A linear-time–branching-time spectrum for behavioral specification theories

Uli Fahrenberg, Axel Legay

Version of Record: https://www.sciencedirect.com/science/article/pii/S2352220819300707
Manuscript_4452994ed0e3ec9dcfd75c3154b0a5da

A Linear-Time–Branching-Time Spectrum for Behavioral Specification Theories

Uli Fahrenberg[a], Axel Legay[b]

[a] *École polytechnique, Palaiseau, France*
[b] *Université catholique de Louvain, Belgium*

Abstract

We propose behavioral specification theories for most equivalences in the linear-time–branching-time spectrum. Almost all previous work on specification theories focuses on bisimilarity, but there is a clear interest in specification theories for other preorders and equivalences. We show that specification theories for preorders cannot exist and develop a general scheme which allows us to define behavioral specification theories, based on disjunctive modal transition systems, for most equivalences in the linear-time–branching-time spectrum.

Keywords: specification theory; linear-time–branching-time spectrum; disjunctive modal transition system

1. Introduction

Models and specifications are central objects in theoretical computer science. In model-based verification, models of computing systems are held up against specifications of their behaviors, and methods are developed to check whether or not a given model satisfies a given specification.

In recent years, behavioral specification theories have seen some popularity [1, 4, 5, 8, 13–15, 25, 26, 29, 33]. Here, the specification formalism is an extension of the modeling formalism, so that specifications have an operational interpretation and models are verified by comparing their operational behavior against the specification's behavior. Popular examples of such specification theories are modal transition systems [4,14,25], disjunctive modal transition systems [8,13,29], and acceptance specifications [15,33]. Also relations to contracts and interfaces have been exposed [5, 34], as have extensions for real-time and quantitative specifications and for models with data [6, 7, 10, 16, 17].

Except for the work by Vogler *et al.* in [13, 14], behavioral specification theories have been developed only to characterize bisimilarity. While bisimilarity is an important equivalence relation on models, there are many others which also are of interest. Examples include nested and k-nested simulation [2, 21], ready or $\frac{2}{3}$-simulation [28], trace equivalence [23], impossible futures [38], or the failure semantics of [12–14, 32, 37] and others.

In order to initiate a systematic study of specification theories for different semantics, we exhibit in this paper specification theories for most of the equivalences in van Glabbeek's linear-time–branching-time spectrum [36], see Figure 1.

To develop our systemization, we first have to clarify what precisely is meant by a specification theory. This is similar to the attempt at a uniform framework of specifications in [5], but our focus is more general. Inspired by the seminal work of Pnueli [32], Larsen [26], and Hennessy and Milner [22], we develop the point of view that a behavioral specification theory is an expressive specification formalism equipped with a mapping from models to their characteristic formulae and with a refinement preorder which generalizes the satisfaction relation between models and specifications.

We then introduce a general scheme of linear and branching relation families and show that variants of these characterize most of the preorders and equivalences in the linear-time–branching-time spectrum (notably also all of the ones mentioned above). We transfer our scheme to disjunctive modal transition systems and use it to define a linear-time–branching-time spectrum of refinement preorders, each giving rise to a specification theory for a different equivalence in the linear-time–branching-time spectrum.

Specification theories as we define them here are useful for incremental design and verification, as specifications can be refined until a sufficient level of detail is reached. The specification theories developed for bisimilarity in [1, 4, 8, 15, 25, 26, 29, 33] also include operations of conjunction and composition, hence allowing for compositional design and verification. What we present here is a first fundamental study of specification theories for equivalences other than bisimilarity, and we leave compositionality for future work.

To sum up, the contributions of this paper are as follows:

- a clarification of the basic theory of behavioral specification theories;
- a uniform treatment of most of the relations in the linear-time–branching-time spectrum;
- a uniform linear-time–branching-time spectrum of specification theories.

This article is a revised and extended version of the paper [19] which has been presented at the 43rd International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2017) in Limerick, Ireland. Compared to [19], and in addition to numerous small changes and improvements, motivation and examples, proofs of all results, as well as two additional sections on a game-based setting have been added to the paper.

2. Specification Theories

We start this paper by introducing and clarifying some concepts related to models and specifications from [22, 26, 32]. Let Mod be a set of models.

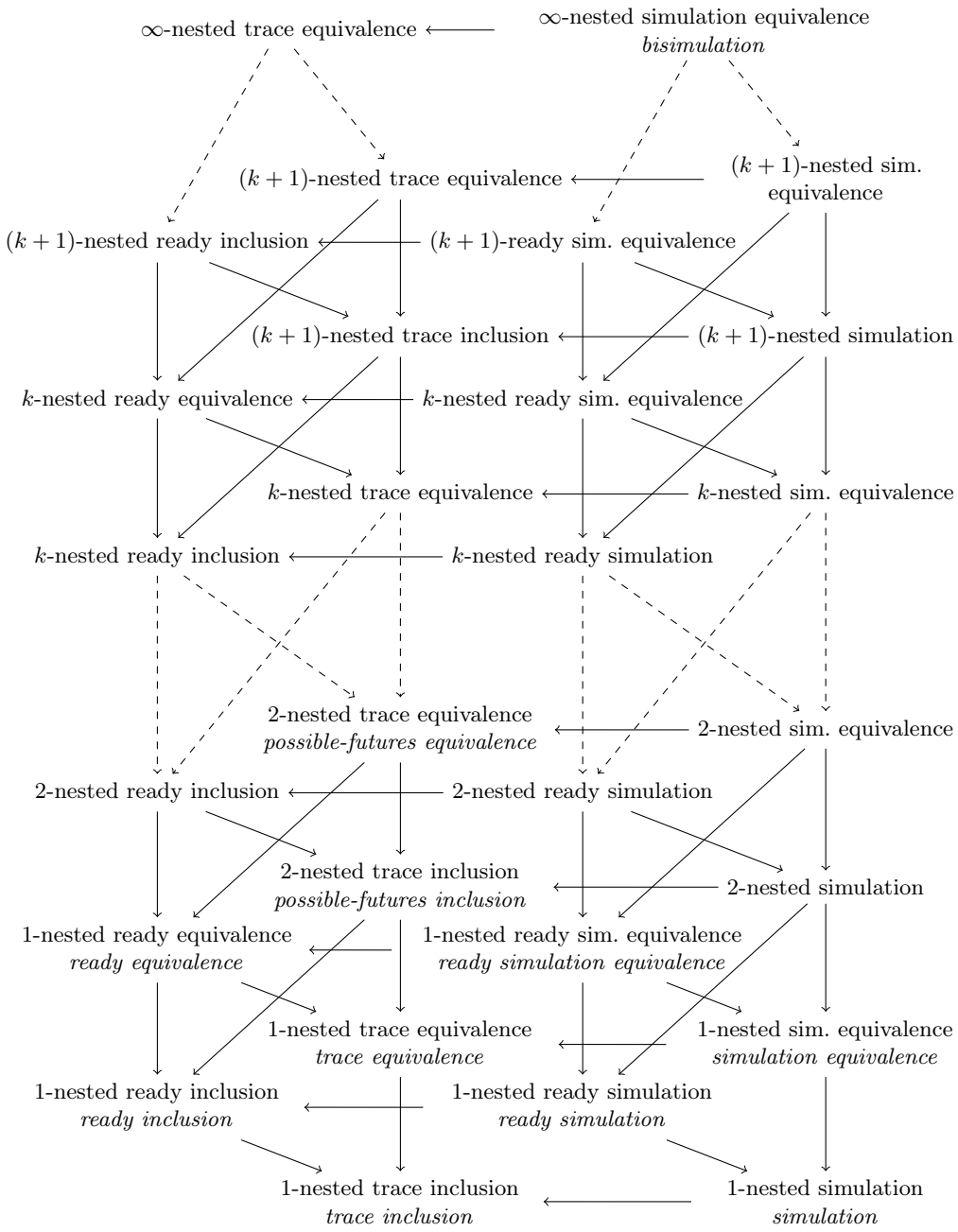

Figure 1: The linear-time–branching-time spectrum. The nodes are different preorders and equivalences, and an edge $R_1 \longrightarrow R_2$ or $R_1 \dashrightarrow R_2$ indicates that R_1 implies R_2 and that they are inequivalent in general.

Definition 1. A *specification formalism* for Mod is a structure $(\mathsf{Spec}, \models)$, where Spec is a set of specifications and $\models \subseteq \mathsf{Mod} \times \mathsf{Spec}$ is the satisfaction relation.

The models in Mod serve to represent computing systems, and the specifications in Spec represent properties of such systems. The *model checking* problem is, given $\mathcal{I} \in \mathsf{Mod}$ and $\mathcal{S} \in \mathsf{Spec}$, to decide whether $\mathcal{I} \models \mathcal{S}$.

Definition 2. For $\mathcal{S} \in \mathsf{Spec}$, $[\![\mathcal{S}]\!] = \{\mathcal{I} \in \mathsf{Mod} \mid \mathcal{I} \models \mathcal{S}\}$ denotes its set of *implementations*.

That is, $[\![\mathcal{S}]\!]$ is the set of models which adhere to the specification $\mathcal{S}$. Note that $\models$ and $[\![\cdot]\!]$ are inter-definable: for $\mathcal{I} \in \mathsf{Mod}$ and $\mathcal{S} \in \mathsf{Spec}$, $\mathcal{I} \models \mathcal{S}$ iff $\mathcal{I} \in [\![\mathcal{S}]\!]$.

Definition 3. For $\mathcal{S}_1, \mathcal{S}_2 \in \mathsf{Spec}$,

- $\mathcal{S}_1$ is *semantically refined* by $\mathcal{S}_2$, denoted $\mathcal{S}_1 \preceq \mathcal{S}_2$, if $[\![\mathcal{S}_1]\!] \subseteq [\![\mathcal{S}_2]\!]$;
- $\mathcal{S}_1$ is *semantically equivalent* to $\mathcal{S}_2$, denoted $\mathcal{S}_1 \cong \mathcal{S}_2$, if $[\![\mathcal{S}_1]\!] = [\![\mathcal{S}_2]\!]$.

Hence $\mathcal{S}_1 \preceq \mathcal{S}_2$ iff every implementation of $\mathcal{S}_1$ is also an implementation of $\mathcal{S}_2$, that is, if it holds for every model that once it satisfies $\mathcal{S}_1$, it automatically also satisfies $\mathcal{S}_2$.

Definition 4. For $\mathcal{I} \in \mathsf{Mod}$, $\mathsf{Th}(\mathcal{I}) = \{\mathcal{S} \in \mathsf{Spec} \mid \mathcal{I} \models \mathcal{S}\}$ denotes its set of *theories*.

That is, $\mathsf{Th}(\mathcal{I})$ is the set of all specifications which are satisfied by $\mathcal{I}$. Again, $\models$ and Th are inter-definable: for $\mathcal{I} \in \mathsf{Mod}$ and $\mathcal{S} \in \mathsf{Spec}$, $\mathcal{I} \models \mathcal{S}$ iff $\mathcal{S} \in \mathsf{Th}(\mathcal{I})$.

As [26] notes, the functions $[\![\cdot]\!] : \mathsf{Spec} \to 2^{\mathsf{Mod}}$ and $\mathsf{Th} : \mathsf{Mod} \to 2^{\mathsf{Spec}}$ can be extended to functions on sets of specifications and models by $[\![A]\!] = \bigcap_{\mathcal{S} \in A} [\![\mathcal{S}]\!]$ and $\mathsf{Th}(B) = \bigcap_{\mathcal{I} \in A} \mathsf{Th}(\mathcal{I})$, and then $[\![\cdot]\!] : 2^{\mathsf{Spec}} \rightleftarrows 2^{\mathsf{Mod}} : \mathsf{Th}$ forms a Galois connection.

Definition 5. For $\mathcal{I}_1, \mathcal{I}_2 \in \mathsf{Mod}$,

- $\mathcal{I}_1$ is *behaviorally refined* by $\mathcal{I}_2$, denoted $\mathcal{I}_1 \sqsubseteq \mathcal{I}_2$, if $\mathsf{Th}(\mathcal{I}_1) \subseteq \mathsf{Th}(\mathcal{I}_2)$;
- $\mathcal{I}_1$ is *behaviorally equivalent* to $\mathcal{I}_2$, denoted $\mathcal{I}_1 \sqsupseteq\sqsubseteq \mathcal{I}_2$, if $\mathsf{Th}(\mathcal{I}_1) = \mathsf{Th}(\mathcal{I}_2)$;

Hence $\mathcal{I}_1 \sqsupseteq\sqsubseteq \mathcal{I}_2$ iff $\mathcal{I}_1$ and $\mathcal{I}_2$ satisfy precisely the same specifications.

In terminology first introduced in [22], the specification formalism $(\mathsf{Spec}, \models)$ is said to be *adequate* for $\sqsupseteq\sqsubseteq$. In fact, the usual point of view is sightly different: normally, Mod comes equipped with some equivalence relation $\sim$, and then one says that $(\mathsf{Spec}, \models)$ is adequate for $(\mathsf{Mod}, \sim)$ if $\sqsupseteq\sqsubseteq = \sim$. It is clear that $\sim$ is not needed to reason about specification formalisms; we can simply declare that $(\mathsf{Spec}, \models)$ is adequate for whatever model equivalence $\sqsupseteq\sqsubseteq$ it *induces*.

Definition 6. A specification $\mathcal{S} \in \mathsf{Spec}$ is a *characteristic formula* for $\mathcal{I} \in \mathsf{Mod}$ if $\mathcal{I} \models \mathcal{S}$ and for all $\mathcal{I}' \models \mathcal{S}$, $\mathcal{I}' \sqsupseteq\sqsubseteq \mathcal{I}$.

This was introduced in [32]. We record the following property which follows directly from the definitions:

Lemma 7. *A specification $\mathcal{S} \in \mathsf{Spec}$ is a characteristic formula for $\mathcal{I} \in \mathsf{Mod}$ iff it holds for all $\mathcal{I}' \in \mathsf{Mod}$ that $\mathcal{S} \in \mathsf{Th}(\mathcal{I}')$ iff $\mathsf{Th}(\mathcal{I}) = \mathsf{Th}(\mathcal{I}')$.* □

Not surprisingly, characteristic formulae are unique up to semantic equivalence:

Lemma 8. *If $\mathcal{S}_1$ and $\mathcal{S}_2$ are characteristic formulae for $\mathcal{I} \in \mathsf{Mod}$, then $\mathcal{S}_1 \approx \mathcal{S}_2$.*

Proof. By Lemma 7, it holds for all $\mathcal{I}' \in \mathsf{Mod}$ that $\mathcal{I}' \models \mathcal{S}_1$ iff $\mathsf{Th}(\mathcal{I}) = \mathsf{Th}(\mathcal{I}')$, iff $\mathcal{I}' \models \mathcal{S}_2$. □

Again following [32], the specification formalism $(\mathsf{Spec}, \models)$ is said to be *expressive* for Mod if every $\mathcal{I} \in \mathsf{Mod}$ admits a characteristic formula. Our first result seems to have been overlooked in [22, 26, 32]: in an expressive specification formalism, the preorder $\sqsubseteq$ is, in fact, an equivalence.

Proposition 9. *If Spec is expressive for Mod, then $\sqsubseteq = \sqsupseteq$.*

Proof. Let $\mathcal{I}_1, \mathcal{I}_2 \in \mathsf{Mod}$ and assume $\mathcal{I}_1 \sqsubseteq \mathcal{I}_2$. Let $\mathcal{S}_1 \in \mathsf{Spec}$ be a characteristic formula for $\mathcal{I}_1$, then $\mathcal{S}_1 \in \mathsf{Th}(\mathcal{I}_1)$. But $\mathsf{Th}(\mathcal{I}_1) \subseteq \mathsf{Th}(\mathcal{I}_2)$, hence $\mathcal{S}_1 \in \mathsf{Th}(\mathcal{I}_2)$. By Lemma 7, this implies $\mathcal{I}_2 \sqsupseteq \mathcal{I}_1$. □

Example 10. A very simple specification formalism is $\mathsf{Spec} = 2^{\mathsf{Mod}}$, that is, specifications are sets of models. In that case, $\models = \in$ is the element-of relation, and $[\![\mathcal{S}]\!] = \mathcal{S}$, thus $\mathcal{S}_1 \preceq \mathcal{S}_2$ iff $\mathcal{S}_1 \subseteq \mathcal{S}_2$ and $\mathcal{S}_1 \approx \mathcal{S}_2$ iff $\mathcal{S}_1 = \mathcal{S}_2$.

Every $\mathcal{I} \in \mathsf{Mod}$ has characteristic formula $\{\mathcal{I}\} \in \mathsf{Spec}$, hence 2^{Mod} is expressive for Mod, so that $\sqsubseteq = \sqsupseteq$. Further, if $\mathcal{I}_1 \sqsupseteq \mathcal{I}_2$, then $\mathcal{I}_2 \in \{\mathcal{I}_1\}$, hence $\mathcal{I}_1 = \mathcal{I}_2$. We have shown that 2^{Mod} is adequate for equality $=$. □

Example 11. *Hennessy-Milner logic* [22] is a well-known specification formalism for labeled transition systems (see Definition 16 of LTS below). It consists of formulae generated by the abstract syntax

$$\mathsf{HML} \ni \phi, \psi ::= \mathbf{tt} \mid \mathbf{ff} \mid \phi \wedge \psi \mid \phi \vee \psi \mid \langle a \rangle \phi \mid [a]\phi \quad (a \in \Sigma)\,,$$

with semantics defined by $[\![\mathbf{tt}]\!] = \mathsf{LTS}$, $[\![\mathbf{ff}]\!] = \emptyset$, $[\![\phi \wedge \psi]\!] = [\![\phi]\!] \cap [\![\psi]\!]$, $[\![\phi \vee \psi]\!] = [\![\phi]\!] \cup [\![\psi]\!]$, and

$$[\![\langle a \rangle \phi]\!] = \{(S, s^0, T) \in \mathsf{LTS} \mid \exists (s^0, a, s) \in T : (S, s, T) \in [\![\phi]\!]\}\,,$$
$$[\![[a]\phi]\!] = \{(S, s^0, T) \in \mathsf{LTS} \mid \forall (s^0, a, s) \in T : (S, s, T) \in [\![\phi]\!]\}\,.$$

HML admits a semantic form of negation, *complementation*, which is defined inductively by $\mathbf{tt}^c = \mathbf{ff}$, $\mathbf{ff}^c = \mathbf{tt}$, $(\phi \wedge \psi)^c = \phi^c \vee \psi^c$, $(\phi \vee \psi)^c = \phi^c \wedge \psi^c$, $(\langle a \rangle \phi)^c = [a]\phi^c$, and $([a]\phi)^c = \langle a \rangle \phi^c$. It can be shown [3] that for all $\phi \in \mathsf{HML}$, $[\![\phi^c]\!] = \mathsf{Mod} \setminus [\![\phi]\!]$.

Now let $\mathcal{I}_1, \mathcal{I}_2 \in \mathsf{LTS}$ and assume $\mathcal{I}_1 \sqsubseteq \mathcal{I}_2$, then it holds for all $\phi \in \mathsf{HML}$ that $\mathcal{I}_1 \models \phi$ implies $\mathcal{I}_2 \models \phi$. By contraposition, $\mathcal{I}_2 \models \phi^c$ implies $\mathcal{I}_1 \models \phi^c$ for all $\phi \in \mathsf{HML}$, so that $\mathcal{I}_2 \sqsubseteq \mathcal{I}_1$. We have shown that $\sqsubseteq = \sqsupseteq$. In fact, by the Hennessy-Milner theorem [22], $\sqsupseteq$ is bisimilarity, so that HML is adequate for bisimilarity.

Even though $\sqsubseteq = \sqsupseteq$, it can be shown [3] that HML is *not* expressive. □

3. Behavioral Specification Theories

We are ready to introduce what we mean by a behavioral specification theory: an expressive specification formalism with extra structure. This mainly sums up and clarifies ideas already present in [5, 26], but we make a connection between specification theories and characteristic formulae which is new. Specifically, we will see that a central ingredient in a specification theory is a function χ which maps models to their characteristic formulae.

Definition 12. A (behavioral) *specification theory* for Mod is a specification formalism $(\mathsf{Spec}, \models)$ for Mod together with a mapping $\chi : \mathsf{Mod} \to \mathsf{Spec}$ and a preorder $\leq$ on Spec, called *modal refinement*, subject to the following conditions:

- for every $\mathcal{I} \in \mathsf{Mod}$, $\chi(\mathcal{I})$ is a characteristic formula for $\mathcal{I}$;
- for all $\mathcal{I} \in \mathsf{Mod}$ and all $\mathcal{S} \in \mathsf{Spec}$, $\mathcal{I} \models \mathcal{S}$ iff $\chi(\mathcal{I}) \leq \mathcal{S}$.

The equivalence relation $\equiv \; = \; \leq \cap \geq$ on Spec is called *modal equivalence*. Note that specification theories are indeed expressive; also, $\models$ is fully determined by $\leq$.

In a categorical sense, the function $\chi : \mathsf{Mod} \to \mathsf{Spec}$ is a *section* of the Galois connection $[\![\cdot]\!] : 2^{\mathsf{Spec}} \rightleftarrows 2^{\mathsf{Mod}} : \mathsf{Th}$. Indeed, we have $\chi(\mathcal{I}) \in \mathsf{Th}(\mathcal{I})$ for all $\mathcal{I} \in \mathsf{Mod}$ and $\mathcal{I}' \sqsupseteq \mathcal{I}$ for all $\mathcal{I}' \in [\![\chi(\mathcal{I})]\!]$, and these properties are characterizing for χ. Further, $\mathsf{Th}(\mathcal{I}) = \{\mathcal{S} \mid \chi(\mathcal{I}) \leq \mathcal{S}\} = \chi(\mathcal{I})\uparrow$ is the *upward closure* of $\chi(\mathcal{I})$.

We sum up a few consequences of the definition: modal refinement (equivalence, resp.) implies semantic refinement (equivalence, resp.), and on characteristic formulae, all refinements and equivalences collapse.

Proposition 13. *Let* $(\mathsf{Spec}, \chi, \leq)$ *be a specification theory for* Mod.

1. *For all* $\mathcal{S}_1, \mathcal{S}_2 \in \mathsf{Spec}$, $\mathcal{S}_1 \leq \mathcal{S}_2$ *implies* $\mathcal{S}_1 \preceq \mathcal{S}_2$ *and* $\mathcal{S}_1 \equiv \mathcal{S}_2$ *implies* $\mathcal{S}_1 \cong \mathcal{S}_2$.
2. *For all* $\mathcal{I}_1, \mathcal{I}_2 \in \mathsf{Mod}$, *the following are equivalent:* $\chi(\mathcal{I}_1) \leq \chi(\mathcal{I}_2)$, $\chi(\mathcal{I}_2) \leq \chi(\mathcal{I}_1)$, $\chi(\mathcal{I}_1) \preceq \chi(\mathcal{I}_2)$, $\chi(\mathcal{I}_2) \preceq \chi(\mathcal{I}_1)$, $\mathcal{I}_1 \sqsupseteq \mathcal{I}_2$.

Proof. The first claim follows from transitivity of $\leq$: if $\mathcal{I} \in [\![\mathcal{S}_1]\!]$, then $\chi(\mathcal{I}) \leq \mathcal{S}_1 \leq \mathcal{S}_2$, hence $\chi(\mathcal{I}) \leq \mathcal{S}_2$, thus $\mathcal{I} \in [\![\mathcal{S}_2]\!]$.

For the second claim, let $\mathcal{I}_1, \mathcal{I}_2 \in \mathsf{Mod}$.

- If $\chi(\mathcal{I}_1) \leq \chi(\mathcal{I}_2)$, then $\chi(\mathcal{I}_1) \preceq \chi(\mathcal{I}_2)$ by the first part.

- If $\chi(\mathcal{I}_1) \preceq \chi(\mathcal{I}_2)$, then $[\![\chi(\mathcal{I}_1)]\!] \subseteq [\![\chi(\mathcal{I}_2)]\!]$. But $\mathcal{I}_1 \in [\![\chi(\mathcal{I}_1)]\!]$, hence $\mathcal{I}_1 \in [\![\chi(\mathcal{I}_2)]\!]$, which, as $\chi(\mathcal{I}_2)$ is characteristic, implies $\mathcal{I}_1 \sqsubseteq \mathcal{I}_2$. Also, $\mathcal{I}_1 \in [\![\chi(\mathcal{I}_2)]\!]$ implies $\chi(\mathcal{I}_1) \leq \chi(\mathcal{I}_2)$.
- Assume $\mathcal{I}_1 \sqsubseteq \mathcal{I}_2$ and let $\mathcal{I} \in [\![\chi(\mathcal{I}_1)]\!]$. Then $\mathcal{I} \sqsubseteq \mathcal{I}_1$, hence $\mathcal{I} \sqsubseteq \mathcal{I}_2$, which implies $\mathcal{I} \in [\![\chi(\mathcal{I}_2)]\!]$. We have shown that $\chi(\mathcal{I}_1) \preceq \chi(\mathcal{I}_2)$.

We have shown that $\chi(\mathcal{I}_1) \leq \chi(\mathcal{I}_2)$ iff $\chi(\mathcal{I}_1) \preceq \chi(\mathcal{I}_2)$ iff $\mathcal{I}_1 \sqsubseteq \mathcal{I}_2$, and reversing the roles of $\mathcal{I}_1$ and $\mathcal{I}_2$ gives the other equivalences. □

The second part of the proposition means that the mapping $\chi : \mathsf{Mod} \to \mathsf{Spec}$ is an *embedding up to equivalence*: for all $\mathcal{I}_1, \mathcal{I}_2 \in \mathsf{Mod}$, $\mathcal{I}_1 \sqsubseteq \mathcal{I}_2$ iff $\chi(\mathcal{I}_1) \equiv \chi(\mathcal{I}_2)$ iff $\chi(\mathcal{I}_1) \cong \chi(\mathcal{I}_2)$. Because of this, most work in specification theories *identifies* models $\mathcal{I}$ with their characteristic formulae $\chi(\mathcal{I})$; for reasons of clarity, we will not make this identification here.

We finish this section with a lemma which shows that the property of $\chi(\mathcal{I})$ being characteristic formulae follows when $\leq$ is symmetric on models.

Lemma 14. *Let* Spec *be a set,* $\chi : \mathsf{Mod} \to \mathsf{Spec}$ *a mapping and* $\leq \subseteq \mathsf{Spec} \times \mathsf{Spec}$ *a preorder. If the restriction of* $\leq$ *to the image of* χ *is symmetric, then* $(\mathsf{Spec}, \chi, \leq)$ *is a specification theory for* Mod.

Proof. We know that $\chi(\mathcal{I}_1) \leq \chi(\mathcal{I}_2)$ iff $\chi(\mathcal{I}_2) \leq \chi(\mathcal{I}_1)$ for all $\mathcal{I}_1, \mathcal{I}_2 \in \mathsf{Mod}$. Let $\mathcal{I} \in \mathsf{Mod}$; we need to show that $\chi(\mathcal{I})$ is a characteristic formula for $\mathcal{I}$.

First, by reflexivity of $\leq$, $\chi(\mathcal{I}) \leq \chi(\mathcal{I})$ implies $\mathcal{I} \models \chi(\mathcal{I})$. Now let $\mathcal{I}' \in \mathsf{Mod}$ and assume $\mathcal{I}' \models \chi(\mathcal{I})$, that is, $\chi(\mathcal{I}') \leq \chi(\mathcal{I})$. We show that $\mathsf{Th}(\mathcal{I}') \supseteq \mathsf{Th}(\mathcal{I})$. Let $\mathcal{S} \in \mathsf{Th}(\mathcal{I})$, then $\mathcal{I} \models \mathcal{S}$, that is, $\chi(\mathcal{I}) \leq \mathcal{S}$. But $\leq$ is transitive, so $\chi(\mathcal{I}') \leq \chi(\mathcal{I}) \leq \mathcal{S}$ implies $\chi(\mathcal{I}') \leq \mathcal{S}$. Hence $\mathcal{I}' \models \mathcal{S}$, so that $\mathcal{S} \in \mathsf{Th}(\mathcal{I}')$.

We have shown that $\chi(\mathcal{I}') \leq \chi(\mathcal{I})$ implies $\mathsf{Th}(\mathcal{I}') \supseteq \mathsf{Th}(\mathcal{I})$. By symmetry of $\leq$ on the image of χ, $\chi(\mathcal{I}') \leq \chi(\mathcal{I})$ implies $\chi(\mathcal{I}) \leq \chi(\mathcal{I}')$, which in turn implies $\mathsf{Th}(\mathcal{I}) \supseteq \mathsf{Th}(\mathcal{I}')$. We have proven that $\mathcal{I}' \models \chi(\mathcal{I})$ implies $\mathsf{Th}(\mathcal{I}') = \mathsf{Th}(\mathcal{I})$. □

Example 15. We have seen that Hennessy-Milner logic is not expressive, hence HML cannot serve as basis for a specification theory for LTS. The standard remedy for expressivity is to add recursion to the logic, see [3, 27]; we will in Sect. 4 below expose a specification theory based on Hennessy-Milner logic with recursion and maximal fixed points.

For our other example, $\mathsf{Spec} = 2^{\mathsf{Mod}}$, we can let $\chi(\mathcal{I}) = \{\mathcal{I}\}$ and $\leq = \subseteq$. Then $\mathcal{I} \in \mathcal{S}$ iff $\{\mathcal{I}\} \subseteq \mathcal{S}$, *i.e.* $\mathcal{I} \models \mathcal{S}$ iff $\chi(\mathcal{I}) \leq \mathcal{S}$. This shows that $(2^{\mathsf{Mod}}, \chi, \subseteq)$ is a specification theory for Mod (which is adequate and expressive for equality).

4. Disjunctive Modal Transition Systems

We proceed to recall disjunctive modal transition systems and how these can serve as a specification theory for bisimilarity. The material in this section is well-known, but our definitions from the previous sections allow for much more succinctness, for example in Proposition 19 below.

From now on, Mod will be the set LTS of (finite) *labeled transition systems* over a fixed finite alphabet Σ:

Definition 16. A *labeled transition system* (S, s^0, T) consists of a finite set of states S, an initial state $s^0 \in S$, and transitions $T \subseteq S \times \Sigma \times S$ labeled with symbols from Σ.

Recall [30,31] that two LTS (S_1, s_1^0, T_1) and (S_2, s_2^0, T_2) are *bisimilar* if there exists a relation $R \subseteq S_1 \times S_2$ such that $(s_1^0, s_2^0) \in R$ and for all $(s_1, s_2) \in R$,

- for all $(s_1, a, t_1) \in T_1$, there is $(s_2, a, t_2) \in T_2$ with $(t_1, t_2) \in R$,
- for all $(s_2, a, t_2) \in T_2$, there is $(s_1, a, t_1) \in T_1$ with $(t_1, t_2) \in R$.

Definition 17. A *disjunctive modal transition system* (DMTS) is a tuple $\mathcal{D} = (S, S^0, \dashrightarrow, \longrightarrow)$ consisting of finite sets $S \supseteq S^0$ of states and initial states, a *may*-transition relation $\dashrightarrow \subseteq S \times \Sigma \times S$, and a *disjunctive must*-transition relation $\longrightarrow \subseteq S \times 2^{\Sigma \times S}$. It is assumed that for all $(s, N) \in \longrightarrow$ and all $(a, t) \in N$, $(s, a, t) \in \dashrightarrow$.

DMTS were introduced in [29], but note that we permit several (or no) initial states here. The set of DMTS is denoted DMTS.

As customary, we write $s \overset{a}{\dashrightarrow} t$ instead of $(s, a, t) \in \dashrightarrow$ and $s \longrightarrow N$ instead of $(s, N) \in \longrightarrow$. The intuition is that may-transitions $s \overset{a}{\dashrightarrow} t$ specify which transitions are permitted in an implementation, whereas a must-transition $s \longrightarrow N$ stipulates a disjunctive requirement: at least one of the choices $(a, t) \in N$ has to be implemented.

Definition 18. A *modal refinement* of two DMTS $\mathcal{D}_1 = (S_1, S_1^0, \dashrightarrow_1, \longrightarrow_1)$, $\mathcal{D}_2 = (S_2, S_2^0, \dashrightarrow_2, \longrightarrow_2)$ is a relation $R \subseteq S_1 \times S_2$ for which it holds of all $(s_1, s_2) \in R$ that

- $\forall s_1 \overset{a}{\dashrightarrow}_1 t_1 : \exists s_2 \overset{a}{\dashrightarrow}_2 t_2 : (t_1, t_2) \in R$;
- $\forall s_2 \longrightarrow_2 N_2 : \exists s_1 \longrightarrow_1 N_1 : \forall (a, t_1) \in N_1 : \exists (a, t_2) \in N_2 : (t_1, t_2) \in R$;

and such that for all $s_1^0 \in S_1^0$, there exists $s_2^0 \in S_2^0$ for which $(s_1^0, s_2^0) \in R$.

Let $\leq \subseteq \mathsf{DMTS} \times \mathsf{DMTS}$ be the relation defined by $\mathcal{D}_1 \leq \mathcal{D}_2$ iff there exists a modal refinement as above (a *witness* for $\mathcal{D}_1 \leq \mathcal{D}_2$). Clearly, $\leq$ is a preorder.

LTS are embedded into DMTS as follows. For an LTS $\mathcal{I} = (S, s^0, T)$, let $\chi(\mathcal{I}) = (S, \{s^0\}, \dashrightarrow, \longrightarrow)$ be the DMTS with $\dashrightarrow = T$ and $\longrightarrow = \{(s, \{(a, t)\}) \mid (s, a, t) \in T\}$. The following proposition reformulates well-known facts about DMTS and modal refinement.

Proposition 19. $(\mathsf{DMTS}, \chi, \leq)$ *is a specification theory for* LTS *adequate for bisimilarity.*

Proof. In lieu of Lemma 14, we show that $\leq$ is bisimilarity, hence symmetric, on the image of χ. Let $\mathcal{I}_1, \mathcal{I}_2 \in \mathsf{LTS}$ and assume $\chi(\mathcal{I}_1) \leq \chi(\mathcal{I}_2)$. Write $\mathcal{I}_1 = (S_1, s_1^0, T_1)$, $\mathcal{I}_2 = (S_2, s_2^0, T_2)$, $\chi(\mathcal{I}_1) = (S_1, \{s_1^0\}, \dashrightarrow_1, \longrightarrow_1)$, and $\chi(\mathcal{I}_2) = (S_2, \{s_2^0\}, \dashrightarrow_2, \longrightarrow_2)$.

We have a relation $R \subseteq S_1 \times S_2$ such that $(s_1^0, s_2^0) \in R$ and for all $(s_1, s_2) \in R$, $\forall s_1 \overset{a}{\dashrightarrow}_1 t_1 : \exists s_2 \overset{a}{\dashrightarrow}_2 t_2 : (t_1, t_2) \in R$ and $\forall s_2 \longrightarrow_2 N_2 : \exists s_1 \longrightarrow_1 N_1 : \forall (a, t_1) \in N_1 : \exists (a, t_2) \in N_2 : (t_1, t_2) \in R$. Let $(s_1, s_2) \in R$. We show that R is a bisimulation.

Let $(s_1, a, t_1) \in T_1$. Then $s_1 \overset{a}{\dashrightarrow}_1 t_1$, so that we have a transition $s_2 \overset{a}{\dashrightarrow}_2 t_2$ with $(t_1, t_2) \in R$. By definition of $\chi(\mathcal{I}_1)$, $(s_2, a, t_2) \in T_2$.

Let $(s_2, a, t_2) \in T_2$. Then $s_2 \longrightarrow_2 N_2 = \{(a, t_2)\}$, hence there is $s_1 \longrightarrow_1 N_1$ such that $\forall (a, t_1) \in N_1 : \exists (a, t_2') \in N_2 : (t_1, t_2') \in R$. But then $t_2' = t_2$, and by definition of $\chi(\mathcal{I}_2)$, $N_1 = \{(a, t_1)\}$ must be a one-element set, hence $(s_1, a, t_1) \in T_1$ and $(t_1, t_2) \in R$.

We have shown that $\chi(\mathcal{I}_1) \leq \chi(\mathcal{I}_2)$ implies that $\mathcal{I}_1$ and $\mathcal{I}_2$ are bisimilar; the proof of the other direction is similar. □

4.1. Hennessy-Milner Logic with Maximal Fixed Points

It is shown in [9,20] that there is a bijective translation between DMTS and Hennessy-Milner logic with recursion and maximal fixed points [27]. For a finite set X of variables, let $\mathsf{HML}(X)$ be the set of formulae generated as follows:

$$\mathsf{HML}(X) \ni \phi, \psi ::= \mathbf{tt} \mid \mathbf{ff} \mid \phi \wedge \psi \mid \phi \vee \psi \mid \langle a \rangle \phi \mid [a]\phi \mid x \quad (a \in \Sigma, x \in X)$$

A *recursive Hennessy-Milner formula* [9,20,27] is a tuple $\mathcal{H} = (X, X^0, \Delta)$ consisting of finite sets $X \supseteq X^0$ of variables and initial variables and a *declaration* $\Delta : X \to \mathsf{HML}(X)$. The set of such formulae is denoted $\mathsf{HML}^{\mathsf{R}}$. The semantics of a formula $\mathcal{H} \in \mathsf{HML}^{\mathsf{R}}$ is a set $[\![\mathcal{H}]\!] \in \mathsf{LTS}$ which is defined as a maximal fixed point, see [3,9,27] for details.

In [9,20], and extending results of [11,25], it is shown that there is a bijective translation between DMTS and recursive HML formulae. That is, there are mappings $dh : \mathsf{DMTS} \to \mathsf{HML}^{\mathsf{R}}$ and $hd : \mathsf{HML}^{\mathsf{R}} \to \mathsf{DMTS}$ such that $hd \circ dh$ and $dh \circ hd$ are identities.

We can now define modal refinement of recursive HML formulae by $\mathcal{H}_1 \leq \mathcal{H}_2$ iff $hd(\mathcal{H}_1) \leq hd(\mathcal{H}_2)$. We also embed LTS into $\mathsf{HML}^{\mathsf{R}}$ by $\chi(\mathcal{I}) = dh(\chi_{\mathsf{DMTS}}(\mathcal{I}))$, where χ_{DMTS} is the embedding $\mathsf{LTS} \to \mathsf{DMTS}$; this is the usual characteristic-formula construction from, for example, [3].

Proposition 20 ([9, 20]). $(\mathsf{HML}^{\mathsf{R}}, \chi, \leq)$ *is a specification theory for* LTS *adequate for bisimilarity.* □

5. A Specification Theory for Simulation Equivalence

We want to construct specification theories for other interesting relations in the linear-time–branching-time spectrum [36]. Given Proposition 9 and the fact that specification theories are expressive, we know that it is futile to look for specification theories for *preorders* in the spectrum. What we *can* do, however, is find specification theories for the *equivalences* in the spectrum. To warm up, we start out by a specification theory for simulation equivalence.

Recall [24] that a *simulation* of LTS (S_1, s_1^0, T_1), (S_2, s_2^0, T_2) is a relation $R \subseteq S_1 \times S_2$ such that $(s_1^0, s_2^0) \in R$ and for all $(s_1, s_2) \in R$,

- for all $(s_1, a, t_1) \in T_1$, there is $(s_2, a, t_2) \in T_2$ with $(t_1, t_2) \in R$.

LTS (S_1, s_1^0, T_1) and (S_2, s_2^0, T_2) are said to be *simulation equivalent* if there exist a simulation $R^1 \subseteq S_1 \times S_2$ *and* a simulation $R^2 \subseteq S_2 \times S_1$.

Definition 21. Let $\mathcal{D}_1 = (S_1, S_1^0, \dashrightarrow_1, \longrightarrow_1)$, $\mathcal{D}_2 = (S_2, S_2^0, \dashrightarrow_2, \longrightarrow_2)$ be DMTS. A *simulation refinement* consists of two relations $R_1, R_2 \subseteq S_1 \times S_2$ such that

1. $\forall s_1^0 \in S_1^0 : \exists s_2^0 \in S_2^0 : (s_1^0, s_2^0) \in R_1$ and $\forall s_2^0 \in S_2^0 : \exists s_1^0 \in S_1^0 : (s_1^0, s_2^0) \in R_2$;
2. $\forall (s_1, s_2) \in R_1 : \forall s_1 \overset{a}{\dashrightarrow}_1 t_1 : \exists s_2 \overset{a}{\dashrightarrow}_2 t_2 : (t_1, t_2) \in R_1$;
3. $\forall (s_1, s_2) \in R_2 : \forall s_2 \longrightarrow_2 N_2 : \exists s_1 \longrightarrow_1 N_1 : \forall (a, t_1) \in N_1 : \exists (a, t_2) \in N_2 : (t_1, t_2) \in R_2$.

Intuitively, R_1 is a simulation of may-transitions from $\mathcal{D}_1$ to $\mathcal{D}_2$, whereas R_2 is a simulation of disjunctive must-transitions from $\mathcal{D}_2$ to $\mathcal{D}_1$. Let $\leq_{\mathsf{s}} \subseteq \mathsf{DMTS} \times \mathsf{DMTS}$ be the relation defined by $\mathcal{D}_1 \leq_{\mathsf{s}} \mathcal{D}_2$ iff there exists a simulation refinement as above. Clearly, $\leq_{\mathsf{s}}$ is a preorder. A direct proof of the following theorem, similar to the one of Proposition 19, is shown below, but it also follows from the later Theorem 26.

Theorem 22. $(\mathsf{DMTS}, \chi, \leq_{\mathsf{s}})$ *forms a specification theory for* LTS *adequate for simulation equivalence.*

Proof. We show that $\leq_{\mathsf{s}}$ is simulation equivalence, hence symmetric, on the image of χ and apply Lemma 14. Let $\mathcal{I}_1, \mathcal{I}_2 \in \mathsf{LTS}$ and assume $\chi(\mathcal{I}_1) \leq_{\mathsf{s}} \chi(\mathcal{I}_2)$. Write $\mathcal{I}_1 = (S_1, s_1^0, T_1)$, $\mathcal{I}_2 = (S_2, s_2^0, T_2)$, $\chi(\mathcal{I}_1) = (S_1, \{s_1^0\}, \dashrightarrow_1, \longrightarrow_1)$, and $\chi(\mathcal{I}_2) = (S_2, \{s_2^0\}, \dashrightarrow_2, \longrightarrow_2)$.

Let $R_1, R_2 \subseteq S_1 \times S_2$ be relations as of Definition 21. Then $(s_1^0, s_2^0) \in R_1$ and $(s_1^0, s_2^0) \in R_2$. We show that $R_1 \subseteq S_1 \times S_2$ and $R_2^{-1} \subseteq S_2 \times S_1$ are simulations.

Let $(s_1, s_2) \in R_1$ and $(s_1, a, t_1) \in T_1$. Then $s_1 \overset{a}{\dashrightarrow}_1 t_1$, hence there is $s_2 \overset{a}{\dashrightarrow}_2 t_2$ such that $(t_1, t_2) \in R_1$. But then also $(s_2, a, t_2) \in T_2$.

Let $(s_2, s_1) \in R_2^{-1}$ and $(s_2, a, t_2) \in T_2$. Then $s_2 \longrightarrow_2 N_2 = \{(a, s_2)\}$, hence there is $s_1 \longrightarrow_1 N_1$ such that $\forall (a, t_1) \in N_1 : \exists (a, t_2') \in N_2 : (t_1, t_2') \in R_2$. But then $t_2' = t_2$ and $N_1 = \{(a, t_1)\}$, hence $(s_1, a, t_1) \in T_1$ and $(t_2, t_1) \in R_2^{-1}$. □

6. Specification Theories for Branching Equivalences

We proceed to generalize the work in the preceding section and develop DMTS-based specification theories for all *branching* equivalences in the linear-time–branching-time spectrum in Figure 1. Examples of such branching equivalences include the bisimilarity and simulation equivalence which we have already seen, but also ready simulation equivalence [28] and nested simulation equivalence [2,21] are important. We will treat the linear part of the spectrum, which includes relations such as trace equivalence [23], impossible-futures equivalence [38] or failure equivalence [12–14, 32, 37], in the next section.

We start by laying out a scheme which systematically covers all branching relations in the spectrum.

Definition 23. Let $k \geq 0$ and $\mathcal{I}_1 = (S_1, s_1^0, T_1), \mathcal{I}_2 = (S_2, s_2^0, T_2) \in \mathsf{LTS}$. A *branching k-switching relation family* from $\mathcal{I}_1$ to $\mathcal{I}_2$ consists of relations $R^0, \ldots, R^k \subseteq S_1 \times S_2$ such that $(s_1^0, s_2^0) \in R^0$ and

- for all *even* $j \in \{0, \ldots, k\}$ and $(s_1, s_2) \in R^j$:
  - $\forall (s_1, a, t_1) \in T_1 : \exists (s_2, a, t_2) \in T_2 : (t_1, t_2) \in R^j$;
  - if $j < k$, then $\forall (s_2, a, t_2) \in T_2 : \exists (s_1, a, t_1) \in T_1 : (t_1, t_2) \in R^{j+1}$;
- for all *odd* $j \in \{0, \ldots, k\}$ and $(s_1, s_2) \in R^j$:
  - $\forall (s_2, a, t_2) \in T_2 : \exists (s_1, a, t_1) \in T_1 : (t_1, t_2) \in R^j$;
  - if $j < k$, then $\forall (s_1, a, t_1) \in T_1 : \exists (s_2, a, t_2) \in T_2 : (t_1, t_2) \in R^{j+1}$.

Clearly, a simulation is the same as a branching 0-switching relation family. Also, a branching 1-switching relation family is a *nested simulation*: the initial states are related in R^0; any transition in $\mathcal{I}_1$ from a pair $(s_1, s_2) \in R^0$ has to be matched recursively in $\mathcal{I}_2$; and at any point in time, the sense of the matching can switch, in that now transitions in $\mathcal{I}_2$ from a pair $(s_1, s_2) \in R^1$ have to be matched recursively by transitions in $\mathcal{I}_1$. In general, a branching k-switching relation family is a k-nested simulation, see also [21, Definition 8.5.2] which is similar to ours. A branching ∞-switching relation family is a bisimulation: any transition in $\mathcal{I}_1$ has to be matched recursively by one in $\mathcal{I}_2$ and vice versa. We refer to [18] for more motivation.

Definition 24. Let $k \geq 0$ and $\mathcal{I}_1 = (S_1, s_1^0, T_1), \mathcal{I}_2 = (S_2, s_2^0, T_2) \in \mathsf{LTS}$. A *branching k-ready relation family* from $\mathcal{I}_1$ to $\mathcal{I}_2$ is a branching k-switching relation family $R^0, \ldots, R^k \subseteq S_1 \times S_2$ with the extra property that for all $(s_1, s_2) \in R^k$:

- if k is even, then $\forall (s_2, a, t_2) \in T_2 : \exists (s_1, a, t_1) \in T_1$;
- if k is odd, then $\forall (s_1, a, t_1) \in T_1 : \exists (s_2, a, t_2) \in T_2$.

Hence a branching 0-ready relation family is the same as a *ready simulation*: any transition in $\mathcal{I}_1$ has to be matched recursively by one in $\mathcal{I}_2$; and at any point in time, precisely the same actions have to be available in the two states. A branching 1-ready relation family would be a nested ready simulation, and so on. Branching k-switching and k-ready relation families cover all branching relations in the linear-time–branching-time spectrum.

Because of Proposition 9, we are only interested in equivalences. For $k \geq 0$ and $\mathcal{I}_1, \mathcal{I}_2 \in \mathsf{LTS}$, we write $\mathcal{I}_1 \sim_k \mathcal{I}_2$ if there exist a branching k-switching relation family from $\mathcal{I}_1$ to $\mathcal{I}_2$ and another from $\mathcal{I}_2$ to $\mathcal{I}_1$. We write $\mathcal{I}_1 \sim_k^{\mathrm{r}} \mathcal{I}_2$ if there exist a branching k-ready relation family from $\mathcal{I}_1$ to $\mathcal{I}_2$ and another from $\mathcal{I}_2$ to $\mathcal{I}_1$. Then $\sim_0$ is simulation equivalence, $\sim_1$ is nested simulation equivalence, $\sim_\infty$ is bisimilarity, $\sim_0^{\mathrm{r}}$ is ready simulation equivalence, etc.

We proceed to devise specification theories for LTS which are adequate for $\sim_k$ and $\sim_k^{\mathrm{r}}$.

Definition 25. Let $k \geq 0$, $\mathcal{D}_1 = (S_1, S_1^0, \dashrightarrow_1, \longrightarrow_1), \mathcal{D}_2 = (S_2, S_2^0, \dashrightarrow_2, \longrightarrow_2) \in \mathsf{DMTS}$. A *branching k-switching relation family* from $\mathcal{D}_1$ to $\mathcal{D}_2$ consists of relations $R_1^0, \ldots, R_1^k, R_2^0, \ldots, R_2^k \subseteq S_1 \times S_2$ such that

- $\forall s_1^0 \in S_1^0 : \exists s_2^0 \in S_2^0 : (s_1^0, s_2^0) \in R_1^0$ and $\forall s_2^0 \in S_2^0 : \exists s_1^0 \in S_1^0 : (s_1^0, s_2^0) \in R_2^0$;
- for all *even* $j \in \{0, \ldots, k\}$ and $(s_1, s_2) \in R_1^j$:
  - $\forall s_1 \overset{a}{\dashrightarrow}_1 t_1 : \exists s_2 \overset{a}{\dashrightarrow}_2 t_2 : (t_1, t_2) \in R_1^j$;
  - if $j < k$, then $\forall s_2 \longrightarrow_2 N_2 : \exists s_1 \longrightarrow_1 N_1 : \forall (a, t_1) \in N_1 : \exists (a, t_2) \in N_2 : (t_1, t_2) \in R_1^{j+1}$;
- for all *odd* $j \in \{0, \ldots, k\}$ and $(s_1, s_2) \in R_1^j$:
  - $\forall s_2 \longrightarrow_2 N_2 : \exists s_1 \longrightarrow_1 N_1 : \forall (a, t_1) \in N_1 : \exists (a, t_2) \in N_2 : (t_1, t_2) \in R_1^j$;
  - if $j < k$, then $\forall s_1 \overset{a}{\dashrightarrow}_1 t_1 : \exists s_2 \overset{a}{\dashrightarrow}_2 t_2 : (t_1, t_2) \in R_1^{j+1}$;
- for all *even* $j \in \{0, \ldots, k\}$ and $(s_1, s_2) \in R_2^j$:
  - $\forall s_2 \longrightarrow_2 N_2 : \exists s_1 \longrightarrow_1 N_1 : \forall (a, t_1) \in N_1 : \exists (a, t_2) \in N_2 : (t_1, t_2) \in R_2^j$;
  - if $j < k$, then $\forall s_1 \overset{a}{\dashrightarrow}_1 t_1 : \exists s_2 \overset{a}{\dashrightarrow}_2 t_2 : (t_1, t_2) \in R_2^{j+1}$.
- for all *odd* $j \in \{0, \ldots, k\}$ and $(s_1, s_2) \in R_2^j$:
  - $\forall s_1 \overset{a}{\dashrightarrow}_1 t_1 : \exists s_2 \overset{a}{\dashrightarrow}_2 t_2 : (t_1, t_2) \in R_2^j$;
  - if $j < k$, then $\forall s_2 \longrightarrow_2 N_2 : \exists s_1 \longrightarrow_1 N_1 : \forall (a, t_1) \in N_1 : \exists (a, t_2) \in N_2 : (t_1, t_2) \in R_2^{j+1}$;

A *branching k-ready relation family* from $\mathcal{D}_1$ to $\mathcal{D}_2$ is a branching k-switching relation family as above with the extra property that if k is even, then

- $\forall (s_1, s_2) \in R_1^k : \forall s_2 \longrightarrow_2 N_2 : \exists s_1 \longrightarrow_1 N_1 : \forall (a, t_1) \in N_1 : \exists (a, t_2) \in N_2$;
- $\forall (s_1, s_2) \in R_2^k : \forall s_1 \overset{a}{\dashrightarrow}_1 t_1 : \exists s_2 \overset{a}{\dashrightarrow}_2 t_2$;

and if k is odd, then

- $\forall (s_1, s_2) \in R_1^k : \forall s_1 \overset{a}{\dashrightarrow}_1 t_1 : \exists s_2 \overset{a}{\dashrightarrow}_2 t_2$;
- $\forall (s_1, s_2) \in R_2^k : \forall s_2 \longrightarrow_2 N_2 : \exists s_1 \longrightarrow_1 N_1 : \forall (a, t_1) \in N_1 : \exists (a, t_2) \in N_2$.

For $k \geq 0$ and $\mathcal{D}_1, \mathcal{D}_2 \in \mathsf{DMTS}$, we write $\mathcal{D}_1 \leq_k \mathcal{D}_2$ if there exist a branching k-switching relation family from $\mathcal{D}_1$ to $\mathcal{D}_2$. We write $\mathcal{D}_1 \leq_k^{\mathrm{r}} \mathcal{D}_2$ if there exist a branching k-ready relation family from $\mathcal{D}_1$ to $\mathcal{D}_2$. Note that $\leq_0$ is the relation $\leq_{\mathsf{s}}$ from the preceding section.

Theorem 26. *For any $k \geq 0$, $(\mathsf{DMTS}, \chi, \leq_k)$ is a specification theory for* LTS *adequate for $\sim_k$, and $(\mathsf{DMTS}, \chi, \leq_k^{\mathrm{r}})$ is a specification theory for* LTS *adequate for $\sim_k^{\mathrm{r}}$.*

Proof. Let $k \geq 0$. We show that $(\mathsf{DMTS}, \chi, \leq_k)$ is a specification theory for LTS adequate for $\sim_k$; the proof for $\leq_k^{\mathrm{r}}$ is similar. We will apply Lemma 14. Let $\mathcal{I}_1 = (S_1, s_1^0, T_1), \mathcal{I}_2 = (S_2, s_2^0, T_2) \in \mathsf{LTS}$ and write $\chi(\mathcal{I}_1) = (S_1, \{s_1^0\}, \dashrightarrow_1, \longrightarrow_1)$ and $\chi(\mathcal{I}_2) = (S_2, \{s_2^0\}, \dashrightarrow_2, \longrightarrow_2)$; we must prove that $\chi(\mathcal{I}_1) \leq_k \chi(\mathcal{I}_2)$ iff $\mathcal{I}_1 \sim_k \mathcal{I}_2$.

Assume that $\chi(\mathcal{I}_1) \leq_k \chi(\mathcal{I}_2)$ and let $R_1^0, \ldots, R_1^k, R_2^0, \ldots, R_2^k \subseteq S_1 \times S_2$ be a DMTS-branching k-switching relation family from $\chi(\mathcal{I}_1)$ to $\chi(\mathcal{I}_2)$ as of Definition 25. We show that $R_1^0, \ldots, R_1^k$ is an LTS-branching k-switching relation family from $\mathcal{I}_1$ to $\mathcal{I}_2$ as of Definition 23. First, we have $(s_1^0, s_2^0) \in R_1^0$.

Let $j \in \{0, \ldots, k\}$ even and $(s_1, s_2) \in R_1^j$. Let $(s_1, a, t_1) \in T_1$, then $s_1 \overset{a}{\dashrightarrow}_1 t_1$, hence there is $s_2 \overset{a}{\dashrightarrow}_2 t_2$ such that $(t_1, t_2) \in R_1^j$, but then also $(s_2, a, t_2) \in T_2$. If $j < k$, then let $(s_2, a, t_2) \in T_2$, thus $s_2 \longrightarrow_2 N_2 = \{(a, t_2)\}$. Hence there is $s_1 \longrightarrow_1 N_1$ such that $\forall (a, t_1) \in N_1 : \exists (a, t_2') \in N_2 : (t_1, t_2') \in R_1^{j+1}$. But then $t_2' = t_2$ and $N_1 = \{(a, t_1)\}$, hence $(s_1, a, t_1) \in T_1$. The arguments for j odd are similar.

We have shown that $R_1^0, \ldots, R_1^k$ is an LTS-branching k-switching relation family from $\mathcal{I}_1$ to $\mathcal{I}_2$. Analogously, one can show that $R_2^0, \ldots, R_2^k$ is an LTS-branching k-switching relation family from $\mathcal{I}_2$ to $\mathcal{I}_1$. The proof that $\mathcal{I}_1 \sim_k \mathcal{I}_2$ implies $\chi(\mathcal{I}_1) \leq_k \chi(\mathcal{I}_2)$ proceeds along similar lines. □

Remark 27. *There is a setting of* generalized simulation games*, based on Stirling's bisimulation games [35], which generalizes the above constructions and gives them a natural context. We have developed these in a quantitative setting in [18], and we provide an exposition of the approach in Section 8. Generalized simulation games can be lifted to games on DMTS which can be used to define the relations of Definition 25, see Section 9.*

7. Specification Theories for Linear Equivalences

We develop a scheme similar to the one of the previous section to cover all linear relations in the linear-time–branching-time spectrum. For $\mathcal{I} = (S, s^0, T) \in \mathsf{LTS}$, we let $T^* \subseteq S \times \Sigma^* \times S$ be the reflexive, transitive closure of T; a recursive definition is as follows:

- $(s, \varepsilon, s) \in T^*$ for all $s \in S$;
- for all $(s, \tau, t) \in T^*$ and $(t, a, u) \in T$, also $(s, \tau.a, u) \in T^*$.

Definition 28. Let $k \geq 0$ and $\mathcal{I}_1 = (S_1, s_1^0, T_1), \mathcal{I}_2 = (S_2, s_2^0, T_2) \in \mathsf{LTS}$. A *linear k-switching relation family* from $\mathcal{I}_1$ to $\mathcal{I}_2$ consists of relations $R^0, \ldots, R^k \subseteq S_1 \times S_2$ such that $(s_1^0, s_2^0) \in R^0$ and

- for all *even* $j \in \{0, \ldots, k\}$ and $(s_1, s_2) \in R^j$:

- $\forall (s_1, \tau, t_1) \in T_1^* : \exists (s_2, \tau, t_2) \in T_2^*$;
- if $j < k$, then $\forall (s_1, \tau, t_1) \in T_1^* : \exists (s_2, \tau, t_2) \in T_2^* : (t_1, t_2) \in R^{j+1}$;

- for all *odd* $j \in \{0, \ldots, k\}$ and $(s_1, s_2) \in R^j$:
  - $\forall (s_2, \tau, t_2) \in T_2^* : \exists (s_1, \tau, t_1) \in T_1^*$;
  - if $j < k$, then $\forall (s_2, \tau, t_2) \in T_2^* : \exists (s_1, \tau, t_1) \in T_1^* : (t_1, t_2) \in R^{j+1}$;

Hence a linear 0-switching relation family is a *trace inclusion*, and a linear 1-switching relation family is a *impossible-futures inclusion*: any trace in $\mathcal{I}_1$ has to be matched by a trace in $\mathcal{I}_2$, and then any trace from the end of the second trace has to be matched by one from the end of the first trace.

Definition 29. Let $k \geq 0$ and $\mathcal{I}_1 = (S_1, s_1^0, T_1), \mathcal{I}_2 = (S_2, s_2^0, T_2) \in \mathsf{LTS}$. A *linear k-ready relation family* from $\mathcal{I}_1$ to $\mathcal{I}_2$ is a linear k-switching relation family $R^0, \ldots, R^k \subseteq S_1 \times S_2$ with the extra property that for all $(s_1, s_2) \in R^k$:

- if k is even, then $\forall (s_1, \tau, t_1) \in T_1^* : \exists (s_2, \tau, t_2) \in T_2^* : \forall (t_2, a, u_2) \in T_2 : \exists (t_1, a, u_1) \in T_1$;
- if k is odd, then $\forall (s_2, \tau, t_2) \in T_2^* : \exists (s_1, \tau, t_1) \in T_1^* : \forall (t_1, a, u_1) \in T_1 : \exists (t_2, a, u_2) \in T_2$.

Thus a linear 0-ready relation family is a *failure inclusion*: any trace in $\mathcal{I}_1$ has to be matched by a trace in $\mathcal{I}_2$ such that there is an inclusion of *failure sets* of non-available actions. For $k \geq 0$ and $\mathcal{I}_1, \mathcal{I}_2 \in \mathsf{LTS}$, we write $\mathcal{I}_1 \approx_k \mathcal{I}_2$ if there exist a branching k-switching relation family from $\mathcal{I}_1$ to $\mathcal{I}_2$ and another from $\mathcal{I}_2$ to $\mathcal{I}_1$. We write $\mathcal{I}_1 \approx_k^{\mathrm{r}} \mathcal{I}_2$ if there exist a branching k-ready relation family from $\mathcal{I}_1$ to $\mathcal{I}_2$ and another from $\mathcal{I}_2$ to $\mathcal{I}_1$.

For $\mathcal{D} = (S, S^0, \dashrightarrow, \longrightarrow) \in \mathsf{DMTS}$, we define $\dashrightarrow^*, \longrightarrow^* \subseteq S \times \Sigma^* \times S$ recursively as follows:

- $s \overset{\varepsilon}{\dashrightarrow}^* s$ and $s \overset{\varepsilon}{\longrightarrow}^* s$ for all $s \in S$;
- for all $s \overset{\tau}{\dashrightarrow}^* t$ and $t \overset{a}{\dashrightarrow} u$, also $s \overset{\tau.a}{\dashrightarrow}^* u$;
- for all $s \overset{\tau}{\longrightarrow}^* t$, $t \longrightarrow N$, and $(a, u) \in N$, also $s \overset{\tau.a}{\longrightarrow}^* u$.

Definition 30. Let $k \geq 0, \mathcal{D}_1 = (S_1, S_1^0, \dashrightarrow_1, \longrightarrow_1), \mathcal{D}_2 = (S_2, S_2^0, \dashrightarrow_2, \longrightarrow_2) \in \mathsf{DMTS}$. A *linear k-switching relation family* from $\mathcal{D}_1$ to $\mathcal{D}_2$ consists of relations $R_1^0, \ldots, R_1^k, R_2^0, \ldots, R_2^k \subseteq S_1 \times S_2$ such that

- $\forall s_1^0 \in S_1^0 : \exists s_2^0 \in S_2^0 : (s_1^0, s_2^0) \in R_1^0$ and $\forall s_2^0 \in S_2^0 : \exists s_1^0 \in S_1^0 : (s_1^0, s_2^0) \in R_2^0$;
- for all *even* $j \in \{0, \ldots, k\}$ and $(s_1, s_2) \in R_1^j$:
  - $\forall s_1 \overset{\tau}{\dashrightarrow}_1^* t_1 : \exists s_2 \overset{\tau}{\dashrightarrow}_2^* t_2$;
  - if $j < k$, then $\forall s_1 \overset{\tau}{\dashrightarrow}_1^* t_1 : \exists s_2 \overset{\tau}{\dashrightarrow}_2^* t_2 : (t_1, t_2) \in R_1^{j+1}$;

- for all *odd* $j \in \{0, \ldots, k\}$ and $(s_1, s_2) \in R_1^j$:
  - $\forall s_2 \xrightarrow{\tau}_2^* t_2 : \exists s_1 \xrightarrow{\tau}_1^* t_1$;
  - if $j < k$, then $\forall s_2 \xrightarrow{\tau}_2^* t_2 : \exists s_1 \xrightarrow{\tau}_1^* t_1 : (t_1, t_2) \in R_1^{j+1}$;
- for all *even* $j \in \{0, \ldots, k\}$ and $(s_1, s_2) \in R_2^j$:
  - $\forall s_2 \xrightarrow{\tau}_2^* t_2 : \exists s_1 \xrightarrow{\tau}_1^* t_1$;
  - if $j < k$, then $\forall s_2 \xrightarrow{\tau}_2^* t_2 : \exists s_1 \xrightarrow{\tau}_1^* t_1 : (t_1, t_2) \in R_1^{j+1}$;
- for all *odd* $j \in \{0, \ldots, k\}$ and $(s_1, s_2) \in R_2^j$:
  - $\forall s_1 \overset{\tau}{\dashrightarrow}_1^* t_1 : \exists s_2 \overset{\tau}{\dashrightarrow}_2^* t_2$;
  - if $j < k$, then $\forall s_1 \overset{\tau}{\dashrightarrow}_1^* t_1 : \exists s_2 \overset{\tau}{\dashrightarrow}_2^* t_2 : (t_1, t_2) \in R_2^{j+1}$.

A *linear k-ready relation family* from $\mathcal{D}_1$ to $\mathcal{D}_2$ is a linear k-switching relation family as above with the extra property that if k is even, then

- $\forall (s_1, s_2) \in R_1^k : \forall s_1 \overset{\tau}{\dashrightarrow}_1^* t_1 : \exists s_2 \overset{\tau}{\dashrightarrow}_2^* t_2 : \forall t_2 \longrightarrow_2 N_2 : \exists t_1 \longrightarrow_1 N_1 : \forall (a, u_1) \in N_1 : \exists (a, u_2) \in N_2$;
- $\forall (s_1, s_2) \in R_2^k : \forall s_2 \xrightarrow{\tau}_2^* t_2 : \exists s_1 \xrightarrow{\tau}_1^* t_1 : \forall t_1 \overset{a}{\dashrightarrow}_1 u_1 : \exists t_2 \overset{a}{\dashrightarrow}_2 u_2$;

and if k is odd, then

- $\forall (s_1, s_2) \in R_1^k : \forall s_2 \xrightarrow{\tau}_2^* t_2 : \exists s_1 \xrightarrow{\tau}_1^* t_1 : \forall t_1 \overset{a}{\dashrightarrow}_1 u_1 : \exists t_2 \overset{a}{\dashrightarrow}_2 u_2$;
- $\forall (s_1, s_2) \in R_2^k : \forall s_1 \overset{\tau}{\dashrightarrow}_1^* t_1 : \exists s_2 \overset{\tau}{\dashrightarrow}_2^* t_2 : \forall t_2 \longrightarrow_2 N_2 : \exists t_1 \longrightarrow_1 N_1 : \forall (a, u_1) \in N_1 : \exists (a, u_2) \in N_2$;

For $k \geq 0$ and $\mathcal{D}_1, \mathcal{D}_2 \in \mathsf{DMTS}$, we write $\mathcal{D}_1 \preccurlyeq_k \mathcal{D}_2$ if there exists a linear k-switching relation family from $\mathcal{D}_1$ to $\mathcal{D}_2$ and $\mathcal{D}_1 \preccurlyeq_k^{\mathrm{r}} \mathcal{D}_2$ if there exists a linear k-ready relation family from $\mathcal{D}_1$ to $\mathcal{D}_2$.

Theorem 31. *For any $k \geq 0$, $(\mathsf{DMTS}, \chi, \preccurlyeq_k)$ is a specification theory for LTS adequate for $\approx_k$, and $(\mathsf{DMTS}, \chi, \preccurlyeq_k^{\mathrm{r}})$ is a specification theory for LTS adequate for $\approx_k^{\mathrm{r}}$.*

Proof. Let $k \geq 0$. We first show that $(\mathsf{DMTS}, \chi, \preccurlyeq_k)$ is a specification theory for LTS adequate for $\approx_k$. We will apply Lemma 14.

Let $\mathcal{I}_1 = (S_1, s_1^0, T_1), \mathcal{I}_2 = (S_2, s_2^0, T_2) \in \mathsf{LTS}$ and denote $\chi(\mathcal{I}_1) = (S_1, \{s_1^0\}, \dashrightarrow_1, \longrightarrow_1)$ and $\chi(\mathcal{I}_2) = (S_2, \{s_2^0\}, \dashrightarrow_2, \longrightarrow_2)$. We show that $\chi(\mathcal{I}_1) \preccurlyeq_k \chi(\mathcal{I}_2)$ implies $\mathcal{I}_1 \approx_k \mathcal{I}_2$; the other direction is similar.

Assume that $\chi(\mathcal{I}_1) \preccurlyeq_k \chi(\mathcal{I}_2)$ and let $R_1^0, \ldots, R_1^k, R_2^0, \ldots, R_2^k \subseteq S_1 \times S_2$ be a DMTS-linear k-switching relation family from $\chi(\mathcal{I}_1)$ to $\chi(\mathcal{I}_2)$ as of Definition 30. We show that $R_1^0, \ldots, R_1^k$ is an LTS-linear k-switching relation family from $\mathcal{I}_1$ to $\mathcal{I}_2$ as of Definition 28. First, we have $(s_1^0, s_2^0) \in R_1^0$.

Let $j \in \{0, \ldots, k\}$ even and $(s_1, s_2) \in R_1^j$. Let $(s_1, \tau, t_1) \in T_1^*$, then $s_1 \overset{\tau}{\dashrightarrow}_1^* t_1$, hence there is $s_2 \overset{\tau}{\dashrightarrow}_2^* t_2$, implying that $(s_2, \tau, t_2) \in T_2^*$. If $j < k$, then there is also $s_2 \overset{\tau}{\dashrightarrow}_2^* t_2$ such that $(t_1, t_2) \in R_1^{j+1}$, and again $(s_2, \tau, t_2) \in T_2^*$.

Let $j \in \{0, \ldots, k\}$ odd and $(s_1, s_2) \in R_1^j$. Let $(s_2, \tau, t_2) \in T_2^*$, then $s_2 \overset{\tau}{\longrightarrow}_2^* t_2$. Hence there is $s_1 \overset{\tau}{\longrightarrow}_1^* t_1$, *i.e.* $(s_1, \tau, t_1) \in T_1^*$. If $j < k$, then there is $s_1 \overset{\tau}{\longrightarrow}_1^* t_1$, *i.e.* $(s_1, \tau, t_1) \in T_1^*$, such that $(t_1, t_2) \in R_1^{j+1}$.

We have shown that $R_1^0, \ldots, R_1^k$ is an LTS-linear k-switching relation family from $\mathcal{I}_1$ to $\mathcal{I}_2$. Similarly, one can show that $R_2^0, \ldots, R_2^k$ is an LTS-linear k-switching relation family from $\mathcal{I}_2$ to $\mathcal{I}_1$.

Now assume that $\chi(\mathcal{I}_1) \preccurlyeq_k^{\mathrm{r}} \chi(\mathcal{I}_2)$; we show that $\mathcal{I}_1 \approx_k^{\mathrm{r}} \mathcal{I}_2$ (the other direction is again similar). Let $R_1^0, \ldots, R_1^k, R_2^0, \ldots, R_2^k \subseteq S_1 \times S_2$ be a DMTS-linear k-ready relation family from $\chi(\mathcal{I}_1)$ to $\chi(\mathcal{I}_2)$. We show that $R_1^0, \ldots, R_1^k$ is an LTS-linear k-ready relation family from $\mathcal{I}_1$ to $\mathcal{I}_2$; again, the proof that $R_2^0, \ldots, R_2^k$ is an LTS-linear k-ready relation family from $\mathcal{I}_2$ to $\mathcal{I}_1$ is completely analogous. First, we have $(s_1^0, s_2^0) \in R_1^0$.

We already know that $R_1^0, \ldots, R_1^k$ is an LTS-linear k-switching relation family from $\mathcal{I}_1$ to $\mathcal{I}_2$, so we only need to see the extra conditions in Definition 29. Let $(s_1, s_2) \in R_1^k$ and assume k to be even (the proof is similar for k odd). Let $(s_1, \tau, t_1) \in T_1^*$, then $s_1 \overset{\tau}{\dashrightarrow}_1^* t_1$, hence there is $s_2 \overset{\tau}{\dashrightarrow}_2^* t_2$, *i.e.* $(s_2, \tau, t_2) \in T_2^*$, such that $\forall t_2 \longrightarrow_2 N_2 : \exists t_1 \longrightarrow_1 N_1 : \forall (a, u_1) \in N_1 : \exists (a, u_2) \in N_2$.

Let $(t_2, a, u_2) \in T_2$, then $t_2 \longrightarrow_2 N_2 = \{(a, u_2)\}$. Hence there is $t_1 \longrightarrow_1 N_1$ such that $\forall (a, u_1) \in N_1 : \exists (a, u_2') \in N_2$, but then $N_1 = \{(a, u_1)\}$, hence $(t_1, a, u_1) \in T_1$. □

8. Generalized Simulation Games

In order to provide context to the constructions in Sect. 6, we introduce a notion of *generalized simulation game*. This is a generalization of Stirling's bisimulation game [35] which permits to define most of the preorders and equivalences in van Glabbeek's linear-time–branching-time spectrum [36]. See also [18] for a quantitative version of these games.

Let $\mathcal{I}_1 = (S_1, s_1^0, T_1), \mathcal{I}_2 = (S_2, s_2^0, T_2) \in \mathsf{LTS}$. We will define a game played by two players, I and II, which intuitively proceeds as follows. Starting from the initial configuration (s_1^0, s_2^0), player I chooses a transition from s_1^0. Player II then has to match this with a transition with the same label from s_2^0, and the game continues from the new configuration (s_1, s_2) given by the target states of the two chosen transitions. The game is won by player I if she plays a transition which player II cannot match; if this never happens, player II wins.

We will see below that player II has a strategy to always win this game iff there is a *simulation* from $\mathcal{I}_1$ to $\mathcal{I}_2$. In order to characterize other preorders and equivalences, we introduce some variability into the game:

- In any configuration (s_1, s_2), player I may choose to *switch sides* and from now on play transitions from the right (s_2) component instead of the left, which player II then has to answer by matching transitions on the left side. Player I may later choose to switch sides again.

- In any configuration (s_1, s_2), player I may also choose to play a *last* transition which ends the game. If player II can match the transition, then she has won; otherwise, player I wins.

Different combinations of these variations, together with restrictions on when and how often player I is allowed to switch sides, will define games which characterize all branching equivalences in the linear-time–branching-time spectrum.

We formalize the above description. The sets of *extended states* for the players are

$$C_1 = (T_1 \times T_2 \cup T_2 \times T_1)^* ,$$
$$C_2 = (T_1 \times T_2 \cup T_2 \times T_1)^* . (T_1 \cup T_2) .$$

These keep track of which edges have been previously chosen by the players. Note that C_1 contains the empty extended state ε.

A *strategy for player I* is a partial mapping $\theta_1 : C_1 \rightharpoonup T_1 \cup T_2$ such that whenever $\theta_1(((s_1, a_1, t_1), (s'_1, a'_1, t'_1)) \dots ((s_n, a_n, t_n), (s'_n, a'_n, t'_n))) = (s, a, t)$ is defined, then $s = t_n$ or $s = t'_n$. Hence an edge chosen by player I must extend one of the previous two edges. If $\theta_1(\varepsilon) = (s, a, t)$ is defined, then $s = s_1^0$ or $s = s_2^0$. The set of strategies for player I is denoted Θ_1. For $c_1 \in C_1$ and $\theta_1 \in \Theta_1$, the *update* $\mathsf{upd}(c_1)$ of c_1 is defined iff $\theta_1(c_1)$ is defined, and then $\mathsf{upd}(c_1) = c_1.\theta_1(c_1) \in C_2$.

A *strategy for player II* is a partial mapping $\theta_2 : C_2 \rightharpoonup T_1 \cup T_2$ such that whenever $\theta_1(((s_1, a_1, t_1), (s'_1, a'_1, t'_1)) \dots ((s_n, a_n, t_n), (s'_n, a'_n, t'_n)).(s, a, t)) = (s', a', t')$ is defined, then $a = a'$, and

- if $s = t_n$, then $(s', a', t') \in T_2$ and $s' = t'_n$;
- if $s = t'_n$, then $(s', a', t') \in T_1$ and $s' = t_n$.

Hence player II has to play a transition with the same label as the last transition played by player I and on the opposite side of the game. The set of strategies for player II is denoted Θ_2. For $c_2 \in C_2$ and $\theta_2 \in \Theta_2$, the *update* $\mathsf{upd}_2(c_2)$ of c_2 is defined iff $\theta_2(c_2)$ is defined, and then $\mathsf{upd}_2(c_2) = c_2.\theta_2(c_2) \in C_1$.

Now let $(\theta_1, \theta_2) \in \Theta_1 \times \Theta_2$ be a *strategy pair*, then this induces a finite or infinite alternating sequence $(c_1^0, c_2^1, c_1^1, c_2^2, \dots)$ of extended states, where $c_1^0 = \varepsilon$ and for all $j \geq 1$,

- c_2^j is defined iff $\theta_2(c_1^{j-1})$ is defined, and then $c_2^j = \theta_2(c_1^{j-1})$;
- c_1^j is defined iff $\theta_1(c_2^j)$ is defined, and then $c_1^j = \theta_1(c_2^j)$.

Each extended state in the sequence is a prefix of the succeeding one, hence these define a finite or infinite string

$$\sigma(\theta_1, \theta_2) \in C_1 \cup C_2 \cup (T_1 \times T_2 \cup T_2 \times T_1)^\omega .$$

A strategy $\theta_1 \in \Theta_1$ is *winning for player I* if $\sigma(\theta_1, \theta_2) \in C_2$ for all $\theta_2 \in \Theta_2$. A strategy $\theta_2 \in \Theta_2$ is *winning for player II* if $\sigma(\theta_1, \theta_2) \in C_1 \cup (T_1 \times T_2 \cup T_2 \times T_1)^\omega$ for all $\theta_1 \in \Theta_1$. The game is determined, so that player I has a winning strategy iff player II does not.

Remark 32. *As the game is about player II matching transitions played by player I, and once she has done so, past transition labels are ignored, it is clear that it suffices to consider* memory-less *strategies for both players, i.e. strategies where the transitions chosen only depend on the current game configuration instead of all past moves. This is important from an algorithmic point of view, but we will not need it below.*

We introduce a *switch counter* sc which indicates how often player I has switched sides to arrive at a given extended state $c_1 \in C_1 = (T_1 \times T_2 \cup T_2 \times T_1)^*$. Intuitively, $\mathsf{sc}(c_1)$ counts how often the elements in the sequence c_1 switch from being in $T_1 \times T_2$ to being in $T_2 \times T_1$ and vice versa. Hence $\mathsf{sc}(c_1) = 0$ iff $c_1 \in (T_1 \times T_2)^* \cup (T_2 \times T_1)^*$, $\mathsf{sc}(c_1) = 1$ iff $c_1 \in (T_1 \times T_2)^+(T_2 \times T_1)^+ \cup (T_2 \times T_1)^+(T_1 \times T_2)^+$, etc. For $c_2 \in C_2$, we similarly have $\mathsf{sc}(c_2) = 0$ iff $c_2 \in (T_1 \times T_2)^* T_1 \cup (T_2 \times T_1)^* T_2$, $\mathsf{sc}(c_2) = 1$ iff $c_2 \in (T_1 \times T_2)^+ T_2 \cup (T_2 \times T_1)^+ T_1$, etc.

Definition 33. Let $k \geq 0$. A strategy $\theta_1 \in \Theta_1$ is *k-switching* if $\mathsf{sc}(\theta_1(c_1)) \leq k$ for all $c_1 \in C_1$ for which $\theta(c_1)$ is defined. It is *k-ready switching* if $\mathsf{sc}(c_1) \leq k$ for all $c_1 \in C_1$ for which $\theta(c_1)$ is defined.

Hence a 0-switching strategy for player I can never switch sides, whence a 0-ready switching strategy can switch sides once, but must be undefined after. Similarly, a 1-switching strategy can switch sides once, and a 1-ready switching strategy can then switch once more, but no more player I moves are defined after. We denote the sets of k-switching strategies by Θ_1^k and of k-ready switching strategies by $\Theta_1^{k\text{-r}}$. Note that $\Theta_1^k \subseteq \Theta_1^{k\text{-r}}$ for all $k \geq 0$, and $\Theta_1^\infty = \Theta_1^{\infty\text{-r}} = \Theta_1$.

For any subset $\Theta_1' \subseteq \Theta_1$, the *$\Theta_1'$-game* denotes the above game when player I is only permitted to use strategies in Θ_1'.

Proposition 34. *Let $k \geq 0$ and $\mathcal{I}_1, \mathcal{I}_2 \in \mathsf{LTS}$. Then $\mathcal{I}_1 \sim_k \mathcal{I}_2$ iff player II has a winning strategy in the Θ_1^k-game on $\mathcal{I}_1, \mathcal{I}_2$, and $\mathcal{I}_1 \sim_k^{\mathrm{r}} \mathcal{I}_2$ iff player II has a winning strategy in the $\Theta_1^{k\text{-r}}$-game on $\mathcal{I}_1, \mathcal{I}_2$.*

Proof. If $\theta_2 \in \Theta_2$ is winning for player II in the specification Θ_1^k-game, then any strategy pair (θ_1, θ_2) can be used to construct a branching k-switching relation family. Conversely, any branching k-switching relation family can be used to construct a (memory-less) winning player-II strategy in the specification Θ_1^k-game. The proof is similar for the k-ready case. □

Remark 35. *By suitably modifying the* sc *notion, also* preorders *in the spectrum can be characterized. By introducing a notion of* blind *strategy for player I, also linear relations in the spectrum can be covered. See [18] for details.*

9. Specification Games

We can now use the developments in the last section to introduce general specification games on DMTS which can be instantiated to yield specification

theories which are adequate for any equivalence in the linear-time–branching-time spectrum.

Let $\mathcal{D}_1 = (S_1, S_1^0, \dashrightarrow_1, \longrightarrow_1), \mathcal{D}_2 = (S_2, S_2^0, \dashrightarrow_2, \longrightarrow_2) \in \mathsf{DMTS}$. The sets of *extended states* for the players are

$$\begin{aligned}
C_1 &= ((\dashrightarrow_1 \times \dashrightarrow_2) \cup (\longrightarrow_2 \times \longrightarrow_1 \times \Sigma \times S_1 \times \Sigma \times S_2))^* \,, \\
C_2 &= ((\dashrightarrow_1 \times \dashrightarrow_2) \cup (\longrightarrow_2 \times \longrightarrow_1 \times \Sigma \times S_1 \times \Sigma \times S_2))^* . (\dashrightarrow_1 \cup \longrightarrow_2) \,, \\
C_1' &= ((\dashrightarrow_1 \times \dashrightarrow_2) \cup (\longrightarrow_2 \times \longrightarrow_1 \times \Sigma \times S_1 \times \Sigma \times S_2))^* . (\longrightarrow_2 \times \longrightarrow_1) \,, \\
C_2' &= ((\dashrightarrow_1 \times \dashrightarrow_2) \cup (\longrightarrow_2 \times \longrightarrow_1 \times \Sigma \times S_1 \times \Sigma \times S_2))^* . \\
&\qquad\qquad (\longrightarrow_2 \times \longrightarrow_1 \times \Sigma \times S_1) \,.
\end{aligned}$$

This conveys the following intuition: At each round of the game, player I either plays a may-transition in $\mathcal{D}_1$ or a disjunctive must-transition in $\mathcal{D}_2$. In the first case, player II answers with a matching may-transition in $\mathcal{D}_2$, and the game proceeds. In the second case, player II answers with a disjunctive must-transition in $\mathcal{D}_1$, bringing the game into a state where player I now must play a branch (a, t) of the chosen must-transition in $\mathcal{D}_1$. To this, player II must answer with a matching branch in the must-transition in $\mathcal{D}_2$, and then the game can proceed.

A *strategy for player I* hence consists of two partial mappings $\theta_1 : C_1 \rightharpoonup (\dashrightarrow_1 \cup \longrightarrow_2)$, $\theta_1' : C_1' \rightharpoonup \Sigma \times S_1$ such that

- if $c_1 = c_1^1 \dots c_1^n \in C_1$, $\theta_1(c_1)$ is defined, and $c_1^n = ((s_n, a_n, t_n), (s_n', a_n', t_n')) \in (\dashrightarrow_1 \times \dashrightarrow_2)$ or $c_1^n = ((s_n, N_n), (s_n', N_n'), a_n, t_n, a_n', t_n') \in (\longrightarrow_2 \times \longrightarrow_1 \times \Sigma \times S_1 \times \Sigma \times S_2)$, then
  - if $\theta_1(c_1) = (s, a, t) \in \dashrightarrow_1$, then $s = t_n$;
  - if $\theta_1(c_1) = (s, N) \in \longrightarrow_2$, then $s = t_n'$;
- if $c_1' = c_1''.((s, N), (s', N')) \in C_1'$ and $\theta_1'(c_1') = (a, t)$ is defined, then $(a, t) \in N'$.

This says that from an extended state in C_1, player I must choose a transition from one of the previous target states, and from a state in C_1', player I must choose a branch of the must-transition just chosen by player II.

If $\theta_1(\varepsilon)$ is defined, then

- if $\theta_1(\varepsilon) = (s, a, t) \in \dashrightarrow_1$, then $s \in S_1^0$;
- if $\theta_1(\varepsilon) = (s, N) \in \longrightarrow_2$, then $s \in S_2^0$.

A *strategy for player II* consists of two partial mappings $\theta_2 : C_2 \rightharpoonup (\dashrightarrow_2 \cup \longrightarrow_1)$, $\theta_2' : C_2' \rightharpoonup \Sigma \times S_2$ such that

- if $c_2 = c_2^1 \dots c_2^n.\tau \in C_2$ and $\theta_2(c_2)$ is defined, and $c_2^n = ((s_n, a_n, t_n), (s_n', a_n', t_n')) \in (\dashrightarrow_1 \times \dashrightarrow_2)$ or $c_2^n = ((s_n, N_n), (s_n', N_n'), a_n, t_n, a_n', t_n') \in (\longrightarrow_2 \times \longrightarrow_1 \times \Sigma \times S_1 \times \Sigma \times S_2)$, then
  - if $\tau = (s, a, t) \in \dashrightarrow_1$, then $\theta_2(c_2) = (s', a, t') \in \dashrightarrow_2$ with $s' = t_n'$;

– if $\tau = (s, N) \in \longrightarrow_2$, then $\theta_2(c_2) = (s', N') \in \longrightarrow_1$ with $s' = t_n$;

- if $c_2' = c_2''.((s, N), (s', N'), (a, t)) \in C_2'$ and $\theta_2'(c_2') = (a', t')$ is defined, then $(a', t') \in N$ and $a' = a$.

The sets of strategies for players I and II are denoted Θ_1 and Θ_2.

Let $(\theta_1, \theta_1') \in \Theta_1$, $(\theta_2, \theta_2') \in \Theta_2$, $c_1 \in C_1$, $c_2 \in C_2$, $c_1' \in C_1'$, and $c_2' \in C_2'$. We define the update functions:

- If $\theta_1(c_1)$ is defined, then $\mathsf{upd}(c_1) = c_1.\theta_1(c_1) \in C_2$.
- If $\theta_2(c_2)$ is defined, then $\mathsf{upd}(c_2) = c_2.\theta_2(c_2) \in C_1$ if $\theta_2(c_2) \in \dashrightarrow_2$ and $\mathsf{upd}(c_2) = c_2.\theta_2(c_2) \in C_1'$ if $\theta_2(c_2) \in \longrightarrow_1$.
- If $\theta_1'(c_1')$ is defined, then $\mathsf{upd}(c_1') = c_1'.\theta_1'(c_1') \in C_2'$.
- If $\theta_2'(c_2')$ is defined, then $\mathsf{upd}(c_2') = c_2'.\theta_2'(c_2') \in C_1$.

Hence a strategy pair $((\theta_1, \theta_1'), (\theta_2, \theta_2')) \in \Theta_1 \times \Theta_2$ induces, via the update functions, a finite or infinite string

$$\sigma((\theta_1, \theta_1'), (\theta_2, \theta_2')) \in C_1 \cup C_2 \cup C_1' \cup C_2' \cup ((\dashrightarrow_1 \times \dashrightarrow_2) \cup (\longrightarrow_2 \times \longrightarrow_1 \times \Sigma \times S_1 \times \Sigma \times S_2))^\omega .$$

Then $(\theta_1, \theta_1') \in \Theta_1$ is said to be *winning for player I* if $\sigma((\theta_1, \theta_1'), (\theta_2, \theta_2')) \in C_2 \cup C_2'$ for all $(\theta_2, \theta_2') \in \Theta_2$, and $(\theta_2, \theta_2') \in \Theta_2$ is *winning for player II* if $\sigma((\theta_1, \theta_1'), (\theta_2, \theta_2')) \in C_1 \cup C_1' \cup ((\dashrightarrow_1 \times \dashrightarrow_2) \cup (\longrightarrow_2 \times \longrightarrow_1 \times \Sigma \times S_1 \times \Sigma \times S_2))^\omega$ for all $(\theta_1, \theta_1') \in \Theta_1$. The game is determined, *i.e.* player I has a winning strategy iff player II does not.

We introduce a switching counter sc, similarly to the one of the preceding section. For $c_1 \in C_1$,

- $\mathsf{sc}(c_1) = 0$ iff $c_1 \in (\dashrightarrow_1 \times \dashrightarrow_2)^* \cup (\longrightarrow_2 \times \longrightarrow_1 \times \Sigma \times S_1 \times \Sigma \times S_2)^*$;
- $\mathsf{sc}(c_1) = 1$ iff $c_1 \in (\dashrightarrow_1 \times \dashrightarrow_2)^+ (\longrightarrow_2 \times \longrightarrow_1 \times \Sigma \times S_1 \times \Sigma \times S_2)^+ \cup (\longrightarrow_2 \times \longrightarrow_1 \times \Sigma \times S_1 \times \Sigma \times S_2)^+ (\dashrightarrow_1 \times \dashrightarrow_2)^+$;

etc., and for $c = c_1.c' \in C_2 \cup C_1' \cup C_2'$ such that $c_1 \in C_1$ is the longest C_1-prefix of c, $\mathsf{sc}(c) = \mathsf{sc}(c_1)$. We also copy Definition 33 to introduce k-switching and k-ready switching strategies in Θ_1, and denote again the subsets of k-switching strategies by Θ_1^k and of k-ready switching strategies by $\Theta_1^{k\text{-r}}$.

Proposition 36. *Let $k \geq 0$ and $\mathcal{D}_1, \mathcal{D}_2 \in \mathsf{DMTS}$. Then $\mathcal{D}_1 \leq_k \mathcal{D}_2$ iff player II has a winning strategy in the Θ_1^k-game on $\mathcal{D}_1, \mathcal{D}_2$, and $\mathcal{D}_1 \leq_k^{\mathrm{r}} \mathcal{D}_2$ iff player II has a winning strategy in the $\Theta_1^{k\text{-r}}$-game on $\mathcal{D}_1, \mathcal{D}_2$.*

Proof. Similar to the proof of Proposition 34. □

It is again sufficient to consider memory-less strategies for both players, *cf.* Remark 32.

10. Game-Based Proof of Theorem 26

We now show a game-based proof of Theorem 26 which relates $\leq_k$ with $\sim_k$ and $\leq_k^{\mathrm{r}}$ with $\sim_k^{\mathrm{r}}$. This is based on exposing an isomorphism between generalized simulation games on LTS and corresponding specification games on their embeddings into DMTS. Hence it can be used to show the more general result that any restriction $\Theta_1' \subseteq \Theta_1$ in the specification game yields a specification theory adequate for an equivalence relation defined on LTS by a similar restriction of the generalized simulation game.

Proof (of Theorem 26). We show that for $\mathcal{I}_1, \mathcal{I}_2 \in \mathsf{LTS}$, $\chi(\mathcal{I}_1) \leq_k \chi(\mathcal{I}_2)$ iff $\mathcal{I}_1 \sim_k \mathcal{I}_2$ and apply Lemma 14; the proof for the k-ready relations is similar.

The essence of the proof is that the simulation Θ_k-game on $\mathcal{I}_1$, $\mathcal{I}_2$ and the specification Θ_k-game on $\chi(\mathcal{I}_1)$, $\chi(\mathcal{I}_2)$ are isomorphic. We expose an injective mapping Φ, from extended states in the simulation game to extended states in the specification game, which essentially maps transitions in $\mathcal{I}_1$ to may-transitions in $\chi(\mathcal{I}_1)$ and transitions in $\mathcal{I}_2$ to must-transitions in $\chi(\mathcal{I}_2)$.

We then show that extended states outside the image of Φ are unreachable in any specification game, hence Φ is a bijection between extended states in the simulation game and "proper" extended states in the specification game.

Using this, we then extend Φ to an injective mapping from strategies in the simulation game to strategies in the specification game, and we show that strategies outside the image of Φ need not be considered. Also, Φ preserves and reflects the switching counter, and we show that a strategy θ_1 is winning for player I in the simulation game iff $\Phi(\theta_1)$ is winning for player I in the specification game.

Write $\mathcal{I}_1 = (S_1, s_1^0, T_1)$, $\mathcal{I}_2 = (S_2, s_2^0, T_2)$, $\chi(\mathcal{I}_1) = (S_1, \{s_1^0\}, \dashrightarrow_1, \longrightarrow_1)$, and $\chi(\mathcal{I}_2) = (S_2, \{s_2^0\}, \dashrightarrow_2, \longrightarrow_2)$. In this proof, we denote extended states and strategies in the specification game as in Sect. 9, whereas extended states and strategies in the game of Sect. 8 are denoted using tildes.

We define mappings $\Phi_1 : \tilde{C}_1 \to C_1$, $\Phi_2 : \tilde{C}_2 \to C_2$. Let $\phi_1 : (T_1 \times T_2 \cup T_2 \times T_1) \to ((\dashrightarrow_1 \times \dashrightarrow_2) \cup (\longrightarrow_2 \times \longrightarrow_1 \times \Sigma \times S_1 \times \Sigma \times S_2))$ and $\phi_2 : (T_1 \cup T_2) \to (\dashrightarrow_1 \cup \longrightarrow_2)$ be given by

$$\phi_1((s,a,t),(s',a',t')) = \begin{cases} ((s,a,t),(s',a',t')) & \text{if } (s,a,t) \in T_1\,, \\ ((s,\{(a,t)\}),(s',\{(a',t')\}),a',t',a,t) & \text{if } (s,a,t) \in T_2\,, \end{cases}$$

$$\phi_2(s,a,t) = \begin{cases} (s,a,t) & \text{if } (s,a,t) \in T_1\,, \\ (s,\{(a,t)\}) & \text{if } (s,a,t) \in T_2\,, \end{cases}$$

and for $\tilde{c}_1 = \tilde{c}_1^1 \dots \tilde{c}_1^n \in \tilde{C}_1$ and $\tilde{c}_2 = \tilde{c}_1.\tilde{c}_2' \in \tilde{C}_2$, define $\Phi_1(\tilde{c}_1) = \phi_1(\tilde{c}_1^1) \dots \phi_1(\tilde{c}_1^n)$ and $\Phi_2(\tilde{c}_2) = \Phi_1(\tilde{c}_1).\phi_2(\tilde{c}_2')$. We also define $\Phi_3 : (T_1 \times T_2 \cup T_2 \times T_1)^\omega \to (\longrightarrow_2 \times \longrightarrow_1 \times \Sigma \times S_1 \times \Sigma \times S_2)^\omega$ by $\Phi_3(d_1 d_2 \dots) = \phi_1(d_1)\phi_1(d_2) \dots$ and let $\Phi = \Phi_1 \cup \Phi_2 \cup \Phi_3$.

We call extended states in the image of Φ_1, Φ_2 *proper*, and we note that any reachable extended state in C_1 and C_2 is proper: Let $c_1 = c_1^1 \dots c_1^n \in C_1$ and

$j \in \{1, \ldots, n\}$ such that $c_1^j = ((s_j, N_j), (s'_j, N'_j), a_j, t_j, a'_j, t'_j) \in (\longrightarrow_2 \times \longrightarrow_1 \times \Sigma \times S_1 \times \Sigma \times S_2)$. Then $N_j = \{(b_j, u_j)\}$ and $N'_j = \{(b'_j, u'_j)\}$ for some $(s_j, b_j, u_j) \in T_2$ and $(s'_j, b'_j, u'_j) \in T_1$. Now if the extended state c_1 will be reached during any game, then $c_1^1 \ldots c_1^{j-1}.((s_j, N_j), (s'_j, N'_j)) \in C'_1$ must also have been reached, and then $(a_j, t_j) \in N'_j$ and $(a'_j, t'_j) \in N_j$ by the definition of strategies. But N'_j and N_j are one-element sets, so that we must have $a_j = b'_j$, $t_j = u'_j$, $a'_j = b_j$, and $t'_j = u_j$. Hence we can assume that if $c_1^j \in (\longrightarrow_2 \times \longrightarrow_1 \times \Sigma \times S_1 \times \Sigma \times S_2)$, then $c_1^j = ((s_j, \{(b_j, u_j)\}), (s'_j, \{(b'_j, u'_j)\}), b'_j, u'_j, b_j, u_j)$ for some $(s_j, b_j, u_j) \in T_2$ and $(s'_j, b'_j, u'_j) \in T_1$, *i.e.* $c_1^j = \phi_1((s_j, b_j, u_j), (s'_j, b'_j, u'_j))$.

The functions Φ_1 and Φ_2 are also injective, hence they are bijections onto the proper subsets of C_1 and C_2. We have shown that improper extended states are not reachable, hence strategies in Θ_1 and Θ_2 need not be defined on improper extended states.

Next we note that strategies $\theta'_1 : C'_1 \to \Sigma \times S_1$ and $\theta'_2 : C'_2 \to \Sigma \times S_2$ are unique: If $c'_1 = c''_1.((s, N), (s', N')) \in C'_1$ and $\theta'_1(c'_1) = (a, t)$ is defined, then $(a, t) \in N'$, but $N' = \{(b', u')\}$ is a one-element set, hence $a = b'$ and $t = u'$. If $\theta'_1(c'_1)$ is undefined, then the modification of θ'_1 which defines $\theta'_1(c'_1) = (b', u')$ is better for player I. The argument is similar for player II. We can henceforth assume that θ'_1 and θ'_2 always are the strategies defined above.

We extend the mappings Φ_1 and Φ_2 to strategies. Let $\tilde{\theta}_1 \in \tilde{\Theta}_1$, then $\Phi_1(\tilde{\theta}_1) = (\theta_1, \theta'_1)$, where θ'_1 is the unique strategy as above, $\theta_1(c_1) = \phi_2(\tilde{\theta}_1(\Phi_1^{-1}(c_1)))$ for any proper extended state $c_1 \in C_1$, and $\theta_1(c_1)$ undefined for c_1 improper. Similarly, for $\tilde{\theta}_2 \in \tilde{\Theta}_2$, $\Phi_2(\tilde{\theta}_2) = (\theta_2, \theta'_2)$, where θ'_2 is the unique player-II strategy, $\theta_2(c_2) = \phi_2(\tilde{\theta}_2(\Phi_2^{-1}(c_2)))$ for any proper extended state $c_2 \in C_2$, and $\theta_2(c_2)$ undefined for c_2 improper. The so-defined functions $\Phi_1 : \tilde{\Theta}_1 \to \Theta_1$, $\Phi_2 : \tilde{\Theta}_2 \to \Theta_2$ are injective, hence bijections onto their images, which consist precisely of the strategies which are the unique strategies on C'_1 and C'_2 and undefined on improper extended states in C_1 and C_2. Φ_1 also preserves and reflects switching counters: for all $\theta_1 \in \Theta_1$ and $k \geq 0$, $\tilde{\theta}_1 \in \tilde{\Theta}_1^k$ iff $\Phi_1(\tilde{\theta}_1) \in \Theta_1^k$ and $\tilde{\theta}_1 \in \tilde{\Theta}_1^{k\text{-r}}$ iff $\Phi_1(\tilde{\theta}_1) \in \Theta_1^{k\text{-r}}$.

Let $(\tilde{\theta}_1, \tilde{\theta}_2) \in \tilde{\Theta}_1 \times \tilde{\Theta}_2$ be a strategy pair; we will show that $\sigma(\Phi_1(\tilde{\theta}_1), \Phi_2(\tilde{\theta}_2)) = \Phi(\tilde{\sigma}(\tilde{\theta}_1, \tilde{\theta}_2))$. Let $\tilde{c}_1 \in \tilde{C}_1$, then

$$\begin{aligned}\Phi_2(\mathsf{upd}(\tilde{c}_1)) &= \Phi_2(\tilde{c}_1.\tilde{\theta}_1(\tilde{c}_1)) = \Phi_1(\tilde{c}_1).\phi_2(\tilde{\theta}_1(\tilde{c}_1)) \\ &= \Phi_1(\tilde{c}_1).\phi_2(\tilde{\theta}_1(\Phi_1^{-1}(\Phi_1(\tilde{c}_1)))) = \Phi_1(\tilde{c}_1).\theta_1(\Phi_1(\tilde{c}_1)) = \mathsf{upd}(\Phi_1(\tilde{c}_1))\,,\end{aligned}$$

where $\Phi_1(\tilde{\theta}_1) = (\theta_1, \theta'_1)$. This shows that Φ commutes with the update functions on $\tilde{C}_1$ and C_1. Similarly one can show that Φ commutes with the update functions on $\tilde{C}_2$ and C_2, and the updates on C'_1 and C'_2 are unique because θ'_1 and θ'_2 are the unique strategies. Together with $\Phi(\varepsilon) = \varepsilon$ and by induction, this implies that $\Phi(\tilde{\sigma}(\tilde{\theta}_1, \tilde{\theta}_2)) = \sigma(\Phi_1(\tilde{\theta}_1), \Phi_2(\tilde{\theta}_2))$.

We can now finish the proof. Let $k \geq 0$ and assume $\chi(\mathcal{I}_1) \not\leq_k \chi(\mathcal{I}_2)$, then player I has a winning strategy $(\theta_1, \theta'_1) \in \Theta_1^k$ in the specification Θ_1^k-game on $\chi(\mathcal{I}_1)$, $\chi(\mathcal{I}_2)$. We can assume that (θ_1, θ'_1) is in the image of Φ_1, hence there is $\tilde{\theta}_1 \in \tilde{\Theta}_1^k$ such that $\Phi_1(\tilde{\theta}_1) = (\theta_1, \theta'_1)$. We show that $\tilde{\theta}_1$ is winning for player I in

the $\tilde{\Theta}_1^k$-game on $\mathcal{I}_1$, $\mathcal{I}_2$, which will imply $\mathcal{I}_1 \not\sim_k \mathcal{I}_2$. Let $\tilde{\theta}_2 \in \tilde{\Theta}_2$, then

$$\tilde{\sigma}(\tilde{\theta}_1, \tilde{\theta}_2) = \Phi^{-1}(\sigma(\Phi_1(\tilde{\theta}_1), \Phi_2(\tilde{\theta}_2))) \in \Phi^{-1}(C_2) \subseteq \tilde{C}_2 \,.$$

Now assume that $\mathcal{I}_1 \not\sim_k \mathcal{I}_2$ and let $\tilde{\theta}_1 \in \tilde{\Theta}_1^k$ be a winning strategy for player I in the $\tilde{\Theta}_1^k$-game on $\mathcal{I}_1$, $\mathcal{I}_2$. Let $(\theta_1, \theta_1') = \Phi(\tilde{\theta}_1)$, we show that (θ_1, θ_1') is winning for player I in the Θ_1^k-game on $\chi(\mathcal{I}_1)$, $\chi(\mathcal{I}_2)$. Let $(\theta_2, \theta_2') \in \Theta_2$, then we can assume that there is $\tilde{\theta}_2 \in \tilde{\Theta}_2$ such that $\Phi_2(\tilde{\theta}_2) = (\theta_2, \theta_2')$, and

$$\sigma((\theta_1, \theta_1'), (\theta_2, \theta_2')) = \Phi(\tilde{\sigma}(\tilde{\theta}_1, \tilde{\theta}_2) \in \Phi(\tilde{C}_2) \subseteq C_2 \,,$$

hence $\chi(\mathcal{I}_1) \not\leq_k \chi(\mathcal{I}_2)$. □

11. Conclusion

We have in this paper extracted a reasonable and general notion of (behavioral) specification theory, based on previous work by a number of authors on concrete specification theories in different contexts and on the well-established notions of characteristic formulae, adequacy and expressivity.

Using this general concept of specification theory, we have introduced new concrete specification theories, based on disjunctive modal transition systems, for most equivalences in van Glabbeek's linear-time–branching-time spectrum. Previously, only specification theories for bisimilarity have been available, and recent work by Vogler *et al.* calls for work on specification theories for failure equivalence. Both failure equivalence and bisimilarity are part of the linear-time–branching-time spectrum, as are nested simulation equivalence, impossible-futures equivalence, and many other useful relations. We develop specification theories for all branching equivalences in the spectrum, but we miss some of the linear equivalences; notably, possible futures and ready trace equivalence are missing. We believe that these can be captured by small modifications to our setting, but leave this for future work.

Our new specification theories should be useful for example in the setting of the failure semantics of Vogler *et al.*, but also in many other contexts where bisimilarity is not the right equivalence to consider. Using our own previous work on the quantitative linear-time–branching-time spectrum and on quantitative specification theories for bisimilarity, we also plan to lift our work presented here to the quantitative setting.

Specification theories for bisimilarity admit notions of conjunction and composition which enable compositional design and verification, and also the specification theories of Vogler *et al.* have (different) such notions. Using the game-based setting, we believe one can define general notions of conjunction and composition defined by games played on the involved disjunctive modal transition systems. This is left for future work.

References

[1] Luca Aceto, Ignacio Fábregas, David de Frutos-Escrig, Anna Ingólfsdóttir, and Miguel Palomino. On the specification of modal systems. *Sci. Comput. Program.*, 78(12):2468–2487, 2013.

[2] Luca Aceto, Wan Fokkink, Rob J. van Glabbeek, and Anna Ingólfsdóttir. Nested semantics over finite trees are equationally hard. *Inf. Comput.*, 191(2):203–232, 2004.

[3] Luca Aceto, Anna Ingólfsdóttir, Kim G. Larsen, and Jiří Srba. *Reactive Systems*. Cambridge Univ. Press, 2007.

[4] Adam Antonik, Michael Huth, Kim G. Larsen, Ulrik Nyman, and Andrzej Wąsowski. 20 years of modal and mixed specifications. *Bull. EATCS*, 95:94–129, 2008.

[5] Sebastian S. Bauer, Alexandre David, Rolf Hennicker, Kim G. Larsen, Axel Legay, Ulrik Nyman, and Andrzej Wąsowski. Moving from specifications to contracts in component-based design. In Juan de Lara and Andrea Zisman, editors, *FASE*, volume 7212 of *Lect. Notes Comput. Sci.*, pages 43–58. Springer-Verlag, 2012.

[6] Sebastian S. Bauer, Uli Fahrenberg, Line Juhl, Kim G. Larsen, Axel Legay, and Claus Thrane. Weighted modal transition systems. *Form. Meth. Syst. Design*, 42(2):193–220, 2013.

[7] Sebastian S. Bauer, Line Juhl, Kim G. Larsen, Axel Legay, and Jiří Srba. Extending modal transition systems with structured labels. *Math. Struct. Comput. Sci.*, 22(4):581–617, 2012.

[8] Nikola Beneš, Ivana Černá, and Jan Křetínský. Modal transition systems: Composition and LTL model checking. In Tevfik Bultan and Pao-Ann Hsiung, editors, *ATVA*, volume 6996 of *Lect. Notes Comput. Sci.*, pages 228–242. Springer-Verlag, 2011.

[9] Nikola Beneš, Benoît Delahaye, Uli Fahrenberg, Jan Křetínský, and Axel Legay. Hennessy-Milner logic with greatest fixed points as a complete behavioural specification theory. In Pedro R. D'Argenio and Hernán C. Melgratti, editors, *CONCUR*, volume 8052 of *Lect. Notes Comput. Sci.*, pages 76–90. Springer-Verlag, 2013.

[10] Nathalie Bertrand, Axel Legay, Sophie Pinchinat, and Jean-Baptiste Raclet. Modal event-clock specifications for timed component-based design. *Sci. Comput. Program.*, 77(12):1212–1234, 2012.

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

[12] Stephen D. Brookes, C. A. R. Hoare, and A. W. Roscoe. A theory of communicating sequential processes. *J. ACM*, 31(3):560–599, 1984.

[13] Ferenc Bujtor, Lev Sorokin, and Walter Vogler. Testing preorders for dMTS: Deadlock- and the new deadlock/divergence-testing. In *ACSD*, pages 60–69. IEEE Computer Society, 2015.

[14] Ferenc Bujtor and Walter Vogler. Failure semantics for modal transition systems. *ACM Trans. Embedded Comput. Syst.*, 14(4):67, 2015.

[15] Benoît Caillaud and Jean-Baptiste Raclet. Ensuring reachability by design. In Abhik Roychoudhury and Meenakshi D'Souza, editors, *ICTAC*, volume 7521 of *Lect. Notes Comput. Sci.*, pages 213–227. Springer-Verlag, 2012.

[16] Alexandre David, Kim G. Larsen, Axel Legay, Ulrik Nyman, Louis-Marie Traonouez, and Andrzej Wasowski. Real-time specifications. *STTT*, 17(1):17–45, 2015.

[17] Uli Fahrenberg and Axel Legay. General quantitative specification theories with modal transition systems. *Acta Inf.*, 51(5):261–295, 2014.

[18] Uli Fahrenberg and Axel Legay. The quantitative linear-time-branching-time spectrum. *Theor. Comput. Sci.*, 538:54–69, 2014.

[19] Uli Fahrenberg and Axel Legay. A linear-time-branching-time spectrum of behavioral specification theories. In Bernhard Steffen, Christel Baier, Mark van den Brand, Johann Eder, Mike Hinchey, and Tiziana Margaria, editors, *SOFSEM*, volume 10139 of *Lect. Notes Comput. Sci.*, pages 49–61. Springer-Verlag, 2017.

[20] Uli Fahrenberg, Axel Legay, and Louis-Marie Traonouez. Structural refinement for the modal nu-calculus. In Gabriel Ciobanu and Dominique Méry, editors, *ICTAC*, volume 8687 of *Lect. Notes Comput. Sci.*, pages 169–187. Springer-Verlag, 2014.

[21] Jan Friso Groote and Frits W. Vaandrager. Structured operational semantics and bisimulation as a congruence. *Inf. Comput.*, 100(2):202–260, 1992.

[22] Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *J. ACM*, 32(1):137–161, 1985.

[23] C. A. R. Hoare. Communicating sequential processes. *Commun. ACM*, 21(8):666–677, 1978.

[24] Kim G. Larsen. A context dependent equivalence between processes. *Theor. Comput. Sci.*, 49:184–215, 1987.

[25] Kim G. Larsen. Modal specifications. In Joseph Sifakis, editor, *Automatic Verification Methods for Finite State Systems*, volume 407 of *Lect. Notes Comput. Sci.*, pages 232–246. Springer-Verlag, 1989.

[26] Kim G. Larsen. Ideal specification formalism = expressivity + compositionality + decidability + testability + ... In Jos C. M. Baeten and Jan Willem Klop, editors, *CONCUR*, volume 458 of *Lect. Notes Comput. Sci.*, pages 33–56. Springer-Verlag, 1990.

[27] Kim G. Larsen. Proof systems for satisfiability in Hennessy-Milner logic with recursion. *Theor. Comput. Sci.*, 72(2&3):265–288, 1990.

[28] Kim G. Larsen and Arne Skou. Bisimulation through probabilistic testing. In *POPL*, pages 344–352. ACM Press, 1989.

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

[30] Robin Milner. Calculi for synchrony and asynchrony. *Theor. Comput. Sci.*, 25, 1983.

[31] David Michael Ritchie Park. Concurrency and automata on infinite sequences. In Peter Deussen, editor, *TCS*, volume 104 of *Lect. Notes Comput. Sci.*, pages 167–183. Springer-Verlag, 1981.

[32] Amir Pnueli. Linear and branching structures in the semantics and logics of reactive systems. In Wilfried Brauer, editor, *ICALP*, volume 194 of *Lect. Notes Comput. Sci.*, pages 15–32. Springer-Verlag, 1985.

[33] Jean-Baptiste Raclet. Residual for component specifications. *Electr. Notes Theor. Comput. Sci.*, 215:93–110, 2008.

[34] Jean-Baptiste Raclet, Eric Badouel, Albert Benveniste, Benoît Caillaud, Axel Legay, and Roberto Passerone. A modal interface theory for component-based design. *Fund. Inf.*, 108(1-2), 2011.

[35] Colin Stirling. Modal and temporal logics for processes. In Faron Moller and Graham M. Birtwistle, editors, *Banff Higher Order Workshop*, volume 1043 of *Lect. Notes Comput. Sci.*, pages 149–237. Springer-Verlag, 1995.

[36] Rob J. van Glabbeek. The linear time – branching time spectrum I. In Jan A. Bergstra, Alban Ponse, and Scott A. Smolka, editors, *Handbook of Process Algebra*, Chapter 1, pages 3–99. Elsevier, 2001.

[37] Walter Vogler. Failures semantics and deadlocking of modular Petri nets. *Acta Inf.*, 26(4):333–348, 1989.

[38] Walter Vogler. *Modular Construction and Partial Order Semantics of Petri Nets*, volume 625 of *Lect. Notes Comput. Sci.* Springer-Verlag, 1992.