Math. Struct. in Comp. Science (2012), *vol.* 22, *pp.* 581–617. © Cambridge University Press 2012
doi:10.1017/S0960129511000697 First published online 8 May 2012

Extending modal transition systems with structured labels

SEBASTIAN S. BAUER[†], LINE JUHL[‡], KIM G. LARSEN[§], AXEL LEGAY[¶] and JIŘÍ SRBA[‖]

[†]*Institut für Informatik, Ludwig-Maximilians-Universität München, München, Germany*
Email: bauerse@pst.ifi.lmu.de
[‡]*Department of Computer Science, Aalborg University, Selma Lagerlöfs Vej 300, DK-9220 Aalborg Ø, Denmark*
Email: linej@cs.aau.dk
[§]*Department of Computer Science, Aalborg University, Selma Lagerlöfs Vej 300, DK-9220 Aalborg Ø, Denmark*
Email: kgl@cs.aau.dk
[¶]*INRIA/IRISA, Rennes, France*
Email: axel.legay@irisa.fr
[‖]*Department of Computer Science, Aalborg University, Selma Lagerlöfs Vej 300, DK-9220 Aalborg Ø, Denmark*
Email: srba@cs.aau.dk

Received 22 December 2010; revised 2 November 2011

We introduce a novel formalism of label-structured modal transition systems that combines the classical may/must modalities on transitions with structured labels that represent quantitative aspects of the model. On the one hand, the specification formalism is general enough to include models like weighted modal transition systems and allows system developers to employ more complex label refinement than in previously studied theories. On the other hand, the formalism maintains the desirable properties required by any specification theory supporting compositional reasoning. In particular, we study modal and thorough refinement, determinisation, parallel composition, conjunction, quotient and logical characterisation of label-structured modal transition systems.

1. Introduction

Modern computing systems are often large and complex assemblies of numerous reactive and interacting components. The components are often designed by independent teams working under a common agreement of what the interface of each component should be. Consequently, the search for mathematical foundations that support *compositional*

[†] Sebastian Bauer is partially supported by the German Academic Exchange Service (DAAD), grant D/10/46169, and by MT-LAB, VKR Centre of Excellence.
[¶] Axel Legay is partially supported by an 'action de recherche collaborative' ARC (TP)I, INRIA and the European Project COMBEST.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

reasoning about interfaces is a major research goal. The framework should support the inference of properties of the global implementation, and the design and reuse of components.

In a logical setting, interfaces are specifications and components that implement an interface are understood as models/implementations. Specification theories should support various features including:

(1) *refinement*, which allows us to compare specifications as well as to replace one specification by another in a larger design;

(2) *structural composition*, which allows us to combine specifications of different components;

(3) *logical conjunction*, which expresses the intersection of the set of requirements expressed by two or more specifications;

(4) a *quotient operator*, which, given two specifications S and T, synthesises the largest (with respect to refinement) specification that can be composed with S in order to refine T.

For sequential systems, the classical notion of a Denotational Semantics, which was founded by Scott and Strachey, provides a rich mathematical foundation for successfully describing the semantics of many sequential programming languages and systems (Gordon 1979; Stoy 1977) where components, that is, programs, are basically modelled as computable functions from the domain of input values to the domain of output values. Most importantly, the semantics of a composite program is expressed in terms of the semantics of its components, thus supporting compositional reasoning. A similar well-established specification theory for sequential systems is that of Hoare Logic (Hoare 1969), where a program is specified by pairs of pre- and post-conditions on states. In particular, Hoare Logic comes equipped with all the ingredients required for a specification theory as described above, with *'strongest postcondition'* and *'weakest precondition'* transformers providing the means for composing and quotienting specifications with respect to sequential composition.

Process algebras such as CCS (Milner 1980) and CSP (Hoare 1985) provide a corresponding mathematical foundation for concurrent and reactive systems. Here systems are semantically understood as labelled transition systems (Plotkin 1981) describing their interaction capabilities and dynamic evolution. Based on the labelled transition system semantics, several *equivalences* and *preorders* have been proposed (van Glabbeek 1990) in order to capture different aspects of the extensional behaviour of a process. This results in specification theories where both the specification and the implementation are expressed within the same formalism, for example, CCS, and with a preferred preorder or equivalence determining the *satisfaction* of an implementation with respect to a specification. To achieve the goal of compositional analysis, great care has normally been taken to ensure that the preorders and equivalences are substitutive with respect to the various process constructions, for example, parallel composition, including the notions of *observational equivalence* (Milner 1980) and *bisimulation equivalence* (Park 1981; Milner 1983) used in CCS.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

The specification theory of *modal transition systems* (Larsen and Thomsen 1988b) grew out of a series of attempts to achieve a more flexible and easy-to-use compositional development methodology for CCS. As initial motivation, consider the so-called *stepwise refinement method* to be carried out in CCS. A specification S (in CCS) of some desired system is given, and the task is to find an implementation I of S such that $I \equiv S$, where $\equiv$ may be observational (or bisimulation) equivalence. In a first refinement-step, S might be refined to a composite specification of the form $C[S_1, S_2]$, where the context C is some CCS-construct (for example, parallel composition), and S_1 and S_2 are subspecifications. Now we may have different teams working independently towards implementations I_1 and I_2 of the subspecifications S_1 and S_2. Given the congruence property of observational equivalence, it will now suffice to establish the equivalences below to conclude, in a compositional manner, that the assembled implementation $C[I_1, I_2]$ satisfies the original specification S:

$$C[S_1, S_2] \equiv S \tag{1}$$

$$I_1 \equiv S_1 \tag{2}$$

$$I_2 \equiv S_2 \tag{3}$$

However, looking more carefully at the stepwise refinement above, we notice that (2) and (3) require S_i and I_i $(i = 1, 2)$ to be proved congruent, that is, interchangeable, in *any* context and not just interchangeable in the context of C in which they are actually going to be placed. We are therefore asked to prove more than seems necessary. Moreover, the subspecifications S_i $(i = 1, 2)$ may have to specify behaviours that are not at all relevant in the context C. Again it seems that the above compositional analysis can be substantially harder than necessary.

In order to reduce the work of (1–3), the notion of *context-dependent* or *relativised* bisimulation was introduced in Larsen (1985; 1987). Here, in order to reduce the overall effort, the observational equivalence $\equiv$ is relativised with information about the context C. The required proofs $I_i \equiv S_i$ can thus be replaced with proofs of the more specific $I_i \equiv_e S_i$, where e is some partial information (stated as a labelled transition system) about the context C. Larsen and Milner (1987; 1992) applied the relativised bisimulation to the compositional verification of the Alternating Bit Protocol, and Pierce and Sangiorgi (2000) introduced a proof technique for polymorphic Pi-Calculus based on polymorphic types, which can be regarded as a 'disciplined instance' of relativised bisimulation. A more recent use of relativised bisimulation includes the use of environment information to produce environment-specific (reduced) code from embedded system specifications (Larsen *et al.* 2005) and to generate relevant test sequences from real-time specifications (Larsen *et al.* 2004).

The introduction of modal transition system was pre-dated by the simpler formalisms of *partial specifications* (Larsen and Thomsen 1988a) and the corresponding notion of partial bisimulation. Roughly speaking, partial specifications are labelled transition systems with certain (specification) states being interpreted as completely unspecified. As such, partial specifications are very similar to that of processes with divergence and so-called pre-bisimulation (Stirling 1987; Stirling and Walker 1989). However, though it allows simple and intuitive subspecifications for several examples, the specification theory constituted by partial specifications is not closed under either conjunction or quotienting.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

In comparison with partial specifications, the introduction of modal transition systems (Larsen and Thomsen 1988b) resulted in a specification theory that was much closer to logic (see Boudol and Larsen (1992) for a logical characterisation of the expressive power of modal transition systems), and thus still having a behavioural semantics allowing for easy composition with respect to process constructions. In short, modal transition systems are labelled transition systems equipped with two types of transitions: *must* transitions that are mandatory for any implementation and *may* transitions that are optional for an implementation. Hence, the refinement of modal transition systems essentially consists of iteratively resolving the unsettled status of may transitions: either by removing them or by turning them into must transitions.

It is generally accepted that modal transition systems and their extensions (see, for example, Raclet (2008)) possess all the requirements of a good specification theory. There is also no doubt that the formalism is expressive enough to encode complex industrial problems (examples include COMBEST and SPEEDS[†]). Moreover, the model has applications in other contexts, which include the verification of product lines (Fischbein *et al.* 2006; Gruler *et al.* 2008; Larsen *et al.* 2007) and a counterexample-guided abstraction refinement technique for transition systems (Godefroid *et al.* 2001).

When searching for a specification theory for embedded systems, it is not only the functional requirements (Larsen *et al.* 2007; Feuillade and Pinchinat 2007; Bauer *et al.* 2010) of system behaviours that are of importance. The theory should also be capable of expressing constraints for several non-functional properties such as timing, energy-consumption and band-width. Recently, such efforts have been of great interest to the theory community (Caillaud *et al.* 2010; Katoen *et al.* 2009; Delahaye *et al.* 2011; Bertrand *et al.* 2009a; Bertrand *et al.* 2009b; David *et al.* 2010).

It is common for similar proof techniques to be used to argue about the specification formalisms for different non-functional extensions. In the current paper, we present a specification theory that unifies several of the proof techniques described in the literature by introducing a general framework of *label-structured modal transition system*. Specialisations of the framework include, apart from well-known instances like unlabelled/labelled modal transition systems, a new specification theory for weighted and multi-weighted transition systems that were studied only recently (Juhl *et al.* 2010). Other formalisms, such as timed modal transitions systems, can be embedded into the framework, as argued in Bertrand *et al.* (2009a), where the authors show that operations defined on some classes of timed modal specifications can be reduced to questions on modal transition systems by using a region-based abstraction.

In the current paper, we study some classical issues related to the formalism of label-structured modal transition systems:

(i) modal and thorough refinement, consistency and pruning, determinism and deterministic hull (in Section 2);
(ii) parallel composition, conjunction and quotient (in Section 3); and
(iii) logical characterisation including generalised model checking (in Section 4).

[†] For details, see `http://www.combest.eu.com` and `http://www.speeds.eu.com`.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

As a result, we offer a self-contained and full specification theory of label-structured modal transition systems. The theory specialises to some well-known formalisms that have already been extensively studied, but at the same time also provides novel results for instances such as weighted and multi-weighted modal transition systems.

2. Label-structured modal transition systems

We shall now introduce the notion of label-structured modal transition systems and some basic properties of the formalism. However, before we do that, we need to define the notion of labels and label-sets used during the system design and specification refinement.

Definition 2.1 (label-set). A *label-set* is a partially ordered set of labels $(K, \sqsubseteq)$ such that $\bot \in K$ (modelling inconsistency) is the least element of K.

A label $k \in K \setminus \{\bot\}$ is called an *implementation label* if $k' \sqsubseteq k$ implies $k' = k$ for all $k' \in K \setminus \{\bot\}$. In other words, implementation labels are all elements in K just above $\bot$. The set of all implementation labels of $(K, \sqsubseteq)$ is denoted by $\mathrm{Imp}(K, \sqsubseteq)$. With each label $k \in K$, we associate the set $[\![k]\!]$ of all implementation labels below k by

$$[\![k]\!] = \{k' \in \mathrm{Imp}(K, \sqsubseteq) \mid k' \sqsubseteq k\}.$$

Definition 2.2 (well-formed label-set). A label-set $(K, \sqsubseteq)$ with least element $\bot \in K$ is said to be *well formed* if $[\![k]\!] \neq \varnothing$ for every $k \in K \setminus \{\bot\}$.

The well formedness of label-sets ensures consistency of the label refinement relation $\sqsubseteq$, in other words, it should always be possible to refine any label into an implementation label. We can now define label-structured modal transition systems that combine the underlying may/must transition relation known from modal transition systems with the label structure defined above.

Definition 2.3 (label-structured modal transition system). A *label-structured modal transition system* (LSMTS) is a tuple $(S, s_0, (K, \sqsubseteq), \dashrightarrow, \longrightarrow)$ where:

— S is a set of states with initial state $s_0 \in S$;
— $(K, \sqsubseteq)$ is a well-formed label-set;
— $\dashrightarrow \subseteq S \times K \times S$ is the *may* transition relation; and
— $\longrightarrow \subseteq S \times K \times S$ is the *must* transition relation such that $\longrightarrow \subseteq \dashrightarrow$.

We write $s \overset{k}{\dashrightarrow} s'$ if $(s, k, s') \in \dashrightarrow$. If for some $k \in K$ no state $s' \in S$ exists such that $s \overset{k}{\dashrightarrow} s'$, we write $s \not\overset{k}{\dashrightarrow}$, and if there exists some $s' \in S$ such that $s \overset{k}{\dashrightarrow} s'$, we write $s \overset{k}{\dashrightarrow}$. This notation also applies to $\longrightarrow$. By abuse of notation, we use S to denote an LSMTS $(S, s_0, (K_S, \sqsubseteq_S), \dashrightarrow_S, \longrightarrow_S)$, and the subscripts will be omitted if they are clear from the context. The notation (s, S) denotes the LSMTS S with the initial state s_0 replaced by s. The class of all LSMTSs with the well-formed label-set $(K, \sqsubseteq)$ is denoted by $\mathscr{M}_{(K,\sqsubseteq)}$, and we typically use capital letters S, T, U to range over this class.

An LSMTS S is said to be an *implementation* if $\longrightarrow = \dashrightarrow$ and all labels on the transitions are implementation labels, that is, for all $s \overset{k}{\longrightarrow} s'$ in S, we have $k \in \mathrm{Imp}(K, \sqsubseteq)$.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

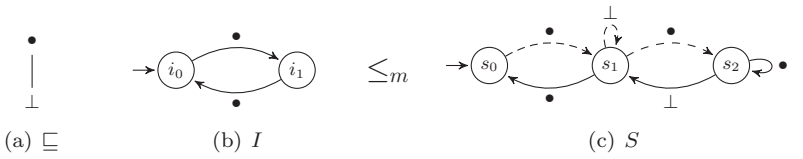


Fig. 1. Unlabelled modal transition system over the label-set $K_{unlabelled}$.

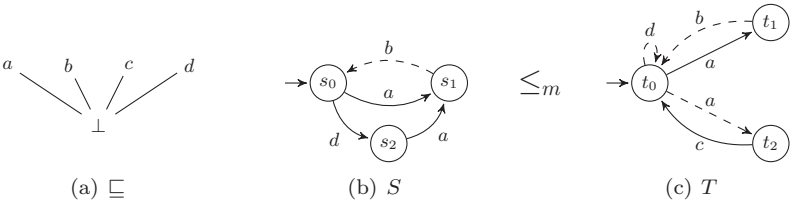


Fig. 2. Modal transition systems over the label-set K_{action}.

The class of all implementations with well-formed label-set $(K, \sqsubseteq)$ is denoted by $\mathscr{I}_{(K,\sqsubseteq)}$, and we typically use capital letters I and J to range over this class.

In the following, we will often use graphs to represent LSMTSs, with the convention that whenever two states are connected by both a must and a may transition under the same label, we will only draw the must transition.

Example 1. The most trivial instance of an LSMTS is obtained by choosing the well-formed label-set $K_{unlabelled} = (\{\bot, \bullet\}, \sqsubseteq)$ where $\sqsubseteq = \{(\bot, \bot), (\bot, \bullet), (\bullet, \bullet)\}$ illustrated in Figure 1(a). This label-set gives rise to unlabelled modal transition systems where $\bullet$ models a single implementation label and $\bot$ is the inconsistency label. An example is shown in Figure 1(b) and (c). The LSMTS I in Figure 1(b) is an implementation because every label is an implementation label and the may and must transition relations coincide. Note that the LSMTS in Figure 1(c) is not an implementation as:

(i) there are transitions labelled with $\bot$; and
(ii) there are several may transitions without the corresponding must ones.

The definition and explanation of modal refinement, denoted $\leqslant_m$, is left to Section 2.2.

Example 2. A well-known instance of the framework is obtained by considering a finite set of actions Σ and then defining a well-formed label-set K_{action} by $K_{action} = (\Sigma \cup \{\bot\}, \sqsubseteq)$ where $a \sqsubseteq b$ if and only if $a = \bot$ or $a = b$. Here all labels (apart from $\bot$) are implementation labels and this setting corresponds exactly to the class of modal transition systems (Larsen and Thomsen 1988b). An illustration of the label-set K_{action} and two examples of modal transition systems are given in Figure 2.

Example 3. As another example of a well-formed label-set demonstrating a more interesting label refinement, we can consider the following structure

$$K_{machine} = (\{drink, coffee, tea, coin, 1EURO, 2EURO, \bot\}, \sqsubseteq)$$

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

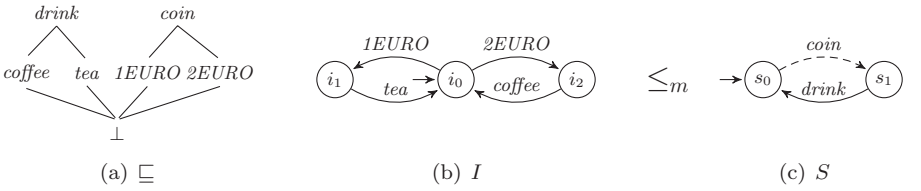


Fig. 3. Vending machines over the label-set $K_{machine}$.

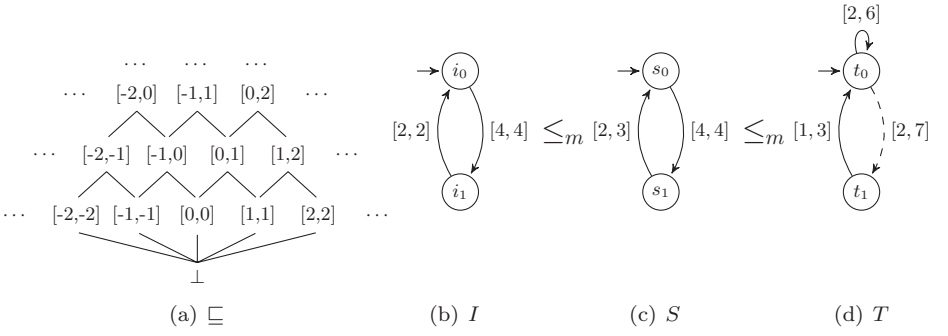


Fig. 4. Weighted modal automata over $K_{weighted}$.

where the ordering $\sqsubseteq$ is given in Figure 3(a). Here it is possible to provide a high-level specification of a vending machine by using the labels *drink* and *coin* that can be later refined in a concrete implementation into the implementation labels *coffee* and *tea*, and *1EURO* and *2EURO*, respectively.

Example 4. Another instance of the framework goes under the name of weighted modal automata. Here the well-formed label-set $K_{weighted}$ is given as a set of integer intervals with the natural inclusion ordering. Formally, $K_{weighted} = (K, \sqsubseteq)$ where

$$K = \{[a, b] \mid a, b \in \mathbb{Z} \text{ s.t. } a \leqslant b\} \cup \{\bot\}$$

and

$$\begin{aligned} [a', b'] &\sqsubseteq [a, b] \text{ if } a \leqslant a' \text{ and } b' \leqslant b \\ \bot &\sqsubseteq [a, b] \\ \bot &\sqsubseteq \bot \end{aligned}$$

for all $a, a', b, b' \in \mathbb{Z}$. It follows that implementation labels are singleton sets of the form $[a, a]$ where $a \in \mathbb{Z}$. See Figure 4 for an illustration of $K_{weighted}$ and three examples of weighted modal automata. The automaton I is an implementation, but S and T are not.

2.1. *Product of labels*

In this section we will discuss a product construction on labels that will allow us to form (by a general construction) new instances of the framework from existing ones.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

Definition 2.4 (product). Let $(K_1, \sqsubseteq_1)$ and $(K_2, \sqsubseteq_2)$ be two label-sets with least elements $\bot_1$ and $\bot_2$, respectively. The *product* $(K_1, \sqsubseteq_1) \otimes (K_2, \sqsubseteq_2)$ of the two label-sets is a label-set $(K, \sqsubseteq)$ where $K = \big((K_1 \setminus \{\bot_1\}) \times (K_2 \setminus \{\bot_2\})\big) \cup \{\bot\}$ and $(k_1', k_2') \sqsubseteq (k_1, k_2)$ if $k_1' \sqsubseteq_1 k_1$ and $k_2' \sqsubseteq_2 k_2$ for all $k_1, k_1' \in K_1 \setminus \{\bot_1\}$ and all $k_2, k_2' \in K_2 \setminus \{\bot_2\}$, and $\bot \sqsubseteq \ell$ for all $\ell \in K$.

It is easy to see that the product construction preserves well formedness and implementations are derived component-wise as stated in the following lemma.

Lemma 2.5. Let $(K_1, \sqsubseteq_1)$ and $(K_2, \sqsubseteq_2)$ be well-formed label-sets. Then:

(1) $(K_1, \sqsubseteq_1) \otimes (K_2, \sqsubseteq_2)$ is a well-formed label-set; and
(2) $\mathrm{Imp}((K_1, \sqsubseteq_1) \otimes (K_2, \sqsubseteq_2)) = \mathrm{Imp}(K_1, \sqsubseteq_1) \times \mathrm{Imp}(K_2, \sqsubseteq_2)$.

Using the product construction of label-sets, we can, for example, combine the previously introduced well-formed label-sets K_{action} and $K_{weighted}$ from Examples 2 and 4 into weighted modal transition systems using the label-set $K_{action} \otimes K_{weighted}$, or into multi-weighted modal transition systems using the label-set

$$K_{action} \otimes K_{weighted} \otimes K_{weighted} \otimes \ldots \otimes K_{weighted},$$

and further combine these with other quantitative aspects.

2.2. *Refinement*

We shall now define the notion of modal refinement that combines the label refinement, given by the partial ordering on the label-set, with the allowed transitions that may be present and the required transitions that must be present. It is a generalisation of the original notion of modal refinement over classical modal transition systems (Larsen and Thomsen 1988b).

Definition 2.6 (modal refinement). Let $S, T \in \mathscr{M}_{(K,\sqsubseteq)}$ be two LSMTSs with initial states s_0 and t_0, respectively. We say that S *modally refines* T, written $S \leqslant_m T$, if there exists a relation $R \subseteq S \times T$ with $(s_0, t_0) \in R$ such that for every $(s, t) \in R$:

(1) whenever $s \overset{k}{\dashrightarrow} s'$, there is $t \overset{\ell}{\dashrightarrow} t'$ such that $k \sqsubseteq \ell$ and $(s', t') \in R$; and
(2) whenever $t \overset{\ell}{\longrightarrow} t'$, there is $s \overset{k}{\longrightarrow} s'$ such that $k \sqsubseteq \ell$ and $(s', t') \in R$.

The *implementation semantics* of an LSMTS $S \in \mathscr{M}_{(K,\sqsubseteq)}$ is defined as the class $[\![S]\!]$ of all implementations refining S, that is, $[\![S]\!] = \{I \in \mathscr{I}_{(K,\sqsubseteq)} \mid I \leqslant_m S\}$.

Example 5. The refinement of modal transition systems labelled with actions (see Example 2) is illustrated in Figure 2. The system S is a modal refinement of the system T, and the relation demonstrating this is given by $\{(s_0, t_0), (s_1, t_1), (s_2, t_0)\}$. Note that S is not an implementation yet as it contains a may transition under b without a must transition under the same label.

Example 6. Consider the label-set $K_{machine}$ from Example 3. A specification of a vending machine is shown in Figure 3(c). It allows a coin to enter and, if this happens, it then requires that a drink is returned to the customer. One of the possible implementations (where

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

all labels are implementation labels and the may and must transition relations coincide) of this specification is given in Figure 3(b). The modal refinement between the implementation and specification is easily demonstrated by the relation $\{(i_0, s_0), (i_1, s_1), (i_2, s_1)\}$.

Example 7. The refinement of weighted modal automata (see Example 4) is illustrated in Figure 4. The relation $\{(s_0, t_0), (s_1, t_0)\}$ is witnessing the modal refinement between Figure 4(d) and 4(c). Note that the refined specification in Figure 4(c) is not an implementation yet as it contains the label [2, 3], which is not an implementation label. We can thus refine it further, ending up with an implementation as shown in Figure 4(b). The witnessing relation is $\{(i_0, s_0), (i_1, s_1)\}$.

Lemma 2.7. The modal refinement relation $\leqslant_m$ is a preorder.

Proof. Reflexivity is trivial. To show transitivity, we let $S, T, U \in \mathscr{M}_{(K,\sqsubseteq)}$ be three LSMTSs with their initial states s_0, t_0, u_0 such that $S \leqslant_m T \leqslant_m U$. From the assumption $S \leqslant_m T$, we know that there exists a witnessing relation $R_1 \subseteq S \times T$, and from the assumption $T \leqslant_m U$ we know that there exists a witnessing relation $R_2 \subseteq T \times U$. We define a relation $R \subseteq S \times U$ by the relational composition of R_1 and R_2, that is,

$$R = \{(s, u) \mid \exists t \in T : (s, t) \in R_1 \text{ and } (t, u) \in R_2\}.$$

We will now show that R proves $S \leqslant_m U$. It is obvious that $(s_0, u_0) \in R$. We let $(s, u) \in R$ be an arbitrary element of R. Let $t \in T$ be a state such that $(s, t) \in R_1$ and $(t, u) \in R_2$. Then:

(1) Assume $s \overset{k_s}{\dashrightarrow} s'$.
From $(s, t) \in R_1$, there exists $t \overset{k_t}{\dashrightarrow} t'$ such that $k_s \sqsubseteq k_t$ and $(s', t') \in R_1$. Then, as $(t, u) \in R_2$, we get $u \overset{k_u}{\dashrightarrow} u'$ such that $k_t \sqsubseteq k_u$ and $(t', u') \in R_2$, hence $(s', u') \in R$ and by transitivity of $\sqsubseteq$ also $k_s \sqsubseteq k_u$.
(2) The other direction is symmetric. □

Modal refinement induces an equivalence relation on LSMTS. We say that S and T are *equivalent*, denoted by $S \equiv_m T$, if both $S \leqslant_m T$ and $T \leqslant_m S$ are satisfied.

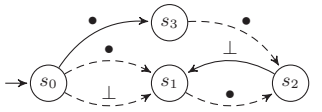
Lemma 2.8. Let $I, J \in \mathscr{I}_{(K,\sqsubseteq)}$ be two implementations. Then $I \leqslant_m J$ implies $I \equiv_m J$.

Proof. Given a relation R witnessing $I \leqslant_m J$, we can use $R^{-1} = \{(j, i) \mid (i, j) \in R\}$ to prove that $J \leqslant_m I$. Let $(j, i) \in R^{-1}$.

(1) Assume that $i \xrightarrow{k} i'$. Remember that this means we also know $i \overset{k}{\dashrightarrow} i'$. From the fact that $(i, j) \in R$, there exists $j \overset{\ell}{\dashrightarrow} j'$ such that $k \sqsubseteq \ell$ and $(i', j') \in R$. Since J is an implementation, we get that $k = \ell$ and $j \xrightarrow{\ell} j'$. It is obvious that $(j', i') \in R^{-1}$.
(2) The other direction is symmetric. □

The notion of modal refinement can be understood as refinement defined at the syntactic level since it directly relates the states of two specifications. A semantically motivated notion of refinement, usually called *thorough refinement*, says that S is a refinement of T if every implementation of S is also an implementation of T.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press



(a) A consistent, but locally inconsistent LSMTS S

(b) The locally consistent result $\rho(S)$ of pruning S

Fig. 5. Example of the pruning operator.

Definition 2.9 (thorough refinement). Let $S, T \in \mathscr{M}_{(K,\sqsubseteq)}$ be two LSMTSs. We say that S *thoroughly refines* T, written $S \leqslant_t T$, if $[\![S]\!] \subseteq [\![T]\!]$.

We expect that modal refinement should imply thorough refinement, as stated in the following soundness theorem. The opposite implication does not hold in general – the details are discussed in Section 2.4.

Theorem 2.10 (soundness). Let $S, T \in \mathscr{M}_{(K,\sqsubseteq)}$ be two LSMTSs. Then $S \leqslant_m T$ implies $S \leqslant_t T$.

Proof. The statement follows immediately from the transitivity of modal refinement (see Lemma 2.7). □

2.3. *Consistency and pruning*

In a similar way to the classical notion of consistency, an LSMTS S is consistent if it has at least one implementation.

Definition 2.11 (consistency). Let $S \in \mathscr{M}_{(K,\sqsubseteq)}$. We say the LSMTS S is *consistent* if $[\![S]\!] \neq \varnothing$.

Consistency is a semantical notion and in the rest of this paper it will be useful to introduce a syntactical notion of consistency, which is called local consistency.

Definition 2.12 (local consistency). Let $S \in \mathscr{M}_{(K,\sqsubseteq)}$. A state $s \in S$ is *locally consistent* if $s \not\xrightarrow{\perp}$. The LSMTS S is *locally consistent* if all states of S are locally consistent.

From our assumption of the well formedness of label-sets, it follows that any locally consistent S has at least one implementation, thus local consistency implies consistency. The converse is not true, as explained in the following example.

Example 8. Consider the LSMTS S in Figure 5(a) with the label-set $K_{unlabelled}$ from Example 1. The system S is clearly not locally consistent, but it is consistent as an implementation with just two states connected by a must (and may) transition labelled with $\bullet$ is an implementation of S.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

We shall now define a pruning operator that removes locally inconsistent states.

Definition 2.13 (pruning). Let $S \in \mathscr{M}_{(K,\sqsubseteq)}$ be an LSMTS and $B \subseteq S$ be a subset of its states. Let

$$pre(B) = \{s \in S \mid s \xrightarrow{k} s' \text{ and } s' \in B \text{ for some } k \in K\}$$

and

$$\begin{aligned} pre^0(B) &= B \\ pre^{j+1}(B) &= pre(pre^j(B)) \text{ for } j \geqslant 0 \\ pre^*(B) &= \bigcup_{j \geqslant 0} pre^j(B). \end{aligned}$$

The *pruning* $\rho(S)$ of S is defined if $s_0 \notin pre^*(\mathsf{bad})$ where $\mathsf{bad} = \{s \in S \mid s \xrightarrow{\perp}\}$, and, in this case, $\rho(S)$ is the LSMTS $(S^\rho, s_0, (K, \sqsubseteq), \dashrightarrow_\rho, \longrightarrow_\rho)$ where

$$\begin{aligned} S^\rho &= S \setminus pre^*(\mathsf{bad}) \\ \dashrightarrow_\rho &= \dashrightarrow \cap \left(S^\rho \times (K \setminus \{\perp\}) \times S^\rho\right) \\ \longrightarrow_\rho &= \longrightarrow \cap \left(S^\rho \times (K \setminus \{\perp\}) \times S^\rho\right). \end{aligned}$$

It is clear that for an LSMTS with n states, we can compute $pre^*(\mathsf{bad})$ using finitely many iterations: more precisely $pre^*(\mathsf{bad}) = pre^n(\mathsf{bad})$ in this case. Also note that the well definedness of pruning is equivalent to the absence of a path of must transitions from the initial state to a locally inconsistent state (which enforces inconsistency through a must transition labelled with $\perp$).

Figure 5 shows the application of the pruning operator ρ to the system S, where it is easy to see that $\rho(S) \leqslant_m S$. Also, pruning does not remove any implementation. A summary of the properties of pruning is given in the following proposition.

Proposition 2.14. Let $S \in \mathscr{M}_{(K,\sqsubseteq)}$. If $\rho(S)$ is defined, then

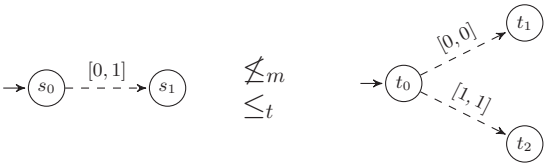
(1) $\rho(S)$ is locally consistent.
(2) $\rho(S) \leqslant_m S$.
(3) $[\![\rho(S)]\!] = [\![S]\!]$.
(4) For any locally consistent $T \in \mathscr{M}_{(K,\sqsubseteq)}$, if $T \leqslant_m S$, then $T \leqslant_m \rho(S)$.

Moreover, $\rho(S)$ is defined if and only if S is consistent.

Proof.

(1)As all labels $\perp$ were removed in $\rho(S)$, it is trivially locally consistent.
(2)We will show that the relation $R = \{(s,s) \mid s \in S \setminus pre^*(\mathsf{bad})\}$ is a refinement relation in order to argue that $\rho(S) \leqslant_m S$. Let $(s,s) \in R$. If $s \xrightarrow{k}_\rho s'$ in $\rho(S)$, then, by the construction, this implies that $s' \in S \setminus pre^*(\mathsf{bad})$. Clearly, we also have $s \xdashrightarrow{k} s'$ in S and $(s',s') \in R$. On the other hand, if $s \xrightarrow{k} s'$ in S, then $s' \notin pre^*(\mathsf{bad})$ as $s \notin pre^*(\mathsf{bad})$, which means that $s \xrightarrow{k}_\rho s'$ also in $\rho(S)$ and $(s',s') \in R$.
(3)The inclusion $[\![\rho(S)]\!] \subseteq [\![S]\!]$ follows from the fact that $\rho(S) \leqslant_m S$. Let $I \in [\![S]\!]$. This means that there is a refinement relation R demonstrating that $I \leqslant_m S$. We will argue

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press



(a) S (b) T

Fig. 6. Incompleteness of modal refinement demonstrated by systems S and T.

that R is also a refinement relation demonstrating $I \leqslant_m \rho(S)$. However, this follows easily from the observation that R cannot contain any state from $pre^*(\mathsf{bad})$, because otherwise a must path to the label $\perp$ will be enforced in I too, but then I is not an implementation.

(4) The same argument as in the previous point also applies here. Any refinement relation demonstrating $T \leqslant_m S$ can also be used to establish $T \leqslant_m \rho(S)$. In order to apply the reasoning as above, it is important that T does not contain any transitions with the label $\perp$.

For the last claim, observe that if $\rho(S)$ is not defined, then there is a must path from its initial state to a state requiring a transition under $\perp$. Any implementation then has to contain such a path to a state requiring a transition under $\perp$, but then it is not an implementation. On the other hand, if $\rho(S)$ is defined, we can change every transition in $\rho(S)$ to a must transition and replace every label by some implementation label below it (possibly using the well formedness of the label-set) in order to construct an implementation of $\rho(S)$, which is also an implementation of S as $\rho(S) \leqslant_m S$. □

2.4. *Determinism and the completeness of refinement*

In general, thorough refinement does not imply modal refinement. A counterexample, using the label-set $K_{weighted}$ (see Example 7), is given in Figure 6. Clearly, the transition $s_0 \overset{[0,1]}{\dashrightarrow} s_1$ cannot be matched by either of the two transitions from t_0 as their labels are less general than $[0, 1]$. Hence $S \not\leqslant_m T$. On the other hand, any implementation of S is either empty or it is a tree of height one with the outgoing edges labelled by either $[0, 0]$ or $[1, 1]$. All such implementations are also refinements of the system T.

It is known that for classical modal transition systems, thorough refinement implies the modal one, under the assumption of determinism (Beneš *et al.* 2009b). We can generalise this result to the class of label-structured modal transition systems. Before we define when an LSMTS is deterministic, we first define when two labels k_1, k_2 are unifiable, that is, if there is another label k that overlaps with k_1 and k_2 with respect to their sets of implementation labels.

Definition 2.15 (unifiable labels). Two labels $k_1, k_2 \in K$ are said to be *unifiable* if there exists $k \in K$ such that $[\![k]\!] \cap [\![k_1]\!] \neq \varnothing$ and $[\![k]\!] \cap [\![k_2]\!] \neq \varnothing$.

Hence, determinism expresses the fact that for any two outgoing may transitions from the same state under two different labels k_1 and k_2, the labels k_1 and k_2 are not unifiable.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

Definition 2.16 (determinism). A LSMTS S is said to be *deterministic* if for any state $s \in S$ and any two transitions $s \overset{k_1}{\dashrightarrow} s'_1$ and $s \overset{k_2}{\dashrightarrow} s'_2$ if k_1 and k_2 are unifiable, then $k_1 = k_2$ and $s'_1 = s'_2$.

Returning to Figure 6, we can realise that the system T is not deterministic as there is a branching of the transitions with labels $[0,0]$ and $[1,1]$, while there exists a label $[0,1]$ such that $[\![[0,1]]\!] \cap [\![[0,0]]\!] \neq \varnothing$ and $[\![[0,1]]\!] \cap [\![[1,1]]\!] \neq \varnothing$.

A very natural assumption that has to be imposed on the label-sets later on in order to show the completeness of modal refinement is completeness of label refinement: inclusion of implementation labels implies label refinement.

Definition 2.17 (completeness of label refinement). Let $(K, \sqsubseteq)$ be a label-set. Label refinement $\sqsubseteq$ is *complete* if for all $k, \ell \in K$, we have $[\![k]\!] \subseteq [\![\ell]\!]$ implies $k \sqsubseteq \ell$.

All examples of label-sets provided in this paper satisfy this property. Note that label refinement is always sound by definition, that is, $k \sqsubseteq \ell$ implies $[\![k]\!] \subseteq [\![\ell]\!]$ by the transitivity of label refinement.

Under the assumption of:

(1) completeness of label refinement,

(2) determinism of the refined LSMTS, and

(3) local consistency of the refining LSMTS,

thorough refinement implies modal refinement.

Theorem 2.18 (completeness). Let $(K, \sqsubseteq)$ be a well-formed label-set for which label refinement $\sqsubseteq$ is complete. Let $S, T \in \mathscr{M}_{(K,\sqsubseteq)}$ with initial states s_0 and t_0, respectively, such that S is locally consistent and T is deterministic. Then $S \leqslant_t T$ implies $S \leqslant_m T$.

Proof. Assume that $S \leqslant_t T$. We define a relation $R \subseteq S \times T$ as the smallest relation satisfying:

(1)$(s_0, t_0) \in R$.

(2)if $(s,t) \in R$, $s \overset{k}{\dashrightarrow} s'$, $t \overset{\ell}{\dashrightarrow} t'$ and $[\![k]\!] \cap [\![\ell]\!] \neq \varnothing$, then $(s', t') \in R$.

We begin by showing a technical result (which we will use later) saying that any $(s,t) \in R$ satisfies $[\![(s,S)]\!] \subseteq [\![(t,T)]\!]$. For $(s_0, t_0) \in R$, we have $[\![(s_0,S)]\!] = [\![S]\!] \subseteq [\![T]\!] = [\![(t_0,T)]\!]$ from the assumption $S \leqslant_t T$. Now let $(s,t) \in R$ be such that $[\![(s,S)]\!] \subseteq [\![(t,T)]\!]$ and assume that $s \overset{k}{\dashrightarrow} s'$, $t \overset{\ell}{\dashrightarrow} t'$ and $[\![k]\!] \cap [\![\ell]\!] \neq \varnothing$. Let $I' \in [\![(s',S)]\!]$ and $m \in [\![k]\!] \cap [\![\ell]\!]$, which exists by the construction. Then, since S is locally consistent, there exists an implementation $(i_0, I) \in [\![(s,S)]\!]$ such that $i_0 \overset{m}{\longrightarrow} i'$ and $(i', I) \leqslant_m I'$. From $[\![(s,S)]\!] \subseteq [\![(t,T)]\!]$, it follows that $I \in [\![(t,T)]\!]$. Then there exists a transition $t \overset{\ell'}{\dashrightarrow} t''$ such that $(i', I) \in [\![(t'',T)]\!]$ and $m \in [\![\ell']\!]$. Now, we have $t \overset{\ell}{\dashrightarrow} t'$ and $t \overset{\ell'}{\dashrightarrow} t''$ such that $m \in [\![\ell]\!] \cap [\![\ell']\!]$, so ℓ and ℓ' are unifiable. As T is deterministic, it follows that $\ell = \ell'$ and $t' = t''$, so $(i', I) \in [\![(t',T)]\!]$. Finally, from $(i', I) \leqslant_m I'$ and Lemma 2.8, it follows that $(i', I) \equiv_m I'$ and hence $I' \in [\![(t',T)]\!]$.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

We will now show that R is a relation witnessing $S \leqslant_m T$. It is clear that $(s_0, t_0) \in R$. Let $(s,t) \in R$. Then:

(1) Assume $s \overset{k}{\dashrightarrow} s'$.

By local consistency of S we can assume that $k \neq \bot$. Then, for each implementation label $m \in [\![k]\!]$, there exists an implementation $I_m \in [\![(s,S)]\!]$ such that $i_0 \overset{m}{\longrightarrow} i'$. We also know that $I_m \in [\![(t,T)]\!]$ because $[\![(s,S)]\!] \subseteq [\![(t,T)]\!]$. Hence there exists a transition $t \overset{\ell_m}{\dashrightarrow} t'_m$ such that $m \in [\![\ell_m]\!]$. We have to show that for all $m \in [\![k]\!]$, the labels ℓ_m are the same. Suppose there are $m_1, m_2 \in [\![k]\!]$ and transitions

$$t \overset{\ell_{m_1}}{\dashrightarrow} t'_{m_1}$$

and

$$t \overset{\ell_{m_2}}{\dashrightarrow} t'_{m_2}$$

such that $m_1 \in [\![\ell_{m_1}]\!]$ and $m_2 \in [\![\ell_{m_2}]\!]$. Then, since $m_1 \in [\![\ell_{m_1}]\!] \cap [\![k]\!]$ and $m_2 \in [\![\ell_{m_2}]\!] \cap [\![k]\!]$ and T is deterministic, it follows that $\ell_{m_1} = \ell_{m_2}$ and $t'_{m_1} = t'_{m_2}$. It then follows that there is a unique transition $t \overset{\ell}{\dashrightarrow} t'$ such that $m \in [\![\ell]\!]$ for all implementation labels $m \in [\![k]\!]$, which means $[\![k]\!] \subseteq [\![\ell]\!]$, which implies $k \sqsubseteq \ell$ by completeness of label refinement. Moreover, by the definition of R, we have $(s', t') \in R$.

(2) Assume $t \overset{\ell}{\longrightarrow} t'$.

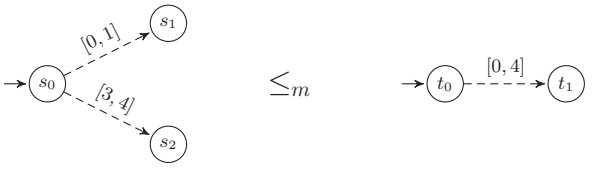
If $\ell = \bot$, there is a transition $s \overset{\bot}{\longrightarrow} s'$ in S, contradicting local consistency of S; hence $\ell \neq \bot$. Then, for each implementation $I \in [\![(t,T)]\!]$, there exists a transition $i_0 \overset{m}{\longrightarrow} i'$ for some label $m \in [\![\ell]\!]$. We know that $[\![(s,S)]\!] \subseteq [\![(t,T)]\!]$, so every implementation $(j_0, J) \in [\![(s,S)]\!]$ has a transition $j_0 \overset{m}{\longrightarrow} j'$. It follows that S must have a transition $s \overset{k}{\longrightarrow} s'$ such that $m \in [\![k]\!]$. If $[\![k]\!] \not\subseteq [\![\ell]\!]$, there would exist an implementation $(\bar{j}_0, \bar{J}) \in [\![(s,S)]\!]$ having $\bar{j}_0 \overset{n}{\longrightarrow} \bar{j}'$ with an implementation label $n \in [\![k]\!]$ not belonging to $[\![\ell]\!]$, which means that there must exist another transition in T, say $t \overset{\ell'}{\dashrightarrow} t''$, such that $n \in [\![\ell']\!]$. But then we have $m \in [\![\ell]\!] \cap [\![k]\!]$ and $n \in [\![\ell']\!] \cap [\![k]\!]$, which contradicts the determinism of T. Thus, we would have $[\![k]\!] \subseteq [\![\ell]\!]$, which implies $k \sqsubseteq \ell$ by the completeness of label refinement. Moreover, by the definition of R, we have $(s', t') \in R$. □

2.5. *Determinisation*

It is a well-known fact that, for almost all existing specification theories, deciding thorough refinement involves more complex decision procedures than those that can be used to decide modal refinement (see, for example, Beneš *et al.* (2009a) dealing with the classical modal transition systems). In Section 2.4, we saw that in the case of deterministic systems, modal and thorough refinements coincide and can be decided by efficient syntactical fixed-point based algorithms. Moreover, it will become evident in Section 3.1 that determinism plays an important role in establishing several soundness results.

It is thus worth studying a general procedure that, given a non-deterministic LSMTS S, computes its smallest deterministic over-approximation $\mathscr{D}(S)$, which is called the deterministic hull.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press



(a) S (b) $\mathcal{D}(S)$

Fig. 7. Determinisation.

Example 9. Consider the system in Figure 7(a) where S is a LSMTS with label-set $K_{weighted}$. It is non-deterministic since there is the label $[0,4]$ for which

$$[\![[0,4]]\!] \cap [\![[0,1]]\!] \neq \varnothing$$

and

$$[\![[0,4]]\!] \cap [\![[3,4]]\!] \neq \varnothing.$$

The best we can do is to approximate S by a deterministic LSMTS in Figure 7(b). However, $\mathscr{D}(S)$ is an over-approximation since $[\![S]\!] \subsetneq [\![\mathscr{D}(S)]\!]$, which is witnessed by the implementation with a single must transition labelled with $[2,2]$.

In the following, we will propose an algorithm that computes, for a given LSMTS S, a deterministic LSMTS $\mathscr{D}(S)$ such that $\mathscr{D}(S)$ is an over-approximation of S with the property that it is a minimal one with respect to modal refinement. The construction is a generalisation of the algorithm presented in Beneš *et al.* (2009b) and dealing specifically with modal transition systems.

We impose the following assumption on the label-set $(K, \sqsubseteq)$, which we will need if we are going to apply the determinisation algorithm. For any set $L \subseteq K$ of pairwise unifiable labels, we require the existence of the least upper bound $lub(L) \in K$ such that $\ell \sqsubseteq lub(L)$ for all $\ell \in L$, and whenever $\ell \sqsubseteq \ell'$ for all $\ell \in L$, we have $lub(L) \sqsubseteq \ell'$.

Definition 2.19 (deterministic hull). Let

$$S = (S, s_0, (K, \sqsubseteq), \dashrightarrow, \longrightarrow)$$

be an LSMTS. The *deterministic hull of* S is defined by the LSMTS

$$\mathscr{D}(S) = (\mathscr{P}(S) \setminus \{\varnothing\}, \{s_0\}, (K, \sqsubseteq), \dashrightarrow_{\mathrm{D}}, \longrightarrow_{\mathrm{D}})$$

where the transition relations $\dashrightarrow_{\mathrm{D}}$ and $\longrightarrow_{\mathrm{D}}$ are defined as follows. Let $\mathscr{T} \in (\mathscr{P}(S) \setminus \{\varnothing\})$ be a state in $\mathscr{D}(S)$. For every maximal, non-empty set $L \subseteq \{k \mid s \overset{k}{\dashrightarrow},\ s \in \mathscr{T}\}$ of pairwise unifiable labels, we have $\mathscr{T} \overset{\ell}{\dashrightarrow}_{\mathrm{D}} \mathscr{T}_\ell$ where $\ell = lub(L)$ and

$$\mathscr{T}_\ell = \{s' \in S \mid s \overset{k}{\dashrightarrow} s',\ s \in \mathscr{T},\ k \in L\}.$$

If, moreover, for each $s \in \mathscr{T}$, we have $s \overset{k}{\longrightarrow} s'$ for some $s' \in \mathscr{T}_\ell$ and some $k \in K$ such that $k \sqsubseteq \ell$, then $\mathscr{T} \overset{\ell}{\longrightarrow}_{\mathrm{D}} \mathscr{T}_\ell$.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

We will now show that $\mathscr{D}(S)$ is the smallest deterministic over-approximation of S.

Theorem 2.20 (soundness and minimality of determinisation). Let $S \in \mathscr{M}_{(K,\sqsubseteq)}$. Then the following hold:

(1) $\mathscr{D}(S)$ is deterministic.
(2) $S \leqslant_m \mathscr{D}(S)$.
(3) For every deterministic $D \in \mathscr{M}_{(K,\sqsubseteq)}$, if $S \leqslant_m D$, then $\mathscr{D}(S) \leqslant_m D$.

Proof.

(1) Let $\mathscr{T}$ be a state in $\mathscr{D}(S)$ and assume that there exist two different transitions $\mathscr{T} \xrightarrow{\ell_1} \mathscr{T}_{\ell_1}$ and $\mathscr{T} \xrightarrow{\ell_2} \mathscr{T}_{\ell_2}$ such that ℓ_1 and ℓ_2 are unifiable. It follows that there exists a least upper bound $\ell \in K$ of ℓ_1 and ℓ_2. Hence, all labels below ℓ_1 are unifiable with all labels below ℓ_2. This contradicts the fact that ℓ_1 and ℓ_2 are the least upper bounds of maximal, unifiable sets of labels.

(2) We define a relation $R \subseteq S \times \mathscr{D}(S)$ by $R = \{(s, \mathscr{T}) \mid s \in \mathscr{T}\}$. Clearly, $(s_0, \{s_0\}) \in R$ for the respective initial states. Now let $(s, \mathscr{T}) \in R$.

(a) We first assume $s \xrightarrow{k} s'$.

So, since $s \in \mathscr{T}$, there exists a maximal, non-empty set $L \subseteq \{k \mid s \xrightarrow{k}, \ s \in \mathscr{T}\}$ of pairwise unifiable labels such that $k \in L$. So there exists a transition $\mathscr{T} \xrightarrow{\ell} \mathscr{T}_\ell$ where $\ell = lub(L)$, which, in particular, means that $k \sqsubseteq \ell$. Moreover, by the definition of $\mathscr{T}_\ell$, we have $s' \in \mathscr{T}_\ell$, and thus $(s', \mathscr{T}_\ell) \in R$.

(b) We now assume $\mathscr{T} \xrightarrow{\ell} \mathscr{T}_\ell$.

We know that for all $t \in \mathscr{T}$ there is $t \xrightarrow{k} t'$ such that $k \sqsubseteq \ell$ and $t' \in \mathscr{T}_\ell$. Since we know that $s \in \mathscr{T}$, there is $s \xrightarrow{k} s'$ such that $k \sqsubseteq \ell$ and $s' \in \mathscr{T}_\ell$, hence $(s', \mathscr{T}_\ell) \in R$.

(3) Let $D \in \mathscr{M}_{(K,\sqsubseteq)}$ be a deterministic LSMTS and assume that $S \leqslant_m D$ witnessed by the relation R. We want to show that $\mathscr{D}(S) \leqslant_m D$. We define a relation $R' \subseteq \mathscr{D}(S) \times D$ by

$$(\mathscr{T}, d) \in R' \qquad \text{if and only if} \qquad \varnothing \neq \mathscr{T} \subseteq \{s \in S \mid (s, d) \in R\}.$$

We will show that R' is a relation witnessing $\mathscr{D}(S) \leqslant_m D$. Clearly, $(\{s_0\}, d_0) \in R'$ for the corresponding initial states. Let $(\mathscr{T}, d) \in R'$. Then:

(a) We assume $\mathscr{T} \xrightarrow{\ell} \mathscr{T}_\ell$ and $\ell = lub(L)$ for some maximal, non-empty set $L \subseteq \{k \mid s \xrightarrow{k}, \ s \in \mathscr{T}\}$. We want to show that there exists $d \xrightarrow{\ell'} d'$ such that $\ell \sqsubseteq \ell'$ and $(\mathscr{T}_\ell, d') \in R'$. Let

$$\mathscr{S} = \{s \in \mathscr{T} \mid s \xrightarrow{k}, \ k \in L\},$$

which is non-empty since L is non-empty and it is a subset of $\{k \mid s \xrightarrow{k}, \ s \in \mathscr{T}\}$. From the assumption $(\mathscr{T}, d) \in R'$, we know that for all $s \in \mathscr{S}$, we have $(s, d) \in R$ by the definition of R'. Let

$$\Delta = \{s \xrightarrow{k} s' \mid s \in \mathscr{S} \text{ and } k \in L\}.$$

For every transition $s \xrightarrow{k} s'$ in Δ, it follows from $(s, d) \in R$ that there exists $d \xrightarrow{k'} d_{k'}$ in D such that $k \sqsubseteq k'$ and $(s', d_{k'}) \in R$. Let L' denote the set of all such k'. Since L

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

is a set of pairwise unifiable labels and for every $k' \in L'$ there is some $k \in L$ such that $k \sqsubseteq k'$, we know that L' is a set of pairwise unifiable labels, too. For every $k' \in L'$, we have a transition $d \overset{k'}{\dashrightarrow} d_{k'}$ for some $d_{k'} \in D$. From the determinism of D, it follows that there exists $d \overset{\ell'}{\dashrightarrow} d'$ such that $d' = d_{k'}$ and $\ell' = k'$ for all $k' \in L'$. Hence, $k \sqsubseteq \ell'$ for all $k \in L$. Since ℓ is the least upper bound of L, we can conclude that $\ell \sqsubseteq \ell'$. Moreover, $(\mathscr{T}_\ell, d') \in R'$ because $\mathscr{T}_\ell = \{s' \in S \mid s \overset{k}{\dashrightarrow} s',\ s \in \mathscr{T},\ k \in L\}$ is non-empty, and for every $s' \in \mathscr{T}_\ell$, we have $(s', d') \in R$.

(b) We assume $d \overset{\ell'}{\longrightarrow} d'$. By the definition of R' we know that $(s, d) \in R$ for all $s \in \mathscr{T}$. Then it follows that, for all $s \in \mathscr{T}$, we have $s \overset{k}{\longrightarrow} s'$ such that $k \sqsubseteq \ell'$ and $(s', d') \in R$. Note that $\mathscr{T}$ is non-empty, so there is some maximal, non-empty set $L \subseteq \{k \mid s \overset{k}{\dashrightarrow},\ s \in \mathscr{T}\}$ of pairwise unifiable labels such that $k \in L$ for all labels k where $s \overset{k}{\dashrightarrow}$ and $k \sqsubseteq \ell'$. This implies that there exists $\mathscr{T} \overset{\ell}{\longrightarrow} \mathscr{T}_\ell$ with $\ell = lub(L)$. We now have to show that $\ell \sqsubseteq \ell'$. Let $k \in L$. Then there exists $s \overset{k}{\dashrightarrow} s'$ with $s \in \mathscr{T}$, and from $(s, d) \in R$, it follows that $k \sqsubseteq \ell'$ by the determinism of D, and $(s', d') \in R$. Since ℓ is the least upper bound of L, it follows that $\ell \sqsubseteq \ell'$. Finally, we need to show that $(\mathscr{T}_\ell, d') \in R'$. But the set $\mathscr{T}_\ell$ is clearly non-empty, and, moreover, for every $s' \in \mathscr{T}_\ell$, we have $(s', d') \in R$. □

We can use the above theorem to prove that given two LSMTSs S and T, whenever S thoroughly refines T, then $\mathscr{D}(S)$ modally (syntactically) refines $\mathscr{D}(T)$.

Corollary 2.21. Let $(K, \sqsubseteq)$ be a well-formed label-set for which the label refinement $\sqsubseteq$ is complete. Let $S, T \in \mathscr{M}_{(K,\sqsubseteq)}$ be locally consistent LSMTSs. If $S \leqslant_t T$, then $\mathscr{D}(S) \leqslant_m \mathscr{D}(T)$.

Proof. By Theorem 2.20, we know that $T \leqslant_m \mathscr{D}(T)$, and by Theorem 2.10, we also know that $T \leqslant_t \mathscr{D}(T)$. By transitivity and the assumption $S \leqslant_t T$, it follows that $S \leqslant_t \mathscr{D}(T)$, which implies $S \leqslant_m \mathscr{D}(T)$ by the completeness of refinement (Theorem 2.18). By the minimality of $\mathscr{D}(S)$, we can conclude that $\mathscr{D}(S) \leqslant_m \mathscr{D}(T)$. □

3. Specification theory

In order to apply label-structured modal transition systems as a specification formalism for software components, we need to define several operators on LSMTSs that are essential for any specification theory supporting compositional reasoning. First, *structural composition* (or *parallel composition*) allows us to combine interacting specifications. Second, *logical composition* (or *conjunction*) of two or more specifications expresses the greatest specification satisfying all the requirements of the given set of specifications. And third, a *quotient operator*, which is dual to parallel composition, such that given a specification T expressing a requirement that needs to be implemented, and a specification S of the components that already exist, the quotient of T by S is the smallest specification that, together with S, satisfies T. All these operators are an important part of any compositional specification theory.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

3.1. *Operators on labels and their product*

Operators on LSMTSs naturally involve different ways of combining labels. In this section, we introduce *label operators* on well-formed label-sets, which will become crucial ingredients for the operators on LSMTSs introduced later on. In the following, label operators are denoted by symbols in circles.

Definition 3.1 (label operator). A (binary) *label operator* on a label-set $(K, \sqsubseteq)$ is a partial function $\odot : K \times K \rightharpoonup K$.

A label operator $\odot$ on $(K, \sqsubseteq)$ is said to be *commutative* if $k \odot \ell$ is defined if and only if $\ell \odot k$ is defined, and if they are defined, then $k \odot \ell = \ell \odot k$. The operator is *associative* if $k \odot (\ell \odot m)$ is defined if and only if $(k \odot \ell) \odot m$, and if they are defined, then $k \odot (\ell \odot m) = (k \odot \ell) \odot m$.

We can also form products of label operators, which are operators on the product of their label-sets.

Definition 3.2 (product of label operators). Given two well-formed label-sets

$$(K_1, \sqsubseteq_1) \text{ and } (K_2, \sqsubseteq_2)$$

and two label operators $\odot_1$ and $\odot_2$, the product of $\odot_1$ and $\odot_2$ is given by the label operator $\odot_1 \times \odot_2$ on $K_1 \otimes K_2$, which is defined as follows:

$$(k_1, k_2)(\odot_1 \times \odot_2)(\ell_1, \ell_2) = \begin{cases} (k_1 \odot_1 \ell_1, k_2 \odot_2 \ell_2) & \text{if } \bot \neq k_i \odot_i \ell_i \text{ is defined for } i \in \{1,2\} \\ \bot & \text{if } k_i \odot_i \ell_i \text{ is defined for } i \in \{1,2\} \\ & \text{and either } k_1 \odot \ell_1 = \bot \text{ or } k_2 \odot \ell_2 = \bot \\ \text{undefined} & \text{otherwise.} \end{cases}$$

It is easy to see that if two label operators, $\odot_1$ and $\odot_2$, are commutative and associative, then so is the product operator $\odot_1 \times \odot_2$.

3.2. *Parallel composition*

We begin with parallel composition, which is an operator that reflects the standard structural composition of implementations at the specification level.

Two LSMTSs with the same well-formed label-set $(K, \sqsubseteq)$ can be structurally composed with respect to a label operator $\oplus$ on $(K, \sqsubseteq)$. Two transitions (in different parallel components) labelled with $k \in K$ and $\ell \in K$ can synchronise if $k \oplus \ell$ is defined. The synchronised transition is then labelled with the label $k \oplus \ell$.

The desired property of parallel composition, which is crucial for any compositional specification theory, is called compositional refinement. It allows a step-wise refinement of individual specifications while guaranteeing that their parallel composition will refine the parallel composition of the original specifications. However, in order to achieve this, we have to impose a natural requirement on the label operator $\oplus$ for composing labels.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

Definition 3.3 (compositional label operator). A label operator $\oplus$ on a well-formed label-set $(K, \sqsubseteq)$ is *compositional* if whenever $k' \sqsubseteq k$ and $\ell' \sqsubseteq \ell$, we have $k' \oplus \ell'$ is defined if and only if $k \oplus \ell$ is defined, and, moreover, in the positive case, $k' \oplus \ell' \sqsubseteq k \oplus \ell$.

As expected, compositionality of label operators is preserved under their products.

Proposition 3.4. If $\oplus_1$ and $\oplus_2$ are compositional label operators on well-formed label-sets $(K_1, \sqsubseteq_1)$ and $(K_2, \sqsubseteq_2)$, respectively, then $\oplus_1 \times \oplus_2$ is a compositional label operator on $(K_1, \sqsubseteq_1) \otimes (K_2, \sqsubseteq_2)$.

The actual definition of $\oplus$ depends on the interpretation of parallel composition for the modelled quantity. We present several possible definitions of $\oplus$ for some of the examples seen so far. All of the defined operators are compositional label operators, which can be further combined using the product construction.

Example 10. In Example 2 we instantiated LSMTSs to modal transition systems labelled with actions, with well-formed label-set $K_{action} = (\Sigma \cup \{\bot\}, \sqsubseteq)$ for a finite set of actions Σ. The label operator for K_{action} can be defined as follows, depending on the desired synchronisation scheme:

— *Synchronisation by shared actions:*

$$a \oplus b = \begin{cases} a & \text{if } a = b \neq \bot \\ \bot & \text{if } a = \bot \text{ or } b = \bot \\ \text{undefined} & \text{otherwise.} \end{cases}$$

— *Complete interleaving:*

$$a \oplus b = \begin{cases} a & \text{if } a \neq \bot \text{ and } b = e \\ b & \text{if } a = e \text{ and } b \neq \bot \\ \bot & \text{if } a = \bot \text{ or } b = \bot \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Here we assume the existence of a special action $e \in K$ such that $s \xrightarrow{e} s$ for every state s.

Example 11. For weighted modal automata (see Example 4), the definition of $\oplus$ depends on how we want to interpret the weights. If the weights on transitions model are, for example, *costs* (or *energy consumption*), then the composition operator may be defined as the sum of intervals:

$$k \oplus \ell = \begin{cases} [i_1 + i_2, j_1 + j_2] & \text{if } k = [i_1, j_1] \text{ and } \ell = [i_2, j_2] \\ \bot & \text{if } k = \bot \text{ or } \ell = \bot. \end{cases}$$

Note that this label operator is total, so every transition in a weighted modal automaton will synchronise with each transition in the other automaton. Other options may include taking the interval intersection as the composition operator if, for example, the weights represent (discrete) time intervals in which a transition can be executed.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

For the rest of this section, we fix a well-formed label-set $(K, \sqsubseteq)$ with a compositional label operator $\oplus$ on it.

Definition 3.5 (parallel composition). Let $S, T \in \mathscr{M}_{(K,\sqsubseteq)}$ be two LSMTSs such that

$$S = (S, s_0, (K, \sqsubseteq), \dashrightarrow_S, \longrightarrow_S)$$
$$T = (T, t_0, (K, \sqsubseteq), \dashrightarrow_T, \longrightarrow_T).$$

The parallel composition of S and T is defined as the LSMTS

$$S \| T = (S \times T, (s_0, t_0), (K, \sqsubseteq), \dashrightarrow, \longrightarrow)$$

where the transition relations $\dashrightarrow$ and $\longrightarrow$ are defined by the rules:

$$\frac{s \overset{k}{\dashrightarrow}_S s' \quad t \overset{\ell}{\dashrightarrow}_T t' \quad k \oplus \ell \text{ is defined}}{(s,t) \overset{k\oplus\ell}{\dashrightarrow} (s',t')}$$

$$\frac{s \overset{k}{\longrightarrow}_S s' \quad t \overset{\ell}{\longrightarrow}_T t' \quad k \oplus \ell \text{ is defined}}{(s,t) \overset{k\oplus\ell}{\longrightarrow} (s',t')}.$$

As we have assumed that the label operator $\oplus$ is compositional, we get the property of compositional refinement, also called independent implementability (de Alfaro and Henzinger 2005). In other words, modal refinement is a precongruence with respect to parallel composition. This is formalised in the following theorem.

Theorem 3.6 (independent implementability). We let $S, S', T, T' \in \mathscr{M}_{(K,\sqsubseteq)}$ be LSMTSs and $\oplus$ be a compositional label operator on $(K, \sqsubseteq)$. If $S' \leqslant_m S$ and $T' \leqslant_m T$, then $S' \| T' \leqslant_m S \| T$.

Proof. We assume R_1 is a relation showing $S' \leqslant_m S$ and R_2 is a relation showing $T' \leqslant_m T$. We define a relation $R \subseteq (S' \times T') \times (S \times T)$ by $((s',t'),(s,t)) \in R$ if and only if $(s',s) \in R_1$ and $(t',t) \in R_2$. We will show that R witnesses $S' \| T' \leqslant_m S \| T$.

It is obvious that $((s'_0, t'_0), (s_0, t_0)) \in R$ where s_0, s'_0, t_0, t'_0 are the initial states of S, S', T, T', respectively. Let $((s',t'),(s,t)) \in R$. Then:

(1) Assume $(s',t') \overset{k'\oplus\ell'}{\dashrightarrow} (\hat{s}',\hat{t}')$.

By the rule of parallel composition, we have $s' \overset{k'}{\dashrightarrow} \hat{s}'$ and $t' \overset{\ell'}{\dashrightarrow} \hat{t}'$. Then, from $(s',s) \in R_1$ and $(t',t) \in R_2$, there exist $s \overset{k}{\dashrightarrow} \hat{s}$ and $t \overset{\ell}{\dashrightarrow} \hat{t}$ such that $k' \sqsubseteq k$, $\ell' \sqsubseteq \ell$, $(\hat{s}',\hat{s}) \in R_1$ and $(\hat{t}',\hat{t}) \in R_2$. From the fact that $\oplus$ is a compositional label operator, it follows that $k' \oplus l' \sqsubseteq k \oplus \ell$, and then $(s,t) \overset{k\oplus\ell}{\dashrightarrow} (\hat{s},\hat{t})$ and $((\hat{s}',\hat{t}'),(\hat{s},\hat{t})) \in R$.

(2) Assume $(s,t) \overset{k\oplus\ell}{\longrightarrow} (\hat{s},\hat{t})$.

By the rule of parallel composition, we have $s \overset{k}{\longrightarrow} \hat{s}$ and $t \overset{\ell}{\longrightarrow} \hat{t}$. Then, from $(s',s) \in R_1$ and $(t',t) \in R_2$, there exist $s' \overset{k'}{\longrightarrow} \hat{s}'$ and $t \overset{\ell'}{\longrightarrow} \hat{t}'$ such that $k' \sqsubseteq k$, $\ell' \sqsubseteq \ell$, $(\hat{s}',\hat{s}) \in R_1$ and $(\hat{t}',\hat{t}) \in R_2$. From the compositionality of the label operator, it follows that $k' \oplus \ell' \sqsubseteq k \oplus \ell$, and then $(s',t') \overset{k'\oplus\ell'}{\longrightarrow} (\hat{s}',\hat{t}')$ and $((\hat{s}',\hat{t}'),(\hat{s},\hat{t})) \in R$.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

Clearly, if $\oplus$ is commutative and associative, then so is the parallel composition (up to isomorphism).

3.3. *Conjunction*

Different component requirements can often be specified by independent teams. The issue of dealing with aspects of multiple viewpoints/properties is thus essential. It should be possible to represent several specifications (viewpoints) for the same implementation and then combine them in a logical manner. This is the objective of the *conjunction* operation.

Two LSMTSs with the same label-set $(K, \sqsubseteq)$ can be conjoined with respect to a label operator $\owedge$ on $(K, \sqsubseteq)$. We first state a necessary condition on $\owedge$ such that the conjunction operator yields the greatest lower bound with respect to the modal refinement relation on LSMTSs.

Definition 3.7 (greatest lower bound operator). A commutative label operator $\owedge$ on a well-formed set $(K, \sqsubseteq)$ is a *greatest lower bound operator* if the following is satisfied:

(1) If $k \owedge \ell$ is defined, then $k \owedge \ell \sqsubseteq k$ and $k \owedge \ell \sqsubseteq \ell$.
(2) If $m \neq \bot$, $m \sqsubseteq k$ and $m \sqsubseteq \ell$, then $k \owedge \ell$ is defined and $m \sqsubseteq k \owedge \ell$.

As in the case of the compositional label operator, it is easy to see that greatest lower bound operators are preserved by the product construction.

Proposition 3.8. Let $\owedge_1$ and $\owedge_2$ be greatest lower bound operators on $(K_1, \sqsubseteq_1)$ and $(K_2, \sqsubseteq_2)$, respectively. Then $\owedge_1 \times \owedge_2$ is a greatest lower bound operator on the product $(K_1, \sqsubseteq_1) \otimes (K_2, \sqsubseteq_2)$.

Again, the actual definition of $\owedge$ depends on the interpretation of conjunction for the modelled quantity.

Example 12. Conjoining labels in $K_{action} = (\Sigma \cup \{\bot\}, \sqsubseteq)$, for a finite set of actions Σ (see Example 2), can be defined as follows:

$$a \owedge b = \begin{cases} a & \text{if } a = b,\ a \neq \bot,\ b \neq \bot \\ \bot & \text{if } a = \bot \text{ or } b = \bot \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Example 13. For the case of weighted modal automata (see Example 4), conjunction $\owedge$ can be defined as the intersection of the intervals, assuming we consider the cost (energy) interpretation:

$$k \owedge \ell = \begin{cases} k \cap \ell & \text{if } k \neq \bot,\ \ell \neq \bot \text{ and } k \cap \ell \neq \varnothing \\ \bot & \text{otherwise.} \end{cases}$$

It is easy to see that the label operators defined in Examples 12 and 13 are greatest lower bound operators on their respective label-sets.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

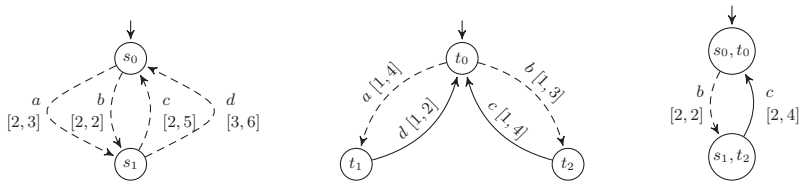


Fig. 8. Pruned conjunction of two LSMTSs S and T.

We fix a well-formed label-set $(K, \sqsubseteq)$ and some greatest lower bound operator $\otimes$ on $(K, \sqsubseteq)$ for the rest of this section.

Definition 3.9 (conjunction). Let $S, T \in \mathscr{M}_{(K,\sqsubseteq)}$ be two LSMTSs such that

$$S = (S, s_0, (K, \sqsubseteq), \dashrightarrow_S, \longrightarrow_S)$$
$$T = (T, t_0, (K, \sqsubseteq), \dashrightarrow_T, \longrightarrow_T).$$

The *conjunction of S and T* is defined by the LSMTS

$$S \wedge T = (S \times T, (s_0, t_0), (K, \sqsubseteq), \dashrightarrow, \longrightarrow)$$

where the transition relations $\dashrightarrow$ and $\longrightarrow$ are defined by the following rules:

$$\frac{s \xrightarrow{k}_S s' \quad t \overset{\ell}{\dashrightarrow}_T t' \quad k \otimes \ell \text{ is defined}}{(s,t) \xrightarrow{k \otimes \ell} (s',t')} \qquad \frac{s \overset{k}{\dashrightarrow}_S s' \quad t \xrightarrow{\ell}_T t' \quad k \otimes \ell \text{ is defined}}{(s,t) \xrightarrow{k \otimes \ell} (s',t')}$$

$$\frac{s \overset{k}{\dashrightarrow}_S s' \quad t \overset{\ell}{\dashrightarrow}_T t' \quad k \otimes \ell \text{ is defined}}{(s,t) \overset{k \otimes \ell}{\dashrightarrow} (s',t')}$$

$$\frac{s \xrightarrow{k}_S s' \quad \left(k \otimes \ell \text{ is not defined for any } \ell \text{ such that } t \overset{\ell}{\dashrightarrow}_T\right)}{(s,t) \xrightarrow{\perp} (s,t)}$$

$$\frac{t \xrightarrow{\ell}_T t' \quad \left(k \otimes \ell \text{ is not defined for any } k \text{ such that } s \overset{k}{\dashrightarrow}_S\right)}{(s,t) \xrightarrow{\perp} (s,t)}.$$

Clearly, conjunction $\wedge$ on LSMTSs is commutative (up to isomorphism) as $\otimes$ is commutative, and, moreover, if $\otimes$ is an associative label operator, then so is conjunction.

Example 14. An example for conjoining specifications is given in Figure 8. Here S and T are LSMTSs with the label-set $K_{action} \otimes K_{weighted}$. Note that the state (s_1, t_1) does not appear in $\rho(S \wedge T)$ since it is locally inconsistent in $S \wedge T$; the LSMTS T requires a transition labelled with the action d and the weight interval $[1,2]$, but S only allows d with the weight interval $[3,6]$.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

We now propose a notion of determinism that is dedicated to the conjunction operation. This definition will be used later to prove that conjunction is the greatest lower bound with respect to the modal refinement ordering.

Definition 3.10 ($\otimes$-determinism). A LSMTS $S \in \mathscr{M}_{(K,\sqsubseteq)}$ is $\otimes$-*deterministic* if for all may transitions $s \xrightarrow{k'} s'$ and $s \xrightarrow{k''} s''$ in S with labels $k', k'' \in K \setminus \{\bot\}$, whenever there is an $\ell \in K \setminus \{\bot\}$ such that both $k' \otimes \ell$ and $k'' \otimes \ell$ are defined, we have $k' = k''$ and $s' = s''$.

It is easy to see that if $\otimes$ is associative, then the construction for conjunction presented in Definition 3.9 preserves $\otimes$-determinism. Note that for modal transition systems with the label-set K_{action}, the notions of $\otimes$-determinism and determinism (as defined in Section 2) coincide. In this case, the determinisation algorithm proposed in Section 2.5 can be applied to compute minimal deterministic LSMTSs of non-deterministic ones.

Under the assumption of $\otimes$-determinism, the conjunction construction yields the greatest lower bound with respect to modal refinement, as stated in the following theorem.

Theorem 3.11 (greatest lower bound of conjunction). Let $S, T, U \in \mathscr{M}_{(K,\sqsubseteq)}$ be locally consistent LSMTSs such that S and T are $\otimes$-deterministic and $S \wedge T$ is consistent. Assume that $\otimes$ is a greatest lower bound label operator. Then:

(1) $\rho(S \wedge T) \leqslant_m S$ and $\rho(S \wedge T) \leqslant_m T$.
(2) If $U \leqslant_m S$ and $U \leqslant_m T$, then $U \leqslant_m \rho(S \wedge T)$.

Proof.

(1) It suffices to show that $\rho(S \wedge T) \leqslant_m S$, the other assertion being symmetric. We define a relation $R \subseteq (S \times T) \times S$ by $R = \{((s,t),s) \mid s \in S, t \in T\}$. We will show that R is a relation witnessing $\rho(S \wedge T) \leqslant_m S$. Clearly, $((s_0,t_0),s_0) \in R$ where s_0 is the initial state of S and t_0 is the initial state of T. Let $((s,t),s) \in R$.

(a) Let $(s,t) \xrightarrow{k \otimes \ell} (s',t')$.
Since $\rho(S \wedge T)$ does not contain any transitions labelled with $\bot$, we know that $k \otimes \ell \neq \bot$. Then there are transitions $s \xrightarrow{k} s'$ and $t \xrightarrow{\ell} t'$. By assumption, we know $k \otimes \ell \sqsubseteq k$, and from the definition of R, we can conclude that $((s',t'),s') \in R$.

(b) We now let $s \xrightarrow{k} s'$.
By the local consistency of S, we can assume that $k \neq \bot$. Suppose T does not have any $t \xrightarrow{\ell} t'$ such that $k \otimes \ell$ is defined, then $(s,t) \xrightarrow{\bot} (s',t)$, contradicting the local consistency of $\rho(S \wedge T)$. So there exists $t \xrightarrow{\ell} t'$ such that $k \otimes \ell$ is defined, so $(s,t) \xrightarrow{k \otimes \ell} (s',t')$. By the assumption about the label operator, we know that $k \otimes \ell \sqsubseteq k$. By the definition of R, we get $((s',t'),s') \in R$.

(2) We can assume a relation R_1 witnessing $U \leqslant_m S$ and a relation R_2 witnessing $U \leqslant_m T$. We define a relation $R \subseteq U \times (S \times T)$ by

$$R = \{(u,(s,t)) \mid (u,s) \in R_1 \text{ and } (u,t) \in R_2\}.$$

We now show that R is witnessing $U \leqslant_m \rho(S \wedge T)$. Clearly, $(u_0,(s_0,t_0)) \in R$ for the initial states. Let $(u,(s,t)) \in R$.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

(a) We assume that $u \overset{m}{\dashrightarrow} u'$.
Then there exists $s \overset{k}{\dashrightarrow} s'$ such that $m \sqsubseteq k$, and $t \overset{\ell}{\dashrightarrow} t'$ such that $m \sqsubseteq \ell$. By Definition 3.7 (2), we get that $k \oslash \ell$ is defined and $m \sqsubseteq k \oslash \ell$, so $(s,t) \overset{k \oslash \ell}{\dashrightarrow} (s',t')$, and $(u',(s',t')) \in R$ by the definition of R.

(b) We now assume that $(s,t) \overset{k \oslash \ell}{\longrightarrow} (s',t')$.
By the local consistency of $\rho(S \wedge T)$, we can assume that $k \oslash \ell \neq \bot$. Then (without loss of generality) there exist $s \overset{k}{\longrightarrow} s'$ with $k \neq \bot$, and $t \overset{\ell}{\dashrightarrow} t'$ with $\ell \neq \bot$ from the local consistency of S and T. It follows from $(u,s) \in R_1$ that there exists $u \overset{m}{\longrightarrow} u'$ such that $(u',s') \in R_1$ and $m \sqsubseteq k$. The local consistency of U implies $m \neq \bot$. We now have to show that $(u',t') \in R_2$. From $(u,t) \in R_2$, there exists $t \overset{\ell'}{\dashrightarrow} t''$ such that $m \sqsubseteq \ell'$ and $(u',t'') \in R_2$. Now, by $m \neq \bot$, $m \sqsubseteq k$ and $m \sqsubseteq \ell'$, we know that $k \oslash \ell'$ is defined (Definition 3.7 (2)). From $\oslash$-determinism of T, we get that $\ell = \ell'$ and $t' = t''$. It then follows that $(u',t') \in R_2$, and thus $(u',(s',t')) \in R$. □

Corollary 3.12. Let $S, T, U \in \mathscr{M}_{(K,\sqsubseteq)}$ be locally consistent LSMTSs such that S and T are $\oslash$-deterministic and $S \wedge T$ is consistent. Then $U \leqslant_m S$ and $U \leqslant_m T$ if and only if $U \leqslant_m \rho(S \wedge T)$. In particular, $[\![S \wedge T]\!] = [\![S]\!] \cap [\![T]\!]$.

Proof. The left-to-right implication is exactly the second part of Theorem 3.11. The other direction follows from the first part of Theorem 3.11 and the transitivity of $\sqsubseteq$. The additional assertion $[\![S \wedge T]\!] = [\![S]\!] \cap [\![T]\!]$ follows from $U \leqslant_m S$ and $U \leqslant_m T$ if and only if $U \leqslant_m \rho(S \wedge T)$ if we take U as an implementation, and by Proposition 2.14 showing that $[\![\rho(S \wedge T)]\!] = [\![S \wedge T]\!]$. □

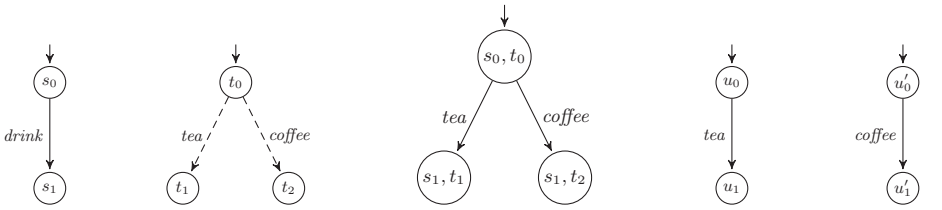
The maximality of the conjunction requires $\oslash$-determinism (see Theorem 3.11 (2)). To see this, consider the example shown in Figure 9, which shows two specifications of a vending machine with the label-set $K_{machine}$ from Example 3. The LSMTS S requires *drink*, and T allows *tea* and *coffee*. For the conjunction $S \wedge T$, we take $\oslash$ as the obvious greatest lower bound for $K_{machine}$ for which $drink \oslash tea = tea$ and $drink \oslash coffee = coffee$. Note that T is *not* $\oslash$-deterministic in this case. It is easy to see that $(S \wedge T) \leqslant_m S$ and $(S \wedge T) \leqslant_m T$. However, U and U' both refine S and T, but $U \not\leqslant_m (S \wedge T) = \rho(S \wedge T)$. With a small extension of the proof for modal transition systems (Delahaye *et al.* 2010), it is easy to prove that there is no LSMTS that is the greatest lower bound for S and T, but the construction of the conjunction in this case is at least safe.

3.4. *Quotient*

The *quotienting* operator is essential in a complete specification theory since it allows the factoring out of behaviours from a larger component. Given two component specifications S and T, the quotient of T by S, written $T \setminus\!\!\setminus S$, is a specification of exactly those components that when composed with S refine T. In other words, the quotient is the largest specification that can be composed with S and still refine T.

As expected, we have to first state the required property for label operators used for quotienting.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press



(a) S (b) T (c) $S \wedge T = \rho(S \wedge T)$ (d) U (e) U'

Fig. 9. Conjunction of two LSMTSs specifying a vending machine.

Definition 3.13 (dual label operators). Let $\ominus$ and $\oplus$ be two label operators on a given well-formed label-set $(K, \sqsubseteq)$. We say that the operator $\ominus$ is a *dual label operator to* $\oplus$ if $m \sqsubseteq \ell \ominus k$ if and only if $k \oplus m \sqsubseteq \ell$.

Example 15. Quotienting labels in $K_{action} = (\Sigma \cup \{\bot\}, \sqsubseteq)$ for the finite set of actions Σ introduced in Example 2 together with the $\oplus$ operator for synchronisation by shared actions given in Example 10 can be defined as identical to $\oplus$. Specifically,

$$a \ominus b = \begin{cases} a & \text{if } a = b \neq \bot \\ \bot & \text{if } a = \bot \text{ or } b = \bot \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Clearly, $\ominus$ is a dual label operator to $\oplus$.

Example 16. A more interesting example of $\ominus$ in $K_{weighted}$ (see Example 4) dual to the composition operator $\oplus$ from Example 11 that sums up weight intervals is given by

$$[a, b] \ominus [a', b'] = \begin{cases} [a - a', b - b'] & \text{if } [a - a', b - b'] \in K_{weighted} \\ \bot & \text{otherwise.} \end{cases}$$

For the rest of this section, we will fix two label operators $\ominus$ and $\oplus$ such that $\ominus$ is a dual label operator to $\oplus$ on a well-formed label-set $(K, \sqsubseteq)$.

Definition 3.14 (quotient). Let $S, T \in \mathscr{M}_{(K,\sqsubseteq)}$ be two LSMTS such that

$$S = (S, s_0, (K, \sqsubseteq), \dashrightarrow_S, \longrightarrow_S)$$
$$T = (T, t_0, (K, \sqsubseteq), \dashrightarrow_T, \longrightarrow_T).$$

The *quotient of T by S* is defined by

$$T \mathbin{\backslash\!\backslash} S = \big((T \times S) \cup \{\mathsf{u}\}, (t_0, s_0), (K, \sqsubseteq), \dashrightarrow, \longrightarrow\big)$$

where u is a new state modelling a universal state, and the transition relations $\dashrightarrow$ and $\longrightarrow$ are defined by the following rules:

$$\frac{t \overset{\ell}{\dashrightarrow}_T t' \quad s \overset{k}{\dashrightarrow}_S s' \quad \ell \ominus k \text{ is defined}}{(t, s) \overset{\ell \ominus k}{\dashrightarrow} (t', s')} \qquad \frac{t \overset{\ell}{\longrightarrow}_T t' \quad s \overset{k}{\longrightarrow}_S s' \quad \ell \ominus k \text{ is defined}}{(t, s) \overset{\ell \ominus k}{\longrightarrow} (t', s')}$$

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

$$\frac{t \xrightarrow{\ell}_T \quad \left(\ell \ominus k \text{ is not defined for any } k \text{ s.t. } s \xrightarrow{k}_S\right)}{(t,s) \xrightarrow{\perp} (t,s)}$$

$$\frac{m \in K \text{ and } m \oplus k \text{ is not defined for any } s \overset{k}{\dashrightarrow}_S}{(t,s) \overset{m}{\dashrightarrow} \mathsf{u}}$$

$$\frac{m \in K}{\mathsf{u} \overset{m}{\dashrightarrow} \mathsf{u}}.$$

The first two rules are derived from the two rules of Definition 3.5, while the third captures the inconsistency present when the larger system T has a must transition that cannot be mimicked by the smaller system S in parallel with any transition from the quotient. To achieve maximality of the quotient, we introduce a universal state u that allows for an arbitrary behaviour. Any allowed behaviour not structurally composable with the allowed behaviour of the smaller system can be safely added to the quotient (leading to the state u) as this will not affect the parallel composition with the smaller system. This is captured by the fourth rule.

Example 17. An example of quotienting is shown in Figure 10. Both specifications T and S have as the label-set $K_{action} \otimes K_{weighted}$ where $K_{action} = (\{a, b, \perp\}, \sqsubseteq)$, and $\rho(T \setminus\!\setminus S)$ is the result of the pruned quotient of T by S with respect to the product of the corresponding label operators from Examples 15 and 16 where the resulting operators are summarised in Figure 11 (these operators are already defined by the product construction, and we just present their combined definitions for clarity). In this way, a may transition under $[-\infty, \infty]$ between two states stands for all may transitions between those states under any label in $K_{weighted}$:

— The quotient $\rho(T \setminus\!\setminus S)$, in state (t_0, s_0), may do the action b with any weight (abbreviated by $[-\infty, \infty]$) and afterwards show arbitrary behaviour (reflected by the universal state) since S has no corresponding transition for action b in state s_0.
— Note that the state (t_3, s_3) is not present in the pruned quotient. If $\rho(T \setminus\!\setminus S)$ allowed a transition labelled with a (which would be possible with the weight interval $[0,0]$), then $(b, [1,4]) \ominus (b, [0,4])$ would yield $\perp$ (on a must transition), turning (t_3, s_3) into an inconsistent state with no implementation.

The quotient $T \setminus\!\setminus S$ is intended to synthesise the largest component that can be composed with S in order to refine T. In existing theories, such as modal transition systems, this maximality property only holds when the specifications are deterministic. As in the case of conjunction, we now propose a general notion of determinism for quotienting.

Definition 3.15 ($\oplus/\ominus$-determinism). Let $S \in \mathscr{M}_{(K,\sqsubseteq)}$ be an LSMTS. We say that S is $\oplus/\ominus$-*deterministic* if, for any $k', k'' \in K \setminus \{\perp\}$:

(1) whenever $s \overset{k'}{\dashrightarrow} s'$ and $s \overset{k''}{\dashrightarrow} s''$ and there exists $m \in K \setminus \{\perp\}$ such that $k' \oplus m$ and $k'' \oplus m$ are defined, we have $k' = k''$ and $s' = s''$; and

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

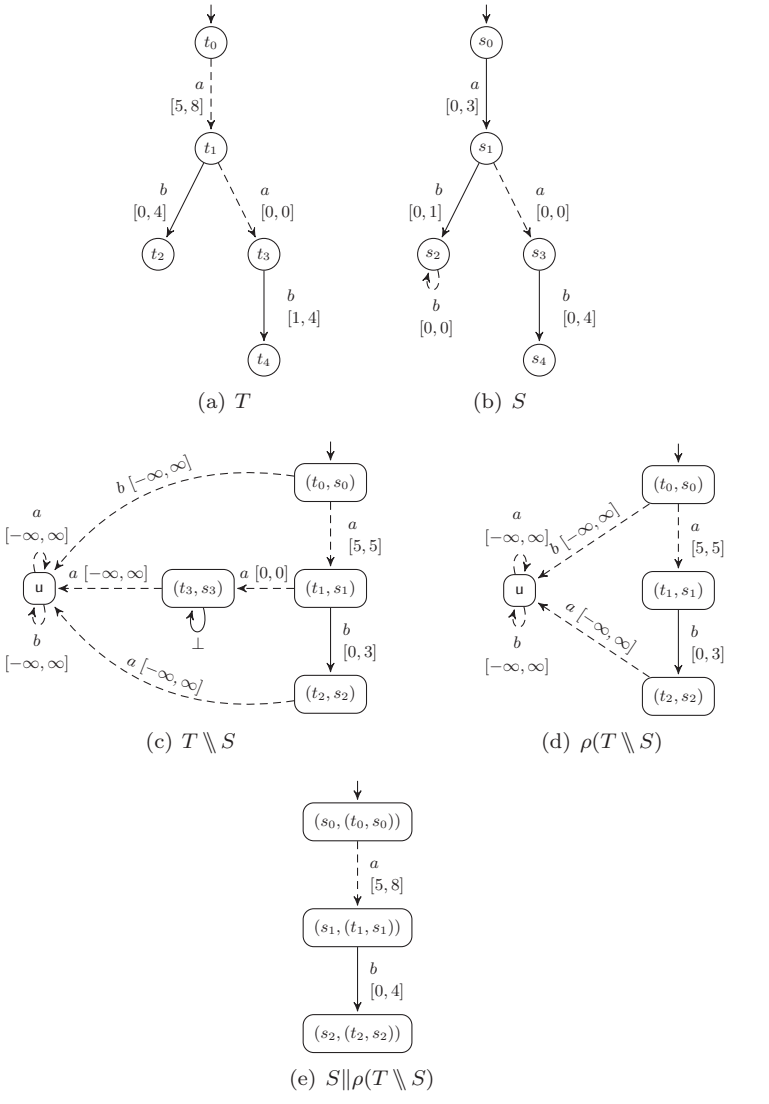


Fig. 10. Quotienting of two LSMTSs.

(2) whenever $s \xrightarrow{k'} s'$ and $s \xrightarrow{k''} s''$ and there exists $\ell \in K \setminus \{\bot\}$ such that $\ell \ominus k'$ and $\ell \ominus k''$ are defined, we have $k' = k''$ and $s' = s''$.

Note that for modal transition systems with the label-set K_{action}, the notions of $\oplus/\ominus$-determinism and determinism (as defined in Section 2) as well as $\oslash$-determinism coincide. Here, in the case of non-deterministic LSMTSs, the determinisation algorithm proposed in Section 2.5 can be applied to compute minimal deterministic versions of them.

Under the assumption of $\oplus/\ominus$-determinism, the quotient construction $T \setminus\!\setminus S$ yields the most general LSMTS that, composed with S, still refines T.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

$$(a,i) \oplus (a',i') = \begin{cases} a, [l+l', r+r'] & \text{if } a = a' \neq \bot,\ i = [l,r],\ i' = [l',r'] \\ \bot & \text{if } a = \bot \text{ or } a' = \bot \text{ or } i = \bot \text{ or } i' = \bot \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$(a,i) \ominus (a',i') = \begin{cases} a, [l-l', r-r'] & \text{if } a = a' \neq \bot,\ i = [l,r],\ i' = [l',r'],\ [l-l', r-r'] \in K_{weighted} \\ \bot & \text{if } a = \bot \text{ or } a' = \bot \text{ or } i = \bot \text{ or } i' = \bot \\ & \text{or } i = [l,r],\ i' = [l',r'] \text{ and } [l-l', r-r'] \notin K_{weighted} \\ \text{undefined} & \text{otherwise} \end{cases}$$

Fig. 11. Label operators $\oplus$ and $\ominus$ for quotienting with the label-set $K_{action} \otimes K_{weighted}$.

Theorem 3.16 (soundness and maximality of quotient). Let $S, T, X \in \mathscr{M}_{(K,\sqsubseteq)}$ be locally consistent LSMTSs such that $T \setminus\!\setminus S$ is consistent. Assume that S is $\oplus/\ominus$-deterministic and that $\ominus$ is the dual label operator to $\oplus$. Then $X \leqslant_m \rho(T \setminus\!\setminus S)$ if and only if $S \| X \leqslant_m T$.

Proof.

($\Longrightarrow$) We assume a relation R_1 witnessing $X \leqslant_m \rho(T \setminus\!\setminus S)$ and define a relation

$$R_2 \subseteq (S \times X) \times T$$

by

$$R_2 = \{((s,x),t) \mid (x,(t,s)) \in R_1\}.$$

We now show that R_2 is a relation witnessing $S \| X \leqslant_m T$. Clearly, $((s_0,x_0),t_0) \in R_2$ for the corresponding initial states.
Let $((s,x),t) \in R_2$. Then:

(1)Assume $(s,x) \overset{k \oplus m}{\dashrightarrow} (s',x')$.
Hence, there exist $s \overset{k}{\dashrightarrow} s'$ and $x \overset{m}{\dashrightarrow} x'$. Since X is locally consistent, we know $m \neq \bot$. From $(x,(t,s)) \in R_1$ and $x \overset{m}{\dashrightarrow} x'$, there exists $(t,s) \overset{\ell \ominus k'}{\dashrightarrow} (t',s'')$ such that $m \sqsubseteq \ell \ominus k'$ and $(x',(t',s'')) \in R_1$. From $m \sqsubseteq \ell \ominus k'$, we get $k' \oplus m \sqsubseteq \ell$ by assumption. Moreover, we have $t \overset{\ell}{\dashrightarrow} t'$ and $s \overset{k'}{\dashrightarrow} s''$. Then, from the $\oplus/\ominus$-determinism of S, it follows that $k = k'$ and $s' = s''$, so $k \oplus m \sqsubseteq \ell$. Thus $(x',(t',s')) \in R_1$, and thus $((s',x'),t') \in R_2$.

(2)Assume $t \overset{\ell}{\longrightarrow} t'$.
By the local consistency of T, we can assume that $\ell \neq \bot$. If there is no transition $s \overset{k}{\longrightarrow} s'$ such that $\ell \ominus k$ is defined, then $(t,s) \overset{\bot}{\longrightarrow}$, which contradicts $(t,s) \in \rho(T \setminus\!\setminus S)$. It follows that there exists $s \overset{k}{\longrightarrow} s'$ such that $\ell \ominus k$ is defined, and $(t,s) \overset{\ell \ominus k}{\longrightarrow} (t',s')$. By a similar argument to the one given above, we know that $\ell \ominus k \neq \bot$. Then $(x,(t,s)) \in R_1$ means that there exists $x \overset{m}{\longrightarrow} x'$ such that $(x',(t',s')) \in R_1$ and $m \sqsubseteq \ell \ominus k$, which implies $k \oplus m \sqsubseteq \ell$. We can then conclude that $(s,x) \overset{k \oplus m}{\longrightarrow} (s',x')$ and $((s',x'),t') \in R_2$.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

($\Longleftarrow$) We assume a relation R_2 witnessing $S \| X \leqslant_m T$ and define a relation

$$R_1 \subseteq X \times (T \times S)$$

by

$$R_1 = \{(x,(t,s)) \mid ((s,x),t) \in R_2\} \cup \{(x,\mathsf{u})\}.$$

We will now show that R_1 is a relation witnessing $X \leqslant_m \rho(T \setminus\!\setminus S)$. Clearly, we have $(x_0,(t_0,s_0)) \in R_1$ for the corresponding initial states. We first let $(x,\mathsf{u}) \in R_1$. Then, clearly, for every transition $x \overset{m}{\dashrightarrow} x'$, we again have $(x',\mathsf{u}) \in R_1$ since the universal state u allows arbitrary behaviour.

We now let $(x,(t,s)) \in R_1$. Then:

(1)Assume $x \overset{m}{\dashrightarrow} x'$.

If there is no transition $s \overset{k}{\dashrightarrow} s'$ such that $k \oplus m$ is defined, then $(t,s) \overset{m}{\dashrightarrow} \mathsf{u}$, and in this case, $(x,\mathsf{u}) \in R_1$, and, clearly, $m \sqsubseteq m$ by reflexivity of $\sqsubseteq$. If there is a transition $s \overset{k}{\dashrightarrow} s'$ such that $k \oplus m$ is defined, then $(s,x) \overset{k\oplus m}{\dashrightarrow} (s',x')$. From $((s,x),t) \in R_2$, there exists $t \overset{\ell}{\dashrightarrow} t'$ such that $((s',x'),t') \in R_2$ and $k \oplus m \sqsubseteq \ell$, implying $m \sqsubseteq \ell \ominus k$. Hence $(t,s) \overset{\ell\ominus k}{\dashrightarrow} (t',s')$ and $(x',(t',s')) \in R_1$.

(2)Assume $(t,s) \overset{\ell\ominus k}{\longrightarrow} (t',s')$.

From the local consistency of $\rho(T \setminus\!\setminus S)$, we know $\ell \ominus k \neq \bot$. So there are $t \overset{\ell}{\longrightarrow} t'$ and $s \overset{k}{\longrightarrow} s'$. From $((s,x),t) \in R_2$, we can then conclude that there exists $(s,x) \overset{k'\oplus m}{\longrightarrow} (s'',x')$ such that $k' \oplus m \sqsubseteq \ell$ and $((s'',x'),t') \in R_2$. So there exist $s \overset{k'}{\longrightarrow} s''$ and $x \overset{m}{\longrightarrow} x'$. The fact that $k' \oplus m \sqsubseteq \ell$ implies $m \sqsubseteq \ell \ominus k'$ by the duality of the operators. From the $\oplus/\ominus$-determinism of S, it follows that $k = k'$ and $s' = s''$. Thus $m \sqsubseteq \ell \ominus k$ and $((s',x'),t') \in R_2$, hence $(x',(t',s')) \in R_1$. □

4. Logical characterisation

Larsen (1989) showed that Hennessy–Milner logic (Hennessy and Milner 1985) can be used as a logical characterisation for modal refinement of modal transition systems (see also Bruns and Godefroid (2000)). In this section we shall extend this result to LSMTSs and study other related topics. For the rest of this section, we fix a well-formed label-set $(K,\sqsubseteq)$.

We will first introduce LSHML as an extension of Hennessy–Milner logic (HML) that is interpreted over LSMTSs, taking into account their label structures. The syntax of the logic is given by the abstract syntax

$$\varphi ::= \mathsf{true} \mid \mathsf{false} \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \langle k \rangle \psi \mid [k]\psi$$

where $k \in K$ is a label. We define the $\vee$-free fragment of LSHML as a set of formulae in LSHML not containing the disjunction operator.

Let $S \in \mathscr{M}_{(K,\sqsubseteq)}$ be an LSMTS. The satisfaction relation between a state $s \in S$ and a formula φ is defined inductively as follows:

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

$s \models \mathsf{true}$
$s \not\models \mathsf{false}$
$s \models \varphi_1 \wedge \varphi_2$ iff $S \models \varphi_1$ and $S \models \varphi_2$
$s \models \varphi_1 \vee \varphi_2$ iff $S \models \varphi_1$ or $S \models \varphi_2$
$s \models \langle k \rangle \varphi$ iff $\exists (s \xrightarrow{\ell} s') : [\![\ell]\!] \subseteq [\![k]\!]$ and $s' \models \varphi$
$s \models [k]\varphi$ iff $\forall (s \overset{\ell}{\dashrightarrow} s')$ s.t. $[\![\ell]\!] \cap [\![k]\!] \neq \varnothing : s' \models \varphi$

We write $S \models \varphi$ iff $s_0 \models \varphi$ where s_0 is the initial state of S.

Example 18. Consider the specification of a vending machine given in Figure 3(c). The specification satisfies the property that after inserting a 1€ coin, a drink is guaranteed, as we have $s_0 \models [1EURO]\langle drink \rangle \mathsf{true}$. On the other hand, we are not guaranteed to receive a cup of tea if a coin is inserted as $s_0 \not\models [coin]\langle tea \rangle \mathsf{true}$.

We are now ready to prove the soundness and completeness theorems for our logic. The following theorem ensures soundness of LSHML, that is, if a formula holds for a specification, then it holds for any of its refinements.

Theorem 4.1 (soundness). Let $T \in \mathscr{M}_{(\mathrm{K},\sqsubseteq)}$ and φ be a LSHML-formula. Then

$$T \models \varphi \quad \Longrightarrow \quad \forall S \leqslant_m T : S \models \varphi \, .$$

Proof. Assume that $S \leqslant_m T$ and $T \models \varphi$. We prove by induction on the structure of φ that $S \models \varphi$ too. Let s_0 and t_0 be the initial states of S and T, respectively. The induction basis, where $\varphi = \mathsf{true}$ and $\varphi = \mathsf{false}$, is trivial.

$\varphi = \varphi_1 \wedge \varphi_2$.
This case follows by the definition of $\models$ and then from the induction hypothesis.

$\varphi = \varphi_1 \vee \varphi_2$.
This case follows in the same way as the case above.

$\varphi = \langle k \rangle \psi$.
From $T \models \langle k \rangle \psi$, there exists $t_0 \xrightarrow{\ell} t$ such that $[\![\ell]\!] \subseteq [\![k]\!]$ and $t \models \psi$. Since $S \leqslant_m T$, there exists $s_0 \xrightarrow{\ell'} s$ such that $\ell' \sqsubseteq \ell$ and $(s, S) \leqslant_m (t, T)$. By the induction hypothesis, we get $s \models \psi$. From the transitivity of $\sqsubseteq$, we have $[\![\ell']\!] \subseteq [\![k]\!]$ and thus $S \models \varphi$.

$\varphi = [k]\psi$.
Let $s_0 \overset{\ell}{\dashrightarrow} s$ such that $[\![k]\!] \cap [\![\ell]\!] \neq \varnothing$. Since $S \leqslant_m T$, we know that there exists $t_0 \overset{\ell'}{\dashrightarrow} t$ in T with $\ell \sqsubseteq \ell'$ and $(s, S) \leqslant_m (t, T)$. Clearly, we have $[\![k]\!] \cap [\![\ell']\!] \neq \varnothing$, and because $T \models [k]\psi$, we know that $t \models \psi$. By the induction hypothesis, we then get $s \models \psi$, and thus $S \models \varphi$. □

We shall now focus on the issue of completeness. We consider two possible definitions:

(1) *Completeness with respect to implementations*: if all implementations of a specification S satisfy a formula of the logic, then so does the specification S.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

(2) *Completeness with respect to modal refinement*[†]: if all formulae satisfied by some specification S are satisfied also by another specification T, then $S \leqslant_m T$.

We will first study completeness with respect to implementations and observe that LSHML-logic is not complete in this case.

Theorem 4.2. The logic LSHML is incomplete with respect to implementations.

Proof. Let T be an LSMTS consisting of a single transition $t_0 \overset{\bullet}{\dashrightarrow} t$ over the label-set $K_{unlabelled}$ from Example 1. Consider the formula $\varphi = \langle\bullet\rangle\mathsf{true} \vee [\bullet]\mathsf{false}$. Since there is no must transition from t_0 and at the same time there is a may transition, we get $t_0 \not\models \varphi$. On the other hand, any implementation of T either contains no transition at all (and then it satisfies $[\bullet]\mathsf{false}$) or it contains at least one outgoing transition (and then it satisfies $\langle\bullet\rangle\mathsf{true}$). Hence any implementation of T satisfies φ and we get the incompleteness result with respect to implementations. □

Note that it is the disjunction that breaks the completeness property in the proof of the above theorem. In fact, we can show completeness if we just consider the $\vee$-free fragment of LSHML.

Theorem 4.3 (completeness with respect to implementations for the $\vee$-free fragment of LSHML). Let $T \in \mathscr{M}_{(K,\sqsubseteq)}$ be a locally consistent specification, and let φ be a $\vee$-free LSHML-formula. Then

$$(\forall I \in [\![T]\!] : I \models \varphi) \quad \Longrightarrow \quad T \models \varphi \ .$$

Proof. We prove the contraposition by showing that for any $\vee$-free LSHML-formula φ

$$\text{if } T \not\models \varphi, \text{ then there exists } I \in [\![T]\!] \text{ such that } I \not\models \varphi \ .$$

The proof is by induction on the structure of the formula φ, and under the assumption that $T \not\models \varphi$, we construct its implementation (i_0, I) such that $i_0 \not\models \varphi$. During the construction, we will write that we add a transition $i_0 \xrightarrow{n} (i_0', I')$ for an implementation (i_0', I'), meaning that together with this transition, we implicitly also add a disjoint copy of I' rooted at i_0' to the implementation I.

The induction basis, where $\varphi = \mathsf{true}$ and $\varphi = \mathsf{false}$, is trivial.

— $\varphi = \varphi_1 \wedge \varphi_2$.
  By the definition of $\models$, either $T \not\models \varphi_1$ or $T \not\models \varphi_2$. We can assume without loss of generality that $T \not\models \varphi_1$. Applying the induction hypothesis, there is $I \in [\![T]\!]$ such that $I \not\models \varphi_1$ and we conclude that $I \not\models \varphi_1 \wedge \varphi_2$.
— $\varphi = \langle k \rangle \psi$.

[†] Completeness with respect to modal refinement is also known as logical characterisation in the literature (Larsen 1989).

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

We assume $T \not\models \langle k \rangle \psi$, which is the case if for all $t_0 \xrightarrow{\ell} t$ we have either:

(1)$[\![\ell]\!] \not\subseteq [\![k]\!]$; or

(2)$(t, T) \not\models \psi$.

We construct an implementation $(i_0, I) \in [\![T]\!]$ as follows. For every $t_0 \xrightarrow{\ell} t$ such that (1) is satisfied, we add the transition $i_0 \xrightarrow{n} (i'_0, I')$ into I where $n \in [\![\ell]\!] \setminus [\![k]\!]$ and $(i'_0, I') \leqslant_m (t, T)$ (the implementation (i'_0, I') exists by local consistency of T and well formedness of the label-set). For every $t_0 \xrightarrow{\ell} t$ such that (2) is satisfied, we have by the induction hypothesis an implementation $(i'_0, I') \in [\![(t, T)]\!]$ such that $i'_0 \not\models \psi$. We then add $i_0 \xrightarrow{m} (i'_0, I')$ to I for some $m \in [\![\ell]\!]$. It is easy to see that $I \leqslant_m T$, and, moreover, $I \not\models \langle k \rangle \psi$ by the construction.

— $\varphi = [k]\psi$.

We assume $T \not\models [k]\psi$. Then there exists $t_0 \xrightarrow{\ell} t$ such that $[\![\ell]\!] \cap [\![k]\!] \neq \varnothing$ and $(t, T) \not\models \psi$. By the induction hypothesis, there exists $(i'_0, I') \in [\![(t, T)]\!]$ such that $i'_0 \not\models \psi$. Let $(i_0, I) \in [\![T]\!]$ be some implementation of T (which exists by local consistency of T) where we add the transition $i_0 \xrightarrow{n} (i'_0, I')$ with $n \in [\![\ell]\!] \cap [\![k]\!]$. Clearly, we still have $I \leqslant_m T$ and, moreover, the transition $i_0 \xrightarrow{n} i'_0$ ensures that $i_0 \not\models [k]\psi$. □

A similar completeness result in the setting of partial Kripke structures can also be found in Antonik and Huth (2006).

We now study completeness with respect to modal refinement, which is the completeness definition considered in Larsen (1989), which showed that for classical modal transition systems (with the label-set K_{action} from Example 2), the LSHML logic is complete with respect to refinement. We first observe that this result does not extend to general LSMTSs.

Theorem 4.4. The logic LSHML is incomplete with respect to modal refinement.

Proof. Consider the systems S and T from Figure 6. By case analysis, it is easy to verify that $s_0 \models \varphi$ if and only if $t_0 \models \varphi$ for any LSHML-formula φ. However, as argued before, $S \not\leqslant_m T$. □

On the other hand, if we just consider deterministic systems, LSHML is complete even with disjunction, as proved below. We let $\mathscr{F}(S) = \{\varphi \mid S \models \varphi\}$ denote the set of all LSHML-formulae satisfied by S.

Theorem 4.5 (completeness with respect to modal refinement for deterministic LSMTSs). Let $S, T \in \mathscr{M}_{(\mathrm{K},\sqsubseteq)}$ be deterministic LSMTSs (see Definition 2.16) and assume that the label refinement relation $\sqsubseteq$ is complete. Then

$$\mathscr{F}(T) \subseteq \mathscr{F}(S) \quad \Longrightarrow \quad S \leqslant_m T \ .$$

Proof.

Assume that $\mathscr{F}(T) \subseteq \mathscr{F}(S)$. We define a relation $R \subseteq S \times T$ by

$$R = \{(s, t) \mid \mathscr{F}((t, T)) \subseteq \mathscr{F}((s, S))\}.$$

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

We will show that R is a relation witnessing $S \leqslant_m T$. Clearly, $(s_0, t_0) \in R$ for the respective initial states. Let $(s, t) \in R$.

— Assume $s \overset{k}{\dashrightarrow} s'$.
Clearly, $t \overset{\ell}{\dashrightarrow} t'$ for some ℓ such that $[\![k]\!] \cap [\![\ell]\!] \neq \varnothing$, otherwise the formula $[k]\mathsf{false}$ is satisfied in (t, T) but not in (s, S), contradicting our assumption that

$$\mathscr{F}((t, T)) \subseteq \mathscr{F}((s, S)).$$

By the determinism of T, there can only be one such ℓ with $[\![k]\!] \cap [\![\ell]\!] \neq \varnothing$.
In order to show a contradiction, we assume that $k \not\sqsubseteq \ell$. By the completeness of label refinement, we get $[\![k]\!] \not\subseteq [\![\ell]\!]$. Thus, there exists some $m \in [\![k]\!] \setminus [\![\ell]\!]$. The formula $[m]\mathsf{false}$ holds in (t, T) because of the choice of m and the absence of any other may transition having any common implementation labels with $[\![k]\!]$, hence, in particular, it also holds for $[\![m]\!] = \{m\}$. However, $[m]\mathsf{false}$ does not hold in (s, S), contradicting the assumption that $\mathscr{F}((t, T)) \subseteq \mathscr{F}((s, S))$. Thus, we can assume the existence of $t \overset{\ell}{\dashrightarrow} t'$ with $k \sqsubseteq \ell$.
Now we need to argue that $(s', t') \in R$. Assume that this is not the case. Then we have $\mathscr{F}((t', T)) \not\subseteq \mathscr{F}((s', S))$, and therefore there is a formula φ' such that $(t', T) \models \varphi'$ and $(s', S) \not\models \varphi'$. Consider the formula $\varphi = [k]\varphi'$. Again $(t, T) \models \varphi$, but $(s, S) \not\models \varphi$, which contradicts the assumption that $\mathscr{F}((t, T)) \subseteq \mathscr{F}((s, S))$. Thus $(s', t') \in R$.

— Assume $t \overset{\ell}{\longrightarrow} t'$.
As in the previous case, there must be a transition $s \overset{k}{\longrightarrow} s'$ such that $[\![k]\!] \subseteq [\![\ell]\!]$, otherwise the formula $\langle\ell\rangle\mathsf{true}$ is satisfied in (t, T) but not in (s, S), contradicting the assumption that $\mathscr{F}((t, T)) \subseteq \mathscr{F}((s, S))$. By the determinism of S, we know that $s \overset{k}{\longrightarrow} s'$ is a unique transition such that $[\![k]\!] \subseteq [\![\ell]\!]$, and from the completeness of label refinement, we know that $k \sqsubseteq \ell$.
We still need to show that $(s', t') \in R$. Assume that this is not the case. The argument is then similar to the previous case by considering the formula $\langle\ell\rangle\varphi'$ where $t' \models \varphi'$ and $s' \not\models \varphi'$. Thus $(s', t') \in R$, which completes the proof. □

5. Conclusion

We have introduced label-structured modal transition systems, which can form a basis for a specification formalism that generalises modal theories such as weighted and multi-weighted modal transition systems. Our work contributes to the long-term objective of unifying existing specification theories through a common framework. We have developed a full specification theory for label-structured modal transition systems, including the notions of modal and thorough refinement, consistency, determinisation and deterministic-hull, together with a number of completeness results, which (as expected) are often conditional on a requirement of determinism. We have shown soundness results for the operators of parallel composition, conjunction and quotient. The specification theory was concluded by suggesting an extension of Hennessy–Milner logic to handle quantitative aspects and by showing the interplay between the logic and the refinement theory in a

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

similar way to that known from the classical theory of labelled transition systems and bisimulation.

Most of the proof techniques were generalisations of the techniques developed for concrete instances of the framework, such as modal transition systems. However, the general theorems provide novel results for particular instances like weighted and multi-weighted modal transition systems. Finally, we considered the uniform and complete presentation of the main aspects of the suggested specification theory as a contribution on its own.

A few recently studied extensions of modal transition systems cannot be captured in our framework, and we will now briefly discuss some of them as possible directions for future research.

Abstract Probabilistic Automata (Delahaye *et al.* 2011) and constraint Markov Chains (Caillaud *et al.* 2010) were recently introduced as stochastic extensions of modal transition systems. One of the major difficulties for our framework is knowing how to capture and generalise the satisfaction relation, which is (unlike in our framework) based on a redistribution of weights from one to several transitions.

The theories based on the optimistic approach introduced in Interface Automata (de Alfaro and Henzinger 2001; de Alfaro *et al.* 2002) are also hard to capture in our framework. Here the semantics of a given specification is viewed as a two-player game. The approach is optimistic in the sense that two specifications can be composed if and only if there exists at least one environment in which they can cooperate. The input and output modalities are orthogonal to may and must ones, which suggests that the label-structured modal transition systems need to be further extended to capture this phenomenon.

Finally, it would be interesting to consider models that manipulate data along the lines of sociable interfaces (de Alfaro *et al.* 2005; Adler *et al.* 2006), and see if they can be described in the framework of label-structured modal transition systems.

References

Adler, B. T., de Alfaro, L., da Silva, L. D., Faella, M., Legay, A., Raman, V., and Roy, P. (2006) Ticc: A tool for interface compatibility and composition. In: Proceedings 18th International Conference on Computer Aided Verification (CAV). *Springer-Verlag Lecture Notes in Computer Science* **4144** 59–62.

Antonik, A. and Huth, M. (2006) Efficient patterns for model checking partial state spaces in CTL ∩ LTL. *Electronic Notes in Theoretical Computer Science* **158** (0) 41–57.

Bauer, S. S., Mayer, P., Schroeder, A. and Hennicker, R. (2010) On weak modal compatibility, refinement, and the mio workbench. In: Esparza, J. and Majumdar, R. (eds.) TACAS. *Springer-Verlag Lecture Notes in Computer Science* **6015** 175–189.

Beneš, N., Křetínský, J., Larsen, K. G. and Srba, J. (2009a) Checking thorough refinement on modal transition systems is exptime-complete. In: Leucker, M. and Morgan, C. (eds.) ICTAC. *Springer-Verlag Lecture Notes in Computer Science* **5684** 112–126.

Beneš, N., Křetínský, J., Larsen, K. G. and Srba, J. (2009b) On determinism in modal transition systems. *Theoretical Computer Science* **410** (41) 4026–4043.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

Bertrand, N., Legay, A., Pinchinat, S. and Raclet, J.-B. (2009a) A compositional approach on modal specifications for timed systems. In: Breitman, K. and Cavalcanti, A. (eds.) ICFEM. *Springer-Verlag Lecture Notes in Computer Science* **5885** 679–697.

Bertrand, N., Pinchinat, S. and Raclet, J.-B. (2009b) Refinement and consistency of timed modal specifications. In: Dediu, A. H., Ionescu, A.-M. and Martín-Vide, C. (eds.) LATA. *Springer-Verlag Lecture Notes in Computer Science* **5457** 152–163.

Boudol, G. and Larsen, K. G. (1992) Graphical versus logical specifications. *Theoretical Computer Science* **106** (1) 3–20.

Bruns, G. and Godefroid, P. (2000) Generalized model checking: Reasoning about partial state spaces. In Palamidessi, C. (ed.) Proceedings of the 11th International Conference on Concurrency Theory (CONCUR'00). *Springer-Verlag Lecture Notes in Computer Science* **1877** 168–182.

Caillaud, B., Delahaye, B., Larsen, K. G., Legay, A., Pedersen, M. L. and Wasowski, A. (2010) Compositional design methodology with constraint Markov chains. In: *Proceedings of 7th International Conference on Quantitative Evaluation of SysTems (QEST)*, IEEE Computer Society 123–132.

David, A., Larsen, K. G., Legay, A., Nyman, U. and Wasowski, A. (2010) Timed i/o automata: a complete specification theory for real-time systems. In: Johansson, K. H. and Yi, W. (eds.) *HSCC*, ACM 91–100.

de Alfaro, L. and Henzinger, T. A. (2001) Interface automata. In: *Proceedings of the 8th ESEC and 9th ACM SIGSOFT FSE*, ACM 109–120.

de Alfaro, L. and Henzinger, T. A. (2005) Interface-based Design. In: Broy, M., Grünbauer, J., Harel, D. and Hoare, C. A. R. (eds.) *Engineering Theories of Software-intensive Systems*, NATO Science Series: Mathematics, Physics, and Chemistry **195**, Springer-Verlag 83–104.

de Alfaro, L., da Silva, L. D., Faella, M., Legay, A., Roy, P. and Sorea, M. (2005) Sociable interfaces. In: Proceedings 5th International Conference on Frontiers of Combining Systems (FROCOS). *Springer-Verlag Lecture Notes in Computer Science* **3717** 81–105.

de Alfaro, L., Henzinger, T. A. and Stoelinga, M. (2002) Timed interfaces. In: Sangiovanni-Vincentelli, A. L. and Sifakis, J. (eds.) EMSOFT. *Springer-Verlag Lecture Notes in Computer Science* **2491** 108–122.

Delahaye, B., Katoen, J.-P., Larsen, K. G., Legay, A., Pedersen, M. L., Sher, F. and Wasowski, A. (2011) Abstract probabilistic automata. In: Jhala, R. and Schmidt, D. (eds.) Proceedings of 12th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI). *Springer-Verlag Lecture Notes in Computer Science* **6538** 324–339.

Delahaye, B., Larsen, K., Legay, A. and Wasowski, A. (2010) On greatest lower bound of modal transition systems. Technical report, INRIA.

Feuillade, G. and Pinchinat, S. (2007) Modal specifications for the control theory of discrete event systems. *Discrete Event Dynamic Systems* **17** (2) 211–232.

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

Godefroid, P., Huth, M. and Jagadeesan, R. (2001) Abstraction-based model checking using modal transition systems. In: Larsen, K. G. and Nielsen, M. (eds.) CONCUR. *Springer-Verlag Lecture Notes in Computer Science* **2154** 426–440.

Gordon, M. (1979) *The Denotational Description of Programming Languages*, Springer-Verlag.

Gruler, A., Leucker, M. and Scheidemann, K. D. (2008) Calculating and modeling common parts of software product lines. In: *SPLC*, IEEE Computer Society 203–212.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

Hennessy, M. and Milner, R. (1985) Algebraic laws for non-determinism and concurrency. *Journal of the ACM* **32** (1) 137–161.

Hoare, C. A. R. (1969) An axiomatic basis for computer programming. *Communications of the ACM* **12** 576–580.

Hoare, C. A. R. (1985) *Communicating Sequential Processes*, Prentice-Hall.

Juhl, L., Larsen, K. G. and Srba, J. (2010) Modal Transition Systems with Weight Intervals.

Katoen, J.-P., Klink, D. and Neuhäußer, M. R. (2009) Compositional abstraction for stochastic systems. In: Ouaknine, J. and Vaandrager, F. W. (eds.) FORMATS. *Springer-Verlag Lecture Notes in Computer Science* **5813** 195–211.

Larsen, K. G. (1985) A context dependent equivalence between processes. In: Brauer, W. (ed.) ICALP. *Springer-Verlag Lecture Notes in Computer Science* **194** 373–382.

Larsen, K. G. (1987) A context dependent equivalence between processes. *Theoretical Computer Science* **49** 184–215.

Larsen, K. G. (1989) Modal specifications. In: Sifakis, J. (ed.) Automatic Verification Methods for Finite State Systems, Proceedings International Workshop, Grenoble, France. *Springer-Verlag Lecture Notes in Computer Science* **407** 232–246.

Larsen, K. G. and Milner, R. (1987) Verifying a protocol using relativized bisimulation. In: Ottmann, T. (ed.) ICALP. *Springer-Verlag Lecture Notes in Computer Science* **267** 126–135.

Larsen, K. G. and Milner, R. (1992) A compositional protocol verification using relativized bisimulation. *Information and Computing* **99** (1) 80–108.

Larsen, K. G. and Thomsen, B. (1988a) Compositional proofs by partial specification of processes. In: Chytil, M., Janiga, L. and Koubek, V. (eds.) MFCS. *Springer-Verlag Lecture Notes in Computer Science* **324** 414–423.

Larsen, K. G. and Thomsen, B. (1988b) A modal process logic. In: *LICS*, IEEE Computer Society 203–210.

Larsen, K. G., Larsen, U. and Wasowski, A. (2005) Color-blind specifications for transformations of reactive synchronous programs. In: Cerioli, M. (ed.) FASE *Springer-Verlag Lecture Notes in Computer Science* **3442** 160–174.

Larsen, K. G., Mikucionis, M. and Nielsen, B. (2004) Online testing of real-time systems using uppaal. In: Grabowski, J. and Nielsen, B. (eds.) FATES. *Springer-Verlag Lecture Notes in Computer Science* **3395** 79–94.

Larsen, K. G., Nyman, U. and Wasowski, A. (2007) Modal i/o automata for interface and product line theories. In: Nicola, R. D. (ed.) ESOP. *Springer-Verlag Lecture Notes in Computer Science* **4421** 64–79.

Milner, R. (1980) A Calculus of Communicating Systems. *Springer-Verlag Lecture Notes in Computer Science* **92**.

Milner, R. (1983) Calculi for synchrony and asynchrony. *Theoretical Computer Science* **25** 267–310.

Park, D. M. R. (1981) Concurrency and automata on infinite sequences. In: Deussen, P. (ed.) Proceedings of the 5th GI-Conference on Theoretical Computer Science. *Springer-Verlag Lecture Notes in Computer Science* **104** 167–183.

Pierce, B. C. and Sangiorgi, D. (2000) Behavioral equivalence in the polymorphic pi-calculus. *Journal of the ACM* **47** 531–584.

Plotkin, G. (1981) A structural approach to operational semantics. FN 19, DAIMI. Computer Science Department, Aarhus University, Denmark.

Raclet, J.-B. (2008) Residual for component specifications. In: Proceedings of the 4th International Workshop on Formal Aspects of Component Software (FACS'07). *Electronic Notes in Theoretical Computer Science* **215** 93–110.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press

Stirling, C. (1987) Modal logics for communicating systems. *Theoretical Computer Science* **49** 311–347.

Stirling, C. and Walker, D. (1989) CCS, liveness, and local model checking in the linear time mu-calculus. In: Sifakis, J. (ed.) Automatic Verification Methods for Finite State Systems, Proceedings International Workshop, Grenoble, France. *Springer-Verlag Lecture Notes in Computer Science* **407** 166–178.

Stoy, J. (1977) *Denotational Semantics: The Scott–Strachey Approach to Progamming Language Theory*, MIT Press.

van Glabbeek, R. J. (1990) The linear time-branching time spectrum (extended abstract). In: Baeten, J. C. M. and Klop, J. W. (eds.) CONCUR. *Springer-Verlag Lecture Notes in Computer Science* **458** 278–297.

https://doi.org/10.1017/S0960129511000697 Published online by Cambridge University Press