

Divide and Congruence III: From Decomposition of Modal Formulas to Preservation of Stability and Divergence

Wan Fokkink

Department of Computer Science
Vrije University Amsterdam, The Netherlands
w.j.fokkink@vu.nl

Rob van Glabbeek

Data61, CSIRO, Sydney, Australia
School of Computer Science and Engineering
University of New South Wales, Sydney, Australia
rvg@cs.stanford.edu

Bas Luttik

Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, The Netherlands
s.p.luttik@tue.nl

In two earlier papers we derived congruence formats with regard to transition system specifications for weak semantics on the basis of a decomposition method for modal formulas. The idea is that a congruence format for a semantics must ensure that the formulas in the modal characterisation of this semantics are always decomposed into formulas that are again in this modal characterisation. The stability and divergence requirements that are imposed on many of the known weak semantics have so far been outside the realm of this method. Stability refers to the absence of a τ -transition. We show, using the decomposition method, how congruence formats can be relaxed for weak semantics that are stability-respecting. This relaxation for instance brings the priority operator within the range of the stability-respecting branching bisimulation format. Divergence, which refers to the presence of an infinite sequence of τ -transitions, escapes the inductive decomposition method. We circumvent this problem by proving that a congruence format for a stability-respecting weak semantics is also a congruence format for its divergence-preserving counterpart.

1 Introduction

Structural operational semantics [25] provides specification languages with an interpretation. It generates a labelled transition system, in which states are the closed terms over a (single-sorted, first-order) signature, and transitions between states carry labels. The transitions between states are obtained from a transition system specification (TSS), which consists of a set of proof rules called transition rules. States in labelled transition systems can be identified by a wide range of behavioural equivalences, based on e.g. branching structure or decorated versions of execution sequences. VAN GLABBEK [14] classified so-called weak semantics, which take into account the internal action τ . A significant number of the weak semantics based on a bisimulation relation carry a stability or divergence requirement. Stability refers to the absence of a τ -transition and divergence to the presence of an infinite sequence of τ -transitions.

In general a behavioural equivalence induced by a TSS is not guaranteed to be a congruence, i.e. the equivalence class of a term $f(p_1, \dots, p_n)$ need not be determined by f and the equivalence classes of its arguments p_1, \dots, p_n . Being a congruence is an important property, for instance in order to fit the equivalence into an axiomatic framework. Respecting stability or preserving divergence sometimes needs to be imposed in order to obtain a congruence relation, for example in case of the priority operator [1].

Behavioural equivalences can be characterised in terms of the observations that an experimenter could make during a session with a process. Modal logic captures such observations. A modal character-

isation of an equivalence on processes consists of a class C of modal formulas such that two processes are equivalent if and only if they satisfy the same formulas in C . For instance, Hennessy-Milner logic [21] constitutes a modal characterisation of (strong) bisimilarity. A cornerstone for the current paper is the work in [5] to decompose formulas from Hennessy-Milner logic with respect to a structural operational semantics in the ntyft format [19] without lookahead. Here the *decomposition* of a modal formula ϕ w.r.t. a term $t = f(p_1, \dots, p_n)$ is a selection of n -tuples of modal formulas, one of which needs to be satisfied by the processes p_i in order for t to satisfy ϕ . Based on this method, congruence formats can be generated for process semantics from their modal characterisation, to ensure that the equivalence is a congruence. Such formats help to avoid repetitive congruence proofs and obtain insight into the congruence property. Key idea is that congruence is ensured if formulas from the modal characterisation of a semantics under consideration are always decomposed into formulas that are again in this modal characterisation. This approach was extended to weak semantics in [13, 10]. It crosses the borders between process algebra, structural operational semantics, process semantics, and modal logic.

Here we expand the latter work to weak semantics that respect stability or preserve divergence. We focus on *branching bisimilarity* and *rooted branching bisimilarity* [18] and consider for each a stability-respecting and two divergence-preserving variants. Divergence-preserving branching bisimilarity [18] is the coarsest congruence relation for the parallel composition operator that only equates processes satisfying the same formulas from the well-known temporal logic CTL^* minus the next-time operator X [17]. With regard to stability the expansion is relatively straightforward: we extend the modal characterisation of the semantics with one clause to capture that a semantics is stability-respecting, and study the decomposition of this additional clause. Next we show how the congruence formats for branching bisimilarity and rooted branching bisimilarity from [13] can be relaxed, owing to the extended modal characterisation for stability-respecting branching bisimilarity. Notably, the transition rules for the priority operator are within the more relaxed formats.

The divergence preservation property escapes the inductive decomposition method, as it concerns an infinite sequence of τ -transitions. We overcome this problem by presenting a general framework for lifting congruence formats from some weak semantics \approx to a finer weak semantics \sim , with the aim to turn congruence formats for stability-respecting weak semantics into congruence formats for their divergence-preserving counterparts. It roughly works as follows. Consider a TSS P in a congruence format for \approx , and for example a unary function symbol f . A new TSS is created out of P , in which many occurrences of τ in transition rules are suppressed by renaming them into some fresh action name. Furthermore, processes are provided with what we call an oracle transition that reveals some pertinent information on the behaviour of the process, such as whether it can diverge. In this way we ensure that \approx and \sim coincide on the transformed TSS. For each closed term p of P we moreover introduce a constant \hat{p} in the transformed TSS such that $p \sim q$ implies $\hat{p} \sim \hat{q}$, and so $\hat{p} \approx \hat{q}$. We take care that this entire transformation preserves the congruence format for \approx . So $\hat{p} \approx \hat{q}$ implies $f(\hat{p}) \approx f(\hat{q})$, and hence, since \approx and \sim coincide on the transformed TSS, $f(\hat{p}) \sim f(\hat{q})$. Finally we cast $f(\hat{p})$ and $f(\hat{q})$ back to processes strongly bisimilar to $f(p)$ and $f(q)$ in the original TSS P , taking care that \sim is preserved. Thus we have gone full circle: $p \sim q$ implies $f(p) \sim f(q)$. This implies that the congruence format for \approx is also a congruence format for \sim . We show four instances where this method can be applied. In two cases \approx is stability-respecting branching bisimilarity and in two cases rooted stability-respecting branching bisimilarity, while in \sim stability is replaced by two different forms of divergence. Thus it can be concluded that the congruence format for stability-respecting branching bisimilarity is also applicable to divergence-preserving as well as weakly divergence-preserving branching bisimilarity; and likewise for the rooted counterparts of these semantics.

An extended abstract of this paper appeared as [11].

2 Preliminaries

This section recalls the basic notions of labelled transition systems and weak semantics and defines stability-respecting and divergence-preserving branching bisimilarity (Sect. 2.1) as well as a modal characterisation of stability-respecting branching bisimilarity (Sect. 2.2). It also presents a brief introduction to structural operational semantics (Sect. 2.3) and recalls some syntactic restrictions on transition rules (Sect. 2.4). We define the notion of a ruloid (Sect. 2.5). Finally, the modal decomposition method for weak semantics is explained (Sect. 2.6).

2.1 Stability-respecting and divergence-preserving branching bisimilarity

A *labelled transition system (LTS)* is a triple $(\mathbb{P}, \text{Act}, \rightarrow)$, with \mathbb{P} a set of *processes*, Act a set of *actions*, and $\rightarrow \subseteq \mathbb{P} \times \text{Act} \times \mathbb{P}$. We normally let $\text{Act} = A \cup \{\tau\}$ where τ is an *internal action* and A some set of external or observable actions not containing τ . We write A_τ for $A \cup \{\tau\}$. We use p, q to denote processes, α, β, γ for elements of A_τ , and a, b for elements of A . We write $p \xrightarrow{\alpha} q$ for $(p, \alpha, q) \in \rightarrow$, $p \xrightarrow{\alpha}$ for $\exists q \in \mathbb{P} : p \xrightarrow{\alpha} q$, and $p \not\xrightarrow{\alpha}$ for $\neg(p \xrightarrow{\alpha})$. Furthermore, $\xRightarrow{\varepsilon}$ denotes the transitive-reflexive closure of $\xrightarrow{\tau}$.

Definition 1 Let $\mathcal{B} \subseteq \mathbb{P} \times \mathbb{P}$ be a symmetric relation.

- \mathcal{B} is a *branching bisimulation* if $p \mathcal{B} q$ and $p \xrightarrow{\alpha} p'$ implies that either $\alpha = \tau$ and $p' \mathcal{B} q$, or $q \xRightarrow{\varepsilon} q' \xrightarrow{\alpha} q''$ for some q' and q'' with $p \mathcal{B} q'$ and $p' \mathcal{B} q''$.
- \mathcal{B} is *stability-respecting* if $p \mathcal{B} q$ and $p \not\xrightarrow{\tau}$ implies that $q \xRightarrow{\varepsilon} q' \not\xrightarrow{\tau}$ for some q' with $p \mathcal{B} q'$.
- \mathcal{B} is *divergence-preserving* if it satisfies the following condition:
 (D) if $p \mathcal{B} q$ and there is an infinite sequence of processes $(p_k)_{k \in \mathbb{N}}$ such that $p = p_0$, $p_k \xrightarrow{\tau} p_{k+1}$ and $p_k \mathcal{B} q$ for all $k \in \mathbb{N}$, then there exists an infinite sequence of processes $(q_\ell)_{\ell \in \mathbb{N}}$ such that $q = q_0$, $q_\ell \xrightarrow{\tau} q_{\ell+1}$ for all $\ell \in \mathbb{N}$, and $p_k \mathcal{B} q_\ell$ for all $k, \ell \in \mathbb{N}$.

The definition of a *weakly divergence-preserving* relation is obtained by omitting the condition “and $p_k \mathcal{B} q_\ell$ for all $k, \ell \in \mathbb{N}$ ”. (The condition “and $p_k \mathcal{B} q$ for all $k \in \mathbb{N}$ ” is then redundant.)

Processes p, q are *branching bisimilar*, denoted $p \Leftrightarrow_b q$, if there exists a branching bisimulation \mathcal{B} with $p \mathcal{B} q$. They are *stability-respecting*, *divergence-preserving* or *weakly divergence-preserving* branching bisimilar, denoted $p \Leftrightarrow_b^s q$, $p \Leftrightarrow_b^\Delta q$ or $p \Leftrightarrow_b^{\Delta^\top} q$, if moreover \mathcal{B} is stability-respecting, divergence-preserving or weakly divergence-preserving, respectively.

We have $\Leftrightarrow_b \supset \Leftrightarrow_b^s \supset \Leftrightarrow_b^{\Delta^\top} \supset \Leftrightarrow_b^\Delta$.

The relations \Leftrightarrow_b , \Leftrightarrow_b^s , \Leftrightarrow_b^Δ and $\Leftrightarrow_b^{\Delta^\top}$ are equivalences [2, 14, 16]. However, they are not *congruences* with respect to most process algebras from the literature, meaning that the equivalence class of a process $f(p_1, \dots, p_n)$, with f an n -ary function symbol, is not always determined by the equivalence classes of its arguments, i.e. the processes p_1, \dots, p_n . Therefore an additional rootedness condition is imposed.

Definition 2 *Rooted branching bisimilarity*, \Leftrightarrow_{rb} , is the largest symmetric relation on \mathbb{P} such that $p \Leftrightarrow_{rb} q$ and $p \xrightarrow{\alpha} p'$ implies that $q \xrightarrow{\alpha} q'$ for some q' with $p' \Leftrightarrow_b q'$. Likewise, *rooted stability-respecting*, *divergence-preserving* or *weakly divergence-preserving branching bisimilarity*, denoted by $p \Leftrightarrow_{rb}^s q$, $p \Leftrightarrow_{rb}^\Delta q$ or $p \Leftrightarrow_{rb}^{\Delta^\top} q$, is the largest symmetric relation \mathcal{R} on \mathbb{P} such that $p \mathcal{R} q$ and $p \xrightarrow{\alpha} p'$ implies that $q \xrightarrow{\alpha} q'$ for some q' with $p' \Leftrightarrow_b^s q'$, $p' \Leftrightarrow_b^\Delta q'$ or $p' \Leftrightarrow_b^{\Delta^\top} q'$, respectively.

Our main aim is to develop congruence formats for stability-respecting and divergence-preserving semantics. These congruence formats will impose syntactic restrictions on the transition rules that are used to generate the underlying LTS. The congruence formats will be determined using the characterising modal logics for the semantics.

We will sometimes refer to strong bisimulation semantics, which ignores the special status of the τ .

Definition 3 A symmetric relation $\mathcal{B} \subseteq \mathbb{P} \times \mathbb{P}$ is a *strong bisimulation* if $p \mathcal{B} q$ and $p \xrightarrow{a} p'$ implies that $q \xrightarrow{a} q'$ for some q' with $p' \mathcal{B} q'$. Processes p, q are *strongly bisimilar*, denoted $p \Leftrightarrow q$, if there exists a strong bisimulation \mathcal{B} with $p \mathcal{B} q$.

The various notions of bisimilarity defined above are examples of so-called behavioural equivalences. For a general formulation of the results in Section 4, it is convenient to formally define a notion of behavioural equivalence that includes at least the examples above. Note that a common feature of their definitions is that they associate with every LTS $G = (\mathbb{P}_G, \text{Act}_G, \rightarrow_G)$ a binary relation \sim_G . (For instance, in the case of strong bisimilarity, the relation \sim_G associated with G is defined as the binary relation $\Leftrightarrow \subseteq \mathbb{P}_G \times \mathbb{P}_G$ such that $p \Leftrightarrow q$ if (and only if) there exists a strong bisimulation $\mathcal{B} \subseteq \mathbb{P}_G \times \mathbb{P}_G$ such that $p \mathcal{B} q$.) One may, thus, think of a behavioural equivalence as a family of binary relations indexed by LTSs. It turns out that we need to impose just one extra condition on such families to arrive at a suitable formalisation of the notion of behavioural equivalence. The condition states that the relation associated with the disjoint union of two LTSs restricted to one of the components coincides with the relation associated with that component.

Definition 4 Two LTSs $G = (\mathbb{P}_G, \text{Act}_G, \rightarrow_G)$ and $H = (\mathbb{P}_H, \text{Act}_H, \rightarrow_H)$ are called *disjoint* if $\mathbb{P}_G \cap \mathbb{P}_H = \emptyset$. In that case $G \uplus H$ denotes their union $(\mathbb{P}_G \cup \mathbb{P}_H, \text{Act}_G \cup \text{Act}_H, \rightarrow_G \cup \rightarrow_H)$.

A *behavioural equivalence* \sim on LTSs is a family of equivalence relations \sim_G , one for every LTS G , such that for each pair of disjoint LTSs $G = (\mathbb{P}_G, \text{Act}_G, \rightarrow_G)$ and H we have $g \sim_G g' \Leftrightarrow g \sim_{G \uplus H} g'$ for any $g, g' \in \mathbb{P}_G$.

The notions of bisimilarity defined above clearly qualify as behavioural equivalences. Given two behavioural equivalences \sim and \approx , we write $\sim \subseteq \approx$ iff $\sim_G \subseteq \approx_G$ for each LTS G .

2.2 Modal logic

Modal logic formulas express properties on the behaviour of processes in an LTS. Following [14], we extend Hennessy-Milner logic [21] with the modal connectives $\langle \varepsilon \rangle \varphi$ and $\langle \hat{\tau} \rangle \varphi$, expressing that a process can perform zero or more, respectively zero or one, τ -transitions to a process where φ holds.

Definition 5 The class \mathbb{O} of *modal formulas* is defined as follows, where I ranges over all index sets and α over A_τ :

$$\mathbb{O} \quad \varphi ::= \bigwedge_{i \in I} \varphi_i \mid \neg \varphi \mid \langle \alpha \rangle \varphi \mid \langle \varepsilon \rangle \varphi \mid \langle \hat{\tau} \rangle \varphi$$

We use abbreviations \top for the empty conjunction, $\varphi_1 \wedge \varphi_2$ for $\bigwedge_{i \in \{1,2\}} \varphi_i$, $\varphi \langle \alpha \rangle \varphi'$ for $\varphi \wedge \langle \alpha \rangle \varphi'$, and $\varphi \langle \hat{\tau} \rangle \varphi'$ for $\varphi \wedge \langle \hat{\tau} \rangle \varphi'$.

$p \models \varphi$ denotes that process p satisfies formula φ . The first two operators represent the standard Boolean operators conjunction and negation. By definition, $p \models \langle \alpha \rangle \varphi$ if $p \xrightarrow{\alpha} p'$ for some p' with $p' \models \varphi$, $p \models \langle \varepsilon \rangle \varphi$ if $p \xRightarrow{\varepsilon} p'$ for some p' with $p' \models \varphi$, and $p \models \langle \hat{\tau} \rangle \varphi$ if either $p \models \varphi$ or $p \xrightarrow{\tau} p'$ for some p' with $p' \models \varphi$.

For each $L \subseteq \mathbb{O}$, we write $p \sim_L q$ if p and q satisfy the same formulas in L . We say that L is a *modal characterisation* of some behavioural equivalence \sim if \sim_L coincides with \sim . We write $\varphi \equiv \varphi'$ if $p \models \varphi \Leftrightarrow p \models \varphi'$ for all processes p . The class L^\equiv denotes the closure of $L \subseteq \mathbb{O}$ under \equiv . Trivially, $p \sim_L q \Leftrightarrow p \sim_{L^\equiv} q$.

Definition 6 [14] The subclasses \mathbb{O}_b and \mathbb{O}_{rb} of \mathbb{O} are defined as follows, where a ranges over A and α over A_τ :

$$\begin{aligned} \mathbb{O}_b \quad \varphi &::= \bigwedge_{i \in I} \varphi_i \mid \neg \varphi \mid \langle \varepsilon \rangle (\varphi \langle \hat{\tau} \rangle \varphi) \mid \langle \varepsilon \rangle (\varphi \langle a \rangle \varphi) \\ \mathbb{O}_{rb} \quad \bar{\varphi} &::= \bigwedge_{i \in I} \bar{\varphi}_i \mid \neg \bar{\varphi} \mid \langle \alpha \rangle \varphi \mid \varphi \quad (\varphi \in \mathbb{O}_b) \end{aligned}$$

\mathbb{O}_b and \mathbb{O}_{rb} are modal characterisations of \leftrightarrow_b and \leftrightarrow_{rb} , respectively (see [13]).

The idea behind stability is: (I) if $p \leftrightarrow_b^s q$ and $p \xRightarrow{\varepsilon} p' \not\xrightarrow{\tau}$ with $p \leftrightarrow_b^s p'$, then $q \xRightarrow{\varepsilon} q' \not\xrightarrow{\tau}$ with $q \leftrightarrow_b^s q'$. In the definition of \leftrightarrow_b^s this was formulated more weakly: (II) if $p \leftrightarrow_b^s q$ and $p \not\xrightarrow{\tau}$, then $q \xRightarrow{\varepsilon} q' \not\xrightarrow{\tau}$ with $q \leftrightarrow_b^s q'$. To argue that formulations (I) and (II) are equivalent, suppose $p \leftrightarrow_b^s q$ and $p \xRightarrow{\varepsilon} p' \not\xrightarrow{\tau}$ with $p \leftrightarrow_b^s p'$. Clearly $q \xRightarrow{\varepsilon} q''$ with $p' \leftrightarrow_b^s q''$. Now the weaker property (II) yields $q'' \xRightarrow{\varepsilon} q' \not\xrightarrow{\tau}$ with $q'' \leftrightarrow_b^s q'$. So $q \xRightarrow{\varepsilon} q' \not\xrightarrow{\tau}$ with $q \leftrightarrow_b^s q'$. That formulations (I) and (II) coincide is important to grasp the following modal characterisation of stability-respecting branching bisimilarity, because the additional clause at the end of the definition of \mathbb{O}_b^s is based on formulation (I).

Definition 7 [14] The subclasses \mathbb{O}_b^s and \mathbb{O}_{rb}^s of \mathbb{O} are defined as follows:

$$\begin{aligned} \mathbb{O}_b^s \quad \varphi &::= \bigwedge_{i \in I} \varphi_i \mid \neg \varphi \mid \langle \varepsilon \rangle (\varphi \langle \hat{\tau} \rangle \varphi) \mid \langle \varepsilon \rangle (\varphi \langle a \rangle \varphi) \mid \langle \varepsilon \rangle (\neg \langle \tau \rangle \top \wedge \bar{\varphi}) \quad (\bar{\varphi} \in \mathbb{O}_{rb}^s) \\ \mathbb{O}_{rb}^s \quad \bar{\varphi} &::= \bigwedge_{i \in I} \bar{\varphi}_i \mid \neg \bar{\varphi} \mid \langle \alpha \rangle \varphi \mid \varphi \quad (\varphi \in \mathbb{O}_b^s) \end{aligned}$$

The additional clause $\langle \varepsilon \rangle (\neg \langle \tau \rangle \top \wedge \bar{\varphi})$ in the definition of \mathbb{O}_b^s expresses stability. The first part $\langle \varepsilon \rangle (\neg \langle \tau \rangle \top \dots)$ captures $p \xRightarrow{\varepsilon} p' \not\xrightarrow{\tau}$, while the second part $\dots \wedge \bar{\varphi}$ captures the stability-respecting branching bisimulation class of p' . Note that since $p' \not\xrightarrow{\tau}$, unrooted and rooted stability-respecting branching bisimilarity coincide for p' , which allows us to take the second part from \mathbb{O}_{rb}^s .

The proof of the following theorem is presented in the appendix.

Theorem 8 $p \leftrightarrow_b^s q \Leftrightarrow p \sim_{\mathbb{O}_b^s} q$ and $p \leftrightarrow_{rb}^s q \Leftrightarrow p \sim_{\mathbb{O}_{rb}^s} q$, for all $p, q \in \mathbb{P}$.

2.3 Structural operational semantics

A *signature* is a set Σ of function symbols f with arity $ar(f)$. A function symbol of arity 0 is called a *constant*. Let V be an infinite set of variables, with typical elements x, y, z ; we assume $|\Sigma|, |A| \leq |V|$. A syntactic object is *closed* if it does not contain any variables. The sets $\mathbb{T}(\Sigma)$ and $\mathbb{T}(\Sigma)$ of terms over Σ and V and closed terms over Σ , respectively, are defined as usual; t, u, v, w denote terms, p, q denote closed terms, and $var(t)$ is the set of variables that occur in term t . A substitution σ is a partial function from V to $\mathbb{T}(\Sigma)$. A closed substitution is a total function from V to closed terms. The domain of substitutions is extended to $\mathbb{T}(\Sigma)$ as usual.

Structural operational semantics [25] generates an LTS in which the processes are the closed terms. The labelled transitions between processes are obtained from a transition system specification, which consists of a set of proof rules called transition rules.

Definition 9 A (positive or negative) *literal* is an expression $t \xrightarrow{\alpha} u$ or $t \not\xrightarrow{\alpha}$. A (transition) *rule* is of the form $\frac{H}{\lambda}$ with H a set of literals called the *premises*, and λ a literal called the *conclusion*; the terms at the left- and right-hand side of λ are called the *source* and *target* of the rule, respectively. A rule $\frac{\emptyset}{\lambda}$ is also written λ . A rule is *standard* if it has a positive conclusion. A *transition system specification (TSS)*, written (Σ, Act, R) , consists of a signature Σ , a set of actions Act , and a collection R of transition rules over Σ . A TSS is *standard* if all its rules are.

A TSS is meant to specify an LTS in which the transitions are the closed positive literals that can be proved using the rules of the TSS. It is straightforward to associate an appropriate notion of provability in the special case of standard TSSs with only positive premises. In the general case, in which the rules of the TSS may have negative premises, consistency is a concern. Literals $t \xrightarrow{\alpha} u$ and $t \not\xrightarrow{\alpha}$ are said to *deny* each other; a notion of provability associated with TSSs is *consistent* if it is not possible to prove two literals that deny each other. To arrive at a consistent notion of provability in the general case, we proceed in two steps: first we define the notion of *irredundant proof*, which on a standard TSS does not allow the derivation of negative literals at all, and then arrive at a notion of *well-supported proof* that allows the derivation of negative literals whose denials are manifestly underivable by irredundant proofs. In [15] it was shown that the notion of well-supported provability is consistent.

Definition 10 [5] Let $P = (\Sigma, Act, R)$ be a TSS. An *irredundant proof* from P of a rule $\frac{H}{\lambda}$ is a well-founded tree with the nodes labelled by literals and some of the leaves marked “hypothesis”, such that the root has label λ , H is the set of labels of the hypotheses, and if μ is the label of a node that is not a hypothesis and K is the set of labels of the children of this node then $\frac{K}{\mu}$ is a substitution instance of a rule in R . The rule $\frac{H}{\lambda}$ is *irredundantly provable* from P , notation $P \vdash_{irr} \frac{H}{\lambda}$, if such a proof exists.

We note that if a leaf in a proof from P is not marked as hypothesis, then it is a substitution instance of a rule without premises in R .

Definition 11 [15] Let $P = (\Sigma, Act, R)$ be a standard TSS. A *well-supported proof* from P of a closed literal λ is a well-founded tree with the nodes labelled by closed literals, such that the root is labelled by λ , and if μ is the label of a node and K is the set of labels of the children of this node, then:

1. either μ is positive and $\frac{K}{\mu}$ is a closed substitution instance of a rule in R ;
2. or μ is negative and for each set N of closed negative literals with $\frac{N}{v}$ irredundantly provable from P and v a closed positive literal denying μ , a literal in K denies one in N .

$P \vdash_{ws} \lambda$ denotes that a well-supported proof from P of λ exists. A standard TSS P is *complete* if for each p and α , either $P \vdash_{ws} p \not\xrightarrow{\alpha}$ or there exists a closed term q such that $P \vdash_{ws} p \xrightarrow{\alpha} q$.

If $P = (\Sigma, Act, R)$ is a complete TSS, then the LTS *associated with* P is the LTS $(T(\Sigma), Act, \rightarrow)$ with $\rightarrow = \{(p, \alpha, q) \mid P \vdash_{ws} p \xrightarrow{\alpha} q\}$. We do not associate an LTS with an incomplete TSS.

2.4 Congruence formats

Let $P = (\Sigma, Act, R)$ be a transition system specification, and let \sim_P be an equivalence relation defined on the set of closed terms $T(\Sigma)$. Then \sim_P is a *congruence* for P if, for each $f \in \Sigma$, we have that $p_i \sim_P q_i$ implies $f(p_1, \dots, p_{ar(f)}) \sim_P f(q_1, \dots, q_{ar(f)})$. Note that this is the case if for each open term $t \in T(\Sigma)$ and each pair of closed substitutions $\rho, \rho' : V \rightarrow T(\Sigma)$ we have $(\forall x \in var(t). \rho(x) \sim_P \rho'(x)) \Rightarrow \rho(t) \sim_P \rho'(t)$.

Recall that we have associated with every complete TSS an LTS of which the states are the closed terms of the TSS, and that a behavioural equivalence \sim associates with every such transition system an

equivalence on its set of states. Thus, \sim associates with every TSS P an equivalence \sim_P on its set of closed terms. By a *congruence format* for a behavioural equivalence \sim we mean a class of TSSs such that for every TSS P in the class the equivalence \sim_P is a congruence. Usually, a congruence format is defined by means of a list of syntactic restrictions on the rules of TSSs. We proceed to recap some terminology for syntactic restrictions on rules [5, 19, 20].

Definition 12 An *ntytt rule* is a rule in which the right-hand sides of positive premises are variables that are all distinct, and that do not occur in the source. An *ntytt rule* is an *ntyxt rule* if its source is a variable, an *ntyft rule* if its source contains exactly one function symbol and no multiple occurrences of variables, and an *nxytt rule* if the left-hand sides of its premises are variables.

The congruence format for strong bisimulation semantics consists of the ntyft and ntyxt rules [19, 20]. Congruence formats for other semantics are generally obtained by imposing additional restrictions on this ntyft/ntyxt format.

Definition 13 A variable in a rule is *free* if it occurs neither in the source nor in right-hand sides of premises. A rule has *lookahead* if some variable occurs in the right-hand side of a premise and in the left-hand side of a premise. A rule is *decent* if it has no lookahead and does not contain free variables.

Each combination of syntactic restrictions on rules induces a corresponding syntactic format for TSSs of the same name. For instance, a TSS is in decent ntyft format if it contains decent ntyft rules only.

Definition 14 A TSS is in *ready simulation format* if it consists of ntyft and ntyxt rules that have no lookahead.

In congruence formats for weak semantics, lookahead of two consecutive actions from A must be forbidden. To see this, consider the extension of CCS [23] with a unary operator f defined by the rule

$$\frac{x \xrightarrow{a} y \quad y \xrightarrow{b} z}{f(x) \xrightarrow{c} z}.$$

Then $ab\mathbf{0} \xleftrightarrow{rb} a\tau b\mathbf{0}$, whereas $f(ab\mathbf{0}) \not\xleftrightarrow{rb} f(a\tau b\mathbf{0})$. Therefore congruence formats for weak semantics are generally obtained by imposing additional restrictions on the ready simulation format.

2.5 Ruloids

To decompose modal formulas, we use a result from [5], where for each standard TSS P in ready simulation format a collection of decent nxytt rules, called *P-ruloids*, is constructed. The precise transformation, which is rather lengthy, can be found in [5]. Here the construction is explained at a somewhat superficial level, which is sufficiently detailed to argue later on that our congruence formats are preserved under this transformation.

First P is converted to a standard TSS in decent ntyft format. In this conversion from [20], free variables in a rule are replaced by arbitrary closed terms, and if the source is of the form x , then this variable is replaced by a term $f(x_1, \dots, x_{ar(f)})$ for each n -ary function symbol f in the signature of P , where the variables $x_1, \dots, x_{ar(f)}$ are fresh. Next, using a construction from [9], left-hand sides of positive premises in rules of P are reduced to variables. In the final transformation step, non-standard rules with a negative conclusion $t \xrightarrow{\alpha}$ are introduced. The motivation is that instead of the notion of well-founded provability of Def. 11, we want a more constructive notion like Def. 10, by making it possible that a negative premise is matched with a negative conclusion. A non-standard rule $\frac{H}{f(x_1, \dots, x_{ar(f)}) \xrightarrow{\alpha}}$ is obtained

by picking one premise from each standard rule with a conclusion of the form $f(x_1, \dots, x_{ar(f)}) \xrightarrow{\alpha} t$, and including the denial of each of the selected premises as a premise in H .

The resulting TSS, which is in decent ntyft format, is denoted by P^+ . In [5] it was established, for all closed literals μ , that $P \vdash_{ws} \mu$ if and only if μ is irredundantly provable from P^+ . The P -ruloids are those decent nxytt rules that are irredundantly provable from P^+ . In [5] it was proved that there is a well-supported proof from P of a transition $\rho(t) \xrightarrow{a} q$, with ρ a closed substitution, if and only if there is a proof of this transition that uses at the root a P -ruloid with source t . This result underlies the decomposition method presented in the next section.

2.6 Decomposition of modal formulas

The decomposition method proposed in [13] gives a special treatment to arguments of function symbols that are deemed *patient*; a predicate marks the arguments that get this special treatment.

Definition 15 [4, 8] Let Γ be a predicate on arguments of function symbols. A standard ntyft rule is a Γ -*patience rule* if it is of the form

$$\frac{x_i \xrightarrow{\tau} y}{f(x_1, \dots, x_{ar(f)}) \xrightarrow{\tau} f(x_1, \dots, x_{i-1}, y, x_{i+1}, \dots, x_{ar(f)})}$$

with $\Gamma(f, i)$. A TSS is Γ -*patient* if it contains all Γ -patience rules. A standard ntytt rule is Γ -*patient* if it is irredundantly provable from the Γ -patience rules; else it is called Γ -*impatient*.

A patience rule for an argument i of a function symbol f expresses that terms $f(p_1, \dots, p_{ar(f)})$ can mimic the τ -transitions of argument p_i (cf. [4, 8]). Typically, in process algebra, there are patience rules for the arguments of the merge and for the first argument of sequential composition, but not for the arguments of the alternative composition $+$ or for the second argument of sequential composition.

Definition 16 [5] Let Γ be a predicate on $\{(f, i) \mid 1 \leq i \leq ar(f), f \in \Sigma\}$. If $\Gamma(f, i)$, then argument i of f is Γ -*liquid*; otherwise it is Γ -*frozen*. An occurrence of x in t is Γ -*liquid* if either $t = x$, or $t = f(t_1, \dots, t_{ar(f)})$ and the occurrence is Γ -liquid in t_i for a liquid argument i of f ; otherwise the occurrence is Γ -*frozen*.

We now show how to decompose formulas from \mathbb{O} . To each term t and formula φ we assign a set $t^{-1}(\varphi)$ of decomposition mappings $\psi : V \rightarrow \mathbb{O}$. Each of these mappings $\psi \in t^{-1}(\varphi)$ has the property that, for all closed substitutions ρ , $\rho(t) \models \varphi$ if $\rho(x) \models \psi(x)$ for all $x \in var(t)$. Vice versa, whenever $\rho(t) \models \varphi$, there is a decomposition mapping $\psi \in t^{-1}(\varphi)$ with $\rho(x) \models \psi(x)$ for all $x \in var(t)$.

Definition 17 [13] Let $P = (\Sigma, A_\tau, R)$ be a Γ -patient standard TSS in ready simulation format. We define $\cdot^{-1} : \mathbb{T}(\Sigma) \times \mathbb{O} \rightarrow \mathcal{P}(V \rightarrow \mathbb{O})$ as the function that for each $t \in \mathbb{T}(\Sigma)$ and $\varphi \in \mathbb{O}$ returns the smallest set $t^{-1}(\varphi) \in \mathcal{P}(V \rightarrow \mathbb{O})$ of decomposition mappings $\psi : V \rightarrow \mathbb{O}$ satisfying the following six conditions. Let t denote a univariate term, i.e. without multiple occurrences of the same variable. (Cases 1–5 associate with every univariate term t a set $t^{-1}(\varphi)$. Then, in Case 6, the definition is generalised to terms that are not univariate, using that every term can be obtained by applying a non-injective substitution to a univariate term.)

1. $\psi \in t^{-1}(\bigwedge_{i \in I} \varphi_i)$ iff there are $\psi_i \in t^{-1}(\varphi_i)$ for each $i \in I$ such that

$$\psi(x) = \bigwedge_{i \in I} \psi_i(x) \quad \text{for all } x \in V$$

2. $\psi \in t^{-1}(\neg\varphi)$ iff there is a function $h : t^{-1}(\varphi) \rightarrow \text{var}(t)$ such that

$$\psi(x) = \begin{cases} \bigwedge_{\chi \in h^{-1}(x)} \neg\chi(x) & \text{if } x \in \text{var}(t) \\ \top & \text{if } x \notin \text{var}(t) \end{cases}$$

3. $\psi \in t^{-1}(\langle\alpha\rangle\varphi)$ iff there is a P -ruloid $\frac{H}{t \xrightarrow{\alpha} u}$ and a $\chi \in u^{-1}(\varphi)$ such that

$$\psi(x) = \begin{cases} \chi(x) \wedge \bigwedge_{x \xrightarrow{\beta} y \in H} \langle\beta\rangle\chi(y) \wedge \bigwedge_{x \not\xrightarrow{\gamma} \in H} \neg\langle\gamma\rangle\top & \text{if } x \in \text{var}(t) \\ \top & \text{if } x \notin \text{var}(t) \end{cases}$$

4. $\psi \in t^{-1}(\langle\varepsilon\rangle\varphi)$ iff one of the following holds:

- (a) either there is a $\chi \in t^{-1}(\varphi)$ such that

$$\psi(x) = \begin{cases} \langle\varepsilon\rangle\chi(x) & \text{if } x \text{ occurs } \Gamma\text{-liquid in } t \\ \chi(x) & \text{otherwise} \end{cases}$$

- (b) or there is a Γ -impatient P -ruloid $\frac{H}{t \xrightarrow{\varepsilon} u}$ and a $\chi \in u^{-1}(\langle\varepsilon\rangle\varphi)$ such that

$$\psi(x) = \begin{cases} \top & \text{if } x \notin \text{var}(t) \\ \langle\varepsilon\rangle\left(\chi(x) \wedge \bigwedge_{x \xrightarrow{\beta} y \in H} \langle\beta\rangle\chi(y) \wedge \bigwedge_{x \not\xrightarrow{\gamma} \in H} \neg\langle\gamma\rangle\top\right) & \text{if } x \text{ occurs } \Gamma\text{-liquid in } t \\ \chi(x) \wedge \bigwedge_{x \xrightarrow{\beta} y \in H} \langle\beta\rangle\chi(y) \wedge \bigwedge_{x \not\xrightarrow{\gamma} \in H} \neg\langle\gamma\rangle\top & \text{otherwise} \end{cases}$$

5. $\psi \in t^{-1}(\langle\hat{\tau}\rangle\varphi)$ iff one of the following holds:

- (a) either $\psi \in t^{-1}(\varphi)$;

- (b) or there is an x_0 that occurs Γ -liquid in t , and a $\chi \in t^{-1}(\varphi)$ such that

$$\psi(x) = \begin{cases} \langle\hat{\tau}\rangle\chi(x) & \text{if } x = x_0 \\ \chi(x) & \text{otherwise} \end{cases}$$

- (c) or there is a Γ -impatient P -ruloid $\frac{H}{t \xrightarrow{\hat{\tau}} u}$ and a $\chi \in u^{-1}(\varphi)$ such that

$$\psi(x) = \begin{cases} \chi(x) \wedge \bigwedge_{x \xrightarrow{\beta} y \in H} \langle\beta\rangle\chi(y) \wedge \bigwedge_{x \not\xrightarrow{\gamma} \in H} \neg\langle\gamma\rangle\top & \text{if } x \in \text{var}(t) \\ \top & \text{otherwise} \end{cases}$$

6. $\psi \in \sigma(t)^{-1}(\varphi)$ for a non-injective substitution $\sigma : \text{var}(t) \rightarrow V$ iff there is a $\chi \in t^{-1}(\varphi)$ such that

$$\psi(x) = \bigwedge_{z \in \sigma^{-1}(x)} \chi(z) \quad \text{for all } x \in V$$

The following theorem will be the key to the forthcoming congruence results.

Theorem 18 [13] Let $P = (\Sigma, A_\tau, R)$ be a Γ -patient complete standard TSS in ready simulation format. For each term $t \in \mathbb{T}(\Sigma)$, closed substitution ρ , and $\varphi \in \mathbb{O}$:

$$\rho(t) \models \varphi \Leftrightarrow \exists \psi \in t^{-1}(\varphi) \forall x \in \text{var}(t) : \rho(x) \models \psi(x)$$

3 Stability-respecting branching bisimilarity as a congruence

We proceed to apply the decomposition method from the previous section to derive congruence formats for stability-respecting branching bisimilarity and rooted stability-respecting branching bisimilarity. The idea behind the construction of these congruence formats is that the format must guarantee that a formula from the characterising logic of the equivalence under consideration is always decomposed into formulas from this same logic (see Prop. 25). This implies the desired congruence results (see Thm. 26 and Thm. 27).

3.1 Congruence format for stability-respecting branching bisimilarity

The definitions of the congruence formats for (rooted) stability-respecting branching bisimilarity, below, presuppose two predicates Λ and \mathfrak{X} on the arguments of the function symbols of a TSS: Λ marks arguments that contain processes that, intuitively, have started executing (but may currently be unable to execute), while \mathfrak{X} marks arguments that contain processes that can execute immediately. For example, in process algebra, Λ and \mathfrak{X} hold for the arguments of the merge $t_1 \parallel t_2$, and for the first argument of sequential composition $t_1 \cdot t_2$; they can contain processes that started to execute in the past, and these processes can continue their execution immediately. Λ and \mathfrak{X} typically do not hold for the second argument of sequential composition; it contains a process that did not yet start to execute, and cannot execute immediately (in absence of the empty process). Λ does not hold and \mathfrak{X} holds for the arguments of alternative composition $t_1 + t_2$; they contain processes that did not yet start to execute, but that can start executing immediately. We will instantiate Γ (from Sect. 2.6) with $\mathfrak{X} \cap \Lambda$.

We proceed to define when a standard ntytt rule is rooted stability-respecting branching bisimulation safe, and base the rooted stability-respecting branching bisimulation format on that notion. The stability-respecting branching bisimulation format will be defined by adding one additional restriction to its rooted counterpart: Λ is universal, i.e., $\Lambda(f, i)$ for all $f \in \Sigma$ and $i = 1, \dots, \text{ar}(f)$. Our aim for the rest of this section will be to prove that the (rooted) stability-respecting branching bisimulation format guarantees that (rooted) stability-respecting branching bisimilarity is a congruence.

Definition 19 Let \mathfrak{X} and Λ be predicates on $\{(f, i) \mid 1 \leq i \leq \text{ar}(f), f \in \Sigma\}$. A standard ntytt rule $r = \frac{H}{t \xrightarrow{\alpha} u}$ is *rooted stability-respecting branching bisimulation safe* w.r.t. \mathfrak{X} and Λ if it satisfies the following conditions.

1. Right-hand sides of positive premises occur only Λ -liquid in u .
2. If $x \in \text{var}(t)$ occurs only Λ -liquid in t , then x occurs only Λ -liquid in r .
3. If $x \in \text{var}(t)$ occurs only \mathfrak{X} -frozen in t , then x occurs only \mathfrak{X} -frozen in H .
4. Suppose that x has exactly one \mathfrak{X} -liquid occurrence in t , and that this occurrence is also Λ -liquid.
 - (a) If x has an \mathfrak{X} -liquid occurrence in a negative premise in H or more than one \mathfrak{X} -liquid occurrence in the positive premises in H , then there is a premise $v \xrightarrow{\tau} \cdot$ in H such that x occurs \mathfrak{X} -liquid in v .

- (b) If there is a premise $w \xrightarrow{\tau} y$ in H and x occurs \mathfrak{K} -liquid in w , then r is $\mathfrak{K} \cap \Lambda$ -patient.

Conditions 1–3 have been copied from the definition of rooted branching bisimulation safeness from [13], and condition 4b is part of condition 4 in that definition. Condition 4a, however, establishes a relaxation of condition 4 in the definition of rooted branching bisimulation safeness, where it was required that x has at most one \mathfrak{K} -liquid occurrence in H , which must be in a positive premise. Here, owing to stability, we can be more tolerant, as long as $x \xrightarrow{\tau}$ can be derived. As a consequence of this relaxation we will see that the rule for the priority operator is rooted stability-respecting branching bisimulation safe, while it is not rooted branching bisimulation safe (see Section 3.5).

Definition 20 A standard TSS is in *rooted stability-respecting branching bisimulation format* if it is in ready simulation format and, for some \mathfrak{K} and Λ , it is $\mathfrak{K} \cap \Lambda$ -patient and only contains rules that are rooted stability-respecting branching bisimulation safe w.r.t. \mathfrak{K} and Λ .

This TSS is in *stability-respecting branching bisimulation format* if moreover Λ is universal.

3.2 Preservation of syntactic restrictions

The definition of modal decomposition is based on the P -ruloids. Therefore we must verify that if P is in rooted stability-respecting branching bisimulation format, then the P -ruloids are rooted stability-respecting branching bisimulation safe (Prop. 24). The key part of the proof is to show that the syntactic restriction of decent rooted stability-respecting branching bisimulation safety is preserved under irredundant provability (Lem. 23).

In the proof of the preservation lemma below, rules with a negative conclusion will play an important role. Therefore the notion of rooted stability-respecting branching bisimulation safety needs to be extended to non-standard rules. The following definition coincides with the definition of rooted branching bisimulation safe for non-standard rules in [13].

Definition 21 An ntytt rule $r = \frac{H}{t \xrightarrow{\tau} \cdot}$ is *rooted stability-respecting branching bisimulation safe* w.r.t. \mathfrak{K} and Λ if it satisfies conditions 2 and 3 of Def. 19.

Lemma 22 Let Q be an $\mathfrak{K} \cap \Lambda$ -patient TSS in decent ntyft format. If an ntytt rule $\frac{H}{t \xrightarrow{\tau} \cdot}$ is provable from Q^+ and x occurs $\mathfrak{K} \cap \Lambda$ -liquid in t , then H contains a premise $v \xrightarrow{\tau} \cdot$ where x occurs $\mathfrak{K} \cap \Lambda$ -liquid in v .

Proof: Recall from Sect. 2.5 that also the TSS Q^+ is in decent ntyft format. Let $\frac{H}{t \xrightarrow{\tau} \cdot}$ be provable from Q^+ , by means of a proof π . We apply structural induction with respect to π .

Induction basis: The case where π has only one node, marked “hypothesis”, is trivial, as then H contains $t \xrightarrow{\tau} \cdot$.

Induction step: Let r be the decent ntyft rule and σ the substitution used at the bottom of π . Let $f(x_1, \dots, x_{ar(f)}) \xrightarrow{\tau} \cdot$ be the conclusion of r . Then $\sigma(f(x_1, \dots, x_{ar(f)})) = t$. Since x occurs $\mathfrak{K} \cap \Lambda$ -liquid in t , it occurs $\mathfrak{K} \cap \Lambda$ -liquid in $\sigma(x_{i_0})$ where i_0 is an $\mathfrak{K} \cap \Lambda$ -liquid argument of f . In view of the patience rule for this argument of f and the construction of rules with a negative premise it follows that r has a premise $x_{i_0} \xrightarrow{\tau} \cdot$. So a rule $\frac{H'}{\sigma(x_{i_0}) \xrightarrow{\tau} \cdot}$ is provable from Q^+ by means of a strict subproof of π , where $H' \subseteq H$. By induction H' contains a premise $v \xrightarrow{\tau} \cdot$ where x occurs $\mathfrak{K} \cap \Lambda$ -liquid in v . \square

Lemma 23 Let P be an $\mathfrak{K} \cap \Lambda$ -patient TSS in decent ntyft format, in which each rule is rooted stability-respecting branching bisimulation safe w.r.t. \mathfrak{K} and Λ . Moreover, P is either a standard TSS or a TSS of the form Q^+ with Q an $\mathfrak{K} \cap \Lambda$ -patient standard TSS in decent ntyft format. Then each ntytt rule irredundantly provable from P is rooted stability-respecting branching bisimulation safe w.r.t. \mathfrak{K} and Λ .

Proof: For all conditions of Def. 19 and Def. 21 except 4a, the preservation proof coincides with the corresponding proof for rooted branching bisimulation safeness in [13, Lem. 5—aliased 4.5]. Therefore we here focus only on condition 4a. Let an ntytt rule $\frac{H}{t \xrightarrow{\alpha} u}$ be irredundantly provable from P , by means of a proof π . We prove, using structural induction with respect to π , that this rule satisfies condition 4a of Def. 19.

Induction basis: Suppose π has only one node, marked “hypothesis”. Then $\frac{H}{t \xrightarrow{\alpha} u}$ equals $\frac{t \xrightarrow{\alpha} u}{t \xrightarrow{\alpha} u}$ (so u is a variable). This rule trivially satisfies condition 4a of Def. 19.

Induction step: Let r be the decent ntyft rule and σ the substitution used at the bottom of π . By assumption, r is decent, ntyft, and rooted stability-respecting branching bisimulation safe w.r.t. \mathfrak{N} and Λ . Let r be of the form

$$\frac{\{v_k \xrightarrow{\beta_k} y_k \mid k \in K\} \cup \{w_\ell \xrightarrow{\gamma_\ell} \mid \ell \in L\}}{f(x_1, \dots, x_{ar(f)}) \xrightarrow{\alpha} v}$$

Then $\sigma(f(x_1, \dots, x_{ar(f)})) = t$ and $\sigma(v) = u$. Moreover, decent ntytt rules $r_k = \frac{H_k}{\sigma(v_k) \xrightarrow{\beta_k} \sigma(y_k)}$ for each $k \in K$ and $r_\ell = \frac{H_\ell}{\sigma(w_\ell) \xrightarrow{\gamma_\ell}}$ for each $\ell \in L$ are irredundantly provable from P by means of strict subproofs of π , where $H = \bigcup_{k \in K} H_k \cup \bigcup_{\ell \in L} H_\ell$. By induction, they are rooted stability-respecting branching bisimulation safe w.r.t. \mathfrak{N} and Λ .

Suppose that x has exactly one \mathfrak{N} -liquid occurrence in t , which is also Λ -liquid. Then there is an $i_0 \in \{1, \dots, ar(f)\}$ with $\mathfrak{N}(f, i_0)$ and $\Lambda(f, i_0)$ such that x has exactly one \mathfrak{N} -liquid occurrence in $\sigma(x_{i_0})$, which is also Λ -liquid. Furthermore, for each $i \in \{1, \dots, ar(f)\} \setminus \{i_0\}$, $\neg \mathfrak{N}(f, i)$ or x occurs only \mathfrak{N} -frozen in $\sigma(x_i)$. Since the ntyft rule r is rooted stability-respecting branching bisimulation safe w.r.t. \mathfrak{N} and Λ , by condition 3 of Def. 19, if $\neg \mathfrak{N}(f, i)$, then x_i occurs only \mathfrak{N} -frozen in v_k for all $k \in K$, as well as in w_ℓ for all $\ell \in L$. We distinguish three possible cases, and argue each time that condition 4a of Def. 19 is satisfied.

CASE 1: x_{i_0} has no \mathfrak{N} -liquid occurrences in the premises of r . Then x has no \mathfrak{N} -liquid occurrences in $\sigma(v_k)$ for $k \in K$ and $\sigma(w_\ell)$ for $\ell \in L$. So by condition 3 of Def. 19 and decency of the r_k and r_ℓ , x has no \mathfrak{N} -liquid occurrences in H .

CASE 2: x_{i_0} has exactly one \mathfrak{N} -liquid occurrence in the premises of r , in v_{k_0} for some $k_0 \in K$. By condition 2 of Def. 19 this occurrence is also Λ -liquid. Then x has exactly one \mathfrak{N} -liquid occurrence in $\sigma(v_{k_0})$, and this occurrence is also Λ -liquid. So by condition 4a of Def. 19, if x has an \mathfrak{N} -liquid occurrence in a negative premise in H_{k_0} or more than one in the positive premises in H_{k_0} , then there is a premise $w \xrightarrow{\tau} \cdot$ in $H_{k_0} \subseteq H$ where x occurs \mathfrak{N} -liquid in w . Furthermore, x has no \mathfrak{N} -liquid occurrences in $\sigma(v_k)$ for $k \in K \setminus \{k_0\}$ and $\sigma(w_\ell)$ for $\ell \in L$. So by condition 3 of Def. 19, x has no \mathfrak{N} -liquid occurrences in H_k for $k \in K \setminus \{k_0\}$ and H_ℓ for $\ell \in L$.

CASE 3: x_{i_0} occurs \mathfrak{N} -liquid in w_{ℓ_0} for some $\ell_0 \in L$ or has more than one \mathfrak{N} -liquid occurrence in the v_k for $k \in K$. Then, by condition 4a of Def. 19, x_{i_0} occurs \mathfrak{N} -liquid in w_{ℓ_1} for some $\ell_1 \in L$ with $\gamma_{\ell_1} = \tau$. By condition 2 of Def. 19 this occurrence is also Λ -liquid. It follows that x occurs $\mathfrak{N} \cap \Lambda$ -liquid in $\sigma(w_{\ell_1})$. In case P is a standard TSS the premise $w_{\ell_1} \xrightarrow{\tau}$ occurs in H , and otherwise, by Lem. 22, $H_{\ell_1} \subseteq H$ contains a premise $w \xrightarrow{\tau}$ where x occurs $\mathfrak{N} \cap \Lambda$ -liquid in w . \square

The following proposition can now be proved in the same way as the corresponding Prop. 2—aliased 4.6—for rooted branching bisimulation safeness in [13].

Proposition 24 Let P be an $\aleph \cap \Lambda$ -patient TSS in ready simulation format, in which each rule is rooted stability-respecting branching bisimulation safe w.r.t. \aleph and Λ . Then each P -ruloid is rooted stability-respecting branching bisimulation safe w.r.t. \aleph and Λ .

3.3 Preservation of modal characterisations

Consider a standard TSS in rooted stability-respecting branching bisimulation format, w.r.t. some \aleph and Λ . Def. 17 yields decomposition mappings $\psi \in t^{-1}(\varphi)$, with $\Gamma := \aleph \cap \Lambda$. In this section we prove that if $\varphi \in \mathbb{O}_b^s$, then $\psi(x) \in \mathbb{O}_b^{s\equiv}$ if x occurs only Λ -liquid in t . (That is why in the stability-respecting branching bisimulation format, Λ must be universal.) Furthermore, we prove that if $\varphi \in \mathbb{O}_{rb}^s$, then $\psi(x) \in \mathbb{O}_{rb}^{s\equiv}$ for all variables x . From these preservation results we will, in Sect. 3.4, deduce the promised congruence results for unrooted and rooted stability-respecting branching bisimilarity, respectively.

In [13] the following proposition was proved in two separate steps: Prop. 3 and Prop. 4—aliased 4.7 and 4.8—in that paper. This was possible since \mathbb{O}_{rb} incorporates \mathbb{O}_b but not vice versa. However, since there is a circular dependency between the definitions of \mathbb{O}_b^s and \mathbb{O}_{rb}^s , here we need to prove the corresponding results for these modal characterisations simultaneously. The third part of the proposition below is merely a tool in proving the other parts.

Proposition 25 Let P be an $\aleph \cap \Lambda$ -patient standard TSS in ready simulation format, in which each rule is rooted stability-respecting branching bisimulation safe w.r.t. \aleph and Λ .

1. For each term t and variable x that occurs only Λ -liquid in t :

$$\varphi \in \mathbb{O}_b^s \Rightarrow \forall \psi \in t^{-1}(\varphi) : \psi(x) \in \mathbb{O}_b^{s\equiv}$$

2. For each term t and variable x :

$$\overline{\varphi} \in \mathbb{O}_{rb}^s \Rightarrow \forall \psi \in t^{-1}(\overline{\varphi}) : \psi(x) \in \mathbb{O}_{rb}^{s\equiv}$$

3. For each term t and variable x that occurs only Λ -liquid and \aleph -frozen in t :

$$\overline{\varphi} \in \mathbb{O}_{rb}^s \Rightarrow \forall \psi \in t^{-1}(\overline{\varphi}) : \psi(x) \in \mathbb{O}_b^{s\equiv}$$

Proof: We apply simultaneous induction on the structure of φ , resp. $\overline{\varphi}$, and the construction of ψ . We only treat the case where t is univariate. The case where t is not univariate can be dealt with in the same way as in the proofs of the corresponding Prop. 3 and Prop. 4—aliased 4.7 and 4.8—in [13]. Let $\psi \in t^{-1}(\varphi)$. If $x \notin \text{var}(t)$ then $\psi(x) \equiv \top \in \mathbb{O}_b^{s\equiv}$. So suppose x occurs exactly once in t .

We start with the first claim of the proposition. The cases where φ is of the form $\bigwedge_{i \in I} \varphi_i$ or $\neg \varphi'$ can be dealt with as in the proof of Prop. 3—aliased 4.7—in [13]. We therefore focus on the other cases.

- $\varphi = \langle \varepsilon \rangle (\varphi_1 \langle \hat{\tau} \rangle \varphi_2)$ with $\varphi_1, \varphi_2 \in \mathbb{O}_b^s$. Let the occurrence of x in t be Λ -liquid. According to Def. 17.4 we can distinguish two cases.

CASE 1: $\psi(x)$ is defined based on Def. 17.4a. Then $\psi(x) = \langle \varepsilon \rangle \chi(x)$ if x occurs \aleph -liquid in t , or $\psi(x) = \chi(x)$ if x occurs \aleph -frozen in t , for some $\chi \in t^{-1}(\varphi_1 \langle \hat{\tau} \rangle \varphi_2)$. By Def. 17.1, $\chi(x) = \chi_1(x) \wedge \chi_2(x)$ with $\chi_1 \in t^{-1}(\varphi_1)$ and $\chi_2 \in t^{-1}(\langle \hat{\tau} \rangle \varphi_2)$. By induction on formula size, $\chi_1(x) \in \mathbb{O}_b^{s\equiv}$. For $\chi_2(x)$, according to Def. 17.5, we can distinguish three cases. Cases 1.1 and 1.2 where $\chi_2(x)$ is defined based on Def. 17.5a and Def. 17.5b, respectively, proceed in the same way as in the proof of [13, Prop. 3]. We focus on the third case.

CASE 1.3: $\chi_2(x)$ is defined based on Def. 17.5c, employing an $\aleph \cap \Lambda$ -impatient P -ruloid $\frac{H}{t \xrightarrow{\tau} u}$ and a $\xi \in u^{-1}(\varphi_2)$. So $\chi_2(x) = \xi(x) \wedge \bigwedge_{x \xrightarrow{\beta} y \in H} \langle \beta \rangle \xi(y) \wedge \bigwedge_{x \xrightarrow{\gamma} \top \in H} \neg \langle \gamma \rangle \top$. By Prop. 24, $\frac{H}{t \xrightarrow{\tau} u}$ is rooted stability-respecting branching bisimulation safe. Since the occurrence of x in t is Λ -liquid, by condition 2 of Def. 19, x occurs only Λ -liquid in u . Therefore, by induction on formula size, $\xi(x) \in \mathbb{O}_b^{s\equiv}$. Case 1.3.1 where the occurrence of x in t is \aleph -frozen still proceeds in the same way as in the proof of [13, Prop. 3]. We focus on the other case.

CASE 1.3.2: The occurrence of x in t is \aleph -liquid. If H has at most one premise of the form $x \xrightarrow{\beta} y$, for which $\beta \neq \tau$, and none of the form $x \xrightarrow{\gamma} \top$, then still the proof proceeds in the same way as in the proof of [13, Prop. 3]. However, the more relaxed condition 4a of Def. 19 allows H to have more than one premise of the form $x \xrightarrow{\beta} y$ with $\beta \neq \tau$, or premises $x \xrightarrow{\gamma} \top$. Only, then H must also contain the premise $x \xrightarrow{\tau} \top$. Thus $\psi(x) \equiv \langle \varepsilon \rangle (\neg \langle \tau \rangle \top \wedge \chi_1(x) \wedge \xi(x) \wedge \bigwedge_{x \xrightarrow{b} y \in H} \langle b \rangle \xi(y) \wedge \bigwedge_{x \xrightarrow{c} \top \in H} \neg \langle c \rangle \top)$. By condition 1 of Def. 19 the right-hand sides y of positive premises in H occur only Λ -liquid in u , so by induction $\xi(y) \in \mathbb{O}_b^{s\equiv}$. Hence the conjuncts $\langle b \rangle \xi(y)$ are in $\mathbb{O}_{rb}^{s\equiv}$. It follows that $\psi(x) \in \mathbb{O}_b^{s\equiv}$.

CASE 2: $\psi(x)$ is defined based on Def. 17.4b, employing an $\aleph \cap \Lambda$ -impatient P -ruloid $\frac{H}{t \xrightarrow{\tau} u}$ and a $\chi \in u^{-1}(\langle \varepsilon \rangle (\varphi_1 \langle \hat{\tau} \rangle \varphi_2))$. By Prop. 24, $\frac{H}{t \xrightarrow{\tau} u}$ is rooted stability-respecting branching bisimulation safe. Since the occurrence of x in t is Λ -liquid, by condition 2 of Def. 19, x occurs only Λ -liquid in u . Therefore, by induction on the construction of ψ , $\chi(x) \in \mathbb{O}_b^{s\equiv}$. Case 2.1 where the occurrence of x in t is \aleph -frozen proceeds in the same way as in the proof of [13, Prop. 3]. We focus on the other case.

CASE 2.2: The occurrence of x in t is \aleph -liquid. Then $\psi(x) = \langle \varepsilon \rangle (\chi(x) \wedge \bigwedge_{x \xrightarrow{\beta} y \in H} \langle \beta \rangle \chi(y) \wedge \bigwedge_{x \xrightarrow{\gamma} \top \in H} \neg \langle \gamma \rangle \top)$. If H has at most one premise of the form $x \xrightarrow{\beta} y$, for which $\beta \neq \tau$, and none of the form $x \xrightarrow{\gamma} \top$, then still the proof proceeds in the same way as in the proof of [13, Prop. 3]. However, the more relaxed condition 4a of Def. 19 allows H to have more than one premise of the form $x \xrightarrow{\beta} y$ with $\beta \neq \tau$, or premises $x \xrightarrow{\gamma} \top$. Only, then H must also contain the premise $x \xrightarrow{\tau} \top$. Thus $\psi(x) \equiv \langle \varepsilon \rangle (\neg \langle \tau \rangle \top \wedge \chi(x) \wedge \bigwedge_{x \xrightarrow{b} y \in H} \langle b \rangle \chi(y) \wedge \bigwedge_{x \xrightarrow{c} \top \in H} \neg \langle c \rangle \top)$. By condition 1 of Def. 19 the right-hand sides y of positive premises in H occur only Λ -liquid in u , so by induction $\chi(y) \in \mathbb{O}_b^{s\equiv}$. Hence the conjuncts $\langle b \rangle \chi(y)$ are in $\mathbb{O}_{rb}^{s\equiv}$. It follows that $\psi(x) \in \mathbb{O}_b^{s\equiv}$.

The proof of the case $\varphi = \langle \varepsilon \rangle (\varphi_1 \langle a \rangle \varphi_2)$ in \mathbb{O}_b from [13, Prop. 3] needs to be adapted in a similar fashion as the case $\varphi = \langle \varepsilon \rangle (\varphi_1 \langle \hat{\tau} \rangle \varphi_2)$. We take the liberty to omit this adaptation here, and continue with the additional clause in the modal characterisation of \mathbb{O}_b^s , compared to \mathbb{O}_b .

- $\varphi = \langle \varepsilon \rangle (\neg \langle \tau \rangle \top \wedge \bar{\varphi})$ with $\bar{\varphi} \in \mathbb{O}_{rb}^s$. Let the occurrence of x in t be Λ -liquid. According to Def. 17.4 we can distinguish two cases.

CASE 1: $\psi(x)$ is defined based on Def. 17.4a. Then $\psi(x) = \langle \varepsilon \rangle \chi(x)$ if x occurs \aleph -liquid in t , or $\psi(x) = \chi(x)$ if x occurs \aleph -frozen in t , for some $\chi \in t^{-1}(\neg \langle \tau \rangle \top \wedge \bar{\varphi})$. By Def. 17.1, $\chi(x) = \chi_1(x) \wedge \chi_2(x)$ with $\chi_1 \in t^{-1}(\neg \langle \tau \rangle \top)$ and $\chi_2 \in t^{-1}(\bar{\varphi})$. By Def. 17.2 there is a function $h : t^{-1}(\langle \tau \rangle \top) \rightarrow \text{var}(t)$ such that $\chi_1(x) = \bigwedge_{\xi \in h^{-1}(x)} \neg \xi(x)$.

CASE 1.1: x occurs \aleph -liquid in t . By Def. 17.3, for each $\xi \in h^{-1}(x)$, $\xi(x)$ is of the form $\bigwedge_{x \xrightarrow{\beta} y \in H} \langle \beta \rangle \top \wedge \bigwedge_{x \xrightarrow{\gamma} \top \in H} \neg \langle \gamma \rangle \top$ for some P -ruloid $\frac{H}{t \xrightarrow{\tau} u}$. Note that such formulas are in \mathbb{O}_{rb}^s . Moreover, by induction on formula size, $\chi_2(x) \in \mathbb{O}_{rb}^{s\equiv}$. Since the occurrence of x in t is

$\aleph \cap \Lambda$ -liquid, there is an $\aleph \cap \Lambda$ -patient ruloid $\frac{x \xrightarrow{\tau} y}{t \xrightarrow{\tau} y'}$. This gives rise to a $\xi \in h^{-1}(x)$ such that $\xi(x) = \neg\langle\tau\rangle\top$. Concluding, $\psi(x) = \langle\varepsilon\rangle(\chi_1(x) \wedge \chi_2(x))$ is of the form $\langle\varepsilon\rangle(\neg\langle\tau\rangle\top \wedge \bar{\varphi}')$ for some $\bar{\varphi}' \in \mathbb{O}_{rb}^s$. So $\psi(x) \in \mathbb{O}_b^s$.

CASE 1.2: x occurs \aleph -frozen in t . Then by condition 3 of Def. 19, x does not occur in H . Since moreover $\omega(x) \equiv \top$ for each $\omega \in t^{-1}(\top)$, by Def. 17.3, $\xi(x) \equiv \top$ for each $\xi \in h^{-1}(x)$. So either $\chi_1(x) \equiv \neg\top$ if $h^{-1}(x)$ is non-empty or $\chi_1(x) \equiv \top$ if $h^{-1}(x)$ is empty. In the first case $\psi(x) \equiv \neg\top$, so then we are done. In the second case, by induction on formula size, using the third claim of this proposition, $\psi(x) \equiv \chi_2(x) \in \mathbb{O}_b^{s\equiv}$.

CASE 2: $\psi(x)$ is defined based on Def. 17.4b. This case proceeds in the same way as case 2 of $\varphi = \langle\varepsilon\rangle(\varphi_1\langle\hat{\tau}\rangle\varphi_2)$.

This completes the proof of the first claim of the proposition. We continue with the second claim of the proposition. The cases where $\bar{\varphi}$ is of the form $\bigwedge_{i \in I} \bar{\varphi}_i$ or $\neg\bar{\varphi}'$ or $\langle\alpha\rangle\varphi$ can be dealt with as in the proof of Prop. 4 in [13]. We focus on the only other case.

- $\bar{\varphi} \in \mathbb{O}_b^s$. The cases where $\bar{\varphi}$ is of the form $\bigwedge_{i \in I} \varphi_i$ or $\neg\varphi'$ or $\langle\varepsilon\rangle(\varphi_1\langle\hat{\tau}\rangle\varphi_2)$ or $\langle\varepsilon\rangle(\varphi_1\langle a \rangle\varphi_2)$ can be dealt with as in the proof of Prop. 4 in [13]. We focus on the only new case here.
- * $\bar{\varphi} = \langle\varepsilon\rangle(\neg\langle\tau\rangle\top \wedge \bar{\varphi}')$ with $\bar{\varphi}' \in \mathbb{O}_{rb}^s$. If the occurrence of x in t is Λ -liquid, then we already proved in the corresponding case for the first claim of the proposition that $\psi(x) \in \mathbb{O}_b^s \subset \mathbb{O}_{rb}^{s\equiv}$. So we can assume that this occurrence is Λ -frozen. According to Def. 17.4 we can distinguish two cases.

CASE 1: $\psi(x)$ is defined based on Def. 17.4a. Then, since x occurs Λ -frozen in t , $\psi(x) = \chi(x)$ for some $\chi \in t^{-1}(\neg\langle\tau\rangle\top \wedge \bar{\varphi}')$. By Def. 17.1, $\chi(x) = \chi_1(x) \wedge \chi_2(x)$ with $\chi_1 \in t^{-1}(\neg\langle\tau\rangle\top)$ and $\chi_2 \in t^{-1}(\bar{\varphi}')$. By induction on formula size, $\chi_1(x) \in \mathbb{O}_{rb}^{s\equiv}$ and $\chi_2(x) \in \mathbb{O}_{rb}^{s\equiv}$. Hence, $\psi(x) = \chi_1(x) \wedge \chi_2(x)$ is in $\mathbb{O}_{rb}^{s\equiv}$.

CASE 2: $\psi(x)$ is defined based on Def. 17.4b, using an $\aleph \cap \Lambda$ -impatient P -ruloid $\frac{H}{t \xrightarrow{\tau} u}$ and a $\chi \in u^{-1}(\langle\varepsilon\rangle(\neg\langle\tau\rangle\top \wedge \bar{\varphi}'))$. As the occurrence of x in t is Λ -frozen,
 $\psi(x) = \chi(x) \wedge \bigwedge_{x \xrightarrow{\beta} y \in H} \langle\beta\rangle\chi(y) \wedge \bigwedge_{x \xrightarrow{\gamma} \top \in H} \neg\langle\gamma\rangle\top$. By induction on the construction of ψ , $\chi(x) \in \mathbb{O}_{rb}^{s\equiv}$. Moreover, by condition 1 of Def. 19 the y occur only Λ -liquid in u , so we proved before that the $\chi(y)$ are in $\mathbb{O}_b^{s\equiv}$. Hence $\psi(x) \in \mathbb{O}_{rb}^{s\equiv}$.

This completes the proof of the second claim of the proposition. We finish with the last claim. The cases where $\bar{\varphi}$ is of the form $\bigwedge_{i \in I} \bar{\varphi}_i$ or $\neg\bar{\varphi}'$ can be dealt with as in the proof of Prop. 3 in [13], and the case $\bar{\varphi} = \varphi \in \mathbb{O}_b$ follows immediately from the first claim. We therefore focus on the remaining case: $\bar{\varphi} = \langle\alpha\rangle\varphi$ with $\varphi \in \mathbb{O}_b$.

$\psi(x)$ is defined based on Def. 17.3, for some P -ruloid $\frac{H}{t \xrightarrow{\alpha} u}$ and $\chi \in u^{-1}(\varphi)$. Since the occurrence of x in t is \aleph -frozen, by condition 3 of Def. 19, x does not occur in H . Hence, $\psi(x) = \chi(x)$. By Prop. 24, $\frac{H}{t \xrightarrow{\alpha} u}$ is rooted stability-respecting branching bisimulation safe. Since the occurrence of x in t is Λ -liquid, by condition 2 of Def. 19, x occurs only Λ -liquid in u . Therefore, by the first claim of this proposition, $\chi(x) \in \mathbb{O}_b^{s\equiv}$. \square

3.4 Congruence results

Now the promised congruence results for \xleftrightarrow{b}^s and \xleftrightarrow{rb}^s can be proved in the same way as their counterparts for \xleftrightarrow{b} and \xleftrightarrow{rb} in [13].

Theorem 26 Let P be a complete standard TSS in stability-respecting branching bisimulation format. Then \leftrightarrow_b^s is a congruence for P . \square

Theorem 27 Let P be a complete standard TSS in rooted stability-respecting branching bisimulation format. Then \leftrightarrow_{rb}^s is a congruence for P . \square

3.5 Application to the priority operator

The *priority* operator [1] is a unary function the definition of which is based on an ordering $<$ on atomic actions. The term $\Theta(p)$ executes the transitions of the term p , with the restriction that a transition $p \xrightarrow{a} q$ only gives rise to a transition $\Theta(p) \xrightarrow{a} \Theta(q)$ if there does not exist a transition $p \xrightarrow{b} q'$ with $b > a$. This intuition is captured by the rule for the priority operator below.

$$\frac{x \xrightarrow{a} y \quad x \not\xrightarrow{b} \text{ for all } b > a}{\Theta(x) \xrightarrow{a} \Theta(y)}$$

In view of the target $\Theta(y)$, by condition 1 of Def. 19, the argument of Θ must be chosen Λ -liquid. And in view of condition 3 of Def. 19, the argument of Θ must be chosen \aleph -liquid. Then the rule above is rooted stability-respecting branching bisimulation safe, if the following condition on the ordering on atomic actions is satisfied: if there is a b such that $b > a$, then $\tau > a$. Namely, this guarantees that condition 4a of Def. 19 is satisfied: if there is a negative premise $x \not\xrightarrow{b}$, then there is also a negative premise $x \not\xrightarrow{\tau}$. Moreover, since the argument of Θ is Λ -liquid, this operator is within the stability-respecting branching bisimulation format.

Corollary 28 \leftrightarrow_b^s and \leftrightarrow_{rb}^s are congruences for the priority operator.

In general, the priority operator Θ does not preserve [rooted] branching bisimilarity (cf. [30, pp. 130–132]), as shown by the following example.

Example 29 Consider the following two LTSs:



Clearly $p \leftrightarrow_b q$. Note that on the other hand $p \not\leftrightarrow_b^s q$, because q is stable while p cannot perform a sequence of τ -transitions to a stable state.

Suppose that $a < b$. Let us try to extend the ordering $<$ such that $\Theta(p) \leftrightarrow_b \Theta(q)$. Since $\Theta(p)$ cannot execute the trace aa , we must declare $a < \tau$, to block this trace in $\Theta(q)$. But then $\Theta(p)$ can only execute an infinite τ -sequence while $\Theta(q)$ can execute a . Hence $\Theta(p) \not\leftrightarrow_b \Theta(q)$ for every ordering $<$ (with $a < b$).

Moreover, if p_0 and q_0 are processes with as only transitions $p_0 \xrightarrow{\tau} p$ and $q_0 \xrightarrow{\tau} q$, then $p_0 \leftrightarrow_{rb} q_0$, but $\Theta(p_0) \not\leftrightarrow_{rb} \Theta(q_0)$ for every ordering $<$ (with $a < b$).

So inevitably, as observed in [13], the rule for the priority operator is not in the rooted branching bisimulation format. Namely, the $\aleph \cap \Lambda$ -liquid argument x in the source occurs \aleph -liquid in the negative premises, which violates the more restrictive condition 4 of the rooted branching bisimulation format.

3.6 Application to sequencing

The binary *sequencing operator* $;$ is a variant of sequential composition that does not rely on a notion of successful termination. Intuitively, the process $p;q$ behaves as its left-hand side argument until that can no longer do any transitions; then it proceeds with its right-hand side argument. The rules below, which appeared already in [3], formalise this behaviour:

$$\frac{x \xrightarrow{\alpha} x'}{x;q \xrightarrow{\alpha} x';y} \quad \frac{x \not\xrightarrow{\alpha} \text{ for all } \alpha \in A_{\tau} \quad y \xrightarrow{\beta} y'}{x;q \xrightarrow{\beta} y'}$$

Suppose that both arguments of $;$ are chosen to be \aleph -liquid and only the first argument of $;$ is chosen to be Λ -liquid. Then it is straightforward to verify that both rules are rooted stability-respecting branching bisimulation safe w.r.t. \aleph and Λ (see Def. 19):

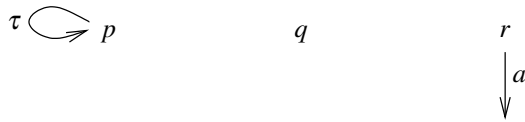
1. The right-hand side x' of the positive premise in the first rule occurs Λ -liquid in $x';y$.
2. In both rules, the variable x has only Λ -liquid occurrences.
3. In both rules, both variables x and y have \aleph -liquid occurrences in the source.
4. In both rules, only the variable x has exactly one \aleph -liquid occurrence in the source that is also Λ -liquid.
 - (a) The variable x does not have more than one \aleph -liquid occurrence in the positive premises of the rules. It does have \aleph -liquid occurrences in negative premises of the second rule, but, since α ranges over all actions in A_{τ} , there is also a premise $x \not\xrightarrow{\tau}$.
 - (b) Clearly the first rule, which has a premise $x \xrightarrow{\tau} x'$ with an \aleph -liquid occurrence of x , is $\aleph \cap \Lambda$ -patient.

Corollary 30 \xleftrightarrow{rb}^s is a congruence for the sequencing operator.

Note that we cannot take Λ to be universal, for then by condition 4b we would need the second rule for $\beta = \tau$ to be $\aleph \cap \Lambda$ -patient. Yet, as is easy to check, \xleftrightarrow{b}^s is a congruence for sequencing. This shows that our (unrooted) stability-respecting branching bisimulation format does not cover all relevant operators from the literature.

Sequencing does not preserve [rooted] branching bisimilarity, as shown by the following example.

Example 31 Consider the following three processes:



Then $p \xleftrightarrow{b} q$, but $p;r = p \not\xleftrightarrow{b} r = q;r$. Note that $p \not\xleftrightarrow{b}^s q$ since q is stable while p cannot perform a sequence of τ -transitions to a stable state.

Using processes p_0 and q_0 as in Ex. 29 shows that also \xleftrightarrow{rb} fails to be a congruence for sequencing.

4 Divergence-preserving branching bisimilarity as a congruence

To get a modal characterisation of (weakly) divergence-preserving branching bisimilarity, a modality needs to be added to the modal logic that captures divergence. A modal characterisation of divergence-preserving branching bisimilarity would be obtained by adding a unary modality Δ to the modal logic

for branching bisimilarity, with the interpretation $p \models \Delta\phi$ if there is an infinite trace $p = p_0 \xrightarrow{\tau} p_1 \xrightarrow{\tau} p_2 \xrightarrow{\tau} \dots$ such that $p_i \models \phi$ for all $i \in \mathbb{N}$.

Modal formulas $\Delta\phi$ to capture divergence elude the inductive decomposition method from Def. 17, because they ask for the existence of an infinite sequence of τ -transitions. The following example shows that the decomposition method does not readily extend to modalities $\Delta\phi$.

Example 32 Let $A = \{a_i, b_i \mid i \in \mathbb{N}\} \cup \{c\}$. The parallel composition operator \parallel is usually defined by the following two rules, where α ranges over A :

$$\frac{x_1 \xrightarrow{\alpha} y}{x_1 \parallel x_2 \xrightarrow{\alpha} y \parallel x_2} \qquad \frac{x_2 \xrightarrow{\alpha} y}{x_1 \parallel x_2 \xrightarrow{\alpha} x_1 \parallel y}$$

We extend the operational semantics of this operator with (asymmetric) communication rules, for all $i \in \mathbb{N}$:

$$\frac{x_1 \xrightarrow{a_i} y_1 \quad x_2 \xrightarrow{b_i} y_2}{x_1 \parallel x_2 \xrightarrow{c} y_1 \parallel y_2}$$

Consider the following two collections of processes p_i, q_i for $i \in \mathbb{N}$. We define $p_i \xrightarrow{\tau} p_{i+1}$ and $p_i \xrightarrow{a_i} \mathbf{0}$ and $q_i \xrightarrow{\tau} q_{i+1}$ and $q_i \xrightarrow{b_i} \mathbf{0}$ for all $i \in \mathbb{N}$. We have $p_0 \parallel q_0 \models \Delta(\langle \varepsilon \rangle \langle c \rangle \top)$. There is no obvious way to decompose this modal property of $p_0 \parallel q_0$ into modal properties of its arguments p_0 and q_0 .

We circumvent this problem by introducing so-called *abstraction-free* TSSs that allow only patience rules and rules without premises to carry a conclusion with the label τ , and by introducing *oracle transitions* $p \xrightarrow{\omega} \checkmark$, where the transition label ω reveals some pertinent information on the behaviour of the process p , such as whether p can diverge. On abstraction-free TSSs with appropriate oracle transitions the equivalences \leftrightarrow_b^s and \leftrightarrow_b^Δ coincide; so there our congruence format for \leftrightarrow_b^s is also a congruence format for \leftrightarrow_b^Δ . We extend this observation to general TSSs by encoding any given TSS into an abstraction-free TSS with oracle transitions, and conclude that on any TSS in the stability-respecting branching bisimulation format \leftrightarrow_b^Δ is a congruence. The same proof strategy shows that on any such TSS also $\leftrightarrow_b^{\Delta\top}$ is a congruence, and it extends to the rooted case.

In Sect. 4.1 we present the method sketched above in more detail. We refrain from introducing the needed machinery, but state exactly which properties of this machinery we require, and prove our main congruence result based on these properties. Instead of dealing with specific equivalences \leftrightarrow_b^Δ and \leftrightarrow_b^s , we work with parametric equivalences \sim and \approx where \sim is finer than \approx ; they will later be instantiated with \leftrightarrow_b^Δ and \leftrightarrow_b^s . This allows a reuse of our work with $\leftrightarrow_b^{\Delta\top}$ and \leftrightarrow_b^s in the roles of \sim and \approx , as well as with $\leftrightarrow_{rb}^{\Delta\top}$ and $\leftrightarrow_{rb}^\Delta$ in the role of \sim and \leftrightarrow_{rb}^s in the role of \approx .

In Sect. 4.2 we introduce the machinery needed in Sect. 4.1 and show that it has the required properties, except for those that depend on the choice of \sim and \approx . Finally, in Sect. 4.3 we apply our framework to derive congruence formats for $\leftrightarrow_b^{\Delta\top}$, \leftrightarrow_b^Δ , $\leftrightarrow_{rb}^{\Delta\top}$ and $\leftrightarrow_{rb}^\Delta$.

4.1 A general framework for lifting congruence formats to finer equivalences

Our general proof idea is illustrated in Fig. 1. Here P (on the left) is a TSS in our congruence format for \approx . We want to show that on P also \sim is a congruence. So consider an operator f , for simplicity depicted as unary. Given two processes p and q in P (closed terms in its signature) with $p \sim q$, we need to show that $f(p) \sim f(q)$. Fig. 1 shows a roundabout trajectory from p to $f(p)$. Imagine a similar trajectory from q to $f(q)$ —not depicted in Fig. 1, but hovering above the page.

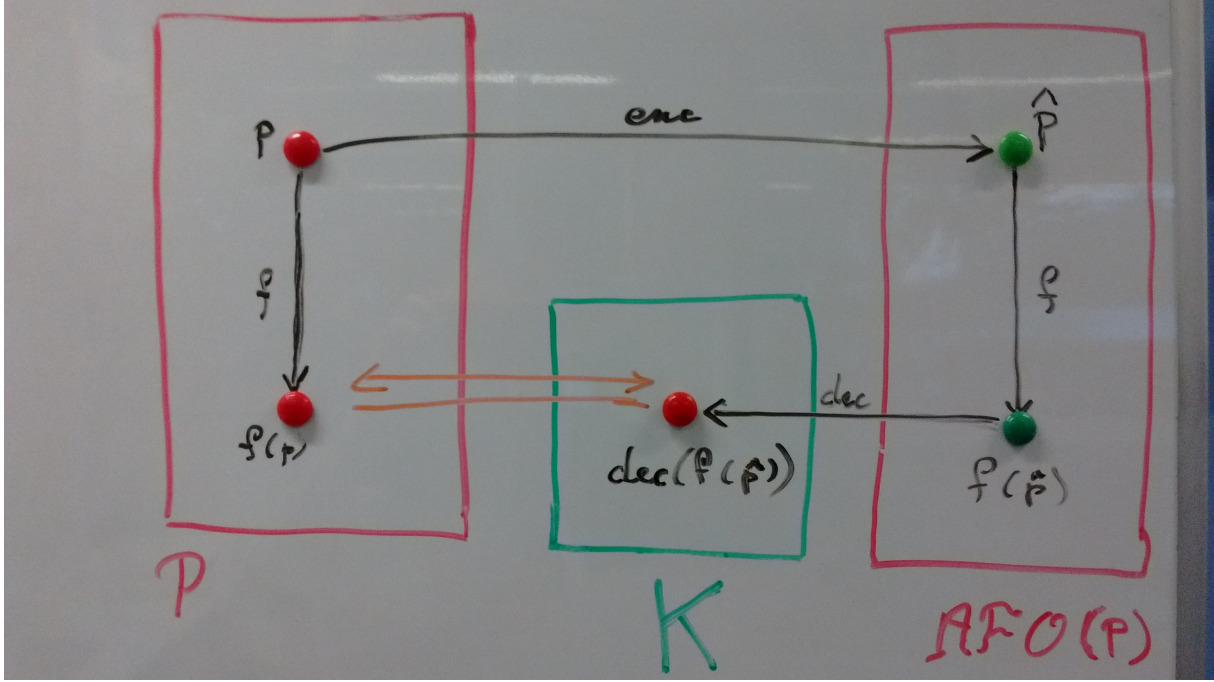


Figure 1: Proof idea

First we apply a transformation \mathcal{AFO} on P , yielding the abstraction-free TSS with oracle transitions $\mathcal{AFO}(P)$, depicted on the right. For each closed term p of P we introduce a constant \hat{p} in $\mathcal{AFO}(P)$, in such a way that $p \sim q$ implies $\hat{p} \sim \hat{q}$. Each n -ary operator f of P remains an n -ary operator f of $\mathcal{AFO}(P)$. Since $\sim \subseteq \approx$ we have $\hat{p} \approx \hat{q}$. We argue that if P is within our congruence format for \approx , then the TSS $\mathcal{AFO}(P)$ is also within this congruence format, and conclude from $\hat{p} \approx \hat{q}$ that $f(\hat{p}) \approx f(\hat{q})$. An important result, deferred to Sect. 4.3, is that on $\mathcal{AFO}(P)$ the equivalences \sim and \approx coincide. Hence $f(\hat{p}) \sim f(\hat{q})$.

Finally, we decode the processes $f(\hat{p})$ and $f(\hat{q})$, aiming to return to the LTS generated by P , but actually ending up in another LTS K . Our decoding function dec exactly undoes the effects of the encoding enc , that sent p to \hat{p} , so that $dec(f(\hat{p}))$ is strongly bisimilar with $f(p)$. A crucial property of the function dec , also deferred to Sect. 4.3, is that it is compositional for \sim , meaning that from $f(\hat{p}) \sim f(\hat{q})$ we may conclude $dec(f(\hat{p})) \sim dec(f(\hat{q}))$. By imposing the requirement that \sim contains strong bisimilarity, this implies that $f(p) \sim f(q)$.

We now formalise this proof idea. An LTS G is called *disjoint* from a complete TSS P if it is disjoint from the TSS G_P associated with P . In that case $P \uplus G$ denotes the union of G_P and G (cf. Def. 4).

Theorem 33 Let \sim and \approx be behavioural equivalences on LTSs, with $\Leftrightarrow \subseteq \sim \subseteq \approx$. Let \mathfrak{F} be a congruence format for \approx , included in the decent ntyft format, and let \mathcal{AFO} be an operation on standard TSSs, where for each TSS $P = (\Sigma, Act, R)$ the signature $\hat{\Sigma}$ of $\mathcal{AFO}(P)$ contains Σ enriched by a fresh constant \hat{p} for each closed term p in $T(\Sigma)$, such that, for each complete standard TSS P in decent ntyft format:

1. also $\mathcal{AFO}(P)$ is a complete standard TSS,
2. if P is in \mathfrak{F} -format then so is $\mathcal{AFO}(P)$,
3. $p \sim_P q \Rightarrow \hat{p} \sim_{\mathcal{AFO}(P)} \hat{q}$,
4. $\sim_{\mathcal{AFO}(P)}$ and $\approx_{\mathcal{AFO}(P)}$ coincide, and

there is an LTS $K = (\mathbb{P}_K, Act_K, \rightarrow_K)$, disjoint from P , as well as a function $dec : T(\hat{\Sigma}) \rightarrow \mathbb{P}_K$ such that:

5. $p \sim_{\mathcal{AF}\mathcal{O}(P)} q \Rightarrow dec(p) \sim_K dec(q)$, and
6. $f(p_1, \dots, p_n) \xrightarrow{P \sqcup K} dec(f(\hat{p}_1, \dots, \hat{p}_n))$ for any n -ary $f \in \Sigma$ and $p_1, \dots, p_n \in T(\Sigma)$.

Then \mathfrak{F} is also a congruence format for \sim .

Proof: Let $P = (\Sigma, Act, R)$ be a complete standard TSS in \mathfrak{F} -format. We will show that \sim_P is a congruence for P . So let $f \in \Sigma$ be an n -ary function symbol, and let $p_i, q_i \in T(\Sigma)$ with $p_i \sim_P q_i$ for $i = 1, \dots, n$. We need to show that $f(p_1, \dots, p_n) \sim_P f(q_1, \dots, q_n)$.

By requirements 1 and 2, $\mathcal{AF}\mathcal{O}(P)$ is a complete standard TSS in \mathfrak{F} -format; hence $\sim_{\mathcal{AF}\mathcal{O}(P)}$ is a congruence for $\mathcal{AF}\mathcal{O}(P)$. By requirement 3 $\hat{p}_i \sim_{\mathcal{AF}\mathcal{O}(P)} \hat{q}_i$ for $i = 1, \dots, n$. By requirement 4 also $\sim_{\mathcal{AF}\mathcal{O}(P)}$ is a congruence for $\mathcal{AF}\mathcal{O}(P)$. So $f(\hat{p}_1, \dots, \hat{p}_n) \sim_{\mathcal{AF}\mathcal{O}(P)} f(\hat{q}_1, \dots, \hat{q}_n)$. Hence, by requirement 5,

$$dec(f(\hat{p}_1, \dots, \hat{p}_n)) \sim_K dec(f(\hat{q}_1, \dots, \hat{q}_n)).$$

Therefore, by two applications of requirement 6, and the definition of a behavioural equivalence, $f(p_1, \dots, p_n) \sim_{P \sqcup K} f(q_1, \dots, q_n)$ and consequently $f(p_1, \dots, p_n) \sim_P f(q_1, \dots, q_n)$. \square

4.2 Abstraction-freeness

In this section we introduce the machinery needed for Thm. 33, namely the conversion $\mathcal{AF}\mathcal{O}$ on TSSs and the function dec into the LTS K . We also establish requirements 1 and 6 of Thm. 33, leaving 2–5 to the applications of Thm. 33 in Sect. 4.3 for specific instances of \sim and \approx . Since we are only interested in TSSs with τ -transitions, we here take the set of actions Act used in Sect. 4.1 to be A_τ .

4.2.1 The conversion $\mathcal{AF}\mathcal{O}$

Let again Γ denote a predicate that marks the (Γ -liquid) arguments of function symbols. In [12] we called a standard TSS *abstraction-free* w.r.t. Γ if only its Γ -patience rules carry the label τ in their conclusion. Here we use a slightly more liberal definition of abstraction-freeness that also allows rules that have no premises, and a conclusion of the form $c \xrightarrow{\tau} d$ for constants c and d .

The following conversion turns any Γ -patient standard TSS $P = (\Sigma, A_\tau, R)$ into a Γ -patient and abstraction-free TSS $\mathcal{AF}\mathcal{O}_\Gamma^{O, \zeta}(P)$. It is parameterised by the choice of a fresh set of actions O , so $O \cap A_\tau = \emptyset$, and a partial function $\zeta : T(\Sigma) \rightarrow O$, which we call an *oracle*. The choice of O and ζ varies for different applications of Thm. 33. This choice will be made in Sect. 4.3 in such a way that requirements 3 and 4 of Thm. 33 are met, for specific instances of \sim and \approx .

Definition 34 Given a Γ -patient standard TSS $P = (\Sigma, A_\tau, R)$. Let $\hat{\Sigma}$ be the signature Σ , enriched with a fresh constant \surd and a fresh constant \hat{p} for each closed term $p \in T(\Sigma)$. Pick a fresh action $\iota \notin A_\tau \cup O$. We define the TSS $\mathcal{AF}\mathcal{O}_\Gamma^{O, \zeta}(P)$ as $(\hat{\Sigma}, A_\tau \cup O \cup \{\iota\}, R')$ where the rules in R' are obtained from the rules in R as follows:

1. R_1 is obtained from R by adding for each rule r and each non-empty subset S of positive τ -premises of r , a copy of r with as only difference that the labels τ in the premises in S are replaced by ι ;
2. R_2 is obtained from R_1 by replacing, in every rule that has a conclusion with the label τ and is not a Γ -patience rule, the τ -label in the conclusion by ι ;
3. R_3 is obtained from R_2 by adding the premise $v \xrightarrow{\iota} \surd$ to each rule with a negative τ -premise $v \xrightarrow{\tau} \surd$;

4. R_4 is obtained from R_3 by the addition of a rule without premises $\hat{p} \xrightarrow{\alpha} \hat{q}$ for each transition $p \xrightarrow{\alpha} q$ ws-provable from P ;
5. R_5 is obtained from R_4 by the addition of a rule $\hat{p} \xrightarrow{\zeta(p)} \checkmark$ for each $p \in T(\Sigma)$ with $\zeta(p)$ defined;
6. R' adds to R_5 a rule $\frac{x_k \xrightarrow{\omega} y}{f(x_1, \dots, x_n) \xrightarrow{\omega} y}$ for each $\omega \in O$, $f \in \Sigma$ and argument k with $\Gamma(f, k)$.

Step 2 above makes the resulting TSS abstraction-free by renaming τ -labels in conclusions of non-patience rules into ι . To ensure that still the same transitions are derived, modulo the conversion of some τ into ι -labels, step 1 above allows positive premises labelled ι to be used instead of τ in all rules, and step 3 achieves the same purpose for negative premises. These three steps result in a Γ -patient and abstraction-free TSS that could be called $\mathcal{AF}_\Gamma(P)$.

Conceptually, steps 1, 2 and 3 on the one hand and steps 4 and 5 on the other hand are independent. Listing steps 1, 2 and 3 before steps 4 and 5 makes it evident that the transformations 1–3 do not apply to the rules added by 4 and 5.

For convenience in proofs, and to better explain steps 4 and 5 of Def. 34, we consider an auxiliary LTS $G = (\mathbb{P}_G, A_\tau, \rightarrow_G)$ with $\mathbb{P}_G = \{\hat{p} \mid p \in T(\Sigma)\}$ and $\hat{p} \xrightarrow{\alpha}_G \hat{q}$ iff $P \vdash_{ws} p \xrightarrow{\alpha} q$, and an auxiliary LTS $H = (\mathbb{P}_H, A_\tau \cup O, \rightarrow_H)$ with $\mathbb{P}_H = \mathbb{P}_G \cup \{\checkmark\}$ and $\rightarrow_H := \rightarrow_G \cup \{\hat{p} \xrightarrow{\zeta(p)} \checkmark \mid \zeta(p) \text{ defined}\}$. The LTS G is simply a disjoint copy of the LTS generated by P .

Lemma 35 If $p \sim_P q$ for some $p, q \in T(\Sigma)$, then $\hat{p} \sim_G \hat{q}$.

Proof: We have $p \xleftrightarrow{P \oplus G} \hat{p}$ for each $p \in T(\Sigma)$, because the relation $\{(p, \hat{p}), (\hat{p}, p) \mid p \in T(\Sigma)\}$ is a strong bisimulation. So $\hat{p} \xleftrightarrow{P \oplus G} p \sim_{P \oplus G} q \xleftrightarrow{P \oplus G} \hat{q}$, using the definition of a behavioural equivalence, and thus $\hat{p} \sim_{P \oplus G} \hat{q}$, using that $\xleftrightarrow{} \subseteq \sim$. Hence $\hat{p} \sim_G \hat{q}$, again by the definition of a behavioural equivalence. \square

The LTS H adds *oracle transitions* to G . The idea is that $\zeta(p)$ is particular for the \sim -equivalence class of $p \in T(\Sigma)$, which on the one hand ensures that $\hat{p} \sim_H \hat{q}$ iff $\hat{p} \sim_G \hat{q}$, and on the other hand enforces that \sim and \approx coincide on H . Namely, if $\hat{p} \approx_H \hat{q}$ then the oracle action of \hat{p} can be matched by \hat{q} (and vice versa), which implies $\hat{p} \sim_H \hat{q}$.

Example 36 Take \sim to be weakly divergence-preserving branching bisimilarity, and \approx to be stability-respecting branching bisimilarity. Let us say that a process p in an LTS is *divergent* if there exists an infinite sequence of processes $(p_k)_{k \in \mathbb{N}}$ such that $p_k \xrightarrow{\tau} p_{k+1}$ for all $k \in \mathbb{N}$, i.e. if $p \models \Delta\top$. Take $O = \{\Delta\top\}$ and let $\zeta(p) = \Delta\top$ iff p is divergent. Thus in H all divergent states of G have a fresh outgoing transition labelled $\Delta\top$.

With this definition of H , we have $\hat{p} \sim_H \hat{q}$ iff $\hat{p} \sim_G \hat{q}$: any weakly divergence-preserving branching bisimulation \mathcal{B} on G relates divergent states with divergent states only, and thus is also a weakly divergence-preserving branching bisimulation on H (when adding $\checkmark \mathcal{B} \checkmark$). Furthermore, as we will show in Prop. 51, due to the construction of H , $\hat{p} \sim_H \hat{q}$ iff $\hat{p} \approx_H \hat{q}$.

Steps 4 and 5 of Def. 34 incorporate the entire LTS H into $\mathcal{AF}_\Gamma(P)$: each state appears as a constant and each transition appears as rule without premises. The operators from Σ can now be applied to arguments of the form \hat{p} . Finally, step 6 lets any term $f(x_1, \dots, x_n)$ inherit the oracle transitions from its Γ -liquid arguments. Steps 4, 5 and 6 preserve abstraction-freeness; for step 4 this uses the relaxed definition of abstraction-freeness that allows to incorporate τ -transitions between constants as rules without premises.

Example 37 Let P have the rules

$$\frac{x_1 \xrightarrow{\tau} y}{g(x_1, x_2, x_3) \xrightarrow{\tau} g(y, x_2, x_3)} \quad \frac{x_1 \xrightarrow{a} y_1 \quad x_1 \xrightarrow{\tau} y_2 \quad x_3 \xrightarrow{\tau} y_3}{g(x_1, x_2, x_3) \xrightarrow{\tau} x_2} \quad \frac{x_2 \xrightarrow{\tau} y \quad x_3 \xrightarrow{\tau} y}{g(x_1, x_2, x_3) \xrightarrow{a} y}$$

where $\Gamma(g, 1)$. Then $\mathcal{AF}\mathcal{O}_\Gamma^{O, \zeta}(P)$ has the rules

$$\begin{array}{c}
\frac{x_1 \xrightarrow{\tau} y}{g(x_1, x_2, x_3) \xrightarrow{\tau} g(y, x_2, x_3)} \quad \frac{x_1 \xrightarrow{a} y_1 \quad x_1 \xrightarrow{\tau} y_2 \quad x_3 \xrightarrow{\tau} y_3}{g(x_1, x_2, x_3) \xrightarrow{l} x_2} \quad \frac{x_2 \xrightarrow{\tau} y \quad x_3 \xrightarrow{\tau} y \quad x_3 \xrightarrow{l} y}{g(x_1, x_2, x_3) \xrightarrow{a} y} \\
\\
\frac{x_1 \xrightarrow{l} y}{g(x_1, x_2, x_3) \xrightarrow{l} g(y, x_2, x_3)} \quad \frac{x_1 \xrightarrow{a} y_1 \quad x_1 \xrightarrow{l} y_2 \quad x_3 \xrightarrow{l} y_3}{g(x_1, x_2, x_3) \xrightarrow{l} x_2} \quad \frac{x_2 \xrightarrow{l} y \quad x_3 \xrightarrow{\tau} y \quad x_3 \xrightarrow{l} y}{g(x_1, x_2, x_3) \xrightarrow{a} y} \\
\\
\frac{x_1 \xrightarrow{a} y_1 \quad x_1 \xrightarrow{l} y_2 \quad x_3 \xrightarrow{\tau} y_3}{g(x_1, x_2, x_3) \xrightarrow{l} x_2} \quad \frac{x_1 \xrightarrow{a} y_1 \quad x_1 \xrightarrow{\tau} y_2 \quad x_3 \xrightarrow{l} y_3}{g(x_1, x_2, x_3) \xrightarrow{l} x_2} \quad \frac{x_1 \xrightarrow{\omega} y}{g(x_1, x_2, x_3) \xrightarrow{\omega} y} \quad (\omega \in O)
\end{array}$$

This example illustrates steps 1, 2, 3 and 6 of Def. 34 only. Since there are no closed terms in this example, steps 4 and 5 are void.

Clearly, for any Γ -patient standard TSS P , the standard TSS $\mathcal{AF}\mathcal{O}_\Gamma^{O, \zeta}(P)$ is Γ -patient and abstraction-free w.r.t. Γ . Henceforth, we drop the superscripts O and ζ , and $\mathcal{AF}\mathcal{O}(P)$ denotes $\mathcal{AF}\mathcal{O}_\Gamma(P)$ for the largest predicate Γ for which P is Γ -patient. The signature of $\mathcal{AF}\mathcal{O}(P)$ contains the signature of P enriched by a fresh constant \hat{p} for each closed term p in P , as required in Thm. 33.

Lemma 38 Let P be a complete standard TSS in ntyft format. If $\hat{p} \sim_H \hat{q}$ for some $p, q \in T(\Sigma)$, then $\hat{p} \sim_{\mathcal{AF}\mathcal{O}(P)} \hat{q}$.

Proof: Since $\mathcal{AF}\mathcal{O}(P)$ has no rules whose source is a variable, the only derivable transitions of the form $\hat{p} \xrightarrow{\alpha} q^*$ with $\alpha \in A_\tau \cup O \cup \{\iota\}$ are the ones from H, with q^* of the form \hat{q} or $\sqrt{}$. For this reason any process \hat{p} in H is strongly bisimilar to the process \hat{p} in $\mathcal{AF}\mathcal{O}(P)$. Using this, the lemma follows just as Lem. 35. \square

As an immediate consequence of Lemmas 35 and 38 we have the following corollary.

Corollary 39 Requirement 3 of Thm. 33 is met if $\hat{p} \sim_G \hat{q}$ implies $\hat{p} \sim_H \hat{q}$. \square

The inference $\hat{p} \sim_G \hat{q} \Rightarrow \hat{p} \sim_H \hat{q}$ depends on the choice of O and ζ , and is deferred to Sect. 4.3.

The oracle inheritance rules in $\mathcal{AF}\mathcal{O}_\Gamma(P)$ —introduced in step 6 of Def. 34—ensure that a closed term $f(p_1, \dots, p_n)$ has an outgoing ω -transition, for $\omega \in O$, iff one of its Γ -liquid arguments p_k has such a transition. Ultimately, all such oracle transitions stem from H. Using that in $\mathcal{AF}\mathcal{O}_\Gamma(P)$ any term $p \in T(\hat{\Sigma})$ can uniquely be written as $\rho(t)$ with $t \in T(\Sigma)$ and $\rho : \text{var}(t) \rightarrow \mathbb{P}_H$, this observation can be phrased as follows.

Observation 40 Let $t \in T(\Sigma)$ and $\rho : \text{var}(t) \rightarrow \mathbb{P}_H$; then $\mathcal{AF}\mathcal{O}_\Gamma(P) \vdash_{ws} \rho(t) \xrightarrow{\omega}$ iff t has a Γ -liquid occurrence of a variable x with $\rho(x) \xrightarrow{\omega}_H$. \square

We proceed to compare provability in a standard TSS P with provability in its abstraction-free counterpart $\mathcal{AF}\mathcal{O}(P)$, in order to verify requirements 1 and 6 of Thm. 33. The most laborious part in this comparison lays in step 4 of Def. 34. Therefore we deal with this step separately from the other five. Hence, we define for any standard TSS $P = (\Sigma, A_\tau, R)$ a TSS $\mathcal{G}(P) = (\hat{\Sigma}, A_\tau, R')$, which constitutes an intermediate between P and $\mathcal{AF}\mathcal{O}(P)$. It is built by only applying step 4 from the construction of $\mathcal{AF}\mathcal{O}(P)$, with R in the role of R_3 .

4.2.2 Comparison of P and $\mathcal{G}(P)$

We restrict our analysis to standard TSSs $P = (\Sigma, A_\tau, R)$ in decent ntyft format. For $p \in T(\hat{\Sigma})$, let $\check{p} \in T(\Sigma)$ be obtained from p by replacing every subterm \hat{q} in p by q . Likewise, for N a set of closed negative literals, $\check{N} := \{\check{p} \xrightarrow{\alpha} \mid (p \xrightarrow{\alpha}) \in N\}$.

When we are merely interested in proving, from P or $\mathcal{G}(P)$, literals of the form $p \xrightarrow{\alpha}$ or $p \xrightarrow{\alpha}$, possibly under some hypotheses, where the target of a positive literal is existentially quantified, we can equivalently use a version of P resp. $\mathcal{G}(P)$ in which all right-hand sides of the premises and conclusions of rules have been dropped. Here we use that P and $\mathcal{G}(P)$ are in decent ntyft format. This we do below.

Lemma 41 Let $P = (\Sigma, A_\tau, R)$ be a standard TSS in decent ntyft format. Let $\alpha \in A_\tau$ and $p \in T(\hat{\Sigma})$.

- If $\mathcal{G}(P) \vdash_{irr} \frac{N}{p \xrightarrow{\alpha}}$, with N a set of closed negative literals, then $P \vdash_{irr} \frac{\check{N} \cup N'}{\check{p} \xrightarrow{\alpha}}$ for a set N' of closed negative literals λ with $P \vdash_{ws} \lambda$.
- If $P \vdash_{irr} \frac{N'}{\check{p} \xrightarrow{\alpha}}$, with N' a set of closed negative literals, then either $P \not\vdash_{ws} \lambda$ for a literal $\lambda \in N'$, or $\mathcal{G}(P) \vdash_{irr} \frac{N}{p \xrightarrow{\alpha}}$ for some N with $\check{N} \subseteq N'$.

Proof: For the first statement we apply induction on the irredundant proof π of $\frac{N}{p \xrightarrow{\alpha}}$ from $\mathcal{G}(P)$.

First assume that p has the form \hat{q} , so that $\check{p} = q$. Since $\mathcal{G}(P)$ has no rules whose source is a variable, the last step of π must be an application of a rule $\hat{q} \xrightarrow{\alpha} \hat{r}$, introduced in step 4 of the construction of $\mathcal{G}(P)$, and $N = \emptyset$. It follows that $P \vdash_{ws} q \xrightarrow{\alpha} r$. Let N' be the unique set of negative literals occurring in the well-supported proof of $q \xrightarrow{\alpha} r$ from P that have no negative literals below them. Then $P \vdash_{irr} \frac{N'}{q \xrightarrow{\alpha} r}$ and each literal $\lambda \in N'$ is *ws*-provable from P .

Alternatively, $p = f(p_1, \dots, p_n)$ with $f \in \Sigma$. Then the last step of π must be an application of a decent ntyft rule $\frac{H}{f(x_1, \dots, x_n) \xrightarrow{\alpha}}$ in P and the substitution $\sigma : \{x_1, \dots, x_n\} \rightarrow T(\hat{\Sigma})$ with $\sigma(x_i) = p_i$ for $i = 1, \dots, n$. Let $\sigma' : \{x_1, \dots, x_n\} \rightarrow T(\Sigma)$ be the substitution with $\sigma'(x_i) = \check{p}_i$ for $i = 1, \dots, n$. With every positive premise $\mu = (t \xrightarrow{\gamma})$ in H we may associate a set of closed negative literals $N_\mu \subseteq N$ such that $\mathcal{G}(P) \vdash_{irr} \frac{N_\mu}{\sigma(t) \xrightarrow{\gamma}}$ and such that N is the union of all N_μ . So, by induction, $P \vdash_{irr} \frac{\check{N}_\mu \cup N'_\mu}{\sigma'(t) \xrightarrow{\gamma}}$ for a set N'_μ of closed negative literals λ with $P \vdash_{ws} \lambda$. Let N' be the union of all the N'_μ . Note that \check{N} is the union of all the \check{N}_μ and the literals $\sigma'(u) \xrightarrow{\gamma}$ with $(u \xrightarrow{\gamma}) \in H$. It follows that $P \vdash_{irr} \frac{\check{N} \cup N'}{\check{p} \xrightarrow{\alpha}}$.

For the second statement assume $P \vdash_{irr} \frac{N'}{\check{p} \xrightarrow{\alpha}}$ and $P \vdash_{ws} \lambda$ for all $\lambda \in N'$. Then $P \vdash_{ws} \check{p} \xrightarrow{\alpha}$. We show that $\mathcal{G}(P) \vdash_{irr} \frac{N}{p \xrightarrow{\alpha}}$ for some N with $\check{N} \subseteq N'$, by induction on the irredundant proof π of $\frac{N'}{\check{p} \xrightarrow{\alpha}}$ from P . First assume that p has the form \hat{q} , so that $\check{p} = q$. Then $P \vdash_{ws} q \xrightarrow{\alpha}$, so $\mathcal{G}(P) \vdash_{irr} \hat{q} \xrightarrow{\alpha}$, meeting the requirement of the lemma.

Alternatively, $p = f(p_1, \dots, p_n)$ with $f \in \Sigma$, so that $\check{p} = f(\check{p}_1, \dots, \check{p}_n)$. Then the last step of π must be an application of a decent ntyft rule $\frac{H}{f(x_1, \dots, x_n) \xrightarrow{\alpha}}$ and the substitution $\sigma' : \{x_1, \dots, x_n\} \rightarrow T(\Sigma)$ with $\sigma'(x_i) = \check{p}_i$ for $i = 1, \dots, n$. Let $\sigma : \{x_1, \dots, x_n\} \rightarrow T(\hat{\Sigma})$ be the substitution with $\sigma(x_i) = p_i$ for $i = 1, \dots, n$. For each positive premise $\mu = (t \xrightarrow{\gamma})$ in H we have $P \vdash_{irr} \frac{N'_\mu}{\sigma'(t) \xrightarrow{\gamma}}$ for some $N'_\mu \subseteq N'$ and $P \vdash_{ws} \lambda$ for all $\lambda \in N'_\mu$, and thus, by induction, $\mathcal{G}(P) \vdash_{irr} \frac{N_\mu}{\sigma(t) \xrightarrow{\gamma}}$ for some N_μ with $\check{N}_\mu \subseteq N'_\mu$. Let N be the union of all the N_μ , and the literals $\sigma(u) \xrightarrow{\gamma}$ with $(u \xrightarrow{\gamma}) \in H$. Then $\check{N} \subseteq N'$. It follows that $\mathcal{G}(P) \vdash_{irr} \frac{N}{p \xrightarrow{\alpha}}$. \square

Proposition 42 Let $P = (\Sigma, A_\tau, R)$ be a complete standard TSS in decent ntyft format, $\alpha \in A_\tau$ and $p \in T(\hat{\Sigma})$.

- $P \vdash_{ws} \check{p} \xrightarrow{\alpha} q^* \Leftrightarrow \exists q \in T(\hat{\Sigma}). q^* = \check{q} \wedge \mathcal{G}(P) \vdash_{ws} p \xrightarrow{\alpha} q.$
- $P \vdash_{ws} \check{p} \not\xrightarrow{\alpha} \Leftrightarrow \mathcal{G}(P) \vdash_{ws} p \not\xrightarrow{\alpha}.$

Proof: “ \Rightarrow ”: We prove both statements by simultaneous induction on the well-founded proof π from $\check{p} \xrightarrow{\alpha} q^*$ or $\check{p} \not\xrightarrow{\alpha}$ from P . First consider the case $P \vdash_{ws} \check{p} \xrightarrow{\alpha} q^*$.

Assume that p has the form \hat{r} , so that $\check{p} = r$. It follows immediately from step 4 of the construction of $\mathcal{G}(P)$ that $\mathcal{G}(P) \vdash_{ws} \hat{r} \xrightarrow{\alpha} \hat{q}^*$. So take $q := \hat{q}^*$.

Alternatively, $p = f(p_1, \dots, p_n)$ with $f \in \Sigma$, so that $\check{p} = f(\check{p}_1, \dots, \check{p}_n)$. Then the last step of π must be an application of a decent ntyft rule $\frac{H}{f(x_1, \dots, x_n) \xrightarrow{\alpha} t}$ and a substitution $\sigma' : V \rightarrow T(\Sigma)$ with $\sigma'(x_i) = \check{p}_i$ for $i = 1, \dots, n$ and $\sigma'(t) = q^*$. Let $\sigma'' : \{x_1, \dots, x_n\} \rightarrow T(\hat{\Sigma})$ be the substitution with $\sigma''(x_i) = p_i$ for $i = 1, \dots, n$. For each negative premise $u \not\xrightarrow{\gamma}$ in H we have $P \vdash_{ws} \sigma'(u) \not\xrightarrow{\gamma}$, and thus, by induction, $\mathcal{G}(P) \vdash_{ws} \sigma''(u) \not\xrightarrow{\gamma}$. Likewise, for each positive premise $\mu = (u \xrightarrow{\gamma} y_\mu)$ in H , $P \vdash_{ws} \sigma'(u) \xrightarrow{\gamma} \sigma'(y_\mu)$, and thus, by induction, $\mathcal{G}(P) \vdash_{ws} \sigma''(u) \xrightarrow{\gamma} r_\mu$ for some r_μ with $\check{r}_\mu = \sigma'(y_\mu)$. Let $\sigma : V \rightarrow T(\hat{\Sigma})$ be a substitution with $\sigma(x_i) = p_i$ for $i = 1, \dots, n$ and $\sigma(y_\mu) = r_\mu$ for each positive premise μ in H . Then $\mathcal{G}(P) \vdash_{ws} \sigma(\mu)$ for each premise μ in H , and thus $\mathcal{G}(P) \vdash_{ws} p \xrightarrow{\alpha} \sigma(t)$. Take $q := \sigma(t)$. Then $\check{q} = \sigma'(t) = q^*$.

Next consider the case $P \vdash_{ws} \check{p} \not\xrightarrow{\alpha}$. Suppose that $\frac{N}{p \xrightarrow{\alpha}}$ is irredundantly provable from $\mathcal{G}(P)$. Then, by Lem. 41, $P \vdash_{irr} \frac{\check{N} \cup N'}{\check{p} \xrightarrow{\alpha}}$ for a set N' of closed negative literals λ with $P \vdash_{ws} \lambda$. By Def. 11 $\check{N} \cup N'$ contains a literal $r' \not\xrightarrow{\gamma}$ such that $r' \not\xrightarrow{\gamma}$ is ws -provable from P by means of a strict subproof of π . By the consistency of \vdash_{ws} , $(r' \not\xrightarrow{\gamma}) \notin N'$. Hence N contains a literal $r \not\xrightarrow{\gamma}$ with $\check{r} = r'$. By induction $\mathcal{G}(P) \vdash_{ws} r \xrightarrow{\gamma}$, and this literal denies a literal in N . From this it follows that $\mathcal{G}(P) \vdash_{ws} p \not\xrightarrow{\alpha}$.

“ \Leftarrow ”: We prove both statements by simultaneous induction on the well-founded proof π of $p \xrightarrow{\alpha} q$ or $p \not\xrightarrow{\alpha}$ from $\mathcal{G}(P)$. First consider the case $\mathcal{G}(P) \vdash_{ws} p \xrightarrow{\alpha} q$.

Assume that p has the form \hat{r} , so that $\check{p} = r$. Since $\mathcal{G}(P)$ has no rules whose source is a variable, the last step of π must be an application of a rule $\hat{r} \xrightarrow{\alpha} \hat{s}$, introduced in step 4 of the construction of $\mathcal{G}(P)$. It follows that $q = \hat{s}$, so $\check{q} = s$, and $P \vdash_{ws} r \xrightarrow{\alpha} s$.

Alternatively, $p = f(p_1, \dots, p_n)$ with $f \in \Sigma$, so that $\check{p} = f(\check{p}_1, \dots, \check{p}_n)$. Then the last step of π must be an application of a decent ntyft rule $\frac{H}{f(x_1, \dots, x_n) \xrightarrow{\alpha} t}$ and a substitution $\sigma : V \rightarrow T(\hat{\Sigma})$ with $\sigma(x_i) = p_i$ for $i = 1, \dots, n$ and $\sigma(t) = q$. Let $\sigma' : V \rightarrow T(\Sigma)$ be the substitution with $\sigma'(x) = \check{\sigma}(x)$ for each $x \in V$ such that $\sigma(x)$ is defined, and undefined otherwise. For each premise μ in H we have $\mathcal{G}(P) \vdash_{ws} \sigma(\mu)$, by means of a subproof of π , and thus, by induction, $P \vdash_{ws} \sigma'(\mu)$. It follows that $P \vdash_{ws} \check{p} \xrightarrow{\alpha} \check{q}$.

Finally, consider the case $\mathcal{G}(P) \vdash_{ws} p \not\xrightarrow{\alpha}$. Suppose that $\frac{N'}{\check{p} \xrightarrow{\alpha}}$ is irredundantly provable from P . Then, by Lem. 41, either $P \not\vdash_{ws} \lambda$ for a literal $\lambda \in N'$, or $\mathcal{G}(P) \vdash_{irr} \frac{N}{p \xrightarrow{\alpha}}$ for some N with $\check{N} \subseteq N'$. In the first case, by the completeness of P , $P \vdash_{ws} \mu$ for a literal μ denying λ , and we are done. So assume the second case. By Def. 11 N contains a literal $r \not\xrightarrow{\gamma}$ such that $r \not\xrightarrow{\gamma}$ is ws -provable from $\mathcal{G}(P)$ by means of a strict subproof of π . By induction $P \vdash_{ws} \check{r} \xrightarrow{\gamma}$, and this literal denies a literal in N' . From this it follows that $P \vdash_{ws} \check{p} \not\xrightarrow{\alpha}$. \square

Corollary 43 Let $P = (\Sigma, A_\tau, R)$ be a complete standard TSS in decent ntyft format. Then also $\mathcal{G}(P)$ is complete.

Proof: Let $p \in T(\hat{\Sigma})$ and $\alpha \in A_\tau$. Since P is complete, either $P \vdash_{ws} \check{p} \xrightarrow{\alpha}$ or $P \vdash_{ws} \check{p} \not\xrightarrow{\alpha}$. Consequently, by Prop. 42, either $\mathcal{G}(P) \vdash_{ws} p \xrightarrow{\alpha}$ or $\mathcal{G}(P) \vdash_{ws} p \not\xrightarrow{\alpha}$. \square

4.2.3 Comparison of $\mathcal{G}(P)$ and $\mathcal{AF}\mathcal{O}(P)$

$\mathcal{G}(P)$ and $\mathcal{AF}\mathcal{O}(P)$ differ on the account of rules with premises and conclusions labelled ι and $\omega \in O$. We note that actions $\omega \in O$ play no role in the next lemma and proposition.

Lemma 44 Let $P = (\Sigma, A_\tau, R)$ be a standard TSS in ntyft format. Let a range over A and p, q over $T(\hat{\Sigma})$.

- $\mathcal{AF}\mathcal{O}(P) \vdash_{irr} \frac{N}{p \xrightarrow{a} q}$ for a set of closed negative literals N iff there is a set of closed negative literals N' such that $\mathcal{G}(P) \vdash_{irr} \frac{N'}{p \xrightarrow{a} q}$ and $N = \{s \xrightarrow{a} \mid (s \xrightarrow{a} \rightarrow) \in N'\} \cup \{t \xrightarrow{\tau} \rightarrow, t \xrightarrow{l} \rightarrow \mid (t \xrightarrow{\tau} \rightarrow) \in N'\}$.
- $\mathcal{AF}\mathcal{O}(P) \vdash_{irr} \frac{N}{p \xrightarrow{\alpha} q}$, with $\alpha = \tau$ or $\alpha = i$, for a set of closed negative literals N , iff there is an N' such that $\mathcal{G}(P) \vdash_{irr} \frac{N'}{p \xrightarrow{\alpha} q}$ and $N = \{s \xrightarrow{a} \mid (s \xrightarrow{a} \rightarrow) \in N'\} \cup \{t \xrightarrow{\tau} \rightarrow, t \xrightarrow{l} \rightarrow \mid (t \xrightarrow{\tau} \rightarrow) \in N'\}$.

Proof: “If”: Both statements follow by a straightforward simultaneous induction on irredundant provability from $\mathcal{G}(P)$.

“Only if”: Both statements follow by a straightforward simultaneous induction on irredundant provability from $\mathcal{AF}\mathcal{O}(P)$. \square

Proposition 45 Let $P = (\Sigma, A_\tau, R)$ be a standard TSS in ntyft format. Let a range over A and p, q over $T(\hat{\Sigma})$.

- $\mathcal{G}(P) \vdash_{ws} p \xrightarrow{a} q \Leftrightarrow \mathcal{AF}\mathcal{O}(P) \vdash_{ws} p \xrightarrow{a} q$.
- $\mathcal{G}(P) \vdash_{ws} p \xrightarrow{\tau} q \Leftrightarrow (\mathcal{AF}\mathcal{O}(P) \vdash_{ws} p \xrightarrow{\tau} q \vee \mathcal{AF}\mathcal{O}(P) \vdash_{ws} p \xrightarrow{l} q)$.
- $\mathcal{G}(P) \vdash_{ws} p \xrightarrow{a} \rightarrow \Leftrightarrow \mathcal{AF}\mathcal{O}(P) \vdash_{ws} p \xrightarrow{a} \rightarrow$.
- $\mathcal{G}(P) \vdash_{ws} p \xrightarrow{\tau} \rightarrow \Leftrightarrow (\mathcal{AF}\mathcal{O}(P) \vdash_{ws} p \xrightarrow{\tau} \rightarrow \wedge \mathcal{AF}\mathcal{O}(P) \vdash_{ws} p \xrightarrow{l} \rightarrow)$.

Proof: Using Lem. 44, the lemma follows by two straightforward inductions on well-supported provability, one for \Rightarrow , covering all four claims, and one for \Leftarrow . \square

Corollary 46 Let P be a standard TSS in decent ntyft format. If P is complete, then so is $\mathcal{AF}\mathcal{O}(P)$.

Proof: By Cor. 43, $\mathcal{G}(P)$ is complete. Hence, for each $p \in T(\hat{\Sigma})$ and $a \in A$, either $\mathcal{G}(P) \vdash_{ws} p \xrightarrow{a} \rightarrow$ or $\mathcal{G}(P) \vdash_{ws} p \xrightarrow{a} q$. So by Prop. 45, either $\mathcal{AF}\mathcal{O}(P) \vdash_{ws} p \xrightarrow{a} \rightarrow$ or $\mathcal{AF}\mathcal{O}(P) \vdash_{ws} p \xrightarrow{a} q$.

Each $p \in T(\hat{\Sigma})$ can be written as $\rho(t)$ with $t \in T(\Sigma)$ and $\rho : \text{var}(t) \rightarrow \mathbb{P}_H$. Let Γ be the largest predicate for which P is Γ -patient. Now $\mathcal{AF}\mathcal{O}_\Gamma(P) \vdash_{ws} \rho(t) \xrightarrow{\tau}$ if t has a Γ -liquid argument x for which $\rho(x) \xrightarrow{\tau} \rightarrow_H$. Otherwise, $\mathcal{AF}\mathcal{O}_\Gamma(P) \vdash_{ws} \rho(t) \xrightarrow{\tau} \rightarrow$.

Likewise, for any $\omega \in O$, $\mathcal{AF}\mathcal{O}_\Gamma(P) \vdash_{ws} \rho(t) \xrightarrow{\omega}$ if t has a Γ -liquid argument x for which $\rho(x) \xrightarrow{\omega} \rightarrow_H$. Otherwise, $\mathcal{AF}\mathcal{O}_\Gamma(P) \vdash_{ws} \rho(t) \xrightarrow{\omega} \rightarrow$.

A closed negative literal $p \xrightarrow{a} \rightarrow$ is called *true* if $\mathcal{AF}\mathcal{O}(P) \vdash_{ws} p \xrightarrow{a} \rightarrow$, *false* if $\mathcal{AF}\mathcal{O}(P) \vdash_{ws} p \xrightarrow{a} q$, and *ambiguous* if neither applies. Above we proved that there are no ambiguous literals labelled $a \in A$ or τ or $\omega \in O$. It remains to show that there are no ambiguous literals labelled ι . Towards a contradiction, assume that $p \xrightarrow{l} \rightarrow$ is ambiguous.

There must exist a closed term q and set of closed negative literals N such that $\mathcal{AF}\mathcal{O}(P) \vdash_{irr} \frac{N}{p \xrightarrow{l} q}$ and no literal in N is false. For if there were no such q and N , the literal $p \xrightarrow{l} \rightarrow$ would be true by Def. 11. Moreover, N must contain an ambiguous literal $r \xrightarrow{l} \rightarrow$, for if all literals in N would be true, then the literal $p \xrightarrow{l} \rightarrow$ would be false by Def. 11. By Lem. 44 there is a set of closed negative literals N' such that $\mathcal{G}(P) \vdash_{irr} \frac{N'}{p \xrightarrow{l} q}$ and $N = \{s \xrightarrow{a} \mid (s \xrightarrow{a} \rightarrow) \in N'\} \cup \{t \xrightarrow{\tau} \rightarrow, t \xrightarrow{l} \rightarrow \mid (t \xrightarrow{\tau} \rightarrow) \in N'\}$. So N contains both $r \xrightarrow{l} \rightarrow$ and $r \xrightarrow{\tau} \rightarrow$. If $\mathcal{G}(P) \vdash_{ws} r \xrightarrow{\tau} \rightarrow$ then $r \xrightarrow{l} \rightarrow$ would be true by Prop. 45. Hence $\mathcal{G}(P) \vdash_{ws} r \xrightarrow{\tau} \rightarrow$, by the completeness of $\mathcal{G}(P)$. So by Prop. 45, $\mathcal{AF}\mathcal{O}(P) \vdash_{ws} r \xrightarrow{\tau} \rightarrow$ or $\mathcal{AF}\mathcal{O}(P) \vdash_{ws} r \xrightarrow{l} \rightarrow$. It follows that one of the literals $r \xrightarrow{l} \rightarrow$ or $r \xrightarrow{\tau} \rightarrow$ must be false, contradicting the absence of false literals in N . \square

4.2.4 Comparison of $\mathcal{AF}\mathcal{O}(P)$ and K

The LTS $K = (\mathbb{P}_K, A_\tau \cup O \cup \{\iota\}, \rightarrow_K)$ has as states $\mathbb{P}_K = \{\text{dec}(p) \mid p \in T(\hat{\Sigma})\}$, and the transitions are the ones generated by the following two rules:

$$\frac{x \xrightarrow{\alpha} y}{\text{dec}(x) \xrightarrow{\alpha} \text{dec}(y)} \quad \frac{x \xrightarrow{\iota} y}{\text{dec}(x) \xrightarrow{\tau} \text{dec}(y)}$$

where α ranges over A_τ . The operator $\text{dec} : T(\hat{\Sigma}) \rightarrow \mathbb{P}_K$ simply sends any process $p \in T(\hat{\Sigma})$ to the state $\text{dec}(p) \in \mathbb{P}_K$. Thus, dec erases all transitions with labels from O and renames labels ι into τ . All other transitions are preserved.

Proposition 47 Let $P = (\Sigma, A_\tau, R)$ be a standard TSS. Let a range over A and p, q over $T(\hat{\Sigma})$.

- $\text{dec}(p) \xrightarrow{a}_K q^* \Leftrightarrow \exists q \in T(\hat{\Sigma}). q^* = \text{dec}(q) \wedge \mathcal{AF}\mathcal{O}(P) \vdash_{ws} p \xrightarrow{a} q.$
- $\text{dec}(p) \xrightarrow{\tau}_K q^* \Leftrightarrow \exists q \in T(\hat{\Sigma}). q^* = \text{dec}(q) \wedge \left(\mathcal{AF}\mathcal{O}(P) \vdash_{ws} p \xrightarrow{\tau} q \vee \mathcal{AF}\mathcal{O}(P) \vdash_{ws} p \xrightarrow{\iota} q \right).$

Proof: Straightforward. □

4.2.5 Verifying requirements 1 and 6 of Thm. 33

That requirement 1 of Thm. 33 is met is immediate by Cor. 46.

We end this section by verifying requirement 6 of Thm. 33. Intuitively, the behaviour of a process $f(p_1, \dots, p_n)$ in P is the same as that of the process $f(\hat{p}_1, \dots, \hat{p}_n)$ in $\mathcal{AF}\mathcal{O}(P)$, except that some τ -transitions of the former are turned into ι -transitions of the latter process, and that some oracle transitions may have been added in the latter. Since any rule in $\mathcal{AF}\mathcal{O}(P)$ with a conclusion labelled by $A_\tau \cup \{\iota\}$ has positive and negative premises with labels from $A_\tau \cup \{\iota\}$ only, these oracle transitions have no influence on the derivation of any transitions from $\mathcal{AF}\mathcal{O}(P)$ with labels in $A_\tau \cup \{\iota\}$. The operator dec removes all oracle transitions and renames ι into τ , thereby returning the behaviour of $f(\hat{p}_1, \dots, \hat{p}_n)$ to match that of $f(p_1, \dots, p_n)$ exactly.

Proposition 48 Let P be a complete standard TSS in decent ntyft format.

Then $f(p_1, \dots, p_n) \Leftrightarrow_{P \oplus K} \text{dec}(f(\hat{p}_1, \dots, \hat{p}_n))$ for any n -ary $f \in \Sigma$ and $p_1, \dots, p_n \in T(\Sigma)$.

Proof: It suffices to show that the relation $\{\check{p}, \text{dec}(p) \mid p \in T(\hat{\Sigma})\}$ is a strong bisimulation.

So let $p \in T(\hat{\Sigma})$. Suppose that $\text{dec}(p) \xrightarrow{a}_K q^*$, with $a \in A$. Then, by Prop. 47, there is a $q \in T(\hat{\Sigma})$ with $q^* = \text{dec}(q)$ and $\mathcal{AF}\mathcal{O}(P) \vdash_{ws} p \xrightarrow{a} q$. Hence, by Prop. 45, $\mathcal{G}(P) \vdash_{ws} p \xrightarrow{a} q$. So, by Prop. 42, $P \vdash_{ws} \check{p} \xrightarrow{a} \check{q}$, which had to be shown.

The case that $\text{dec}(p) \xrightarrow{\tau}_K q^*$ proceeds in the same way.

Now suppose that $P \vdash_{ws} \check{p} \xrightarrow{a} q^*$, with $a \in A$. Then, by Prop. 42, there is a $q \in T(\hat{\Sigma})$ with $q^* = \check{q}$ and $\mathcal{G}(P) \vdash_{ws} p \xrightarrow{a} q$. Hence, by Prop. 45, $\mathcal{AF}\mathcal{O}(P) \vdash_{ws} p \xrightarrow{a} q$. So, by Prop. 47, $\text{dec}(p) \xrightarrow{a}_K \text{dec}(q)$, which had to be shown. Again the case $P \vdash_{ws} \check{p} \xrightarrow{\tau} q^*$ proceeds in the same way. □

4.3 Application of the general framework to divergence-preserving semantics

We now apply Thm. 33 to derive that the stability-respecting branching bisimulation format and its rooted variant are congruence formats for $\Leftrightarrow_b^{\Delta^\top}$ and \Leftrightarrow_b^Δ , and $\Leftrightarrow_{rb}^{\Delta^\top}$ and $\Leftrightarrow_{rb}^\Delta$, respectively.

As congruence format \mathfrak{F} in Thm. 33 we take the (rooted) stability-respecting branching bisimulation format intersected with the decent ntyft format. If a TSS P is in this format, there exist predicates \mathfrak{X}

and Λ such that P is $\aleph \cap \Lambda$ -patient and only contains rules that are rooted stability-respecting branching bisimulation safe w.r.t. \aleph and Λ . Although there may be some freedom in the choice of \aleph and Λ , the predicate $\aleph \cap \Lambda$ is completely determined by P . Namely, each $\aleph \cap \Lambda$ -liquid argument of an operator symbol f must have a patience rule, and conversely, each argument of f that has a patience rule must be $\aleph \cap \Lambda$ -liquid, by conditions 1 and 3 of Def. 19. Hence there can be no ambiguity in the choice of Γ in $\mathcal{AFO}_\Gamma(P)$.

It is straightforward to check that the conversion \mathcal{AFO} on standard TSSs defined in Sect. 4.2 preserves the (rooted) stability-respecting branching bisimulation format, and thus satisfies requirement 2 of Thm. 33. Here it is important that in step 6 of Def. 34 a term $f(x_1, \dots, x_n)$ inherits oracle transitions only from its Γ -liquid arguments—else condition 3 of Def. 19 would be violated. Furthermore, condition 4a of Def. 19 is preserved because premises $v \xrightarrow{\tau}$ are kept in place in step 3 of Def. 34; and condition 4b of Def. 19 is preserved because the transformation in Def. 34 does not introduce new positive premises with the label τ .

4.3.1 A congruence format for weakly divergence-preserving branching bisimilarity

We first apply Thm. 33 with $\leftrightarrow_b^{\Delta\top}$ and \leftrightarrow_b^s in the roles of \sim and \approx . In the construction of the LTS H out of the LTS G (cf. Sect. 4.2) we take $O = \{\Delta\top\}$ and let $\zeta(p) = \Delta\top$ iff p is divergent. As observed in Ex. 36, $\hat{p} \sim_H \hat{q}$ iff $\hat{p} \sim_G \hat{q}$. Hence, with Cor. 39, requirement 3 of Thm. 33 is satisfied. We proceed to show that also requirement 4 is satisfied.

Lemma 49 Let $t \in \mathbb{T}(\Sigma)$ and $\rho : \text{var}(t) \rightarrow \mathbb{P}_H$. In $\mathcal{AFO}_\Gamma(P)$, $\rho(t)$ is divergent iff t has a Γ -liquid occurrence of a variable x such that $\rho(x)$ is divergent.

Proof: For “only if”, suppose $\rho(t)$ is divergent, i.e., there is an infinite sequence of τ -transitions from $\rho(t)$. Since $\mathcal{AFO}_\Gamma(P)$ is abstraction-free, each of these transitions must originate from a τ -transition from a process $\rho(x)$, where x occurs Γ -liquid in t . One of the variables x that occurs Γ -liquid in t must contribute infinitely many of these transitions, so that $\rho(x)$ is divergent.

“If” follows immediately from the fact that the TSS $\mathcal{AFO}_\Gamma(P)$ is Γ -patient. \square

Lemma 50 Let $p \in \mathbb{T}(\hat{\Sigma})$. Then $\mathcal{AFO}_\Gamma(P) \vdash_{ws} p \xrightarrow{\Delta\top}$ iff p is divergent.

Proof: In $\mathcal{AFO}_\Gamma(P)$ any term $p \in \mathbb{T}(\hat{\Sigma})$ can be written as $\rho(t)$ with $t \in \mathbb{T}(\Sigma)$ and $\rho : \text{var}(t) \rightarrow \mathbb{P}_H$.

Suppose $\rho(t)$ is divergent. Then t has a Γ -liquid occurrence of a variable x such that $\rho(x)$ is divergent, by Lem 49. Hence $\rho(x) \xrightarrow{\Delta\top}_H$ by the construction of H . Thus $\mathcal{AFO}_\Gamma(P) \vdash_{ws} \rho(t) \xrightarrow{\Delta\top}$, by Obs. 40. The other direction proceeds likewise. \square

The following proposition states that requirement 4 of Thm. 33 is satisfied indeed.

Proposition 51 On $\mathcal{AFO}(P)$ the equivalences $\leftrightarrow_b^{\Delta\top}$ and \leftrightarrow_b^s coincide.

Proof: It suffices to show that the relation \leftrightarrow_b^s on $\mathcal{AFO}(P)$ is a weakly divergence-preserving branching bisimulation. By definition it is a (stability-respecting) branching bisimulation. Hence it remains to show that it is weakly divergence-preserving. To this end, it suffices to show that if p is divergent and $p \leftrightarrow_b^s q$ then also q is divergent.

Suppose $p(t)$ is divergent. Then by Lem. 50 $\mathcal{AFO}(P) \vdash_{ws} p \xrightarrow{\Delta\top}$. So $\mathcal{AFO}(P) \vdash_{ws} q \xRightarrow{\varepsilon} q' \xrightarrow{\Delta\top}$ for some q' , by the definition of a branching bisimulation. So q' is divergent, by Lem. 50, and thus q is divergent. \square

The following example shows that Prop. 51 would not hold if we had skipped step 6 of Def. 34, inheriting oracle transitions for Γ -liquid arguments, or if we had not used the oracle transitions at all.

Example 52 Let $p \in T(\Sigma)$ be deadlock (a process without outgoing transitions) and $q \in T(\Sigma)$ a process having only a τ -transition to itself, and a τ -transition to p . Then in the LTS G we have $\hat{p} \leftrightarrow_b^s \hat{q}$ but $\hat{p} \not\leftrightarrow_b^{\Delta\top} \hat{q}$. After translation to H , the processes \hat{p} and \hat{q} are distinguished by means of oracle transitions, so that we have $\hat{p} \not\leftrightarrow_b^s \hat{q}$ and $\hat{p} \not\leftrightarrow_b^{\Delta\top} \hat{q}$. Now let Σ feature a unary operator f with as only rule $\frac{x \xrightarrow{\tau} y}{f(x) \xrightarrow{\tau} f(y)}$. If oracle transitions would not be inherited in $\mathcal{AFO}(P)$, then we would have $f(\hat{p}) \leftrightarrow_b^s f(\hat{q})$ but $f(\hat{p}) \not\leftrightarrow_b^{\Delta\top} f(\hat{q})$.

We now verify requirement 5 of Thm. 33.

Proposition 53 $p \leftrightarrow_b^{\Delta\top} q \Rightarrow dec(p) \leftrightarrow_b^K dec(q)$.

Proof: Define the relation \mathcal{B} on the states of K by

$$dec(p) \mathcal{B} dec(q) \Leftrightarrow p \leftrightarrow_b^{\Delta\top} q.$$

It suffices to show that \mathcal{B} is a weakly divergence-preserving branching bisimulation.

- Suppose $p \leftrightarrow_b^{\Delta\top} q$ and $dec(p) \xrightarrow{\alpha} p^\dagger$. By the semantics of dec there are two possibilities.

CASE 1: $p \xrightarrow{\alpha} p'$ for some p' with $p^\dagger = dec(p')$. Since $p \leftrightarrow_b^{\Delta\top} q$, either $\alpha = \tau$ and $p' \leftrightarrow_b^{\Delta\top} q$, or $q \xRightarrow{\varepsilon} q' \xrightarrow{\alpha} q''$ for some q' and q'' with $p \leftrightarrow_b^{\Delta\top} q'$ and $p' \leftrightarrow_b^{\Delta\top} q''$. So either $\alpha = \tau$ and $dec(p') \mathcal{B} dec(q)$, or $dec(q) \xRightarrow{\varepsilon} dec(q') \xrightarrow{\alpha} dec(q'')$ with $dec(p) \mathcal{B} dec(q')$ and $dec(p') \mathcal{B} dec(q'')$.

CASE 2: $\alpha = \tau$, and $p \xrightarrow{l} p'$ for some p' with $p^\dagger = dec(p')$. Since $p \leftrightarrow_b^{\Delta\top} q$, $q \xRightarrow{\varepsilon} q' \xrightarrow{l} q''$ for some q' and q'' with $p \leftrightarrow_b^{\Delta\top} q'$ and $p' \leftrightarrow_b^{\Delta\top} q''$. So $dec(q) \xRightarrow{\varepsilon} dec(q') \xrightarrow{\tau} dec(q'')$ with $dec(p) \mathcal{B} dec(q')$ and $dec(p') \mathcal{B} dec(q'')$.
- Suppose $p \leftrightarrow_b^{\Delta\top} q$ and there is an infinite sequence $(p_k^\dagger)_{k \in \mathbb{N}}$ such that $dec(p) = p_0^\dagger$, $p_k^\dagger \xrightarrow{\tau} p_{k+1}^\dagger$ and $p_k^\dagger \mathcal{B} dec(q)$ for all $k \in \mathbb{N}$. Then there is an infinite sequence $(p_k)_{k \in \mathbb{N}}$ such that $p_0 = p$ and, for all $k \in \mathbb{N}$, $p_k^\dagger = dec(p_k)$, $p_k \leftrightarrow_b^{\Delta\top} q$ and either $p_k \xrightarrow{\tau} p_{k+1}$ or $p_k \xrightarrow{l} p_{k+1}$. We distinguish two cases.

CASE 1: For infinitely many of the k we have $p_k \xrightarrow{l} p_{k+1}$. Then there is an infinite sequence $(p'_j)_{j \in \mathbb{N}}$ such that $p'_0 = p_0$ and $p'_j \xRightarrow{\varepsilon} p'_{j+1}$ for all $j \in \mathbb{N}$. Since $p \leftrightarrow_b^{\Delta\top} q$, there must be an infinite sequence $(q'_j)_{j \in \mathbb{N}}$ such that $q = q'_0$, $q'_j \xRightarrow{\varepsilon} q'_{j+1}$ and $p'_j \leftrightarrow_b^{\Delta\top} q'_j$ for all $j \in \mathbb{N}$. It follows that $dec(q) = dec(q'_0)$ and $dec(p'_j) \xRightarrow{\varepsilon} dec(p'_{j+1})$ for all $j \in \mathbb{N}$. In other words, there exists an infinite sequence $(q_\ell^\dagger)_{\ell \in \mathbb{N}}$ such that $dec(q) = q_0^\dagger$ and $q_\ell^\dagger \xrightarrow{\tau} q_{\ell+1}^\dagger$ for all $\ell \in \mathbb{N}$.

CASE 2: There is an $n \in \mathbb{N}$ such that $p_k \xrightarrow{\tau} p_{k+1}$ for all $k \geq n$. Since $p \leftrightarrow_b^{\Delta\top} q$, there must be a finite sequence $(q_k)_{k=0}^n$ such that $q = q_0$ and, for all $0 \leq k < n$, either $q_k \xRightarrow{\varepsilon} q_{k+1}$ or $q_k \xRightarrow{\varepsilon} p_{k+1}$, and $p_{k+1} \leftrightarrow_b^{\Delta\top} q_{k+1}$. Moreover, since $p_n \leftrightarrow_b^{\Delta\top} q_n$, and $p_k \leftrightarrow_b^{\Delta\top} q \leftrightarrow_b^{\Delta\top} p_n \leftrightarrow_b^{\Delta\top} q_n$ for each $k \geq n$, there must be an infinite sequence $(q_\ell)_{\ell > n}$ such that $q_\ell \xrightarrow{\tau} q_{\ell+1}$ for all $\ell \geq n$. It follows that $dec(q) = dec(q_0) \xRightarrow{\varepsilon} dec(q_n)$ and $dec(q_\ell) \xrightarrow{\tau} dec(q_{\ell+1})$ for all $\ell \geq n$. \square

Corollary 54 \mathfrak{F} is a congruence format for $\leftrightarrow_b^{\Delta\top}$.

As indicated in Sect. 2.5, each standard TSS P in ready simulation format can be converted to a TSS P' in decent ntyft format. In [5] it is shown that this transformation preserves the set of *ws*-provable closed literals, hence \sim is a congruence for P iff P' is. It is not hard to check that if P is in (rooted) stability-respecting branching bisimulation format then so is P' . Thus we obtain the following congruence theorem.

Theorem 55 Let P be a complete standard TSS in stability-respecting branching bisimulation format. Then $\leftrightarrow_b^{\Delta^\top}$ is a congruence for P . \square

4.3.2 A congruence format for rooted weakly divergence-preserving branching bisimilarity

We now apply Thm. 33 with $\leftrightarrow_{rb}^{\Delta^\top}$ and \leftrightarrow_{rb}^s in the roles of \sim and \approx . The choice of O and ζ in the construction of H is the same as above.

Again requirement 3 of Thm. 33 is satisfied: since any two processes $g, h \in \mathbb{P}_G$ are $\leftrightarrow_b^{\Delta^\top}$ -equivalent in G iff they are $\leftrightarrow_b^{\Delta^\top}$ -equivalent in H , it follows immediately from Def. 2 that any two processes $g, h \in \mathbb{P}_G$ are $\leftrightarrow_{rb}^{\Delta^\top}$ -equivalent in G iff they are $\leftrightarrow_{rb}^{\Delta^\top}$ -equivalent in H .

Requirement 4 of Thm. 33 is also satisfied: since on $\mathcal{AF}\mathcal{O}(P)$ the equivalences $\leftrightarrow_b^{\Delta^\top}$ and \leftrightarrow_b^s coincide, it follows immediately from Def. 2 that also $\leftrightarrow_{rb}^{\Delta^\top}$ and \leftrightarrow_{rb}^s coincide.

To check requirement 5 of Thm. 33 define the relation \mathcal{R} on the states of K by

$$\text{dec}(p) \mathcal{R} \text{dec}(q) \iff p \leftrightarrow_{rb}^{\Delta^\top} q.$$

With Def. 2 it is straightforward to check that \mathcal{R} is a rooted weakly divergence-preserving branching bisimulation. Hence requirement 5 is satisfied.

Corollary 56 Rooted \mathfrak{F} is a congruence format for $\leftrightarrow_{rb}^{\Delta^\top}$.

Exactly as above, this yields the following congruence theorem.

Theorem 57 Let P be a complete standard TSS in rooted stability-respecting branching bisimulation format. Then $\leftrightarrow_{rb}^{\Delta^\top}$ is a congruence for P . \square

4.3.3 A congruence format for divergence-preserving branching bisimilarity

Next, we apply Thm. 33 with \leftrightarrow_b^Δ and \leftrightarrow_b^s in the roles of \sim and \approx . In the construction of the LTS H , we let O contain a unique name for each \leftrightarrow_b^Δ -equivalence class of processes in G , and let $\zeta(p)$ be the name of the \leftrightarrow_b^Δ -equivalence class of $\hat{p} \in \mathbb{P}_G$, for any $p \in T(\Sigma)$. Thus in H all states of G have a fresh outgoing transition, labelled with the name of its \sim -equivalence class in G .

With this definition of H , requirement 3 of Thm. 33 is satisfied: any divergence-preserving branching bisimulation \mathcal{B} on G relates states in the same \leftrightarrow_b^Δ -equivalence class only, and thus is also a divergence-preserving branching bisimulation on H (when adding $\sqrt{\mathcal{B}} \sqrt{\mathcal{B}}$).

We proceed to show that also requirement 4 is satisfied. A few lemmas are needed.

Lemma 58 [16] Condition (D) in Def. 1 can be replaced by the following equivalent condition:

(D') if $p \mathcal{B} q$ and there is an infinite sequence of processes $(p_k)_{k \in \mathbb{N}}$ such that $p = p_0$, $p_k \xrightarrow{\tau} p_{k+1}$ and $p_k \mathcal{B} q$ for all $k \in \mathbb{N}$, then there is a process q' such that $q \xrightarrow{\varepsilon} q'$ and $p_k \mathcal{B} q'$ for some $k \in \mathbb{N}$.

That is, the resulting definition also yields the relation \leftrightarrow_b^Δ .

Furthermore, we will employ the following property of \leftrightarrow_b^Δ .

Lemma 59 [16] If $p_0 \xrightarrow{\tau} p_1 \xrightarrow{\tau} \dots \xrightarrow{\tau} p_n$ and $p_0 \leftrightarrow_b^\Delta p_n$, then $p_0 \leftrightarrow_b^\Delta p_i$ for all $i = 0, \dots, n$.

The following proposition tells that requirement 4 of Thm. 33 is satisfied indeed.

Proposition 60 On $\mathcal{AF}\mathcal{O}(P)$ the equivalences \leftrightarrow_b^Δ and \leftrightarrow_b^s coincide.

Proof: It suffices to show that the relation \leftrightarrow_b^s on $\mathcal{AF}\mathcal{O}(P)$ is a divergence-preserving branching bisimulation. By definition it is a (stability-respecting) branching bisimulation. Hence it remains to show that it is divergence-preserving. By Lem. 58 it suffices to show that it satisfies condition (D'). So assume that $p \leftrightarrow_b^s q$ and there is an infinite sequence of processes $(p_k)_{k \in \mathbb{N}}$ such that $p = p_0$, $p_k \xrightarrow{\tau} p_{k+1}$ and $p_k \leftrightarrow_b^s q$ for all $k \in \mathbb{N}$. We need to find a process q' such that $q \xRightarrow{\varepsilon} \xrightarrow{\tau} q'$ and $p_k \leftrightarrow_b^s q'$ for some $k \in \mathbb{N}$. Towards a contradiction, assume that there is no q' with $q \xRightarrow{\varepsilon} \xrightarrow{\tau} q'$ and $p_k \leftrightarrow_b^s q'$ for some $k \in \mathbb{N}$.

Let $p = \rho(t)$ and $q = \rho(u)$ for univariate $t, u \in \mathbb{T}(\Sigma)$ with $\text{var}(t) \cap \text{var}(u) = \emptyset$ and $\rho: \text{var}(t) \cup \text{var}(u) \rightarrow \mathbb{P}_H$. Since $\mathcal{AF}\mathcal{O}(P)$ is abstraction-free, all τ -transitions in the infinite sequence $(p_k \xrightarrow{\tau} p_{k+1})_{k \in \mathbb{N}}$ originate, through patience rules, from τ -transitions of the processes $\rho(x)$ for variables x occurring Γ -liquid in t . Let L be the set of variables x that occur Γ -liquid in t . Then $\rho(x) = h_0^x \xrightarrow{\tau} h_1^x \xrightarrow{\tau} h_2^x \xrightarrow{\tau} \dots$ for each $x \in L$, and for at least one $x \in L$ —call it z —this sequence is infinitely long. For each $k \in \mathbb{N}$ and $x \in L$, let $k_x \leq k$ keep track of how many of the τ -transitions in the sequence $p_0 \xrightarrow{\tau} \dots \xrightarrow{\tau} p_k$ originate from $\rho(x)$. Hence $p_k = \rho_k(t)$ where $\rho_k(x) = h_{k_x}^x$ for each $x \in L$ and $\rho_k(x) = \rho(x)$ for each $x \in \text{var}(t) \setminus L$. Since, by assumption, the sequence of τ -transitions originating from $\rho(z)$ is infinite, for each $\ell \in \mathbb{N}$ there exists a $k \in \mathbb{N}$ such that $k_z = \ell$.

Claim: There is a $j \in \mathbb{N}$ such that $h_n^z \leftrightarrow_b^\Delta h_{j_z}^z$ for all $n \geq j_z$.

Proof of claim: Suppose there is no such j . Then, using Lem. 59, the processes h_ℓ^z for $\ell \in \mathbb{N}$ belong to infinitely many \leftrightarrow_b^Δ -equivalence classes. So there are infinitely many actions $\omega \in O$ such that $\exists \ell. h_\ell^z \xrightarrow{\omega} h$. For each of those actions ω , $\mathcal{AF}\mathcal{O}(P) \vdash_{ws} p_k \xrightarrow{\omega}$ for some $k \in \mathbb{N}$, by Obs. 40, and hence $\mathcal{AF}\mathcal{O}(P) \vdash_{ws} q \xRightarrow{\varepsilon} q'' \xrightarrow{\omega}$ for some q'' with $p_k \leftrightarrow_b^s q''$, since $p_k \leftrightarrow_b^s q$. As we have assumed that there is no q' with $q \xRightarrow{\varepsilon} \xrightarrow{\tau} q'$ and $p_k \leftrightarrow_b^s q'$ for some $k \in \mathbb{N}$, we have $q'' = q$, and thus $\mathcal{AF}\mathcal{O}(P) \vdash_{ws} q \xrightarrow{\omega}$. However, according to Obs. 40 there can only be finitely many actions $\omega \in O$ with $\mathcal{AF}\mathcal{O}(P) \vdash_{ws} q \xrightarrow{\omega}$, namely one for each variable occurring Γ -liquid in u . ■

Let ω be the name of the \leftrightarrow_b^Δ -equivalence class of the process $\rho_j(z) = h_{j_z}^z$. Then $\rho_j(z) \xrightarrow{\omega} h$, and consequently $\mathcal{AF}\mathcal{O}(P) \vdash_{ws} \rho_j(t) \xrightarrow{\omega}$ by Obs. 40. Since $\rho_j(t) \leftrightarrow_b^s q$, it follows that $\mathcal{AF}\mathcal{O}(P) \vdash_{ws} q \xRightarrow{\varepsilon} q'' \xrightarrow{\omega}$ for some process q'' with $\rho_j(t) \leftrightarrow_b^s q''$. As we assumed that there is no q' with $q \xRightarrow{\varepsilon} \xrightarrow{\tau} q'$ and $p_k \leftrightarrow_b^s q'$ for some $k \in \mathbb{N}$, we have $q'' = q$, and thus $\mathcal{AF}\mathcal{O}(P) \vdash_{ws} q = \rho(u) \xrightarrow{\omega}$. So by Obs. 40 u has a Γ -liquid occurrence of a variable y with $\rho(y) \xrightarrow{\omega} h$. It follows that in G we have $h_{j_z}^z = \rho_j(z) \leftrightarrow_b^\Delta \rho(y)$. Thus, by Lem. 58, there exists a $g \in \mathbb{P}_G$ with $\rho(y) \xRightarrow{\varepsilon} \xrightarrow{\tau} g$ and $h_\ell^z \leftrightarrow_b^\Delta g$ for some $\ell \geq j_z$. Since the TSS $\mathcal{AF}\mathcal{O}(P)$ is Γ -patient, we have $q = \rho(u) \xRightarrow{\varepsilon} \xrightarrow{\tau} \rho'(u)$ for a substitution ρ' with $\rho'(y) = g$ and $\rho'(x) = \rho(x)$ for all variables $x \neq y$. As, using the claim, $\rho(y) \leftrightarrow_b^s \rho_j(z) = h_{j_z}^z \leftrightarrow_b^\Delta h_\ell^z \leftrightarrow_b^s g = \rho'(y)$, and \leftrightarrow_b^s is a congruence on $\mathcal{AF}\mathcal{O}(P)$, it follows that $\rho(u) \leftrightarrow_b^s \rho'(u)$. Since $p \leftrightarrow_b^s q = \rho(u) \leftrightarrow_b^s \rho'(u)$, it suffices to take $q' := \rho'(u)$, so that $q \xRightarrow{\varepsilon} \xrightarrow{\tau} q'$ and $p \leftrightarrow_b^s q'$. This contradicts our assumption that no such q' exists. □

We now verify requirement 5 of Thm. 33.

Proposition 61 $p \leftrightarrow_b^\Delta \mathcal{AF}\mathcal{O}(P) q \Rightarrow \text{dec}(p) \leftrightarrow_b^K \text{dec}(q)$.

Proof: Define the relation \mathcal{B} on the states of K by

$$\text{dec}(p) \mathcal{B} \text{dec}(q) \Leftrightarrow p \leftrightarrow_b^\Delta q.$$

By Lem. 58 it suffices to show that \mathcal{B} is a branching bisimulation satisfying condition (D'). That it is a branching bisimulation follows exactly as in the proof of Prop. 53. To show that it satisfies (D'), suppose $p \leftrightarrow_b^\Delta q$ and there is an infinite sequence $(p_k^\dagger)_{k \in \mathbb{N}}$ such that $\text{dec}(p) = p_0^\dagger$, $p_k^\dagger \xrightarrow{\tau} p_{k+1}^\dagger$ and $p_k^\dagger \mathcal{B} \text{dec}(q)$ for all $k \in \mathbb{N}$. Then there is an infinite sequence $(p_k)_{k \in \mathbb{N}}$ such that $p_0 = p$ and, for all $k \in \mathbb{N}$, $p_k^\dagger = \text{dec}(p_k)$, $p_k \leftrightarrow_b^\Delta q$ and either $p_k \xrightarrow{\tau} p_{k+1}$ or $p_k \xrightarrow{!} p_{k+1}$. We distinguish two cases.

CASE 1: There is a $k \in \mathbb{N}$ with $p_k \xrightarrow{l} p_{k+1}$ —consider the first such k . Then $p \xRightarrow{\varepsilon} p_k \xrightarrow{l} p_{k+1}$. Since $p \xleftrightarrow{b}^{\Delta} q$, there is a q' with $q \xRightarrow{\varepsilon} \xrightarrow{l} q'$ and $p_{k+1} \xleftrightarrow{b}^{\Delta} q'$. Hence there is a $\text{dec}(q')$ with $\text{dec}(q) \xRightarrow{\varepsilon} \xrightarrow{\tau} \text{dec}(q')$ and $\text{dec}(p_{k+1}) \mathcal{B} \text{dec}(q')$.

CASE 2: There is no such k . Then, since $\xleftrightarrow{b}^{\Delta}$ satisfies (D'), there is a q' such that $q \xRightarrow{\varepsilon} \xrightarrow{\tau} q'$ and $p_k \xleftrightarrow{b}^{\Delta} q'$ for some $k \in \mathbb{N}$. It follows that there is a $\text{dec}(q')$ such that $\text{dec}(q) \xRightarrow{\varepsilon} \xrightarrow{\tau} \text{dec}(q')$ and $\text{dec}(p_{k+1}) \mathcal{B} \text{dec}(q')$. \square

Corollary 62 \mathfrak{F} is a congruence format for $\xleftrightarrow{b}^{\Delta}$.

Exactly as above, this yields the following congruence theorem.

Theorem 63 Let P be a complete standard TSS in stability-respecting branching bisimulation format. Then $\xleftrightarrow{b}^{\Delta}$ is a congruence for P . \square

4.3.4 A congruence format for rooted divergence-preserving branching bisimilarity

We now apply Thm. 33 with $\xleftrightarrow{rb}^{\Delta}$ and \xleftrightarrow{rb}^s in the roles of \sim and \approx . The choice of O and ζ in the construction of H is the same as in Sect. 4.3.3.

Again requirement 3 of Thm. 33 is satisfied: since any two processes $g, h \in \mathbb{P}_G$ are $\xleftrightarrow{b}^{\Delta}$ -equivalent in G iff they are $\xleftrightarrow{b}^{\Delta}$ -equivalent in H , it follows immediately from Def. 2 that any two processes $g, h \in \mathbb{P}_G$ are $\xleftrightarrow{rb}^{\Delta}$ -equivalent in G iff they are $\xleftrightarrow{rb}^{\Delta}$ -equivalent in H .

Requirement 4 of Thm. 33 is also satisfied: since on $\mathcal{AFC}(P)$ the equivalences $\xleftrightarrow{b}^{\Delta}$ and \xleftrightarrow{b}^s coincide, it follows immediately from Def. 2 that also $\xleftrightarrow{rb}^{\Delta}$ and \xleftrightarrow{rb}^s coincide.

To check requirement 5 of Thm. 33 define the relation \mathcal{R} on the states of K by

$$\text{dec}(p) \mathcal{R} \text{dec}(q) \Leftrightarrow p \xleftrightarrow{rb}^{\Delta} q.$$

With Def. 2 it is straightforward to check that \mathcal{R} is a rooted weakly divergence-preserving branching bisimulation. Hence requirement 5 is satisfied.

Corollary 64 Rooted \mathfrak{F} is a congruence format for $\xleftrightarrow{rb}^{\Delta}$.

Exactly as above, this yields the following congruence theorem.

Theorem 65 Let P be a complete standard TSS in rooted stability-respecting branching bisimulation format. Then $\xleftrightarrow{rb}^{\Delta}$ is a congruence for P . \square

5 Related work

Ulidowski [27, 28, 29] proposed congruence formats, inside GSOS [6], for weak semantics that take into account non-divergence, called *convergence* in [14]. In [27] he introduces the *ISOS* format, and shows that the weak convergent *refusal simulation* preorder is a precongruence for all TSSs in the ISOS format. The GSOS format—in our terminology the *decent nxyft* format—allows only decent ntyft rules with variables as the left-hand sides of premises. The ISOS format is contained in the intersection of the GSOS format and our stability-preserving branching bisimulation format. Its additional restriction is that no variable may occur multiple times as the left-hand side of a positive premise, or both as the left-hand side of a positive premise and in the conclusion of a rule.

In [28, 29] he employs *Ordered SOS* (OSOS) TSSs [28]. An OSOS TSS allows no negative premises, but includes priorities between rules: $r < r'$ means that r can only be applied if r' cannot. An OSOS specification can be seen as, or translated into, a GSOS specification with negative premises. Each rule r with exactly one higher-priority rule $r' > r$ is replaced by a number of rules, one for each (positive) premise of r' ; in the copy of r , this premise is negated. For a rule r with multiple higher-priority rules r' , this replacement is carried out for each such r' .

The ebo and bbo formats from [28] target unrooted convergent *delay* and branching bisimulation equivalence, respectively. The bbo format is more liberal than the ebo format, which in turn is more liberal than the ISOS format. The rebo and rbbo formats from [29] target rooted convergent delay and branching bisimulation equivalence, respectively. These rooted formats are more liberal than their unrooted counterparts, and the rbbo format is more liberal than the rebo format.

If patience rules are not allowed to have a lower priority than other rules, then the (r)bbo format, upon translation from OSOS to GSOS, can be seen as a subformat of our (rooted) stability-respecting branching bisimulation format. The basic idea is that in the rbbo format all arguments of so-called τ -preserving function symbols [29], which are the only ones allowed to occur in targets, are declared Λ -liquid; in the bbo format, all arguments of function symbols are declared Λ -liquid. Moreover, all arguments of function symbols that occur as the left-hand side of a positive premise are declared \aleph -liquid. Patience rules are in the (r)bbo format however, under strict conditions, allowed to be dominated by other rules, which in our setting gives rise to patience rules with negative premises. This is outside the realm of our rooted stability-respecting branching bisimulation format. On the other hand, the TSSs of the process algebra $\text{BPA}_{\varepsilon\delta\tau}$, the binary Kleene star and deadlock testing (see [10]), for which rooted convergent branching bisimulation equivalence is a congruence, are outside the rbbo format but within the rooted stability-respecting branching bisimulation format.

6 Conclusions

We showed how the method from [13] for deriving congruence formats through modal decomposition can be applied to weak semantics that are stability-respecting. We used (rooted and unrooted) stability-respecting branching bisimulation equivalence as a notable example. Moreover, we developed a general method for lifting congruence formats from a weak semantics to a finer semantics, and used it to show that congruence formats for stability-respecting branching bisimulation equivalence are also congruence formats for their divergence-preserving counterparts. This research provides a deeper insight into the link between modal logic and congruence formats, and strengthens the framework from [13] for the derivation of congruence formats for weak semantics.

Almost every weak semantics has stability-respecting and divergence-preserving variants. Such variants have been studied most widely in the literature for branching, η -, delay and weak bisimulation equivalence, but they have for instance also been considered for decorated trace semantics, such as *exhibited behaviour equivalence* [26], the *generalised failure preorder* [22], *refusal equivalence* [24], and *copy+refusal equivalence* [27]. We expect that the methods developed in this paper, combined with the methods from [13, 10], can serve as a cornerstone in the generation of congruence formats for such semantics. In particular, we conjecture that this will straightforwardly yield congruence formats for stability-respecting and divergence-preserving variants of η -, delay and weak bisimulation equivalence.

Admittedly, the whole story is quite technical and intricate. Partly this is because we build on a rich body of earlier work in the realm of structural operational semantics: the notions of well-supported proofs and complete TSSs from [15]; the ntyft/ntyxt format from [19, 7]; the transformation to ruloids, which

for the main part goes back to [9]; and the work on modal decomposition and congruence formats from [5]. In spite of these technicalities, we have arrived at a relatively simple framework for the derivation of congruence formats for weak semantics. Namely, for this one only needs to: (1) provide a modal characterisation of the weak semantics under consideration; (2) study the class of modal formulas that result from decomposing this modal characterisation, and formulate syntactic restrictions on TSSs to bring this class of modal formulas within the original modal characterisation; and (3) check that these syntactic restrictions are preserved under the transformation to ruloids. Steps (2) and (3) are very similar in structure for different weak semantics, as exemplified by the way we obtained a congruence format for stability-respecting branching bisimulation equivalence. And the resulting congruence formats tend to be more liberal and elegant than existing congruence formats in the literature.

Our intention is to carve out congruence formats for all weak semantics in the spectrum from [14] that have reasonable congruence properties. At first we expected that the current third instalment would allow us to do so. However, it turns out that convergent weak semantics as considered in for instance [28, 29, 31] still need extra work. The modal characterisations of these semantics are three-valued [14], which requires an extension of the modal decomposition technique to a three-valued setting.

References

- [1] J.C.M. BAETEN, J.A. BERGSTRA & J.W. KLOP (1986): *Syntax and defining equations for an interrupt mechanism in process algebra*. *Fundamenta Informaticae* 9(2), pp. 127–167.
- [2] T. BASTEN (1996): *Branching bisimulation is an equivalence indeed!* *Information Processing Letters* 58(3), pp. 141–147.
- [3] B. BLOOM (1994): *When is partial trace equivalence adequate?* *Formal Aspects of Computing* 6, pp. 317–338.
- [4] B. BLOOM (1995): *Structural operational semantics for weak bisimulations*. *Theoretical Computer Science* 146(1/2), pp. 25–68.
- [5] B. BLOOM, W.J. FOKKINK & R.J. VAN GLABBEK (2004): *Precongruence formats for decorated trace semantics*. *ACM Transactions on Computational Logic* 5(1), pp. 26–78.
- [6] B. BLOOM, S. ISTRAIL & A.R. MEYER (1995): *Bisimulation can't be traced*. *Journal of the ACM* 42(1), pp. 232–268.
- [7] R.N. BOL & J.F. GROOTE (1996): *The meaning of negative premises in transition system specifications*. *Journal of the ACM* 43(5), pp. 863–914.
- [8] W.J. FOKKINK (2000): *Rooted branching bisimulation as a congruence*. *Journal of Computer and System Sciences* 60(1), pp. 13–37.
- [9] W.J. FOKKINK & R.J. VAN GLABBEK (1996): *Ntyfi/ntyxt rules reduce to ntree rules*. *Information and Computation* 126(1), pp. 1–10.
- [10] W.J. FOKKINK & R.J. VAN GLABBEK (2016): *Divide and congruence II: Delay and weak bisimilarity*. In *Proc. LICS'16*, pp. 778–787, ACM/IEEE.
- [11] W.J. FOKKINK, R.J. VAN GLABBEK & B. LUTTIK (2017): *Divide and congruence III: Stability & divergence*. In *Proc. CONCUR'17, LIPIcs* 85, pp. 11:1–11:15, Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- [12] W.J. FOKKINK, R.J. VAN GLABBEK & P. DE WIND (2006): *Divide and congruence: From decomposition of modalities to preservation of branching bisimulation*. In *Proc. FMCO'05, LNCS* 4111, pp. 195–218, Springer.

- [13] W.J. FOKKINK, R.J. VAN GLABBEK & P. DE WIND (2012): *Divide and congruence: From decomposition of modal formulas to preservation of branching and η -bisimilarity*. *Information and Computation* 214, pp. 59–85.
- [14] R.J. VAN GLABBEK (1993): *The linear time-branching time spectrum II: The semantics of sequential systems with silent moves*. In *Proc. CONCUR'93*, LNCS 715, pp. 66–81. Springer.
- [15] R.J. VAN GLABBEK (2004): *The meaning of negative premises in transition system specifications II*. *Journal of Logic and Algebraic Programming* 60/61, pp. 229–258.
- [16] R.J. VAN GLABBEK, B. LUTTIK & N. TRČKA (2009): *Branching bisimilarity with explicit divergence*. *Fundamenta Informaticae*, 93(4) pp. 371–392.
- [17] R.J. VAN GLABBEK, B. LUTTIK & N. TRČKA (2009): *Computation tree logic with deadlock detection*. *Logical Methods in Computer Science* 5(4):5.
- [18] R.J. VAN GLABBEK & W.P. WEIJLAND (1996): *Branching time and abstraction in bisimulation semantics*. *Journal of the ACM* 43(3), pp. 555–600.
- [19] J.F. GROOTE (1993): *Transition system specifications with negative premises*. *Theoretical Computer Science* 118(2), pp. 263–299.
- [20] J.F. GROOTE & F.W. VAANDRAGER (1992): *Structured operational semantics and bisimulation as a congruence*. *Information and Computation* 100(2), pp. 202–260.
- [21] M. HENNESSY & R. MILNER (1985): *Algebraic laws for non-determinism and concurrency*. *Journal of the ACM* 32(1), pp. 137–161.
- [22] R. LANGERAK (1989): *A testing theory for LOTOS using deadlock detection*. In E. Brinksma, G. Scollo & C.A. Vissers, editors: *Proceedings 9th IFIP WG6.1 International Symposium on Protocol Specification, Testing, and Verification*, Enschede, The Netherlands.
- [23] R. MILNER (1989): *Communication and Concurrency*. Prentice-Hall.
- [24] I.C.C. PHILLIPS (1987): *Refusal testing*. *Theoretical Computer Science* 50, pp. 241–284.
- [25] G.D. PLOTKIN (2004): *A structural approach to operational semantics*. *Journal of Logic and Algebraic Programming* 60/61, pp. 17–139. Originally appeared in 1981.
- [26] L. POMELLO (1986): *Some equivalence notions for concurrent systems – An overview*. In G. Rozenberg, editor: *Advances in Petri Nets 1985*, LNCS 222, Springer, pp. 381–400.
- [27] I. ULIDOWSKI (1992): *Equivalences on observable processes*. In *Proc. LICS'92*, pp. 148–159. IEEE.
- [28] I. ULIDOWSKI & I. PHILLIPS (2002): *Ordered SOS rules and process languages for branching and eager bisimulations*. *Information and Computation* 178, pp. 180–213, 2002.
- [29] I. ULIDOWSKI & S. YUEN (2000): *Process languages for rooted eager bisimulation*. In *Proc. CONCUR'00*, LNCS 1877, pp. 275–289. Springer.
- [30] F.W. VAANDRAGER (1990): *Algebraic Techniques for Concurrency and their Application*. PhD thesis, University of Amsterdam.
- [31] D. WALKER (1990): *Bisimulation and divergence*. *Information and Computation* 85, pp. 202–241, 1990.

A Modal characterisations

We prove Thm. 8, which states that \mathbb{O}_b^s is a modal characterisation of $p \leftrightarrow_b^s q$, and \mathbb{O}_{rb}^s of $p \leftrightarrow_{rb}^s q$. So we need to prove, given an LTS $(\mathbb{P}, \rightarrow)$, that $p \leftrightarrow_b^s q \Leftrightarrow p \sim_{\mathbb{O}_b^s} q$ and $p \leftrightarrow_{rb}^s q \Leftrightarrow p \sim_{\mathbb{O}_{rb}^s} q$ for all $p, q \in \mathbb{P}$.

Proof: (\Rightarrow) We prove by structural induction on φ , resp. $\bar{\varphi}$, that $p \leftrightarrow_b^s q \wedge p \models \varphi \Rightarrow q \models \varphi$ for all $\varphi \in \mathbb{O}_b^s$, and $p \leftrightarrow_{rb}^s q \wedge p \models \bar{\varphi} \Rightarrow q \models \bar{\varphi}$ for all $\bar{\varphi} \in \mathbb{O}_{rb}^s$. The converse implications ($q \models \varphi \Rightarrow p \models \varphi$ and $\bar{q} \models \bar{\varphi} \Rightarrow p \models \bar{\varphi}$) follow by symmetry.

- $\varphi = \bigwedge_{i \in I} \varphi_i$. Then $p \models \varphi_i$ for all $i \in I$. By induction $q \models \varphi_i$ for all $i \in I$, so $q \models \bigwedge_{i \in I} \varphi_i$.
- $\varphi = \neg \varphi'$. Then $p \not\models \varphi'$. By induction $q \not\models \varphi'$, so $q \models \neg \varphi'$.
- $\varphi = \langle \varepsilon \rangle \varphi_1 \langle \hat{\tau} \rangle \varphi_2$. Then for some n there are $p_0, \dots, p_n \in \mathbb{P}$ with $p_0 = p$, $p_i \xrightarrow{\tau} p_{i+1}$ for $i \in \{0, \dots, n-1\}$, and $p_n \models \varphi_1 \langle \hat{\tau} \rangle \varphi_2$. We apply induction on n .
 - $n = 0$ Then $p \models \varphi_1$, so by induction on formula size, $q \models \varphi_1$. Furthermore, either (1) $p \models \varphi_2$ or (2) there is a $p' \in \mathbb{P}$ with $p \xrightarrow{\tau} p'$ and $p' \models \varphi_2$. In case (1), by induction on formula size, $q \models \varphi_2$, so $q \models \langle \varepsilon \rangle \varphi_1 \langle \hat{\tau} \rangle \varphi_2$. In case (2), since $p \leftrightarrow_b^s q$, by Def. 1 either (2.1) $p' \leftrightarrow_b^s q$ or (2.2) $q \xrightarrow{\varepsilon} q' \xrightarrow{\tau} q''$ with $p \leftrightarrow_b^s q'$ and $p' \leftrightarrow_b^s q''$. In case (2.1), by induction on formula size, $q \models \varphi_2$. In case (2.2), by induction on formula size, $q' \models \varphi_1$ and $q'' \models \varphi_2$. In both cases, $q \models \langle \varepsilon \rangle \varphi_1 \langle \hat{\tau} \rangle \varphi_2$.
 - $n > 0$ Since $p \xrightarrow{\tau} p_1$, and $p \leftrightarrow_b^s q$, according to Def. 1 there are two possibilities.
 1. Either $p_1 \leftrightarrow_b^s q$. Since $p_1 \models \langle \varepsilon \rangle \varphi_1 \langle \hat{\tau} \rangle \varphi_2$, by induction on n , $q \models \langle \varepsilon \rangle \varphi_1 \langle \hat{\tau} \rangle \varphi_2$.
 2. Or $q \xrightarrow{\varepsilon} q' \xrightarrow{\tau} q''$ with $p_1 \leftrightarrow_b^s q''$. Since $p_1 \models \langle \varepsilon \rangle \varphi_1 \langle \hat{\tau} \rangle \varphi_2$, by induction on n , $q'' \models \langle \varepsilon \rangle \varphi_1 \langle \hat{\tau} \rangle \varphi_2$. Hence $q \models \langle \varepsilon \rangle \varphi_1 \langle \hat{\tau} \rangle \varphi_2$.
- $\varphi = \langle \varepsilon \rangle \varphi_1 \langle a \rangle \varphi_2$. Then for some n there are $p_0, \dots, p_n \in \mathbb{P}$ with $p_0 = p$, $p_i \xrightarrow{\tau} p_{i+1}$ for $i \in \{0, \dots, n-1\}$, and $p_n \models \varphi_1 \langle a \rangle \varphi_2$. We apply induction on n .
 - $n = 0$ Then $p \models \varphi_1$, and there is a $p' \in \mathbb{P}$ with $p \xrightarrow{a} p'$ and $p' \models \varphi_2$. Since $p \leftrightarrow_b^s q$, by Def. 1 $q \xrightarrow{\varepsilon} q' \xrightarrow{a} q''$ with $p \leftrightarrow_b^s q'$ and $p' \leftrightarrow_b^s q''$. By induction on formula size, $q' \models \varphi_1$ and $q'' \models \varphi_2$. Hence $q \models \langle \varepsilon \rangle \varphi_1 \langle a \rangle \varphi_2$.
 - $n > 0$ This case goes exactly as the the case $n > 0$ above.
- $\varphi = \langle \varepsilon \rangle (\neg \langle \tau \rangle \top \wedge \bar{\varphi})$ with $\bar{\varphi} \in \mathbb{O}_{rb}^s$. Then for some n there are $p_0, \dots, p_n \in \mathbb{P}$ with $p_0 = p$, $p_i \xrightarrow{\tau} p_{i+1}$ for $i \in \{0, \dots, n-1\}$, and $p_n \models \neg \langle \tau \rangle \top \wedge \bar{\varphi}$. We apply induction on n .
 - $n = 0$ Then $p \models \bar{\varphi}$ and $p \not\xrightarrow{\tau}$. Since p and q are related by a *stability-respecting* branching bisimulation, there is a q' with $q \xrightarrow{\varepsilon} q' \not\xrightarrow{\tau}$ and $p \leftrightarrow_b^s q'$. Because p and q' are both stable, $p \leftrightarrow_b^s q'$ implies $p \leftrightarrow_{rb}^s q'$. So by induction on formula size, $q' \models \bar{\varphi}$. Hence $q \models \langle \varepsilon \rangle (\neg \langle \tau \rangle \top \wedge \bar{\varphi})$.
 - $n > 0$ This case goes exactly as the the case $n > 0$ above.
- $\bar{\varphi} = \bigwedge_{i \in I} \bar{\varphi}_i$. Then $p \models \bar{\varphi}_i$ for all $i \in I$. By induction $q \models \bar{\varphi}_i$ for all $i \in I$, so $q \models \bigwedge_{i \in I} \bar{\varphi}_i$.
- $\bar{\varphi} = \neg \bar{\varphi}'$. Then $p \not\models \bar{\varphi}'$. By induction $q \not\models \bar{\varphi}'$, so $q \models \neg \bar{\varphi}'$.
- $\bar{\varphi} = \langle \alpha \rangle \bar{\varphi}$ with $\bar{\varphi} \in \mathbb{O}_b^s$. Then $p \xrightarrow{\alpha} p'$ for some p' with $p \models \bar{\varphi}$. By Def. 2, $q \xrightarrow{\alpha} q'$ for some q' with $p' \leftrightarrow_b^s q'$. So by induction $q' \models \bar{\varphi}$, and hence $q \models \langle \alpha \rangle \bar{\varphi}$.
- $\bar{\varphi} = \varphi \in \mathbb{O}_b^s$. Since $p \leftrightarrow_{rb}^s q$ implies $p \leftrightarrow_b^s q$, we obtain $q \models \varphi$ by the cases treated above.

(\Leftarrow) We first prove that $\sim_{\mathbb{O}_b^s}$ is a branching bisimulation. The relation is clearly symmetric. Let $p \sim_{\mathbb{O}_b^s} q$. Suppose $p \xrightarrow{\alpha} p'$. If $\alpha = \tau$ and $p' \sim_{\mathbb{O}_b^s} q$, then the first condition of Def. 1 is fulfilled. So we can assume that either (i) $\alpha \neq \tau$ or (ii) $p' \not\sim_{\mathbb{O}_b^s} q$. We define two sets:

$$\begin{aligned} Q' &= \{q' \in \mathbb{P} \mid q \xRightarrow{\varepsilon} q' \wedge p \not\sim_{\mathbb{O}_b^s} q'\} \\ Q'' &= \{q'' \in \mathbb{P} \mid \exists q' \in \mathbb{P} : q \xRightarrow{\varepsilon} q' \xrightarrow{\alpha} q'' \wedge p' \not\sim_{\mathbb{O}_b^s} q''\} \end{aligned}$$

For each $q' \in Q'$, let $\varphi_{q'}$ be a formula in \mathbb{O}_b^s such that $p \models \varphi_{q'}$ and $q' \not\models \varphi_{q'}$. (Such a formula always exists because \mathbb{O}_b^s is closed under negation \neg .) We define

$$\varphi = \bigwedge_{q' \in Q'} \varphi_{q'}$$

Similarly, for each $q'' \in Q''$, let $\psi_{q''}$ be a formula in \mathbb{O}_b^s such that $p' \models \psi_{q''}$ and $q'' \not\models \psi_{q''}$. We define

$$\psi = \bigwedge_{q'' \in Q''} \psi_{q''}$$

Clearly, $\varphi, \psi \in \mathbb{O}_b^s$, $p \models \varphi$ and $p' \models \psi$. We distinguish two cases.

1. $\alpha \neq \tau$. Since $p \models \langle \varepsilon \rangle \varphi \langle \alpha \rangle \psi \in \mathbb{O}_b^s$ and $p \sim_{\mathbb{O}_b^s} q$, also $q \models \langle \varepsilon \rangle \varphi \langle \alpha \rangle \psi$. Hence $q \xRightarrow{\varepsilon} q' \xrightarrow{\alpha} q''$ with $q' \models \varphi$ and $q'' \models \psi$. By the definition of φ and ψ it follows that $p \sim_{\mathbb{O}_b^s} q'$ and $p' \sim_{\mathbb{O}_b^s} q''$.
2. $\alpha = \tau$ and $p' \not\sim_{\mathbb{O}_b^s} q$. Let $\tilde{\varphi} \in \mathbb{O}_b^s$ such that $p' \models \tilde{\varphi}$ and $p, q \not\models \tilde{\varphi}$. Since $p \models \langle \varepsilon \rangle \varphi \langle \hat{\tau} \rangle (\tilde{\varphi} \wedge \psi) \in \mathbb{O}_b^s$ and $p \sim_{\mathbb{O}_b^s} q$, also $q \models \langle \varepsilon \rangle \varphi \langle \hat{\tau} \rangle (\tilde{\varphi} \wedge \psi)$. So $q \xRightarrow{\varepsilon} q' \xrightarrow{\hat{\tau}} q''$ with $q' \models \varphi \langle \hat{\tau} \rangle (\tilde{\varphi} \wedge \psi)$. By definition of φ it follows that $p \sim_{\mathbb{O}_b^s} q'$. Thus $q' \not\models \tilde{\varphi}$, so $q' \xrightarrow{\tau} q''$ with $q'' \models \tilde{\varphi} \wedge \psi$. By the definition of ψ it follows that $p' \sim_{\mathbb{O}_b^s} q''$.

Both cases imply that the first condition of Def. 1 is fulfilled, i.e. that $\sim_{\mathbb{O}_b^s}$ is a branching bisimulation.

Next we show that $\sim_{\mathbb{O}_b^s}$ is stability-respecting. Let $p \sim_{\mathbb{O}_b^s} q$ and $p \xrightarrow{\tau} \cdot$. Define Q' and $\varphi \in \mathbb{O}_b^s \subseteq \mathbb{O}_{rb}^s$ as above. Then $p \models \langle \varepsilon \rangle (\neg \langle \tau \rangle \top \wedge \varphi)$, and thus also $q \models \langle \varepsilon \rangle (\neg \langle \tau \rangle \top \wedge \varphi)$. Hence $q \xRightarrow{\varepsilon} q'$ for some q' with $q' \models \neg \langle \tau \rangle \top \wedge \varphi$. So $q \not\xrightarrow{\tau} \cdot$ and by the definition of φ it follows that $p \sim_{\mathbb{O}_b^s} q'$, which had to be shown.

Finally we show that $\sim_{\mathbb{O}_{rb}^s}$ is a rooted stability-respecting branching bisimulation. Let $p \sim_{\mathbb{O}_{rb}^s} q$ and $p \xrightarrow{\alpha} p'$. Let $Q_\alpha = \{q' \in \mathbb{P} \mid q \xrightarrow{\alpha} q' \wedge p' \not\sim_{\mathbb{O}_b^s} q'\}$. For each $q' \in Q_\alpha$, let $\chi_{q'}$ be a formula in \mathbb{O}_b^s such that $p' \models \chi_{q'}$ and $q' \not\models \chi_{q'}$. We define $\chi = \bigwedge_{q' \in Q_\alpha} \chi_{q'}$. Since $p \models \langle \alpha \rangle \chi$ we have $q \models \langle \alpha \rangle \chi$, so $q \xrightarrow{\alpha} q'$ for some q' with $q' \models \chi$. By the definition of χ it follows that $p' \sim_{\mathbb{O}_b^s} q'$, which had to be shown. \square