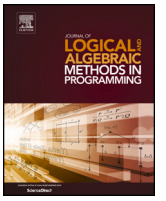
J. Log. Algebraic Methods Program. 138 (2024) 100954

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

journal homepage: www.elsevier.com/locate/jlamp



Coherent modal transition systems refinement

Davide Basile [a,*], Maurice H. ter Beek [a], Alessandro Fantechi [a,b], Stefania Gnesi [a]

[a] Formal Methods and Tools lab, ISTI–CNR, Via G. Moruzzi 1, Pisa, 56124, Italy
[b] DINFO, University of Florence, Via S. Marta 3, Firenze, 50139, Italy

A B S T R A C T

Modal Transition Systems (MTS) are a well-known formalism that extend Labelled Transition Systems (LTS) with the possibility of specifying necessary and permitted behaviour. Coherent MTS (CMTS) have been introduced to model Software Product Lines (SPL) based on a correspondence between the necessary and permitted modalities of MTS transitions and their associated actions, and the core and optional features of SPL. In this paper, we address open problems of the coherent fragment of MTS and introduce the notions of refinement and thorough refinement of CMTS. Most notably, we prove that refinement and thorough refinement coincide for CMTS, while it is known that this is not the case for MTS. We also define (thorough) equivalence and strong bisimilarity of both MTS and CMTS. We show their relations and, in particular, we prove that also strong bisimilarity and equivalence coincide for CMTS, whereas they do not for MTS. Finally, we extend our investigation to CMTS equipped with Constraints (MTSC), originally introduced to express alternative behaviour, and we prove that novel notions of refinement and strong thorough refinement coincide for MTSC, and so do their extensions to strong (thorough) equivalence and strong bisimilarity.

1. Introduction

Modal Transition Systems (MTS) [1] extend Labelled Transition Systems (LTS) [2] by distinguishing two types of transitions, meant to describe necessary and optional behaviour in a system specification by means of transitions that must *necessarily* be implemented and transitions that may *optionally* be implemented. MTS come with a concept of *refinement*, which represents a step of the design process, namely the one in which some optional behaviour is discarded while other optional behaviour becomes necessary. Stepwise refinement of an MTS eventually results in an *implementation*, which is an LTS in which no further refinement is possible. Refinement and a notion of *equivalence* of MTS are critical for enabling formal reasoning on the correctness of a system's design and implementation. Refinement enables gradually refining an abstract specification into a concrete one, ensuring that each step is correct, while equivalence allows comparing different specifications to determine if they are functionally equivalent, ensuring they behave in the same way (also regarding optional and necessary behaviour). MTS are a well-known specification theory and significant advances have been made so far [3,4].

It is known that the (modal) refinement of MTS is not complete, as amply discussed in the literature (cf., e.g., [5]). In other words, there are cases in which two MTS are not in a refinement relation although the set of implementations of one MTS is included in the set of implementations of the other MTS (the latter relation is called *thorough refinement*). Furthermore, while determining MTS refinement can be computed in polynomial time, determining thorough refinement of MTS requires EXPTIME [6].

MTS were recognised in [7–9] as a promising formalism to describe in a compact way all possible behaviour of the products of a Software Product Line (SPL). An SPL concerns a product line (or family) of closely related, yet customisable products (or variants) identified by relevant features (i.e., user distinguishable units of functionality) of the product domain [10–13].

* Corresponding author. First author.
E-mail address: davide.basile@isti.cnr.it (D. Basile).

https://doi.org/10.1016/j.jlamp.2024.100954
Received 14 August 2023; Received in revised form 20 February 2024; Accepted 20 February 2024

Available online 28 February 2024

Several variants of MTS have been proposed in the literature to express behavioural variability typical of SPL [14–20]. In this paper, we consider one of them and refer to it as *Coherent MTS* (CMTS) [19]. In CMTS, the actions that label transitions are identified as features of an SPL. The association of features with actions can be traced back to [7–9,14]. Since a feature cannot be both mandatory and optional, it follows that transitions of CMTS labelled with the same action must also have the same modality. This restriction is coined *coherence* [19]. As a consequence, when an optional action is discarded in an implementation, all transitions labelled with that action must be discarded in that implementation.

In this paper, we introduce notions of (thorough) refinement, (thorough) equivalence and strong bisimilarity for such CMTS [19], and address their mutual relations. Furthermore, we extend these relations to CMTS equipped with Constraints (MTSC), originally introduced as MTSυ in [19] to express alternative behaviour. Fig. 1 illustrates our main contributions. Most notably, refinement of CMTS is proved to be sound *and* complete, while refinement is known to be sound but not complete for MTS (cf., e.g., [5,6]). We summarise our contributions (cf. Fig. 1) and relate them to the literature.

1. Originally, in [19], CMTS are defined as a suitable restriction of MTS, with modalities assigned to the transitions. We provide a polished formal definition of CMTS by assigning modalities to the actions and show that our notion of CMTS corresponds to the one from [19] (Lemma 2).
2. We introduce the notion of refinement ($\preceq_c$) between CMTS. We define equivalence ($\equiv_m$, resp. $\equiv_c$) and strong bisimilarity ($\sim_m$, resp. $\sim_c$) of both MTS and CMTS, and show their relations (Theorem 3). For MTS, strong bisimilarity implies equivalence (Lemma 1), but we demonstrate that these relations do not coincide for MTS (Fig. 3), while we do prove their coincidence for the coherent case (Theorem 2).
3. We show that there are cases of CMTS that are in a refinement relation, but do not follow the informal intuition. We explain this in detail (§4.2):
   (a) We define the notion of persistent transition removal and show that CMTS refinement does not satisfy this property (Figs. 5–7);
   (b) We introduce a notion of syntactic CMTS refinement that we prove equivalent (modulo strong bisimilarity) to CMTS refinement (Theorem 6). Syntactic CMTS refinement follows our intuition of CMTS refinement and extends the notion of product derivation from [19], which is only defined as a relation between a CMTS and a product LTS; we prove that syntactic CMTS refinement enjoys the persistent transition removal property (Theorem 5);
   (c) We provide conditions by which CMTS refinement satisfies the persistent transition removal property (Theorem 7).
4. We introduce the notion of thorough refinement ($\preceq_{ct}$) of CMTS. Contrary to refinement of MTS, which is not complete (Fig. 2), we prove that CMTS refinement is complete, i.e., CMTS thorough refinement coincides with CMTS refinement (Theorem 8). The proof exploits the equivalence between CMTS refinement and syntactic CMTS refinement (§4.2). Notably, it follows that thorough refinement is decidable in polynomial time for the coherent fragment of MTS, while it is known to be decidable in EXPTIME for MTS [6].
5. We extend the correspondence of CMTS refinement and CMTS thorough refinement to the notions of equivalence ($\equiv_c$) and thorough equivalence ($\equiv_{ct}$) of CMTS (Theorem 10).
6. We extend these notions of refinement and equivalence to MTSC (§5):
   (a) We introduce notions of refinement ($\preceq_{cc}$), thorough refinement ($\preceq_{cct}$), and strong thorough refinement ($\preceq_{ccst}$) between MTSC and show their relations (Theorems 11–13), proving MTSC refinement to be complete with respect to MTSC strong thorough refinement.
   (b) We introduce notions of strong bisimilarity ($\sim_{cc}$) and (strong) thorough equivalence ($\equiv_{cc(s)t}$) of CMTS and show that, differently from CMTS, for MTSC only *strong* thorough equivalence coincides with strong bisimilarity (Theorem 14), while strong bisimilarity solely implies MTSC thorough equivalence (Lemma 15).

Outline Section 2 discusses related work. Section 3 contains background on MTS. The core of the paper is formed by Sections 4 (CMTS) and 5 (MTSC), which contain the main contributions. Section 6 presents our conclusion and some future work. An appendix contains all proofs of the results of the paper.

2. Related work

MTS and their variants are widely studied in the literature. In [5], four different refinement relations are studied extensively, including thorough refinement, and an MTS is said to be *consistent* if it admits at least one non-empty implementation. MTS that allow inconsistent specifications, where transitions can be necessary but not permitted, are called Mixed Transition Systems [3,21].

In [6, Corollary 4.6], it is proved that, similarly to modal refinement, thorough refinement is decidable in polynomial time for deterministic MTS, whilst thorough refinement is generally decidable in EXPTIME for non-deterministic MTS. We note that the classes of deterministic MTS and CMTS do not coincide. While thorough refinement does not always imply modal refinement of MTS, in [22, Lemma 3.6] it is proved that thorough refinement implies modal refinement of a deterministic overapproximation of (non-deterministic) MTS.

In [5, Theorem 3], it is proved that any alternative notion $\preceq_{alt}$ of modal refinement that is both sound and complete cannot be decided in polynomial time unless *P=NP*. This is obtained by reducing the problem of deciding thorough refinement to the problem of deciding whether a 3-DNF formula is a tautology. However, in this case, thorough refinement considers all implementations

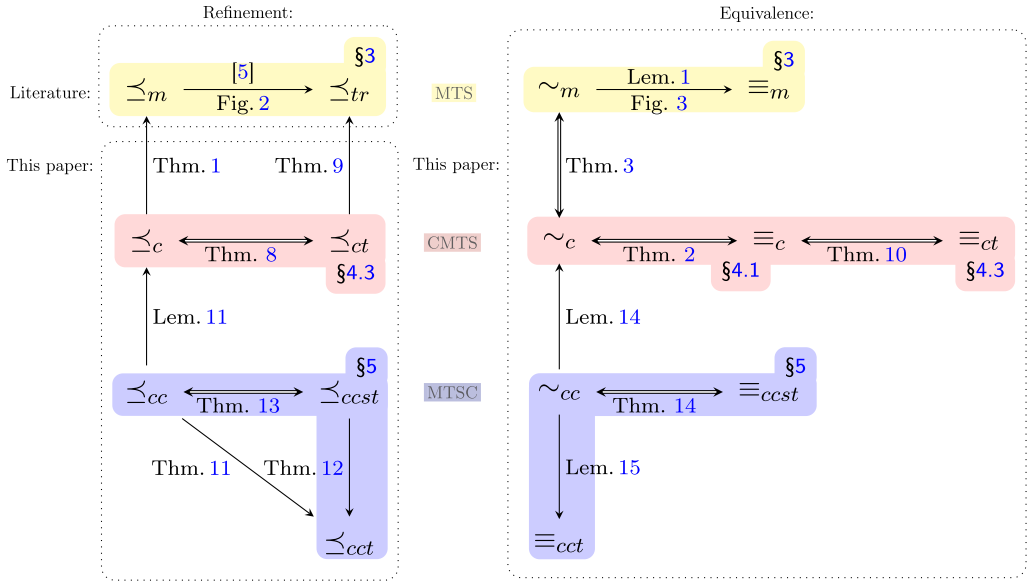


Fig. 1. Overview of the main contributions of this paper: the yellow relations in the top row concern MTS, the red relations in the middle row concern CMTS, the blue relations in the bottom row concern MTSC; a single arrow represents implication, while a double arrow represents coincidence. It is also indicated which results are original and which are taken from the literature (viz., the results on MTS in the upper left box). Note that no specific theorem is central to all the others.

obtained through modal refinement $\preceq_m$, and not only those obtained using the alternative notion $\preceq_{alt}$. The problem of proposing an alternative notion of modal refinement that is both sound and complete with respect to its set of implementations is left open [5]. The main challenge is to argue that the considered set of implementations is also interesting from a practical point of view. In this paper, we prove that CMTS refinement is sound and complete (Theorem 8). Moreover, CMTS refinement has practical applications in the field of SPL, as mentioned in the Introduction and further discussed below.

Parametric MTS (PMTS) [23–25] were introduced to enhance the expressiveness of MTS. PMTS are LTS equipped with an obligation function Φ, which is a parametric Boolean proposition over the outgoing transitions from each state. The satisfying assignments of Φ yield the allowed combinations of outgoing transitions. When Φ is not parametric, PMTS are called Boolean MTS (BMTS). PMTS are capable of expressing, among others, *persistent* choices (i.e., once some outgoing transition is enabled, it must be enabled also everywhere else). It is shown that MTS are a special case of BMTS, and that BMTS are a special case of PMTS. In this paper, we show that also CMTS are a special case of MTS (Lemma 2). Thorough refinement is computable in NEXPTIME for both BMTS and PMTS, while we show in this paper that thorough refinement is polynomial for CMTS (Theorem 8). Modal refinement of MTS, BMTS, and PMTS is not complete, whereas we show in this paper that CMTS refinement is complete (Lemma 10). The deterministic variants of PMTS and BMTS are called, respectively, DPMTS and DBMTS. When restricting to only deterministic systems, similarly to CMTS, also DBMTS modal refinement is complete, whereas DPMTS modal refinement is still not complete.

The application of variants of the MTS formalism to product lines has been published in many top venues over the years (with various case studies, more below). In [7–9], it was noted that, under the assumption that the features of an SPL are identified with the actions of an MTS, modal refinement is not capable of capturing a notion of conformance suitable for SPL. Two issues were signalled. First, an action may not be the action of both a necessary and an optional transition, since a feature is either mandatory or optional. Second, if a transition labelled with an optional action occurs as a necessary transition in an LTS implementation, then all transitions labelled with the same action must occur as necessary transitions in that LTS. In [19], the first and second issue are solved by requiring MTS to be 'coherent' (i.e., CMTS) and the products to be consistently derived, respectively. This notion of 'consistent' product derivation requires that whenever an optional transition is discarded in an implementation, all transitions sharing the same label must also be discarded. This consistency requirement mimicks the aforementioned persistency of PMTS [23–25] and it is not to be confused with the above mentioned notion of consistency as studied in [5]. Hence, just like PMTS, also CMTS can express persistency.

Another issue, amply signalled in the literature for both MTS and CMTS (cf, e.g., [19]), is that so-called cross-tree constraints in feature models that regard *alternative* behaviour cannot be captured (e.g., it is not possible to enforce that the occurrence of one action in an LTS implementation excludes that of another action in the same implementation). In [19], this is solved by equipping CMTS with (variability) constraints, denoted by MTSυ, and called MTSC in this paper. Refinement and equivalence of MTSC are discussed in Section 5.

Table 1 illustrates a comparison of MTS and their dialects based on their ability to express alternative, non-deterministic, and persistent behaviour, as well as whether the refinement is complete or not. This illustrates that among various variants of MTS, only CMTS possess the ability to express non-determinism while maintaining a sound and complete refinement relation. We contend that, while modelling SPL may involve expressing alternative and persistent behaviour, non-determinism and complete refinement are two fundamental properties inherent to any specification language.

Table 1
A comparison of different variants of MTS based on (i) their ability to express alternative and persistent behaviour, which are typical characteristics of SPL, as well as (ii) their ability to express non-deterministic behaviour and to enjoy a notion of refinement that guarantees soundness and completeness with respect to the set of implementations; these last two characteristics are desirable for any specification language yet, as of now, the only variant that satisfies both these features is CMTS.

	persistent behaviour	alternative behaviour	non-deterministic behaviour	complete refinement
MTS	✗	✗	✓	✗
PMTS	✓	✓	✓	✗
DPMTS	✓	✓	✗	✗
BMTS	✓	✓	✓	✗
DBMTS	✓	✓	✗	✓
CMTS	✓	✗	✓	✓
MTSC	✓	✓	✓	✗

During the last decade, several other formalisms with a transition system semantics were developed for SPL modelling and analysis. These include first and foremost the seminal work on Featured Transition Systems (FTS) [26–28], but it is worth mentioning also process-algebraic approaches [29–36] and Feature (Petri) nets [37]. Undoubtedly, MTS and FTS are the most studied formalisms for SPL modelling and analysis. In [38], it was proved that CMTS with variability constraints (i.e., MTSυ, called MTSC in this paper) are equally expressive as FTSs. This complements the expressiveness results in [39–41], namely that MTS are less expressive than FTS and that FTS can be encoded into equivalent sets of multiple MTS.

Remark This paper presents advancements in the theory of CMTS and MTSC by focusing on the challenge of establishing a sound and complete notion of refinement. Additionally, the paper includes insightful examples to clarify the technical aspects involved. For practical illustrations of SPL case studies modelled using CMTS and MTSC, we refer to [8,9,16–20,30,38].

3. Background

We start by discussing some background on MTS. The standard definition of MTS accounts for two sets of transitions, *permitted* (or *may*) transitions, denoted by $\Delta_{\Diamond}$, and *necessary* (or *must*) transitions, denoted by $\Delta_{\Box}$, such that $\Delta_{\Box} \subseteq \Delta_{\Diamond}$, i.e., all (necessary) transitions are permitted. A transition $(q,a,q') \in \Delta_{\Diamond}$ is also denoted as $q \xrightarrow{a}_{\Diamond} q'$ and likewise $q \xrightarrow{a}_{\Box} q'$ if $(q,a,q') \in \Delta_{\Box}$. The reader may be misled to think that $q \xrightarrow{a}_{\Diamond} q'$ excludes $q \xrightarrow{a}_{\Box} q'$, and vice versa that $q \xrightarrow{a}_{\Box} q'$ excludes $q \xrightarrow{a}_{\Diamond} q'$. However, the first statement is not always true and the second is always false, since $\Delta_{\Box} \subseteq \Delta_{\Diamond}$. For our purpose, it is irrelevant to indicate that a transition is permitted. For the sake of simplifying the presentation, we thus opt for a slightly revised definition of MTS, where we partition the set of transitions into *optional* and *necessary* transitions, and no longer indicate the fact that all transitions are *permitted*.

Definition 1 *(MTS). A* Modal Transition System (MTS) *is a 5-tuple* $(Q, A, \overline{q}, \Delta_{\bigcirc}, \Delta_{\Box})$*, with set* Q *of states, set* A *of actions, initial state* $\overline{q} \in Q$*, and transition relation* $\Delta \subseteq Q \times A \times Q$ *partitioned into* optional transitions*, denoted by* $\Delta_{\bigcirc}$*, and* necessary transitions*, denoted by* $\Delta_{\Box}$*, i.e.,* $\Delta_{\bigcirc} \cap \Delta_{\Box} = \varnothing$*. If* $(q,a,q') \in \Delta_{\bigcirc}$*, then we also write* $q \xrightarrow{a}_{\bigcirc} q'$*, and likewise we also write* $q \xrightarrow{a}_{\Box} q'$ *for* $(q,a,q') \in \Delta_{\Box}$*. We write* $q \xrightarrow{a} q'$ *when* $(q,a,q') \in \Delta$*.*

Note that the standard definition of MTS is $(Q, A, \overline{q}, \Delta_{\Diamond}, \Delta_{\Box})$, where $\Delta_{\Diamond} = \Delta_{\bigcirc} \cup \Delta_{\Box}$. An LTS is an MTS where $\Delta_{\bigcirc} = \varnothing$. In the sequel, the conversion from an MTS $(Q, A, \overline{q}, \Delta_{\bigcirc}, \Delta_{\Box})$ with $\Delta_{\bigcirc} = \varnothing$ to an LTS $(Q, A, \overline{q}, \Delta)$ with $\Delta = \Delta_{\Box}$ is implicit. Moreover, we will use subscripts or superscripts to indicate the origin of an element of a tuple, i.e., $S = (Q_S, A_S, \overline{s}, \Delta_{\bigcirc}^S, \Delta_{\Box}^S)$.

We now define modal refinement of MTS.

Definition 2 *(modal refinement). An MTS* S *is a* (modal) refinement *of an MTS* T*, denoted by* $S \preceq_m T$*, if and only if there exists a* refinement relation $\mathcal{R} \subseteq Q_S \times Q_T$ *such that* $(\overline{s}, \overline{t}) \in \mathcal{R}$ *and for all* $(s,t) \in \mathcal{R}$*, the following holds:*

1. *whenever* $t \xrightarrow{a}_{\Box} t'$*, for some* $t' \in Q_T$ *and* $a \in A_T$*, then* $a \in A_S$*,* $\exists\, s' \in Q_S : s \xrightarrow{a}_{\Box} s'$*, and* $(s',t') \in \mathcal{R}$*, and*
2. *whenever* $s \xrightarrow{a} s'$*, for some* $s' \in Q_S$ *and* $a \in A_S$*, then* $a \in A_T$*,* $\exists\, t' \in Q_T : t \xrightarrow{a} t'$*, and* $(s',t') \in \mathcal{R}$*.*

We also say that S *(modally) refines* T *when* $S \preceq_m T$*.*

Intuitively, S modally refines T if any necessary transition of T can be mimicked by a necessary transition of S, and every transition of S can be mimicked by a transition of T.

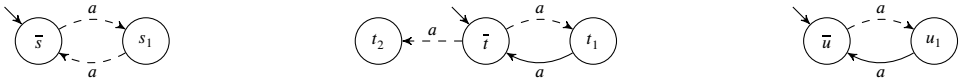


Fig. 2. Three MTS S, T, and U, where $S \preceq_{tr} T$, but $S \not\preceq_m T$, and $S \not\preceq_{tr} U$ and $S \not\preceq_m U$; solid arcs depict necessary transitions, dashed arcs depict optional transitions (reproduced from [6]).

Fig. 3. Two MTS S and T such that $S \equiv_m T$, but $S \nsim_m T$.

The set of implementations of an MTS M is defined as the set of LTS that are a modal refinement of M. Indeed, LTS cannot be further refined and are considered implementations.

Definition 3 *(implementations). Given an MTS M, its set of* implementations*, denoted by $Impl(M)$, is the set $\{ L \text{ is an LTS} \mid L \preceq_m M \}$ of LTS.*

In other words, every LTS refinement of an MTS M is an *implementation* of M.[1]

Two MTS can also be compared through their sets of implementations. This relation, defined next, is called *thorough refinement* (cf., e.g., [4–6]).

Definition 4 *(thorough refinement). An MTS S* thoroughly refines *an MTS T, denoted by $S \preceq_{tr} T$, if $Impl(S) \subseteq Impl(T)$.*

In [5], it is shown that modal refinement implies thorough refinement. In other words, modal refinement is *sound*, i.e., each time an MTS S modally refines an MTS T, it follows that the set of implementations of T contains also the implementations of S. However, the contrary is not true, i.e., modal refinement is not *complete*. Fig. 2, reproduced from [6], shows an example where the set of implementations of T contains also the implementations of S, but S does not modally refine T. In all figures of this paper, dashed arcs are used to depict optional transitions ($\Delta_{\bigcirc}$), while solid arcs depict necessary transitions ($\Delta_{\Box}$).

The relation of modal equivalence formally defined below is taken from [42, Definition 14]. Two MTS are equivalent if both modally refine each other.

Definition 5 *(modal equivalence). An MTS S is* (modal) equivalent *to an MTS T, denoted by $S \equiv_m T$, if and only if $S \preceq_m T$ and $T \preceq_m S$.*

As usual, strong bisimilarity of MTS is defined by additionally requiring the two modal refinement relations to be such that one is the transpose of the other, formally defined next.

Definition 6 *(strong bisimilarity). An MTS S is strongly bisimilar to an MTS T, denoted by $S \sim_m T$, if and only if there exists a refinement relation $\mathcal{R}$ for $S \preceq_m T$ and $\mathcal{R}^T$ for $T \preceq_m S$.*

It is easy to see that strong bisimilarity of MTS implies modal equivalence of MTS.

Lemma 1 *($\sim_m$ implies $\equiv_m$). Let S and T be two MTS. If $S \sim_m T$, then $S \equiv_m T$.*

As mentioned in the Introduction, the proof of this lemma as well as the proofs of all the results that we present in the rest of the paper can be found in the Appendix.

Similarly to LTS, equivalence of MTS is sometimes too coarse, since it may identify as equal two MTS that are not bisimilar, as illustrated in Fig. 3, where $S \preceq_m T$ and $T \preceq_m S$ follow from the two refinement relations $\mathcal{R}_{S \preceq_m T} = \{(\bar{s}, \bar{t}), (s_1, t_1), (s_2, t_1), (s_3, t_2)\}$ and $\mathcal{R}_{T \preceq_m S} = \{(\bar{t}, \bar{s}), (t_1, s_1), (t_2, s_3)\}$, respectively, but $\mathcal{R}_{S \preceq_m T} \neq (R_{T \preceq_m S})^T$.

4. Coherent MTS

We now define Coherent MTS (CMTS), originally introduced in [19], their modal refinement, and equivalence relations. CMTS have been introduced to model the behaviour of SPL by identifying the features of an SPL as the actions of an MTS. In [19], CMTS were (informally) defined by imposing a further constraint on the definition of MTS (cf. Lemma 2 below), whilst here we polish and formalise their definition by assigning the modalities to the actions. It follows that, differently from MTS, in CMTS the transitions

[1] In the sequel, we will implicitly consider an implementation to be the minimal LTS in its class of strong bisimulations, where strong bisimulation is the equivalence relation between two LTS used in this paper.

labelled with the same action are tied together. This means that all transitions labelled with one and the same action are either all necessary actions or all optional actions. Accordingly, if, as the result of a refinement, one optional transition is turned into a necessary transition, then this happens to all other transitions labelled with that same action. This notion was called *persistency* in [23–25]. Similarly, if an optional transition is removed, then this occurs for all other transitions labelled with that same action.[2] Next we formally define CMTS. Compared to MTS, where the modalities are assigned to the transitions, we underline that in CMTS the modalities are assigned to the actions of the transitions. A transition thus inherits the modality of its action.

Definition 7 *(coherent MTS). A* Coherent Modal Transition System (CMTS) *is a 5-tuple* $(Q, A_{\bigcirc}, A_{\Box}, \bar{s}, \Delta)$*, with set* Q *of states, set* $A = A_{\bigcirc} \cup A_{\Box}$ *of actions partitioned into* optional actions*, denoted by* $A_{\bigcirc}$*, and* necessary actions*, denoted by* $A_{\Box}$*, i.e.,* $A_{\bigcirc} \cap A_{\Box} = \emptyset$*, initial state* $\bar{s} \in Q$*, and transition relation* $\Delta \subseteq Q \times A \times Q$*. Moreover, let* $\Delta_{\bigcirc} = \{\, s \xrightarrow{a} s' \in \Delta \mid a \in A_{\bigcirc},\ s, s' \in Q \,\}$ *and let* $\Delta_{\Box} = \{\, s \xrightarrow{a} s' \in \Delta \mid a \in A_{\Box},\ s, s' \in Q \,\}$.

From now on, without loss of generality, we only consider MTS whose elements are non-redundant, i.e., all transitions are reachable and each state and action appear in at least one transition. An LTS is a CMTS where $A_{\bigcirc} = \emptyset$. The next lemma states the conditions (informally stated in [19]) under which an MTS is *coherent*, i.e., if two transitions share the same action, then they must also share the same modality. This condition can be used to check whether it is possible to translate an MTS into a syntactically equivalent CMTS.

Lemma 2 *(coherence). Let* S *be an MTS such that the following holds:*

$$\forall a \in A_S .(\exists s_1, s_2 \in Q_S : s_1 \xrightarrow{a}_{\Box} s_2) \implies (\nexists s_3, s_4 \in Q_S : s_3 \xrightarrow{a}_{\bigcirc} s_4)$$

Then $S' = (Q_S, A_{\bigcirc}, A_{\Box}, \bar{s}, \Delta^S_{\bigcirc} \cup \Delta^S_{\Box})$*, where* $A_{\bigcirc} = \{\, a \mid s \xrightarrow{a} s' \in \Delta^S_{\bigcirc} \,\}$ *and* $A_{\Box} = \{\, a \mid s \xrightarrow{a} s' \in \Delta^S_{\Box} \,\}$ *is a CMTS.*

Note that any MTS defined as a 5-tuple $(Q, A, \bar{s}, \Delta_{\bigcirc}, \Delta_{\Box})$ can be redefined as a 5-tuple $(Q, A_{\bigcirc}, A_{\Box}, \bar{s}, \Delta)$ using the translation provided in Lemma 2. However, whenever the conditions in Lemma 2 are not met (cf., e.g., MTS U in Fig. 4), then $A_{\bigcirc}$ and $A_{\Box}$ do not partition A, a necessary condition for CMTS.

4.1. CMTS refinement

We now introduce the modal refinement of CMTS.

In Section 4.2, we will demonstrate the equivalence (modulo strong bisimilarity) between the forthcoming (coinductive) notion of CMTS refinement (Definition 8) and a simpler operational notion of refinement (cf. Theorem 6). The operational notion of refinement is based on two operations. The first operation involves removing optional transitions with the same label, while the second operation transforms optional transitions with the same label into must transitions (cf. Definition 10).

However, while the operational definition of refinement may be simpler, Definition 8 plays a crucial role in understanding the relationship between CMTS refinement and modal refinement (i.e., Definition 2). Specifically, Cases 1a and 2 of Definition 8 are inherited from modal refinement, while Cases 1b and 1c are unique to this new notion of refinement.

Furthermore, Definition 8 is essential for all the proofs in this paper (cf. the Appendix), as it enables the use of coinductive reasoning.

Intuitively, whenever $\mathcal{R}$ is a relation proving $S \preceq_c T$, for all pairs $(s,t) \in \mathcal{R}$, in addition to the constraints of modal refinement, it is also required that for every optional transition $\delta = t \xrightarrow{a}_{\bigcirc} t'$ of T, *either* δ is mimicked by a transition $s \xrightarrow{a}_{\bigcirc} s'$ in S (and in this case S may contain other transitions labelled with a) and $(s', t') \in \mathcal{R}$, *or* the action a does not occur as label of any transition of S.

It is important to emphasise that this additional constraint plays a critical role in comprehending the distinctions between Definition 2 and Definition 8. Specifically, it sheds light on why CMTS refinement achieves completeness while MTS refinement falls short in this regard (cf. Section 6).

Definition 8 *(CMTS refinement). A CMTS* S *is a* CMTS refinement *of a CMTS* T*, denoted by* $S \preceq_c T$*, if and only if there exists a* refinement relation $\mathcal{R} \subseteq Q_S \times Q_T$ *such that* $(\bar{s}, \bar{t}) \in \mathcal{R}$ *and for all* $(s,t) \in \mathcal{R}$*, the following holds:*

1. *whenever* $t \xrightarrow{a} t'$*, for some* $t' \in Q_T$*, then either*
   (a) $a \in A^T_{\Box}$, $a \in A^S_{\Box}$, $\exists\, s' \in Q_S : s \xrightarrow{a}_{\Box} s'$, *and* $(s', t') \in \mathcal{R}$*, or*
   (b) $a \in A^T_{\bigcirc}$, $a \in A_S$, $\exists\, s' \in Q_S : s \xrightarrow{a} s'$, *and* $(s', t') \in \mathcal{R}$*, or*
   (c) $a \in A^T_{\bigcirc}$ *and* $a \notin A_S$*;*

[2] We use the intuition that in a refinement $S \preceq_m T$, S is obtained by modifying T. In general, however, S and T are two systems created independently from each other. Rather than "removing" optional transitions, we should say that for a pair of states (s,t) in the refinement relation, for some transition $t \xrightarrow{a}_{\bigcirc} t'$ in T, there is no corresponding transition $s \xrightarrow{a}_{\bigcirc} s'$ in S; likewise for "turning" optional transitions into necessary transitions.

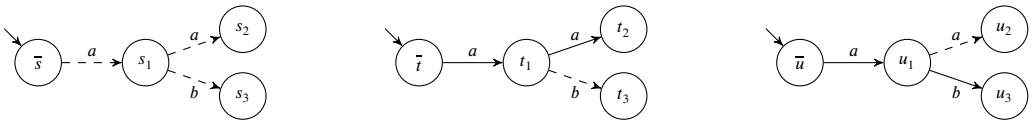


Fig. 4. Two CMTS S and T and an MTS U such that $T \preceq_c S$ and $U \not\preceq_c S$ (U is not a CMTS).

2. *whenever* $s \xrightarrow{a} s'$, *for some* $s' \in Q_S$ *and* $a \in A_S$, *then* $a \in A_T$, $\exists\, t' \in Q_T : t \xrightarrow{a} t'$, *and* $(s', t') \in \mathcal{R}$.

Note that in Case 1a we know that $t \xrightarrow{a}_{\Box} t'$ since $a \in A^T_{\Box}$. Fig. 4 depicts examples of CMTS refinement. In the above definition, the fact that both S and T are CMTS ensures that all transitions with the same action in S are either all optional or all necessary transitions; the same for T. Hence, if an optional transition δ of T is turned into a necessary transition in S, then all transitions in S labelled with the same action as δ are necessary transitions. Cases 1b and 1c of Definition 8 are used to ensure that whenever an optional transition of T is removed in S, all other transitions of S labelled with the same action must also be removed (with some exceptions, cf. Section 4.2). We show that CMTS refinement is a conservative extension of modal refinement.

Theorem 1 ($\preceq_c$ *implies* $\preceq_m$). *Let* S *and* T *be two CMTS. If* $S \preceq_c T$, *then* $S \preceq_m T$.

The algorithm for computing a CMTS refinement is similar to that for computing a modal refinement. One starts with a set R containing all pairs of states of S and T. The algorithm repeats a loop until no more pairs are removed from R. At each iteration of the loop, the pairs not satisfying the conditions of Definition 8 are removed from R. With respect to modal refinement, one needs to check two additional conditions (Cases 1b and 1c). These conditions are local to the pairs of states under scrutiny. The computed set R is the largest CMTS refinement relation between S and T. Let $n = |Q_S \times Q_T|$. Moreover, let $FS(s)$ be defined as the set of outgoing transitions from a state s, i.e., $FS(s) = \{\, (s, a, s') \in \Delta_S \mid a \in A_S \text{ and } s' \in Q_S \,\}$. Let s_m and t_m be the states with the maximum number of outgoing transitions, i.e., for all $s \in Q_S$ and $t \in Q_T$, $|FS(s_m)| \geq |FS(s)|$ and $|FS(t_m)| \geq |FS(t)|$. Let $k = |FS(s_m)| \times |FS(t_m)|$. In the worst case, at each iteration, all elements of R are visited and only one pair is removed. In this worst case, there are $\frac{n(n+1)}{2}$ iterations. At each iteration, in the worst case, one needs to perform $k \times 2$ comparisons among transitions of S and T, and a search through the set R, which in the worst case requires visiting all n pairs. The resulting upper bound for the complexity is $O((n^3 + n^2) \times k) = O(n^3)$. In Section 4.2, we will show that the complexity of computing CMTS refinement can be reduced to the complexity of computing strong bisimulation.

The following lemma relates the actions of two CMTS in a CMTS refinement relation. Whenever $S \preceq_c T$, the set of optional actions of T contains the optional actions of S, and the set of necessary actions of S contains the necessary actions of T. Furthermore, the set of optional actions of T that are not optional actions of S contains the set of necessary actions of S that are not necessary actions of T.

Lemma 3. *Let* S *and* T *be two CMTS such that* $S \preceq_c T$. *It holds that* $A^S_{\bigcirc} \subseteq A^T_{\bigcirc}$, $A^T_{\Box} \subseteq A^S_{\Box}$, *and* $A^S_{\Box} \setminus A^T_{\Box} \subseteq A^T_{\bigcirc} \setminus A^S_{\bigcirc}$.

Similarly to modal refinement, also CMTS refinement is a preorder.

Lemma 4. *CMTS refinement is a preorder.*

Equivalence and strong bisimilarity of CMTS are defined similarly to the case of MTS. In particular, two CMTS are equivalent if they are mutually in a CMTS refinement relation with each other. Moreover, if these relations are such that one is the transpose of the other, then the two CMTS are also strongly bisimilar.

Definition 9 *(CMTS equivalence and CMTS strong bisimilarity). Let* S *and* T *be two CMTS. Then* S *is* CMTS equivalent *to* T, *denoted by* $S \equiv_c T$, *if and only if* $S \preceq_c T$ *and* $T \preceq_c S$. *Moreover,* S *is* CMTS strongly bisimilar *to* T, *denoted by* $S \sim_c T$, *if and only if there exists a CMTS refinement relation* $\mathcal{R}$ *that proves* $S \preceq_c T$ *and* $\mathcal{R}^T$ *proves* $T \preceq_c S$.

In a modal refinement $S \preceq_m T$, whenever all transitions of S and T are necessary, then the relation becomes a strong bisimilarity between S and T. This is also true for CMTS refinement. Instead, in modal refinement, whenever all transitions of S and T are optional, then the relation becomes a simulation (T simulates S). This is different from CMTS refinement. Indeed, in this case, if also S and T have the same sets of (all optional) actions, then the relation becomes again a bisimulation between S and T.

Fig. 5 shows two CMTS S and T such that $S \sim_c T$, where $\mathcal{R}_{S \preceq_c T} = \{(\bar{s}, \bar{t}), (s_1, t_1), (s_1, t_2)\}$ and $\mathcal{R}_{T \preceq_c S} = \{(\bar{t}, \bar{s}), (t_1, s_1), (t_2, s_1)\}$. Fig. 6 shows two CMTS S and U such that $S \sim_c U$ where $\mathcal{R}_{S \preceq_c U} = \{(\bar{s}, \bar{u}), (s_2, u_3), (s_2, u_2), (s_3, u_1), (s_1, u_1)\}$ and $\mathcal{R}_{U \preceq_c S} = (\mathcal{R}_{S \preceq_c U})^T$. Contrary to MTS, equivalence and strong bisimilarity coincide for CMTS.

Theorem 2 ($\equiv_c$ *and* $\sim_c$ *coincide). Let* S *and* T *be two CMTS such that* $S \equiv_c T$, *where* $\mathcal{R}$ *proves* $S \preceq_c T$. *Then* $\mathcal{R}^T$ *proves* $T \preceq_c S$.

Fig. 5. Two CMTS S and T such that $S \sim_c T$, where some optional transition δ of T has no 'syntactic' correspondent in S, but S does have some transition with the same action as δ.

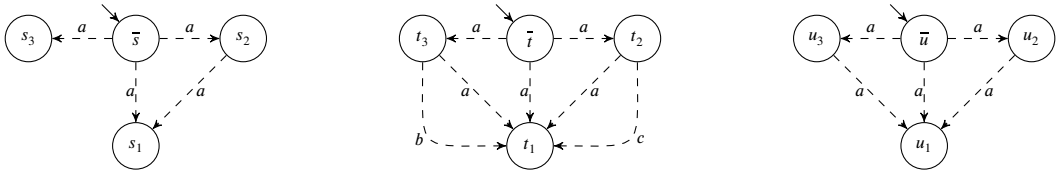


Fig. 6. Three CMTS S, T, and U such that $S \preceq_c T$, $S \npreceq_c T$, $U \preceq_c T$, and $S \sim_c U$; S is obtained from T by removing all transitions labelled with b and c but only one of the five transitions labelled with a, whose source state is reachable in S, and S is not strongly bisimilar to T; on the converse, $U \preceq_c T$ is a syntactic CMTS refinement.

The next result shows that CMTS and MTS strong bisimilarity coincide.

Theorem 3 *($\sim_c$ and $\sim_m$ coincide). Let S and T be two CMTS, then $S \sim_m T$ if and only if $S \sim_c T$.*

In the following, we denote $s \sim_m t$ whenever two states $s \in Q_S$ and $t \in Q_T$ are strongly bisimilar, i.e., $S \sim_m T$ iff $\overline{s} \sim_m \overline{t}$ (recall that $\overline{s}$ and $\overline{t}$ are the initial states of S and T, respectively). We now discuss the minimisation of CMTS, which will be used in Section 4.2. We consider a CMTS to be minimal if it has no strongly bisimilar states. The next result shows how, given a CMTS S with two bisimilar states s_1 and s_2, it is possible to construct a strongly bisimilar CMTS S'' such that one of the two bisimilar states, say s_2, has been removed. The CMTS S'' is obtained from S by removing the state s_2, removing all the outgoing transitions of s_2, and changing the target of all transitions incoming to s_2 into s_1. Finally, S'' is computed from the obtained intermediate CMTS S' by removing the unreachable actions, transitions, and states from S' (an action a is considered unreachable (i.e., redundant) if there is no reachable transition labelled with a). Note that while we assume CMTS to have no redundant elements, we explicitly clarify the process of removing any redundant elements.

Theorem 4 *(remove bisimilar states). Let S be a CMTS, let $\mathcal{R}_b$ be a relation showing that $s_1 \sim_c s_2$ for two states $s_1, s_2 \in Q_S$ such that $s_2 \neq \overline{s}$, and let $BS(s_2) = \{ (s,a,s_2) \mid s \in Q_S,\ a \in A_S \}$. Now let $S' = (Q_S \setminus \{s_2\}, A^S_{\bigcirc}, A^S_{\square}, \overline{s}, \Delta_S \setminus BS(s_2) \cup \{ (s,a,s_1) \mid (s,a,s_2) \in \Delta_S \})$ and let S'' be obtained from S' by removing the unreachable actions, states, and transitions of S', if any. Then it follows that $S \sim_c S''$.*

By applying the above result iteratively, it is possible to transform a CMTS with bisimilar states into a strongly bisimilar one without bisimilar states, i.e., a minimised one. In Fig. 5, we have that $t_1 \sim_c t_2$. Indeed, T is not minimal, and S can be obtained from T by applying Theorem 4.

The next corollary shows that CMTS strong bisimilarity preserves the CMTS refinement relation with other CMTS.

Corollary 1. *Let S and T be two CMTS such that $S \preceq_c T$, and let S'' be a CMTS such that $S \sim_m S''$. Then $S'' \preceq_c T$.*

Combining the previous results, it follows that it is possible to minimise a CMTS S whilst preserving its CMTS refinement relations with other CMTS.

4.2. Syntactic CMTS refinement

We note that the intuitive explanation of CMTS refinement given so far is useful to provide a rough idea, but it is not precise. The intuitive explanation we used is the property stating that whenever an optional transition is removed, then this occurs for all other transitions labelled with that same action. We call this property *persistent transition removal*. Indeed, this property does not hold in all CMTS refinements, i.e., it is sometimes possible to remove an optional transition δ without forcing the removal of all other transitions labelled with the same action. This is the case, e.g., when the removal of δ produces a new CMTS that is strongly bisimilar to the original one. In this case, removing δ does not force to remove all optional transitions labelled with the same action (cf. Fig. 5). It is also possible to remove an optional transition without forcing the removal of all other transitions labelled with the same action in case S and T are not strongly bisimilar, such as, e.g., in Fig. 6. Finally, Fig. 7 shows yet another example where both S and T are minimised.

These refinements do not retract from our intuition of CMTS refinement. Indeed, they are all strongly bisimilar to refinements satisfying the persistent transition removal property. To prove this, we need some additional results, which are described in this section. First, if we remove an optional action from a CMTS, then we obtain a CMTS refinement.

Lemma 5 *(remove action). Let S be a CMTS and, given some $b \in A^S_{\bigcirc}$, let $S' = (Q'_S, A^S_{\bigcirc} \setminus \{b\}, A^S_{\square}, \overline{s}, \Delta'_S)$, where $Q'_S = \{ s \in Q_S \mid s \text{ is reachable} \}$ and $\Delta'_S = \{ (s,a,s') \in \Delta_S \mid a \in A_S \setminus \{b\},\ s \text{ is reachable} \}$. Then it holds that $S' \preceq_c S$.*

We also obtain a CMTS refinement if we turn an optional action into a necessary action.

Lemma 6 *(shift action right). Let S be a CMTS and, given some $b \in A^S_{\bigcirc}$, let $S' = (Q_S, A^S_{\bigcirc} \setminus \{b\}, A^S_{\square} \cup \{b\}, \overline{s}, \Delta_S)$. Then it holds that $S' \preceq_c S$.*

Finally, by Lemma 3, we know that, given $S' \preceq_c S$, it holds that $A^S_{\square} \subseteq A^{S'}_{\square}$. Hence, if we remove a necessary action or turn it into an optional action, then the obtained system is not a CMTS refinement of the original one. When a CMTS S is obtained from a CMTS T by repeatedly using the above operations (either removing an optional action or turning it into a necessary action), then we say that S is a *syntactic CMTS refinement* of T.

Definition 10 *(syntactic CMTS refinement). Let S and T be CMTS such that S is obtained from T by iteratively applying one of the operations 'remove action' and 'shift action right' from Lemmata 5 and 6, respectively. Then S is a* syntactic CMTS refinement *of T.*

Note that we use the term 'syntactic' because, e.g., in a syntactic refinement, loops are never unfolded. Definition 10 extends the notion of *product derivation* from [19, Definition 10], which is only defined for implementations (cf. Definition 12 below). Fig. 6 shows an example of syntactic CMTS refinement $U \preceq_c T$, where U is obtained from T by applying twice the operations from Lemma 5 on the actions b and c. Fig. 7 shows another example of syntactic CMTS refinement $U \preceq_c T$.

Syntactic CMTS refinement satisfies the property of persistent transition removal. Indeed, the removal of an optional transition causes the removal of all transitions labelled with the same action. We now formalise the property of *persistency*, which entails the persistent transition removal. Whenever S is a persistent (CMTS) refinement of T, for all actions $a \in A_S$ the cardinality of the set of transitions labelled with a in S and in T is equal. In other words, if in a CMTS refinement $S \preceq_c T$ one optional transition of T has been removed from S without causing the removal of all other transitions labelled with the same action, then the refinement is non-persistent. This is also the case if S contains a greater number of transitions than T with the same label.

Definition 11 *(persistent refinement). Let S and T be two CMTS such that $S \preceq_c T$. We say that S is a* persistent *refinement of T, whenever for all actions $a \in A_S$, it holds that $|\{ (s, a, s') \in \Delta_S \}| = |\{ (t, a, t') \in \Delta_T \}|$. Otherwise, we say that $S \preceq_c T$ is* non-persistent.

Note that we assume that there are no redundant actions in S. Hence, whenever $a \in A_S$, the cardinality of the set of transitions labelled with a in S is greater than zero. Definition 11 entails the persistent transition removal (related to Lemma 5) and generalises the notion of persistency given in [23–25] (cf. Section 2), which is only related to Lemma 6 (i.e., if an optional transition is turned into a necessary transition, then this happens to all other transitions sharing the same label).

Definition 11 may seem unconventional for Definition 8 (i.e., CMTS refinement) but it is not surprising in the context of Definition 10 (i.e., syntactic CMTS refinement). In this section, we will show the equivalence (modulo strong bisimilarity) of Definition 10 and Definition 8.

First, we show that syntactic CMTS refinement is persistent.

Theorem 5 *(syntactic CMTS refinement is persistent). Let S and T be two CMTS such that S is a syntactic CMTS refinement of T. Then $S \preceq_c T$ is a persistent refinement.*

In Figs. 6 and 7, $U \preceq_c T$ is a persistent refinement. For example, consider the set of optional actions of the CMTS U from Fig. 7, namely, $\{a\}$ (cf. Definition 11). Both U and T have two transitions labelled with a (note that b is not an action in U).

The following theorem states that for any CMTS refinement $S \preceq_c T$, there exists a syntactic CMTS refinement T' of T such that $S \sim_c T'$ and, by Corollary 1, $T' \preceq_c T$. In other words, any S that is a CMTS refinement of T can be computed starting from T (modulo bisimilarity) by applying the operations from Lemmata 5 and 6. Intuitively, the CMTS T' is computed as follows. Firstly, select the necessary actions in S that are not necessary actions in T. By Lemma 3, we know that the selected actions are optional in T. Therefore, we turn them into necessary actions in T' (using the operation from Lemma 6), i.e., $A^{T'}_{\square} = A^S_{\square}$. In the second step, we remove from T' the remaining optional actions of T that are neither optional nor necessary actions in S (using the operation from Lemma 5), i.e., $A^{T'}_{\bigcirc} = A^S_{\bigcirc}$.

Theorem 6 *(CMTS refinement and syntactic CMTS refinement). Let S and T be two CMTS such that $S \preceq_c T$ and let $T' = (Q_{T'}, A^S_{\bigcirc}, A^S_{\square}, \overline{t}, \Delta_{T'})$, where $Q_{T'} = \{ s \in Q_T \mid s \text{ is reachable} \}$ and $\Delta_{T'} = \{ (t, a, t') \in \Delta_T \mid a \in A^S_{\bigcirc} \cup A^S_{\square} \}$. Then it holds that $S \sim_c T'$.*

Note that Theorem 6 does not hold for MTS refinement. Indeed, given a modal refinement $S \preceq_m T$, it is not always possible to compute an MTS that is strongly bisimilar to S by removing transitions or turning optional transitions into necessary transitions. For instance, Fig. 3 shows a refinement $S \preceq_m T$, where S is obtained from T by *adding* a transition. Figs. 6 and 7 show two CMTS refinements $S \preceq_c T$ and two syntactic CMTS refinements $U \preceq_c T$, where $S \sim_c U$, i.e., U is obtained from S and T by applying Theorem 6 (U in Figs. 6 and 7 takes the role of T' in Theorem 6).

Theorem 6 suggests an alternative procedure for computing CMTS refinement. Indeed, checking $S \preceq_c T$ can be done as follows. Firstly compute T' (from Theorem 6). If this is not possible (e.g., $A^T_{\square} \not\subseteq A^S_{\square}$), then (by Lemma 3) $S \not\preceq_m T$. Otherwise, check whether

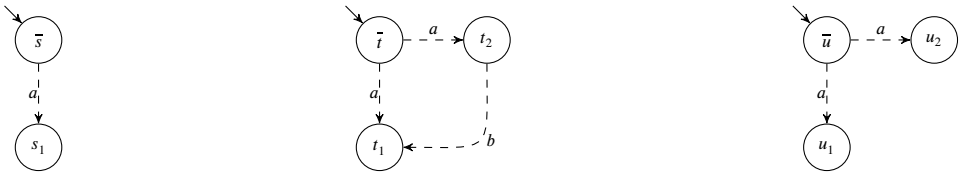


Fig. 7. Three CMTS S, T, and U with a non-persistent refinement $S \preceq_c T$ (note that both CMTS are minimised) and a syntactic CMTS refinement $U \preceq_c T$, where $S \sim_c U$.

$S \sim_c T'$ to determine whether $S \preceq_c T$. This demonstrates that the complexity of computing CMTS refinement is equivalent to the complexity of computing strong bisimulation.

As a corollary, whenever two CMTS share the same sets of actions, then the CMTS refinement relation implies strong bisimilarity.

Corollary 2. *Let S and T be two CMTS such that $S \preceq_c T$, $A^S_\Box = A^T_\Box$, and $A^S_\bigcirc = A^T_\bigcirc$. Then $S \sim_c T$.*

The above results also provide us with an upper bound on the maximum number of 'canonical' CMTS refinements, i.e., canonical elements of the equivalence class induced by strong bisimilarity. Intuitively, when computing a refinement, for each optional action of S, there are three options: either keep it optional, remove it, or turn it into a necessary action.

Corollary 3. *Let $R_S = \{\, S' \mid S' \preceq_c S \text{ and } \forall S'' \in R_S \,.\, S'' \nsim_c S' \,\}$ be a set of CMTS refinements of S that are not strongly bisimilar between each other. Then, for any CMTS S, it holds that $|R_S| \leq 3^{|A^S_\bigcirc|}$.*

We have shown that syntactic CMTS refinement is persistent. We now study the conditions under which CMTS refinement is persistent. Inspecting Fig. 7, we note that there exists a CMTS T' such that $S \preceq_c T'$, $T' \preceq_c T$, and T' is not minimal. T' is obtained from T by only removing those optional actions of T that have also been removed in S. Indeed, the example in Fig. 7 is a non-persistent refinement because T' is not minimal. Another case could be when S is not minimal. We formalise this intuition in the next lemma.

Lemma 7. *Let S and T be two CMTS such that $S \preceq_c T$ is a non-persistent CMTS refinement. Let $T' = (Q_{T'}, A^T_\bigcirc \cap A_S, A^T_\Box, \bar{t}, \Delta_{T'})$, where $Q_{T'} = \{\, s \in Q_T \mid s \text{ is reachable} \,\}$ and $\Delta_{T'} = \{\, (t,a,t') \in \Delta_T \mid a \in A_{T'} \text{ and } t \text{ is reachable} \,\}$. Then either (i) there exist $t_1, t_2 \in Q_{T'}$ such that $t_1 \neq t_2$ and $t_1 \sim_c t_2$, or (ii) there exist $s_1, s_2 \in Q_S$ such that $s_1 \neq s_2$ and $s_1 \sim_c s_2$.*

The next theorem states the conditions under which a CMTS refinement is persistent. Intuitively, if both S and T' (from Lemma 7) are minimised, then $S \preceq_c T$ cannot be non-persistent or otherwise this would be a contradiction to Lemma 7.

Theorem 7 *(persistent CMTS refinement). Let S and T be two CMTS such that $S \preceq_c T$ and T' is the CMTS from Lemma 7. If both S and T' are minimised, then $S \preceq_c T$ is a persistent CMTS refinement.*

Hence, we have shown that non-persistent refinements are ruled out when only using syntactic CMTS refinement or when the refined CMTS is minimal and the CMTS to be refined, polished from the optional actions that we want to remove, is minimal.

Finally, the following lemma states the conditions under which it is possible to turn a necessary action into an optional action whilst maintaining CMTS refinement with another CMTS. This will be useful in the next section when comparing CMTS refinement with CMTS thorough refinement. The main insight is that the necessary action of S that is turned into an optional action, is an optional action also in T, where $S \preceq_c T$.

Lemma 8 *(shift action left). Let S and T be two CMTS with $a \in A^S_\Box \cap A^T_\bigcirc$ and such that $S \preceq_c T$. Then $S' \preceq_c T$, where $S' = (Q_S, A^S_\bigcirc \cup \{a\}, A^S_\Box \setminus \{a\}, \bar{s}, \Delta_S)$.*

4.3. CMTS thorough refinement

The notion of implementation can be defined for CMTS refinement analogously to the way this was done for MTS. The set of implementations of a CMTS S is the set of LTS that are CMTS refinements of S.

Definition 12 *(CMTS implementation). Given a CMTS S, the set $\mathit{Impl}_C(S)$ of* CMTS implementations *of S is the set $\{\, L \text{ is an LTS} \mid L \preceq_c S \,\}$ of LTS.*

Definition 12 is a readaptation of [19, Definition 10]. In [19], CMTS implementations are called 'consistent' products according to the SPL point of view. In the sequel, we define thorough refinement of CMTS analogously to the notion of thorough refinement

Fig. 8. Two CMTS S and T, where $S \preceq_{tr} T$, $S \not\preceq_{m} T$, $S \not\preceq_{ct} T$, and $S \not\preceq_{c} T$.

of MTS. We say that a CMTS S is a thorough refinement of a CMTS T, if the set of CMTS implementations of T contains the set of CMTS implementations of S.

Definition 13 *(CMTS thorough refinement). Let S and T be CMTS. Then S is a* CMTS thorough refinement *of T, denoted by $S \preceq_{ct} T$, if $Impl_C(S) \subseteq Impl_C(T)$.*

Similarly to modal refinement, also CMTS refinement implies CMTS thorough refinement. This result is obtained by exploiting Lemma 4, i.e., the transitivity of CMTS refinement.

Lemma 9 *($\preceq_c$ implies $\preceq_{ct}$). Given two CMTS S and T, it holds that $S \preceq_c T$ implies $S \preceq_{ct} T$.*

Contrary to the non-coherent case, when we restrict ourselves to consider only CMTS, we obtain that the converse of Lemma 9 also holds. This result exploits Theorem 6, i.e., the correspondence between CMTS refinement and syntactic CMTS refinement. Intuitively, while in a non-deterministic choice of an MTS it is possible to have an implementation excluding a 'bad' branch (i.e., violating refinement) in favour of a 'good' branch (cf. Fig. 2), this is not possible for CMTS. The reason is that a (syntactic) CMTS refinement either removes or keeps all transitions that are labelled with the same action.

The following lemma states that CMTS thorough refinement implies CMTS refinement. Given a thorough refinement $S \preceq_{ct} T$, the proof uses an implementation I of S and, by the hypothesis, of T, which is obtained by turning all optional actions of S into necessary actions. Subsequently, starting from that implementation I, the operation of Lemma 8 is applied iteratively to show that $S \preceq_c T$.

Lemma 10 *($\preceq_{ct}$ implies $\preceq_c$). Given two CMTS S and T, it holds that $S \preceq_{ct} T$ implies $S \preceq_c T$.*

The main result of this section, the coincidence between CMTS refinement and CMTS thorough refinement, is stated next. It follows trivially from Lemmata 9 and 10.

Theorem 8 *($\preceq_{ct}$ and $\preceq_c$ coincide). CMTS refinement and CMTS thorough refinement coincide.*

Fig. 2 shows an example of two MTS such that $S \preceq_{tr} T$, but $S \not\preceq_{m} T$. It also trivially holds that $S \not\preceq_{c} T$ and $S \not\preceq_{ct} T$, because T is not a CMTS. Fig. 8 displays two CMTS S and T that are similar to the MTS displayed in Fig. 2 (but with some actions b). Indeed, for the two CMTS it still holds that $S \preceq_{tr} T$ and $S \not\preceq_{m} T$. However, contrary to the MTS thorough refinement that holds for the MTS in Fig. 2, CMTS thorough refinement does not hold for the CMTS in Fig. 8, i.e., $S \not\preceq_{ct} T$. Indeed, T has only two implementations obtainable through syntactic CMTS refinement. The first is obtained by removing action a (Lemma 5), and it contains only the state $\bar{t}$ and no transition. The second implementation of T is obtained by turning action a into a necessary action (Lemma 6), and it contains all states and transitions of T. On the converse, S also has an implementation that is obtained by removing action b and turning action a into a necessary action. This implementation is not strongly bisimilar to any of the implementations of T, i.e., $S \not\preceq_{ct} T$.

This coincidence of $\preceq_c$ and $\preceq_{ct}$ is significant. In [6], it has been shown that verifying thorough refinement of MTS can be accomplished in EXPTIME. Conversely, by restricting to CMTS, thorough refinement can be computed using the same algorithm for computing CMTS refinement, thus resulting in polynomial complexity, as discussed earlier.

Via Lemma 10, we also prove that $\preceq_{ct}$ is a conservative extension of $\preceq_{tr}$.

Theorem 9 *($\preceq_{ct}$ implies $\preceq_{tr}$). Let S and T be two CMTS. If $S \preceq_{ct} T$, then $S \preceq_{tr} T$.*

We conclude this section by examining the concept of equivalence that is generated by thorough refinement. It is important to note that thorough strong bisimilarity ($\sim_{ct}$) is not considered, because thorough refinement is founded on set inclusion. Two CMTS are deemed thoroughly equivalent if they share identical sets of CMTS implementations.

Definition 14 *(CMTS thorough equivalence). Two CMTS S and T are thorough equivalent, denoted by $S \equiv_{ct} T$, if and only if $Impl_C(S) = Impl_C(T)$*

Analogous to CMTS refinement, the notions of CMTS equivalence and CMTS thorough equivalence also coincide, as demonstrated next.

Theorem 10 *($\equiv_c$ and $\equiv_{ct}$ coincide). Let S and T be two CMTS. Then $S \equiv_c T$ if and only if $S \equiv_{ct} T$.*

Fig. 5 shows two CMTS S and T such that $S \equiv_c T$ and $S \equiv_{ct} T$, and Figs. 6 and 7 show two CMTS S and U such that $S \equiv_c U$ and $S \equiv_{ct} U$.

5. CMTS with constraints

In this section, we extend our investigation of CMTS refinement to CMTS equipped with constraints. We consider variability constraints as defined in [19], on the basis of the earlier proposals of [16,17], to cope with the inability of CMTS to express alternative behaviour in the context of SPL (cf. Section 2).

We associate with an MTS a set of *variability constraints* that must be taken into account when deriving the set of implementations. These are essentially conditions on the presence of actions in the LTS.

Definition 15 *(variability constraints). Let $L = (Q, \Sigma, \overline{q}, \Delta)$ be an LTS. Then a* variability constraint *is a propositional formula φ over a given set of atoms. Moreover, a set of variability constraints Φ is rendered as the propositional formula $\varphi' = \bigwedge_{\varphi \in \Phi} \varphi$. The LTS L satisfies φ (denoted by $L \vDash \varphi$) if and only if $I \vDash \varphi$, where I is the interpretation of φ defined as: for all atoms a in φ, $a = true$ if $a \in \Sigma$ and $a = false$ otherwise.*

Recall that we only consider LTS whose alphabet Σ only contains actions that are reachable in the corresponding LTS. In [19], constraints were actually defined using a syntax borrowed from variability modelling, where in particular three specific types of constraints are used (*requires* and *excludes*, which are so-called "cross-tree constraints" in feature models [11–13], and *alternative*, which is the n-ary logical exclusive or (xor) operation). The requires constraint, denoted by $a \,\mathrm{REQ}\, b$, says that if an optional action a is asserted in a refinement, then also the optional action b is asserted. It is expressed by the propositional formula $a \implies b$. The excludes constraint, denoted by $a \,\mathrm{EXC}\, b$, says that if an optional action a is asserted in a refinement, then the optional action b is removed, and vice versa. It is expressed by the propositional formula $a \implies \neg b$. The alternative constraint, denoted by $a_1 \,\mathrm{ALT}\, a_2 \cdots \mathrm{ALT}\, a_n$, says that precisely one of the optional actions $a_1, a_2, \ldots, a_n$, for some $n > 1$, is asserted in a refinement. It is expressed by the propositional formula $a_1 \oplus a_2 \oplus \cdots \oplus a_n$, where $\oplus$ denotes *exclusive disjunction*.

In the following, we will use the notation $a \,\mathrm{REQ}\, b$ for some example constraints.

Definition 16 *(CMTS with constraints). A CMTS with constraints (MTSC)*[3] *is a pair (S, Φ), where S is a CMTS and Φ is a set of (variability) constraints according to Definition 15.*

The constraints of an MTSC are only evaluated on implementations. The implementations of an MTSC are the LTS obtained through MTSC refinement that satisfy the given constraints. Fig. 9 shows some examples of MTSC.

Definition 17 *(MTSC implementation). Given an MTSC (S, Φ), the set $Impl_C(S, \Phi)$ of* implementations *of (S, Φ) is the set $\{ L \mid (L, \Phi') \preceq_{cc} (S, \Phi) \wedge L \vDash \Phi' \}$ of LTS.*

Fig. 10 shows examples of implementations of MTSC. Defining constraints on CMTS is useful to compare two MTSC also when their constraints are differing. Two MTSC are in (MTSC) refinement relation whenever their corresponding CMTS are in (CMTS) refinement relation and the constraints of the refined MTSC are entailing the constraints of the MTSC to be refined.

Definition 18 *(MTSC refinement). An MTSC (S, Φ) is a* refinement *of an MTSC (T, Φ'), denoted by $(S, \Phi) \preceq_{cc} (T, \Phi')$, if $S \preceq_c T$ and $\Phi \implies \Phi'$.*

Recall that we also consider as LTS those MTS with an empty set of optional actions. MTSC implementations are called *valid products* in [19, Definition 12]. Fig. 9 shows examples of MTSC refinement. The fact that MTSC refinement implies CMTS refinement is straightforward.

Lemma 11 *($\preceq_{cc}$ implies $\preceq_c$). Let (S, Φ) and (T, Φ') be two MTSC. Then $(S, \Phi) \preceq_{cc} (T, \Phi')$ implies $S \preceq_c T$.*

The next lemma shows the relation between an LTS entailing a set of constraints ϕ and another set of constraints ϕ' that are entailed by ϕ. This result is auxiliary to the subsequent lemma below.

Lemma 12. *Let L be an LTS and ϕ and ϕ' be two sets of constraints. It holds that $(L \vDash \phi \wedge \phi \implies \phi')$ implies $(L \vDash \phi')$.*

As anticipated, Lemma 12 allows us to provide an equivalent definition of the set of implementations of an MTSC. Whilst in Definition 17, the implementations of an MTSC are satisfying constraints that entail those of the original MTSC (i.e., Definition 17 uses $\preceq_{cc}$), in the next definition the set of constraints is fixed in both the MTSC and its implementations (i.e., Lemma 13 uses $\preceq_c$).

[3] In [19], the notation MTSυ is used rather than MTSC.

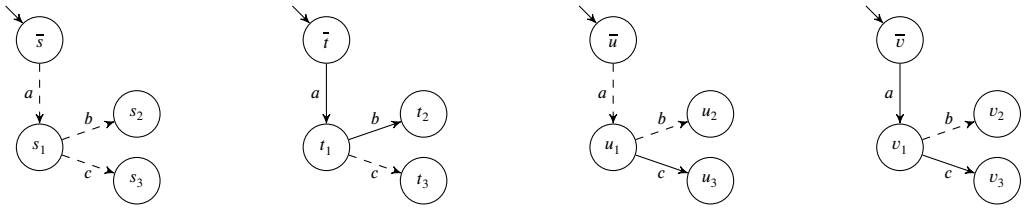


Fig. 9. Some examples of MTSC refinement: (i) $(T,\emptyset) \not\preceq_{cc} (S,\{a \,\mathrm{REQ}\, b\})$, i.e., we have that $T \preceq_c S$, but $\mathit{true} \not\Longrightarrow (a \implies b)$; (ii) $(U,\{a \,\mathrm{REQ}\, b\}) \preceq_{cc} (S,\{a \,\mathrm{REQ}\, b\})$ and (iii) $(V,\emptyset) \not\preceq_{cc} (S,\{a \,\mathrm{REQ}\, b\})$; moreover, $T \preceq_{ct} S$, but $(T,\emptyset) \not\preceq_{cct} (S,\{a \,\mathrm{REQ}\, b\})$ (i.e., the implementation of T obtained by removing the transition labelled with c is not an implementation of $(S,\{a \,\mathrm{REQ}\, b\})$ due to the constraint $a \,\mathrm{REQ}\, b$).

Fig. 10. Two CMTS S and T.

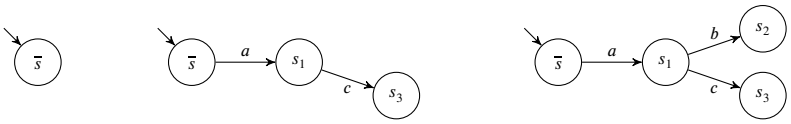


Fig. 11. Three (minimal) LTS of the $\mathit{Impl}(S)$ of the CMTS S of Fig. 10; all other LTS in $\mathit{Impl}(S)$ are strongly bisimilar to one of these three (cf. Footnote 1, Page 5).

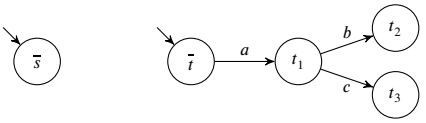


Fig. 12. Two (minimal) LTS of the $\mathit{Impl}(T)$ of the CMTS T of Fig. 10; all other LTS in $\mathit{Impl}(T)$ are strongly bisimilar to one of these two (cf. Footnote 1, Page 5); moreover, $\mathit{Impl}(T,\emptyset) = \mathit{Impl}(T) = \mathit{Impl}(S,\{a \,\mathrm{REQ}\, b\})$, where S is the CMTS of Fig. 10.

Lemma 13. *Let* $\mathit{Impl}_C{}'(S,\Phi) = \{\, L \mid L \preceq_c S \wedge L \models \Phi \,\}$. *Then* $\mathit{Impl}_C{}'(S,\Phi) = \mathit{Impl}_C(S,\Phi)$.

We now extend the notion of thorough refinement to MTSC. We consider two MTSC (with different set of constraints) to be in a thorough refinement relation whenever the implementations of one MTSC are included into the implementations of the other.

Definition 19 *(MTSC thorough refinement). An MTSC* (S,Φ_S) *is a* thorough refinement *of an MTSC* (T,Φ_T)*, denoted by* $(S,\Phi_S) \preceq_{cct} (T,\Phi_T)$*, if* $\mathit{Impl}_C(S,\Phi_S) \subseteq \mathit{Impl}_C(T,\Phi_T)$.

Fig. 9 shows an example where CMTS thorough refinement does not imply MTSC thorough refinement. Moreover, Fig. 10 shows that MTSC thorough refinement does not imply CMTS thorough refinement.

The relation between thorough refinement and refinement of MTSC is similar to the case of MTS, i.e., MTSC refinement is not complete. Indeed, MTSC thorough refinement does not imply MTSC refinement. For example, consider Figs. 10–12. It holds that $(S,\{a \,\mathrm{REQ}\, b\}) \preceq_{cct} (T,\emptyset)$, but $(S,\{a \,\mathrm{REQ}\, b\}) \not\preceq_{cc} (T,\emptyset)$. Similarly, $(T,\emptyset) \preceq_{cct} (S,\{a \,\mathrm{REQ}\, b\})$, but $(T,\emptyset) \not\preceq_{cc} (S,\{a \,\mathrm{REQ}\, b\})$.

Furthermore, MTSC thorough refinement does not imply CMTS thorough refinement. Continuing the previous example, depicted in Figs. 10–12, it holds that $(S,\{a \,\mathrm{REQ}\, b\}) \preceq_{cct} (T,\emptyset)$, but $S \not\preceq_{ct} T$ (note that $T \preceq_{ct} S$).

The next theorem shows that MTSC refinement implies MTSC thorough refinement.

Theorem 11 *($\preceq_{cc}$ implies $\preceq_{cct}$). Let* (S,Φ) *and* (T,Φ') *be two MTSC. Then* $(S,\Phi) \preceq_{cc} (T,\Phi')$ *implies* $(S,\Phi) \preceq_{cct} (T,\Phi')$.

The equivalence between the two notions of refinement is desirable as it provides both a precise and an efficient way of characterising refinement.

We now introduce a stronger notion of thorough refinement, for which we can prove the equivalence with the notion of refinement of MTSC. This stronger notion is built upon CMTS thorough refinement.

Definition 20 *(MTSC strong thorough refinement). An MTSC* (S,Φ) *is a* strong thorough refinement *of an MTSC* (T,Φ')*, denoted by* $(S,\Phi) \preceq_{ccst} (T,\Phi')$*, if* $S \preceq_{ct} T$ *and* $\Phi \implies \Phi'$.

We show that strong thorough refinement is indeed a stronger notion than thorough refinement of MTSC.

Theorem 12 *($\preceq_{ccst}$ implies $\preceq_{cct}$). Let (S,Φ) and (T,Φ') be two MTSC. Then $(S,\Phi) \preceq_{ccst} (T,\Phi')$ implies $(S,\Phi) \preceq_{cct} (T,\Phi')$.*

Consider again the example depicted in Figs. 10–12. It holds that $(T,\{a\,\mathrm{REQ}\,b\}) \preceq_{ccst} (S,\{a\,\mathrm{REQ}\,b\})$ and $(T,\{a\,\mathrm{REQ}\,b\}) \preceq_{cct} (S,\{a\,\mathrm{REQ}\,b\})$. Indeed, it holds that $Impl_C(T,\{a\,\mathrm{REQ}\,b\}) = Impl_C(T,\emptyset)$ and $T \preceq_{ct} S$.

We now prove the correspondence between strong thorough refinement and refinement of MTSC. This means that MTSC refinement is complete with respect to the notion of strong thorough refinement.

Theorem 13 *($\preceq_{cc}$ and $\preceq_{ccst}$ coincide). Let (S,Φ') and (T,Φ) be MTSC. Then $(S,\Phi) \preceq_{cc} (T,\Phi')$ if and only if $(S,\Phi) \preceq_{ccst} (T,\Phi')$.*

We now conclude this section by discussing the notions of MTSC strong bisimilarity and MTSC thorough equivalence.

Definition 21 *(MTSC strong bisimilarity). Two MTSC (S,Φ_S) and (T,Φ_T) are* MTSC strongly bisimilar*, denoted by $(S,\Phi_S) \sim_{cc} (T,\Phi_T)$, if and only if $S \sim_c T$ and $\Phi_S \iff \Phi_T$.*

Note that, by Theorem 2, the notion of MTSC equivalence coincides with MTSC strong bisimilarity. It is easy to see that MTSC strong bisimilarity implies CMTS strong bisimilarity.

Lemma 14 *($\sim_{cc}$ implies $\sim_c$). Let (S,Φ_S) and (T,Φ_T) be two MTSC. Then $(S,\Phi_S) \sim_{cc} (T,\Phi_T)$ implies $S \sim_c T$.*

We introduce MTSC thorough equivalence, defined as the equivalence of the set of implementations. Similar to the case of CMTS, we do not consider MTSC thorough bisimilarity, as MTSC thorough refinement relies on the equivalence of the set of implementations.

Definition 22 *(MTSC thorough equivalence). Two MTSC (S,Φ_S) and (T,Φ_T) are* MTSC thorough equivalent*, denoted by $(S,\Phi_S) \equiv_{cct} (T,\Phi_T)$, if and only if $Impl_C(S,\Phi_S) = Impl_C(T,\Phi_T)$.*

Similarly to the case of CMTS, we have that MTSC strong bisimilarity implies MTSC thorough equivalence.

Lemma 15 *($\sim_{cc}$ implies $\equiv_{cct}$). Let (S,Φ_S) and (T,Φ_T) be two MTSC. Then $(S,\Phi_S) \sim_{cc} (T,\Phi_T)$ implies $(S,\Phi_S) \equiv_{cct} (T,\Phi_T)$*

The converse does not hold. Consider once again the examples depicted in Figs. 10–12, and the MTSC $(S,\{a\,\mathrm{REQ}\,b\})$ and $(T,\{a\,\mathrm{REQ}\,b\})$. It holds that $(S,\{a\,\mathrm{REQ}\,b\}) \equiv_{cct} (T,\{a\,\mathrm{REQ}\,b\})$, as can be seen in Fig. 10 (note that $Impl_C(T,\{a\,\mathrm{REQ}\,b\}) = Impl_C(T,\emptyset)$). For the converse, since $S \nsim_c T$, we have that $(S,\{a\,\mathrm{REQ}\,b\}) \nsim_{cc} (T,\{a\,\mathrm{REQ}\,b\})$. Moreover, whilst it holds that $(T,\{a\,\mathrm{REQ}\,b\}) \equiv_{cct} (T,\emptyset)$, we have that $(T,\{a\,\mathrm{REQ}\,b\}) \nsim_{cc} (T,\emptyset)$.

Finally, we introduce the notion of MTSC strong thorough equivalence.

Definition 23 *(MTSC strong thorough equivalence). MTSC (S,Φ_S) and (T,Φ_T) are* MTSC strong thorough equivalent*, denoted by $(S,\Phi_S) \equiv_{ccst} (T,\Phi_T)$, if and only if $(S,\Phi_S) \equiv_{ct} (T,\Phi_T)$ and $\Phi_S \iff \Phi_T$.*

Differently from MTSC thorough equivalence, MTSC strong thorough equivalence does coincide with MTSC strong bisimilarity.

Theorem 14 *($\equiv_{ccst}$ and $\sim_{cc}$ coincide). Let (S,Φ_S) and (T,Φ_T) be MTSC. Then $(S,\Phi_S) \equiv_{ccst} (T,\Phi_T)$ if and only if $(S,\Phi_S) \sim_{cc} (T,\Phi_T)$.*

Consider again the previous example, depicted in Figs. 10–12, and recall that $(T,\{a\,\mathrm{REQ}\,b\}) \nsim_{cc} (T,\emptyset)$ and $(S,\{a\,\mathrm{REQ}\,b\}) \nsim_{cc} (T,\{a\,\mathrm{REQ}\,b\})$. We see that $(T,\{a\,\mathrm{REQ}\,b\}) \not\equiv_{ccst} (T,\emptyset)$. Moreover, also $(S,\{a\,\mathrm{REQ}\,b\}) \not\equiv_{ccst} (T,\{a\,\mathrm{REQ}\,b\})$.

6. Conclusion and future work

In this paper, we have addressed open problems in the coherent fragment of MTS (CMTS). We have provided a polished formal definition of CMTS, introduced the notions of refinement, thorough refinement, and equivalence of CMTS, and addressed their relations. We have shown that CMTS refinement is both sound and—contrary to MTS—complete, i.e., thorough refinement of CMTS coincides with CMTS refinement.

Furthermore, we have introduced a notion of syntactic CMTS refinement that extends the SPL notion of product derivation from [19] and enjoys the persistent transition removal property. We have demonstrated that CMTS refinement is equivalent to syntactic CMTS refinement modulo strong bisimilarity. We have extended the correspondence between CMTS refinement and CMTS thorough refinement to the notions of equivalence and thorough equivalence of CMTS.

Finally, we have introduced and studied such refinement and equivalence relations also for CMTS with constraints (MTSC). We have demonstrated that MTSC allow expressing alternative behaviour at the cost of losing the correspondence between refinement and thorough refinement. However, MTSC refinement is complete with respect to MTSC strong thorough refinement and MTSC thorough equivalence does not coincide with strong bisimilarity, while MTSC strong thorough equivalence does.

Our results provide a solid theoretical foundation for the use of CMTS as a suitable specification theory, capable of expressing non-deterministic behaviour while preserving completeness of refinement, which none of the other existing MTS dialects currently supports (cf. Table 1).

Future work We are currently investigating an alternative notion of modal refinement of MTS. To the best of our knowledge, the research challenge posed in [5] regarding establishing a complete (and reasonable) notion of MTS refinement remains unresolved to date. In this regard, the coinductive notion of CMTS refinement given in Definition 8 suggests a possible solution.

We conjecture that completeness of MTS refinement can be achieved by suitably incorporating into Definition 2 (MTS refinement) the additional constraints imposed by Definition 8 that target the optional transitions of the system to be refined. These constraints could be further generalised as follows: instead of constraining each action to be either necessary or optional throughout the entire model, it could be sufficient to impose this constraint only on the actions labeling transitions reachable by the same sequence of actions. Further research is necessary to confirm or reject this conjecture.

In [42], the issue of so-called "unintuitive" refinements (i.e., implementations not reflecting the expected behaviour) arising from MTS weak refinement was addressed. Similarly, it is also necessary to demonstrate that all MTS refinements disregarded due to the inclusion of the supplementary constraints, violate some currently unknown requirement.

CRediT authorship contribution statement

Davide Basile: Writing – original draft, Visualization, Funding acquisition, Formal analysis, Conceptualization. **Maurice H. ter Beek:** Writing – review & editing, Visualization, Funding acquisition. **Alessandro Fantechi:** Writing – review & editing, Funding acquisition, Conceptualization. **Stefania Gnesi:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

We thank the reviewers for their useful comments.

Partially funded by the Italian MUR PRIN 2020TL3X8X project T-LADIES (Typeful Language Adaptation for Dynamic, Interacting and Evolving Systems), by the MUR PRIN 2022 PNRR P2022A492B project ADVENTURE (ADVancEd iNtegraTed evalUation of Railway systEms), financed by the EU – Next Generation EU, and by the CNR project "Formal Methods in Software Engineering 2.0", CUP B53C24000720005.

Appendix A. Proofs

In this appendix, we provide the proofs of all the results presented in this paper.

A.1. MTS

Lemma 1 *($\sim_m$ implies $\equiv_m$). Let S and T be two MTS. If $S \sim_m T$, then $S \equiv_m T$.*

Proof. Straightforward from the definition of strong bisimilarity. □

A.2. Coherent MTS

Lemma 2 *(coherence). Let S be an MTS such that the following holds:*

$$\forall a \in A_S . (\exists s_1, s_2 \in Q_S : s_1 \xrightarrow{a}_\Box s_2) \implies (\nexists s_3, s_4 \in Q_S : s_3 \xrightarrow{a}_\bigcirc s_4)$$

Then $S' = (Q_S, A_\bigcirc, A_\Box, \overline{s}, \Delta^S_\bigcirc \cup \Delta^S_\Box)$, where $A_\bigcirc = \{ a \mid s \xrightarrow{a} s' \in \Delta^S_\bigcirc \}$ and $A_\Box = \{ a \mid s \xrightarrow{a} s' \in \Delta^S_\Box \}$ is a CMTS.

Proof. By contradiction, assume that S' is not a CMTS, i.e., there exists an $a \in A_\bigcirc \cap A_\Box$. From the non-redundancy of actions, it follows that in S there are two transitions $s_1 \xrightarrow{a}_\Box s_2$ and $s_3 \xrightarrow{a}_\bigcirc s_4$, for some $s_1, s_2, s_3, s_4 \in Q_S$. A contradiction is reached. □

A.3. CMTS refinement

Theorem 1 ($\preceq_c$ *implies* $\preceq_m$). *Let S and T be two CMTS. If $S \preceq_c T$, then $S \preceq_m T$.*

Proof. Trivially, since modal refinement (cf. Definition 2) only requires Cases 1a and 2 of Definition 8. □

Lemma 3. *Let S and T be two CMTS such that $S \preceq_c T$. It holds that $A^S_{\bigcirc} \subseteq A^T_{\bigcirc}$, $A^T_{\square} \subseteq A^S_{\square}$, and $A^S_{\square} \setminus A^T_{\square} \subseteq A^T_{\bigcirc} \setminus A^S_{\bigcirc}$.*

Proof. Let $\mathcal{R}$ be the relation that proves that $S \preceq_c T$. We first prove $A^T_{\square} \subseteq A^S_{\square}$. By contradiction, assume that there exists an $a \in A^T_{\square} \setminus A^S_{\square}$. Hence, there exists a reachable transition $\delta \in \Delta^T_{\square}$ labelled with a (recall that we assume that there are no redundant actions and transitions). Let t be the source state of δ. By the definition of CMTS refinement, there exists a pair $(s,t) \in \mathcal{R}$ for some $s \in Q_S$ such that $s \xrightarrow{a}_{\square} s' \in \Delta_S$ and $a \in A^S_{\square}$. We reached a contradiction.

We now show $A^S_{\bigcirc} \subseteq A^T_{\bigcirc}$. By contradiction, assume that there exists an $a \in A^S_{\bigcirc} \setminus A^T_{\bigcirc}$. Hence, there exists a reachable transition $\delta \in \Delta^S_{\bigcirc}$ labelled with a (again, we assume that there are no redundant actions and transitions). Let s be the source state of δ. By the definition of CMTS refinement, there exists a pair $(s,t) \in \mathcal{R}$ for some $t \in Q_T$ such that $t \xrightarrow{a} t' \in \Delta_T$ and $a \in A_T$. Since $a \notin A^T_{\bigcirc}$, it must be the case that $a \in A^T_{\square}$, hence $a \in A^S_{\square}$. We reached a contradiction.

Finally, we prove $A^S_{\square} \setminus A^T_{\square} \subseteq A^T_{\bigcirc} \setminus A^S_{\bigcirc}$. Let $a \in A^S_{\square} \setminus A^T_{\square}$. Since S is coherent, we have $a \notin A^S_{\bigcirc}$. Moreover, there exists a reachable transition $s \xrightarrow{a} s' \in \Delta_S$ (once again, we assume that there are no redundant actions and transitions). By the definition of CMTS refinement, there exists a pair $(s,t) \in \mathcal{R}$ for some $t \in Q_T$ such that $t \xrightarrow{a} t' \in \Delta_T$, $a \in A_T$ and $(s',t') \in \mathcal{R}$. Since $A^T_{\square} \subseteq A^S_{\square}$, it must be the case that $a \in A^T_{\bigcirc}$. □

Lemma 4. *CMTS refinement is a preorder.*

Proof. For reflexivity, we observe that Case 1c of Definition 8 is never satisfied, so reflexivity trivially holds.

For transitivity, we need to show that $S \preceq_c T$ and $T \preceq_c U$ implies $S \preceq_c U$. Assume that $S \preceq_c T$ and $T \preceq_c U$ hold. Let $\mathcal{R}$ and $\mathcal{R}'$ be the relations that prove $S \preceq_c T$ and $T \preceq_c U$, respectively. Consider the relation $\mathcal{R}'' = \{\, (s,u) \mid (s,t) \in \mathcal{R},\ (t,u) \in \mathcal{R}',\ t \in Q_T \,\}$. We now show that $\mathcal{R}''$ proves $S \preceq_c U$. Firstly, $(\overline{s},\overline{u}) \in \mathcal{R}''$. Then, for all $(s,u) \in \mathcal{R}''$, by construction, there exists a $t \in Q_T$ such that $(s,t) \in \mathcal{R}$ and $(t,u) \in \mathcal{R}'$, and the following holds:

1. Whenever $u \xrightarrow{a} u'$, with a such that
   (a) $a \in A^U_{\square}$. Since $(t,u) \in \mathcal{R}'$, we have $a \in A^T_{\square}$, $t \xrightarrow{a}_{\square} t'$ and $(t',u') \in \mathcal{R}'$. Since $(s,t) \in \mathcal{R}$, we have $a \in A^S_{\square}$, $s \xrightarrow{a}_{\square} s'$, and $(s',t') \in \mathcal{R}$. By construction, $(s',u') \in \mathcal{R}''$;
   (b) $a \in A^U_{\bigcirc}$. Since $(t,u) \in \mathcal{R}'$, we have either
      i. $a \in A_T$, $t \xrightarrow{a} t'$, and $(t',u') \in \mathcal{R}'$. By Lemma 3, $a \notin A^U_{\square}$ implies $a \notin A^T_{\square}$. Since $a \in A^U_{\bigcirc}$ and $a \in A_T$, it must be the case that $a \in A^T_{\bigcirc}$. Since $(s,t) \in \mathcal{R}$, we have either that
         A. $a \in A_S$, $s \xrightarrow{a} s'$, and $(s',t') \in \mathcal{R}$. By construction, we have $(s',u') \in \mathcal{R}''$;
         B. $a \notin A_S$;
      ii. $a \notin A_T$. Since $S \preceq_c T$, by Lemma 3, $a \notin A_S$.
2. Whenever $s \xrightarrow{a} s'$, since $(s,t) \in \mathcal{R}$, we have $t \xrightarrow{a} t'$ and $(s',t') \in \mathcal{R}$. Since $(t,u) \in \mathcal{R}'$, we have $u \xrightarrow{a} u'$ such that $(t',u') \in \mathcal{R}'$. By construction, $(s',u') \in \mathcal{R}''$. □

Theorem 2 ($\equiv_c$ *and* $\sim_c$ *coincide). Let S and T be two CMTS such that $S \equiv_c T$, where $\mathcal{R}$ proves $S \preceq_c T$. Then $\mathcal{R}^T$ proves $T \preceq_c S$.*

Proof. Firstly, by Lemma 3 and the hypothesis, we have $A^S_{\bigcirc} = A^T_{\bigcirc}$ and $A^S_{\square} = A^T_{\square}$. We know that $(\overline{t},\overline{s}) \in \mathcal{R}^T$. Then, for all $(t,s) \in \mathcal{R}^T$, the following holds:

1. Whenever $s \xrightarrow{a} s'$, then:
   (a) if $a \in A^S_{\square}$, by $(s,t) \in \mathcal{R}$ and $S \preceq_c T$, we have $a \in A_T$ and $t \xrightarrow{a} t'$ such that $(s',t') \in \mathcal{R}$. Since $A^S_{\square} = A^T_{\square}$, we have $t \xrightarrow{a}_{\square} t'$ and $(t',s') \in \mathcal{R}^T$;
   (b) if $a \in A^S_{\bigcirc}$, by $(s,t) \in \mathcal{R}$ and $S \preceq_c T$, we have $a \in A_T$ and $t \xrightarrow{a} t'$ such that $(s',t') \in \mathcal{R}$. Hence, $(t',s') \in \mathcal{R}^T$;
2. Whenever $t \xrightarrow{a} t'$, then since $(s,t) \in \mathcal{R}$ and $a \in A_T$, by hypothesis, $a \in A_S$, and thus $s \xrightarrow{a} s'$ such that $(s',t') \in \mathcal{R}$. Hence, $(t',s') \in \mathcal{R}^T$. □

Theorem 3 ($\sim_c$ *and* $\sim_m$ *coincide). Let S and T be two CMTS, then $S \sim_m T$ if and only if $S \sim_c T$.*

Proof. For the direction $S \sim_c T$ implies $S \sim_m T$, by hypothesis, we have $S \preceq_c T$ and $T \preceq_c S$, where $\mathcal{R}$ proves $S \preceq_c T$. By Theorem 2, we have that $\mathcal{R}^T$ proves $T \preceq_c S$. Finally, by Theorem 1, we have that $\mathcal{R}$ shows $S \preceq_m T$ and $\mathcal{R}^T$ shows $T \preceq_m S$.

For the direction $S \sim_m T$ implies $S \sim_c T$, we assume that $S \sim_m T$ and let $\mathcal{R}_b$ be the relation that shows it. By the definition of $\sim_m$, Cases 1a and 2 of Definition 8 are satisfied (for both directions of the refinement) by the definition of modal refinement. We never have Case 1c for both directions of the refinement, because $A_T = A_S$ by Lemma 3. For Case 1b of Definition 8, we have that, for the direction $S \preceq_c T$, for all $(s,t) \in \mathcal{R}_b$: whenever $t \xrightarrow{a}_{\bigcirc} t'$, by the hypothesis $(t,s) \in \mathcal{R}_b^T$, it holds that $a \in A_S$, $s \xrightarrow{a} s'$, and $(t',s') \in R_b^T$; hence $(s',t') \in \mathcal{R}_b$. We also have that, for the direction $T \preceq_c S$, for all $(t,s) \in \mathcal{R}_b^T$: whenever $s \xrightarrow{a}_{\bigcirc} s'$, by $(s,t) \in \mathcal{R}_b$, it holds that $a \in A_T$, $t \xrightarrow{a} t'$, and $(s',t') \in R_b$; hence $(t',s') \in \mathcal{R}_b^T$. □

Theorem 4 *(remove bisimilar states). Let S be a CMTS, let $\mathcal{R}_b$ be a relation showing that $s_1 \sim_c s_2$ for two states $s_1, s_2 \in Q_S$ such that $s_2 \neq \overline{s}$, and let $BS(s_2) = \{ (s,a,s_2) \mid s \in Q_s,\ a \in A_S \}$. Now let $S' = (Q_S \setminus \{s_2\}, A^S_{\bigcirc}, A^S_{\Box}, \overline{s}, \Delta_S \setminus BS(s_2) \cup \{ (s,a,s_1) \mid (s,a,s_2) \in \Delta_S \})$ and let S'' be obtained from S' by removing the unreachable actions, states, and transitions of S', if any. Then it follows that $S \sim_c S''$.*

Proof. Let $\mathcal{R} = \{ (s,s) \mid s \in Q_S \cap Q_{S''} \} \cup \{ (s,s') \mid s \in Q_S \setminus Q_{S''},\ s' \in Q_{S''},\ s \sim_m s' \}$ be the relation that proves that $S \sim_m S''$. By hypothesis, $(\overline{s},\overline{s}) \in \mathcal{R}$. For all $(s,s'') \in \mathcal{R}$, either $s \sim_m s''$ and the statement follows trivially, or $s = s''$. In the latter case, either the transition δ is in $\Delta_S \cap \Delta_{S''}$ (trivial), or we have one of the following two situations:

1. If $\delta \in \Delta_{S''} \setminus \Delta_S$, then by construction $\delta = s'' \xrightarrow{a} s_1$, for some $a \in A_S$. As $s'' = s$, it holds that $s \xrightarrow{a} s_2 \in \Delta_S$ and $s_1 \sim_m s_2$, by hypothesis;
2. If $\delta \in \Delta_S \setminus \Delta_{S''}$, then by construction $\delta = s \xrightarrow{a} s_2$, for some $a \in A_S$. As $s = s''$, it holds that $s'' \xrightarrow{a} s_1 \in \Delta_{S''}$ and $s_1 \sim_m s_2$, by hypothesis. □

Corollary 4. *Let S and T be two CMTS such that $S \preceq_c T$, and let S'' be a CMTS such that $S \sim_m S''$. Then $S'' \preceq_c T$.*

Proof. By the definition of $\sim_m$ and transitivity of $\preceq_c$. □

A.4. Syntactic CMTS refinement

Lemma 5 *(remove action). Let S be a CMTS and, given some $b \in A^S_{\bigcirc}$, let $S' = (Q'_S, A^S_{\bigcirc} \setminus \{b\}, A^S_{\Box}, \overline{s}, \Delta'_S)$, where $Q'_S = \{ s \in Q_S \mid s \text{ is reachable} \}$ and $\Delta'_S = \{ (s,a,s') \in \Delta_S \mid a \in A_S \setminus \{b\},\ s \text{ is reachable} \}$. Then it holds that $S' \preceq_c S$.*

Proof. The relation $\mathcal{R} = \{ (s,s) \mid s \in Q'_S \}$ proves that $S' \preceq_c S$. Firstly, $(\overline{s},\overline{s}) \in \mathcal{R}$. Then for all $(s,s) \in \mathcal{R}$, the following holds:

1. Whenever $s \xrightarrow{a} s' \in \Delta_S$, then
   (a) if $a \in A^S_{\Box}$, then by construction $s \xrightarrow{a}_{\Box} s' \in \Delta_{S'}$ and $(s',s') \in \mathcal{R}$;
   (b) if $a \in A^S_{\bigcirc} \cap A^{S'}$, then by construction $s \xrightarrow{a} s' \in \Delta_{S'}$ and $(s',s') \in \mathcal{R}$;
2. Whenever $s \xrightarrow{a} s' \in \Delta_{S'}$, then by construction $(s',s') \in \mathcal{R}$. □

Lemma 6 *(shift action right). Let S be a CMTS and, given some $b \in A^S_{\bigcirc}$, let $S' = (Q_S, A^S_{\bigcirc} \setminus \{b\}, A^S_{\Box} \cup \{b\}, \overline{s}, \Delta_S)$. Then it holds that $S' \preceq_c S$.*

Proof. The relation $\mathcal{R} = \{ (s,s) \mid s \in Q_S \}$ proves that $S' \preceq_c S$. Firstly, $(\overline{s},\overline{s}) \in \mathcal{R}$. Then for all $(s,s) \in \mathcal{R}$ the following holds:

1. Whenever $s \xrightarrow{a} s' \in \Delta_S$, then
   (a) if $a \in A^S_{\Box}$, then by construction $s \xrightarrow{a}_{\Box} s' \in \Delta_{S'}$ and $(s',s') \in \mathcal{R}$;
   (b) if $a \in A^S_{\bigcirc}$, then by construction $s \xrightarrow{a} s' \in \Delta_{S'}$ and $(s',s') \in \mathcal{R}$;
2. Whenever $s \xrightarrow{a} s' \in \Delta_{S'}$, then by construction $(s',s') \in \mathcal{R}$. □

Theorem 5 *(syntactic CMTS refinement is persistent). Let S and T be two CMTS such that S is a syntactic CMTS refinement of T. Then $S \preceq_c T$ is a persistent refinement.*

Proof. Let $\mathcal{R}$ be the relation that shows that $S \preceq_c T$. Each syntactic CMTS refinement is computed by repeatedly applying the operations from Lemmata 5 and 6. From the proofs of these lemmata, we know that $\mathcal{R} = \{ (s,s) \mid s \in Q_S \}$ and $Q_S \subseteq Q_T$. By contradiction, assume that there exists an $a \in A_S$ not satisfying Definition 11. By Lemma 3, $A_S \subseteq A_T$, and by Definition 8 it holds that for all $a \in A_S$, if $(s,a,s') \in \Delta_S$ then $\exists t \in Q_T . (s,t) \in \mathcal{R}$ and if $(t,a,t') \in \Delta_T$ then $\exists s \in Q_S . (s,t) \in \mathcal{R}$. There are two cases.

First, $|T_a^S| < |T_a^T|$, where $T_a^S = \{ (s,a,s') \in \Delta_S \}$ and $T_a^T = \{ (t,a,t') \in \Delta_T \}$. We have $\Delta_S = \{ (t,a,t') \in \Delta_T \mid a \in A_S \}$. By hypothesis, $T_a^T \setminus T_a^S \neq \emptyset$. Now pick a transition $\delta = (t,a,t') \in T_a^T \setminus T_a^S$. Since $(s,t) \in \mathcal{R}$, for some $s \in Q_S$, by $S \preceq_c T$, whenever $t \xrightarrow{a} t'$, then:

1. If $a \in A_\Box^T$, then $s \xrightarrow{a}_\Box s'$ and $(s',t') \in \mathcal{R}$. Since $s = t$ and $s' = t'$, it follows that $t \xrightarrow{a} t' \in T_a^S$, and thus we reach a contradiction;
2. If $a \in A_\bigcirc^T$ and $a \in A_\bigcirc^S$, then $s \xrightarrow{a} s'$ and $(s',t') \in \mathcal{R}$. Since $s = t$ and $s' = t'$, it follows that $t \xrightarrow{a} t' \in T_a^S$, and thus we reach a contradiction;
3. If $a \in A_\bigcirc^T$ and $a \notin A_\bigcirc^S$, then we reach a contradiction.

Second, $|T_a^T| < |T_a^S|$. By hypothesis, $T_a^S \setminus T_a^T \neq \emptyset$. We pick a transition $\delta = (s,a,s') \in T_a^S \setminus T_a^T$. Since $(s,t) \in \mathcal{R}$, for some $t \in Q_T$, by $S \preceq_c T$ it holds that $t \xrightarrow{a} t'$ and $(s',t') \in \mathcal{R}$. Since $s = t$ and $s' = t'$ it follows that $s \xrightarrow{a} s' \in T_a^T$ and thus we reach a contradiction. □

Theorem 6 *(CMTS refinement and syntactic CMTS refinement). Let S and T be two CMTS such that $S \preceq_c T$ and let $T' = (Q_{T'}, A_\bigcirc^S, A_\Box^S, \bar{t}, \Delta_{T'})$, where $Q_{T'} = \{ s \in Q_T \mid s \text{ is reachable} \}$ and $\Delta_{T'} = \{ (t,a,t') \in \Delta_T \mid a \in A_\bigcirc^S \cup A_\Box^S \}$. Then it holds that $S \sim_c T'$.*

Proof. Since $S \preceq_c T$, by Lemma 3, it holds that $A_\bigcirc^S \subseteq A_\bigcirc^T$, $A_\Box^T \subseteq A_\Box^S$, and $A_\Box^S \setminus A_\Box^T \subseteq A_\bigcirc^T \setminus A_\bigcirc^S$. Hence, since $A_\bigcirc^S \subseteq A_\bigcirc^T$ and $A_\Box^S \subseteq A_T$, it follows that $A_{T'} \subseteq A_T$ and thus the construction of $\Delta_{T'}$ is sound.

Let $\mathcal{R}'$ be the relation that proves that $S \preceq_c T$. We show that $\mathcal{R} = \{(s,t) \mid (s,t) \in \mathcal{R}', t \in Q_{T'}\}$ shows that $S \preceq_m T'$ and $\mathcal{R}^T$ shows $T' \preceq_m S$ (i.e., $S \sim_m T$, cf. Definition 5). We have that $\mathcal{R} \subseteq \mathcal{R}'$. By Theorem 3 $S \sim_m T$ implies $S \sim_c T$. Therefore, $S \sim_m T$ suffices to also prove that $S \preceq_c T'$ and $T' \preceq_c S$. It holds that $(\bar{s},\bar{t}) \in \mathcal{R}$, because $(\bar{s},\bar{t}) \in \mathcal{R}'$ and $\bar{t} \in Q_{T'}$. Then, each $(s,t) \in \mathcal{R}$ is such that $(s,t) \in \mathcal{R}'$, and the following holds:

1. Whenever $t \xrightarrow{a}_\Box t' \in \Delta_{T'}$, there are two cases:
   (a) if $a \in A_\Box^T$, then by construction $t \xrightarrow{a}_\Box t' \in \Delta_T$. Since $(s,t) \in \mathcal{R}'$, by $S \preceq_c T$, we know that $s \xrightarrow{a}_\Box s'$ and $(s',t') \in \mathcal{R}'$. Furthermore, $t' \in Q_{T'}$. Hence $(s',t') \in \mathcal{R}$;
   (b) otherwise, $a \in A_\Box^S \setminus A_\Box^T$ and thus, by Lemma 3, $a \in A_\bigcirc^T \setminus A_\bigcirc^S$ and $t \xrightarrow{a}_\bigcirc t' \in \Delta_T$. Since $(s,t) \in \mathcal{R}'$ and $a \in A_S$, by $S \preceq_c T$, we know that $s \xrightarrow{a} s' \in \Delta_S$ and $(s',t') \in \mathcal{R}'$. Since S is coherent, $s \xrightarrow{a}_\Box s' \in \Delta_S$. Since $t' \in Q_{T'}$, it holds $(s',t') \in \mathcal{R}$;
2. Whenever $s \xrightarrow{a} s'$, since $(s,t) \in \mathcal{R}'$, by $S \preceq_c T$, we know that $t \xrightarrow{a} t' \in \Delta_T$ and $(s',t') \in \mathcal{R}'$. By construction, $t \xrightarrow{a} t' \in \Delta_{T'}$ and $t' \in Q_{T'}$. Hence $(s',t') \in \mathcal{R}$.

We can see that $\mathcal{R}$ is a relation that shows that $S \preceq_m T'$. We now show that $\mathcal{R}^T$ proves $T' \preceq_m S$. Firstly, $(\bar{t},\bar{s}) \in \mathcal{R}^T$ holds. Then, for any pair $(t,s) \in \mathcal{R}^T$, the following holds:

1. Whenever $s \xrightarrow{a}_\Box s'$, since $\Delta_\Box^S \subseteq \Delta_S$, we also have $s \xrightarrow{a} s'$. Since $(s,t) \in \mathcal{R}$, we have $t \xrightarrow{a} t' \in \Delta_{T'}$ and $(s',t') \in \mathcal{R}$. Hence $(t',s') \in \mathcal{R}^T$. Finally, since $a \in A_\Box^S$ and $A_\Box^S = A_\Box^{T'}$, we have $t \xrightarrow{a}_\Box t' \in \Delta_{T'}$;
2. Whenever $t \xrightarrow{a} t' \in \Delta_{T'}$, since $A_\bigcirc^{T'} = A_\bigcirc^S \subseteq A_\bigcirc^T$, by construction $t \xrightarrow{a} t' \in \Delta_T$. Since $(s,t) \in \mathcal{R}'$, $S \preceq_c T$, and $a \in A_\bigcirc^S$, it holds that $s \xrightarrow{a} s'$ and $(s',t') \in \mathcal{R}'$. Since $(s,t) \in \mathcal{R}$, we know that $s \xrightarrow{a} s' \in \Delta_S$, $t \xrightarrow{a} t' \in \Delta_{T'}$, and $(s',t') \in \mathcal{R}'$, and by construction we also have $(s',t') \in \mathcal{R}$. Hence $(t',s') \in \mathcal{R}^T$.

We showed that $\mathcal{R}$ proves $S \preceq_m T'$ and $\mathcal{R}^T$ proves $T' \preceq_m S$. Thus $S \sim_m T'$. □

Corollary 2. *Let S and T be two CMTS such that $S \preceq_c T$, $A_\Box^S = A_\Box^T$, and $A_\bigcirc^S = A_\bigcirc^T$. Then $S \sim_c T$.*

Proof. Straightforward from Theorem 6. □

Corollary 3. *Let $R_S = \{ S' \mid S' \preceq_c S \text{ and } \forall S'' \in R_S \,.\, S'' \not\sim_c S' \}$ be a set of CMTS refinements of S that are not strongly bisimilar between each other. Then, for any CMTS S, it holds that $|R_S| \leq 3^{|A_\bigcirc^S|}$.*

Proof. By Lemma 3, we know that no CMTS refinement of S can be obtained by either removing a necessary action from S or by turning a necessary action of S into an optional action. Hence, for each action $a \in A_\bigcirc^S$, there are three options: either ignore the action, remove the action from A_S, or turn the action into a necessary action. Each choice produces a CMTS refinement of S by Lemmata 5 and 6. Hence, a total of $3^{|A_\bigcirc^S|}$ CMTS refinements is obtained 'syntactically'. By Theorem 6, any other CMTS refinement of S is strongly bisimilar to a CMTS refinement obtained through the above process. □

Lemma 7. *Let S and T be two CMTS such that $S \preceq_c T$ is a non-persistent CMTS refinement. Let $T' = (Q_{T'}, A^T_\bigcirc \cap A_S, A^T_\square, \bar{t}, \Delta_{T'})$, where $Q_{T'} = \{\, s \in Q_T \mid s \text{ is reachable} \,\}$ and $\Delta_{T'} = \{\, (t,a,t') \in \Delta_T \mid a \in A_{T'} \text{ and } t \text{ is reachable} \,\}$. Then either (i) there exist $t_1, t_2 \in Q_{T'}$ such that $t_1 \neq t_2$ and $t_1 \sim_c t_2$, or (ii) there exist $s_1, s_2 \in Q_S$ such that $s_1 \neq s_2$ and $s_1 \sim_c s_2$.*

Proof. First, we have $T' \preceq_c T$ by Lemma 5. Moreover, by Theorem 6, we have $S \sim_c S'$, where $S' = (Q_{T'}, A^S_\bigcirc, A^S_\square, \bar{t}, \Delta_{T'})$. We show that $A^{T'} = A^{S'}$. By Lemma 3, it holds that $A^S_\square \supseteq A^T_\square = A^{T'}_\square$, $A^S_\bigcirc \subseteq A^T_\bigcirc$, and $A^S_\bigcirc \subseteq A_S$; hence $A^S_\bigcirc \subseteq A^{T'}_\bigcirc$. Furthermore, we observe the following:

1. For the direction $A^{T'} \subseteq A^{S'}$, if $a \in A^{T'}_\square$, since $A^{T'}_\square \subseteq A^{S'}_\square$, then $a \in A^{S'}_\square$. If $a \in A^{T'}_\bigcirc$, then $a \in A^T_\bigcirc \cap A_S$, by definition of $A^{T'}_\bigcirc$, hence $a \in A_S = A_{S'}$;
2. For the direction $A^{T'} \supseteq A^{S'}$, if $a \in A^{S'}_\bigcirc$, then $a \in A^{T'}_\bigcirc$ because $A^{S'}_\bigcirc = A^S_\bigcirc \subseteq A^{T'}_\bigcirc$. If $a \in A^{S'}_\square$, then either $a \in A^T_\square = A^{T'}_\square$, and otherwise, by Lemma 3, $A^S_\square \setminus A^T_\square \subseteq A^T_\bigcirc \setminus A^S_\bigcirc$ and $A^{S'}_\square = A^S_\square$, and it holds $a \in A^T_\bigcirc$, thus $a \in A^T_\bigcirc \cap A_S = A^{T'}_\bigcirc$.

We proved $A^{T'} = A^{S'}$. By construction of S' and T', it holds that S' is obtained from T' using the operation from Lemma 6, and so we have that $S' \preceq_c T'$ and, by transitivity, $S \preceq_c T'$.

Note that in T' we have only removed transitions from T whose actions are not present in S. Hence, also $S \preceq_c T'$ is non-persistent. Let $\mathcal{R}$ be the relation that proves $S \preceq_c T'$.

We know that $A^{T'} = A^S$. By Definition 8, it holds that for all $a \in A_S$, if $(s,a,s') \in \Delta_S$ then $\exists\, t \in Q_T \,.\, (s,t) \in \mathcal{R}$, and if $(t,a,t') \in \Delta_T$ then $\exists\, s \in Q_S \,.\, (s,t) \in \mathcal{R}$.

The proof now proceeds by cases of Definition 11.

In the first case there exists an action $a \in A_S$ such that $|T^S_a| < |T^{T'}_a|$, where $T^S_a = \{\, (s,a,s') \in \Delta_S \,\}$ and $T^{T'}_a = \{\, (t,a,t') \in \Delta'_T \,\}$. We now prove that there exists a state $s \in Q_S$ and two states $t_1 \neq t_2 \in Q_{T'}$ such that $(s,t_1),(s,t_2) \in \mathcal{R}$. By contradiction, assume that for all $(s_1,t_1),(s_2,t_2) \in \mathcal{R}$, if $t_1 \neq t_2$, then $s_1 \neq s_2$. Assuming this hypothesis we will reach the contradiction $|T^{T'}_a| = |T^S_a|$. We proceed by iteration on the elements of $T^{T'}_a$ and T^S_a.

At the first iteration, we pick two transitions $(t_1,a,t'_1),(t_2,a,t'_2) \in T^{T'}_a$ (note that $|T^{T'}_a| \geq 2$). Since $a \in A_S$, there exist two transitions $s_1 \xrightarrow{a} s'_1, s_2 \xrightarrow{a} s'_2 \in \Delta_S$ such that $(s_1,t_1),(s'_1,t'_1),(s_2,t_2),(s'_2,t'_2) \in \mathcal{R}$. Moreover, either $t_1 \neq t_2$ or $t'_1 \neq t'_2$. Hence it holds that either $s_1 \neq s_2$ or $s'_1 \neq s'_2$ (by the hypothesis 'for all $(s_1,t_1),(s_2,t_2) \in \mathcal{R}$, if $t_1 \neq t_2$, then $s_1 \neq s_2$'). This means that $|T^S_a| \geq 2$.

At the nth iteration, we have proved that $|T^{T'}_a| \geq n$ and $|T^S_a| \geq n$. Pick a transition $(t_{n+1},a,t'_{n+1}) \in T^{T'}_a$. By the hypothesis $|T^S_a| < |T^{T'}_a|$ we have that, for all transitions picked at previous iterations, either their source (resp., target) is different from the source (resp., target) of the transition at the current iteration. Formally, for all $i \leq n$, $(t_i,a,t'_i) \in T^{T'}_a \,.\, t_i \neq t_{n+1}$ or $t'_i \neq t'_{n+1}$. This means $|T^{T'}_a| \geq n+1$. Since $a \in A_S$, there exists a transition $s_{n+1} \xrightarrow{a} s'_{n+1} \in \Delta_S$ such that $(s_{n+1},t_{n+1}),(s'_{n+1},t'_{n+1}) \in \mathcal{R}$. It holds that, for all $i \leq n$, $(s_i,a,s'_i) \in T^S_a \,.\, s_i \neq s_{n+1}$ or $s'_i \neq s'_{n+1}$ (by the hypothesis 'for all $(s_1,t_1),(s_2,t_2) \in \mathcal{R}$, if $t_1 \neq t_2$, then $s_1 \neq s_2$'). Indeed, if by contradiction, for some i, it were to hold that $s_i = s_{n+1}$ and $s'_i = s'_{n+1}$, then we would have that for either pairs $(s_i,t_i),(s_{n+1},t_{n+1}) \in \mathcal{R}$ or $(s'_i,t'_i),(s'_{n+1},t'_{n+1}) \in \mathcal{R}$ it holds that $t_i \neq t_{n+1}$ and $s_i = s_{n+1}$, or $t'_i \neq t'_{n+1}$ and $s'_i = s'_{n+1}$, respectively, violating our assumption. Hence $|T^S_a| \geq n+1$.

When we reach the size of $T^{T'}_a$ (i.e., the iteration $m = |T^{T'}_a|$) we have proved $|T^{T'}_a| = |T^S_a|$, violating the hypothesis that $S \preceq_c T'$ is non-persistent. So we proved that there exists a state $s \in Q_S$ and two states $t_1 \neq t_2 \in Q_{T'}$ such that $(s,t_1),(s,t_2) \in \mathcal{R}$.

We show that the relation $\mathcal{R}' = \{\, (t_\ell,t_r) \mid t_\ell,t_r \in Q_{T'}, \exists s \in Q_S \text{ such that } (s,t_\ell),(s,t_r) \in \mathcal{R} \,\}$ proves $t_1 \preceq_c t_2$. First, $(t_1,t_2) \in \mathcal{R}'$. Then, for any $(t_\ell,t_r) \in \mathcal{R}'$, it holds $\exists\, s \in Q_S$ such that $(s,t_\ell),(s,t_r) \in \mathcal{R}$ and:

1. Whenever $t_r \xrightarrow{a'} t'_r$ and
   (a) $a' \in A^{T'}_\square$, then by $(s,t_r) \in \mathcal{R}$, we have $a' \in A^S_\square$ and $s \xrightarrow{a'}_\square s' \,.\, (s',t'_r) \in \mathcal{R}$. By $(s,t_\ell) \in \mathcal{R}$, we have $t_\ell \xrightarrow{a'} t'_\ell \,.\, (s',t'_\ell) \in \mathcal{R}$. Since T' is coherent, we have $t_\ell \xrightarrow{a'}_\square t'_\ell$. Hence (t'_ℓ,t'_r) to $\mathcal{R}'$;
   (b) $a' \in A^{T'}_\bigcirc$, then by construction $a' \in A_S$ and by $(s,t_r) \in \mathcal{R}$, we have $s \xrightarrow{a'} s' \,.\, (s',t'_r) \in \mathcal{R}$. By $(s,t_\ell) \in \mathcal{R}$, we have $t_\ell \xrightarrow{a'} t'_\ell \,.\, (s',t'_\ell) \in \mathcal{R}$. Hence (t'_ℓ,t'_r) to $\mathcal{R}'$;
2. Whenever $t_\ell \xrightarrow{a'} t'_\ell$, since $A^{T'} = A^S$, by construction, we have $a' \in A_S$, and since $(s,t_\ell) \in \mathcal{R}$, we have $s \xrightarrow{a} s' \,.\, (s',t'_\ell) \in \mathcal{R}$. Since $(s,t_r) \in \mathcal{R}$, we have $t_r \xrightarrow{a} t'_r \,.\, (s',t'_r) \in \mathcal{R}$. Hence (t'_ℓ,t'_r) to $\mathcal{R}'$.

We now show that $\mathcal{R}'^T$ proves $t_2 \preceq_c t_1$. First consider $(t_2,t_1) \in \mathcal{R}'^T$. For any $(t_r,t_\ell) \in \mathcal{R}'^T$:

1. Whenever $t_\ell \xrightarrow{a'} t'_\ell$, then
   (a) $a' \in A^{T'}_\square$, and by $(t_\ell,t_r) \in \mathcal{R}'$, we have $t_r \xrightarrow{a'} t'_r$. Since T' is coherent, we have $t_r \xrightarrow{a'}_\square t'_r$ and $(t'_\ell,t'_r) \in \mathcal{R}'$, thus $(t'_r,t'_\ell) \in \mathcal{R}'^T$;

(b) $a' \in A^{T'}_{\bigcirc}$ and by $(t_\ell, t_r) \in \mathcal{R}'$, we have $t_r \xrightarrow{a'} t'_r$ and $(t'_\ell, t'_r) \in \mathcal{R}'$, thus $(t'_r, t'_\ell) \in \mathcal{R}'^T$;

2. Whenever $t_r \xrightarrow{a'} t'_r$, then by $(t_\ell, t_r) \in \mathcal{R}'$ and $a' \in A_{T'}$, we have $t_\ell \xrightarrow{a'} t'_\ell$ and $(t'_\ell, t'_r) \in \mathcal{R}'$, thus $(t'_r, t'_\ell) \in \mathcal{R}'^T$.

We have showed that $t_1 \sim_c t_2$.

In the second case, there exists an action $a \in A_S$ such that $|T^{T'}_a| < |T^S_a|$. We prove that there exists a state $t \in Q_{T'}$ and two states $s_1 \neq s_2 \in Q_S$ such that $(s_1, t), (s_2, t) \in \mathcal{R}$. The proof for the second case is symmetrical to the proof for the first case. We report it for completeness. By contradiction, assume that for all $(s_1, t_1), (s_2, t_2) \in \mathcal{R}$, if $s_1 \neq s_2$ then $t_1 \neq t_2$. We will reach the contradiction that $|T^S_a| = |T^{T'}_a|$. We proceed by iteration on the elements of $T^{T'}_a$ and T^S_a. At the first iteration, we pick two transitions $(s_1, a, s'_1) \in T^S_a$ and $(s_2, a, s'_2) \in T^S_a$. Note that since $A_S = A_{T'}$, it must be that $|T^S_a| \geq 2$. Since $a \in A_{T'}$, there exist two transitions $t_1 \xrightarrow{a} t'_1 \in \Delta_{T'}$ and $t_2 \xrightarrow{a} t'_2 \in \Delta_{T'}$ such that $(s_1, t_1), (s'_1, t'_1), (s_2, t_2), (s'_2, t'_2) \in \mathcal{R}$. Moreover, either $t_1 \neq t_2$ or $t'_1 \neq t'_2$. This means that $|T^{T'}_a| \geq 2$.

At the nth iteration, we have proved that $|T^{T'}_a| \geq n$ and $|T^S_a| \geq n$. Pick a transition $(s_{n+1}, a, s'_{n+1}) \in T^S_a$. By the hypothesis $|T^{T'}_a| < |T^S_a|$, we have that, for all $i \leq n$, $(s_i, a, s'_i) \in T^S_a \,.\, s_i \neq s_{n+1}$ or $s'_i \neq s'_{n+1}$. This means that $|T^S_a| \geq n+1$. Since $a \in A_T$, there exists a transition $t_{n+1} \xrightarrow{a} t'_{n+1} \in \Delta_T$ such that $(s_{n+1}, t_{n+1}), (s'_{n+1}, t'_{n+1}) \in \mathcal{R}$. It holds that, for all $i \leq n$, $(t_i, a, t'_i) \in T^{T'}_a \,.\, t_i \neq t_{n+1}$ or $t'_i \neq t'_{n+1}$. Indeed, if by contradiction, for some i, it were to hold that $t_i = t_{n+1}$ and $t'_i = t'_{n+1}$, then we would have that for either pairs $(s_i, t_i), (s_{n+1}, t_{n+1}) \in \mathcal{R}$ or $(s'_i, t'_i), (s'_{n+1}, t'_{n+1}) \in \mathcal{R}$ it holds that $s_i \neq s_{n+1}$ and $t_i = t_{n+1}$, or $s'_i \neq s'_{n+1}$ and $t'_i = t'_{n+1}$, respectively, violating our assumption. Hence $|T^{T'}_a| \geq n+1$.

When we reach the size of T^S_a, i.e. the iteration $m = |T^S_a|$, we have proved $|T^{T'}_a| = |T^S_a|$, violating the hypothesis that $S \preceq_c T'$ is non-persistent. So we proved that there exists a state $t \in Q_{T'}$ and two states $s_1 \neq s_2 \in Q_S$ such that $(s_1, t), (s_2, t) \in \mathcal{R}$.

We show that the relation $\mathcal{R}'' = \{ (s_\ell, s_r) \mid s_\ell, s_r \in Q_S, \exists t \in Q_{T'}$ such that $(s_\ell, t) \in \mathcal{R}, (s_r, t) \in \mathcal{R} \}$ proves $s_1 \preceq_c s_2$. First, $(s_1, s_2) \in \mathcal{R}''$. Then, for any $(s_\ell, s_r) \in \mathcal{R}''$, it holds that $\exists t \in Q_{T'}$ such that $(s_\ell, t) \in \mathcal{R}, (s_r, t) \in \mathcal{R}$ and:

1. Whenever $s_r \xrightarrow{a'} s'_r$ and
   (a) $a' \in A^S_{\square}$, then by $(s_r, t) \in \mathcal{R}$, we have $a' \in A_{T'}$ and $t \xrightarrow{a'} t' \,.\, (s'_r, t') \in \mathcal{R}$. By $(s_\ell, t) \in \mathcal{R}$ and $a' \in A_S$, we have $s_\ell \xrightarrow{a'} s'_\ell, (s'_\ell, t') \in \mathcal{R}$. Since S is coherent, we have $s_\ell \xrightarrow{a'}_{\square} s'_\ell$. Hence (s'_ℓ, s'_r) to $\mathcal{R}''$.
   (b) $a' \in A^S_{\bigcirc}$, then by $(s_r, t) \in \mathcal{R}$, we have $a' \in A_{T'}$ and $t \xrightarrow{a'} t' \,.\, (s'_r, t') \in \mathcal{R}$. By $(s_\ell, t) \in \mathcal{R}$ and $a' \in A_S$, we have $s_\ell \xrightarrow{a'} s'_\ell, (s'_\ell, t') \in \mathcal{R}$. Hence (s'_ℓ, s'_r) to $\mathcal{R}''$.
2. Whenever $s_l \xrightarrow{a'} s'_l$ then since $A^{T'} = A^S$, by construction, we have $a' \in A_{T'}$, and since $(s_l, t) \in \mathcal{R}$, we have $t \xrightarrow{a'} t' \,.\, (s'_l, t') \in \mathcal{R}$. Since $(s_r, t) \in \mathcal{R}$ we have $s_r \xrightarrow{a'} s'_r$. Hence (s'_r, s'_l) to $\mathcal{R}''$.

We now show that $\mathcal{R}''^T$ proves $s_2 \preceq_c s_1$. Firstly, $(s_2, s_1) \in \mathcal{R}''^T$. For any $(s_r, s_\ell) \in \mathcal{R}''^T$:

1. Whenever $s_\ell \xrightarrow{a'} s'_\ell$ and
   (a) $a' \in A^S_{\square}$, then by $(s_\ell, s_r) \in \mathcal{R}''$, we have $s_r \xrightarrow{a'} s'_r$. Since S is coherent, we have $s_r \xrightarrow{a'}_{\square} s'_r$ and $(s'_\ell, s'_r) \in \mathcal{R}''$, thus $(s'_r, s'_\ell) \in \mathcal{R}''^T$;
   (b) $a' \in A^S_{\bigcirc}$, then by $(s_\ell, s_r) \in \mathcal{R}''$, we have $s_r \xrightarrow{a'} s'_r$ and $(s'_\ell, s'_r) \in \mathcal{R}''$, thus $(s'_r, s'_\ell) \in \mathcal{R}''^T$;
2. Whenever $s_r \xrightarrow{a'} s'_r$, then by $(s_\ell, s_r) \in \mathcal{R}''$ and $a' \in S$, we have $s_\ell \xrightarrow{a'} s'_\ell$ and $(s'_\ell, s'_r) \in \mathcal{R}''$, thus $(s'_r, s'_\ell) \in \mathcal{R}''^T$.

We have showed that $s_1 \sim_c s_2$. □

Theorem 7 *(persistent CMTS refinement). Let S and T be two CMTS such that $S \preceq_c T$ and T' is the CMTS from Lemma 7. If both S and T' are minimised, then $S \preceq_c T$ is a persistent CMTS refinement.*

Proof. By contradiction, assume $S \preceq_c T$ is non-persistent. By Lemma 7, it follows that either T' has two states $t_1 \neq t_2$ such that $t_1 \sim_c t_2$, contradicting the fact that T' is minimal, or S has two states $s_1 \neq s_2$ such that $s_1 \sim_c s_2$, contradicting the fact that S is minimal. □

Lemma 8 *(shift action left). Let S and T be two CMTS with $a \in A^S_{\square} \cap A^T_{\bigcirc}$ and such that $S \preceq_c T$. Then $S' \preceq_c T$, where $S' = (Q_S, A^S_{\bigcirc} \cup \{a\}, A^S_{\square} \setminus \{a\}, \overline{s}, \Delta_S)$.*

Proof. Assume that $\mathcal{R}_1$ proves $S \preceq_c T$. Then we show that the relation $\mathcal{R} = \mathcal{R}_1$ proves $S' \preceq_c T$. Firstly, $(\overline{s}, \overline{t}) \in \mathcal{R}$. Then for each $(s, t) \in \mathcal{R}$, it holds:

1. Whenever $t \xrightarrow{a'} t' \in \Delta_T$:
   (a) if $a' \in A^T_{\square}$, then $a' \neq a$. Since $(s,t) \in \mathcal{R}_1$, by $S \preceq_c T$, we have $s \xrightarrow{a'}_{\square} s' \in \Delta_S \cap \Delta_{S'}$ and $(s',t') \in \mathcal{R}_1$. Hence $(s',t') \in \mathcal{R}$;
   (b) if $a' \in A^T_{\bigcirc}$ and $a' \in A_S$, then by $S \preceq_c T$, we have $s \xrightarrow{a'} s' \in \Delta_S \cap \Delta_{S'}$ and $(s',t') \in \mathcal{R}_1$. Hence $(s',t') \in \mathcal{R}$;
   (c) if $a' \in A^T_{\bigcirc}$ and $a' \notin A_S$, then by construction also $a' \notin A_{S'}$;
2. Whenever $s \xrightarrow{a'} s' \in \Delta_{S'}$, then by construction $a' \in A_S$, thus $s \xrightarrow{a'} s' \in \Delta_S$, and since $(s,t) \in \mathcal{R}_1$, then by $S \preceq_c T$, we have $t \xrightarrow{a'} t' \in \Delta_T$ and $(s',t') \in \mathcal{R}_1$. Hence $(s',t') \in \mathcal{R}$.

By construction, $\mathcal{R}$ proves that $S' \preceq_c T$. □

A.5. CMTS thorough refinement

Lemma 9 *($\preceq_c$ implies $\preceq_{ct}$). Given two CMTS S and T, it holds that$S \preceq_c T$ implies $S \preceq_{ct} T$.*

Proof. Pick an implementation $I_S \in Impl_C(S)$. By hypothesis, $S \preceq_c T$ and by the definition of CMTS, we know that implementation $I_S \preceq_c S$. Hence, by transitivity, $I_S \preceq_c T$ and $I_S \in Impl_C(T)$. Hence $Impl_C(S) \subseteq Impl_C(T)$. □

Lemma 10 *($\preceq_{ct}$ implies $\preceq_c$). Given two CMTS S and T, it holds that $S \preceq_{ct} T$ implies $S \preceq_c T$.*

Proof. This proof proceeds as follows. We start from an implementation I_S of S and, by the hypothesis, of T, which is obtained by turning all optional actions of S into necessary actions. Subsequently, starting from that implementation I_S, the operation of Lemma 8 is applied iteratively to show that $S \preceq_c T$.

Let $I_S = (Q_S, \varnothing, A_S, \overline{s}, \Delta_S)$. Since I_S can be obtained from S by repeatedly applying the operation from Lemma 6, we have $I_S \preceq_c S$. Moreover, $I_S \in Impl_C(S)$ by Definition 12. By hypothesis, $I_S \sim_c I_T$ for some implementation $I_T \preceq_c T$. By transitivity, $I_S \preceq_c T$.

We now show that $A^S_{\bigcirc} \subseteq A^T_{\bigcirc}$. For all actions $a \in A^S_{\bigcirc}$, by construction $a \in A^{I_S}_{\square}$ and, by Lemma 3, $a \in A_T$. To show that $a \in A^T_{\bigcirc}$, it suffices to consider another implementation $I'_S = (Q'_S, \varnothing, A^S_{\square}, \overline{s}, \Delta'_S)$, where $Q'_S = \{\, s \in Q_S \mid s \text{ is reachable} \,\}$ and $\Delta'_S = \{\, (s,a,s') \in \Delta_S \mid a \in A^S_{\square} \,\}$. We have that $I'_S \preceq_c S$ by repeatedly applying the operation from Lemma 5 until exhaustion. Moreover, $I'_S \in Impl_C(S)$ by Definition 12. By hypothesis and transitivity, it holds that $I'_S \preceq_c T$. Since, by Lemma 3, $A^T_{\square} \subseteq A^{I'_S}_{\square} = A^S_{\square}$, $a \in A^S_{\bigcirc}$, and $a \in A_T$, it holds that $a \in A^T_{\bigcirc}$.

If $A^S_{\bigcirc} = \varnothing$, then $S = I_S$ and the statement follows. Hence, assume $A^S_{\bigcirc} \neq \varnothing$. We pick an action $a \in A^S_{\bigcirc}$, thus $a \in A^T_{\bigcirc}$. From I_S, T, and a we build a CMTS $S^1 = (Q_S, \{a\}, A_S \setminus \{a\}, \overline{s}, \Delta_S)$ such that $S^1 \preceq_c T$ by Lemma 8. We re-iterate this process. From S^1, T, and an action $b \in A^S_{\bigcirc}$ such that $b \neq a$, we build a CMTS $S^2 = (Q_S, \{a,b\}, A_S \setminus \{a,b\}, \overline{s}, \Delta_S)$ such that $S^2 \preceq_c T$ by Lemma 8. After performing $n = |A^S_{\bigcirc}|$ iterations, we have built a CMTS $S^n = (Q_S, A^S_{\bigcirc}, A_S \setminus A^S_{\bigcirc}, \overline{s}, \Delta_S)$ such that $S^n \preceq_c T$ by Lemma 8. The statement holds by observing that $S^n = S$. □

Theorem 8 *($\preceq_{ct}$ and $\preceq_c$ coincide). CMTS refinement and CMTS thorough refinement coincide.*

Proof. Straightforward from Lemmata 9 and 10. □

Theorem 9 *($\preceq_{ct}$ implies $\preceq_{tr}$). Let S and T be two CMTS. If $S \preceq_{ct} T$, then $S \preceq_{tr} T$.*

Proof. By Lemma 10, it holds that $S \preceq_c T$. By Theorem 1, it holds that $S \preceq_m T$. From [5], we know that $S \preceq_m T$ implies $S \preceq_{tr} T$. □

Theorem 10 *($\equiv_c$ and $\equiv_{ct}$ coincide). Let S and T be two CMTS. Then $S \equiv_c T$ if and only if $S \equiv_{ct} T$.*

Proof. For the direction CMTS equivalence implies CMTS thorough equivalence, by hypothesis, we have $S \preceq_c T$ and $T \preceq_c S$. By Lemma 9, we have $Impl_C(S) \subseteq Impl_C(T)$ and $Impl_C(T) \subseteq Impl_C(S)$, hence $Impl_C(S) = Impl_C(T)$.

For the direction CMTS thorough equivalence implies CMTS equivalence, by hypothesis, we have $Impl_C(S) \subseteq Impl_C(T)$ and, by Lemma 10, it holds that $S \preceq_c T$. Similarly, by hypothesis, we have $Impl_C(T) \subseteq Impl_C(S)$ and, by Lemma 10, it follows that $T \preceq_c S$. By Definition 9, we have $S \equiv_c T$. □

A.6. CMTS with constraints

Lemma 11 *($\preceq_{cc}$ implies $\preceq_c$). Let (S,Φ) and (T,Φ') be two MTSC. Then $(S,\Phi) \preceq_{cc} (T,\Phi')$ implies $S \preceq_c T$.*

Proof. Straightforward from the definition of MTSC. □

Lemma 12. *Let L be an LTS and ϕ and ϕ' be two sets of constraints. It holds that $(L \vDash \phi \wedge \phi \implies \phi')$ implies $(L \vDash \phi')$.*

Proof. Let I be the interpretation of ϕ derived from L according to Definition 15, where $I \vDash \phi$, and since $\phi \implies \phi'$, it follows that $I \vDash \phi'$, i.e., $L \vDash \phi'$. □

Lemma 13. *Let $Impl_C{}'(S,\Phi) = \{\, L \mid L \preceq_c S \wedge L \vDash \Phi \,\}$. Then $Impl_C{}'(S,\Phi) = Impl_C(S,\Phi)$.*

Proof. The inclusion $Impl_C{}'(S,\Phi) \subseteq Impl_C(S,\Phi)$ is trivial. To prove the inclusion $Impl_C(S,\Phi) \subseteq Impl_C{}'(S,\Phi)$, by Definitions 17 and 18, we have that $L \in Impl_C(S,\Phi)$ is defined as $L \preceq_c S$ and $L \vDash \Phi'$, for some Φ' such that $\Phi' \implies \Phi$. By Lemma 12, it follows that $L \vDash \Phi$. By Definition of $Impl_C{}'(S,\Phi)$, we have that $L \in Impl_C{}'(S,\Phi)$. □

Theorem 11 *($\preceq_{cc}$ implies $\preceq_{cct}$). Let (S,Φ) and (T,Φ') be two MTSC. Then $(S,\Phi) \preceq_{cc} (T,\Phi')$ implies $(S,\Phi) \preceq_{cct} (T,\Phi')$.*

Proof. By hypothesis, $S \preceq_c T$ and $\Phi \implies \Phi'$. By Lemma 9, $S \preceq_{ct} T$, i.e., $Impl_C(S) \subseteq Impl_C(T)$. By Lemma 13, $L \in Impl_C(S,\Phi)$ iff $L \in Impl_C(S)$ and $L \vDash \Phi$, and, similarly, $L \in Impl_C(T,\Phi')$ iff $L \in Impl_C(T)$ and $L \vDash \Phi'$. The statement follows by Lemma 12 and the hypothesis $\Phi \implies \Phi'$. □

Theorem 12 *($\preceq_{ccst}$ implies $\preceq_{cct}$). Let (S,Φ) and (T,Φ') be two MTSC. Then $(S,\Phi) \preceq_{ccst} (T,\Phi')$ implies $(S,\Phi) \preceq_{cct} (T,\Phi')$.*

Proof. By hypothesis, $Impl_C(S) \subseteq Impl_C(T)$. We must prove $Impl_C(S,\Phi) \subseteq Impl_C(T,\Phi')$. Let $L \in Impl_C(S,\Phi)$, then by Lemma 13, we have $L \vDash \Phi$. By Lemma 12 and the hypothesis $\Phi \implies \Phi'$, we have $L \vDash \Phi'$. By Lemma 13, we have that $L \preceq_c S$, hence $L \in Impl_C(S)$, and by hypothesis $L \in Impl_C(T)$. By Definition 12, $L \preceq_c T$, hence $L \in Impl_C(T,\Phi')$. □

Theorem 13 *($\preceq_{cc}$ and $\preceq_{ccst}$ coincide). Let (S,Φ') and (T,Φ) be MTSC. Then $(S,\Phi) \preceq_{cc} (T,\Phi')$ if and only if $(S,\Phi) \preceq_{ccst} (T,\Phi')$.*

Proof. For the direction $\preceq_{cc}$ implies $\preceq_{ccst}$, by hypothesis, $S \preceq_c T$ and $\Phi \implies \Phi'$. By Lemma 9, $S \preceq_{ct} T$, from which the statement follows. Similarly, for the direction $\preceq_{ccst}$ implies $\preceq_{cc}$, by hypothesis, $S \preceq_{ct} T$ and $\Phi \implies \Phi'$. By Lemma 10, $S \preceq_c T$, from which the statement follows. □

Lemma 14 *($\sim_{cc}$ implies $\sim_c$). Let (S,Φ_S) and (T,Φ_T) be two MTSC. Then $(S,\Phi_S) \sim_{cc} (T,\Phi_T)$ implies $S \sim_c T$.*

Proof. Straightforward from the definition of MTSC equivalence. □

Lemma 15 *($\sim_{cc}$ implies $\equiv_{cct}$). Let (S,Φ_S) and (T,Φ_T) be two MTSC. Then $(S,\Phi_S) \sim_{cc} (T,\Phi_T)$ implies $(S,\Phi_S) \equiv_{cct} (T,\Phi_T)$*

Proof. By hypothesis, $(S,\Phi_S) \preceq_{cc} (T,\Phi_T)$, and hence $Impl_C(S,\Phi_S) \subseteq Impl_C(T,\Phi_T)$. By hypothesis, also $(T,\Phi_T) \preceq_{cc} (S,\Phi_S)$, hence $Impl_C(S,\Phi_S) \supseteq Impl_C(T,\Phi_T)$. It follows that $Impl_C(S,\Phi_S) = Impl_C(T,\Phi_T)$. □

Theorem 14 *($\equiv_{ccst}$ and $\sim_{cc}$ coincide). Let (S,Φ_S) and (T,Φ_T) be MTSC. Then $(S,\Phi_S) \equiv_{ccst} (T,\Phi_T)$ if and only if $(S,\Phi_S) \sim_{cc} (T,\Phi_T)$.*

Proof. The equivalence follows by Theorem 10. □

References

[1] K.G. Larsen, B. Thomsen, A modal process logic, in: Proceedings 3rd Symposium on Logic in Computer Science (LICS'88), IEEE, 1988, pp. 203–210.

[2] R.M. Keller, Formal verification of parallel programs, Commun. ACM 19 (7) (1976) 371–384, https://doi.org/10.1145/360248.360251.

[3] A. Antonik, M. Huth, K.G. Larsen, U. Nyman, A. Wąsowski, 20 years of modal and mixed specifications, Bull. Eur. Assoc. Theor. Comput. Sci. 95 (2008) 94–129, https://vbn.aau.dk/files/16474238/BEATCS2008.pdf.

[4] J. Křetínský, 30 years of modal transition systems: survey of extensions and analysis, in: L. Aceto, G. Bacci, G. Bacci, A. Ingólfsdóttir, A. Legay, R. Mardare (Eds.), Models, Algorithms, Logics and Tools, in: LNCS, vol. 10460, Springer, 2017, pp. 36–74.

[5] K.G. Larsen, U. Nyman, A. Wąsowski, On modal refinement and consistency, in: L. Caires, V.T. Vasconcelos (Eds.), Proceedings 18th International Conference on Concurrency Theory (CONCUR'07), in: LNCS, vol. 4703, Springer, 2007, pp. 105–119.

[6] N. Beneš, J. Křetínský, K.G. Larsen, J. Srba, EXPTIME-completeness of thorough refinement on modal transition systems, Inf. Comput. 218 (2012) 54–68, https://doi.org/10.1016/j.ic.2012.08.001.

[7] D. Fischbein, S. Uchitel, V.A. Braberman, A foundation for behavioural conformance in software product line architectures, in: Proceedings ISSTA Workshop on Role of Software Architecture for Testing and Analysis (ROSATEA'06), ACM, 2006, pp. 39–48.

[8] A. Fantechi, S. Gnesi, A behavioural model for product families, in: Proceedings 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT International Symposium on Foundations of Software Engineering (ESEC/FSE'07), ACM, 2007, pp. 521–524.

[9] A. Fantechi, S. Gnesi, Formal modeling for product families engineering, in: Proceedings 12th International Software Product Line Conference (SPLC'08), IEEE, 2008, pp. 193–202.

[10] S. Apel, D.S. Batory, C. Kästner, G. Saake, Feature-Oriented Software Product Lines: Concepts and Implementation, Springer, 2013.
[11] K.C. Kang, S.G. Cohen, J.A. Hess, W.E. Novak, A.S. Peterson, Feature-Oriented Domain Analysis (FODA) Feasibility Study, Tech. Rep. CMU/SEI-90-TR-21, Carnegie Mellon University, November 1990, http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=11231.
[12] P.-Y. Schobbens, P. Heymans, J.-C. Trigaux, Feature diagrams: a survey and a formal semantics, in: Proceedings 14th IEEE International Conference on Requirements Engineering (RE'06), IEEE, 2006, pp. 136–145.
[13] M.H. ter Beek, K. Schmid, H. Eichelberger, Textual Variability Modeling Languages: An Overview and Considerations, Proceedings 23rd International Systems and Software Product Line Conference (SPLC'19), vol. 2, ACM, 2019, pp. 82:1–82:7.
[14] K.G. Larsen, U. Nyman, A. Wąsowski, Modal I/O automata for interface and product line theories, in: R. De Nicola (Ed.), Proceedings 16th European Symposium on Programming (ESOP'07), in: LNCS, vol. 4421, Springer, 2007, pp. 64–79.
[15] K. Lauenroth, K. Pohl, S. Töhning, Model checking of domain artifacts in product line engineering, in: Proceedings 24th International Conference on Automated Software Engineering (ASE'09), IEEE, 2009, pp. 269–280.
[16] P. Asirelli, M.H. ter Beek, A. Fantechi, S. Gnesi, A logical framework to deal with variability, in: D. Méry, S. Merz (Eds.), Proceedings 8th International Conference on Integrated Formal Methods (iFM'10), in: LNCS, vol. 6396, Springer, 2010, pp. 43–58.
[17] P. Asirelli, M.H. ter Beek, A. Fantechi, S. Gnesi, Formal description of variability in product families, in: Proceedings 15th International Software Product Line Conference (SPLC'11), IEEE, 2011, pp. 130–139.
[18] M.H. ter Beek, F. Mazzanti, A. Sulova, VMC: a tool for product variability analysis, in: D. Giannakopoulou, D. Méry (Eds.), Proceedings 18th International Symposium on Formal Methods (FM'12), in: LNCS, vol. 7436, Springer, 2012, pp. 450–454.
[19] M.H. ter Beek, A. Fantechi, S. Gnesi, F. Mazzanti, Modelling and analysing variability in product families: model checking of modal transition systems with variability constraints, J. Log. Algebraic Methods Program. 85 (2) (2016) 287–315, https://doi.org/10.1016/j.jlamp.2015.11.006.
[20] A. Fantechi, S. Gnesi, Refinement of behavioural models for variability description, in: L. Petre, E. Sekerinski (Eds.), From Action Systems to Distributed Systems: The Refinement Approach, Chapman and Hall/CRC, 2016, pp. 155–169, Ch. 11.
[21] D. Dams, R. Gerth, O. Grumberg, Abstract interpretation of reactive systems, ACM Trans. Program. Lang. Syst. 19 (2) (1997) 253–291, https://doi.org/10.1145/244795.244800.
[22] N. Beneš, J. Křetínský, K.G. Larsen, J. Srba, On determinism in modal transition systems, Theor. Comput. Sci. 410 (41) (2009) 4026–4043, https://doi.org/10.1016/j.tcs.2009.06.009.
[23] N. Beneš, J. Křetínský, K.G. Larsen, M.H. Møller, J. Srba, Parametric modal transition systems, in: T. Bultan, P. Hsiung (Eds.), Proceedings 9th International Symposium on Automated Technology for Verification and Analysis (ATVA'11), in: LNCS, vol. 6996, Springer, 2011, pp. 275–289.
[24] J. Kretínský, S. Sickert, On refinements of Boolean and parametric modal transition systems, in: Z. Liu, J. Woodcock, H. Zhu (Eds.), Proceedings 10th International Colloquium on Theoretical Aspects of Computing (ICTAC'13), in: LNCS, vol. 8049, Springer, 2013, pp. 213–230.
[25] N. Beneš, J. Křetínský, K.G. Larsen, M.H. Møller, S. Sickert, J. Srba, Refinement checking on parametric modal transition systems, Acta Inform. 52 (2–3) (2015) 269–297, https://doi.org/10.1007/s00236-015-0215-4.
[26] A. Classen, P. Heymans, P.-Y. Schobbens, A. Legay, J.-F. Raskin, Model checking lots of systems: efficient verification of temporal properties in software product lines, in: Proceedings 32nd ACM/IEEE International Conference on Software Engineering (ICSE'10), ACM, 2010, pp. 335–344.
[27] A. Classen, M. Cordy, P.-Y. Schobbens, P. Heymans, A. Legay, J.-F. Raskin, Featured transition systems: foundations for verifying variability-intensive systems and their application to LTL model checking, IEEE Trans. Softw. Eng. 39 (8) (2013) 1069–1089, https://doi.org/10.1109/TSE.2012.86.
[28] M. Cordy, X. Devroey, A. Legay, G. Perrouin, A. Classen, P. Heymans, P.-Y. Schobbens, J.-F. Raskin, A decade of featured transition systems, in: M.H. ter Beek, A. Fantechi, L. Semini (Eds.), From Software Engineering to Formal Methods and Tools, and Back, in: LNCS, vol. 11865, Springer, 2019, pp. 285–312.
[29] A. Gruler, M. Leucker, K.D. Scheidemann, Modeling and model checking software product lines, in: G. Barthe, F.S. de Boer (Eds.), Proceedings 10th International Conference on Formal Methods for Open Object-Based Distributed Systems (FMOODS'08), in: LNCS, vol. 5051, Springer, 2008, pp. 113–131.
[30] S. Gnesi, M. Petrocchi, Towards an Executable Algebra for Product Lines, Proceedings 16th International Software Product Line Conference (SPLC'12), vol. 2, ACM, 2012, pp. 66–73.
[31] M.H. ter Beek, A. Lluch Lafuente, M. Petrocchi, Combining Declarative and Procedural Views in the Specification and Analysis of Product Families, Proceedings 17th International Software Product Line Conference (SPLC'13), vol. 2, ACM, 2013, pp. 10–17.
[32] M.H. ter Beek, E.P. de Vink, Using mCRL2 for the analysis of software product lines, in: Proceedings 2nd FME Workshop on Formal Methods in Software Engineering (FormaliSE'14), IEEE, 2014, pp. 31–37.
[33] M. Tribastone, Behavioral relations in a process algebra for variants, in: Proceedings 18th International Software Product Line Conference (SPLC'14), ACM, 2014, pp. 82–91.
[34] M.H. ter Beek, A. Legay, A. Lluch Lafuente, A. Vandin, A framework for quantitative modeling and analysis of highly (re)configurable systems, IEEE Trans. Softw. Eng. 46 (3) (2020) 321–345, https://doi.org/10.1109/TSE.2018.2853726.
[35] P. Chrszon, C. Dubslaff, S. Klüppelholz, C. Baier, ProFeat: feature-oriented engineering for family-based probabilistic model checking, Form. Asp. Comput. 30 (1) (2018) 45–75, https://doi.org/10.1007/s00165-017-0432-4.
[36] H. Beohar, B. König, S. Küpper, A. Silva, Conditional transition systems with upgrades, Sci. Comput. Program. 186 (2020), https://doi.org/10.1016/j.scico.2019.102320.
[37] R. Muschevici, J. Proença, D. Clarke, Feature nets: behavioural modelling of software product lines, Softw. Syst. Model. 15 (4) (2016) 1181–1206, https://doi.org/10.1007/s10270-015-0475-z.
[38] M.H. ter Beek, F. Damiani, S. Gnesi, F. Mazzanti, L. Paolini, On the expressiveness of modal transition systems with variability constraints, Sci. Comput. Program. 169 (2019) 1–17, https://doi.org/10.1016/j.scico.2018.09.006.
[39] H. Beohar, M. Varshosaz, M.R. Mousavi, Basic behavioral models for software product lines: expressiveness and testing pre-orders, Sci. Comput. Program. 123 (2016) 42–60, https://doi.org/10.1016/j.scico.2015.06.005.
[40] M. Varshosaz, H. Beohar, M.R. Mousavi, Basic behavioral models for software product lines: revisited, Sci. Comput. Program. 168 (2018) 171–185, https://doi.org/10.1016/j.scico.2018.09.001.
[41] M. Varshosaz, L. Luthmann, P. Mohr, M. Lochau, M.R. Mousavi, Modal transition system encoding of featured transition systems, J. Log. Algebraic Methods Program. 106 (2019) 1–28, https://doi.org/10.1016/j.jlamp.2019.03.003.
[42] D. Fischbein, V.A. Braberman, S. Uchitel, A sound observational semantics for modal transition systems, in: M. Leucker, C. Morgan (Eds.), Proceedings 6th International Colloquium on Theoretical Aspects of Computing (ICTAC'09), in: LNCS, vol. 5684, Springer, 2009, pp. 215–230.