



Centrum voor Wiskunde en Informatica
REPORT*RAPPORT*

The tyft/tyxt format reduces to tree rules

W.J. Fokkink

Computer Science/Department of Software Technology

CS-R9367 1993

The Tyft/Tyxt Format Reduces to Tree Rules

Willem Jan Fokkink

CWI

P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

e-mail: wan@cwi.nl

Abstract. Groote and Vaandrager [5] introduced the *tyft/tyxt format* for transition system specifications (TSSs), and established that for each TSS in this format that is *well-founded*, the strong bisimulation it induces is a congruence. In this paper, we construct for each TSS in tyft/tyxt format an equivalent TSS that consists of *tree rules* only. As a corollary we can give an affirmative answer to an open question, namely whether the well-foundedness condition in the Congruence Theorem of [5] can be dropped. These results extend to tyft/tyxt with negative premises and predicates.

1 Introduction

A current method to provide programming and specification languages with an operational semantics is based on the use of transition systems, advocated by Plotkin [7]. Given a set of states, the transitions between these states are obtained inductively from a transition system specification (TSS), containing transition rules. Such a rule, together with a number of transitions, may imply the validity of another transition.

We will consider a specific type of transition systems, in which states are the closed terms generated by a single sorted signature, and transitions are supplied with labels. A great deal of the operational semantics of formal languages in Plotkin style that have been defined over the years, are within the scope of this format.

To distinguish such labelled transition systems, many different equivalences have been defined, the finest of which is the strong bisimulation equivalence of Park [6]. In general, this equivalence is not a congruence, i.e. the equivalence class of a term $f(p_1, \dots, p_m)$ modulo strong bisimulation is not always determined by the equivalence classes of the terms p_i . However, congruence is an essential property, for instance, to fit the equivalence into an axiomatic framework.

Several formats have been developed which ensure that the bisimulation equivalence induced by a TSS in such a format is always a congruence. A first proposal was made by De Simone [8], which was generalised by Bloom, Istrail, and Meyer [1] to the GSOS format. Next, Groote and Vaandrager [5] introduced the tyft/tyxt format, and proved a Congruence Theorem for TSSs in this format that satisfy a well-foundedness criterion.

Up to now, it has been an open question whether or not well-foundedness is an essential ingredient of the Congruence Theorem. The requirement popped

up in the proof, but no counter-example was found to show that the theorem breaks down if well-foundedness were omitted from it. In this paper, we prove that the Congruence Theorem does hold for general TSSs in tyft/tyxt format, i.e. that the requirement of well-foundedness can be omitted.

In fact, we will establish a stronger result, namely that for each TSS in tyft/tyxt format, there is an equivalent TSS consisting of ‘tree rules’ only. A tree rule is a well-founded rule of the form

$$\frac{\{z_i \xrightarrow{a_i} y_i \mid i \in I\}}{f(x_1, \dots, x_m) \xrightarrow{a} t}$$

where the y_i and the x_j are all different variables and are the only variables that occur in the rule, the z_i are variables, f is a function symbol, and t is any term. Using terminology from [5], we can say that a tree rule is a pure and well-founded xyft rule. Since tree rules are well-founded, the reduction of tyft/tyxt format to tree format will immediately imply that the Congruence Theorem concerning the tyft/tyxt format can do without well-foundedness.

Last summer, Rob van Glabbeek independently deduced the same result, which he announced in [3]. His proof is along the same lines as the one presented in this paper.

The major advantage of our main theorem is that it facilitates reasoning about the tyft/tyxt format. Because often it is much easier to prove a theorem for TSSs in tree format than for TSSs in tyft/tyxt format. For example, this is the case with the Congruence Theorem itself. Another striking example consists of Theorems 8.6.6 and 8.9.1 in [5]. With our result at hand, the complicated proof of the second theorem can be skipped, because now the second theorem follows immediately from the first one.

About all TSSs in Plotkin style that have been defined over the years are well-founded. So in this sense, the practical implication of removing well-foundedness from the Congruence Theorem for tyft/tyxt will probably be quite small. But this removal does increase considerably the convenience of applying the tyft/tyxt format, since the user no longer has to recall and check the complicated well-foundedness criterion.

Groote [4] added negative premises to tyft/tyxt, resulting in the ntyft/ntyxt format, and proved that the Congruence Theorem extends to well-founded TSSs in ntyft/ntyxt format. We will show that the reduction of tyft/tyxt rules to tree rules can be lifted to the positive part of rules in ntyft/ntyxt format, but a simple example will learn that this reduction cannot be applied to the negative premises. Again, we will find that the Congruence Theorem concerning the ntyft/ntyxt format can do without well-foundedness.

Finally, Verhoef [9] has defined the panth format, which adds predicates to ntyft/ntyxt, and proved that the Congruence Theorem holds for well-founded TSSs in panth format. We will show that all our results extend to the panth format too.

Acknowledgements. Chris Verhoef is thanked for useful comments, and special thanks go to Rob van Glabbeek and Frits Vaandrager for suggesting some

substantial improvements.

2 Preliminaries

This section contains the basic definitions.

2.1 The Signature

In the sequel we assume a (*single sorted*) *signature* Σ , which consists of a set F of function symbols, together with their arities. Moreover, we assume an infinite set of variables V ,¹ disjoint with F .

The collection $\mathbb{T}(\Sigma)$ of (*open*) *terms* is defined as the least set satisfying:

- each variable from V is in $\mathbb{T}(\Sigma)$,
- if $f \in F$ has arity n , and $t_1, \dots, t_n \in \mathbb{T}(\Sigma)$, then $f(t_1, \dots, t_n) \in \mathbb{T}(\Sigma)$.

A term is called *closed* if it does not contain any variables; the collection of closed terms is denoted by $T(\Sigma)$.

A *substitution* is a mapping $\sigma : V \rightarrow \mathbb{T}(\Sigma)$. Each substitution is extended to a mapping from terms to terms in the standard way.

2.2 Transition System Specifications

In the sequel we assume a set of labels. An expression \xrightarrow{a} with a a label denotes a binary relation between terms, and a pair $t \xrightarrow{a} t'$ is called a *transition*. A transition is called closed if it involves closed terms.

A (*transition*) *rule* is an expression of the form

$$\frac{\{t_i \xrightarrow{a_i} t'_i \mid i \in I\}}{t \xrightarrow{a} t'}$$

with I an index set, the t_i, t'_i, t, t' terms and the a_i, a labels. The expressions $t_i \xrightarrow{a_i} t'_i$ are called the *premises*, and $t \xrightarrow{a} t'$ the *conclusion* of the rule. The notion of substitution extends to transitions and rules as expected.

A *transition system specification* (TSS) is a collection of transition rules. Assume a TSS R , and a rule of the form

$$\frac{\{t_i \xrightarrow{a_i} t'_i \mid i \in I\}}{t \xrightarrow{a} t'}$$

This rule is *provable* from R if there is a *proof* for it in R , which consists of an upwardly branching tree in which all upward paths are finite. Moreover, the nodes of the tree must be labelled by transitions, such that the root has label $t \xrightarrow{a} t'$, and for each node we have:

¹ In several constructions we will assume the existence of ‘fresh’ variables, i.e. variables that have not yet been used in the construction. Some caution is needed to ensure the existence of such fresh variables at any time, but clearly this technical problem is not of a serious nature.

- either the node has a label $t_i \xrightarrow{a_i} t'$ for some $i \in I$, and there are no nodes above it,
- or the node has label $u \xrightarrow{b} u'$, and the nodes directly above it have labels $u_j \xrightarrow{b_j} u'_j$ for $j \in J$, and there is a rule $r \in R$ and a substitution σ such that $\sigma(r)$ equals

$$\frac{\{u_j \xrightarrow{b_j} u'_j \mid j \in J\}}{u \xrightarrow{b} u'}$$

We say that a transition $t \xrightarrow{a} t'$ is *provable* from R , notation $R \vdash t \xrightarrow{a} t'$, if the rule with no premises and conclusion $t \xrightarrow{a} t'$ is provable from R .

Finally, we say that a rule r together with a substitution σ *deduces* a transition $t \xrightarrow{a} t'$ from R if all the premises of r under σ are provable from R , and the conclusion of r under σ results to $t \xrightarrow{a} t'$.

Two TSSs are (*transition*) *equivalent* if exactly the same closed transitions are provable from both.

Lemma 1. *If all rules in S are provable from R , then all rules provable from S are provable from R .*

Lemma 2. *A transition $t \xrightarrow{a} t'$ is provable from R iff there is a rule $r \in R$ that deduces $t \xrightarrow{a} t'$ from R .*

The proofs of these two lemmas are left to the reader.

2.3 Strong Bisimulation

Definition 3. Assume a TSS R . Two closed terms p_0, q_0 are *R -bisimilar*, notation $p_0 \xleftrightarrow{R} q_0$, if there exists a symmetric relation $B \subseteq T(\Sigma) \times T(\Sigma)$ such that

- $p_0 B q_0$,
- if $p B q$ and $R \vdash p \xrightarrow{a} p'$, then $R \vdash q \xrightarrow{a} q'$ and $p' B q'$ for some q' .

2.4 The Tyft/Tyxt Format

In general, bisimulation equivalence is not a *congruence*, i.e. it may be the case that $p_i \xleftrightarrow{R} q_i$ for $i = 1, \dots, n$, but $f(p_1, \dots, p_n)$ and $f(q_1, \dots, q_n)$ are not R -bisimilar. To deal with this problem, Groote and Vaandrager [5] have introduced the *tyft/tyxt format*. If a TSS is in this format, and it satisfies a well-foundedness criterion, then the bisimulation it induces is a congruence.

Definition 4. A transition rule is a *tyft* rule if it is of the form

$$\frac{\{t_i \xrightarrow{a_i} y_i \mid i \in I\}}{f(x_1, \dots, x_m) \xrightarrow{a} t}$$

where the x_k and y_i are all different variables. Similarly, a *tyxt* rule is of the form

$$\frac{\{t_i \xrightarrow{a_i} y_i \mid i \in I\}}{x \xrightarrow{a} t}$$

with x and the y_i all different variables. A TSS is said to be in *tyft/tyxt* format if it consists of tyft and tyxt rules only.

Definition 5. Assume a set $T = \{t_i \xrightarrow{a_i} t' \mid i \in I\}$ of transitions. Its ‘dependency graph’ is a directed graph, with the collection of variables V as vertices, and with as edges the collection

$$\{(x, y) \mid x \text{ and } y \text{ occur in } t_i \text{ and } t' \text{ respectively, for some } i \in I\}.$$

The set T is called *well-founded* if any backward chain of edges in its dependency graph is finite. A transition rule is well-founded if its collection of premises is so, and a TSS is well-founded if all its rules are well-founded.

Example 1. Examples of sets of transitions that are *not* well-founded are

- $\{y \xrightarrow{a} y\}$,
- $\{y_1 \xrightarrow{a} y_2, y_2 \xrightarrow{b} y_1\}$,
- $\{y_{i+1} \xrightarrow{a} y_i \mid i = 0, 1, 2, \dots\}$.

The following Congruence Theorem originates from [5].

Theorem 6. *If a TSS R is well-founded and in tyft/tyxt format, then $\leftrightarrow_{\underline{R}}$ is a congruence.*

In the next section we will see that the requirement of well-foundedness in this theorem can be dropped.

3 Reducing Tyft Rules to Tree Rules

The following lemma, originating from [5], indicates that we can refrain from tyxt rules.

Lemma 7. *For each TSS R in tyft/tyxt format, there is an equivalent TSS in tyft format.*

Proof. Replace each tyxt rule r in R by a collection of tyft rules $\{r_f \mid f \in F\}$, where each r_f is obtained by substituting $f(x_1, \dots, x_n)$ for x in r , with x_1, \dots, x_n variables that do not yet occur in r . The collection of tyft rules R' that is thus obtained is equivalent to R , because clearly for each proof in R' of a certain closed transition there is a proof in R of the same transition, and vice versa. \square

The next lemma will be crucial in the proof of the main theorem.

Lemma 8. For substitutions σ and ρ with $\sigma\rho = \sigma$, there exists a substitution $\bar{\rho}$ such that:

1. $\sigma\bar{\rho} = \sigma$.
2. $\bar{\rho}\rho = \bar{\rho}$.
3. $\bar{\rho}^2 = \bar{\rho}$.
4. If $\rho(x) = x$, then $\bar{\rho}(x) = x$.
5. If $\rho^n(x)$ is a variable for all $n \geq 0$, then $\bar{\rho}(x)$ is a variable.

Proof. Since $\sigma\rho = \sigma$, it follows that $\sigma\rho^n(x) = \sigma(x)$ for $n \geq 1$. So the size of the $\rho^n(x)$ (that is, the number of function symbols they contain) cannot grow beyond the size of $\sigma(x)$. Since $\rho^{n+1}(x)$ is obtained from $\rho^n(x)$ by replacing variables by terms, $\rho^{n+1}(x)$ has at least the size of $\rho^n(x)$. So for n sufficiently great, the terms $\rho^n(x)$ all have the same size. Hence, for such n , $\rho^{n+1}(x)$ is obtained from $\rho^n(x)$ by replacing variables by variables.

Let W be the collection of variables y for which $\rho^n(y)$ is a variable for all $n \geq 0$. Define a binary relation \sim on W by $y \sim z$ if $\rho^m(y) = \rho^n(z)$ for certain m and n . Note that \sim is an equivalence relation. Under $\bar{\rho}$, the elements of each equivalence class $C \subseteq W$ are contracted to one variable from this class as follows:

- If $\rho(y_0) = y_0$ for some $y_0 \in C$, then for all $y \in C$ $\rho^n(y) = y_0$ for some n . This implies $\rho(y) \neq y$ for $y \in C \setminus \{y_0\}$, so y_0 is uniquely determined. We put $\bar{\rho}(y) = y_0$ for $y \in C$.
- If $\rho(y) \neq y$ for all $y \in C$, then we just pick some $y_0 \in C$ and put $\bar{\rho}(y) = y_0$ for $y \in C$.

By definition, for each $y \in W$ there are m and n such that $\rho^m\bar{\rho}(y) = \rho^n(y)$. After applying σ to both sides we get $\sigma\bar{\rho}(y) = \sigma(y)$ for $y \in W$.

Now consider any variable x , for which we define $\bar{\rho}(x)$ as follows. We already noted that for N sufficiently great all variables in $\rho^N(x)$ are in W . We obtain $\bar{\rho}(x)$ by replacing each variable y in $\rho^N(x)$ by the contraction $\bar{\rho}(y)$ that has been selected just now. Clearly $\bar{\rho}(x)$ does not depend on the choice of N .

Since $\sigma\bar{\rho}(y) = \sigma(y)$ for variables y in $\rho^N(x)$, we have $\sigma\bar{\rho}(x) = \sigma\rho^N(x) = \sigma(x)$. And properties 2-5 follow immediately from the construction of $\bar{\rho}$. \square

The following simple example shows that Lemma 8 cannot do without the σ .

Example 2. Assume a function f of arity one, and define $\rho(x) = f(x)$. Suppose that there exists a substitution $\bar{\rho}$ with $\bar{\rho}\rho = \bar{\rho}$. Then

$$\bar{\rho}(x) = \bar{\rho}\rho(x) = \bar{\rho}(f(x)) = f(\bar{\rho}(x))$$

But $f(\bar{\rho}(x))$ has greater size than $\bar{\rho}(x)$, so we have a contradiction.

Definition 9. A tyft rule

$$\frac{\{t_i \xrightarrow{a_i} y_i \mid i \in I\}}{f(x_1, \dots, x_m) \xrightarrow{a} t}$$

is said to be a *xyft* rule if all the t_i are single variables.

We shall now prove that the tyft/tyxt format reduces to xyft rules, which will be an intricate affair. Then a simple argument will learn that the tyft/tyxt format reduces even to tree rules.

Theorem 10. *For each TSS R in tyft/tyxt format, there is an equivalent TSS in xyft format.*

Proof. According to Lemma 7, we may assume R in tyft format. We shall prove R equivalent with the TSS S of xyft rules that are provable from R . According to Lemma 1, transitions provable from S are provable from R . We now show that a closed transition $p \xrightarrow{a} p'$ provable from R is provable from S , using ordinal induction to the length of a shortest proof P for $p \xrightarrow{a} p'$ in R .

First, assume that P has length one. Then apparently there is a rule $r \in R$ with no premises which conclusion results to $p \xrightarrow{a} p'$ under a substitution σ . Since r has no premises, it is a tree rule, and r together with σ deduces $p \xrightarrow{a} p'$ from S . So Lemma 2 implies $S \vdash p \xrightarrow{a} p'$.

Next, suppose that we have proved the case for a proof in R with length smaller than α , and let P have length α . We will construct from P a sequence of proofs Q_n in R for tyft rules r_n that, together with a σ_n , deduce $p \xrightarrow{a} p'$ from S . Each Q_n will be a sub-tree of P , where its nodes are furnished with new labels, which under σ_n yield the original labels of P . The 'limit' of the Q_n will be a proof Q in R for a xyft rule r that deduces $p \xrightarrow{a} p'$ from S .

Let $r_0 \in R$ together with a substitution σ_0 constitute the last step in P . The premises of r_0 under σ_0 are all provable from R by a strict sub-proof of P , so according to the induction hypothesis these transitions are provable from S . Hence, r_0 together with σ_0 deduces $p \xrightarrow{a} p'$ from S . Our proof Q_0 for r_0 in R consists simply of a bottom node labelled by the conclusion of r_0 and upper nodes labelled by the premises of r_0 .

Next, suppose that we have constructed a proof Q_{n-1} in R for a tyft rule r_{n-1} , which together with a σ_{n-1} deduces $p \xrightarrow{a} p'$ from S . Let r_{n-1} be of the form

$$\frac{\{t_i \xrightarrow{a_i} y_i \mid i \in I\}}{f(x_1, \dots, x_m) \xrightarrow{a} t}$$

Let $I_0 \subseteq I$ be the subset of i 's for which the term t_i is not a single variable, but of the form $g_i(u_{i1}, \dots, u_{im_i})$.

The premises of r_{n-1} are labels of upper nodes in Q_{n-1} . Since Q_{n-1} is a sub-tree of P , the premises correspond with nodes in P . For $i \in I_0$, let $s_i \in R$ and τ_i together constitute the step in P to the node which corresponds with the premise $t_i \xrightarrow{a_i} y_i$. Ordinal induction implies that the premises of s_i under τ_i are provable from S . To obtain Q_n , the rules s_i will be imported into Q_{n-1} , so assume that each s_i contains only fresh variables, to avoid name clashes.

Since $\sigma_{n-1}(t_i \xrightarrow{a_i} y_i)$ equals the label of the corresponding node in P , it follows that s_i is of the form

$$\frac{\{t_j \xrightarrow{b_j} y_j \mid j \in J_i\}}{g_i(x_{i1}, \dots, x_{im_i}) \xrightarrow{a_i} v_i}$$

with $\tau_i(x_{ik}) = \sigma_{n-1}(u_{ik})$ and $\tau_i(v_i) = \sigma_{n-1}(y_i)$.

Let σ_n be a substitution equal to σ_{n-1} for variables in Q_{n-1} and equal to the τ_i for variables in the s_i . Moreover, define a substitution ρ_n by:

$$\begin{aligned} \rho_n(x_{ik}) &= u_{ik} && \text{for } i \in I_0 \text{ and } k = 1, \dots, m_i \\ \rho_n(y_i) &= v_i && \text{for } i \in I_0 \\ \rho_n(x) &= x && \text{otherwise} \end{aligned}$$

Note that $\sigma_n \rho_n = \sigma_n$:

$$\begin{aligned} \sigma_n \rho_n(x_{ik}) &= \sigma_n(u_{ik}) = \sigma_{n-1}(u_{ik}) = \tau_i(x_{ik}) = \sigma_n(x_{ik}) \\ \sigma_n \rho_n(y_i) &= \sigma_n(v_i) = \tau_i(v_i) = \sigma_{n-1}(y_i) = \sigma_n(y_i) \end{aligned}$$

So Lemma 8 indicates a substitution $\bar{\rho}_n$ with:

1. $\sigma_n \bar{\rho}_n = \sigma_n$.
2. $\bar{\rho}_n \rho_n = \bar{\rho}_n$.
3. $\bar{\rho}_n^2 = \bar{\rho}_n$.
4. If $\rho_n(x) = x$, then $\bar{\rho}_n(x) = x$.

Since $\bar{\rho}_n \rho_n = \bar{\rho}_n$, it follows that

$$\bar{\rho}_n(g_i(x_{i1}, \dots, x_{im_i}) \xrightarrow{a_i} v_i) = \bar{\rho}_n(g_i(u_{i1}, \dots, u_{im_i}) \xrightarrow{a_i} y_i),$$

and so the rule $\bar{\rho}_n(s_i)$ is of the form

$$\frac{\{\bar{\rho}_n(t_j \xrightarrow{b_j} y_j) \mid j \in J_i\}}{\bar{\rho}_n(t_i \xrightarrow{a_i} y_i)}$$

We adapt Q_{n-1} to a proof Q_n in R as follows:

- For $i \in I_0$, extend Q_{n-1} above the node labelled by $t_i \xrightarrow{a_i} y_i$ with new nodes that have labels $t_j \xrightarrow{b_j} y_j$ for $j \in J_i$.
- Apply $\bar{\rho}_n$ to all labels in the extended version of Q_{n-1} .

Since we have applied $\bar{\rho}_n$ to all the nodes in Q_{n-1} , and since the new steps in Q_n match with the rules $\bar{\rho}_n(s_i)$, it follows that Q_n constitutes a proof in R for some rule r_n .

Due to property 4 of $\bar{\rho}_n$, the rule r_n has conclusion $f(x_1, \dots, x_m) \xrightarrow{a} \bar{\rho}_n(t)$, and premises $\bar{\rho}_n(t_i) \xrightarrow{a_i} y_i$ for $i \in I \setminus I_0$ and $\bar{\rho}_n(t_j) \xrightarrow{b_j} y_j$ for $i \in I_0$ and $j \in J_i$. Hence, r_n is a tyft rule. And since $\sigma_n \bar{\rho}_n = \sigma_n$, it follows that r_n together with σ_n deduces $p \xrightarrow{a} p'$ from S .

Finally, the property $\sigma_n \bar{\rho}_n = \sigma_n$ ensures that σ_n applied to Q_n produces the original labels of P .

In general, r_n is not yet a xyft rule, because although we have removed from r_n all premises of r_{n-1} that do not have a single variable as left-hand side, we may have introduced other premises in r_n that are of this form. Therefore, we repeat the construction above again and again, to obtain sequences $\{Q_n\}_{n=0}^\infty$

and $\{r_n\}_{n=0}^\infty$ and $\{\sigma_n\}_{n=0}^\infty$ and $\{\bar{\rho}_n\}_{n=1}^\infty$, where Q_n is a proof in R for r_n , and r_n together with σ_n deduces $p \xrightarrow{a} p'$ from S .

We construct the limit Q of the proofs Q_n . The tree structure of Q is simply the limit of the trees Q_n ; this is well-defined, because Q_n incorporates Q_{n-1} . However, the labels of the nodes in Q cannot be determined so easily, because the labels in the Q_n are not consistent; if a certain node in Q_{n-1} has label l , then in Q_n it is renamed to $\bar{\rho}_n(l)$. To resolve this complication, we need some extra machinery.

If $\bar{\rho}_n(x) \neq x$, then it follows from $\bar{\rho}_n^2 = \bar{\rho}_n$ that x cannot occur in any term $\bar{\rho}_n(y)$. To obtain Q_n , we have applied $\bar{\rho}_n$ at all its labels, so x does not occur in Q_n . This implies $\bar{\rho}_m(x) = x$ for $m > n$. Hence, we can define a substitution $\bar{\rho}$ as follows:

$$\begin{aligned} \bar{\rho}(x) &= \bar{\rho}_n(x) && \text{if } \bar{\rho}_n(x) \neq x \text{ for some } n \\ \bar{\rho}(x) &= x && \text{otherwise} \end{aligned}$$

Furthermore, let σ be a substitution that equals σ_n for variables in Q_n for all n . Since $\sigma_n \bar{\rho}_n = \sigma_n$ for all n , we have $\sigma \bar{\rho} = \sigma$. So according to Lemma 8 there exists a substitution $\hat{\rho}$ such that:

1. $\sigma \hat{\rho} = \sigma$.
2. $\hat{\rho} \bar{\rho} = \hat{\rho}$.
4. If $\bar{\rho}(x) = x$, then $\hat{\rho}(x) = x$.
5. If $\bar{\rho}^n(x)$ is a variable for $n \geq 0$, then $\hat{\rho}(x)$ is a variable.

Since $\hat{\rho} \bar{\rho} = \hat{\rho}$, it follows that $\hat{\rho} \bar{\rho}_n = \hat{\rho}$ for all n .

Now we can determine the labels of Q . If a node has label l in Q_{n-1} , then in Q we furnish it with the label $\hat{\rho}(l)$. This definition does not depend on the choice of n , because although in Q_n the label is adapted to $\bar{\rho}_n(l)$, the equality $\hat{\rho} \bar{\rho}_n = \hat{\rho}$ ensures that the resulting label in Q would remain the same.

Since Q is a sub-tree of P , each upward path in Q must be finite. And if a step in P matches with a rule $s \in R$ together with a τ , then the same step in Q matches with s together with $\hat{\rho}\tau$. Hence, Q is a proof in R for a rule r .

We check that r is xyft. First, consider a premise of r . It was introduced in some r_k and maintained in all subsequent r_n , so apparently in r_k it had the form $z \xrightarrow{b} y$, and $\bar{\rho}^n(z)$ is a variable for all $n \geq 0$. So according to property 5 $\hat{\rho}(z)$ is a variable. Moreover, $\bar{\rho}_n(y) = y$ for all n , so due to property 4 $\hat{\rho}(y) = y$. Summarizing, the premise in r has the form $\hat{\rho}(z) \xrightarrow{b} y$ with $\hat{\rho}(z)$ a variable. Clearly, the conclusion of r equals $f(x_1, \dots, x_m) \xrightarrow{a} \hat{\rho}(t)$ (where t is the right-hand side of the conclusion of some r_n). So r is xyft.

Since $\sigma \hat{\rho} = \sigma$, the conclusion of r under σ results to $p \xrightarrow{a} p'$, and the premises of r under σ are all provable from S . So according to Lemma 2 $S \vdash p \xrightarrow{a} p'$. \square

Although according to Theorem 10, the tyft/tyxt format reduces to the more restrictive xyft format, this is by no means an argument to abandon the tyft/tyxt format, because a simple TSS in tyft/tyxt format may take a much more complicated form if it is described in xyft format. This is demonstrated by the following example.

Example 3. Assume two functions a, b of arity zero, a function f of arity one, and a label l , and consider the TSS in tyft format that consists of the following two rules:

$$a \xrightarrow{l} a \qquad \frac{a \xrightarrow{l} y}{a \xrightarrow{l} f(y)}$$

To describe this TSS in xyft format, we need an infinite number of rules: $a \xrightarrow{l} f^n(a)$ for $n = 0, 1, 2, \dots$

The auxiliary function symbol b is present to avoid that the TSS can be described by the single rule $a \xrightarrow{l} x$.

Before proving our main theorem, first we define what is a tree rule. The following terminology originates from [5].

Definition 11. Assume a tyft rule of the form

$$\frac{\{t_i \xrightarrow{a_i} y_i \mid i \in I\}}{f(x_1, \dots, x_m) \xrightarrow{a} t}$$

The variables that occur in this rule and are unequal to the x_k and y_i , are called the *free* variables of the rule. A tyft rule is called *pure* if it does not contain any free variables.

Definition 12. A *tree* rule is a pure and well-founded xyft rule.

Theorem 13. For each TSS R in tyft/tyxt format, there is an equivalent TSS in tree format.

Proof. According to Theorem 10, we may assume R in xyft format. We prove R equivalent with the TSS T of tree rules that can be derived from R .

Since all rules in T can be derived from R , it follows from Lemma 1 that each transition provable from T is also provable from R . We check the converse, namely that a closed transition $p \xrightarrow{a} p'$ provable from R is provable from T .

Fix a rule r in R that together with a substitution σ deduces $p \xrightarrow{a} p'$ from R . Let r be of the form

$$\frac{\{z_i \xrightarrow{a_i} y_i \mid i \in I\}}{f(x_1, \dots, x_m) \xrightarrow{a} t}$$

Using ordinal induction, we may assume $T \vdash \sigma(z_i \xrightarrow{a_i} y_i)$ for $i \in I$.

We now construct from r a rule r' in T that deduces $p \xrightarrow{a} p'$ from T , by removing all premises from r that are not well-founded or that contain free variables, and by replacing free variables in t by their values under σ .

1. Remove each 'loop' in the premises of r , either of the form

$$y_1 \xrightarrow{a_2} y_2 \quad y_2 \xrightarrow{a_3} y_3 \quad \cdots \quad y_{n-1} \xrightarrow{a_n} y_n \quad y_n \xrightarrow{a_1} y_1$$

- or of the form $y_{i+1} \xrightarrow{a_i} y_i$ with $i = 0, 1, 2, \dots$

2. Remove all premises $z_i \xrightarrow{a_i} y_i$ from the new rule for which z_i is a free variable.
3. Finally, replace each free variable z in t by $\sigma(z)$.

Clearly, the resulting rule r' is a tree rule, and it is provable from R . Moreover, r' together with σ deduces $p \xrightarrow{a} p'$ from T . \square

Since tree rules are well-founded tyft rules, Theorem 13 implies that the Congruence Theorem for the tyft/tyxt format can do without well-foundedness.

Corollary 14. *If a TSS R is in tyft/tyxt format, then \leftrightarrow_R is a congruence.*

We give an example of a small TSS in xyft format that can only be described by infinitely many rules in tree format.

Example 4. Assume two functions a, b of arity zero, a function f of arity one, and a label l , and consider the TSS in xyft format that consists of the following three rules:

$$a \xrightarrow{l} a \qquad \frac{y \xrightarrow{l} y}{f(y) \xrightarrow{l} f(y)} \qquad a \xrightarrow{l} f(a)$$

To describe this TSS in tree format takes an infinite number of rules: $f^n(a) \xrightarrow{l} f^n(a)$ for $n = 0, 1, 2, \dots$ together with $a \xrightarrow{l} f(a)$.

The extra rule $a \xrightarrow{l} f(a)$ prevents that the TSS can be described by the following two tree rules:

$$a \xrightarrow{l} a \qquad \frac{x \xrightarrow{l} y}{f(x) \xrightarrow{l} f(y)}$$

4 Extensions to Other Formats

4.1 The Ntyft/Ntyxt Format

Groote [4] has extended the tyft/tyxt format by allowing negative premises in the transition rules, which are expressions of the form $t \xrightarrow{a} p'$. A transition $p \xrightarrow{a} p'$ is provable from a TSS R if there exists a rule r in R of the form

$$\frac{\{t_i \xrightarrow{a_i} t'_i \mid i \in I\} \cup \{s_j \xrightarrow{b_j} q \mid j \in J\}}{f(x_1, \dots, x_m) \xrightarrow{a} t}$$

together with a substitution σ , such that

- $R \vdash \sigma(t_i \xrightarrow{a_i} t'_i)$ for $i \in I$,
- $R \not\vdash \sigma(s_j) \xrightarrow{b_j} q$ for all $q \in T(\Sigma)$ and $j \in J$,
- the conclusion of r under σ results to $p \xrightarrow{a} p'$.

Negative premises may give rise to ‘contradictions’, due to rules such as

$$\frac{x \xrightarrow{a} \neg}{x \xrightarrow{a} y}$$

Such contradictions are avoided by considering only TSSs that allow a *stratification*, which ensures that for each rule of the TSS its conclusion is in a sense ‘greater’ than its premises. For a formal definition of this notion we refer to [4].

Groote has deduced a Congruence Theorem for stratifiable, well-founded TSSs that are in the so-called *ntyft/ntyxt* format, which requires a transition rule to have premises of the form $t \xrightarrow{a} y$ and $t \xrightarrow{a} \neg$, and a conclusion of the form $f(x_1, \dots, x_m) \xrightarrow{a} t$ or $x \xrightarrow{a} t$. Moreover, the variables at the right-hand side of the premises and at the left-hand side of the conclusion must all be different.

Without any further complications, we can repeat the construction from the previous section to show that each stratifiable TSS in *ntyft/ntyxt* format is equivalent to a stratifiable, well-founded TSS with rules that have premises of the form $z \xrightarrow{a} y$ and $t \xrightarrow{a} \neg$, and a conclusion of the form $f(x_1, \dots, x_m) \xrightarrow{a} t$. Moreover, the variables at the right-hand side of the premises and at the left-hand side of the conclusion are all different, and are the only variables that occur in the rule. As a corollary, we see that the well-foundedness condition in the Congruence Theorem for the *ntyft/ntyxt* format can be dropped.

Corollary 15. *If a stratifiable TSS R is in *ntyft/ntyxt* format, then \leftrightarrow_R is a congruence.*

We conjecture that in general, terms in negative premises can *not* be reduced to single variables, which is suggested by the following simple example in Basic Process Algebra (BPA). This formalism assumes an alphabet A , representing both a set of labels and a collection of functions with arity zero. Furthermore, it contains the functions $+$ and \cdot , both of arity two, denoting alternative and sequential composition respectively.

Example 5. We add two functions f and g with arity one and a label ok to the signature of BPA, and extend the operational semantics by the following two transition rules. Fix an $a \in A$.

$$\frac{x \xrightarrow{a} y_1 \quad y_1 \xrightarrow{a} y_2}{f(x) \xrightarrow{ok} a} \qquad \frac{f(x) \xrightarrow{ok} \neg}{g(x) \xrightarrow{ok} a}$$

The extended TSS is stratifiable and in *ntyft/ntyxt* format. We conjecture that the premise $f(x) \xrightarrow{ok} \neg$ cannot be reduced.

An obvious attempt to delete the negative premise would be to replace the second rule by the following two rules.

$$\frac{x \xrightarrow{a} \neg}{g(x) \xrightarrow{ok} a} \qquad \frac{x \xrightarrow{a} y \quad y \xrightarrow{a} \neg}{g(x) \xrightarrow{ok} a}$$

However, this adapted TSS is not equivalent with the original one. For example, $g(aa + ab)$ can do an *ok* transition in the new TSS, but not in the old one.

4.2 The Panth Format

Baeten and Verhoef [2] have extended the tyft/tyxt format with predicates, i.e. not only relations $t \xrightarrow{a} t'$, but also predicates such as $t \xrightarrow{a} \surd$ are allowed to occur in transition rules. The definition of strong bisimulation, Definition 3, is adapted accordingly by adding a third condition:

- if pBq , then $p \xrightarrow{a} \surd$ iff $q \xrightarrow{a} \surd$.

Moreover, Verhoef [9] has extended the ntyft/ntyxt format with predicates such as $t \xrightarrow{a} \surd$ and $t \xrightarrow{a} \not\surd$. A Congruence Theorem holds for well-founded TSSs that are in the so-called *panth* format, which requires a transition rule to have premises of the form $t \xrightarrow{a} y$ and $t \xrightarrow{a} \surd$ and $t \xrightarrow{a} \not\surd$ and $t \xrightarrow{a} \not\surd$, and a conclusion of the form $f(x_1, \dots, x_m) \xrightarrow{a} t$ or $x \xrightarrow{a} t$ or $f(x_1, \dots, x_m) \xrightarrow{a} \surd$ or $x \xrightarrow{a} \surd$. Moreover, the variables at the right-hand side of the premises and at the left-hand side of the conclusion must all be different.

Without any further complications, we can repeat the construction from the previous section to show that each stratifiable TSS in panth format is equivalent to a stratifiable, well-founded TSS, of which each rule has premises of the form $z \xrightarrow{a} y$ and $z \xrightarrow{a} \surd$ and $t \xrightarrow{a} \not\surd$ and $t \xrightarrow{a} \not\surd$, and a conclusion of the form $f(x_1, \dots, x_m) \xrightarrow{a} t$ or $f(x_1, \dots, x_m) \xrightarrow{a} \surd$. Furthermore, the variables at the right-hand side of the premises and at the left-hand side of the conclusion are all different, and are the only variables that occur in the rule. As a corollary, we see that the well-foundedness condition in the Congruence Theorem for the panth format can be dropped.

Corollary 16. *If a stratifiable TSS R is in panth format, then \leftrightarrow_R is a congruence.*

References

1. B. Bloom, S. Istrail, and A.R. Meyer. Bisimulation can't be traced: preliminary report. In *Proceedings 15th ACM Symposium on Principles of Programming Languages*, San Diego, California, pages 229–239, 1988.
2. J.C.M. Baeten and C. Verhoef. A congruence theorem for structured operational semantics with predicates. In E. Best, editor, *Proceedings CONCUR 93*, Hildesheim, LNCS 715, pages 477–492. Springer-Verlag, 1993.
3. R.J. van Glabbeek. Full abstraction in structural operational semantics. In M. Nivat, C. Rattray, T. Rus, and G. Scollo, editors, *Proceedings 3rd AMAST Conference*, Twente, The Netherlands, June 1993, Workshops in Computing, pages 77–84. Springer-Verlag, 1993.
4. J.F. Groote. Transition system specifications with negative premises. In J.C.M. Baeten and J.W. Klop, editors, *Proceedings CONCUR 90*, Amsterdam, LNCS 458, pages 332–341. Springer-Verlag, 1990.

5. J.F. Groote and F.W. Vaandrager. Structured operational semantics and bisimulation as a congruence. *Information and Computation*, 100(2):202–260, 1992.
6. D.M.R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *5th GI Conference*, LNCS 104, pages 167–183. Springer-Verlag, 1981.
7. G.D. Plotkin. A structural approach to operational semantics. Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.
8. R. de Simone. Higher-level synchronising devices in MELJE–SCCS. *Theoretical Computer Science*, 37:245–267, 1985.
9. C. Verhoef. A congruence theorem for structured operational semantics with predicates and negative premises. Report CSN-93/18, Eindhoven University of Technology, Eindhoven, 1993.