Math. Struct. in Comp. Science (1992), *vol.* 2, *pp.* 257–275

Processes as terms: non-well-founded models for bisimulation

J. J. M. M. RUTTEN†

CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

Received 30 January 1991; revised 19 February 1992

A compositional semantics characterizing bisimulation equivalence is derived from transition system specifications in the SOS style, satisfying certain syntactic conditions. We use Aczel's nonstandard set theory for solving a recursive equation for a domain of processes. It contains non-well-founded elements modelling possibly infinite behaviour. Semantic interpretations of syntactic operators are obtained by defining the operational semantics for terms consisting of both syntactic and semantic (processes) entities. Finally, we return to standard set theory by observing that a similar, though less general, result can be obtained with the use of complete metric spaces.

1. Introduction

A *labelled transition system* (LTS) is a triple $< S, A, \rightarrow >$, consisting of a set S of states, a set A of transition labels, and a transition relation $\rightarrow \subseteq S \times A \times S$ (Plotkin, 1981). Every LTS induces a (strong) *bisimulation* equivalence on the set of states (Park, 1981). We show how to derive from *transition system specifications* (sets of axioms and rules for defining LTS's) satisfying the syntactic requirements of Groote and Vaandrager (1989), a compositional semantics that characterizes bisimulation in the sense that it assigns the same meaning to bisimilar states. This extends our previous results (Rutten, 1990) on the same topic, since the class of TSS's that can be handled is larger.

First, a (not necessarily compositional) operational semantics M_T is defined for an arbitrary LTS T. It assigns to each state its unfolding under the transition relation. These unfoldings are represented as elements of a class P of commutative, tree-like structures called *processes*, satisfying

$$P = \mathcal{P}(A \times P).$$

(Here $\mathcal{P}(A \times P)$ is the class of all subsets of $A \times P$, the Cartesian product of A and P.)

The process domain P is formally defined in Aczel's theory of non-well-founded sets (Aczel, 1988). This theory is based on the usual set-theoretic axioms, with the axiom of foundation replaced by a strong version of its negation, the *anti-foundation-axiom* (AFA).

† This work was partially supported by ESPRIT BRA (3020) Integration. This paper is an improved version of J.J.M.M. Rutten, Non-well-founded sets and programming language semantics, Proceedings MFPS'91, Pittsburgh, 1991.

Aczel formulates AFA in a very intuitive fashion by viewing sets as graphs and the equality of sets as their being bisimilar (in a sense closely related to the original notion of Park). The existence of non-well-founded sets, like the set a satisfying $a = \{a\}$, is an immediate consequence of AFA. The semantic universe P mentioned above will contain such non-well-founded sets. A simple example is the process p satisfying $p = \{< a, \emptyset >, < b, p >\}$, which represents an infinite binary tree with a choice at every node between doing a and terminating, or doing b and continuing with p again.

There are two useful facts that hold in the above theory. First, the process domain P (which can be viewed itself as an LTS) is strongly extensional, meaning that bisimilar processes are equal. The second fact is called the *Solution Lemma* and is a direct consequence of (in fact equivalent to) AFA. It states the existence of a unique solution for systems of recursive equations. Using these two facts, one can prove that the semantics M_T assigns the same element in P to bisimilar states.

Next, LTS's defined by means of transition system specifications (TSS) are considered. A TSS is a set of rules (and axioms) for defining transitions. These rules follow the syntactic structure of the states $s \in S$, which are now assumed to be terms over some (single-sorted) signature Σ. Thus, we consider LTS's of the form $T = < Terms(\Sigma), A, \rightarrow >$. We show how to use these rules for the definition of semantic interpretations of the syntactic operators $f \in \Sigma$.

The main idea, which also explains the slogan in the title, is to consider an expression like $f(p_1, \ldots, p_k)$ (where k is the arity of f and $p_1, \ldots, p_k$ are processes in P) as a mixed *term*, containing both syntactic (f) and semantic ($p_1, \ldots, p_k$) entities. The semantic interpretation of f is obtained by taking the operational semantics of mixed terms $f(p_1, \ldots, p_k)$, for any k-tuple of processes.

This yields a compositional semantics C_T for T, which is shown to be equal to M_T when the syntactic restrictions on the format of the transition rules introduced in Groote and Vaandrager (1989) are satisfied. We repeat the main result from that paper, namely that the bisimulation equivalence induced by such a TSS is a congruence, and use it to establish $M_T = C_T$.

Finally it is shown that (in standard set theory) *complete metric spaces*, which have often been used in the semantics of programming languages (Nivat, 1979; de Bakker and Zucker, 1982), offer a good alternative to the above non-standard theory. The two main facts of Aczel's theory that are used here, namely the strong extensionality of (domains like) P and the Solution Lemma, also hold, though in a different form, for complete metric spaces. The reader interested in deriving compositional models from TSS's, but not too eager to study non-standard set theory, can skip Section 2 and consult Section 7 for the metric theory that can be used instead.

2. Non-well-founded sets

We shall work in the universe of *non-well-founded* sets as presented by Aczel (1988). (See Barwise and Etchemendy (1988) for a summary. See also Forti and Honsell (1983) for an alternative approach.)

At the basis of Aczel's work lies the conception of sets as graphs. Every set A gives rise

to a graph by taking as nodes the transitive closure of A, and as (directed) edges all pairs x and y with $y \in x$. Conversely, every graph is associated with a unique set.

It is this latter observation that Aczel turns into an axiom, the so-called *anti-foundation-axiom* (AFA). More formally it says: every graph has a unique decoration. Here, a *decoration* for a graph is a function D that assigns to every node of the graph a set such that for each node x

$$D(x) = \{D(y) : y \text{ is a child of } x\}.$$

An immediate consequence of AFA is the existence of non-well-founded sets: consider the one node graph with one edge leading from this node to itself. Since this graph has, by AFA, a decoration, there exists a set a with $a = \{a\}$ (which is, moreover, unique). The set-theoretic framework Aczel works in is determined by the usual axioms of Zermelo-Fraenkel (ZFC), from which the axiom of foundation is omitted (yielding ZFC$^-$), and to which AFA is added. The resulting collection of axioms is denoted by ZFC$^-$/AFA. (In Aczel (1988), the (relative) consistency of ZFC$^-$/AFA is shown.)

We shall make use of two principles that are a direct consequence of AFA: the *solution lemma* and the principle of *strong extensionality*.

The *solution lemma* asserts the existence of a unique solution for a class of *systems* of (recursive) equations. It is formulated as follows. Consider a set X of variables x (formally these variables are called *atoms* or *Urelemente*). A *system* of equations is a collection

$$\{x = a_x\}_{x \in X},$$

where, for every x, the set a_x may contain any of the variables occurring on the left side of any of the equations (a simple example of a system of equations is $\{x = \{x\}\}$). A solution for such a system is a collection π of sets $\{\pi(x)\}_{x \in X}$ such that, for every x,

$$\pi(x) = a_x[\pi(x_1), \pi(x_2), \ldots],$$

where we use the rather informal notation $a_x[\pi(x_1), \pi(x_2), \ldots]$ to denote the set that is obtained from a_x by substituting every variable x_i in a_x by $\pi(x_i)$, for any i. Now we can formulate the following theorem.

Theorem 2.1. (**Solution Lemma**) Every system of equations has a unique solution.

In order to formulate the principle of *strong extensionality*, we first have to introduce the notion of $\in$-*bisimulation*. (Actually it is just called *bisimulation* in Aczel's book. The $\in$ prefix is used to distinguish it from the usual notion of bisimulation, to be defined in the next section.)

Definition 2.2. (**$\in$-bisimulation**) A binary relation R on sets is called an $\in$-bisimulation if for all sets a and b with aRb,

$$\forall x \in a \exists y \in b[xRy]$$

$$\forall x \in b \exists y \in a[xRy].$$

Two sets a and b are called $\in$-*bisimilar* (notation $a \sim_\in b$) if there exists an $\in$-bisimulation relation R with aRb.

The principle of strong extensionality says that whenever two sets are $\in$-bisimilar, they are equal.

Theorem 2.3. (Strong extensionality) For all sets a and b,

$$a \sim_{\in} b \Leftrightarrow a = b.$$

The principle of strong extensionality gives us a way of dealing with equality of non-well-founded sets; e.g., it can be used to prove $a = b$ for $a = \{a\}$ and $b = \{b\}$. (Note that the usual axiom of extensionality does not help here.)

Finally, we mention a theorem stating the existence of fixed-points for a class of recursive domain equations. Again a definition first:

Definition 2.4. A class operator Φ assigns to each class X a class ΦX. A class operator is set-continuous if, for each class X,

$$\Phi X = \bigcup\{\Phi x : x \text{ is a subset of } X\}.$$

Aczel shows that every set-continuous class operator has a smallest and a largest fixed-point. The smallest fixed-point contains all well-founded elements that are present in the largest, which, moreover, may contain non-well-founded sets. We shall use only largest fixed-points, which are characterized in the following theorem:

Theorem 2.5. (Largest fixed-point) Let Φ be a set-continuous class operator. Let

$$J_\Phi = \bigcup\{x : x \text{ is a subset of } \Phi x\}.$$

Then J_Φ is the largest fixed-point of Φ.

Now we can solve recursive domain equations in the usual way by associating with such an equation a class operator. The fixed-points of this operator will satisfy the domain equation.

3. Models for bisimulation

As a starting point for our semantic considerations, we take the notion of *labelled transition system* (LTS) in the style of Plotkin's structural operational semantics (SOS). For every LTS T a semantics M_T will be defined that assigns to every state of T its tree-like unfolding under the transition relation of T. This semantics is characterized by the fact that it assigns the same value to bisimilar states.

First, the notion of labelled transition system is introduced:

Definition 3.6. (LTS) A labelled transition system is a triple $(S, A, \rightarrow)$ consisting of a set of *states* S, a set of *labels* A, and a transition relation $\rightarrow \subseteq S \times A \times S$. We shall write $s \xrightarrow{a} s'$ for $(s, a, s') \in \rightarrow$.

Definition 3.7. (Bisimulation) Let $T = (S, A, \rightarrow)$ be an LTS. A relation $R \subseteq S \times S$ is called a (strong) bisimulation if for all $a \in A$ and $s, t \in S$ with sRt,

$$s \xrightarrow{a} s' \Rightarrow \exists t' \in S\,[t \xrightarrow{a} t' \wedge s'Rt']$$

and

$$t \xrightarrow{a} t' \Rightarrow \exists s' \in S\,[s \xrightarrow{a} s' \wedge s'Rt'].$$

Two states are *bisimilar* in T, notation $s \sim_T t$, if there exists a bisimulation relation R

with sRt. Note that bisimilarity is itself a bisimulation relation (the largest); it is also an equivalence relation on states.

Next we introduce for every LTS $T = (S, A, \rightarrow)$ a semantics M_T, which maps every state $s \in S$ onto its tree-like unfolding under the transition relation $\rightarrow$. It has as a co-domain the set P of *processes*, which is defined as follows (we shall often use the convention of writing $(x, y \in)X$ to introduce a set X with special elements x and y):

Definition 3.8. Let $(p, q \in)P$ be the largest class satisfying

$$P = \mathcal{P}(A \times P),$$

where the set A is the set of labels of T. Formally, P is obtained as the largest fixed-point of the class operator Φ that assigns to every class X the class $\mathcal{P}(A \times X)$ (see also Chapter 8 of Aczel (1988)). It is straightforward to show that Φ is set-continuous. (The interpretation of $\mathcal{P}(A \times X)$ is of importance, however; it should be the class of all *subsets* of $A \times X$. This distinction between sets and classes also explains why there is no problem of cardinality.)

So far, nothing has been said about the set-theoretic nature of the set A of labels, which is used in the definition of P above. One possibility would be to take a specific collection of some well known sets (such as the natural numbers). More generally, one can take labels to be *atoms* or *Urelements*, which are also used in the formulation of the Solution Lemma above. Then the set theory we work in should be extended to deal with these as well. Intuitively, atoms are to be simply seen as given basic building blocks that may be used in the construction of sets. Rather than going into the details of formulating such a set theory here, we refer to Aczel (1988) for some more discussion, and to, for instance, Barwise (1975).

Interestingly, the domain P can itself be viewed as a transition system as follows: let $T_P = < P, A, \rightarrow_P >$, where $\rightarrow_P$ is given, for all $p, q \in P$ and $a \in A$, by

$$p \xrightarrow{a}_P q \Leftrightarrow < a, q > \in p.$$

Let $\sim_P$ indicate the bisimilarity relation on P induced by T_P.

The next theorem follows from the principle of strong extensionality.

Theorem 3.9. The domain P is strongly extensional. That is, for all $p, q \in P$,

$$p \sim_P q \Rightarrow p = q.$$

Proof. We show, for all $p, q \in P$,

$$p \sim_P q \Rightarrow p \sim_\in q.$$

This, with the principle of strong extensionality, leads to the theorem. Let $p \sim_P q$. Then there exists a bisimulation $R \subseteq P \times P$ with pRq. We define

$$S = S_1 \cup S_2 \cup S_3 \cup S_4 \cup S_5$$

with

$$\begin{aligned} S_1 &= R \\ S_2 &= \{(< a, p >, < a, q >) : pRq, a \in A\} \end{aligned}$$

$$\begin{aligned} S_3 &= \{(\{a\},\{a\}) : a \in A\} \\ S_4 &= \{(\{a,p\},\{a,q\}) : pRq, a \in A\} \\ S_5 &= \{(a,a) : a \in A\}. \end{aligned}$$

Clearly pSq. We have to prove that S is an $\in$-bisimulation. (Then $p \sim_\in q$.) Consider two sets c and d with cSd and let $c' \in c$. There should exist $d' \in d$ with $c'Sd'$.

First, suppose cS_1d. Then $c = r$ and $d = t$, for some $r,t \in P$, and c' is of the form $< a,r' >$, for some $a \in A$ and $r' \in P$. Since rRt, there exists $t' \in P$ such that $< a,t' > \in t$ and $r'Rt'$. Thus, $< a,r' > S_2 < a,t' >$ and hence $< a,r' > S < a,t' >$.

Second, suppose cS_2d. Then $c =< a,r >$ and $d =< a,t >$, for some $a \in A$ and $r,t \in P$ with rRt. Note that, as usual, $< x,y >$ is shorthand for $\{\{x\},\{x,y\}\}$. Thus, c' is either $\{a\}$ or $\{a,r\}$. We can take $d' \in d$ to be either $\{a\}$ or $\{a,t\}$, satisfying $c'S_3d'$ or $c'S_4d'$, respectively.

The other cases are similar. Note that in the last case cS_5d, the fact that the set A is assumed to consist of atoms is used: elements $a \in A$ do not have a set structure. □

For every LTS T a model $M_T : S \to P$ is defined as follows:

Definition 3.10. Let $T = (S,A,\to)$ be an LTS. We define a model (operational semantics) $M_T : S \to P$ by, for any $s \in S$,

$$M_T(s) = \{< a, M_T(s') > : s \xrightarrow{a} s'\}.$$

We can justify this recursive definition by an application of the Solution Lemma: consider the system of equations

$$\{x_s = \{< a, x_{s'} > : s \xrightarrow{a} s'\}\}_{s \in S},$$

where $\{x_s\}_{s \in S}$ is a collection of variables, one for each state $s \in S$. Let π be a unique solution for this system. Then we can define

$$M_T(s) = \pi(x_s).$$

The fact that $\pi(x_s)$ is in P is a direct consequence of the fact that P is the *largest* class satisfying the equation used for its definition.

This model is of interest because it assigns the same meaning to states that are bisimilar. We prove this next. (See also van Glabbeek and Rutten (1989) and Abramsky (1991) for similar results using complete metric spaces and complete partially ordered spaces, respectively.)

Theorem 3.11. Let $\sim_T \subseteq S \times S$ denote the bisimilarity relation induced by the labelled transition system $T = (S,A,\to)$. Then

$$\forall s,t \in S [s \sim_T t \Leftrightarrow M_T(s) = M_T(t)].$$

Proof. Let $s,t \in S$.

($\Leftarrow$): Suppose $M_T(s) = M_T(t)$. We define a relation $R_1 \subseteq S \times S$ by

$$R_1 = \{(s',t') : M_T(s') = M_T(t')\}.$$

From the definition of M_T it is straightforward that R_1 is a bisimulation relation on S and that sRt. Thus $s \sim_T t$.

($\Rightarrow$): Consider s and t with $s \sim_T t$. According to Theorem 3.9, it is sufficient to show that $M_T(s)$ and $M_T(t)$ are bisimilar. Let $R_2 \subseteq P \times P$ be defined by

$$R_2 = \{(M_T(s'), M_T(t')) : s' \sim_T t'\}.$$

It is not difficult to show that R_2 is a bisimulation. Hence $M_T(s) \sim_P M_T(t)$. □

4. Transition system specifications

Often LTS's have some structure. In particular, the set of states is given as the set of terms over some signature, and the transition relation is defined by means of axioms and rules following the syntactic structure of the states.

Therefore, for the rest of this paper let Σ be a single-sorted signature. Function symbols $f \in \Sigma$ come with an arity $a(f)$, which is left implicit. Let $(x \in) Var$ be a set of variables. The set of terms $(s, t \in) T(\Sigma, Var)$ possibly containing variables is defined as usual. The set of terms without variables, called closed terms, is indicated by $Terms(\Sigma)$ or simply $Terms$. For any term $t \in T(\Sigma, Var)$, the set of variables occurring in t is denoted by $Var(t)$.

The set of substitutions $(\sigma \in) Subst$ consists of all partial functions from Var to $Terms$. Substitutions are extended to terms in the usual way. The application of a substitution σ to a term t is denoted by $t(\sigma)$.

Definition 4.12. A transition system specification (TSS) for Σ is a collection $(R \in) \mathscr{R}$ of rules of the form

$$\frac{\{t_i \xrightarrow{a_i} t_i' : i \in I\}}{t \xrightarrow{a} t'},$$

where I is some set of indices, $a_i, a \in A$ (the set of action labels), $t_i, t_i', t, t' \in T(\Sigma, Var)$. The elements $t_i \xrightarrow{a_i} t_i'$ are called premises, and $t \xrightarrow{a} t'$ is called the conclusion of this rule. If $I = \emptyset$, the rule is called an axiom. The set of all variables occurring in R is denoted by $Var(R)$.

Definition 4.13. An expression of the form $s \xrightarrow{a} s'$ with $s, s' \in Terms$ is called a transition. Note that transitions do not contain variables (unlike terms used in rules). A proof tree for $s \xrightarrow{a} s'$ from a TSS $\mathscr{R}$ is defined as follows (let $R \in \mathscr{R}$ be as above):

1. If $I = \emptyset$ and if there is a substitution σ with domain $Var(R)$ such that $s = t(\sigma)$ and $s' = t'(\sigma)$, then $s \xrightarrow{a} s'$ is a proof tree for $s \xrightarrow{a} s'$.
2. If there is a substitution σ with domain $Var(R)$ such that $s = t(\sigma)$ and $s' = t'(\sigma)$, and if there exist proof trees τ_i for the transitions $t_i(\sigma) \xrightarrow{a_i} t_i'(\sigma)$, for all $i \in I$, then a proof tree for $s \xrightarrow{a} s'$ is obtained by forming the tree with root $s \xrightarrow{a} s'$ and as immediate subtrees the proof trees τ_i.
3. Clauses 1 and 2 define all proof trees.

Note that proof trees always have a finite height. If a proof tree for $s \xrightarrow{a} s'$ from $\mathscr{R}$ exists, then we write $\mathscr{R} \vdash s \xrightarrow{a} s'$. Every TSS $\mathscr{R}$ induces an LTS $< Terms, A, \rightarrow >$ with

$$\rightarrow = \{(s, a, s') : \mathscr{R} \vdash s \xrightarrow{a} s'\}.$$

Example 4.14. Consider the signature Σ_B

$$\Sigma_B = Act \cup \{\epsilon, \delta\} \cup RecVar \cup \{\cdot, +\}$$

consisting of a set Act of atomic actions, two special symbols ϵ and δ, a set $(X \in) RecVar$ of recursion variables, and two operators $\cdot$ and $+$. (The signature Σ_B is called Basic Process Algebra with ϵ and δ (see, for example, Groote and Vaandrager (1989)), here extended with recursion.)

All elements are constants except for the last two, which are binary operators. The interpretation of $\cdot$, for concatenation, and $+$, for nondeterministic choice, is as usual. Let $A = Act \cup \{\surd\}$. The label $\surd$ is used to indicate termination. A TSS $\mathscr{R}_B$ for Σ_B is defined as follows. It consists, for every $a \in A$, of the following axioms and rules:

1. $$a \xrightarrow{a} \epsilon$$

2. $$\epsilon \xrightarrow{\surd} \delta$$

3. We assume the presence of a collection $\{s_X : s_X \in Terms(\Sigma_B) \wedge X \in RecVar\}$ of declarations, giving the body s_X for each recursion variable X. Then we have for all $X \in RecVar$ the following rule:

$$\frac{s_X \xrightarrow{a} y}{X \xrightarrow{a} y}$$

4. $$\frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'}$$

5. $$\frac{x \xrightarrow{a} x'}{y + x \xrightarrow{a} x'}$$

6. For all $a \neq \surd$,

$$\frac{x \xrightarrow{a} x'}{x \cdot y \xrightarrow{a} x' \cdot y}$$

7. $$\frac{x \xrightarrow{\surd} x' \quad y \xrightarrow{a} y'}{x \cdot y \xrightarrow{a} y'}$$

Let $T = <Terms(\Sigma_B), A, \rightarrow>$ be the LTS induced by $\mathscr{R}_B$. Consider the function M_T given by Definition 3.10. It yields for $(a \cdot b) + c \in Terms\ (\Sigma_B)$ the following process:

$$M_T((a \cdot b) + c) = \{< a, \{< b, \{< \surd, \emptyset >\} >\} >,\ < c, \{< \surd, \emptyset >\} >\}.$$

For another example, let $X \in RecVar$ and let $s_X = a + (X \cdot b)$, then

$$M_T(X) = \{ab^n\surd : n \geq 0\},$$

where $ab^n\surd$ is used as an abbreviation for

$$\{< a, \{< b, \cdots \{< b, \{< \surd, \emptyset >\} >\} \cdots >\} >\}$$

with n occurrences of b. Note that the declarations of recursion variables X need not be guarded in X (in the standard sense that, for example, $a \cdot X$ is, and $X \cdot a$ is not, guarded in X).

5. Processes as terms

For the rest of the paper let $\mathscr{R}$ be a TSS for Σ, let $T = < Terms, A, \rightarrow_T >$ be the LTS induced by $\mathscr{R}$ and let P be as in Definition 3.8. Our aim is to develop a systematic way of associating with every syntactic operator $f \in \Sigma$ of arity k a function $\tilde{f} : P^k \rightarrow P$, which can be seen as its semantic interpretation, and to use these functions for defining a compositional semantics for T.

To this end, we shall extend the signature Σ with the collection of processes P. Next, the TSS $\mathscr{R}$ will be extended in order to define also transitions for terms containing both function symbols from Σ and processes. In the next section, this extended signature will be used to construct $\tilde{f}$. The key idea will be to consider an expression like $f(p_1, \ldots, p_k)$ as a term – albeit a mixed one in the sense that it consists both of a syntactic entity f and semantic entities $p_1, \ldots, p_k$. Then its meaning will be directly given by the extended transition system.

Definition 5.15. Let $\Sigma_P = \Sigma \cup P$. All processes $p \in P$ have arity 0 (and hence are to be considered as constants of the extended signature). The set of closed terms over Σ_P is denoted by $(\rho \in) PTerms$, the collection of process (or mixed) terms. Further, the TSS $\mathscr{R}$ is extended to a TSS $\mathscr{R}_P$ for Σ_P by putting

$$\mathscr{R}_P = \mathscr{R} \cup \{p \xrightarrow{a} q : < a, q > \in p\}.$$

The LTS induced by $\mathscr{R}_P$ is indicated by

$$T_P = < PTerms, A, \rightarrow_{PT} > .$$

The bisimilarity equivalence induced by T_P is denoted by $\sim_{PT}$.

Example 4.14. (continued) Suppose $p, q \in P$ with $p \xrightarrow{b}_{PT} q$. Then $(a \cdot p) + c$ is an example of a process term. One of its possible transition sequences is

$$(a \cdot p) + c \xrightarrow{a}_{PT} \epsilon \cdot p \xrightarrow{b}_{PT} q.$$

Terms in *Terms* are assigned a meaning by the semantic function $M_T : Terms \rightarrow P$ of Definition 3.10. Now, mixed terms in *PTerms* have by the same definition an operational model $M_{PT} : PTerms \rightarrow P$ satisfying, for all $\rho \in PTerms$,

$$M_{PT}(\rho) = \{< a, M_{PT}(\rho') > : \rho \xrightarrow{a}_{PT} \rho'\}.$$

Given the fact that $Terms \subseteq PTerms$, one would expect, for all $s \in Terms$,

$$M_T(s) = M_{PT}(s).$$

This does not hold in general, however, as the following example shows.

Example 5.16. Let $\Sigma = \{a\}$. Suppose $\mathscr{R}$ contains only one (admittedly rather silly) rule,

$$\frac{y \xrightarrow{a} y}{a \xrightarrow{a} a}.$$

Obviously, $M_T(a) = \emptyset$. In contrast, as a direct consequence of the fact that P is the largest fixed point of its defining equation, it contains a process p with $p = \{< a, p >\}$. Hence $M_{PT}(a) = p$.

Intuitively, the problem is that in the above rule no relation exists between the left side of the conclusion and the variables occurring in the premisses. Therefore, only rules of a restricted format are considered. Next it will be shown that for such rules the above equality of M_T and M_{PT} on elements of *Terms* does indeed hold.

Definition 5.17. Let R be a rule of the form

$$\frac{\{t_i \xrightarrow{a_i} t_i' : i \in I\}}{t \xrightarrow{a} t'}.$$

Let $Var(R)$ be the set of variables occurring in R. The set $\Pi(R)$ of so-called produced variables is defined as follows. It is the union $\Pi(R) = \bigcup_n \Pi_n$ of a sequence of sets $(\Pi_n)_n$, defined inductively by

$$\Pi_0 = Var(t), \quad \Pi_{n+1} = \bigcup \{Var(t_i') : Var(t_i) \subseteq \Pi_n\}.$$

Now the rule R is called inductive if $Var(R) = \Pi(R)$. A TSS $\mathscr{R}$ is inductive if all rules in $\mathscr{R}$ are.

The reason why such rules are called inductive is that their set of premisses can be inductively structured as follows. For every $i \in I$, an induction coefficient $\omega(i)$ is defined as the smallest natural number k such that

$$Var(t_i') \subseteq \Pi_k.$$

In Groote and Vaandrager (1989), the syntactic format of TSS's is extensively studied. (See also the next section of this paper.) It is not very difficult to prove that a rule is *inductive* if and only if it is *pure* in the sense of Groote and Vaandrager (1989).

Theorem 5.18. If $\mathscr{R}$ is inductive, then, for all $s \in$ Terms,

$$M_T(s) = M_{PT}(s).$$

Proof. We shall prove

1. $\rightarrow_T \subseteq \rightarrow_{PT}$
2. For all $s \in Terms$, $\rho \in PTerms$, if $s \xrightarrow{a}_{PT} \rho$, then $\rho \in Terms$ and $s \xrightarrow{a}_T \rho$.

From Clauses 1 and 2, it immediately follows that $B \subseteq P \times P$, given by

$$B = \{(M_T(s), M_{PT}(s)) : s \in Terms\},$$

is a bisimulation on P. Then the theorem follows by the strong extensionality of P.
Clause 1 is trivial, so let us proceed with Clause 2. Consider a proof tree for $s \xrightarrow{a}_{PT} \rho$. The proof is by induction on the height of the proof tree.
(1): Suppose the height of the proof tree is 1. Then it is the instantiation of an axiom in

$\mathscr{R}_{PT}$. Since $s \in Terms$, it must be an axiom in $\mathscr{R}$, say $t \xrightarrow{a} t'$: let $\sigma : Var(R) \to PTerms$ be a substitution with $t\sigma = s$ and $t'\sigma = \rho$. Since $\mathscr{R}$ is inductive and since there are no premisses, $Var(t') \subseteq Var(t)$. Then $s \in Terms$ implies $\rho \in Terms$ and, moreover, $s \xrightarrow{a}_T \rho$.
(n+1): Suppose the height of the proof tree for $s \xrightarrow{a}_{PT} \rho$ is $n+1$ and suppose Clause 2 holds for all transitions that have a proof tree of height less than or equal to n. Since there are only axioms in $\mathscr{R}_{PT} \setminus \mathscr{R}_T$, the transition $s \xrightarrow{a}_{PT} \rho$ is the instantiation of the conclusion of a rule $R \in \mathscr{R}_T$, say,

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

with substitution $\sigma : Var(R) \to PTerms$. We show by induction on $\omega(i)$ (defined immediately after Definition 5.17 above) that, for all $i \in I$,

— $t_i'\sigma \in Terms$
— $t_i\sigma \xrightarrow{a_i}_T t_i'\sigma$.

Suppose this holds for all i with $\omega(i) \leq k$ and consider i with $\omega(i) = k+1$. Since

$$Var(t_i) \subseteq \Pi_k = Var(t) \cup \bigcup\{Var(t_i') : \omega(i) \leq k\},$$

this assumption implies for all $x \in \Pi_k$ that $x\sigma \in Terms$. Because $t_i\sigma \xrightarrow{a_i}_{PT} t_i'\sigma$ has a proof tree of height less than or equal to n, it follows by the general induction hypothesis that $t_i'\sigma \in Terms$ and $t_i\sigma \xrightarrow{a_i}_T t_i'\sigma$.
Note that from the above it follows that σ is a substitution of type $\sigma : Var(R) \to Terms$. Thus we can conclude that $s \xrightarrow{a_i}_T \rho$. □

6. Compositional semantics for bisimulation

For every syntactic operator $f \in \Sigma$ a semantic interpretation $\tilde{f}$ can now be defined as follows:

Definition 6.19. Let $\mathscr{R}_P$ be the TSS of the previous section for the extended signature Σ_P. (Recall that we have a model $M_{PT} : PTerms \to P$.) Let $f \in \Sigma$ with arity $k \geq 0$. We define $\tilde{f} : P^k \to P$, for all $p_1, \ldots, p_k$, by

$$\tilde{f}(p_1, \ldots, p_k) = M_{PT}(f(p_1, \ldots, p_k)).$$

(Note that this is well defined, since $f(p_1, \ldots, p_k)$ is a process term in $PTerms$.)

Example 4.14 (continued) According to the above definition, the semantic operators belonging to the signature Σ_B are given by

$$\begin{aligned}
\tilde{a} &= \{< a, \{< \surd, \emptyset >\} >\} \\
\tilde{\varepsilon} &= \{< \surd, \emptyset >\} \\
\tilde{\delta} &= \emptyset \\
p_1 \tilde{+} p_2 &= p_1 \cup p_2 \\
p_1 \tilde{\cdot} p_2 &= \{< a, q_1 \tilde{\cdot} p_2 > : < a, q_1 > \in p_1 \wedge a \neq \surd\} \\
&\cup \{< a, q_2 > : < \surd, q_1 > \in p_1 \wedge < a, q_2 > \in p_2\}.
\end{aligned}$$

As usual, a semantic interpretation of the operators in Σ induces a compositional model for *Terms*.

Definition 6.20. The mapping C_T : *Terms* $\rightarrow P$ is defined inductively as follows. Let $t \in$ *Terms* be of the form $t = f(t_1, \ldots, t_k)$; then

$$C_T(f(t_1, \ldots, t_k)) = \tilde{f}(C_T(t_1), \ldots, C_T(t_k)).$$

The question whether C_T is equal (or correct with respect) to M_T now naturally arises. The following example shows that in general the answer is no.

Let $\Sigma = \{a, \delta, \pi\}$, consisting of two constants and a unary operator, respectively. Let $\mathscr{R}$ be a TSS for Σ given by

$$a \xrightarrow{a} \delta$$

$$\pi(a) \xrightarrow{b} \delta.$$

Then $M_T(\pi(a)) = \{< b, \emptyset >\}$, whereas

$$C_T(\pi(a)) = \tilde{\pi}(\tilde{a}) = \tilde{\pi}(\{< a, \emptyset >\}) = \emptyset.$$

The latter equality follows from the fact that the extended TSS $\mathscr{R}_P$ does not contain any rule for deriving a transition from $\pi(\{< a, \emptyset >\})$.

The point of this example is that the second axiom in $\mathscr{R}$ does not allow replacement of one term by another one that is bisimilar to it. More precisely, on the one hand the terms a and $\{< a, \emptyset >\}$ (over the extended signature Σ_P) are bisimilar: they can both take an a step, to δ and $\emptyset$ respectively, neither can take further steps. On the other hand, however, $\pi(a)$ and $\pi(\{< a, \emptyset >\})$ are not bisimilar: the former can take a b step, whereas the latter cannot. In other words, the bisimilarity equivalence $\sim_{PT}$ induced by $\mathscr{R}_P$ is not a congruence.

(However, note that $\sim_T$, the bisimilarity equivalence induced by $\mathscr{R}$, *is* a congruence.)

In Groote and Vaandrager (1989), a condition on the syntactic format of the rules of TSS's is given that ensures the induced bisimilarity equivalence to be a congruence. It will forbid rules like the second one in the example above. We shall first give the format and theorem of Groote and Vaandrager (1989); next we show how this can be applied here.

Definition 6.21. Let $\mathscr{R}$ be a TSS for Σ and let R be a rule in $\mathscr{R}$. The rule R is in *tyft*-format if it has the following form

$$\frac{\{t_i \xrightarrow{a_i} y_i : i \in I\}}{f(x_1, \ldots, x_k) \xrightarrow{a} t},$$

and it is in *tyxt*-format if it is of the form

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

where $f \in \Sigma$, with arity k; the terms t and t_i, for $i \in I$ are in $T(\Sigma, Var)$; and all of the variables in $\{x_1, \ldots, x_k\} \cup \{y_i : i \in I\}$ in the first case, and all of the variables in $\{x\} \cup \{y_i : i \in I\}$ in the second case, are pairwise distinct variables in *Var*. The TSS $\mathscr{R}$ is in *tyft*/*tyxt*-format (terminology of Groote and Vaandrager (1989)) if all its rules

are either in *tyft* or in *tyxt* format. It is in BSOS (bisimulation structured operational semantics, our terminology) format if it is both inductive and in *tyft/tyxt*-format.

As mentioned in the previous section, a rule is inductive if and only if it is *pure* in the sense of Groote and Vaandrager (1989) (pure is defined there as the conjunction of being *well-founded* and *closed*). The notion of well-foundedness is introduced as a technical condition needed to prove the main theorem (which follows in a moment). Here it is natural to require the rules to be inductive (which implies that they are well-founded) since this condition is also needed for Theorem 5.18.

Theorem 6.22. (Groote and Vaandrager, 1989) If $\mathscr{R}$ is in BSOS format, then $\sim_T$, the bisimilarity relation corresponding to the TSS T (induced by $\mathscr{R}$), is a congruence. That is, for all $f \in \Sigma$ of arity k, all $u_1, \ldots, u_k, v_1, \ldots, v_k$ in *Terms*,

$$\forall i \in \{1, \ldots, k\}[u_i \sim_T v_i] \Rightarrow f(u_1, \ldots, u_k) \sim_T f(v_1, \ldots, v_k).$$

In the appendix a proof of the above theorem can be found, which is a slight variation of the one given in Groote and Vaandrager (1989). In that paper, some examples are given (similar to the one above with $\Sigma = \{a, \delta, \pi\}$) showing that none of the conditions of the theorem can be missed.

Now consider a TSS $\mathscr{R}$ that is in BSOS format. We shall prove that the compositional semantics of Definition 6.20 equals the operational semantics M_T of Definition 3.10. We shall use two lemmas:

Lemma 6.23. Let $=_{PT}$ be the equality induced by M_{PT}; i.e., $\rho =_{PT} \rho'$ if and only if $M_{PT}(\rho) = M_{PT}(\rho')$. Then $=_{PT}$ is a congruence.

Proof. By Theorem 3.11, $=_{PT} = \sim_{PT}$. If $\mathscr{R}$ is in BSOS format, then $\mathscr{R}_P$ is also, since all axioms of the form $p \xrightarrow{a} q$ are in BSOS format. Therefore by Theorem 6.22, $\sim_{PT}$ is a congruence. □

Lemma 6.24. For all $\rho \in PTerm$, $\rho =_{PT} M_{PT}(\rho)$.

Proof. We have $\rho \sim_{PT} M_{PT}(\rho)$ as an immediate consequence of the observation that $\{(\rho, M_{PT}(\rho)) : \rho \in PTerm\}$ is a bisimulation relation on *PTerm*. The lemma follows by Theorem 3.11. □

Theorem 6.25. For all $t \in Terms$, $M_T(t) = C_T(t)$.

Proof. Let $t \in Terms$. We use induction on the syntactic structure of t. Let $t = f(t_1, \ldots, t_k)$, for some $f \in \Sigma$ and $t_1, \ldots, t_k \in Terms$, and suppose the theorem holds for all $t_1, \ldots, t_k$. Then

$$\begin{aligned}
M_T(t) &= M_T(f(t_1, \ldots, t_k)) \\
&= (\mathscr{R} \text{ is inductive, Theorem 5.18}) \\
&\quad M_{PT}(f(t_1, \ldots, t_k)) \\
&= (\text{by Lemma 6.23 and Lemma 6.24}) \\
&\quad M_{PT}(f(M_{PT}(t_1), \ldots, M_{PT}(t_k))) \\
&= (\text{Definition 6.19}) \\
&\quad \tilde{f}(M_{PT}(t_1), \ldots, M_{PT}(t_k))
\end{aligned}$$

$$
\begin{aligned}
&= \text{(Theorem 5.18)} \\
&\quad \tilde{f}(M_T(t_1),\ldots,M_T(t_k)) \\
&= \text{(induction hypothesis)} \\
&\quad \tilde{f}(C_T(t_1),\ldots,C_T(t_k)) \\
&= \text{(Definition 6.20)} \\
&\quad C_T(f(t_1,\ldots,t_k)) \\
&= C_T(t).
\end{aligned}
$$

□

Example 4.14 (continued) The following equalities hold:

$$
\begin{aligned}
M_T(a) &= \tilde{a} \\
M_T(\epsilon) &= \tilde{\epsilon} \\
M_T(\delta) &= \tilde{\delta} \\
M_T(s_1 + s_2) &= M_T(s_1)\tilde{+}M_T(s_2) \\
M_T(s_1 \cdot s_2) &= M_T(s_1)\tilde{\cdot}M_T(s_2) \\
M_T(X) &= M_T(s_X).
\end{aligned}
$$

If s_X equals, for instance, $a + (X \cdot b)$ then

$$M_T(X) = \tilde{a}\tilde{+}(M_T(X)\tilde{\cdot}\tilde{b}).$$

7. Discussion: comparison with metric spaces

Is it really necessary to resort to set theory, moreover to a non-standard one, if one wants to develop semantics of programming languages? No, not really, is the expected (and for some also very much welcome) answer. Below we shall show how all of the above constructions can be mimicked in a standard theoretic setting – or rather, without the need for (purely) set-theoretic considerations at all. (In fact, only some of the constructions of Section 3 have to be revised.) At the same time, this alternative approach will be less general (as we will see), and that is where an argument for the use of non-well-founded sets may be found.

The mathematical structures that will be used are complete metric spaces, which seem to be closest to non-well-founded sets (rather than the (even) more standard complete partial orders).

In de Bakker and Zucker (1982), a method is presented for constructing complete metric spaces as solutions of recursive domain equations. It was later generalized in America and Rutten (1989), where a larger family of equations is solved, and where, moreover, uniqueness of many such solutions is proved. Thus, a domain equation very similar to the one used for the construction of P (Definition 3.8) can be solved over a category of complete metric spaces. That is, there exists a complete metric space P_M such that

$$P_M \cong \mathscr{P}_{compact}(A \times P_M).$$

Here $\mathcal{P}_{compact}(X)$ is, for any complete metric space X, the complete metric space consisting of all compact subsets of X. (See, for instance, Engelking (1977) or America and Rutten (1989) for all standard definitions concerning metric spaces.) The symbol $\cong$ expresses that both sides of the equation are isomorphic: there exist distance-preserving mappings

$$i : P_M \to \mathcal{P}_{compact}(A \times P_M)$$

and

$$j : \mathcal{P}_{compact}(A \times P_M) \to P_M$$

such that $j \circ i = id_{P_M}$ and $i \circ j = id_{\mathcal{P}_{compact}(A \times P_M)}$. This equation was first solved in de Bakker and Zucker (1982). In America and Rutten (1989) the uniqueness (up to isomorphism) of the solution is proved.

Note that P is strictly equal to $\mathcal{P}(A \times P)$, whereas here the equation is solved up to isomorphism.

Also P_M can be turned into an LTS by defining, for all $p, q \in P_M$ and $a \in A$,

$$p \xrightarrow{a}_{P_M} q \Leftrightarrow < a, q > \in i(p).$$

Moreover, P_M is again strongly extensional (as pointed out in van Glabbeek and Rutten (1989)). That is, if two processes are bisimilar, then they are equal. (For P this is proved in Theorem 3.9, using the general principle of strong extensionality, a direct consequence of AFA.) It can be shown by defining $\epsilon = \sup\{(p_1, p_2) : p_1 \sim_{P_M} p_2\}$, and proving $\epsilon \leq \frac{1}{2}\epsilon$, hence $\epsilon = 0$. From this it follows that any two bisimilar processes have distance 0, and thus are equal.

Using P_M instead of P, the operational semantics of Definition 3.10 can be adapted as follows. For an LTS $T = (S, A, \to)$, a model $M_T : S \to P_M$ is given, for any $s \in S$, by

$$M_T(s) = j(\{< a, M_T(s') > : \; s \xrightarrow{a} s'\}).$$

The well-definedness of M_T (that in Definition 3.10 is based on the Solution Lemma) derives from the fact that M_T can be obtained as the unique solution of the following function: let $\Phi : (S \to^1 P_M) \to (S \to^1 P_M)$ be defined by, for any $F \in (S \to^1 P_M)$ and $s \in S$,

$$\Phi(F)(s) = j(\{< a, F(s') > : \; s \xrightarrow{a} s'\}).$$

(Here $(S \to^1 P_M)$ is the set of non-expansive (non-distance-increasing, weakly contractive) mappings from S to P_M.) Now Φ is contracting (strictly distance-decreasing) and hence (by Banach's theorem) has a unique fixed-point M_T: see Rutten (1990).

There is one important requirement that has to be met in the above definition of Φ. The set

$$\{< a, F(s') > : \; s \xrightarrow{a} s'\}$$

should be compact. This imposes an essential restriction on the LTS's to which the above construction can be applied: the transition relation $\to$ of $T = (S, A, \to)$ must be finitely (or, more generally, compactly) branching. That is, for every $s \in S$ the set

$$\{< a, s' > : \; s \xrightarrow{a} s'\}$$

must be finite (or compact). (This can be weakened to *image-finiteness* if one uses the powerset constructor that yields all *closed* subsets in the definition of P_M.)

This difference between the use of non-well-founded sets, where arbitrary LTS's can be considered, and complete metric spaces constitutes a point in favour of the former. Note that, for example, the use of unguarded recursion is ruled out in the metric case: declarations like $s_X = a + (X \cdot b)$ give rise to infinitely branching transition systems.

Once the definitions of Section 3 have been adapted along the lines sketched above, all following definitions and theorems go through as before. In particular, the signature Σ can again be extended with processes from P_M as constants. (Maybe one final technical observation is needed here: if the LTS T is finitely branching, then the extended LTS T_P is compactly branching.) And thus, the definition of a compositional semantics C_T (Definition 6.20) remains unaltered, as well as the proof of the equality of M_T and C_T (Theorem 6.25).

Summarizing, non-well-founded sets and complete metric spaces are very similar in two respects. First, both P and P_M are strongly extensional. Second, both the recursive specifications of M_T (one with P and one with P_M as co-domain) have a unique solution. In the first case, this is due to the Solution Lemma, in the second to Banach's theorem. In both cases, the function specification should satisfy some guardedness condition: systems of equations allow the occurrence of variables to the right of the equality only *within* other sets; a function specification, like that of Φ above, defines a contraction only if its argument (F) occurs at the right of the equality guarded by some distance-decreasing construct. (In order to understand the latter remark, the definitions of the metrics involved should be made explicit. Again we refer to America and Rutten (1989).)

In Rutten (1991) it is shown that the non-well-founded domain of hereditarily finite sets is isomorphic to a subset of the metric completion of the collection of all well-founded hereditarily finite sets (thus giving a metric version of Mislove et al. (1989), where complete partial orders are used). A point of further research is how to relate non-well-founded domains and metric domains in general.

Acknowledgements: Gordon Plotkin's *stressing* of the fact (already present in Aczel (1989)) that the domain of processes is a transition system, eventually led to the idea of processes as terms. Discussions with Samson Abramsky, Jaco de Bakker, Franck van Breugel, Daniele Turi and Jeroen Warmerdam are gratefully acknowledged. The anonymous referees are thanked for their constructive comments.

REFERENCES

Abramsky, S. (1991) A domain equation for bisimulation. *Information and Computation*, **92** 161–218.

Aczel, P. (1988) *Non-well-founded sets*, number 14 in CSLI Lecture Notes.

America, P. and Rutten, J.J.M.M. (1989) Solving reflexive domain equations in a category of complete metric spaces. *Journal of Computer and System Sciences*, **39**(3) 343–375.

Barwise, J. (1975) *Admissible sets and structures*, Springer-Verlag.

Barwise, J. and Etchemendy, J. (1988) *The Liar: An Essay in Truth and Circularity*, Oxford University Press.

de Bakker, J.W. and Zucker, J.I. (1982) Processes and the denotational semantics of concurrency. *Information and Control*, **54** 70–120.

Engelking, R. (1977) *General Topology*, Polish Scientific Publishers.

Forti, M. and Honsell, F. (1983) Set theory with free construction principles. *Annali Scuola Normale Superiore, Pisa*, X(3) 493–522.

Groote, J. F. and Vaandrager, F. (1989) Structured operational semantics and bisimulation as a congruence. In: G. Ausiello, M. Dezani-Ciancaglini, and S. Ronchi Della Rocca, editors, Proceedings 16th ICALP, *Lecture Notes in Computer Science* **372** 423–438. Springer-Verlag. To appear in Information and Computation.

Mislove, M. W., Moss, L. S. and Oles, F. J. (1989) Non-well-founded sets obtained from ideal fixed points. In *Proc. of the Fourth IEEE Symposium on Logic in Computer Science* 263–272. To appear in Information and Computation.

Nivat, M. (1979) Infinite words, infinite trees, infinite computations. In: J. W. de Bakker and J. van Leeuwen, editors, Foundations of Computer Science III, Part 2, *Math. Centre Tracts* **109** 3–52.

Park, D. M. R. (1981) Concurrency and automata on infinite sequences. In: P. Deussen, editor, Proceedings 5th GI conference, *Lecture Notes in Computer Science* **104** 15–32. Springer-Verlag.

Plotkin, G. D. (1981) *A structural approach to operational semantics*, Technical Report DAIMI FN-19, Aarhus University, Computer Science Department.

Rutten, J. J. M. M. (1990) Deriving denotational models for bisimulation from Structured Operational Semantics. In: M. Broy and C. B. Jones, editors, *Programming concepts and methods, proceedings of the IFIP Working Group 2.2/2.3 Working Conference, Sea of Galilee* 155–177. North-Holland.

Rutten, J. J. M. M. (1991) *Hereditarily-finite sets and complete metric spaces.* Technical Report CS-R9148, Centre for Mathematics and Computer Science, Amsterdam.

van Glabbeek, R. J. and Rutten, J. J. M. M. (1989) The processes of De Bakker and Zucker represent bisimulation equivalence classes. In: *J. W. de Bakker, 25 jaar semantiek* 243–246, CWI, Amsterdam.

Appendix

In the proof of the theorem below, only rules that are (inductive and) in *tyft* format are considered. This is sufficient since any TSS containing rules in *tyxt* format can be transformed into a TSS (yielding the same LTS) containing only rules in *tyft* format, as follows. For every rule R in *tyxt* format, and every operator $f \in \Sigma$, a new rule is added by taking R in which x (the left side of the conclusion) is replaced by $f(x_1, \cdots, x_n)$, where n is the arity of f and the new variables have been chosen disjointly from the ones already present in R.

Theorem 6.22. If $\mathscr{R}$ is in BSOS format, then $\sim_T$, the bisimilarity relation corresponding to the TSS T (induced by $\mathscr{R}$), is a congruence.

Proof. Let $W \subseteq Terms \times Terms$ be the smallest congruence relation that contains $\sim_T$. In other words, W is the smallest relation such that

1. $\sim_T \subseteq W$
2. For all $f \in \Sigma$ of arity k, and all $u_1, \ldots, u_k$ and $v_1, \ldots, v_k$ in *Terms*,

$$\text{if } u_i W v_i, \text{ for all } 1 \leq i \leq k, \text{ then } f(u_1, \ldots, u_k) W f(v_1, \ldots, v_k).$$

It is sufficient to show that W is a bisimulation, since this implies $W \subseteq \sim_T$. By symmetry, this is equivalent to proving, for all $u, v \in Terms$ with uWv: if $u \xrightarrow{a} u'$, then there exists $v' \in Terms$ with $v \xrightarrow{a} v'$ and $u'Wv'$.
Let $u, v \in Terms$ with uWv and suppose $u \xrightarrow{a} u'$. We have to show the existence of $v' \in Terms$ with $v \xrightarrow{a} v'$ and $u'Wv'$. If $u \sim_T v$ then this follows from the fact that $\sim_T$ is a bisimulation relation and the fact that $\sim_T \subseteq W$.
Next, suppose that $u = f(\bar{u})$ and $v = f(\bar{v})$ for $f \in \Sigma$ of arity k (with $\bar{u} = u_1, \ldots, u_k$ and $\bar{v} = v_1, \ldots, v_k$); suppose that $u_i W v_i$, for all $1 \leq i \leq k$. Consider a proof tree τ for the transition $f(\bar{u}) \xrightarrow{a} u'$, and let $R \in \mathscr{R}$ be the last rule used in this proof tree in combination with a substitution σ. By the remark preceding this theorem, we may assume that R is in *tyft* format. So, let

$$R = \frac{\{t_i \xrightarrow{a_i} y_i : i \in I\}}{f(\bar{x}) \xrightarrow{a} t},$$

where the terms t and t_i, for $i \in I$ are in $T(\Sigma, Var)$, $\bar{x} = x_1, \ldots, x_k$, and all of the variables in $\{x_1, \ldots, x_k\} \cup \{y_i : i \in I\}$, are pairwise distinct variables in Var. We have $x_i(\sigma) = u_i$, for all $1 \leq i \leq k$, and $t(\sigma) = u'$.
We shall define a substitution σ' such that

$$f(\bar{x})(\sigma') = f(\bar{v})$$

$$f(\bar{v}) \xrightarrow{a} t(\sigma')$$

$$t(\sigma) W t(\sigma').$$

We proceed by induction on the height of τ. Suppose the height of τ is $n+1$. Assume that for all s and t in $Terms$ with sWt, and transitions $s \xrightarrow{a} s'$ with a proof tree of height less than or equal to n, there exists $t' \in Terms$ with $t \xrightarrow{a} t'$ and $s'Wt'$.
Recall from Definition 5.17 that $Var(R) = \bigcup_l \Pi_l$. We shall, by induction on l, construct a sequence $\sigma_0 \subseteq \sigma_1 \subseteq \cdots$ (using here and below a set notation for substitutions) such that for all $l \geq 0$,

1. $domain(\sigma_l) = \Pi_l$
2. For all $z \in domain(\sigma_l)$: $z(\sigma) W z(\sigma_l)$
3. For all $i \in I$, if $\omega(i) = l$ then $t_i(\sigma_l) \xrightarrow{a_i} y_i(\sigma_l)$

(o): Take $\sigma_0 = \{(x_1, v_1), \ldots, (x_k, v_k)\}$. Then clause 1 holds since

$$domain(\sigma_0) = \{x_1, \ldots, x_k\} = \Pi_0.$$

Clause 2 holds since $u_i W v_i$, for all $1 \leq i \leq k$. Clause 3 trivially holds since $\omega(i) > 0$, for all $i \in I$. Now $f(\bar{x})(\sigma_0) = f(\bar{v})$. (Note that here the fact is used that the left side of the conclusion of the rule R contains only one function symbol (f) and that all the variables in $\bar{x}$ are pairwise distinct.)
(l+1): Suppose we have defined $\sigma_0, \ldots, \sigma_l$ satisfying clauses 1, 2 and 3 above. We extend σ_l by putting

$$\sigma_{l+1} = \sigma_l \cup \{(y_i, w_{y_i}) : \omega(i) = l+1\},$$

where for every $i \in I$ with $\omega(i) = l+1$, a term w_{y_i} is chosen as follows:

Let $i \in I$ with $\omega(i) = l + 1$. By definition of Π_{l+1}, this implies $Var(t_i) \subseteq \Pi_l$. By clause 2 (for l) we have, for all variables $z \in Var(t_i)$, that $z(\sigma) W z(\sigma_l)$. Since W is a congruence, this implies $t_i(\sigma) W t_i(\sigma_l)$. Consider the transition $t_i(\sigma) \xrightarrow{a_i} y_i(\sigma)$. Using the definition of W, we distinguish between two cases. First, if $t_i(\sigma) \sim_T t_i(\sigma_l)$ then there exists, by the fact that $\sim_T$ is a bisimulation, a term $w \in Terms$ such that $t_i(\sigma_l) \xrightarrow{a_i} w$ and $y_i(\sigma) \sim_T w$, hence $y_i(\sigma) W w$. Take $w_{y_i} = w$.

Second suppose that there exist an operator $g \in \Sigma$ and sequences (of length the arity of g) of terms $\bar{v}_1$ and $\bar{v}_2$ such that $t_i(\sigma) = g(\bar{v}_1)$ and $t_i(\sigma_l) = g(\bar{v}_2)$, and $\bar{v}_1 W \bar{v}_2$ (using an obvious shorthand). Since the transition $g(\bar{v}_1) \xrightarrow{a_i} y_i(\sigma)$ has a proof tree of height smaller than or equal to n, the induction hypothesis gives us the existence of a term w such that $g(\bar{v}_2) \xrightarrow{a_i} w$ and $y_i(\sigma) W w$. Also in this case, take $w_{y_i} = w$.

Note that in this construction the fact is used that all y_i's are mutually distinct.

It is immediate from the definition of σ_{l+1} that the clauses 1, 2 and 3 hold for $l + 1$.

Now define $\sigma' = \bigcup_l \sigma_l$. Then $f(\bar{x})(\sigma') = f(\bar{v})$. Since for all $i \in I$ we have $t_i(\sigma') \xrightarrow{a_i} y_i(\sigma')$, we have constructed a proof for the transition $f(\bar{v}) \xrightarrow{a} t(\sigma')$. Finally, because of the facts that $Var(t) \subseteq \bigcup_l \Pi_l = domain(\sigma')$, that for all $z \in domain(\sigma')$, $z(\sigma) W z(\sigma')$, and that W is a congruence, we have $t(\sigma) W t(\sigma')$. □