mathrm{Tran}_\mu(s)$ be arbitrary. Clearly, $(t_i, s') \in M$ iff $(f_i, s') \notin M$. Furthermore M induces a valuation $\tilde{y}$ of y (via the choice of t_i/f_i) and $\tilde{z}$ of z (via the choice of $(*, s_i)$), such that $\neg\varphi(\hat{x}, \tilde{y}, \tilde{z})$ holds.

As we know that $\forall z : \varphi(\hat{x}, \hat{y}, z)$ holds, this means that $\hat{y}$ and $\tilde{y}$ differ. Therefore, there is at least one index j such that $\hat{y}_j \neq \tilde{y}_j$ and thus either $y_j \in \nu$ and $(f_j, s') \in M$ or $y_j \notin \nu$ and $(t_j, s') \in M$.

To show that $s \leq_{\mathrm{m}} t$, we use Lemma 2.5. Clearly, Condition (1) holds. We now show that $\mathrm{match}_t(M) \in \mathrm{Tran}_\nu(t)$. Clearly, $\mathrm{match}_t(M) = \{(*, t')\} \cup \{(t_i, t') \mid (t_i, s') \in M\} \cup \{(f_i, t') \mid (f_i, s') \in M\} \cup \{(t_i, u_i) \mid (t_i, s') \in M \text{ and } y_i \notin \nu\} \cup \{(f_i, v_i) \mid (f_i, s') \in M \text{ and } y_i \in \nu\}$. Due to our observation that $\tilde{y}$ and $\hat{y}$ differ, at least one of (t_i, u_i) or (f_i, v_i) is contained in $\mathrm{match}_t(M)$. Therefore, $\mathrm{match}_t(M) \in \mathrm{Tran}_\nu(t)$ and thus $s \leq_{\mathrm{m}} t$.

Let us now assume that $\forall x \exists y \forall z : \varphi(x, y, z)$ does not hold. We show that $s \not\leq_{\mathrm{m}} t$. Our assumption means that $\exists x \forall y \exists z : \neg\varphi(x, y, z)$ holds. Let us fix such a valuation $\hat{x}$ of x and choose $\mu = \{x_i \mid \hat{x}_i = \mathbf{tt}\}$. Let $\nu \subseteq P'$ be arbitrary and let us fix the valuation $\hat{y}$ of y represented by ν. Let us further fix a valuation $\hat{z}$ of z such that $\neg\varphi(\hat{x}, \hat{y}, \hat{z}) = \mathbf{tt}$. Choose $M = \{(t_i, s') \mid \hat{y}_i = \mathbf{tt}\} \cup \{(f_i, s') \mid \hat{y}_i \neq \mathbf{tt}\} \cup \{(*, s_i) \mid \hat{z}_i = \mathbf{tt}\} \cup \{(*, s')\}$. Clearly, $M \in \mathrm{Tran}_\mu(s)$.

We now show that $\mathrm{match}_t(M) \notin \mathrm{Tran}_\nu(t)$. To be in $\mathrm{Tran}_\nu(t)$ a set has to contain at least one of (t_i, u_i) or (f_i, v_i). However, $\mathrm{match}_t(M)$ does not contain any of these, as M contains (t_i, s') iff $y_i \in \nu$ and (f_i, s') iff $y_i \notin \nu$. (See our observation about $s' \leq_{\mathrm{m}}^{\mu,\nu} u_i$ and $s' \leq_{\mathrm{m}}^{\mu,\nu} v_i$ above.) Therefore, $s \not\leq_{\mathrm{m}} t$. □

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.

5 Modal Refinement Checking by Reduction to QBF

In this section, we show how to solve the modal refinement problem on parameter-free PMTS and on general PMTS using QBF solvers. Although modal refinement

is computationally hard (Π_2^{P}-complete on parameter-free PMTS and Π_4^{P}-complete on PMTS), by using QBF solvers we obtain a feasible method for modal refinement checking, as documented by our preliminary experiments.

As mentioned, in order to decide whether modal refinement holds between two states, a reduction to a quantified Boolean formula will be used. Recall the definition of the problem QBF_n^Q where $Q \in \{\exists, \forall\}$ given in Definition 4.1.

5.1 Construction for parameter-free PMTS

Due to the completeness of QBF problems and our result in Proposition 4.4, it is possible to polynomially reduce modal refinement on parameter-free PMTS to $\mathrm{QBF}_2^{\forall}$. However, following the construction in Proposition 4.4, we have to perform fixed-point iterations in order to compute the refinement relation, resulting in numerous invocations of the external QBF solver. Additionally, this approach is not applicable in the PMTS case. We shall instead provide a direct reduction of modal refinement checking to $\mathrm{QBF}_3^{\exists}$.

Let $s \in S_1$ and $t \in S_2$ be states of two arbitrary parameter-free PMTSs $\mathcal{M}_1 = (S_1, T_1, \emptyset, \Phi_1)$ and $\mathcal{M}_2 = (S_2, T_2, \emptyset, \Phi_2)$. Furthermore let

$$Ap = \underbrace{(S_1 \times S_2)}_{X_R} \uplus \underbrace{T_1}_{X_{T_1}} \uplus \underbrace{(S_1 \times T_2)}_{X_{T_2}}$$

be a set of atomic propositions. The intended meaning is that $(u, v) \in X_R$ is assigned $\mathtt{tt}$ if and only if it is also contained in the modal refinement relation R, while X_{T_1} and X_{T_2} encode transitions. A state from S_1 is attached to the set T_2, because the variables used to encode $N \in \mathrm{Tran}(t)$ with $t \in S_2$ must be unique for different states of S_1 in order to move the $\exists$ quantification over conjuncts in the formula.

We now construct a formula $\Psi_{s,t} \in \mathcal{B}(Ap)$ satisfying

$$s \leq_m t \quad \textit{iff} \quad \exists X_R \forall X_{T_1} \exists X_{T_2} \Psi_{s,t} \in \mathrm{QBF}_3^{\exists} \,. \tag{5.1}$$

To this end, we shall use a macro $\psi_{u,v}$ capturing the condition which has to be satisfied by any element $(u, v) \in R$. Furthermore, we ensure that (s, t) is assigned $\mathtt{tt}$ by every satisfying assignment for the formula by placing it directly in the conjunction:

$$\Psi_{s,t} = (s, t) \wedge \bigwedge_{(u,v) \in X_R} \big((u, v) \Rightarrow \psi_{u,v}\big) \,. \tag{5.2}$$

It remains to define the formula $\psi_{u,v}$ that should guarantee that $(u, v) \in R$. We start with the modal refinement condition as a blueprint:

$$\begin{aligned} \forall M \in \mathrm{Tran}(u) : \exists N \in \mathrm{Tran}(v) : \;& \forall (a, u') \in M : \exists (a, v') \in N : (u', v') \in R \;\wedge \\ & \forall (a, v') \in N : \exists (a, u') \in M : (u', v') \in R \,. \end{aligned}$$

As M and N are subsets of $T_1(u)$ and $T_2(v)$, respectively, and are finite, the inner quantifiers can be expanded causing only a polynomial growth of the formula size. Further, Tran sets are replaced by the original definition and the outer quantifiers

are moved in front of $\Psi_{s,t}$. As the state obligations are defined over a different set of atomic propositions ($\Phi(v) \in \mathcal{B}((\Sigma \times S) \cup P) \not\subseteq \mathcal{B}(Ap)$), a family of mapping functions π_x is introduced where x is either a state from S_1 or a pair of states from $S_1 \times S_2$.

$$\begin{aligned} \pi_x : \mathcal{B}(\Sigma \times S) &\to \mathcal{B}(Ap) \\ \mathbf{tt} &\mapsto \mathbf{tt} \\ (a, v) &\mapsto (x, a, v) \quad \textit{for } a \in \Sigma,\, v \in S \\ \neg\varphi &\mapsto \neg\, \pi_x(\varphi) \\ \varphi_1 \wedge \varphi_2 &\mapsto \pi_x(\varphi_1) \wedge \pi_x(\varphi_2) \\ \varphi_1 \vee \varphi_2 &\mapsto \pi_x(\varphi_1) \vee \pi_x(\varphi_2) \end{aligned} \tag{5.3}$$

Applying these steps to the blueprint yields the following result:

$$\psi_{u,v} = \pi_u\left(\Phi_1\left(u\right)\right) \Rightarrow \pi_{u,v}\left(\Phi_2\left(v\right)\right) \wedge \varphi_{u,v} \tag{5.4}$$

$$\begin{aligned} \varphi_{u,v} = &\bigwedge_{\substack{u^* \in X_{T_1} \\ u^* = (u,a,u')}} \left(u^* \Rightarrow \bigvee_{\substack{v^* \in X_{T_2} \\ v^* = (u,v,a,v')}} \left(v^* \wedge \left(u', v'\right)\right)\right) \\ &\wedge \bigwedge_{\substack{v^* \in X_{T_2} \\ v^* = (u,v,a,v')}} \left(v^* \Rightarrow \bigvee_{\substack{u^* \in X_{T_1} \\ u^* = (u,a,u')}} \left(u^* \wedge \left(u', v'\right)\right)\right) . \end{aligned} \tag{5.5}$$

Theorem 5.1 *For states s, t of a parameter-free PMTS, we have*

$$s \leq_m t \quad \textit{iff} \quad \exists X_R \forall X_{T_1} \exists X_{T_2} \Psi_{s,t} \in \mathrm{QBF}_3^{\exists} .$$

Before proving the soundness and the correctness of the construction for parameter-free PMTS, a lemma is introduced to simplify this proof.

Lemma 5.2 *Let $(u, v) \in S_1 \times S_2$ be a pair of states. Let $\mathcal{A}_{X_R}$, $\mathcal{A}_{X_{T_1}}$ and $\mathcal{A}_{X_{T_2}}$ be partial valuations for the respective sets of atomic propositions that appear in the indices. Furthermore let $R \subseteq S_1 \times S_2$, $M \in \mathrm{Tran}_\emptyset(u)$ and $N \in \mathrm{Tran}_\emptyset(v)$ such that $\mathcal{A}_{X_R} = R$, $\mathcal{A}_{X_{T_1}} \supseteq \pi_u(M)$ and $\mathcal{A}_{X_{T_2}} \supseteq \pi_{u,v}(N)$. Then $\mathcal{A}_{X_R} \cup \mathcal{A}_{X_{T_1}} \cup \mathcal{A}_{X_{T_2}} \models \varphi_{u,v}$ if and only if*

$$\begin{aligned} R \cup M \cup N \models \quad & \forall (a, s') \in M : \exists (a, t') \in N : (s', t') \in R \\ \wedge \quad & \forall (a, t') \in N : \exists (a, s') \in M : (s', t') \in R . \end{aligned}$$

Proof We assume the conditions of the lemma, the set $\mathcal{A}_X = \mathcal{A}_{X_R} \cup \mathcal{A}_{X_{T_1}} \cup \mathcal{A}_{X_{T_2}}$ and $\mathcal{A}_R = R \cup M \cup N$. Additionally, we only consider one half of the conjunction, as the other one is proven analogously.

$$\begin{aligned} & \mathcal{A}_R \models \quad \forall (a, s') \in M : \exists (a, t') \in N : (s', t') \in R \\ \textit{iff} \quad & \mathcal{A}_R \models \bigwedge_{(a,s') \in T_1(s)} \left((a, s') \in M \Rightarrow \bigvee_{(a,t') \in T_2(t)} \left((a, t') \in N \wedge \left((s', t') \in R\right)\right)\right) \\ \textit{iff} \quad & \mathcal{A}_X \models \bigwedge_{\substack{s^* \in X_{T_1} \\ s^* = (s,a,s')}} \left(s^* \Rightarrow \bigvee_{\substack{t^* \in X_{T_2} \\ t^* = (s,t,a,t')}} \left(t^* \wedge \left(s', t'\right)\right)\right) \end{aligned}$$

As M and N are finite sets, $\forall$ and $\exists$ quantifiers may simply be expanded. In the second step, we apply π_x and substitute $\in$ with atomic propositions. □

Soundness and Completeness of Theorem 5.1 For soundness, assume $s \leq_m t$ with the modal refinement relation R. As the partial valuation for X_R, we set $\mathcal{A}_{X_R} = R$. Furthermore let $\mathcal{A}_{X_{T_1}} \subseteq X_{T_1}$ be an arbitrary assignment. We now construct an assignment $\mathcal{A}_{X_{T_2}}$, such that

$$\mathcal{A} = \mathcal{A}_{X_R} \cup \mathcal{A}_{X_{T_1}} \cup \mathcal{A}_{X_{T_2}} \models \Psi_{s,t}$$

holds. Without adding anything to $\mathcal{A}_{X_{T_2}}$, clearly $\mathcal{A} \models (s,t)$ and $\mathcal{A} \models (u,v) \Rightarrow \psi_{u,v}$ for all $(u,v) \in X_R \cap \overline{R}$ hold.

Let now $(u,v) \in R$ be an arbitrary pair of states. If $\mathcal{A} \not\models \pi_u(\Phi(u))$, then $\mathcal{A} \models \psi_{u,v}$ and $\mathcal{A} \models (u,v) \Rightarrow \psi_{u,v}$. Hence we assume now $\mathcal{A} \models \pi_u(\Phi(u))$. Since $(u,v) \in R$, there exists for all $M \in \mathrm{Tran}_\emptyset(u)$ a set N, such that the condition holds, which is included in the assignment $\mathcal{A}_{X_{T_2}} \supseteq \pi_{u,v}(N)$. This can safely be done due to the prefixing and with Lemma 5.2 we get $\mathcal{A} \models \varphi_{u,v}$ and $\mathcal{A} \models (u,v) \Rightarrow \psi_{u,v}$.

As a valuation $\mathcal{A}$ can be constructed for a fixed modal refinement relation, such that for all subsets of X_{T_1} it satisfies the formula, $\exists X_R \forall X_{T_1} \exists X_{T_2} \Psi_{s,t} \in \mathrm{QBF}_3^\exists$ holds.

For completeness, we now assume

$$\exists X_R \forall X_{T_1} \exists X_{T_2} \Psi_{s,t} \in \mathrm{QBF}_3^\exists \ .$$

Then there exists a partial valuation $\mathcal{A}_{X_R} \subseteq \mathcal{A}$ for X_R, which satisfies $\Psi_{s,t}$. R is simply constructed by setting $R = \mathcal{A}_{X_R}$. Clearly $(s,t) \in R$. Let now $(u,v) \in R$ be an arbitrary pair of states. As (5.4) is satisfied for this pair, either $\Phi(u)$ is unsatisfiable and there simply exists no $M \in \mathrm{Tran}_\emptyset(s)$ or for the chosen $M = \pi_u^{-1}(\mathcal{A}_{X_{T_1}})$ exists a $N = \pi_{u,v}^{-1}(\mathcal{A}_{X_{T_2}})$. By Lemma 5.2 the modal refinement condition holds for this arbitrary pair. Hence R is a modal refinement relation.

Finally, we show that the reduction indeed takes only polynomial time. For this observe that (5.5) is in $\mathcal{O}(|T_1(u)||T_2(v)|)$. Therefore (5.4) is in $\mathcal{O}(|T_1(u)||T_2(v)| + |\Phi_1(u)| + |\Phi_2(v)|)$. Leading to a total formula size of

$$\mathcal{O}\Big(|S_1||S_2|\big(|T_1||T_2| + |\Phi_1| + |\Phi_2|\big)\Big) \ .$$

5.2 Construction for PMTS

We now reduce the modal refinement on PMTS to $\mathrm{QBF}_4^\forall$, which corresponds directly to the complexity result for this problem proved in the present article. Nevertheless, due to the first existential quantification in $\forall\exists\forall\exists$ alternation sequence, we can still guess the refinement relation using the QBF solver rather than to do the lengthy fixed-point computation.

In the PMTS case, we have to find for all parameter valuations for the system of s a valuation for the system of t, such that there exists a modal refinement relation containing (s,t). We simply choose universally a valuation for the parameters of the left system (the underlying system of s) and then existentially for the right system (the underlying system of t). Prior to checking modal refinement, the valuations

are fixed, so the PMTS becomes a parameter-free PMTS. This is accomplished by extending Ap with P_1 and P_2 and adding the necessary quantifiers to the formula. Thus we obtain the following:

Theorem 5.3 *For states s, t of a PMTS, we have*

$$s \leq_m t \quad \textit{iff} \quad \forall P_1 \exists P_2 \exists X_R \forall X_{T_1} \exists X_{T_2} \Psi_{s,t} \in \mathrm{QBF}_4^{\forall} \ .$$

5.3 Experimental Results

We now show how our method performs in practice by implementing the reduction and linking it to the QBF solver Quantor[2]. In order to evaluate whether our solution scales, we generated several random samples of MTS and parametric MTS with a different number of parameters ranging from 0 to 20 (as displayed in Table 5.1 in parentheses). The systems use two different actions and the edges were generated randomly so that they create a tree with some additional "noise" edges to reach the specified number of outgoing transitions. In order to simulate a step-wise refinement, we took these randomly generated systems and performed a few randomised operations, such as insertion and removal of actions, states and transitions, as well as changes to the transition labels or to the obligation function of a state. Table 5.1 is split according to the instances where the modal refinement (after a few randomised changes were performed) remained satisfied and where it became unsatisfied. The degree of the random graph is the number of outgoing transitions from each state. Given the fixed two-letter action alphabet, the higher the degree the more nondeterminism is present in the generated systems.

The entries in the tables are average running times in seconds with around 90% of the time used by the external QBF solver. The standard deviation in our experiments ranged from 10 to 60%, a few exceptions are noted with a footnote. The reason why the refinement checking takes occasionally significantly more time is likely due to the fact that the generated graphs are hard instances for the QBF solver (for the chosen heuristic). The experiments were performed on an Intel Core i7 (2.7 Ghz) with 16 GB RAM using Java 1.8.

On the one hand, we can observe that the number of parameters does not generally play any major role in the running time, excluding some rare cases. The running times on PMTS with 20 parameters are close to parameter-free PMTS. Therefore, the greatest theoretical complexity threat—the number of parameters requiring in general a search through exponentially many combinations—seems to be in practice eliminated by the use of QBF solvers.

On the other hand, observe that the running time is affected by the level of nondeterminism (degree). This effect is more emphasised as the number of parameters increases. However, the level of nondeterminism in many applications is often quite low [18], hence this dependency does not pose a serious problem in practice.

The method is implemented within the tool MoTraS [37], which is equipped with a graphical as well as a command-line interface. More information on the tool and the data used to perform this experimental evaluation can be found at `https://www7.in.tum.de/~kretinsk/motras.html`.

[2] `http://fmv.jku.at/quantor/`

Table 5.1: Experimental results: degree is the number of outgoing transitions and the size of the problem is the number of states

degree 2 (satisfied)	50	100	150	200	250	300
MTS	0.17	0.36	0.82	1.44	2.31	3.4
PMTS (0)	0.08	0.34	0.79	1.45	2.42	3.23
PMTS (1)	0.08	0.39	0.93	1.69	2.70	3.82
PMTS (5)	0.09	0.42	0.92	1.79	2.79	3.91
PMTS (20)	0.09	0.57	1.11	4.53	32.38 [3]	37.85 [4]

degree 2 (unsatisfied)	50	100	150	200	250	300
MTS	0.09	0.36	0.81	1.50	2.34	3.21
PMTS (0)	0.08	0.36	0.78	1.51	2.38	3.43
PMTS (1)	0.09	0.39	0.82	1.62	2.71	3.74
PMTS (5)	0.09	0.39	0.89	1.66	2.83	3.80
PMTS (20)	0.09	0.42	0.97	2.01	2.93	4.32

degree 3 (satisfied)	50	100	150	200	250	300
MTS	0.31	0.94	2.22	4.03	6.59	9.48
PMTS (0)	0.20	0.91	2.23	4.00	6.65	9.56
PMTS (1)	0.27	1.04	2.73	4.60	7.79	11.61
PMTS (5)	0.24	1.10	2.73	5.05	8.21	12.19
PMTS (20)	0.34	1.34	3.06	6.34	11.21	227.55 [5]

degree 3 (unsatisfied)	50	100	150	200	250	300
MTS	0.20	0.82	2.13	3.79	6.36	8.85
PMTS (0)	0.21	0.87	2.17	4.05	6.59	8.85
PMTS (1)	0.26	0.99	2.48	4.55	7.46	10.99
PMTS (5)	0.28	1.13	2.83	4.96	8.42	11.63
PMTS (20)	0.28	1.11	3.22	5.83	8.98	12.69

degree 5 (satisfied)	50	100	150	200	250	300
MTS	0.79	2.89	6.69	12.88	19.79	30.27
PMTS (0)	0.89	3.54	8.64	16.83	26.29	38.27
PMTS (1)	2.24	8.86	22.02	41.94	81.86	113.59
PMTS (5)	3.32	8.64	16.91	38.17	58.31	95.79
PMTS (20)	1.42	7.59	21.69	80.29 [6]	61.84	116.24

degree 5 (unsatisfied)	50	100	150	200	250	300
MTS	0.72	2.81	6.50	12.45	19.45	29.58
PMTS (0)	0.83	4.23	8.51	14.89	25.33	37.79
PMTS (1)	1.79	9.25	29.66	56.53	69.69	83.31
PMTS (5)	2.89	7.60	20.14	42.45	64.86	89.23
PMTS (20)	1.21	6.87	20.75	40.51	47.54	97.75

[3] Standard deviation: 40.29
[4] Standard deviation: 47.56
[5] Standard deviation: 197.36
[6] Standard deviation: 96.79

6 Conclusion

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 specialises to the well-known refinements already studied on the subclasses of PMTS. We provided a comprehensive overview of complexity of refinement checking on PMTS and its subclasses, showed a direct encoding of the problem into QBF and discussed its applicability on preliminary experiments with random graphs. The conclusion we can draw is that the refinement checking scales well with the growing number of parameters but it is more sensitive to the degree of nondeterminism. A more thorough experimental evaluation on larger case studies is left as future work.

Acknowledgements We would like to thank to Sebastian Bauer for suggesting the traffic light example and for allowing us to use his figure environments. The research leading to the results in this article has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under Grant Agreement nr. 318490 (SENSATION) and Grant Agreement nr. 601148 (CASSTING), from the Sino-Danish Basic Research Center IDEA4CPS funded by the Danish National Research Foundation and the National Science Foundation China. The research was further funded in part by the European Research Council (ERC) under the grant agreement 267989 (QUAREM), by the Austrian Science Fund (FWF) project S11402-N23 (RiSE) and by the Czech Science Foundation grant No. P202/12/G061. Nikola Beneš has been supported by the MEYS project No. CZ.1.07/2.3.00/30.0009 Employment of Newly Graduated Doctors of Science for Scientific Excellence.

References

1. Aceto, L., Fábregas, I., de Frutos-Escrig, D., Ingólfsdóttir, A., Palomino, M.: Graphical representation of covariant-contravariant modal formulae. In: EXPRESS, pp. 1–15 (2011)
2. de Alfaro, L., Henzinger, T.A.: Interface automata. In: ESEC / SIGSOFT FSE, pp. 109–120 (2001)
3. Alur, R., Henzinger, T.A., Kupferman, O., Vardi, M.Y.: Alternating refinement relations. In: CONCUR, pp. 163–178 (1998)
4. Antonik, A., Huth, M., Larsen, K.G., Nyman, U., Wasowski, A.: 20 years of modal and mixed specifications. Bulletin of the EATCS no. 95 pp. 94–129 (2008)
5. Antonik, A., Huth, M., Larsen, K.G., Nyman, U., Wasowski, A.: Complexity of decision problems for mixed and modal specifications. In: Proceedings of the 11th International Conference on Foundations of Software Science and Computation Structures (FOSSACS'08), *LNCS*, vol. 4962, pp. 112–126 (2008)
6. Balcazar, J.L., Gabarró, J., Santha, M.: Deciding bisimilarity is P-complete. Formal aspects of computing **4**(6 A), 638–648 (1992)
7. Bauer, S.S., Fahrenberg, U., Juhl, L., Larsen, K.G., Legay, A., Thrane, C.R.: Quantitative refinement for weighted modal transition systems. In: MFCS, *LNCS*, vol. 6907, pp. 60–71. Springer (2011)
8. Beneš, N., Černá, I., Křetínský, J.: Disjunctive modal transition systems and generalized LTL model checking. Technical report FIMU-RS-2010-12, Faculty of Informatics, Masaryk University, Brno (2010)

9. Beneš, N., Delahaye, B., Fahrenberg, U., Křetínský, J., Legay, A.: Hennessy-Milner logic with greatest fixed points as a complete behavioural specification theory. In: P.R. D'Argenio, H.C. Melgratti (eds.) CONCUR, *Lecture Notes in Computer Science*, vol. 8052, pp. 76–90. Springer (2013)
10. Beneš, N., Křetínský, J., Larsen, K.G., Møller, M.H., Srba, J.: Parametric modal transition systems. Technical report FIMU-RS-2011-03, Faculty of Informatics, Masaryk University, Brno (2011)
11. Beneš, N., Křetínský, J., Larsen, K.G., Srba, J.: EXPTIME-completeness of thorough refinement on modal transition systems. Inf. Comput. **218**, 54–68 (2012)
12. Beneš, N., Křetínský, J.: Process algebra for modal transition systemses. In: L. Matyska, M. Kozubek, T. Vojnar, P. Zemcík, D. Antos (eds.) MEMICS, *OASICS*, vol. 16, pp. 9–18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany (2010)
13. Beneš, N., Křetínský, J., Larsen, K., Srba, J.: EXPTIME-completeness of thorough refinement on modal transition systems. Information and Computation **218**, 54–68 (2012)
14. Beneš, N., Křetínský, J., Larsen, K.G., Møller, M.H., Srba, J.: Parametric modal transition systems. In: ATVA, pp. 275–289 (2011)
15. Beneš, N., Křetínský, J., Larsen, K.G., Møller, M.H., Srba, J.: Dual-priced modal transition systems with time durations. In: LPAR, pp. 122–137 (2012)
16. Beneš, N., Křetínský, J., Larsen, K.G., Srba, J.: Checking thorough refinement on modal transition systems is EXPTIME-complete. In: Theoretical Aspects of Computing - ICTAC 2009, 6th International Colloquium. Proceedings, *LNCS*, vol. 5684. Springer (2009)
17. Beneš, N., Křetínský, J., Larsen, K.G., Srba, J.: On determinism in modal transition systems. Theoretical Computer Science **410**(41), 4026–4043 (2009)
18. Beneš, N., Křetínský, J., Larsen, K.G., Srba, J.: On determinism in modal transition systems. Theor. Comput. Sci. **410**(41), 4026–4043 (2009)
19. Beneš, N., Černá, I., Křetínský, J.: Modal transition systems: Composition and LTL model checking. In: ATVA, pp. 228–242 (2011)
20. Bertrand, N., Legay, A., Pinchinat, S., Raclet, J.B.: Modal event-clock specifications for timed component-based design. Sci. Comput. Program. **77**(12), 1212–1234 (2012). DOI 10.1016/j.scico.2011.01.007
21. Boudol, G., Larsen, K.G.: Graphical versus logical specifications. In: CAAP, pp. 57–71 (1990)
22. Boudol, G., Larsen, K.G.: Graphical versus logical specifications. Theor. Comput. Sci. **106**(1), 3–20 (1992)
23. Caillaud, B., Delahaye, B., Larsen, K.G., Legay, A., Pedersen, M.L., Wasowski, A.: Compositional design methodology with constraint markov chains. In: QEST, pp. 123–132 (2010)
24. Campetelli, A., Gruler, A., Leucker, M., Thoma, D.: *Don't Know* for multi-valued systems. In: ATVA, pp. 289–305 (2009)
25. Čerāns, K., Godskesen, J.C., Larsen, K.G.: Timed modal specification - theory and tools. In: CAV, pp. 253–267 (1993)
26. Dams, D., Gerth, R., Grumberg, O.: Abstract interpretation of reactive systems. ACM Trans. Program. Lang. Syst. **19**(2), 253–291 (1997)
27. Dams, D., Namjoshi, K.S.: The existence of finite abstractions for branching time model checking. In: LICS, pp. 335–344 (2004)

28. David, A., Larsen, K.G., Legay, A., Nyman, U., Wasowski, A.: ECDAR: An environment for compositional design and analysis of real time systems. In: ATVA, pp. 365–370 (2010)
29. Fecher, H., Schmidt, H.: Comparing disjunctive modal transition systems with an one-selecting variant. J. of Logic and Alg. Program. **77**(1-2), 20–39 (2008)
30. Fecher, H., Steffen, M.: Characteristic mu-calculus formulas for underspecified transition systems. ENTCS **128**(2), 103–116 (2005)
31. Godefroid, P., Huth, M., Jagadeesan, R.: Abstraction-based model checking using modal transition systems. In: Proc. CONCUR'01, *LNCS*, vol. 2154, pp. 426–440. Springer (2001)
32. Godefroid, P., Nori, A.V., Rajamani, S.K., Tetali, S.: Compositional may-must program analysis: unleashing the power of alternation. In: POPL, pp. 43–56 (2010)
33. Gruler, A., Leucker, M., Scheidemann, K.D.: Modeling and model checking software product lines. In: G. Barthe, F.S. de Boer (eds.) FMOODS, *Lecture Notes in Computer Science*, vol. 5051, pp. 113–131. Springer (2008)
34. Huth, M., Jagadeesan, R., Schmidt, D.A.: Modal transition systems: A foundation for three-valued program analysis. In: Proc. of ESOP'01, *LNCS*, vol. 2028, pp. 155–169. Springer (2001)
35. Jacobs, B., Poll, E.: A logic for the java modeling language JML. In: FASE, pp. 284–299 (2001)
36. Juhl, L., Larsen, K.G., Srba, J.: Modal transition systems with weight intervals. J. Log. Algebr. Program. **81**(4), 408–421 (2012)
37. Křetínský, J., Sickert, S.: MoTraS: A tool for modal transition systems and their extensions. In: D.V. Hung, M. Ogawa (eds.) ATVA, *Lecture Notes in Computer Science*, vol. 8172, pp. 487–491. Springer (2013). Tool accessible at `https://www7.in.tum.de/~kretinsk/motras.html`
38. Křetínský, J., Sickert, S.: On refinements of Boolean and parametric modal transition systems. In: Z. Liu, J. Woodcock, H. Zhu (eds.) ICTAC, *Lecture Notes in Computer Science*, vol. 8049, pp. 213–230. Springer (2013)
39. Křetínský, J., Sickert, S.: On refinements of Boolean and parametric modal transition systems. CoRR **abs/1304.5278** (2013)
40. Larsen, K.G., Nyman, U., Wasowski, A.: Modal I/O automata for interface and product line theories. In: ESOP, pp. 64–79 (2007)
41. Larsen, K.G., Nyman, U., Wasowski, A.: On modal refinement and consistency. In: Proc. of CONCUR'07, *LNCS*, vol. 4703, pp. 105–119. Springer (2007)
42. Larsen, K.G., Thomsen, B.: A modal process logic. In: LICS, pp. 203–210. IEEE Computer Society (1988)
43. Larsen, K.G., Xinxin, L.: Equation solving using modal transition systems. In: LICS, pp. 108–117. IEEE Computer Society (1990)
44. Liskov, B., Wing, J.M.: A behavioral notion of subtyping. ACM Trans. Program. Lang. Syst. **16**(6), 1811–1841 (1994)
45. Lüttgen, G., Vogler, W.: Modal interface automata. Logical Methods in Computer Science **9**(3) (2013). DOI 10.2168/LMCS-9(3:4)2013. URL `http://dx.doi.org/10.2168/LMCS-9(3:4)2013`
46. Lynch, N.: I/O automata: A model for discrete event systems. In: 22nd Annual Conference on Information Sciences and Systems, pp. 29–38. Princeton University (1988)

47. Namjoshi, K.S.: Abstraction for branching time properties. In: CAV, pp. 288–300 (2003)
48. Nanz, S., Nielson, F., Nielson, H.R.: Modal abstractions of concurrent behaviour. In: Proc. of SAS'08, *LNCS*, vol. 5079, pp. 159–173. Springer (2008)
49. Papadimitriou, C.H.: Computational complexity. Addison-Wesley Publishing Co., Inc., Reading, MA, USA (1994)
50. Raclet, J.B.: Quotient de spécifications pour la réutilisation de composants. Ph.D. thesis, Université de Rennes I (2007). (In French)
51. Raclet, J.B., Badouel, E., Benveniste, A., B.Caillaud, Legay, A., Passerone, R.: A modal interface theory for component-based design. Fundamenta Informaticae **108**(1-2), 119–149 (2011)
52. Raclet, J.B., Badouel, E., Benveniste, A., Caillaud, B., Passerone, R.: Why are modalities good for interface theories? In: ACSD, pp. 119–127. IEEE (2009)
53. Sawa, Z., Jančar, P.: Behavioural equivalences on finite-state systems are PTIME-hard. Computing and informatics **24**(5), 513–528 (2005)
54. Uchitel, S., Chechik, M.: Merging partial behavioural models. In: Proc. of FSE'04, pp. 43–52. ACM (2004)