# An alternative formulation of operational conservativity with binding terms

C.A. Middelburg

*Computing Science Department, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, Netherlands*

*Department of Philosophy, Utrecht University, P.O. Box 80126, 3508 TC Utrecht, Netherlands*

**Abstract**

In a previous paper, the approach to structural operational semantics using transition system specifications (TSSs) was extended to deal with variable binding operators. It was shown that in the new setting a generalization of the transition rule format known as the panth format guarantees that bisimulation equivalence is a congruence for meaningful TSSs. In the current paper, it is shown that certain syntactic criteria to determine whether a TSS is an operational conservative extension of another TSS, originating from Fokkink and Verhoef, are applicable to the new setting as well. This result can for example be used to simplify proofs of axiomatic conservativity and completeness in the case where an existing process calculus is extended with new features.

*Key words:* structural operational semantics, operational conservative extension, transition system specification, variable binding operator, source dependency

## 1 Introduction

Transition system specifications (TSSs) are used in an approach to structural operational semantics (SOS) that considers transition systems where the states are the closed terms over a given signature. The notion of TSS was first introduced in Ref. [1]. The original TSSs define binary transition relations by means of transition rules with positive premises. The notion of TSS was generalized in Refs. [2–5] to TSSs that define unary and binary transition relations by means of transition rules with positive and negative premises.

*Email address:* keesm@win.tue.nl (C.A. Middelburg).
*URL:* http://www.win.tue.nl/~keesm (C.A. Middelburg).

In Ref. [6], it was generalized further to cover variable binding operators. The new TSSs can amongst other things deal with: the integration operator $\int$ of real time ACP [7], the sum operator $\sum$ of $\mu$CRL [8], and the recursion operator $\mu$ of CSP [9] and CCS [10]. It was found that the notions of bisimulation equivalence and panth format generalize naturally to the new TSSs, and moreover that in the new setting bisimulation equivalence is still a congruence for meaningful TSSs in panth format.

The notion of TSS was first generalized to cover variable binding operators in Ref. [11]. In Ref. [6], an alternative extension was introduced that keeps the new TSSs more closely related to the original ones. In Ref. [11], no transition rule format is given that guarantees that bisimulation equivalence is a congruence. However, in that paper syntactic criteria are given to determine whether a TSS is an operational conservative extension of another TSS. In this paper, it is shown that those syntactic criteria are applicable to the new setting as well.

The explanation of the meaning of TSSs given in this paper differs from the one given in Ref. [6]. The new explanation uses less model-theoretic notions and more proof-theoretic notions. In that way, it conveys a more intuitive understanding of the meaning of TSSs.

It was recently found that the generalized panth format given in Ref. [6] could be made somewhat less restrictive. In this paper, the new relaxed version of the format is presented. In various applications of TSSs, it is impractical and unnecessary to provide the terms of certain sorts with an operational semantics because there exists a fully established semantics for them. The sort that represents the time domain in process calculi with timing, usually $\mathbb{N}$ or $\mathbb{R}_{\geq 0}$, is a typical example. By distinguishing such sorts, the generalized panth format can be relaxed further and transition relations can be parametrized. In Ref. [6], where such sorts are called given sorts, these matters have been discussed. In this paper, that discussion is adapted to the new explanation of the meaning of TSSs.

The TSSs introduced in Ref. [6] are TSSs that define transition relations on binding terms. Binding terms, first introduced in Ref. [12], are basically second-order terms of a restricted kind, suitable to deal with variable binding operators. As a result, binding terms are not meant to deal with general second-order operators. They do not support higher-order operators other than the second-order operators that can be regarded as variable binding operators. Consequently, the new TSSs are for example not intrinsically appropriate to provide higher-order process calculi with an operational semantics. Approaches to structural operational semantics for the higher-order case have, for example, been studied in Refs. [13,14].

The structure of this paper is as follows. Section 2 covers the preliminaries needed in the remainder of the paper. In Section 3, the basic approach to structural operational semantics using TSSs, which does not cover variable binding operators, is presented. The extension to deal with variable binding operators is introduced in Section 4. In Section 5, operational conservativity of TSSs is defined and syntactic criteria to determine whether a TSS is an operational conservative extension of another TSS are given which are applicable to the setting with variable binding operators. The adapted discussion about given sorts can be found in Section 6. Finally, in Section 7, some concluding remarks are made.

## 2 Preliminaries

In this section, we briefly review the basic notions on which the material presented in this paper is founded and establish the notation and terminology used.

### *2.1 Signatures, terms and equations*

We assume a set $\mathcal{S}$ of *sorts* (type symbols), a set $\mathcal{O}$ of *operators* (function symbols) and a set $\mathcal{V}$ of *variables*. Each operator $o \in \mathcal{O}$ has a sequence of *argument* sorts $\langle s_1, \ldots, s_n \rangle \in \mathcal{S}^*$ and a *result* sort $s \in \mathcal{S}$. Each variable $x \in \mathcal{V}$ has a sort $s \in \mathcal{S}$. It is assumed that the sets $\mathcal{V}$ and $\mathcal{O}$ are disjoint. We use the notation $o : s_1 \times \ldots \times s_n \to s$ to indicate that $o$ is an operator of which the sequence of argument sorts is $\langle s_1, \ldots, s_n \rangle$ and the result sort is $s$. We use the notation $x : s$ to indicate that $x$ is a variable of which the sort is $s$.

Constants are regarded as nullary operators, i.e. operators of which the sequence of argument sorts has length 0.

A (many-sorted) *signature* is a pair $\Sigma = (S, O)$, with $S \subseteq \mathcal{S}$ and $O \subseteq \mathcal{O}$, such that for all $o \in O$, if $o : s_1 \times \ldots \times s_n \to s$, then $s_1, \ldots, s_n, s \in S$.

Let $\Sigma = (S, O)$ be a signature. Then the *variable domain* for $\Sigma$, written $\mathcal{V}_\Sigma$, is the set $\{x \in \mathcal{V} \mid \exists s \in S \bullet x : s\}$.

Let $\Sigma = (S, O)$ be a signature and $X \subseteq \mathcal{V}_\Sigma$. For each $s \in S$, there is a set $\mathcal{T}_\Sigma(X)_s$ of *terms* over $\Sigma$ and $X$ of sort $s$. These sets are the smallest sets satisfying:

(1) if $x \in X$ and $x : s$, then $x \in \mathcal{T}_\Sigma(X)_s$;

(2) if $o \in O$, $o : s_1 \times \ldots \times s_n \to s$, and $t_1 \in \mathcal{T}_\Sigma(X)_{s_1}, \ldots, t_n \in \mathcal{T}_\Sigma(X)_{s_n}$, then $o(t_1, \ldots, t_n) \in \mathcal{T}_\Sigma(X)_s$.

Nullary operators are used as terms: we write $o$ for the term $o()$. The set $\mathcal{T}_\Sigma(X)$ of terms over $\Sigma$ and $X$ is the set $\bigcup\{\mathcal{T}_\Sigma(X)_s \mid s \in S\}$. For each $t \in \mathcal{T}_\Sigma(X)$, we write $\mathrm{s}(t)$ for the sort $s \in S$ such that $t \in \mathcal{T}_\Sigma(X)_s$. We write $\mathcal{T}_\Sigma$ for the set $\mathcal{T}_\Sigma(\mathcal{V}_\Sigma)$. The set $\mathcal{T}_\Sigma$ is called the set of terms over $\Sigma$. A term over $\Sigma$ is also called a $\Sigma$-term.

A term $t$ is *closed* if it does not contain variables. We write $\mathcal{CT}_{\Sigma s}$ for the set $\mathcal{T}_\Sigma(\emptyset)_s$ of closed $\Sigma$-terms of sort $s$ and we write $\mathcal{CT}_\Sigma$ for the set $\mathcal{T}_\Sigma(\emptyset)$ of closed $\Sigma$-terms.

A *substitution* of terms over $\Sigma$ and $X$ for variables in $X$ is a sort-respecting function $\sigma : X \to \mathcal{T}_\Sigma(X)$. A substitution $\sigma$ extends from variables to terms in the obvious way: $\sigma(t)$ is the term obtained by simultaneously replacing in $t$ all occurrences of variables $x$ by $\sigma(x)$. We usually write $t\sigma$ for $\sigma(t)$. We write $[t_1, \ldots, t_n/x_1, \ldots, x_n]$ for the substitution $\sigma$ such that $\sigma(x_1) = t_1, \ldots, \sigma(x_n) = t_n$ and $\sigma(x) = x$ if $x \notin \{x_1, \ldots, x_n\}$. A substitution $\sigma : X \to \mathcal{T}_\Sigma(X)$ is *closed* if $\sigma(x) \in \mathcal{CT}_\Sigma$ for all $x \in X$.

Let $\Sigma = (S, O)$ be a signature and $X \subseteq \mathcal{V}_\Sigma$. Then the set $\mathcal{E}_\Sigma(X)$ of *equations* over $\Sigma$ and $X$ is the smallest set satisfying:

if $t_1, t_2 \in \mathcal{T}_\Sigma(X)_s$ for some $s \in S$, then $t_1 = t_2 \in \mathcal{E}_\Sigma(X)$.

We write $\mathcal{E}_\Sigma$ for the set $\mathcal{E}_\Sigma(\mathcal{V}_\Sigma)$. The set $\mathcal{E}_\Sigma$ is called the set of equations over $\Sigma$. An equation over $\Sigma$ is also called a $\Sigma$-equation.

An equation $e$ is *closed* if both terms occurring in it are closed. We write $\mathcal{CE}_\Sigma$ for the set $\mathcal{E}_\Sigma(\emptyset)$ of closed $\Sigma$-equations.

Let $E \subseteq \mathcal{E}_\Sigma(X)$ and $e \in \mathcal{E}_\Sigma(X)$. Then $e$ is *derivable* from $E$, written $E \vdash e$, if it is justified by the following rules:

(1) if $t_1 = t_2 \in E$, then $E \vdash t_1 = t_2$;
(2) if $t \in \mathcal{T}_\Sigma(X)$, then $E \vdash t = t$;
(3) if $E \vdash t_1 = t_2$, then $E \vdash t_2 = t_1$;
(4) if $E \vdash t_1 = t_2$ and $E \vdash t_2 = t_3$, then $E \vdash t_1 = t_3$;
(5) if $E \vdash t_1 = t_2$, $E \vdash t'_1 = t'_2$, $x \in X$ and $x : \mathrm{s}(t'_1)$, then $E \vdash t_1[t'_1/x] = t_2[t'_2/x]$.

Let $\Sigma = (S, O)$ be a signature. Then an *algebra* $\mathcal{A}$ with signature $\Sigma$ consists of:

(1) for each $s \in S$, a non-empty set $\mathcal{A}_s$, called the *carrier* of $s$;
(2) for each $o \in O$, $o : s_1 \times \ldots \times s_n \to s$, a function $o^{\mathcal{A}} : \mathcal{A}_{s_1} \times \ldots \times \mathcal{A}_{s_n} \to \mathcal{A}_s$, called the *interpretation* of $o$.

An algebra with signature $\Sigma$ is also called a $\Sigma$-algebra. Sometimes, we loosely write $\mathcal{A}$ for the set $\bigcup\{\mathcal{A}_s \mid s \in S\}$.

Let $\mathcal{A}$ be an algebra with signature $\Sigma = (S, O)$ and $X \subseteq \mathcal{V}_\Sigma$. Then an *assignment* in $\mathcal{A}$ for variables in $X$ is a sort-respecting function $\alpha : X \to \mathcal{A}$. For every assignment $\alpha : X \to \mathcal{A}$, $x \in X$, $x : s$, and $d \in \mathcal{A}_s$ $(s \in S)$, we write $\alpha(x \to d)$ for the assignment $\alpha' : X \to \mathcal{A}$ such that $\alpha'(y) = \alpha(x)$ if $y \neq x$ and $\alpha'(x) = d$.

Let $\mathcal{A}$ be a $\Sigma$-algebra, $X \subseteq \mathcal{V}_\Sigma$, and $\alpha : X \to \mathcal{A}$ be an assignment in $\mathcal{A}$ for variables in $X$. Then the *term evaluation* function extending $\alpha$ is the sort-respecting function $\alpha^* : \mathcal{T}_\Sigma(X) \to \mathcal{A}$ recursively defined by

(1) $\alpha^*(x) = \alpha(x)$ ;
(2) $\alpha^*(o(t_1, \ldots, t_n)) = o^{\mathcal{A}}(\alpha^*(t_1), \ldots, \alpha^*(t_n))$ .

Let $\mathcal{A}$ be a $\Sigma$-algebra, $X \subseteq \mathcal{V}_\Sigma$, and $t_1 = t_2 \in \mathcal{E}_\Sigma(X)$. Then $t_1 = t_2$ *holds* in $\mathcal{A}$, written $\mathcal{A} \models t_1 = t_2$, if $\alpha^*(t_1) = \alpha^*(t_2)$ for all assignments $\alpha : X \to \mathcal{A}$.

Let $E \subseteq \mathcal{E}_\Sigma(X)$. Then $\mathcal{A}$ is a *model* of $E$, written $\mathcal{A} \models E$, if $\mathcal{A} \models e$ for all $e \in E$.

Let $\mathcal{A}$ be a $\Sigma$-algebra and $E$ be a set of $\Sigma$-equations. Then $E$ is a *sound* axiomatization of $\mathcal{A}$ (for closed terms) if for all $e \in \mathcal{CE}_\Sigma$: $E \vdash e \;\Rightarrow\; \mathcal{A} \models e$; and $E$ is a *complete* axiomatization of $\mathcal{A}$ (for closed terms) if for all $e \in \mathcal{CE}_\Sigma$: $E \vdash e \;\Leftarrow\; \mathcal{A} \models e$.

Let $\Sigma = (S, O)$ be a signature and $X \subseteq \mathcal{V}_\Sigma$ such that for all $s \in S$, $\mathcal{T}_\Sigma(X)_s \neq \emptyset$. Then the *algebra of terms* over $\Sigma$ and $X$, written $\mathcal{T}_\Sigma(X)$, is the $\Sigma$-algebra where

(1) for each $s \in S$, the carrier of $s$ is $\mathcal{T}_\Sigma(X)_s$;
(2) for each $o \in O$, $o : s_1 \times \ldots \times s_n \to s$, the interpretation of $o$ is the function $o^{\mathcal{T}_\Sigma(X)} : \mathcal{T}_\Sigma(X)_{s_1} \times \ldots \times \mathcal{T}_\Sigma(X)_{s_n} \to \mathcal{T}_\Sigma(X)_s$ such that for all $t_1 \in \mathcal{T}_\Sigma(X)_{s_1}$, $\ldots$, $t_n \in \mathcal{T}_\Sigma(X)_{s_n}$, $o^{\mathcal{T}_\Sigma(X)}(t_1, \ldots, t_n) = o(t_1, \ldots, t_n)$.

The *algebra of closed terms* over $\Sigma$, written $\mathcal{CT}_\Sigma$, is the *algebra of terms* over

$\Sigma$ and $\emptyset$.

Let $\mathcal{A}$ be an algebra with signature $\Sigma = (S, O)$. Then a (sort-respecting) equivalence relation $\sim \subseteq \mathcal{A} \times \mathcal{A}$ is a *congruence* on $\mathcal{A}$ if for each $o \in O$, $o : s_1 \times \ldots \times s_n \to s$, we have for all $a_1, a'_1 \in \mathcal{A}_{s_1}, \ldots, a_n, a'_n \in \mathcal{A}_{s_n}$:

$$a_1 \sim a'_1, \ldots, a_n \sim a'_n \;\Rightarrow\; o^{\mathcal{A}}(a_1, \ldots, a_n) \sim o^{\mathcal{A}}(a'_1, \ldots, a'_n) \ .$$

Let $\sim$ be an equivalence relation on a set $A$. Then we write $[a]_\sim$, where $a \in A$, for the equivalence class $\{a' \in A \mid a \sim a'\}$; and we write $A/\sim$ for the quotient set $\{[a]_\sim \mid a \in A\}$.

Let $\mathcal{A}$ be an algebra with signature $\Sigma = (S, O)$ and $\sim \subseteq \mathcal{A} \times \mathcal{A}$ be a congruence on $\mathcal{A}$. Then the *quotient algebra* of $\mathcal{A}$ by $\sim$, written $\mathcal{A}/\sim$, is the $\Sigma$-algebra where

(1) for each $s \in S$, the carrier of $s$ is $\mathcal{A}_s/\sim$;
(2) for each $o \in O$, $o : s_1 \times \ldots \times s_n \to s$, the interpretation of $o$ is the function $o^{\mathcal{A}/\sim} : \mathcal{A}_{s_1}/\sim \times \ldots \times \mathcal{A}_{s_n}/\sim \to \mathcal{A}_s/\sim$ such that for all $a_1 \in \mathcal{A}_{s_1}, \ldots, a_n \in \mathcal{A}_{s_n}$, $o^{\mathcal{A}/\sim}([a_1]_\sim, \ldots, [a_n]_\sim) = [o^{\mathcal{A}}(a_1, \ldots, a_n)]_\sim$.

## 3 The basic approach

In this section, we introduce the approach to structural operational semantics using TSSs that define unary and binary transition relations by means of transition rules with positive and negative premises. In this approach, developed in Refs. [1–5], variable binding operators are not covered.

### *3.1 Transition system specifications*

The main constituent of a transition system specification is a collection of transition rules defining certain transition relations. Each transition rule is made up of transition formulas. We will define transition formulas and transition rules over a signature and a domain of transition predicates. Therefore, we first define the notion of domain of transition predicates. Roughly speaking, a domain of transition predicates consists of unary and binary predicates (relation symbols), each predicate being given a sequence of argument sorts.

We assume a set $\mathcal{P}$ of *predicates*. Each predicate $p \in \mathcal{P}$ has a sequence of *argument* sorts $\langle s_1, \ldots, s_n \rangle \in \mathcal{S}^*$. It is assumed that the sets $\mathcal{V}$, $\mathcal{O}$ and $\mathcal{P}$ are mutually disjoint. We use the notation $p : s_1 \times \ldots \times s_n$ to indicate that $p$ is a predicate of which the sequence of argument sorts is $\langle s_1, \ldots, s_n \rangle$.

Let $\Sigma = (S, O)$ be a signature. Then a *domain of transition predicates* on $\Sigma$-terms is a set $\Pi \subseteq \mathcal{P}$ such that for all $p \in \Pi$, if $p : s_1 \times \ldots \times s_n$, then $s_1, \ldots, s_n \in S$ and $n = 1$ or $2$.

Transition predicates are defined here in an uncommon way to anticipate the generalization to parametrized transition predicates discussed in Section 6.

Next, we define the notions of positive and negative transition formula. We also introduce the notion of denial of a transition formula and make the notion of closed transition formula precise.

Let $\Pi$ be a domain of transition predicates on $\Sigma$-terms. Then the set $\mathcal{F}^{+}_{\Sigma,\Pi}$ of *positive transition formulas* over $\Sigma$ and $\Pi$ and the set $\mathcal{F}^{-}_{\Sigma,\Pi}$ of *negative transition formulas* over $\Sigma$ and $\Pi$ are the smallest sets satisfying:

if $p \in \Pi$, $p : s_1 \times \ldots \times s_n$, and $t_1 \in \mathcal{T}_{\Sigma s_1}, \ldots, t_n \in \mathcal{T}_{\Sigma s_n}$,
then $p(t_1, \ldots, t_n) \in \mathcal{F}^{+}_{\Sigma,\Pi}$;

if $p \in \Pi$, $p : s_1 \times \ldots \times s_n$, and $t_1 \in \mathcal{T}_{\Sigma s_1}, \ldots, t_n \in \mathcal{T}_{\Sigma s_n}$,
then $\neg p(t_1, \ldots, t_n) \in \mathcal{F}^{-}_{\Sigma,\Pi}$.

Bear in mind that $p \in \Pi$ implies $1 \leq n \leq 2$. We use in general postfix notation for unary predicates and infix notation for binary predicates. We write $\mathcal{F}_{\Sigma,\Pi}$ for $\mathcal{F}^{+}_{\Sigma,\Pi} \cup \mathcal{F}^{-}_{\Sigma,\Pi}$. For $\phi \in \mathcal{F}_{\Sigma,\Pi}$, $\overline{\phi}$, the *denial* of $\phi$, is defined as follows:

$$\overline{p(t_1, \ldots, t_m)} = \neg p(t_1, \ldots, t_m) \;, \qquad \overline{\neg p(t_1, \ldots, t_m)} = p(t_1, \ldots, t_m) \;.$$

A positive or negative transition formula $\phi$ is *closed* if all terms occurring in it are closed. We write $\mathcal{CF}^{+}_{\Sigma,\Pi}$ for $\{\phi \in \mathcal{F}^{+}_{\Sigma,\Pi} \mid \phi \text{ is closed}\}$ and $\mathcal{CF}^{-}_{\Sigma,\Pi}$ for $\{\phi \in \mathcal{F}^{-}_{\Sigma,\Pi} \mid \phi \text{ is closed}\}$. Furthermore, we write $\mathcal{CF}_{\Sigma,\Pi}$ for $\mathcal{CF}^{+}_{\Sigma,\Pi} \cup \mathcal{CF}^{-}_{\Sigma,\Pi}$.

In the following definition, the notion of transition rule is defined. The notions of substitution instance and closed substitution instance of a transition rule are also introduced.

Let $\Pi$ be a domain of transition predicates on $\Sigma$-terms. Then the set $\mathcal{R}_{\Sigma,\Pi}$ of *transition rules* over $\Sigma$ and $\Pi$ is the smallest set satisfying:

if $\Phi \subseteq \mathcal{F}_{\Sigma,\Pi}$ and $\psi \in \mathcal{F}^{+}_{\Sigma,\Pi}$, then $\dfrac{\Phi}{\psi} \in \mathcal{R}_{\Sigma,\Pi}$.

Let $r = \dfrac{\Phi}{\psi}$ be a transition rule. Then the transition formulas in $\Phi$ are the *premises* of $r$ and the transition formula $\psi$ is the *conclusion* of $r$. A transition rule $r$ is *closed* if all formulas occurring in it are closed. Substitution extends from terms to formulas and rules as expected. For every substitution $\sigma : \mathcal{V}_\Sigma \to \mathcal{T}_\Sigma$ and transition rule $r$, the transition rule $\sigma(r)$ is a *substitution instance* of $r$. If $\sigma$ is a closed substitution, the transition rule $\sigma(r)$ is a *closed* substitution instance of $r$.

We are now ready to define the notion of transition system specification.

A *transition system specification* (TSS) is a triple $P = (\Sigma, \Pi, R)$, where

(1) $\Sigma$ is a signature;
(2) $\Pi$ is a domain of transition predicates on $\Sigma$-terms;
(3) $R \subseteq \mathcal{R}_{\Sigma,\Pi}$.

We write $\mathrm{si}(R)$ for the set of all substitution instances of $r \in R$ and $\mathrm{csi}(R)$ for the set of all closed substitution instances of $r \in R$.

**Example 1** *We consider the signature* $\Sigma_\mathsf{C} = (\{\mathsf{C}\}, \{\mathsf{0_c}, \mathsf{s_c}\})$*, with* $\mathsf{0_c}\colon \to \mathsf{C}$ *and* $\mathsf{s_c} : \mathsf{C} \to \mathsf{C}$*, and the transition predicate domain* $\Pi_\mathsf{C} = \{\xrightarrow{\mathsf{inc}}, \xrightarrow{\mathsf{dec}}, \xrightarrow{\mathsf{even}}\}$*, with* $\xrightarrow{\mathsf{inc}} : \mathsf{C} \times \mathsf{C}$*,* $\xrightarrow{\mathsf{dec}} : \mathsf{C} \times \mathsf{C}$ *and* $\xrightarrow{\mathsf{even}} : \mathsf{C} \times \mathsf{C}$*. The signature* $\Sigma_\mathsf{C}$ *introduces terms intended to be used as expressions for counters. A counter can freely be incremented, but it can only be decremented once for each time it has been incremented. The idea is that the term* $\mathsf{0_c}$ *represents a counter that cannot be decremented and that the term* $\mathsf{s_c}(t)$*, where* $t \in \mathcal{CT}_{\Sigma_\mathsf{C}}$*, represents a counter that can be decremented once more than the counter represented by* $t$*. In addition, it can be checked whether a counter can be decremented for an even number of times. This operational behaviour is modeled by the TSS* $\mathrm{P}_\mathsf{C} = (\Sigma_\mathsf{C}, \Pi_\mathsf{C}, \mathrm{R}_\mathsf{C})$*, where* $\mathrm{R}_\mathsf{C}$ *consists of the following transition rules:*

$$\frac{}{x \xrightarrow{\mathsf{inc}} \mathsf{s_c}(x)} \quad \frac{y \xrightarrow{\mathsf{inc}} x}{x \xrightarrow{\mathsf{dec}} y} \quad \frac{}{\mathsf{0_c} \xrightarrow{\mathsf{even}} \mathsf{0_c}} \quad \frac{\neg(x \xrightarrow{\mathsf{even}} x)}{\mathsf{s_c}(x) \xrightarrow{\mathsf{even}} \mathsf{s_c}(x)} \ .$$

*An example of a closed substitution instance of a transition rule from* $\mathrm{R}_\mathsf{C}$ *is*

$$\frac{\mathsf{0_c} \xrightarrow{\mathsf{inc}} \mathsf{s_c}(\mathsf{0_c})}{\mathsf{s_c}(\mathsf{0_c}) \xrightarrow{\mathsf{dec}} \mathsf{0_c}} \ .$$

*It is obtained from the second transition rule by means of a closed substitution* $\sigma$ *such that* $\sigma(x) = \mathsf{s_c}(\mathsf{0_c})$ *and* $\sigma(y) = \mathsf{0_c}$*.*

### *3.2 Proofs from TSSs*

In the following definition, we introduce a general notion of proof from a TSS by allowing to prove transition rules. The proof of a transition rule $\frac{\Phi}{\psi}$ corresponds to the proof of the transition formula $\psi$ under the assumptions $\Phi$.

Let $P = (\Sigma, \Pi, R)$ be a TSS. Then a *proof* of a transition rule $\frac{\Phi}{\psi}$ from $P$ is a well-founded, upwardly branching tree of which the nodes are labelled by formulas in $\mathcal{F}_{\Sigma,\Pi}$, such that

(1) the root is labelled by $\psi$;
(2) if a node is labelled by $\psi'$ and $\Phi'$ is the set of labels of the nodes directly above this node, then

$$\text{either} \quad \psi' \in \Phi \text{ and } \Phi' = \emptyset \quad \text{or} \quad \frac{\Phi'}{\psi'} \in \mathrm{si}(R) \ .$$

A transition rule $r$ is *provable* from $P$, written $P \vdash r$, if there exists a proof of $r$ from $P$. A positive transition formula $\phi$ is *provable* from $P$, written $P \vdash \phi$, if there exists a proof of $\frac{\emptyset}{\phi}$ from $P$.

In the following definition, we introduce the notion of well-supported proof from a TSS. It incorporates a form of negation as failure.

Let $P = (\Sigma, \Pi, R)$ be a TSS. Then a *well-supported proof* of a closed transition formula $\psi$ from $P$ is like a proof of $\frac{\emptyset}{\psi}$ from $P$, but admitting under 2 additionally

$$\text{or} \quad \psi' \text{ is negative and for all sets } N \subseteq \mathcal{CF}^{-}_{\Sigma,\Pi} \text{ such that } P \vdash \frac{N}{\psi'} \text{ there exists a } \phi' \in \Phi' \text{ such that } \overline{\phi'} \in N \ .$$

A closed transition formula $\phi$ is *ws-provable* from $P$, written $P \vdash_{\mathrm{ws}} \psi$, if there exists a well-supported proof of $\phi$ from $P$.

In a well-supported proof, it is allowed to infer the denial of a closed positive transition formula $\phi$, if it is manifestly impossible to infer $\phi$ because every conceivable proof of $\phi$ involves a negative premise of which the denial has already been proved. This fits in with the idea that the only closed positive transition formulas that hold in the intended model of a TSS are those inferable from the transition rules under assumption of closed negative transition formulas that do not lead to inconsistencies. However, in the case where this principle is applied, it is not precluded that there still exists a closed positive transition formula of which it is not possible to establish whether it holds in the intended model or not. Therefore, we also introduce the notion of complete TSS.

Let $P = (\Sigma, \Pi, R)$ be a TSS. Then $P$ is *complete* if for all $\phi \in \mathcal{CF}_{\Sigma,\Pi}$, either $P \vdash_{\mathrm{ws}} \phi$ or $P \vdash_{\mathrm{ws}} \overline{\phi}$.

Only complete TSSs are considered to be meaningful in this paper. This choice is dictated by the observation that in virtually all applications of TSSs, it is essential that it can be established for every closed positive transition formula whether it holds in the intended model or not. It is, for example, the case with transition rule formats guaranteeing that bisimulation equivalence is a congruence and syntactic criteria to determine operational conservativity.

**Example 2** *We consider the TSS* $\mathrm{P}_{\mathsf{C}}$ *of Example 1. The following is a well-supported proof of* $\mathsf{s_c}(\mathsf{s_c}(\mathsf{0_c})) \xrightarrow{\mathsf{even}} \mathsf{s_c}(\mathsf{s_c}(\mathsf{0_c}))$ *from* $\mathrm{P}_{\mathsf{C}}$*:*

$$\mathsf{0_c} \xrightarrow{\mathsf{even}} \mathsf{0_c}$$

$$\neg(\mathsf{s_c}(\mathsf{0_c}) \xrightarrow{\mathsf{even}} \mathsf{s_c}(\mathsf{0_c}))$$

$$\mathsf{s_c}(\mathsf{s_c}(\mathsf{0_c})) \xrightarrow{\mathsf{even}} \mathsf{s_c}(\mathsf{s_c}(\mathsf{0_c}))$$

### *3.3 Models of TSSs*

The models of a TSS are known as transition systems. We define transition systems with respect to a signature and a domain of transition predicates.

Let $\Pi$ be a domain of transition predicates on $\Sigma$-terms. A *transition system* $\mathcal{TS}$ for $\Sigma$ and $\Pi$ consists of:

> for each $p \in \Pi$, $p : s_1 \times \ldots \times s_n$, a relation $p^{\mathcal{TS}} \subseteq \mathcal{CT}_{\Sigma\, s_1} \times \ldots \times \mathcal{CT}_{\Sigma\, s_n}$, called the *interpretation* of $p$.

So transition predicates are interpreted as relations on sets of closed terms.

The following definition makes precise what it means for a closed transition formula to hold in a transition system.

Let $\mathcal{TS}$ be a transition system for signature $\Sigma$ and domain of transition predicates $\Pi$. For $\phi \in \mathcal{CF}_{\Sigma,\Pi}$, $\phi$ *holds in* $\mathcal{TS}$, written $\mathcal{TS} \models \phi$, is defined as follows:

(1) $\mathcal{TS} \models p(t_1, \ldots, t_n)$ if $(t_1, \ldots, t_n) \in p^{\mathcal{TS}}$,
(2) $\mathcal{TS} \models \neg p(t_1, \ldots, t_n)$ if $(t_1, \ldots, t_n) \notin p^{\mathcal{TS}}$.

For $\Phi \subseteq \mathcal{CF}_{\Sigma,\Pi}$, we write $\mathcal{TS} \models \Phi$ to indicate that $\mathcal{TS} \models \phi$ for all $\phi \in \Phi$.

A transition system $\mathcal{TS}$ for $\Sigma$ and $\Pi$ corresponds to the set $F \subseteq \mathcal{CF}^{+}_{\Sigma,\Pi}$ such that, for all $p(t_1, \ldots, t_n) \in \mathcal{CF}^{+}_{\Sigma,\Pi}$, $p(t_1, \ldots, t_n) \in F \Leftrightarrow \mathcal{TS} \models p(t_1, \ldots, t_n)$. Hence, in the light of the last definition, a transition relation on $\Sigma$-terms can be regarded as a set of closed positive transition formulas over $\Sigma$ and $\Pi$. Therefore, closed positive transition formulas are sometimes loosely called *transitions*. This correspondence also clarifies the value attached in Section 3.2 to TSSs being complete.

Now, we can make precise what it means for a transition system to be a model of a TSS and what it means for a transition system to be well-supported by a

TSS.

Let $P = (\Sigma, \Pi, R)$ be a TSS and $\mathcal{TS}$ be a transition system for $\Sigma$ and $\Pi$. Then $\mathcal{TS}$ is a *model* of $P$, written $\mathcal{TS} \models P$, if for all $\psi \in \mathcal{CF}^{+}_{\Sigma,\Pi}$:

$$\mathcal{TS} \models \psi \;\Leftarrow\; \exists \frac{\Phi}{\psi} \in \mathrm{csi}(R) \bullet \mathcal{TS} \models \Phi \;,$$

and $\mathcal{TS}$ is *well-supported* by $P$ if for all $\psi \in \mathcal{CF}^{+}_{\Sigma,\Pi}$:

$$\mathcal{TS} \models \psi \;\Rightarrow\; \exists \Phi \subseteq \mathcal{CF}^{-}_{\Sigma,\Pi} \bullet P \vdash \frac{\Phi}{\psi} \wedge \mathcal{TS} \models \Phi \;.$$

If $\mathcal{TS}$ is a model of $P$ that is well-supported by $P$, we say that $\mathcal{TS}$ is a *well-supported* model of $P$. For $\phi \in \mathcal{CF}_{\Sigma,\Pi}$, we write $P \models_{\mathrm{ws}} \phi$ to indicate that $\mathcal{TS} \models \phi$ for all well-supported models $\mathcal{TS}$ of $P$.

The definition of model expresses that a transition system is a model of a TSS if it obeys the transition rules of the TSS. The definition of well-supportedness expresses that a transition system is well-supported by a TSS if each of its transitions is justified by the transition rules of the TSS and this justification is founded, i.e. it does not make use of the transition itself. We have that $\vdash_{\mathrm{ws}}$ is sound for all well-supported models of a TSS, that is $P \vdash_{\mathrm{ws}} \psi \;\Rightarrow\; P \models_{\mathrm{ws}} \psi$ (Proposition 11 in Ref. [15]).

Suppose that $P = (\Sigma, \Pi, R)$ is a complete TSS and $\mathcal{TS}$ is a transition system for $\Sigma$ and $\Pi$. It is easy to check that the notion of well-supported proof is defined in such a way that $\mathcal{TS}$ is well-supported by $P$ iff for all $\psi \in \mathcal{CF}^{+}_{\Sigma,\Pi}$, $\mathcal{TS} \models \psi \;\Rightarrow\; P \vdash_{\mathrm{ws}} \psi$. From this and the soundness result for $\vdash_{\mathrm{ws}}$, it follows that a complete TSS has a unique well-supported model. Its transitions are exactly the ones justified by a well-supported proof.

Let $P = (\Sigma, \Pi, R)$ be a complete TSS. Then the *intended* model of $P$, written $\mathcal{TS}_P$, is the unique well-supported model of $P$. $\mathcal{TS}_P$ is also called the transition system *associated with* $P$.

Notice that every TSS without negative premises is complete. Moreover, for TSSs without negative premises, $\vdash$ and $\vdash_{\mathrm{ws}}$ coincide on closed transition formulas.

**Example 3** *We consider again the TSS* $\mathrm{P}_{\mathsf{C}}$ *of Example 1. Let* $\mathcal{E}$ *be the smallest subset of* $\mathcal{CT}_{\Sigma_{\mathsf{C}}}$ *satisfying (1)* $\mathsf{0}_{\mathsf{c}} \in \mathcal{E}$ *and (2) if* $t \in \mathcal{E}$*, then* $\mathsf{s}_{\mathsf{c}}(\mathsf{s}_{\mathsf{c}}(t)) \in \mathcal{E}$*. The intended model* $\mathcal{TS}_{\mathrm{P}_{\mathsf{C}}}$ *has* $\{(t, \mathsf{s}_{\mathsf{c}}(t)) \mid t \in \mathcal{CT}_{\Sigma_{\mathsf{C}}}\}$*,* $\{(\mathsf{s}_{\mathsf{c}}(t), t) \mid t \in \mathcal{CT}_{\Sigma_{\mathsf{C}}}\}$*, and* $\{(t, t) \mid t \in \mathcal{E}\}$ *as interpretations of the transition predicates* $\xrightarrow{\mathsf{inc}}$*,* $\xrightarrow{\mathsf{dec}}$*, and* $\xrightarrow{\mathsf{even}}$*, respectively.*

Bisimulation equivalence is a frequently used equivalence to abstract from irrelevant details of operational semantics. We define bisimulation equivalence with respect to a TSS.

Let $P = (\Sigma, \Pi, R)$ be a TSS. Then a *bisimulation* $B$ based on $P$ is a sort-respecting symmetric binary relation $B \subseteq \mathcal{CT}_\Sigma \times \mathcal{CT}_\Sigma$ such that:

(1) if $B(t_1, t'_1)$ and $\mathcal{TS}_P \models p(t_1, t_2)$, then $\exists t'_2 \bullet \mathcal{TS}_P \models p(t'_1, t'_2)$ and $B(t_2, t'_2)$;
(2) if $B(t_1, t'_1)$ and $\mathcal{TS}_P \models p(t_1)$, then $\mathcal{TS}_P \models p(t'_1)$.

Two closed $\Sigma$-terms $t$ and $t'$ are *bisimulation equivalent* in $P$, written $t \leftrightarrow_P t'$, if there exists a bisimulation $B$ such that $B(t, t')$.

The transition rule format known as the panth format guarantees that bisimulation equivalence is a congruence.

Let $P = (\Sigma, \Pi, R)$ be a TSS. Then a transition rule $r \in R$ is in *panth format* if it satisfies:

(1) the second argument of each premise of $r$ that has the form $p(t_1, t_2)$ is a variable;
(2) the second argument of each premise of $r$ that has the form $\neg p(t_1, t_2)$ is a closed term;
(3) the first argument of the conclusion of $r$ has one of the following forms: $x$ or $o(x_1, \ldots, x_n)$;
(4) the variables that occur as second argument of a premise that has the form $p(t_1, t_2)$ or in the first argument of the conclusion are mutually distinct.

The TSS $P$ is in *panth format* if each transition rule $r \in R$ is in panth format.

**Theorem 4 (Congruence)** *Let $P = (\Sigma, \Pi, R)$ be a complete TSS in panth format. Then $\leftrightarrow_P$ is a congruence on the algebra of closed terms over $\Sigma$.*

**Proof.** In the one-sorted case, it follows immediately from Theorem 4.5 in Ref. [4] and Corollary 5.7 in Ref. [16]. However, it is immediately clear that those theorems go through in the many-sorted case seeing that their proofs do not depend on the lack of many-sortedness. □

Consider a TSS $P = (\Sigma, \Pi, R)$ in panth format. Then it is certain that we can construct $\mathcal{CT}_\Sigma / \leftrightarrow_P$, the quotient algebra of the algebra of closed terms over

$\Sigma$ by bisimulation equivalence. If this $\Sigma$-algebra is intended to be a model of some set of $\Sigma$-equations, then this algebra is usually called its *bisimulation* model.

**Example 5** *We consider once more the TSS* $\mathrm{P}_{\mathsf{C}}$ *of Example 1. It is not in panth format because its second and fourth transition rule are not in panth format. We can replace those two rules by rules in panth format such that the transition system associated with the resulting TSS is the same as the transition system associated with the original TSS. The transition rules of the new TSS are as follows:*

$$\frac{}{x \xrightarrow{\mathsf{inc}} \mathsf{s_C}(x)} \quad \frac{}{\mathsf{s_C}(x) \xrightarrow{\mathsf{dec}} x} \quad \frac{}{\mathsf{0_C} \xrightarrow{\mathsf{even}} \mathsf{0_C}} \quad \frac{\{\neg(x \xrightarrow{\mathsf{even}} t) \mid t \in \mathcal{CT}_{\Sigma_\mathsf{C}}\}}{\mathsf{s_C}(x) \xrightarrow{\mathsf{even}} \mathsf{s_C}(x)} \ .$$

*It is easy to see that the new TSS, say* $\mathrm{P}'_{\mathsf{C}}$*, is also complete. Hence,* $\leftrightarrows_{\mathrm{P}'_\mathsf{C}}$ *is a congruence on the algebra of closed terms over* $\Sigma_\mathsf{C}$*. In this particular case, this result is not really relevant because* $\leftrightarrows_{\mathrm{P}'_\mathsf{C}}$ *is the identity relation on* $\mathcal{CT}_{\Sigma_\mathsf{C}}$*.*

## 4 Variable binding operators

The generalization of the relevant notions – such as signature, term, equation, algebra, transition rule, bisimulation equivalence and panth format – needed to deal with variable binding operators is rather straightforward. Additional rules to derive equations ensue from it.

### *4.1 Signatures, terms, equations and algebras*

For clearness' sake, we now call the elements of $\mathcal{S}$ *base sorts*. To begin with, we need other sorts, which are built up from base sorts. If $S \subseteq \mathcal{S}$ and $s_1, \ldots, s_n, s \in S$, then $s_1, \ldots, s_n \,.\, s$ is a *binding sort* over $S$. We write $\mathcal{B}(S)$ for the union of $S$ and the set of all binding sorts over $S$. The carrier of a base sort $s$ consists of objects which are called ordinary objects. The carrier of a binding sort $s_1, \ldots, s_n \,.\, s$ consists of functions from the cartesian product of the carriers of the base sorts $s_1, \ldots, s_n$ to the carrier of the base sort $s$. Binding sorts are used for variable binding in arguments of operators. Argument sorts of operators may be binding sorts; result sorts must be base sorts. Suppose that $o : \mathfrak{s}_1 \times \ldots \times \mathfrak{s}_n \to s$. If $\mathfrak{s}_i = s_{i1}, \ldots, s_{in_i} \,.\, s_i$ $(1 \leq i \leq n)$, then $o$ binds $n_i$ variables, of base sorts $s_{i1}, \ldots, s_{in_i}$, in the $i$th argument. Otherwise, i.e. if $\mathfrak{s}_i \in S$, it does not bind any variable in the $i$th argument. Sorts of variables may also be binding sorts.

A *binding signature* is now a pair $\Sigma = (S, O)$, with $S \subseteq \mathcal{S}$ and $O \subseteq \mathcal{O}$, such that for all $o \in O$, if $o : \mathfrak{s}_1 \times \ldots \times \mathfrak{s}_n \to s$, then $\mathfrak{s}_1, \ldots, \mathfrak{s}_n, s \in \mathcal{B}(S)$.

For a binding signature $\Sigma$, the variable domain $\mathcal{V}_\Sigma$ is the set $\{x \in \mathcal{V} \mid \exists s \in \mathcal{B}(S) \bullet x : s\}$.

Let $\Sigma = (S, O)$ be a binding signature and $X \subseteq \mathcal{V}_\Sigma$. For each $\mathfrak{s} \in \mathcal{B}(S)$, there is a set $\mathcal{T}_\Sigma(X)_\mathfrak{s}$ of *binding terms* over $\Sigma$ and $X$ of sort $\mathfrak{s}$. These sets are the smallest sets satisfying:

(1) if $x \in X$ and $x : s$, with $s \in S$, then $x \in \mathcal{T}_\Sigma(X)_s$;
(2) if $x \in X$, $x : s_1, \ldots, s_n \,.\, s$, and $t_1 \in \mathcal{T}_\Sigma(X)_{s_1}, \ldots, t_n \in \mathcal{T}_\Sigma(X)_{s_n}$, then $x(t_1, \ldots, t_n) \in \mathcal{T}_\Sigma(X)_s$;
(3) if $x_1, \ldots, x_n \in X$, $x_1 : s_1, \ldots, x_n : s_n$, $t \in \mathcal{T}_\Sigma(X)_s$, with $s_1, \ldots, s_n, s \in S$, and $x_1, \ldots, x_n$ are mutually distinct, then $x_1, \ldots, x_n \,.\, t \in \mathcal{T}_\Sigma(X)_{s_1,\ldots,s_n.s}$;
(4) if $o \in O$, $o : \mathfrak{s}_1 \times \ldots \times \mathfrak{s}_n \to s$, and $t_1 \in \mathcal{T}_\Sigma(X)_{\mathfrak{s}_1}, \ldots, t_n \in \mathcal{T}_\Sigma(X)_{\mathfrak{s}_n}$, then $o(t_1, \ldots, t_n) \in \mathcal{T}_\Sigma(X)_s$.

Rule 2 shows that variables of binding sorts have arguments. Notice that binding terms formed by application of rule 3 serve only as argument of operators. Binding terms of which the sort is a base sort are ordinary terms.

In the case of binding terms, the notion of closed term must be generalized. An occurrence of a variable $x$ in a binding term $t$ is *bound* if the occurrence is in a subterm of the form $x_1, \ldots, x_n \,.\, t'$ with $x \in \{x_1, \ldots, x_n\}$; otherwise it is *free*. If $x$ has at least one bound occurrence in $t$, it is called a *bound variable* of $t$. If $x$ has at least one free occurrence in $t$, it is called a *free variable* of $t$. A binding term $t$ is *closed* if it is a binding term without free variables. We still write $\mathcal{CT}_\Sigma$ for the set of closed binding terms.

The extension of a substitution $\sigma$ from variables to binding terms differs in two ways from the extension of a substitution from variables to ordinary terms. First of all, only free occurrences of variables are replaced and bound variables are renamed if needed to avoid free occurrences of variables in the replacing terms becoming bound. Secondly, if $\sigma(x) = x_1, \ldots, x_n \,.\, t$, a term of the form $x(t_1, \ldots, t_n)$ is replaced as a whole by the term $t[\sigma(t_1), \ldots, \sigma(t_n)/x_1, \ldots, x_n]$. Substitution is only defined up to change of bound variables. This is justified because binding terms that can be obtained from each other by change of bound variables are not distinguished semantically.

In the case of binding signatures, the definition of the notion of equation has to be adapted to include equations of which both sides are terms of a binding sort. Two additional rules are available to derive such equations:

(1) if $x_1, \ldots, x_n \,.\, t \in \mathcal{T}_\Sigma(X)$, $y_1, \ldots, y_n \in X$, $y_1 : \mathrm{s}(x_1), \ldots, y_n : \mathrm{s}(x_n)$, and $y_1, \ldots, y_n$ are mutually distinct, then $E \vdash x_1, \ldots, x_n \,.\, t = y_1, \ldots, y_n \,.$

$t[y_1, \ldots, y_n/x_1, \ldots, x_n]$;

(2) if $E \vdash t_1 = t_2$, $\mathrm{s}(t_1) \in S$, $x_1, \ldots, x_n \in X$ and $\mathrm{s}(x_1), \ldots, \mathrm{s}(x_n) \in S$, then $E \vdash x_1, \ldots, x_n \,.\, t_1 = x_1, \ldots, x_n \,.\, t_2$.

In the case of binding signatures, an algebra differs in two ways from an ordinary algebra. Firstly, there are also carriers for the binding sorts, as explained above. That is, the algebra with signature $\Sigma = (S, O)$ consists of:

(1) for each $\mathfrak{s} \in \mathcal{B}(S)$, a non-empty set $\mathcal{A}_\mathfrak{s}$, called the *carrier* of $\mathfrak{s}$, such that if $\mathfrak{s} \in \mathcal{B}(S) - S$, $\mathfrak{s} = s_1, \ldots, s_n \,.\, s$, then $\mathcal{A}_\mathfrak{s} \subseteq \mathcal{A}_{s_1} \times \ldots \times \mathcal{A}_{s_n} \to \mathcal{A}_s$;

(2) for each $o \in O$, $o : \mathfrak{s}_1 \times \ldots \times \mathfrak{s}_n \to s$, a function $o^\mathcal{A} : \mathcal{A}_{\mathfrak{s}_1} \times \ldots \times \mathcal{A}_{\mathfrak{s}_n} \to \mathcal{A}_s$, called the *interpretation* of $o$.

Secondly, the algebra must satisfy the restriction that each assignment $\alpha$ can be extended to a term evaluation function such that:

(1) $\alpha^*(x) = \alpha(x)$ ;

(2) $\alpha^*(x(t_1, \ldots, t_n)) = \alpha(x)(\alpha^*(t_1), \ldots, \alpha^*(t_n))$ ;

(3) $\alpha^*(x_1, \ldots, x_n \,.\, t)$ is the $f \in \mathcal{A}_{\mathrm{s}(x_1),\ldots,\mathrm{s}(x_n).\mathrm{s}(t)}$ such that, for all $d_1 \in \mathcal{A}_{\mathrm{s}(x_1)}$, $\ldots$, $d_n \in \mathcal{A}_{\mathrm{s}(x_n)}$, $f(d_1, \ldots, d_n) = (\alpha(x_1 \to d_1) \ldots (x_n \to d_n))^*(t)$ ;

(4) $\alpha^*(o(t_1, \ldots, t_n)) = o^\mathcal{A}(\alpha^*(t_1), \ldots, \alpha^*(t_n))$ .

The restriction concerning term evaluation is automatically satisfied by ordinary algebras. Frequently used ways to construct algebras, such as the term algebra construction and the quotient algebra construction, still work in the presence of variable binding operators. For a formal treatment of algebras in the presence of variable binding operators, the reader is referred to Ref. [6].

Henceforth, we usually say signature and term instead of binding signature and binding term, respectively, if it is clear that the latter are meant.

**Example 6** *In CCS [10], the operator $\mu$ is used to define processes recursively. For example, the expression $\mu x \,.\, ax$ denotes the solution of the equation $x = ax$, i.e. the process that will keep on performing action $a$ forever. The operator $\mu$ is in essence a unary variable binding operator that binds one variable in its argument. In the current setting, $\mu x \,.\, t$ becomes simply an abbreviation for $\mu(x \,.\, t)$.*

### *4.2 Transition systems, bisimulation equivalence and the panth format*

In this subsection, we only consider transition predicates that do not bind variables in their arguments. In Ref. [6], we consider transition predicates that may bind variables in their second argument. That yields a slightly weaker congruence result: if transition predicates that bind variables in their second argument are used, the congruence result is limited to TSSs that are *well-founded* (see Ref. [6] for details).

In the case of binding signatures, a transition system differs in one way from an ordinary transition system: terms are identified if they can be obtained from each other by change of bound variables. This is formalized as follows. First of all, we introduce $\dot{\approx}$, the (sort-respecting) congruence on terms induced by change of bound variables. Next, we adapt the definition of the notion of transition system such that transition predicates are interpreted as relations on equivalence classes of closed terms with respect to $\dot{\approx}$. That is, a transition system for signature $\Sigma$ and domain of transition predicates $\Pi$ consists of:

for each $p \in \Pi$, $p\colon s_1 \times \ldots \times s_n$, a relation $p^{\mathcal{TS}} \subseteq \mathcal{CT}_{\Sigma\, s_1}/\dot{\approx} \times \ldots \times \mathcal{CT}_{\Sigma\, s_n}/\dot{\approx}$.

For closed transition formulas $\phi$, *$\phi$ holds in $\mathcal{TS}$*, still written $\mathcal{TS} \models \phi$, is now defined as follows:

(1) $\mathcal{TS} \models p(t_1, \ldots, t_n)$ if $([t_1]_{\dot{\approx}}, \ldots, [t_n]_{\dot{\approx}}) \in p^{\mathcal{TS}}$,
(2) $\mathcal{TS} \models \neg p(t_1, \ldots, t_n)$ if $([t_1]_{\dot{\approx}}, \ldots, [t_n]_{\dot{\approx}}) \notin p^{\mathcal{TS}}$.

The definitions of the notions of TSS, model of a TSS, well-supported model of a TSS, complete TSS and TSS in panth format do not have to be adapted. A bisimulation based on a TSS $P = (\Sigma, \Pi, R)$ must have the following additional properties:

(3) if $t \dot{\approx} t'$, then $B(t, t')$;
(4) if $B(x_1, \ldots, x_n \,.\, t, y_1, \ldots, y_n \,.\, t')$, then $\forall t_1 \in \mathcal{CT}_{\Sigma\, \mathrm{s}(x_1)}, \ldots, t_n \in \mathcal{CT}_{\Sigma\, \mathrm{s}(x_n)} \bullet B(t[t_1, \ldots, t_n / x_1, \ldots, x_n], t'[t_1, \ldots, t_n / y_1, \ldots, y_n])$.

With these adaptations, Theorem 4, the congruence theorem, goes through in the presence of variable binding operators. However, that theorem goes even through in the case where we relax the panth format as follows (see Corollary 4.10 in Ref. [6]). A transition rule $r \in R$ is in *generalized panth format* if it satisfies:

(1) the second argument of each premise of $r$ that has the form $p(t_1, t_2)$ has one of the following forms:
$x$ or $x(t'_1, \ldots, t'_n)$
where each $t'_i$ $(1 \leq i \leq n)$ is a closed term;
(2) the second argument of each premise of $r$ that has the form $\neg p(t_1, t_2)$ is a closed term;
(3) the first argument of the conclusion of $r$ has one of the following forms:
$x$ or $x(u_1, \ldots, u_n)$ or $o(u_1, \ldots, u_n)$
where each $u_i$ $(1 \leq i \leq n)$ has the form $y$ or $x_1, \ldots, x_n \,.\, y(x_1, \ldots, x_n)$;
(4) the variables that occur as a free variable in the second argument of a premise or the first argument of the conclusion are mutually distinct.

This transition rule format is a minor improvement of the one from Definition 4.3 in Ref. [6].

**Example 7** *We consider again the recursion operator* $\mu$ *from Example 6. The transition rules anticipated for this operator include for each action* $\alpha$ *the following transition rule concerning a transition predicate* $\xrightarrow{\alpha}$ *capturing "is capable of first performing action* $\alpha$ *and then proceeding as":*

$$\frac{z(\mu x \,.\, z(x)) \xrightarrow{\alpha} x'}{\mu x \,.\, z(x) \xrightarrow{\alpha} x'} \;.$$

*This transition rule is in generalized panth format.*

## 5 Conservative extensions

Frequently, bisimulation models, and their axiomatizations, seem to extend other bisimulation models, and their axiomatizations, smoothly. If the bisimulation models extend in a certain way, proofs of axiomatic conservativity and completeness can be simplified. This kind of extension is conveyed by the notion of operational conservativity. First, we will define what an operational conservative extension of a TSS is and give syntactic criteria to determine whether a TSS is an operational conservative extension of another TSS. After that, we will define what an axiomatic conservative extension of a set of equations is and give results explaining the relationship between operational conservativity, axiomatic conservativity and completeness.

### *5.1 Operational conservativity*

First the notions of sum of signatures and sum of TSSs are introduced.

Let $\Sigma = (S, O)$ and $\Sigma' = (S', O')$ be signatures and $P = (\Sigma, \Pi, R)$ and $P' = (\Sigma', \Pi', R')$ be TSSs. Then the sum of $\Sigma$ and $\Sigma'$, written $\Sigma \oplus \Sigma'$, is the signature $(S \cup S', O \cup O')$ and the sum of $P$ and $P'$, written $P \oplus P'$, is the TSS $(\Sigma \oplus \Sigma', \Pi \cup \Pi', R \cup R')$.

Next we make precise what an operational conservative extension of a TSS is.

Let $P = (\Sigma, \Pi, R)$ and $P' = (\Sigma', \Pi', R')$ be TSSs. Then $P \oplus P'$ is an *operational conservative extension* of $P$ if $P \oplus P'$ is a complete TSS and for all $\phi \in \mathcal{CF}_{\Sigma \oplus \Sigma', \Pi \cup \Pi'}$ such that the first argument of $\phi$ is a $\Sigma$-term we have $\mathcal{TS}_P \models \phi \Leftrightarrow \mathcal{TS}_{P \oplus P'} \models \phi$.

Suppose that $P = (\Sigma, \Pi, R)$ and $P' = (\Sigma', \Pi', R')$ are TSSs and that $P \oplus P'$ is complete. It is straightforward to check that $P \oplus P'$ is an operational conservative extension of $P$ iff for all $N \subseteq \mathcal{CF}^{-}_{\Sigma\oplus\Sigma',\Pi\cup\Pi'}$ and for all $\psi \in \mathcal{CF}^{+}_{\Sigma\oplus\Sigma',\Pi\cup\Pi'}$ such that the first argument of $\psi$ is a $\Sigma$-term we have $P \vdash \frac{N}{\psi} \Leftrightarrow P \oplus P' \vdash \frac{N}{\psi}$. This characterization of operational conservativity can also be used as its definition in the case where the restriction is dropped that $P \oplus P'$ is complete. This is done in Ref. [11]. However, as explained at the end of Section 3.2, only complete TSSs are considered meaningful in this paper. Besides, it follows immediately from the definition given in this paper that $\leftrightarroweq_P \subseteq \leftrightarroweq_{P\oplus P'}$ if $P \oplus P'$ is an operational conservative extension of $P$, whereas it does not follow immediately from the definition given in Ref. [11].

Next, we will introduce the notion of source-dependency of a transition rule. After that, source-dependency is used in formulating a sufficient condition for a TSS to be an operational conservative extension of another TSS. In the definition of source-dependency and the following theorem, an occurrence of a variable in a (binding) term $t$ is called *firmly free* if the occurrence is free and not in one of the terms $t_1, \ldots, t_n$ of a subterm of the form $x(t_1, \ldots, t_n)$. Besides, a term $t$ is called *firmly fresh* for a signature $\Sigma$, if there is an occurrence of a subterm $t'$ with $t' \notin \mathcal{T}_\Sigma$ in $t$ (possibly $t$ itself) that is not in one of the terms $t_1, \ldots, t_n$ of a subterm of the form $x(t_1, \ldots, t_n)$.

Let $r$ be a transition rule. Then the set of *source-dependent* variables in $r$, written $\mathrm{sd}(r)$, is the smallest set satisfying:

(1) if $x$ occurs firmly free in the first argument of the conclusion of $r$, then $x \in \mathrm{sd}(r)$;
(2) if $p(t, t')$ is a premise of $r$, for all variables $x'$ that occur free in $t$ we have $x' \in \mathrm{sd}(r)$, and $y$ occurs firmly free in $t'$, then $y \in \mathrm{sd}(r)$.

The transition rule $r$ is *source-dependent* if for all variables $x$ that occur free in $r$ we have $x \in \mathrm{sd}(r)$.

Notice that, because of the way in which substitution works for terms of the form $x(t_1, \ldots, t_n)$, substitution instances of a term may contain no trace of certain occurrences of subterms of the term. Firmly free occurrences of variables and firmly fresh terms are without this vanishing character. The following theorem does not go through if firmly free is replaced by free or firmly fresh is replaced by fresh anywhere in the definition of source-dependency or in the theorem itself.

**Theorem 8 (Operational conservativity)** *Let $P = (\Sigma, \Pi, R)$ and $P' = (\Sigma', \Pi', R')$ be TSSs such that $P \oplus P'$ is complete. Let, for each $r \in R'$, $\rho(r)$ be $r$ with the premises restricted to those premises of which the first (and possibly only) argument is a $\Sigma$-term. Then $P \oplus P'$ is an operational* *conservative extension of $P$ if the following conditions are satisfied:*

1. *for each $r \in R$, $r$ is source-dependent;*
2. *for each $r \in R'$, either the first argument of the conclusion of $r$ is firmly fresh for $\Sigma$ or there exists a premise $p(t,t')$ or $p(t)$ of $r$ such that:*
   - *(a) $t$ is a $\Sigma$-term;*
   - *(b) for each variable $x$ that occurs free in $t$ we have that $x \in \mathrm{sd}(\rho(r))$;*
   - *(c) either $t'$ is firmly fresh for $\Sigma$ or $p \notin \Pi$.*

**Proof.** We prove a more general result, namely that $P \oplus P'$ satisfies the characterization of operational conservativity given before even in the case where $P \oplus P'$ is not complete. This result is the counterpart of Theorem 3.20 in Ref. [11] in a different setting for variable binding operators. The essential differences are in the details of the structure of terms and the details of substitution. The proof presented in Ref. [11] makes use of three lemmas. It is only through those lemmas that the proof depends on the details of the structure of terms and the details of substitution. Adapted to the notations and terminology used in this paper, the lemmas concerned are as follows:

1. for $t \in \mathcal{T}_{\Sigma \oplus \Sigma'}$, if $t$ is firmly fresh for $\Sigma$, then $\sigma(t) \notin \mathcal{T}_\Sigma$;
2. for $t \in \mathcal{T}_\Sigma$, if $\sigma(x) \in \mathcal{T}_\Sigma$ for all free variables $x$ of $t$, then $\sigma(t) \in \mathcal{T}_\Sigma$;
3. for $t \in \mathcal{T}_\Sigma$, if $\sigma(t) \in \mathcal{T}_\Sigma$, then $\sigma(x) \in \mathcal{T}_\Sigma$ for all free variables $x$ of $t$ with at least one firmly free occurrence.

The proofs of these lemmas are straightforward by structural induction on $t$. □

For completeness, we discuss the subordinate differences between Theorem 3.20 in Ref. [11] and Theorem 8 in this paper. The distinction between formal and actual variables, formal and actual terms, formal and actual substitutions, and formal and actual transition rules is irrelevant in the case of Theorem 8. Besides, Theorem 8 is not as refined – the special position of sorts for which there are no firmly fresh terms is not taken into account – and it does not cover parametrized transition relations. The refinement referred to does not pose any problem, but it will clutter up the definition of source-dependency and the formulation of the operational conservativity theorem. For analogous reasons, parametrized transition relations are not dealt with here (cf. Section 6). In order to grasp the proof presented in Ref. [11], it is useful to know that the following notations and terminology are used. The set of all free variables of $t$ is denoted by $FV(t)$ and the set of all free variables of $t$ that have a firmly free occurrence in $t$ is denoted by $EV(t)$. Firmly fresh terms are simply called fresh terms.

**Example 9** *We consider a fragment of CCS without restriction, relabeling,* *and recursion. CCS assumes a set* $\mathsf{N}$ *of names. The set* $\mathsf{A}$ *of actions is defined by* $\mathsf{A} = \mathsf{N} \cup \overline{\mathsf{N}} \cup \{\tau\}$*, where* $\overline{\mathsf{N}} = \{\overline{a} \mid a \in \mathsf{N}\}$*. Elements* $\overline{a} \in \overline{\mathsf{N}}$ *are called co-names and* $\tau$ *is called the silent step. The signature of the TSS for this fragment of CCS consists of the sort* $\mathsf{P}$ *of processes, the inaction constant* $0\colon \to \mathsf{P}$*, an action prefix operator* $\alpha : \mathsf{P} \to \mathsf{P}$ *for each action* $\alpha \in \mathsf{A}$*, the choice operator* $+ : \mathsf{P} \times \mathsf{P} \to \mathsf{P}$*, and the composition operator* $\mid : \mathsf{P} \times \mathsf{P} \to \mathsf{P}$*. The transition predicate domain consists of a binary transition predicate* $\xrightarrow{\alpha} : \mathsf{P} \times \mathsf{P}$ *for each* $\alpha \in \mathsf{A}$*. The transition rules are the ones given below (*$\alpha \in \mathsf{A}$*,* $a \in \mathsf{N}$*):*

$$\frac{}{\alpha x \xrightarrow{\alpha} x} \qquad \frac{x \xrightarrow{\alpha} x'}{x + y \xrightarrow{\alpha} x'} \qquad \frac{y \xrightarrow{\alpha} y'}{x + y \xrightarrow{\alpha} y'}$$

$$\frac{x \xrightarrow{\alpha} x'}{x \mid y \xrightarrow{\alpha} x' \mid y} \quad \frac{y \xrightarrow{\alpha} y'}{x \mid y \xrightarrow{\alpha} x \mid y'} \quad \frac{x \xrightarrow{a} x',\ y \xrightarrow{\overline{a}} y'}{x \mid y \xrightarrow{\tau} x' \mid y'} \quad \frac{x \xrightarrow{\overline{a}} x',\ y \xrightarrow{a} y'}{x \mid y \xrightarrow{\tau} x' \mid y'} \ .$$

*We can extend this fragment of CCS with the recursion operator* $\mu : \mathsf{P} \,.\, \mathsf{P} \to \mathsf{P}$*. This requires the addition of the transition rules for this operator given in Example 7. This addition satisfies the conditions of Theorem 8. Consequently, the extension with the recursion operator is an operational conservative extension.*

### *5.2 Axiomatic conservativity and completeness*

First we make precise what an axiomatic conservative extension of a set of equations is.

Let $\Sigma$ and $\Sigma'$ be signatures. Let $E$ be a set of equations over $\Sigma$ and $E''$ be a set of equations over $\Sigma \oplus \Sigma'$ such that $E \subseteq E''$. Then $E''$ is an *axiomatic conservative extension* of $E$ (for closed terms) if for all $e \in \mathcal{CE}_\Sigma$ we have $E \vdash e \Leftrightarrow E'' \vdash e$.

The following two theorems suggest how operational conservativity of extensions can be used in proofs of axiomatic conservativity and completeness proofs.

**Theorem 10 (Axiomatic conservativity)** *Let* $P = (\Sigma, \Pi, R)$ *and* $P' = (\Sigma', \Pi', R')$ *be TSSs. Let* $E$ *be a set of equations over* $\Sigma$ *and* $E''$ *be a set of equations over* $\Sigma \oplus \Sigma'$ *such that* $E \subseteq E''$*. Then* $E''$ *is an axiomatic conservative extension of* $E$ *if the following conditions are satisfied:*

*(1)* $P \oplus P'$ *is an operational conservative extension of* $P$*;*
*(2)* $E$ *is a complete axiomatization of* $\mathcal{CT}_\Sigma / {\leftrightarrows}_P$*;*
*(3)* $E''$ *is a sound axiomatization of* $\mathcal{CT}_{\Sigma \oplus \Sigma'} / {\leftrightarrows}_{P \oplus P'}$*.*

**Proof.** Suppose that $E'' \vdash t_1 = t_2$ for $t_1, t_2 \in \mathcal{CT}_\Sigma$. Soundness of $E''$ implies $t_1 \leftrightarroweq_{P \oplus P'} t_2$. Operational conservativity of $P \oplus P'$ implies $t_1 \leftrightarroweq_P t_2$. Completeness of $E$ implies $E \vdash t_1 = t_2$. The other direction is trivial. $\square$

**Theorem 11 (Complete axiomatization)** *Let $P = (\Sigma, \Pi, R)$ and $P' = (\Sigma', \Pi', R')$ be TSSs. Let $E$ be a set of equations over $\Sigma$ and $E''$ be a set of equations over $\Sigma \oplus \Sigma'$ such that $E \subseteq E''$. Then $E''$ is a complete axiomatization of $\mathcal{CT}_{\Sigma \oplus \Sigma'} / \leftrightarroweq_{P \oplus P'}$ if the conditions of Theorem 10 as well as the following condition are satisfied:*

*(4) for each $t \in \mathcal{CT}_{\Sigma \oplus \Sigma'}$, there exists a $t' \in \mathcal{CT}_\Sigma$ such that $E'' \vdash t = t'$.*

**Proof.** Suppose that $t_1 \leftrightarroweq_{P \oplus P'} t_2$ for $t_1, t_2 \in \mathcal{CT}_{\Sigma \oplus \Sigma'}$. Because of condition 4, there exist $u_1, u_2 \in \mathcal{CT}_\Sigma$ such that $E'' \vdash t_1 = u_1$ and $E'' \vdash t_2 = u_2$. Soundness of $E''$ implies $t_1 \leftrightarroweq_{P \oplus P'} u_1$ and $t_2 \leftrightarroweq_{P \oplus P'} u_2$. Together with $t_1 \leftrightarroweq_{P \oplus P'} t_2$, we have $u_1 \leftrightarroweq_{P \oplus P'} u_2$. Operational conservativity of $P \oplus P'$ implies $u_1 \leftrightarroweq_P u_2$. Completeness of $E$ implies $E \vdash u_1 = u_2$. Because $E \subseteq E''$, also $E'' \vdash u_1 = u_2$. Together with $E'' \vdash t_1 = u_1$ and $E'' \vdash t_2 = u_2$, we have $E'' \vdash t_1 = t_2$. $\square$

**Example 12** *We consider again the fragments of CCS of Example 9. Suppose that we have a set $E$ of axioms that is complete for the bisimulation model of the fragment without the recursion operator and a set $E'$ of additional axioms concerning the recursion operator that are all sound for the bisimulation model of the fragment with the recursion operator. The sets in question can easily be found in Ref. [17] and Ref. [10], respectively. It is already known that the TSS for the fragment with the recursion operator is an operational conservative extension of the TSS for the fragment without the recursion operator. Therefore, it follows immediately that $E \cup E'$ is an axiomatic conservative extension of $E$.*

## 6 Given sorts

In various applications of TSSs, it is impractical and unnecessary to provide the terms of certain sorts with an operational semantics because there exists a fully established semantics for them. We will call such sorts *given sorts*. The sort that represents the time domain in versions of process calculi with timing is a typical example of a given sort. In the case of given sorts, a transition system differs in one way from a transition system as defined before: terms of given sorts are identified if they are semantically equivalent. This is formalized as follows. First of all, we introduce $\ddot{\approx}$, the least (sort-respecting) congruence on terms that includes both $\approx$ and the equivalence induced by the semantics for the terms of given sorts. Next, we adapt the definition of the notion of transition system such that a transition system consists of relations on equivalence classes of closed terms with respect to $\ddot{\approx}$. That is, a transition system for signature $\Sigma$ and domain of transition predicates $\Pi$ consists of:

for each $p \in \Pi$, $p\colon s_1 \times \ldots \times s_n$, a relation $p^{\mathcal{TS}} \subseteq \mathcal{CT}_{\Sigma\, s_1}/\ddot{\approx} \times \ldots \times \mathcal{CT}_{\Sigma\, s_n}/\ddot{\approx}$.

For closed transition formulas $\phi$, $\phi$ *holds in* $\mathcal{TS}$, still written $\mathcal{TS} \models \phi$, is now defined as follows:

(1) $\mathcal{TS} \models p(t_1, \ldots, t_n)$ if $([t_1]_{\ddot{\approx}}, \ldots, [t_n]_{\ddot{\approx}}) \in p^{\mathcal{TS}}$,
(2) $\mathcal{TS} \models \neg p(t_1, \ldots, t_n)$ if $([t_1]_{\ddot{\approx}}, \ldots, [t_n]_{\ddot{\approx}}) \notin p^{\mathcal{TS}}$.

The definitions of the notions of TSS, model of a TSS, well-supported model of a TSS, complete TSS, TSS in panth format, and bisimulation equivalence in a TSS do not have to be adapted.

With these adaptations, we still have the following result. If $P = (\Sigma, \Pi, R)$ is a complete TSS in generalized panth format, then $\leftrightarroweq_P$ is a congruence on the algebra of closed terms over $\Sigma$. However, this result goes through in the case where we relax the generalized panth format as follows. A transition rule $r \in R$ is in *relaxed generalized panth format* if it satisfies the restrictions for the generalized panth format with restriction 3 modified as follows:

(3) the first argument of the conclusion of $r$ has one of the following forms:
$$x \quad \text{or} \quad x(u_1, \ldots, u_n) \quad \text{or} \quad o(u_1, \ldots, u_n)$$
where each $u_i$ $(1 \leq i \leq n)$ has the form $y$ or $x_1, \ldots, x_n \,.\, y(x_1, \ldots, x_n)$ or is a term of a given sort.

This modification permits, for each given sort $s$, that a term of sort $s$ is used where the original generalized panth format only permits that a variable of sort $s$ is used.

Distinguishing given sorts does not only make it possible to relax the panth format. It also allows for TSSs with transition predicates parametrized by closed terms of given sorts. We can relax the restriction that a transition predicate $p$ is a predicate $p\colon s_1 \times \ldots \times s_n$ with $1 \leq n \leq 2$ to the restriction that a transition predicate $p$ is a predicate $p : s_1 \times \ldots \times s_n$ with at most two sorts among $s_1, \ldots, s_n$ that are not given sorts. Suppose that $p$ is a parametrized transition predicate $p : s_1 \times \ldots \times s_n$ and $i_1, \ldots, i_k$ $(n - 2 \leq k \leq n - 1)$ are the indices of the given sorts in increasing order. We can take a fresh predicate $p_{T_1,\ldots,T_k}$ for each equivalence class $T_1$ of closed terms of sort $s_{i_1}, \ldots,$ equivalence class $T_k$ of closed terms of sort $s_{i_k}$. It is easy to see that carrying on in this way, we can reduce any TSS with parametrized transition predicates to a TSS without them, while preserving bisimulation equivalence. Consequently, the congruence and operational conservativity results given in this paper can be generalized to cover transition predicates parametrized by closed terms of given sorts.

**Example 13** *We consider the signature* $\Sigma_{\mathsf{T}} = (\{\mathsf{T}\}, \{\mathsf{0_t}, \mathsf{s_t}\}) \oplus \Sigma_{\mathbb{N}}$*, with* $\mathsf{0_t}\colon \to \mathsf{T}$ *and* $\mathsf{s_t} : \mathsf{T} \to \mathsf{T}$*, where* $\Sigma_{\mathbb{N}} = (\{\mathbb{N}\}, \{0, 1, +, \cdot\})$ *is the signature of the theory of natural numbers. We declare* $\mathbb{N}$ *to be a given sort. We also consider the transition predicate domain* $\Pi_{\mathsf{T}} = \{\mapsto\}$ *with* $\mapsto : \mathsf{T} \times \mathbb{N} \times \mathsf{T}$*. So* $\mapsto$ *is a transition predicate parametrized by closed terms of the given sort* $\mathbb{N}$*. The signature* $\Sigma_{\mathsf{T}}$ *introduces terms intended to be used as expressions for timers. The idea is that the term* $\mathsf{0_t}$ *represents a timer that expires immediately and that the term* $\mathsf{s_t}(t)$*, where* $t \in \mathcal{CT}_{\Sigma_{\mathsf{T}}}$*, represents a timer that expires one time unit later than the timer represented by* $t$*. After idling for one time unit, the timer represented by* $\mathsf{s_t}(t)$ *behaves like the timer represented by* $t$*. The operational behaviour of timers is modeled by the TSS* $\mathrm{P}_{\mathsf{T}} = (\Sigma_{\mathsf{T}}, \Pi_{\mathsf{T}}, \mathrm{R}_{\mathsf{T}})$*, where* $\mathrm{R}_{\mathsf{T}}$ *consists of the following transition rules:*

$$\frac{}{\mathsf{0_t} \overset{0}{\mapsto} \mathsf{0_t}} \qquad \frac{x \overset{n}{\mapsto} y}{\mathsf{s_t}(x) \overset{n+1}{\longmapsto} y} \qquad \frac{x \overset{n}{\mapsto} y}{\mathsf{s_t}(x) \overset{n}{\mapsto} \mathsf{s_t}(y)} \;.$$

*Two examples of closed transition formulas over* $\Sigma_{\mathsf{T}}$ *and* $\Pi_{\mathsf{T}}$ *are*

$$\mathsf{s_t}(\mathsf{0_t}) \overset{1}{\mapsto} \mathsf{0_t} \; \textit{and} \; \mathsf{s_t}(\mathsf{0_t}) \overset{0+1}{\longmapsto} \mathsf{0_t} \;.$$

*Both transition formulas refer to the same transition because* $1$ *and* $0 + 1$ *are semantically equivalent.*

## 7 Concluding remarks

I have been able to reformulate an operational conservativity theorem from Ref. [11] in my preferred setting to deal with variable binding operators, viz. the setting introduced in Ref. [6]. This is not a very deep result. Yet, for several reasons, it is surely worth making mention of. In the first place, different from the setting to deal with variable binding operators introduced in Ref. [11], the one introduced in Ref. [6] does not require to make a distinction between two kinds of variables, terms, substitutions, etc. Such a distinction hinders smooth generalizations of definitions and results concerning TSSs without support for variable binding operators. Secondly, the question whether variable binding operators fit in with the basic concepts, constructions, and results concerning algebras has only been answered affirmative in the case of the setting introduced in Ref. [6] (see e.g. Ref. [12]). Because my need to extend the approach to structural operational semantics developed in Refs. [1–5] to deal with variable binding operators stems from the work on process algebra with timing presented in Refs. [18,19], I found the second point very important.

Furthermore, I have given an alternative explanation of the meaning of TSSs with negative premises. In my opinion, this explanation conveys a more intuitive understanding of the meaning of TSSs with negative premises than previous explanations. In general, those explanations put the emphasis on rather artificial notions, such as the notion of a three-valued stable model, which are difficult to grasp (see e.g. Ref. [5,6,11]). In the alternative explanation given in this paper, I have made an attempt to introduce only notions that are relevant to a clear understanding of such issues as the issue whether bisimulation equivalence based on a TSS is also a congruence for that TSS and the issue whether a TSS is an operational conservative extension of another TSS.

## Acknowledgements

The work presented in this paper has been partly carried out while the author was at Eindhoven Embedded Systems Institute, Eindhoven University of Technology.

## References

[1] J.F. Groote, F.W. Vaandrager, Structured operational semantics and bisimulation as a congruence, Information and Computation 100 (1992) 202–260.

[2] J.C.M. Baeten, C. Verhoef, A congruence theorem for structured operational semantics with predicates, in: E. Best (Ed.), CONCUR'93, LNCS 715, Springer-Verlag, 1993, pp. 477–492.

[3] J.F. Groote, Transition system specifications with negative premises, Theoretical Computer Science 118 (1993) 263–299.

[4] C. Verhoef, A congruence theorem for structured operational semantics with predicates and negative premises, Nordic Journal of Computing 2 (1995) 274–302.

[5] R.N. Bol, J.F. Groote, The meaning of negative premises in transition system specifications, Journal of the ACM 43 (1996) 863–914.

[6] C.A. Middelburg, Variable binding operators in transition system specifications, Journal of Logic and Algebraic Programming 47 (2001) 15–45.

[7] J.C.M. Baeten, J.A. Bergstra, Real time process algebra, Formal Aspects of Computing 3 (2) (1991) 142–188.

[8] J.F. Groote, A. Ponse, The syntax and semantics of $\mu$CRL, in: A. Ponse, C. Verhoef, S.F.M. van Vlijmen (Eds.), Algebra of Communicating Processes 1994, Workshop in Computing Series, Springer-Verlag, 1995, pp. 26–62.

[9] C.A.R. Hoare, Communicating Sequential Processes, Prentice-Hall, 1985.

[10] R. Milner, Communication and Concurrency, Prentice-Hall, 1989.

[11] W.J. Fokkink, C. Verhoef, A conservative look at operational semantics with variable binding, Information and Computation 146 (1998) 24–54.

[12] Sun Yong, An algebraic generalization of Frege structures – Binding algebras, Theoretical Computer Science 211 (1999) 189–232.

[13] K.L. Bernstein, A congruence theorem for structured operational semantics of higher-order languages, in: LICS '98, IEEE Computer Science Press, 1998, pp. 153–164.

[14] B. Bloom, CHOCOLATE: Calculi of higher order communication and lambda terms, in: Symposium on Principles of Programming Languages, ACM Press, 1994, pp. 339–347.

[15] R.J. Glabbeek, The meaning of negative premises in transition system specifications II, in: F. Meyer auf der Heide, B. Monien (Eds.), Proceedings 23th ICALP, LNCS 1099, Springer Verlag, 1996, pp. 502–513.

[16] W.J. Fokkink, R.J. Glabbeek, Ntyft/ntyxt rules reduce to ntree rules, Information and Computation 126 (1996) 1–10.

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

[18] J.C.M. Baeten, C.A. Middelburg, Process algebra with timing: Real time and discrete time, in: J.A. Bergstra, A. Ponse, S.A. Smolka (Eds.), Handbook of Process Algebra, Elsevier, 2001, pp. 627–684.

[19] J.C.M. Baeten, C.A. Middelburg, Process Algebra with Timing, Springer Verlag, EATCS Monographs Series, 2002.