The One-Selecting Variant of Disjunctive Modal Transition Systems

Studienarbeit

Heiko Schmidt

Rankestr. 3
24118 Kiel
Mat.-Nr.: 484590

`hsc@informatik.uni-kiel.de`

Lehrstuhl Softwaretechnologie
Institut für Informatik und Praktische Mathematik
Christian-Albrechts-Universität zu Kiel
Betreuer: Dr. Harald Fecher

Abgegeben am: 27.06.2005

The One-Selecting Variant of Disjunctive Modal Transition Systems

Heiko Schmidt

Christian-Albrechts-University Kiel, Germany
hsc@informatik.uni-kiel.de

Abstract

Disjunctive modal transition systems (DMTS) describe classes of transition systems (TS) via a simulation relation. Consequently, they are an appropriate formalism for expressing underspecification of systems. An alternative definition of simulation on DMTSs is presented. It is shown that the class of TSs described by a given DMTS with respect to the original simulation can also be expressed by a DMTS with respect to the alternative simulation (where the latter DMTS has at most twice as many states).

1 Introduction

1.1 Process Semantics

This work deals with the specification and underspecification of concurrent systems. More exactly, we describe the observable behaviour of a system rather than the system itself. The behaviour of a system is called *process*. We consider processes that perform some kind of *actions*, i.e. steps of observable behaviour. It remains unspecified what kind of system we describe by defining a process and what kind of actions this system performs. For example, the system might be a program with actions being variable assignments or it might be an agent in a reactive system with actions being communications between agents.

It is a common approach to describe processes using so called *labelled transition systems* [8]. We will henceforth simply call them transition systems and abbreviate them by TS. A transition system is a directed graph with a designated root node and labelled edges. The nodes are identified with the states of the process. Edges stand for transitions of the system: An edge from state s to state s' having label a, shortly written $s \xrightarrow{a} s'$, means that if a process is in state s and an action a occurs, it will evolve into state s'.

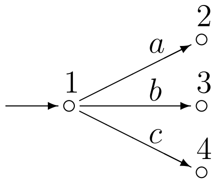

Fig. 1. Transition System $\mathcal{T}$

Non-determinism is allowed, i.e. there may be states s, s', $\bar{s}'$ and a label a such that $s \xrightarrow{a} s'$ and $s \xrightarrow{a} \bar{s}'$. In this case, an action a can lead to state s' or $\bar{s}'$. We do not specify how this decision is made.

Figure 1 shows a simple example of a transition system. Usually there exist many transition systems that describe the same process, i.e. the same system behaviour in terms of actions. Thus in order to specify processes using transition systems, there is the need for an equivalence relation that defines which transition systems are to be identified. Then processes can be seen as equivalence classes of transition systems with regard to some chosen equivalence relation. As presented in [8], various equivalences have been established. The most common relations include *trace equivalence* and *bisimulation equivalence*.

Trace equivalence [9,8] is defined as follows: The *trace set* of a transition system is the set of possible sequences of labels that are passed when starting in the root state and repeatedly following possible transitions. Then two transition systems are called *trace equivalent*, if they have equal trace sets.

Trace equivalence is not sufficient if the branching structure is of importance. Thus in most cases one prefers "finer" relations, e.g. the widely-used bisimulation equivalence [18,8]: A bisimulation between two transition systems $\mathcal{T}_1$ and $\mathcal{T}_2$ is a relation between the two state sets such that the root states are related and the following property holds: For every transition $s_1 \xrightarrow{a} s_1'$ in $\mathcal{T}_1$ and every s_2 related to s_1, there exists some s_2' related to s_1' such that $s_2 \xrightarrow{a} s_2'$, and vice verca, for every transition $s_2 \xrightarrow{a} s_2'$ in $\mathcal{T}_2$ and every s_1 related to s_2, there exists some s_1' related to s_2' such that $s_1 \xrightarrow{a} s_1'$. Two transition systems are called *bisimilar* (or *bisimulation equivalent*) if there exists a bisimulation.

1.2 The Need for Underspecification

We have seen how a process can be specified using a transition system and an equivalence relation. However, in many applications it is useful to *underspecify*, i.e. to not specify the system in every detail, but instead leave parts of the system open to allow different implementations. For example, when developing a system in top-down manner, one starts with a general framework of the system, leaving many parts open that are to be implemented later. This framework can be described using underspecification.

Another reason for having holes in the specification might be that the left out parts are simply not of interest for the task one wants to achieve. For example, in *model checking*, it is checked whether a formula in some modal logic holds for a given system. The validity of the formula might not be depending on some parts of the system, which thus can be left open. The given, fully specified system can be turned into an abstract, underspecified system with less states, which can be model checked more efficiently. This technique is called *abstraction*.

As a further application, underspecification can be useful if processes are used as semantics domain. For example, modeling languages, like UML [20,21], are themselves underspecified, as they do not specify every detail of a program. Thus underspecification is suitable for defining the semantics of such modeling languages.

1.3 Underspecification Techniques

If we want to underspecify, we need to find formalisms that do not describe a single process, but rather a class of processes, namely those that are supposed to be implementations of the underspecification. It is possible, for example, to use a formula in some modal logic as specification and regard the class of transition systems satisfying the formula as implementations. Although this is useful in some applications, others demand more operational based approaches. These use transition systems and non-symmetric relations similar to trace and bisimulation equivalence, which have already been introduced in Section 1.1 as formalisms for (full) specification. In the following, we summarize some techniques used for underspecification.

1.3.1 Underspecification via Trace Inclusion

For the trace set approach, one can regard *trace inclusion*. One can define some (concrete) transition system to be an implementation of an (abstract) transition system, if the trace set of the abstract system is a subset of the trace set of the concrete system (or vice versa, i.e. if the trace set of the concrete system is a subset of the abstract system). Using one of these techniques, it is possible to express that an implementation may allow either more or less behaviour than the specification. However, this approach does not satisfy, as it is not possible to allow combinations of this, i.e. allow more behaviour in one part of the system and less behaviour in another one.

1.3.2 Underspecification via Simulation

An adaption of trace inclusion that takes the branching structure into account is obtained via *simulations* [17]. A transition system $\mathcal{T}_1$ (safety-)simulates another transition system $\mathcal{T}_2$, if there exists a relation between the state sets of $\mathcal{T}_1$ and $\mathcal{T}_2$ such that the root states are related and the following property

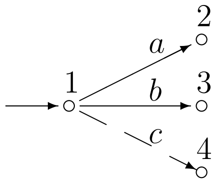

Fig. 2. A Transition System with Must and May Transitions

holds: For every transition $s_1 \xrightarrow{a} s_1'$ in $\mathcal{T}_1$ and every s_2 related to s_1, there exists some s_2' related to s_1' such that $s_2 \xrightarrow{a} s_2'$. In this case, $\mathcal{T}_2$ reverse-simulates (or liveness-simulates) $\mathcal{T}_1$. Now underspecification can be introduced using one of these two notions: One can define some (concrete) transition system $\mathcal{T}_1$ to be an implementation of an (abstract) transition system $\mathcal{T}_2$, if $\mathcal{T}_1$ simulates (respectively reverse-simulates) $\mathcal{T}_2$. However, similar to the trace set approach, we have the problem of not being able to define combinations of safety and liveness properties, i.e. all behaviour can be neglected when the safety-simulation approach is used, respectively arbitrary behaviour can be added if the liveness-simulation approach is used. For example, it is not possible to express the class of all transition systems, where at the beginning actions a and b have to be possible, action c is allowed but no further actions are allowed at the beginning.

1.3.3 Underspecification via Must and May Transitions

In order to address this problem, one can use a modification of transition systems that features two kinds of transitions, one transition relation to denote the steps that are mandatory for the implementation, called *must transitions*, and the other to indicate those steps which may occur, but are not necessary for the implementation, called *may transitions*. This approach was followed independently by Larsen and Thomsen, who introduced *modal transition systems* [15], and Dams, who called them *mixed transition systems* [4,5]. We can solve the problem mentioned at the end of section 1.3.2 by turning the transition labelled with c into a may transition (Figure 2). Graphically, we represent must transitions as solid arrows, whereas may transitions are drawn as dashed arrows.

1.3.4 Underspecification via Disjunctive Modal Transition Systems

However, the approach with must and may transitions is still not sufficient to model all behaviours that appear in practice. Reconsider transition system $\mathcal{T}$ from Figure 1. Three transitions were possible in the initial state. One might like to express that two different implementations allow two different successor states after performing an action a. However, an action a shall be possible in every implementation. Thus we cannot use two may transitions to the two

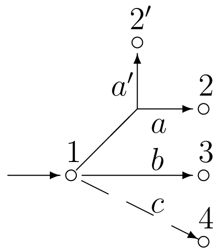

Fig. 3. Disjunctive Modal Transition System $\mathcal{U}$

states, because this would also allow an implementation with no action a at all. Two must transitions would not serve either, because they would mean that both must be possible in *every* implementation. A solution to this problem is a further extension of transition systems. These are called *disjunctive modal transition systems (DMTS)* [16]. They allow *hypertransitions*, i.e. transitions starting in a single state, but ending in sets of states. Graphically, we represent hypertransitions by dividing the head of an arrow such that it points to all target states. For different targets of a single hypertransition we also allow different labels, thus we draw them behind the division. Figure 3 shows an example of a DMTS, named $\mathcal{U}$. It expresses that every implementation has to provide an action b, may provide an action c, and must provide one of the actions a or a'.[1]

The class of implementations of a DMTS is defined via a simulation relation that relates abstract DMTSs and concrete transition systems. The commonly used relation is called *disjunctive modal simulation*. For every hypertransition it demands that at least one of the targets can be reached in every implementation. Thus it also allows that an implementation can contain transitions to more than one target of the hypertransition.

1.4 Underspecification via 1-Selecting Modal Transition Systems

In this work, we introduce a new simulation relation for disjunctive modal transition systems, called *1-selecting modal simulation*. It differs from disjunctive modal simulation only in the way hypertransitions are interpreted. In contrast to the disjunctive approach, 1-selecting modal simulation demands that for every hypertransition *exactly one* of its heads corresponds to a transition in the implementation. Thus example $\mathcal{U}$ in Figure 3 does not allow an implementation in which both actions a and a' can be performed, if we use it

[1] In fact, this could have also been expressed with the must and may transition approach presented in section 1.3.3, if we allow more than one root state. One could use a transition system with two components that both look like the one in Figure 2, where in one of them action a is replaced by a'. However, such a reduction is no longer possible if, for example, we add to Figure 3 a transition from state $2'$ to state 1.

in the context of 1-selecting modal simulation. In the context of disjunctive modal simulation, this is allowed. To differentiate between the two interpretations, DMTSs used in the context of 1-selecting modal simulation will be called *1-selecting modal transition systems (1MTS)*.

We compare the expressive power of these two approaches. The hope is to gain more expressiveness using the new approach. By expressiveness we mean the class of classes [2] of transition systems that can be described by a disjunctive, repectively 1-selecting modal transition system. One formalism is more expressive than another, if the expressiveness of the first is a proper superclass of the second. In a first step, this work gives the partial answer that 1MTSs are at least as expressive as the common DMTSs by showing that the class of TSs described by a given DMTSs can also be expressed by a 1MTS that has at most twice as many states. It is future work to check whether the 1-selecting approach is even more expressive than the disjunctive approach.

1.5 Outline

The outline of this work is as follows: Section 2 introduces transition systems together with the concept of bisimulation. In Section 3, DMTSs are introduced together with the two considered simulation relations. Subsection 3.1 defines the syntax of DMTSs. Subsections 3.2 and 3.3 define disjunctive modal simulation, respectively 1-selecting simulation. In Subsection 3.4, we show that the two simulation notions for DMTSs harmonize with the common bisimulation notion for transition systems. Our definition of disjunctive modal simulation is a slight variation of the one proposed by Larsen and Xinxin in [16]. Subsection 3.5 justifies this variation by showing that it does not change the expressiveness. In Section 4, we present the proof that 1-selecting modal simulation is at least as expressive as disjunctive modal simulation. Related work is discussed in Section 5 and the conclusion together with future work is presented in Section 6.

2 Transition Systems

Transition systems are directed graphs, where edges are labelled and we have a designated root state. Formally:

Definition 2.1 (TS) *A* transition system *(TS) is a tuple* $(S, L, \longrightarrow, s^0)$*, where* S *is a set of states,* L *is a set of labels,* $\longrightarrow \subseteq S \times L \times S$ *is the* transition

[2] Readers with set theoretical expertise might complain, that a class of classes does not exist, as long as at least one of the classes to be put into the collection is a proper class, which is the case here. We ignore this problem, because otherwise we could not define terms like expressiveness. It would have been possible to present all results in this work without giving definitions that require "classes of classes", but it would have been less comfortable to read.

relation *and* s^0 *is the* root state.

We denote the class of all TSs by $\mathbb{TS}$. Instead of $(s, a, s') \in\longrightarrow$, we usually write $s \xrightarrow{a} s'$. The most common equivalence notion on transition systems is bisimulation, defined as follows:

Definition 2.2 (bisimulation) *For* $i \in \{1, 2\}$*, let* $\mathcal{T}_i = (S_i, L, \longrightarrow_i, s_i^0) \in \mathbb{TS}$*. A* bisimulation *between* $\mathcal{T}_1$ *and* $\mathcal{T}_2$ *is a relation* $R \subseteq S_1 \times S_2$ *such that the following properties hold:*

- $(s_1^0, s_2^0) \in R$
- *For all* $(s_1, s_2) \in R$ *and* $a \in L$ *we have*
  - $s_1 \xrightarrow{a}_1 s_1' \Rightarrow \exists s_2' \in S_2 : s_2 \xrightarrow{a}_2 s_2' \wedge (s_1', s_2') \in R$
  - $s_2 \xrightarrow{a}_2 s_2' \Rightarrow \exists s_1' \in S_1 : s_1 \xrightarrow{a}_1 s_1' \wedge (s_1', s_2') \in R$

$\mathcal{T}_1$ *and* $\mathcal{T}_2$ *are called* bisimilar *if there exists a bisimulation between them. In that case, we write* $\mathcal{T}_1 \sim \mathcal{T}_2$*.*

3 Disjunctive Modal Transition Systems

We will now introduce disjunctive modal transition systems together with two different notions of simulation that formalize, whether a TS is an implementation of a disjunctive modal transition system.

3.1 Syntax

Disjunctive modal transition systems extend TSs. Instead of a single root state, they allow a set of root states. They feature two types of transitions: must and may transitions. We demand that every must transition must also appear as may transition, but there may be may transitions that do not appear as must transitions. This requirement seems reasonable, transitions required need to be allowed. Both must and may transitions may be hypertransitions, i.e. have more than one successor state as target. If every transition has only one target, the must transitions are exactly the may transitions and the root state set contains only one element, we call the system *fully determined.* The formal definition is as follows:

Definition 3.1 (DMTS) *A* disjunctive modal transition system *(DMTS) is a tuple* $(U, L, \longmapsto, \dashrightarrow, U^0)$*, where* U *is a set of states,* L *is a set of labels,* $\longmapsto \subseteq U \times \mathcal{P}(L \times U)$ *is the* must transition relation*,* $\dashrightarrow \subseteq U \times \mathcal{P}(L \times U)$ *is the* may transition relation*, and* U^0 *is the set of* root states*, satisfying the condition* $\longmapsto \subseteq \dashrightarrow$*. It is* fully determined *iff* $|U^0| = 1$*,* $\longmapsto = \dashrightarrow$ *and* $\forall (u, M) \in\dashrightarrow$: $|M| = 1$.

We denote the class of all DMTSs by $\mathbb{DMTS}$. Instead of $(u, M) \in\longmapsto$ (respectively $(u, M) \in\dashrightarrow$), we usually write $u \longmapsto M$ (respectively $u \dashrightarrow$

M). Furthermore, for all $u \in U$, we define $\mathcal{M}_u \stackrel{\text{def}}{=} \{M \mid u \longmapsto M\}$ and $\mathcal{N}_u \stackrel{\text{def}}{=} \{M \mid u \dashrightarrow M\}$. Then obviously $\mathcal{M}_u \subseteq \mathcal{N}_u$ holds.

Figure 3 shows an example of a DMTS. In the graphical representation, hypertransitions are drawn as arrows having several heads, with every head having its own label. Must transitions are represented by solid arrows, whereas may transitions are drawn as dashed arrows. We do not draw may transitions if they also exist as must transitions. Due to the requirement $\longmapsto \subseteq \dashrightarrow$, we know that such may transitions always exist implicitly.

The class of all fully determined DMTSs corresponds to the class of all TSs:

Proposition 3.2 *Define*

$$\pi : \mathbb{TS} \to \{\mathcal{U} \in \mathbb{DMTS} \mid \mathcal{U} \textit{ is fully determined }\};$$
$$(S, L, \longrightarrow, s^0) \mapsto (S, L, \dashrightarrow, \dashrightarrow, \{s^0\}),$$

where $\dashrightarrow \stackrel{\text{def}}{=} \{(s, \{(a, s')\}) \mid s \stackrel{a}{\longrightarrow} s'\}$. *Then* π *is an isomorphism.*

Proof. Obvious. □

3.2 Disjunctive Modal Simulation

We define disjunctive modal simulation, the common simulation relation used with DMTSs. Similar to the bisimulation notion on TSs, a disjunctive modal simulation is a relation relating states in the TS with states in the DMTS, such that several properties are satisfied: The TS needs to have a root state corresponding to a state in the root state set of the DMTS. Furthermore, every state s' in the TS that can be reached by an action a needs to have an equivalent counterpart u' in the DMTS that is not forbidden to be reached with an action a. On the other hand, for every must transition to a set of targets M in the DMTS, there must be at least one $(a, u') \in M$ that has a corresponding step in the TS.

A TS *disjunctively simulates* a given DMTS, if there exists a disjunctive modal simulation between them. We can define some TS $\mathcal{T}$ to be an implementation of an DMTS $\mathcal{U}$, if $\mathcal{T}$ disjunctively simulates $\mathcal{U}$. The formal definition is as follows:

Definition 3.3 (disjunctive modal simulation) *Let* $\mathcal{T} = (S, L, \longrightarrow, s^0) \in \mathbb{TS}$ *and* $\mathcal{U} = (U, L, \longmapsto, \dashrightarrow, U^0) \in \mathbb{DMTS}$. *A* disjunctive modal simulation *between* $\mathcal{T}$ *and* $\mathcal{U}$ *is a relation* $R \subseteq S \times U$ *such that the following properties hold:*

- $\exists u \in U^0 : (s^0, u) \in R$
- *For all* $(s, u) \in R$ *we have*
  - $\forall a \in L, s' \in S : s \stackrel{a}{\longrightarrow} s' \Rightarrow \exists M \in \mathcal{N}_u, u' \in U : (a, u') \in M \wedge (s', u') \in R$
  - $\forall M \in \mathcal{M}_u : \exists (a, u') \in M, s' \in S : s \stackrel{a}{\longrightarrow} s' \wedge (s', u') \in R$

$\mathcal{T}$ disjunctively simulates $\mathcal{U}$ *if there exists a disjunctive modal simulation between them. In that case, we write* $\mathcal{T} \prec_{\text{DMTS}} \mathcal{U}$*. The expressiveness of disjunctive modal simulation on DMTSs is*

$$E_{\text{DMTS}} \stackrel{\text{def}}{=} \{\mathcal{S} \subseteq \mathbb{TS} \mid \exists \mathcal{U} \in \mathbb{DMTS} : \mathcal{S} = \{\mathcal{T} \in \mathbb{TS} \mid \mathcal{T} \prec_{\text{DMTS}} \mathcal{U}\}\}.$$

We will call a TS $\mathcal{T}$ an *implementation* of a DMTS $\mathcal{U}$, if $\mathcal{T}$ disjunctively simulates $\mathcal{U}$. A DMTS is *implementable*, if there exists an implementation for it. An example of a non-implementable DMTS is

$$\mathcal{U}^{-} \stackrel{\text{def}}{=} (\{1\}, \emptyset, \{(1, \emptyset)\}, \{(1, \emptyset)\}, \{1\}).$$

A disjunctive modal simulation must relate the start state of the TS with 1. Then, since $1 \longmapsto \emptyset$ is a must transition, one target in the empty set needs to have a corresponding transition in the TS, which is certainly not possible.

3.3 1-Selecting Modal Simulation

We will now introduce a new simulation notion for DMTSs called 1-selecting modal simulation. The concept of 1-selecting modal simulation is similar to the one of disjunctive modal simulation. Again, we have a simulation relation that is required to satisfy several properties: As for the disjunctive approach, an implementing TS needs to have a root state corresponding to a state in the root state set of the DMTS. The interpretation of hypertransitions differs from disjunctive modal simulation: One needs to choose functions γ that pick for each hypertransition one of its heads. Then every state s' in the TS that can be reached by an action a needs to have an equivalent counterpart $\hat{u}'$ that has been chosen by γ and is not forbidden to be reached with an action a. On the other hand, for every chosen must transition target reachable with action a, there must be a corresponding state in the TS reachable with a.

A TS *1-selecting simulates* a given DMTS, if there exists a 1-selecting modal simulation between them. Thus we have an alternative way of interpreting DMTSs: One can define some TS $\mathcal{T}$ to be an implementation of an DMTS $\mathcal{U}$, if $\mathcal{T}$ 1-selecting simulates $\mathcal{U}$. If used in this way, we will henceforth call the system 1MTS. We denote by $\mathbb{1MTS}$ the class of all 1MTSs (thus syntactically $\mathbb{1MTS} = \mathbb{DMTS}$). We will continue to use the term DMTS if the system is used in the context of disjunctive modal simulation. The formal definition of 1-selecting modal simulation is as follows:

Definition 3.4 (1-selecting modal simulation) *Let* $\mathcal{T} = (S, L, \longrightarrow, s^0) \in \mathbb{TS}$ *and* $\hat{\mathcal{U}} = (\hat{U}, L, \stackrel{\wedge}{\longmapsto}, \stackrel{\wedge}{\dashrightarrow}, \hat{U}^0) \in \mathbb{1MTS}$*. An* 1-selecting modal simulation *between* $\mathcal{T}$ *and* $\hat{\mathcal{U}}$ *is a relation* $\hat{R} \subseteq S \times \hat{U}$ *such that the following properties hold:*

- $\exists \hat{u} \in \hat{U}^0 : (s^0, \hat{u}) \in \hat{R}$
- *For all* $(s, \hat{u}) \in \hat{R}$ *there exists a function* $\gamma : \mathcal{N}_{\hat{u}} \to L \times \hat{U}$ *such that*
  · $\forall \hat{M} \in \mathcal{N}_{\hat{u}} : \gamma(\hat{M}) \in \hat{M}$

- $\forall a \in L, s' \in S : s \xrightarrow{a} s' \Rightarrow \exists \hat{u}' \in \hat{U} : (a, \hat{u}') \in \gamma(\mathcal{N}_{\hat{u}}) \wedge (s', \hat{u}') \in \hat{R}$
- $\forall (a, \hat{u}') \in \gamma(\mathcal{M}_{\hat{u}}) : \exists s' \in S : s \xrightarrow{a} s' \wedge (s', \hat{u}') \in \hat{R}$

$\mathcal{T}$ 1-selecting simulates $\hat{\mathcal{U}}$ *if there exists a 1-selecting modal simulation between them. In that case, we write* $\mathcal{T} \prec_{1\text{MTS}} \hat{\mathcal{U}}$. *The expressiveness of 1-selecting modal simulation on 1MTSs is*

$$E_{1\text{MTS}} \stackrel{\text{def}}{=} \{\mathcal{S} \subseteq \mathbb{TS} \mid \exists \hat{\mathcal{U}} \in \mathbb{1MTS} : \mathcal{S} = \{\mathcal{T} \in \mathbb{TS} \mid \mathcal{T} \prec_{1\text{MTS}} \hat{\mathcal{U}}\}\}.$$

We will call a TS $\mathcal{T}$ an *implementation* of an 1MTS $\hat{\mathcal{U}}$, if $\mathcal{T}$ 1-selecting simulates $\hat{\mathcal{U}}$. An 1MTS is *implementable*, if there exists an implementation for it. Reconsider $\mathcal{U}^-$, the example of a non-implementable DMTS from Subsection 3.2. $\mathcal{U}^-$ is also an non-implementable 1MTS. In fact, for an 1MTS, it is enough to have a may transition instead of a must transition, i.e.

$$\hat{\mathcal{U}}^- \stackrel{\text{def}}{=} (\{1\}, \emptyset, \emptyset, \{(1, \emptyset)\}, \{1\})$$

is not implementable if understood as 1MTS, but it is implementable if understood as DMTS. In the latter case, the TS $(\{1'\}, \emptyset, \emptyset, 1')$ is an implementation. It is not an implementation for the 1MTS, because one cannot find a function γ for $(1', 1) \in \hat{R}$ satisfying $\forall \hat{M} \in \{\emptyset\} : \gamma(\hat{M}) \in \hat{M}$.

One can characterize the difference between the two simulation notions as follows: 1-selecting modal simulation demands that for every must hypertransition there exists *exactly one* head of the hypertransition that corresponds with a transition in the implementation. For disjunctive modal simulation, we require *at least one* head of every must hypertransition to have a corresponding transition in the implementation. One the other hand, 1-selecting modal simulation demands that for every may hypertransition there may exist *at most one* head of the hypertransition that corresponds with a transition in the implementation. For disjunctive modal simulation, we allow *arbitrarily many* heads of a may hypertransition to have corresponding transitions in the implementation.

Thus, for 1-selecting modal simulation, one has to choose a particular head for every hypertransition that may, respectively must be implemented. This is formalized by functions $\gamma : \mathcal{N}_{\hat{u}} \to L \times \hat{U}$, which for every outgoing transition choose one of its targets. If $\mathcal{N}_{\hat{u}} = \emptyset$, the empty function satisfies all our requirements.

Note however, that the above characterization does not imply that two must hypertransitions will always have two corresponding transitions in the implementation. For example, TS $\mathcal{T}_1$ is an implementation for 1MTS $\hat{\mathcal{U}}_1$ (see Figure 4). Every must hypertransition has exactly one target that corresponds to a transition in the implementation. For both hypertransitions, it is the one leading to state 3. This leads to a single transition in the implementation.

Fig. 4. 1MTS $\hat{\mathcal{U}}_1$ (left) and TS $\mathcal{T}_1$ (right)

3.4 Compatibility of Bisimulation and Modal Simulations

The following proposition states that disjunctive modal simulation and 1-selecting modal simulation are extensions of bisimulation, i.e. for fully determined DMTSs, which correspond to TSs, the three notions coincide.

Proposition 3.5 *Let $\mathcal{T}_1, \mathcal{T}_2 \in \mathbb{TS}$ and $\mathcal{U} \stackrel{\text{def}}{=} \pi(\mathcal{T}_2)$, where π is the isomorphism from proposition 3.2. Then the following statements are equivalent:*

(i) $\mathcal{T}_1 \sim \mathcal{T}_2$

(ii) $\mathcal{T}_1 \prec_{\text{DMTS}} \mathcal{U}$

(iii) $\mathcal{T}_1 \prec_{\text{1MTS}} \mathcal{U}$

Proof. (i) $\Rightarrow$ (ii): It is easily checked that a given bisimulation between $\mathcal{T}_1$ and $\mathcal{T}_2$ is also a disjunctive modal simulation between $\mathcal{T}_1$ and $\mathcal{U}$.

(ii) $\Rightarrow$ (iii): Any given disjunctive modal simulation between $\mathcal{T}_1$ and $\mathcal{U}$ is also a 1-selecting modal simulation between $\mathcal{T}_1$ and $\mathcal{U}$. The choices for functions γ is trivial, because $\mathcal{U}$ is fully determined and thus there is only one possible function for each hypertransition.

(iii) $\Rightarrow$ (i): Again, it is easily checked that a given 1-selecting modal simulation between $\mathcal{T}_1$ and $\mathcal{U}$ is also a bisimulation between $\mathcal{T}_1$ and $\mathcal{T}_2$. □

Furthermore, disjunctive modal simulation and 1-selecting modal simulation are closed under bisimilarity:

Proposition 3.6 *Let $\mathcal{T}_1, \mathcal{T}_2 \in \mathbb{TS}$ and $\mathcal{U} \in \mathbb{DMTS}$.*

(i) $\mathcal{T}_1 \sim \mathcal{T}_2 \wedge \mathcal{T}_2 \prec_{\text{DMTS}} \mathcal{U} \Rightarrow \mathcal{T}_1 \prec_{\text{DMTS}} \mathcal{U}$.

(ii) $\mathcal{T}_1 \sim \mathcal{T}_2 \wedge \mathcal{T}_2 \prec_{\text{1MTS}} \mathcal{U} \Rightarrow \mathcal{T}_1 \prec_{\text{1MTS}} \mathcal{U}$.

Proof. Let $\bar{R}$ be a bisimulation between $\mathcal{T}_1$ and $\mathcal{T}_2$ and R be a disjunctive modal simulation (respectively 1-selecting modal simulation) between $\mathcal{T}_2$ and $\mathcal{U}$. Then it is straightforwardly checked that $\bar{R} \circ R = \{(s_1, u) \mid \exists s_2 : (s_1, s_2) \in \bar{R} \wedge (s_2, u) \in R\}$ is a disjunctive modal simulation (respectively 1-selecting modal simulation) between $\mathcal{T}_1$ and $\mathcal{U}$. □

3.5 Disjunctive Modal Transition Systems with Single Target May Transitions

Our definition of DMTSs allows must and may hypertransitions, i.e. both must and may transitions are allowed to point to sets of states. However, the approach followed by Larsen and Xinxin in [16] is to have must hypertransitions, but use ordinary transitions with single targets for may steps. Thus a disjunctive modal transition system of that kind is defined as follows:

Definition 3.7 ($\overline{\textsf{DMTS}}$) *A $\overline{\textsf{DMTS}}$ is a tuple $(U, L, \longmapsto, \dashrightarrow, U^0)$, where U is a set of states, L is a set of labels, $\longmapsto \subseteq U \times \mathcal{P}(L \times U)$ is the* must transition relation*, $\dashrightarrow \subseteq U \times L \times U$ is the* may transition relation*, and U^0 is the set of* root states*, satisfying the condition*

$$\forall u \in U, M \subseteq L \times U : u \longmapsto M \Rightarrow \forall (a, u') \in M : u \overset{a}{\dashrightarrow} u'.$$

We denote the class of all $\overline{\textsf{DMTS}}$s by $\overline{\mathbb{DMTS}}$. In $\overline{\textsf{DMTS}}$s, $\dashrightarrow$ is a subset of $U \times L \times U$ instead of $U \times \mathcal{P}(L \times U)$. As a consequence, the requirement $\longmapsto \subseteq \dashrightarrow$ in DMTSs must be reformulated for $\overline{\textsf{DMTS}}$s. The condition looks more complicated, but simply expresses that every target in a must hypertransition must be allowed to be reached. Furthermore, the definition of $\mathcal{N}_u$ takes a new form: $\mathcal{N}_u \overset{\text{def}}{=} \{\{(a, u')\} \mid u \overset{a}{\dashrightarrow} u'\}$, whereas the notation $\mathcal{M}_u$ will be used as for DMTSs. Using these meanings for $\mathcal{M}_u$ and $\mathcal{N}_u$ in the context of $\overline{\textsf{DMTS}}$s, the definition of disjunctive modal simulation can also be applied to $\overline{\textsf{DMTS}}$s instead of DMTSs. The expressiveness of disjunctive modal simulation on $\overline{\textsf{DMTS}}$s is

$$E_{\overline{\text{DMTS}}} \overset{\text{def}}{=} \{\mathcal{S} \subseteq \mathbb{TS} \mid \exists \overline{\mathcal{U}} \in \overline{\mathbb{DMTS}} : \mathcal{S} = \{\mathcal{T} \in \mathbb{TS} \mid \mathcal{T} \prec_{\text{DMTS}} \overline{\mathcal{U}}\}\}.$$

It turns out that in the context of disjunctive modal simulation DMTSs and $\overline{\textsf{DMTS}}$s have the same expressive power, thus can be used interchangeably:

Proposition 3.8 $E_{\text{DMTS}} = E_{\overline{\text{DMTS}}}$.

Proof. We show $E_{\text{DMTS}} \subseteq E_{\overline{\text{DMTS}}}$ and $E_{\text{DMTS}} \supseteq E_{\overline{\text{DMTS}}}$.

- Let $\mathcal{S} \subseteq \mathbb{TS}$ and $\mathcal{U} = (U, L, \longmapsto, \dashrightarrow, U^0) \in \mathbb{DMTS}$ such that $\mathcal{S} = \{\mathcal{T} \in \mathbb{TS} \mid \mathcal{T} \prec_{\text{DMTS}} \mathcal{U}\}$. We need to find $\overline{\mathcal{U}} \in \overline{\mathbb{DMTS}}$ such that $\mathcal{S} = \overline{\mathcal{S}} \overset{\text{def}}{=} \{\mathcal{T} \in \mathbb{TS} \mid \mathcal{T} \prec_{\text{DMTS}} \overline{\mathcal{U}}\}$. Define $\overline{\mathcal{U}} \overset{\text{def}}{=} (U, L, \longmapsto, \dashrightarrow', U^0)$, where $u \overset{a}{\dashrightarrow}{}' u' \overset{\text{def}}{\Leftrightarrow} \exists M \subseteq L \times U : u \dashrightarrow M \wedge (a, u') \in M$. We show $\mathcal{S} \subseteq \overline{\mathcal{S}}$ and $\mathcal{S} \supseteq \overline{\mathcal{S}}$.
  - Let $\mathcal{T} \in \mathcal{S}$, i.e. $\mathcal{T} \prec_{\text{DMTS}} \mathcal{U}$. Given a disjunctive modal simulation R between $\mathcal{T}$ and $\mathcal{U}$, it is straightforwardly checked that R is also a disjunctive modal simulation between $\mathcal{T}$ and $\overline{\mathcal{U}}$.
  - Now let $\mathcal{T} \in \overline{\mathcal{S}}$, i.e. $\mathcal{T} \prec_{\text{DMTS}} \overline{\mathcal{U}}$. Given a disjunctive modal simulation R between $\mathcal{T}$ and $\overline{\mathcal{U}}$, it is straightforwardly checked that R is also a disjunctive modal simulation between $\mathcal{T}$ and $\mathcal{U}$.
- Let $\overline{\mathcal{S}} \subseteq \mathbb{TS}$ and $\overline{\mathcal{U}} = (U, L, \longmapsto, \dashrightarrow', U^0) \in \overline{\mathbb{DMTS}}$ such that $\overline{\mathcal{S}} = \{\mathcal{T} \in$

Fig. 5. 1MTS $\hat{\mathcal{U}}$ (left) and $\overline{\text{1MTS}}$ $\overline{\mathcal{U}}$ (right)

$\mathbb{TS} \mid \mathcal{T} \prec_{\text{DMTS}} \overline{\mathcal{U}}\}$. We need to find $\mathcal{U} \in \mathbb{DMTS}$ such that $\overline{\mathcal{S}} = \mathcal{S} \overset{\text{def}}{=} \{\mathcal{T} \in \mathbb{TS} \mid \mathcal{T} \prec_{\text{DMTS}} \mathcal{U}\}$. Define $\mathcal{U} \overset{\text{def}}{=} (U, L, \longmapsto, \dashrightarrow, U^0)$, where $u \dashrightarrow M \overset{\text{def}}{\Leftrightarrow} \exists (a, u') \in L \times U : u \overset{a}{\dashrightarrow}{}' u' \wedge M = \{(a, u')\}$. We show $\overline{\mathcal{S}} \subseteq \mathcal{S}$ and $\overline{\mathcal{S}} \supseteq \mathcal{S}$.

- Let $\mathcal{T} \in \overline{\mathcal{S}}$, i.e. $\mathcal{T} \prec_{\text{DMTS}} \overline{\mathcal{U}}$. Given a disjunctive modal simulation R between $\mathcal{T}$ and $\overline{\mathcal{U}}$, it is straightforwardly checked that R is also a disjunctive modal simulation between $\mathcal{T}$ and $\mathcal{U}$.
- Now let $\mathcal{T} \in \mathcal{S}$, i.e. $\mathcal{T} \prec_{\text{DMTS}} \mathcal{U}$. Given a disjunctive modal simulation R between $\mathcal{T}$ and $\mathcal{U}$, it is straightforwardly checked that R is also a disjunctive modal simulation between $\mathcal{T}$ and $\overline{\mathcal{U}}$. □

We have seen that for DMTSs there would be no need to allow subsets as targets of may transitions, i.e. one could also use $\overline{\text{DMTS}}$s without losing expressiveness. However, it is not clear if we can restrict ourselves to single target may transitions in 1MTSs, i.e. whether $E_{\text{1MTS}} = E_{\overline{\text{1MTS}}}$ holds, where $\overline{\mathbb{1MTS}} \overset{\text{def}}{=} \overline{\mathbb{DMTS}}$ and the expressiveness of 1-selecting modal simulation on $\overline{\text{1MTS}}$s is defined as

$$E_{\overline{\text{1MTS}}} \overset{\text{def}}{=} \{\mathcal{S} \subseteq \mathbb{TS} \mid \exists \overline{\mathcal{U}} \in \overline{\mathbb{1MTS}} : \mathcal{S} = \{\mathcal{T} \in \mathbb{TS} \mid \mathcal{T} \prec_{\text{1MTS}} \overline{\mathcal{U}}\}\}.$$

At least a straightforward approach for a reduction from 1MTS to $\overline{\text{1MTS}}$ does not succeed: Consider the 1MTS $\hat{\mathcal{U}}$ and the $\overline{\text{1MTS}}$ $\overline{\mathcal{U}}$ shown in Figure 5. $\hat{\mathcal{U}}$ allows as possible implementations those that either do a single a-step or a single b-step or no step at all. $\overline{\mathcal{U}}$ does not have the same class of implementations, because it would also allow an additional implementation in which both an a- and a b-step is possible.

4 Expressiveness

We show that 1MTSs are at least as expressive as DMTS:

Theorem 4.1 $E_{\text{DMTS}} \subseteq E_{\text{1MTS}}$.

Proof. Let $\mathcal{S} \subseteq \mathbb{TS}$ and $\mathcal{U} = (U, L, \longmapsto, \dashrightarrow, U^0) \in \mathbb{DMTS}$ such that $\mathcal{S} = \{\mathcal{T} \in \mathbb{TS} \mid \mathcal{T} \prec_{\text{DMTS}} \mathcal{U}\}$. We need to find $\hat{\mathcal{U}} \in \mathbb{1MTS}$ such that $\mathcal{S} = \hat{\mathcal{S}} \overset{\text{def}}{=} \{\mathcal{T} \in \mathbb{TS} \mid \mathcal{T} \prec_{\text{1MTS}} \hat{\mathcal{U}}\}$. Define $\hat{\mathcal{U}} = (\hat{U}, L, \overset{\wedge}{\longmapsto}, \overset{\wedge}{\dashrightarrow}, \hat{U}^0)$, where

$$\hat{U} \overset{\text{def}}{=} U \times \{0, 1\}, \text{ where we usually write } u_i \text{ instead of } (u, i),$$

$$u_i \hat{\longmapsto} \hat{M} \overset{\text{def}}{\Leftrightarrow} \exists M \in \mathcal{M}_u : M = \hat{M} = \emptyset \vee$$
$$\exists(\bar{a}, \bar{u'}) \in M : \hat{M} = \{(a, u'_0) \mid (a, u') \in M\} \cup \{(\bar{a}, \bar{u'}_1)\},$$
$$u_i \hat{\dashrightarrow} \hat{M} \overset{\text{def}}{\Leftrightarrow} u_i \hat{\longmapsto} \hat{M} \vee \exists M \in \mathcal{N}_u, (a, u') \in M : \hat{M} = \{(a, u'_0)\},$$
$$\hat{U}^0 \overset{\text{def}}{=} U^0 \times \{0, 1\}.$$

Note that this definition satisfies $\hat{\longmapsto} \subseteq \hat{\dashrightarrow}$. In order to show $\mathcal{S} = \hat{\mathcal{S}}$, we prove both inclusions and start with $\mathcal{S} \subseteq \hat{\mathcal{S}}$. Let $\mathcal{T} = (S, L, \longrightarrow, s^0) \in \mathcal{S}$, i.e. $\mathcal{T} \prec_{\text{DMTS}} \mathcal{U}$. We need to show $\mathcal{T} \prec_{\text{1MTS}} \hat{\mathcal{U}}$. There is a disjunctive modal simulation $R \subseteq S \times U$ between $\mathcal{T}$ and $\mathcal{U}$. Define $\hat{R} \subseteq S \times \hat{U}$:

$$s\hat{R}u_i \overset{\text{def}}{\Leftrightarrow} sRu$$

We prove that $\hat{R}$ is an 1-selecting modal simulation between $\mathcal{T}$ and $\hat{\mathcal{U}}$.

- Since R is a disjunctive modal simulation, we can choose $u \in U^0$ such that $s^0 Ru$. Hence $s^0 \hat{R} u_0$, as required.
- Let $(s, u_i) \in \hat{R}$, hence $(s, u) \in R$. We need to define $\gamma : \mathcal{N}_{u_i} \to L \times \hat{U}$. We first consider $\hat{M} \in \mathcal{M}_{u_i}$ and define $\gamma(\hat{M})$. The case $\hat{M} \in \mathcal{N}_{u_i} \setminus \mathcal{M}_{u_i}$ will be considered later. We know $\emptyset \notin \mathcal{M}_u$, because sRu implies (by means of the last property of a disjunctive modal simulation) the existence of some element in each $M \in \mathcal{M}_u$. Thus there exist $M \in \mathcal{M}_u, (\bar{a}, \bar{u'}) \in M$ such that $\hat{M} = \{(a, u'_0) \mid (a, u') \in M\} \cup \{(\bar{a}, \bar{u'}_1)\}$. Furthermore, since R is a disjunctive modal simulation, we can choose $(a, u') \in M$ such that there exists $s' \in S$ with $s \xrightarrow{a} s'$ and $s'Ru'$. Define:

  $$\gamma(\hat{M}) \overset{\text{def}}{=} \begin{cases} (\bar{a}, \bar{u'}_0) & \text{if } \exists \bar{s'} \in S : s \xrightarrow{\bar{a}} \bar{s'} \wedge \bar{s'} R \bar{u'}; \\ (a, u'_0) & \text{otherwise.} \end{cases}$$

  Now we consider the case $\hat{M} \in \mathcal{N}_{u_i} \setminus \mathcal{M}_{u_i}$. We can choose $M \in \mathcal{N}_u$ and $(a, u') \in M$ such that $\hat{M} = \{(a, u'_0)\}$. Define for $\hat{M} \in \mathcal{N}_{u_i} \setminus \mathcal{M}_{u_i}$:

  $$\gamma(\hat{M}) \overset{\text{def}}{=} (a, u'_0)$$

  We prove the three remaining properties of an 1-selecting modal simulation.
  - $\gamma(\hat{M}) \in \hat{M}$ for all $\hat{M} \in \mathcal{N}_{u_i}$ follows directly from the definition of γ.
  - Let $a \in L, s' \in S$ such that $s \xrightarrow{a} s'$. Since R is a disjunctive modal simulation, we can choose $M \in \mathcal{N}_u, u' \in U$ such that $(a, u') \in M$ and $s'Ru'$. By definition of $\hat{\dashrightarrow}$, we have $u_i \hat{\dashrightarrow} \{(a, u'_0)\}$. Certainly $\gamma(\{(a, u'_0)\}) = (a, u'_0)$. Hence $(a, u'_0) \in \gamma(\mathcal{N}_{u_i})$ and $s'Ru'$. By definition of $\hat{R}$, we have $s'\hat{R}u'_0$.
  - Let $(a, u'_i) \in \gamma(\mathcal{M}_{u_i})$. There exists $\hat{M} \in \mathcal{M}_{u_i}$ such that $\gamma(\hat{M}) = (a, u'_i)$. By definition of γ, there exists $s' \in S$ such that $s \xrightarrow{a} s'$ and $s'Ru'$. By definition of $\hat{R}$, we obtain $s'\hat{R}u'_i$.

This completes the proof of $\mathcal{S} \subseteq \hat{\mathcal{S}}$. It remains to show that $\mathcal{S} \supseteq \hat{\mathcal{S}}$. Let $\mathcal{T} = (S, L, \longrightarrow, s^0) \in \hat{\mathcal{S}}$, i.e. $\mathcal{T} \prec_{\text{1MTS}} \hat{\mathcal{U}}$. We need to show $\mathcal{T} \prec_{\text{DMTS}} \mathcal{U}$. There is a 1-selecting modal simulation $\hat{R} \subseteq S \times \hat{U}$ between $\mathcal{T}$ and $\hat{\mathcal{U}}$. Define $R \subseteq S \times U$:

$sRu \overset{\text{def}}{\Leftrightarrow} \exists i \in \{0,1\} : s\hat{R}u_i$

We prove that R is a disjunctive modal simulation between $\mathcal{T}$ and $\mathcal{U}$.

- Since $\hat{R}$ is an 1-selecting modal simulation, we can choose $u_i \in \hat{U}^0$ such that $s^0\hat{R}u_i$. Then s^0Ru, as required.
- Let $(s,u) \in R$. By definition of R, we can choose $i \in \{0,1\}$ such that $s\hat{R}u_i$. Choose a function $\gamma : \mathcal{N}_{u_i} \to L \times \hat{U}$ such that the properties of an 1-selecting modal simulation are satisfied.
  - Let $a \in L, s' \in S$ such that $s \xrightarrow{a} s'$. Since $\hat{R}$ is an 1-selecting modal simulation, we can choose $u'_i \in \hat{U}$ such that $(a,u'_i) \in \gamma(\mathcal{N}_{u_i})$ and $s'\hat{R}u'_i$. The latter implies $s'Ru'$, hence it remains to show $\exists M \in \mathcal{N}_u, u' \in U : (a,u') \in M$. Choose $\hat{M} \in \mathcal{N}_{u_i}$ such that $(a,u'_i) \in \hat{M}$. If $\hat{M} \in \mathcal{N}_{u_i} \setminus \mathcal{M}_{u_i}$, there exist $M \in \mathcal{N}_u$ and $(a,u') \in M$ such that $\hat{M} = \{(a,u'_0)\}$, as required. If $\hat{M} \in \mathcal{M}_{u_i}$, $\hat{M} \neq \emptyset$ implies the existence of $M \in \mathcal{M}_u$ and $(\bar{a},\bar{u'}) \in M$ such that $\hat{M} = \{(a,u'_0) \mid (a,u') \in M\} \cup \{(\bar{a},\bar{u'}_1)\}$. We have $u \longmapsto M$ and $(a,u') \in M$, where the latter is true, because in the case $i = 0$, we have $(a,u'_0) \in \{(a,u'_0) \mid (a,u') \in M\}$; and if $i = 1$, we have $(a,u'_1) = (\bar{a},\bar{u'_1})$, thus $(a,u') = (\bar{a},\bar{u'}) \in M$. By definition of a DMTS, $u \longmapsto M$ implies $u \dashrightarrow M$; thus we have $M \in \mathcal{N}_u$ and $(a,u') \in M$, as required.
  - Let $M \in \mathcal{M}_u$. We have $M \neq \emptyset$, because otherwise we would have $u_i \hat{\longmapsto} \emptyset$ by definition of $\hat{\longmapsto}$ and γ would satisfy $\gamma(\emptyset) \in \emptyset$, which is a contradiction. Consequently we can choose $(\bar{a},\bar{u'}) \in M$. By definition of $\hat{\longmapsto}$, we have $u_i \hat{\longmapsto} \hat{M}$, where $\hat{M} = \{(a,u'_0) \mid (a,u') \in M\} \cup \{(\bar{a},\bar{u'}_1)\}$. Let $(\bar{\bar{a}},\bar{\bar{u}}'_k) \overset{\text{def}}{=} \gamma(\hat{M})$. Then $(\bar{\bar{a}},\bar{\bar{u}}') \in M$ and, since $\hat{R}$ is an 1-selecting modal simulation, $\exists s' : s \xrightarrow{\bar{\bar{a}}} s' \wedge s'\hat{R}\bar{\bar{u}}'_k$. By definition of R, $s'\hat{R}\bar{\bar{u}}'_k$ implies $s'R\bar{\bar{u}}'$. $\square$

For any DMTS $\mathcal{U}$ we found an 1MTS $\hat{\mathcal{U}}$ such that $\mathcal{U}$ and $\hat{\mathcal{U}}$ have the same class of implementations. Our construction doubled the number of states. In most cases it is possible to introduce even less additional states. One only needs to guarantee that for every hypertransition with n targets in $\mathcal{U}$, at least $\lceil \log_2(n) \rceil$ of these targets appear with a component 1 in the state set of $\hat{\mathcal{U}}$. This is enough to build n different target sets for $\hat{\mathcal{U}}$ such that every set includes every target with a component 0 (plus certain targets with component 1, which make the sets distinct).

5 Related Work

The approach of extending transition systems by a second transition relation that expresses which steps *may* appear in an implementation was followed independently by Larsen and Thomsen, who introduced *modal transition systems* [15,14], and by Dams, who called them *mixed transition systems* [4,5]. Modal transition systems have been defined in this work. Mixed transition systems are modal transition systems without the requirement that every must transition has to be contained in the may transition relation. Larsen and Xinxin were the first to extend their must and may transition approach by hypertransitions, which resulted in their definition of *disjunctive modal transition systems* [16], called $\overline{\mathsf{DMTS}}$ in this work. They also defined a preorder on specifications, which expresses whether one specification is a *refinement* of another one. In that case, it allows fewer implementations. In the case that one of the specifications is fully determined, this notion corresponds to our disjunctive modal simulation. In the same paper, Larsen and Xinxin also showed how a $\overline{\mathsf{DMTS}}$ can be used to express the solution set of an equation system within process algebra.

Kripke modal transition systems (KMTS) [10,12], state-based versions of modal transition systems, are used as a model for abstraction in order to investigate more efficient model checking. KMTS do not have action labels on transitions. Instead, every state is labelled with the set of propositions holding there. Then validity of properties expressed in modal logics like the μ-calculus [13] can be checked with respect to all possible implementations. Obviously, besides the two outcomes *true* (all implementations satisfy the formula) and *false* (all implementations do not satisfy the formula), underspecification introduces a third truth value *unknown* (some implementations satisfy the formula, others do not), resulting in three-valued approaches for program analysis [11].

Generalized Kripke modal transition systems are KMTSs that also feature hypertransitions, thus are the state-based version of DMTSs. These were used by Shoham and Grumberg in [22].

Alfaro et al. used in [7] a DMTS-like approach for the underspecification of turn-based games [2], extending these structures by must and may transitions and hypertransitions. This resulted in their definition of *abstract game structures*. Validity of formulas in the alternating-time μ-calculus [3] can be checked, which again raises the need for three-valued semantics.

In [6], Dams and Namjoshi have uncovered yet another transition system notion called *focused transition systems (FTS)*. The corresponding abstraction framework is complete in the sense that for every system one can find a finite abstraction such that a given correctness property can be shown. The authors extend mixed transition systems by two types of hypertransitions, so called *focus* and *de-focus* steps. Satisfaction of formulas is defined via a game on the focused transition system and an alternating tree automaton [19] for the given formula. The notion of refinement is also defined via a game. The de-focus set of the FTS leads to some kind of hypertransition. It is future work to check, how exactly FTSs are related to the common disjunctive or our 1-selecting hypertransition approach.

6 Conclusion and Future Work

We have introduced an alternative simulation relation for disjunctive modal transition systems and have shown that this type of interpretation is at least as expressive as the common approach. It is future work to check whether our simulation notion is even more expressive. Furthermore, our 1-selecting approach can be generalized to an n-selecting approach, in which every hypertransition carries a number that determines, how many of the targets may be reachable in implementations at maximum. It remains to be examined whether this extension increases expressiveness or allows less complex representations. Yet another underspecification approach might consider DMTS-like transition systems, where every hypertransition can only have a single action label (that could be drawn in front of the division). Are these systems less expressive; and if not, do DMTSs allow significantly smaller representations?

References

[1] *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science.* IEEE Computer Society Press, 2004.

[2] S. Abramsky. Semantics of interaction. In *Trees in Algebra and Programming – CAAP'96, Proc. 21st Int. Coll., Linköping*, volume 1059, page 1. Springer-Verlag, 1996.

[3] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002.

[4] D. Dams. *Abstract interpretation and partition refinement for model checking.* PhD thesis, Technische Universiteit Eindhoven, The Netherlands, 1996.

[5] D. Dams, R. Gerth, and O. Grumberg. Abstract interpretation of reactive systems. *ACM Transactions on Programming Languages and Systems*, 19(2):253–291, 1997.

[6] D. Dams and K. S. Namjoshi. The existence of finite abstractions for branching time model checking. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science* [1].

[7] L. de Alfaro, P. Godefroid, and R. Jagadeesan. Three-valued abstractions of games: Uncertainty, but with precision. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science* [1].

[8] R. v. Glabbeek. The linear time – branching time spectrum I; the semantics of concrete, sequential processes. In J. Bergstra, A. Ponse, and S. Smolka, editors, *Handbook of Process Algebra*, chapter 1, pages 3–99. Elsevier, 2001. Available at http://boole.stanford.edu/pub/spectrum1.ps.gz.

[9] C. A. R. Hoare. A model for communicating sequential processes. In *On the Construction of Programs*, pages 229–254. Cambridge University Press, 1980.

[10] M. Huth. Model checking modal transition systems using Kripke structures. In A. Cortesi, editor, *VMCAI 2002*, volume 2294 of *LNCS*, pages 302+. Springer-Verlag, 2002.

[11] M. Huth, R. Jagadeesan, and D. Schmidt. Modal transition systems: A foundation for three-valued program analysis. *Lecture Notes in Computer Science*, 2028:155+, 2001.

[12] M. R. Huth, R. Jagadeesan, and D. A. Schmidt. A domain equation for refinement of partial systems. *Mathematical Structures in Computer Science*, 2003.

[13] D. Kozen. Results on the propositional μ-calculus. In *Proceedings of the 9th Colloquium on Automata, Languages and Programming*, pages 348–359. Springer-Verlag, 1982.

[14] K. G. Larsen. Modal specifications. In *Proceedings of the international workshop on Automatic verification methods for finite state systems*, pages 232–246. Springer-Verlag New York, Inc., 1990.

[15] K. G. Larsen and B. Thomsen. A modal process logic. In *Proceedings of the 3rd Annual IEEE Symposium on Logic in Computer Science*, pages 203–210. IEEE Computer Society Press, 1988.

[16] K. G. Larsen and L. Xinxin. Equation solving using modal transition systems. In *Proceedings of the 5th Annual IEEE Symposium on Logic in Computer Science*, pages 108–117. IEEE Computer Society Press, 1990.

[17] R. Milner. An algebraic definition of simulation between programs. In *IJCAI*, pages 481–489, 1971.

[18] R. Milner. *A Calculus for Communicating Systems*, volume 92 of *LNCS*. Springer-Verlag, 1980.

[19] D. Niwiński. Fixed point characterization of infinite behavior of finite-state systems. *Theoretical Computer Science*, 189(1-2):1–69, 1997.

[20] Object Management Group. *UML 2.0 Infrastructure Specification*, 2003. http://www.omg.org/cgi-bin/doc?ptc/03-09-15.

[21] Object Management Group. *UML 2.0 Superstructure Specification*, 2003. http://www.omg.org/cgi-bin/doc?ptc/03-08-02.

[22] S. Shoham and O. Grumberg. Monotonic abstraction-refinement for CTL. In *TACAS*, pages 546–560, 2004.