Algebraic representation of bisimulation for the tagh-format

A. Sokolova[1,2] & E.P. de Vink[1,3]

[1] Department of Mathematics and Computer Science, Eindhoven University of Technology

[2] Institute of Informatics, Skopje

[3] LIACS, Leiden University

a.sokolova@tue.nl, evink@win.tue.nl

Abstract

Process behavior is often described in terms of structural operational semantics. For a wide class of such rules, viz. those adhering to the GSOS format or to the tagh-format, it is possible to obtain automatically a finite axiomatization that characterizes strong bisimulation. Here we go a bit further and show for this format how to obtain canonical representatives of bisimulation equivalence classes under a condition of well-foundedness. This we do by constructing the initial algebra in the variety of semi-lattices over the axiomatization of a system.

1 Introduction

Structured operational semantics has become the predominant method of describing the operational semantics of process languages in terms of transition systems. Starting from the founding work of Bloom, Istrael and Meyer on GSOS in [9] and important contributions of De Simone [12] and Groote and Vaandrager [15], various other formats have been studied each with their respective merits, including the property of bisimulation being a congruence. Here, we mention: ntyft/ntyxt [14], panth [26], tree-rules [13], the work in [25], and the tagh-format of [5].

In [1], Aceto, Bloom and Vaandrager provide a method of generating a sound and complete axiomatization of strong bisimulation for transition system specification in the GSOS-format. Recently, in [5] this result has been slightly generalized in the setting of the tagh-format in order to deal with explicit termination. In this paper we focus on the question of deciding the bisimilarity of terms with respect to some given transition system specification. In general, this problem is known to be unsolvable (cf. [9]). However, under certain linearity and well-foundedness conditions on the transition rules a decision procedure can be given. More specifically, we address the following problem: Given a transition system specification over a signature Σ and rules confirming to the GSOS or tagh format, determine whether two ground terms in $T(\Sigma)$ are bisimilar without constructing a bisimulation itself.

For rules in GSOS-format and tagh-format the resulting transition system is finitely branching and computable. Moreover, as shown in [1, 5], a head normalization property allows one to trade an arbitrary operation in favour of the basic operations of nondeterministic choice and action prefixing. Together with well-foundedness of the underlying transition relation this induces an elimination theorem, stating that any term can be reduced to an equivalent basic term in a basic process algebra.

In [18] a description of the free algebra for a basic process language was presented. Here we adapt their result by dealing with action prefixing instead of sequential composition and incorporating the dagger-construction of [1] and explicit termination as in [4]. In a similar manner as in [18] we provide a technique for the construction of the initial algebra. The construction takes place in the variety $\mathcal{V}_{\dagger,\epsilon}$ defined by $E_{\dagger,\epsilon}$, the equational theory for bisimulation for the basic process language. The free $\mathcal{V}_{\dagger,\epsilon}$-algebra is given by set-theoretic means as a universe set with suitably defined operations, and an epimorphism is defined that maps each term of the signature of $\mathcal{V}_{\dagger,\epsilon}$ to its canonical unique representative in the free $\mathcal{V}_{\dagger,\epsilon}$-algebra. Combining this with the results of [1] and [5] we arrive at a procedure for determining whether any two terms with operations defined by a well-founded transition system specification in GSOS-format or tagh format are bisimilar or not, simply comparing their image in the initial algebra.

2 Preliminaries

2.1 Tagh-systems

We assume acquaintance with an SOS system as a pair $S_S = (\Sigma_S, R_S)$ of a finite signature Σ_S and a finite set R_S of rules. For further details see for instance [23, 2, 7].

The SOS system that we will use as basic in this paper is the system $S_{\dagger,\epsilon} = (\Sigma_{\dagger,\epsilon}, R_{\dagger,\epsilon})$. Here, the signature $\Sigma_{\dagger,\epsilon}$ is given by $\Sigma_{\dagger,\epsilon} = \{+, \delta, \epsilon\} \cup \{a(.) \mid a \in Act\} \cup \{\dagger_B(.) \mid B \subseteq Act\}$, where $+$ is a binary operation, δ and ϵ are constants, there is one unary operation for each action $a \in Act$, and also one unary operation $\dagger_B$ for each subset of Act. We use the notation ax or $a.x$ for $a(x)$ and $x \dagger B$ for $\dagger_B(x)$. The set of rules $R_{\dagger,\epsilon}$ consists of the following.

$$ax \xrightarrow{a} x \tag{1}$$

$$\frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'}, \quad \frac{y \xrightarrow{a} y'}{x + y \xrightarrow{a} y'} \tag{2}$$

$$\frac{x \xrightarrow{a} x'}{x \dagger B \xrightarrow{a} x'}(a \notin B) \tag{3}$$

$$\epsilon \downarrow, \quad \frac{x \downarrow}{(x+y) \downarrow}, \quad \frac{y \downarrow}{(x+y) \downarrow} \tag{4}$$

$$\frac{x \downarrow}{x \dagger B \downarrow} \tag{5}$$

The above is in fact the system for the basic process language from [5] with nondeterministic choice $+$, a prefixing operation for every action a in the set of actions Act, a unary one-step restriction, or initial-blocking, operation $\dagger_B$ for every subset of actions, an explicit termination constant ϵ, and a deadlock constant δ which has no transitions (cf. [4]). The symbol $\downarrow$ is used for postfix notation of the termination predicate, meaning that a term can terminate immediately.

Definition 2.1 *Let $S_S = (\Sigma_S, R_S)$ be an SOS system with termination. A binary relation R on $T(\Sigma_S)$ is a bisimulation if for all $x, y \in T(\Sigma_S)$ such that $R(x, y)$ the following conditions are met*

1. *if $x \xrightarrow{a} x'$ then there exists y' such that $y \xrightarrow{a} y'$ and $R(x', y')$;*
2. *if $y \xrightarrow{a} y'$ then there exists x' such that $x \xrightarrow{a} x'$ and $R(x', y')$;*
3. *$x \downarrow$ iff $y \downarrow$, i.e. x can terminate if and only if y can.*

Two terms x and y are bisimilar (notation $x \sim_S y$) if there exists a bisimulation relation R such that $R(x, y)$.

A familiar result states that bisimulation can be captured equationally, i.e. two terms can be proved to be equal using the equations exactly when the two terms are bisimilar (see [20, 21]).

Theorem 2.2 *The equational theory given by the axioms*

$$x + y = y + x \tag{6}$$
$$(x + y) + z = x + (y + z) \tag{7}$$
$$x + x = x \tag{8}$$
$$x + \delta = x \tag{9}$$
$$(x + y) \dagger B = x \dagger B + y \dagger B \tag{10}$$
$$(a.x) \dagger B = a.x,\ a \notin B \tag{11}$$
$$(a.x) \dagger B = \delta,\ a \in B \tag{12}$$
$$\delta \dagger B = \delta \tag{13}$$
$$\epsilon \dagger B = \epsilon \tag{14}$$

is sound and complete with respect to bisimulation for $S_{\dagger,\epsilon}$. □

We will apply the algebraic results, that are presented in the sequel, to a subclass of SOS-systems that are characterized by the format of their rules. Here, we focus on the tagh-format of [5] that extends the well-known GSOS-format of [9] with termination.

Definition 2.3

1. *A tagh transition rule for an* n*-ary operation* f *is a deduction rule of the format*

$$\frac{\{x_i \stackrel{a_{ip}}{\to} y_{ip} |\ i \in I, p \in P_i\} \quad \{x_j \stackrel{b}{\not\to}\ |\ j \in J, b \in B_j\} \quad \{x_k \downarrow |\ k \in K\}}{f(x_1, \ldots, x_n) \stackrel{a}{\to} C[x_m, y_{ip}]}$$

   where $I, J, K \subseteq \{1, \ldots, n\}$, *for* $i \in I$, P_i *a nonempty finite index set, for* $j \in J$, B_j *a finite set of actions, and, for* $m \in \{1, \ldots, n\}, i \in I, p \in P_i$, x_m, y_{ip} *are pairwise distinct variables, that are the only variables that may occur in the context* $C[x_m, y_{ip}]$.
   The arguments $x_i, i \in I$ *are called active, while the* $x_j, j \in J$ *are called negative.*

2. *A tagh termination rule for an* n*-ary operation* f *is a deduction rule of the format*

$$\frac{\{x_k \downarrow\ |\ k \in K\}}{f(x_1, \ldots, x_n) \downarrow}$$

   where $x_1, \ldots, x_n$ *are pairwise distinct variables and* $K \subseteq \{1, \ldots, n\}$.

A tagh-system is SOS system where all the rules for all the operations, except those of $\Sigma_{\dagger,\epsilon}$, are tagh rules.

Since the tagh-format is an instance of the panth-format, by a general result of [26] we have the following lemma.

Lemma 2.4 *Bisimulation is a congruence for every tagh-system.* $\square$

The main result of [5], exploiting similar techniques of [1] for GSOS-systems, is a procedure to automatically obtain for a tagh-system a sound and complete system of equations that characterize bisimulation.

Theorem 2.5 *There exists a procedure to obtain for any tagh-system S_T, a disjoint extension S'_T and an equational theory E'_T such that E'_T is head-normalizing, sound and complete with respect to strong bisimulation.* $\square$

2.2 Free algebras and SOS systems

In this subsection we present basic notions from universal algebra and discuss the connection to our problem of finding a system of representatives of terms modulo bisimulation. For more details on free algebras see, for instance, [11, 22, 24].

An algebra is a pair $\mathbf{A} = (A; \Sigma)$ where A is a carrier set and Σ is a signature, or a set of basic operations. A variety $\mathcal{V}$ is a class of algebras of the same type (i.e., over the same signature) that is closed under homomorphic images, subalgebras and direct products. By Birkhoff's variety theorem, every variety is an equational class, meaning that it consists of all the algebras that are models of a certain equational theory. Hence we usually think of a variety as a class of algebras defined by identities.

Definition 2.6 *Let $\mathcal{V}$ be a variety. A free algebra in $\mathcal{V}$ with base set X is an algebra $\mathbf{F} = (F; \Sigma)$ such that*

- $\mathbf{F}$ *belongs to* $\mathcal{V}$.
- *The set $X \subseteq F$ generates F.*
- *For every algebra $\mathbf{A} = (A, \Sigma) \in \mathcal{V}$, any mapping from X to A can be extended to a homomorphism from $\mathbf{F}$ to $\mathbf{A}$, i.e. $\mathbf{F}$ satisfies the universal mapping property.*

An initial algebra in $\mathcal{V}$ is the free algebra with an empty base.

Let Σ be a signature. Let $\mathcal{V}_0$ denote the variety over Σ that has no defining identities, i.e. $\mathcal{V}_0$ contains all the algebras of type Σ. The free algebra in $\mathcal{V}_0$ with base X is the algebra of Σ-terms built over the base X. This is also called *the absolutely free algebra with base X of type* Σ, further on denoted by $T(\Sigma_X)$. In case of the initial algebra in $\mathcal{V}_0$ we write $T(\Sigma)$, it exists if and only if the signature contains constants.

For later reference we include here a classical characterization of free algebras in a given variety.

Theorem 2.7 *Let $\mathcal{V}$ be a variety of algebras of type Σ. Let $E_{\mathcal{V}}$ denote the underlaying equational theory and $=_{\mathcal{V}}$ the equality in $E_{\mathcal{V}}$. Then the free algebra in $\mathcal{V}$ with base X (up to isomorphism) is $T(\Sigma_X)/=_{\mathcal{V}}$.* $\square$

In a free algebra of a given variety only the identities of the whole variety hold and nothing else, hence it is free of any other laws. In an absolutely free algebra no laws hold, all the terms are distinct elements of the absolutely free algebra. For an example of a free algebra consider the variety of semigroups i.e. all algebras with a single binary operation that satisfies the associative law. The free semigroup with base X is isomorphic with X^+, all words over the alphabet X with the operation of concatenation. Every group can also be considered a member of this variety but, as more laws hold, is not free in the variety of semigroups.

Having an algorithmic construction of free algebras, with a countable base, in a variety $\mathcal{V}$ is known as solving the word problem for free algebras in $\mathcal{V}$, and is equivalent to the decision problem of the equational theory of $\mathcal{V}$. Namely, it gives a procedure to decide whether two given terms in signature Σ (over a countable set of variables) are equal modulo identities of $\mathcal{V}$.

There are various ways of constructing free algebras in a given variety and they vary from rather specific constructions for certain varieties (e.g. [24, 22, 17, 18, 19]) to more general constructions (for example [10, 3]). Roughly speaking, in the sense of Definition 2.6, there are two main approaches. The first consists of constructing a carrier set that consists of a certain kind of objects, defining operations on it of type Σ and proving that the conditions of Definition 2.6 are met. The second is by obtaining, via reduction, a system of representatives for $=_{\mathcal{V}}$, i.e. by constructing F as a subset of $T(\Sigma_X)$ containing exactly one element for each $=_{\mathcal{V}}$-congruence class. Note that the first and the second construction could be linked together if we construct the isomorphism between the object $\mathbf{F}$ obtained with the first construction and $T(\Sigma_X)/=_{\mathcal{V}}$ or construct the corresponding epimorphism from $T(\Sigma_X)$

onto $\mathbf{F}$. We follow this later approach when constructing the free algebras in Section 3.

By now we have mentioned the main ingredients used in this paper. Throughout the paper we will use the following notation:

S_S	–	an SOS system
Σ_S	–	the signature of S_S
$T(\Sigma_S)$	–	the set of ground terms over a signature Σ_S
$\sim_S$	–	bisimulation w.r.t. S_S
$\mathbf{A}_S$	–	the algebra $T(\Sigma_S)/\sim_S$
E_S	–	equational theory for bisimulation w.r.t. S_S
$=_S$	–	equality w.r.t. E_S
$\mathcal{V}_S$	–	the variety defined by E_S
$\mathbf{F}_S^X$	–	the free algebra in $\mathcal{V}_S$ with base X
$\mathbf{F}_S$	–	the initial algebra in $\mathcal{V}_S$

According to the convention above, the equational theory $E_{\dagger,\epsilon}$ consists of the identities of Theorem 2.2, i.e. the laws (6) up to (14).

Next we introduce several subsystems of $S_{\dagger,\epsilon}$ that will be used in the sequel. The simplest one is S_F (with F referring to *FINTREE* as coined in [1]), with signature $\Sigma_F = \{+, \delta\} \cup \{a(.)|\ a \in Act\}$ and rules (1) and (2). The system $S_{F,\epsilon}$ with signature $\Sigma_{F,\epsilon} = \{+, \delta, \epsilon\} \cup \{a(.)|\ a \in Act\}$ and rules (1),(2) and (4) is an extension of S_F with explicit termination. What remains of $S_{\dagger,\epsilon}$ without termination is the system $S_\dagger$ with signature $\Sigma_\dagger = \Sigma_{\dagger,\epsilon} \setminus \{\epsilon\}$ and rules (1), (2) and (3).

Note that E_F consists of the identities (6) thru (9); $E_{F,\epsilon}$ has the same identities, while $E_\dagger$ has the identities (6) thru (13). For systems that do not include termination the definition of bisimulation is restricted to the first two conditions of Definition 2.1 as usual.

Since $E_{\dagger,\epsilon}$ is a disjoint extension of $E_{F,\epsilon}$ we have the following fact.

Lemma 2.8 *For $x, y \in T(\Sigma_{F,\epsilon})$ it holds that $x =_{\dagger,\epsilon} y$ iff $x =_{F,\epsilon} y$* $\square$

The following lemma explains the connection between free algebras and the issue of finding canonical representatives for bisimulation equivalence classes.

Lemma 2.9 *Let S_S be an SOS system with signature Σ_S such that bisimulation $\sim_S$ is a congruence. Assume that E_S is a sound and complete equational theory for S_S. Let $\mathcal{V}_S$ be the variety induced by E_S. Then the initial $\mathcal{V}_S$-algebra $\mathbf{F}_S$ is isomorphic to $\mathbf{A}_S$, i.e. to $T(\Sigma_S)/\sim_S$.*

Proof Since E_S is sound and complete for $\sim_S$, we have that $x \sim_S y \Leftrightarrow x =_S y$ for any $x, y \in T(\Sigma_S)$. Hence $\sim_S$ is in fact the congruence generated by the defining identities of $\mathcal{V}_S$ and $\mathbf{F}_S \cong T(\Sigma_S)/\sim_S$. □

Thus, under the conditions of Lemma 2.9, the problem of obtaining canonical representatives for bisimulation is equivalent to the problem of constructing the initial algebra in the corresponding variety.

3 Free $\mathcal{V}_{\dagger,\epsilon}$ algebras

In this section we focus on the construction of the free algebra with a given base in the variety $\mathcal{V}_{\dagger,\epsilon}$ induced by the laws of Theorem 2.2. By Lemma 2.9 above this amounts to constructing the quotient algebra $\mathbf{A}_{\dagger,\epsilon}$.

First, we focus on the representation of $\mathbf{A}_{F,\epsilon}$ for which we borrow some ideas from [18]. The construction of [18] of a free algebra for BPA, as it is called in [6], is a construction of the first kind explained above. However, following our basic observation, this can easily be turned into a construction of the second kind, by defining a suitable epimorphism.

As mentioned before, $\mathcal{V}_{F,\epsilon}$ denotes the variety defined by the identities of $E_{F,\epsilon}$ in the signature $\Sigma_{F,\epsilon}$. Note that, in view of the laws of $E_{F,\epsilon}$, any $\mathcal{V}_{F,\epsilon}$-algebra is a semi-lattice with respect to $+$, with unit δ. We shall use the description of a free semi-lattice with unit for the free objects in $\mathcal{V}_{F,\epsilon}$.

Let, for the moment, X be some given set such that $X \cap \mathcal{P}(X) = \emptyset$. Put $SL(X) = \{\emptyset\} \cup X \cup \mathcal{P}_{\geq 2}(X)$. We define an operation $*$ on $SL(X)$ as follows:

$$\begin{array}{lcl} \emptyset * z & = & z * \emptyset \; = \; z \\ x * x' & = & \{x, x'\}, \quad x \neq x' \\ x * x & = & x \\ y * y' & = & y \cup y' \\ y * x & = & x * y \; = \; \{x\} \cup y \end{array}$$

where $x, x' \in X, y, y' \in \mathcal{P}_{\geq 2}(X), z \in SL(X)$.

Note that the operation $*$ is essentially the familiar union of sets, only each singleton is replaced by the element itself. This is done since we want $X \subseteq SL(X)$ to hold, as the SL-operator will be applied recursively in the sequel. Also note that the operation $*$ is well-defined since we assumed $X \cap \mathcal{P}(X) = \emptyset$.

Lemma 3.1 *The structure $(SL(X), *)$ is the free semi-lattice with unit $\emptyset$ and base X.*

Proof We prove that $SL(X)$ belongs to the variety of semi-lattices with a unit, that it is generated by X and that it has the universal mapping property.

Let $y \in \mathcal{P}_{\geq 2}(X)$. Then $y = \{x_1, \ldots, x_k\} \subseteq X$, for some $k \geq 2$. Thus, $y = x_1 * \{x_2, \ldots, x_k\}$ if $k > 2$ and $y = x_1 * x_2$ if $k = 2$. This implies that $\mathcal{P}_{\geq 2}(X)$ is generated by X.

¿From the definition of $*$ it is obvious that $SL(X)$ is a semi-lattice. Let $(S, \circ)$ be any semi-lattice with unit 0, and $f \colon X \to S$ any map. We extend f to a homomorphism $f^* \colon SL(X) \to S$ as follows:

$$
\begin{aligned}
&f^*(\emptyset) = 0 \\
&f^*(x) = f(x) \text{ for } x \in X \\
&f^*(y) = f^*(x_1) \circ \cdots \circ f^*(x_k) \text{ for } y = \{x_1, \ldots, x_k\} \in \mathcal{P}_{\geq 2}(X)
\end{aligned}
$$

It is, using the definition of $*$, straightforwardly checked that f^* is indeed a homomorphism. From this the universal mapping property follows. $\square$

For the construction of our free algebras we need to introduce yet another operator.

wDefine $\Sigma_p = \Sigma_{F,\epsilon} - \{+, \delta, \epsilon\}$, i.e. Σ_p is the signature consisting of prefixing operations only. For practical reasons we shall denote the absolutely free algebra with base X over this signature by $\overline{X}$. Clearly $\overline{X}$ is the set of all Σ_p terms built from X.

Using the operators $SL(\cdot)$ and $\overline{(\cdot)}$ we construct the universe $F^X_{F,\epsilon}$ of the free $\mathcal{V}_{F,\epsilon}$-algebra with base X. We define a sequence of sets F_i, $i \geq 0$ by

$$
\begin{aligned}
F_0 &= \overline{X \cup \{\epsilon, \emptyset\}} \\
F_{i+1} &= \overline{SL(F_i \setminus \mathcal{P}(F_i))}
\end{aligned}
$$

and we put $F^X_{F,\epsilon} = \bigcup_{i \geq 0} F_i$. Note that all the sets $F_i \setminus \mathcal{P}(F_i)$ are eligible as argument of SL, since $(F_i \setminus \mathcal{P}(F_i)) \cap \mathcal{P}(F_i \setminus \mathcal{P}(F_i)) = \emptyset$. Also observe that $F_i \cap \mathcal{P}_1(F_i) = \emptyset$, so we can write $F_i \setminus \mathcal{P}(F_i)$ instead of $F_i \setminus (\{\emptyset\} \cup \mathcal{P}_{\geq 2}(F_i))$.

The next lemma collects some properties that are needed later.

Lemma 3.2

(i) *The set $F_i \setminus \mathcal{P}(F_i)$ does not contain sets as its elements, for $i \geq 0$, if X does not.*

(ii) *For $i \geq 0$, if an element x of F_{i+1} is a set, then $x \subseteq F_i \setminus \mathcal{P}(F_i)$.*

(*iii*) $F_i \setminus \mathcal{P}(F_i) \subseteq F_{i+1} \setminus \mathcal{P}(F_{i+1})$ *for* $i \geq 0$.

(*iv*) $F_i \subseteq F_{i+1}$ *for* $i \geq 0$.

Proof

(*i*) Clearly, $F_0 \setminus \mathcal{P}(F_0)$ does not contain sets. Let $x \in F_i \setminus \mathcal{P}(F_i)$, for $i \geq 1$. Then $x \in F_i$ and either $x \in \overline{SL(F_{i-1} \setminus \mathcal{P}(F_{i-1}))} \setminus SL(F_{i-1} \setminus \mathcal{P}(F_{i-1}))$ in which case x is not a set, or $x \in SL(F_{i-1} \setminus \mathcal{P}(F_{i-1}))$. In the later case there are two more possibilities. The first case is that $x \in F_{i-1} \setminus \mathcal{P}(F_{i-1})$ and by the inductive hypothesis x is not a set. The second case is $x = \{x_1, \ldots, x_k\}$, $k \geq 2$, $x_1, \ldots, x_k \in F_{i-1} \setminus \mathcal{P}(F_{i-1})$ which implies $x_1, \ldots, x_k \in F_i$, i.e. $x \in \mathcal{P}(F_i)$ which is a contradiction.

(*ii*) Let $x \in F_{i+1}$ be a set, for $i \geq 0$. Then $x \in SL(F_i \setminus \mathcal{P}(F_i))$ and by (*i*) we have that $x \notin F_i \setminus \mathcal{P}(F_i)$, i.e. $x \subseteq F_i \setminus \mathcal{P}(F_i)$.

(*iii*) By construction we have $F_i \setminus \mathcal{P}(F_i) \subseteq F_{i+1}$. In addition, by (*i*) if $x \in F_i \setminus \mathcal{P}(F_i)$ then x is not a set, hence $x \notin \mathcal{P}(F_{i+1})$.

(*iv*) This is a consequence of (*iii*) and the definition of the sets F_i.

$\square$

From the lemma we obtain as a corollary the following.

Corollary 3.3 *The sequence* $(SL(F_i \setminus \mathcal{P}(F_i))|\ i \geq 0)$ *is a chain of semi-lattices, the sequence* $(F_i|\ i \geq 0)$ *is a chain of algebras of type* Σ_p *and* $F^X_{F,\epsilon} = \bigcup_{i \geq 0} F_i = \bigcup_{i \geq 0} SL(F_i \setminus \mathcal{P}(F_i))$. $\square$

Thus $F^X_{F,\epsilon}$ inherits interpretations for $+$ and prefixing operations from the F_i's. Additionally, we can take $\emptyset$ is an interpretation for δ. Therefore $F^X_{F,\epsilon}$ can be turned into a $\Sigma_{F,\epsilon}$-algebra, $\mathbf{F}^X_{F,\epsilon}$, in a natural way.

Note that the elements of $F^X_{F,\epsilon}$ have the form $a_1 a_2 \ldots a_n x$ where x is either ϵ or $\emptyset$, or $x = \{x_1, \ldots, x_k\}$ for some $x_1, \ldots, x_k \in F^X_{F,\epsilon}$ which are not sets. The *length* of $a_1 a_2 \ldots a_n x$ is the number n of unary prefix operations that appear as top operations in this element, and the *depth* of $a_1 a_2 \ldots a_n x$ is the number of pairs of braces appearing in x (0 for $x = \epsilon$ or $x = \emptyset$). In this informal way we have defined two functions, *length* and *depth*, that map elements of $F^X_{F,\epsilon}$ to natural numbers.

By now we have the ingredients for our first result.

Theorem 3.4 $\mathbf{F}^X_{F,\epsilon}$ *is a free* $\mathcal{V}_{F,\epsilon}$*-algebra with base* X.

Proof The proof goes by induction on *length* and *depth*. First of all $F^X_{F,\epsilon} \in \mathcal{V}_{F,\epsilon}$, since by construction it is easy to see that the $+$ operation satisfies the laws of a semi-lattice with unit $\emptyset$. Next we check that X generates $F^X_{F,\epsilon}$. Clearly, X generates F_0. If $x = ax' \in F^X_{F,\epsilon} \setminus F_0$ then by *length*-induction x' is generated by X and so is x. Finally if $x = \{x_1, \ldots, x_k\}$ then by *depth*-induction all the x_i's are generated by X and so is $x = x_1 + \cdots + x_k$. Let $(F, \Sigma_{F,\epsilon})$ be any other $\mathcal{V}_{F,\epsilon}$-algebra and $f \colon X \to F$ be any mapping. Then we can extend f to a homomorphism $f' \colon F^X_{F,\epsilon} \to F$ by putting $f'(\epsilon) = \epsilon$, $f'(\emptyset) = \delta$, $f'(ax) = af'(x)$, $f'(\{x_1, \ldots, x_n\}) = f'(x_1) + \cdots + f'(x_n)$. An easy inductive argument on *length* and *depth* shows that f' is indeed a well-defined homomorphism as desired. $\square$

For the remainder of this section we work with an empty base, so $X = \emptyset$. We define the mapping $e_F \colon T(\Sigma_{F,\epsilon}) \to F_{F,\epsilon}$ by

$$\begin{array}{lcl} e_F(\epsilon) & = & \epsilon \\ e_F(\delta) & = & \emptyset \\ e_F(ax') & = & a.e_F(x') \\ e_F(x' + x'') & = & e_F(x') + e_F(x'') \end{array}$$

where the $+$ in the righthand side is the $+$ of $F_{F,\epsilon}$. Clearly e_F is an epimorphism, so $F_{F,\epsilon} = e_F(T(\Sigma_{F,\epsilon}))$. From the general theory we have $T(\Sigma_{F,\epsilon})/ker(e_F) \cong \mathbf{F}_{F,\epsilon}$, hence two terms in $T(\Sigma_{F,\epsilon})$ are equivalent modulo the identities of $\mathcal{V}_{F,\epsilon}$ exactly when they have the same image under e_F, i.e. for all $x, y \in T(\Sigma_{F,\epsilon})$ we have that

$$x =_{F,\epsilon} y \iff e_F(x) = e_F(y) \tag{15}$$

Therefore the set $\{e_F(x) \mid x \in T(\Sigma_{F,\epsilon})\}$ constitutes a system of representatives of equivalence modulo $E_{F,\epsilon}$.

So far we have constructed the free $\mathcal{V}_{F,\epsilon}$-algebra and we have shown how to obtain canonical representatives of congruence classes of $T(\Sigma_{F,\epsilon})$. Next we extend the above construction for $\mathcal{V}_{\dagger,\epsilon}$. Note that the identities given in Theorem 2.5 for the unary one-step restriction operations allow for elimination of these operations. In order to make this explicit we define mappings

$e_B : T(\Sigma_{F,\epsilon}) \to T(\Sigma_{F,\epsilon})$, for each subset $B \subseteq Act$, by

$$\begin{array}{rcl} e_B(\epsilon) & = & \epsilon \\ e_B(\delta) & = & \delta \\ e_B(x' + x'') & = & e_B(x') + e_B(x'') \\ e_B(ax) & = & \delta \text{ if } a \in B \\ e_B(ax) & = & ax \text{ if } a \notin B \end{array}$$

and a mapping $e_\dagger : T(\Sigma_{\dagger,\epsilon}) \to T(\Sigma_{F,\epsilon})$ by $e_\dagger(x) = x$ if $x \in T(\Sigma_{F,\epsilon})$ and

$$\begin{array}{rcl} e_\dagger(ax) & = & ae_\dagger(x) \\ e_\dagger(x' + x'') & = & e_\dagger(x') + e_\dagger(x'') \\ e_\dagger(x \dagger B) & = & e_B(e_\dagger(x)) \end{array}$$

otherwise. An inductive argument shows that the mappings $e_\dagger$ and e_B, for $B \subseteq Act$ are well-defined. Moreover, $e_\dagger$ is surjective, since for $x \in T(\Sigma_{F,\epsilon})$ we have $x = e_\dagger(x)$, and respects, by definition, all the operations of $\Sigma_{F,\epsilon}$. Additionally we have the following.

Lemma 3.5 *For any $x \in T(\Sigma_{F,\epsilon})$ it holds that $x \dagger B =_{\dagger,\epsilon} e_B(x)$ and for any $x \in T(\Sigma_{\dagger,\epsilon})$ we have $x =_{\dagger,\epsilon} e_\dagger(x)$. For any $x, y \in T(\Sigma_{F,\epsilon})$ it holds that $x =_{F,\epsilon} y \implies e_B(x) =_{F,\epsilon} e_B(y)$.*

Proof By a simple inductive argument on the structure of $\Sigma_{F,\epsilon}$-terms it follows that $x \dagger B =_{\dagger,\epsilon} e_B(x)$ for all $x \in T(\Sigma_{F,\epsilon})$. Then, once again by induction but now on the structure of $\Sigma_{\dagger,\epsilon}$ terms, we get that $x =_{\dagger,\epsilon} e_\dagger(x)$ for all $x \in T(\Sigma_{\dagger,\epsilon})$. If for two $\Sigma_{F,\epsilon}$-terms x, y and $B \subseteq Act$ it holds that $x =_{F,\epsilon} y$, then, as $S_{\dagger,\epsilon}$ is a disjoint extension of $S_{F,\epsilon}$, it also holds that $x =_{\dagger,\epsilon} y$. Hence $x \dagger B =_{\dagger,\epsilon} y \dagger B$ and, by the above, $e_B(x) =_{\dagger,\epsilon} e_B(y)$. But, since $e_B(x), e_B(y)$ are $\Sigma_{F,\epsilon}$-terms as well, we obtain $e_B(x) =_{F,\epsilon} e_B(y)$ by another appeal to the disjointness of the extension of $S_{\dagger,\epsilon}$ over $S_{F,\epsilon}$. $\Box$

The lemma above states that any $\Sigma_{\dagger,\epsilon}$-term x can be reduced to a $\Sigma_{F,\epsilon}$-term, viz. $e_\dagger(x)$. There are two reasons underlying the well-definedness of the mappings $e_\dagger$ and e_B and the proof of Lemma 3.5. The first is known as a head-normalization property. Namely for each $\Sigma_{\dagger,\epsilon}$-term x there either exist $\Sigma_{\dagger,\epsilon}$-terms x_1, x', and an action a_1 such that $x = a_1 x_1 + x'$, or $x = \epsilon$, or $x = \delta$. Moreover, the transition relation is finitely branching for $S_{\dagger,\epsilon}$. The second reason is that the rules for $\dagger_B$-operations are well-founded, i.e. no infinite sequence of rewritings is possible. We will exploit this observation below, when generalizing the above result.

So far we have that $T(\Sigma_{\dagger,\epsilon}) \xrightarrow{e_\dagger} T(\Sigma_{F,\epsilon}) \xrightarrow{e_F} F_{F,\epsilon}$, and both $e_\dagger$ and e_F are surjective mappings that respect the operations of $\Sigma_{F,\epsilon}$. Hence $e_F \circ e_\dagger$ is itself surjective, i.e. $(e_F \circ e_\dagger)(T(\Sigma_{\dagger,\epsilon})) = F_{F,\epsilon}$, and it respects the operations of $\Sigma_{F,\epsilon}$. Moreover, the following holds.

Lemma 3.6 *For all $x, y \in T(\Sigma_{\dagger,\epsilon})$ it holds that $x =_{\dagger,\epsilon} y \iff (e_F \circ e_\dagger)(x) = (e_F \circ e_\dagger)(y)$.*

Proof Let $x, y \in T(\Sigma_{\dagger,\epsilon})$. Then we have that $x =_{\dagger,\epsilon} y$ iff $e_\dagger(x) =_{\dagger,\epsilon} e_\dagger(y)$, by Lemma 3.5, iff $e_\dagger(x) =_{F,\epsilon} e_\dagger(y)$, by Lemma 2.8, iff $e_F(e_\dagger(x)) = e_F(e_\dagger(y))$, by equation (15). □

Next we extend the signature of $\mathbf{F}_{F,\epsilon}$ by adding the one-step restriction operations. We put, for $x \in F_{F,\epsilon}$, $x \dagger B = (e_F \circ e_\dagger)(x_0 \dagger B)$ if $x_0 \in T(\Sigma_{\dagger,\epsilon})$ is such that $(e_F \circ e_\dagger)(x_0) = x$ and $B \subseteq Act$. We check that this indeed defines operations $(\cdot) \dagger B$ on $F_{F,\varepsilon}$.

Lemma 3.7 *The operations $(\cdot) \dagger B$ on $F_{F,\epsilon}$ are well-defined.*

Proof Pick $x \in F_{F,\epsilon}$. Let $x', x'' \in T(\Sigma_{\dagger,\epsilon})$ be such that $e_F(e_\dagger(x')) = x$ and $e_F(e_\dagger(x'')) = x$. Hence $e_F(e_\dagger(x')) = e_F(e_\dagger(x'')$ and, by equation (15), $e_\dagger(x') =_{F,\epsilon} e_\dagger(x'')$. Therefore, by one of the properties of Lemma 3.5, we obtain $e_B(e_\dagger(x')) =_{F,\epsilon} e_B(e_\dagger(x''))$. This is, by the definition of $e_\dagger$, equivalent to $e_\dagger(x' \dagger B) =_{F,\epsilon} e_\dagger(x'' \dagger B)$. Using equation (15) we conclude that $e_F(e_\dagger(x' \dagger B)) = e_F(e_\dagger(x'' \dagger B))$, which was to be shown. □

Let $\mathbf{F}_{\dagger,\epsilon}$ denote the algebra obtained by extending $\mathbf{F}_{F,\epsilon}$ to a $\Sigma_{\dagger,\epsilon}$-algebra using the operations $(\cdot) \dagger B$ as given above.

Theorem 3.8 *The algebra $\mathbf{F}_{\dagger,\epsilon}$ is the initial algebra in the variety $\mathcal{V}_{\dagger,\epsilon}$.*

Proof As stated before $e_F \circ e_\dagger$ is an epimorphism from $T(\Sigma_{\dagger,\epsilon})$ onto $\mathbf{F}_{\dagger,\epsilon}$ such that, by Lemma 3.6, $ker(e_F \circ e_\dagger)$ is equal to $=_{\dagger,\epsilon}$. Hence $\mathbf{F}_{\dagger,\epsilon} \cong T(\Sigma_{\dagger,\epsilon})/ker(e_F \circ e_\dagger) = T(\Sigma_{\dagger,\epsilon})/ =_{\dagger,\epsilon}$ and the statement follows by Theorem 2.7. □

The next consequence is of special interest to us as a characterization of bisimulation.

Corollary 3.9 *For any $x, y \in T(\Sigma_{\dagger,\epsilon})$ it holds that $x \sim_{\dagger,\epsilon} y \Leftrightarrow (e_F \circ e_\dagger)(x) = (e_F \circ e_\dagger)(y)$.*

Proof Immediate from Theorem 3.8 and the results of Section 2. □

Thus the set $\{(e_F \circ e_\dagger)(x) \mid x \in T(\Sigma_{\dagger,\epsilon})\}$ is a system of representatives for bisimulation equivalence for $S_{\dagger,\epsilon}$. Hence, we have some canonical representatives which are not terms. Such can be achieved by introducing an ordering on terms, but there is no need for doing so; there is already a procedure for determining whether two terms have the same canonical representative. It should also be noted that the application of the mappings e_F and $e_\dagger$ can be done mechanically. This is in fact the term rewriting mechanism underlaying the equational theory.

4 More free algebras

In the previous section we discussed the construction of the initial $\mathcal{V}_{F,\epsilon}$-algebra $e_F(T(\Sigma_{F,\epsilon}))$ as well as the construction of the initial $\mathcal{V}_{\dagger,\epsilon}$-algebra $e_F(e_\dagger(T(\Sigma_{\dagger,\epsilon})))$. We can do the same for other operations defined by SOS rules. In fact, the procedure applies well for an SOS system in tagh-format under some mild conditions.

The two essential elements for the argumentation above are a head-normalization property and a well-foundedness criterion. Head-normalization is free for SOS system S in tagh-format; it always holds in an equivalent extension S' of S. Thus, for tagh-systems that can be shown to be well-founded we can construct a mapping $e\colon T(\Sigma_{S'}) \to T(\Sigma_{\dagger,\epsilon})$ as before. From this an initial algebra and a system of representatives can be constructed.

Definition 4.1 *Let S_S be an SOS system with signature Σ_S. A term $x \in T(\Sigma_S)$ is well-founded if there is no infinite sequence of S_S-transitions starting in x, i.e. there does not exist an infinite sequence of terms $(x_i|\ i \geq 0)$ and an infinite sequence of actions $(a_i|\ i \geq 0)$ such that $x = x_0$ and $x_i \xrightarrow{a_i}_S x_{i+1}$ for $i \geq 0$. The SOS system S_S is well-founded if all terms in $T(\Sigma_S)$ are well-founded.*

Unfortunately well-foundedness of SOS rules is undecidable in general. But as shown in [1], it is decidable via syntactical well-foundedness for a rather wide class of *GSOS* rules. Since termination does not at all affect well-foundedness of rules, the same holds for tagh-format rules.

A tagh system is called *linear* if each variable occurs at most once in the target $(C[x_m, y_{ip}])$ and for each active argument i, at most one of the following holds: x_i occurs in the target; at most one of y_{ip} does. For linear tagh-systems well-foundedness is indeed decidable, in fact it is equivalent to the problem of determining whether a linear system of diophantine equations has a solution in the set of natural numbers.

When all the rules that extend $S_{\dagger,\epsilon}$ are well founded, then together with head-normalization property, by Noetherian induction ([11, 16]), it follows that a counterpart of Lemma 3.6 holds for all extra operations of the tagh-system $E_{S'}$.

More concretely, since there are no infinite transitions and there is no infinite branching, we can define a function "maximal number of transitions in a row possible", *mntp*, that maps any term $x \in T(\Sigma_{S'})$ to a natural number $mntp(x)$ which is the number of transitions in a maximal transition sequence starting from x. Then by induction on *mntp* of terms from the head-normalization property it follows that for any $S_{S'}$ term x there is an $S_{\dagger,\epsilon}$ term y such that x is bisimilar to y i.e. $x =_{S'} y$ in $\mathbf{A}_{S'}$.

Thus, in the same manner as before, we can define a surjective mapping e which respects the $\Sigma_{\dagger,\epsilon}$ operations and $e(x) =_{S'} x$ for $x \in T(\Sigma_{S'})$. This way we get that $T(\Sigma_{S'}) \stackrel{e}{\to} T(\Sigma_{\dagger,\epsilon}) \stackrel{e_F \circ e_\dagger}{\to} F_{\dagger,\epsilon}$. Subsequently, we extend the signature of $\mathbf{F}_{\dagger,\epsilon}$ to the signature $\Sigma_{S'}$ thus obtaining the algebra $\mathbf{F}_{S'}$ with universe $F_{S'} = F_{\dagger,\epsilon}$. Namely, if f is an n-ary operation in $\Sigma_{S'}$ and $e(x_1), \ldots, e(x_n) \in F_{\dagger,\epsilon}$ then we put $f(e(x_1), \ldots, e(x_n)) = e(f(x_1, \ldots, x_n))$ where $f(x_1, \ldots, x_n)$ in the right hand side denotes an existing term in $T(\Sigma_{S'})$.

In the exactly same way as before one can prove the following.

Lemma 4.2 *For all $x, y \in T(\Sigma_{S'})$ it holds $x =_{S'} y \Leftrightarrow (e_F \circ e_\dagger \circ e)(x) = (e_F \circ e_\dagger \circ e)(y)$.* □

The lemma is the main ingredient of the next theorem.

Theorem 4.3 *The structure $\mathbf{F}_{S'}$ is the initial algebra in the variety $\mathcal{V}_{S'}$.* □

From the theorem, in turn, it follows that

Corollary 4.4 *For any $x, y \in T(\Sigma_S)$, terms in the original signature, it holds that $x \sim_S y \Leftrightarrow (e_F \circ e_\dagger \circ e)(x) = (e_F \circ e_\dagger \circ e)(y)$.* □

So, we start of from a transition system specification S in tagh-format. By exploiting the result from [1, 5] we obtain a disjoint extension S' of which the accompanying equational theory $E_{S'}$ is sound and complete for bisimulation with respect to S'. The theory developed in the present paper, enables us to construct the initial $\mathcal{V}_{S'}$ algebra via a suitable epimorphism. Via this mapping we are given a system of canonical representatives for bisimulation for the starting tagh system S. Thus, in the end, the problem whether two S-terms x, y are bisimilar can be answered: they are if $e_F(e_\dagger(e(x))) = e_F(e_\dagger(e(y)))$.

References

[1] L. Aceto, B. Bloom, F. Vaandrager: *Turning SOS Rules into Equations*, Information and Computation, 111:1-52, 1994.

[2] L.Aceto, W. Fokking, C. Verhoef: *Structural Operational Semantics*, in J.A. Bergstra, A. Ponse and S.A. Smolka, editors, Handbook of Process Algebra, Elsevier Science, 197 - 292, 2001.

[3] J. Adámek *Free Algebras and automata realisations in the language of categories*, Comment. Math. Univ. Carolinae 15:589-602, 1974.

[4] J.C.M. Baeten, *Embedding untimed into timed process algebra; the case for explicit termination*, in L. Aceto and B. Victor, editors, Proc. EXPRESS'00, State College, Pa., USA, ELNTCS 39, Elsevier, pp.45-62, 2000. http://www.elsevier.nl/locate/entcs/volume39.html. Extended version to appear in MSCS.

[5] J.C.M. Baeten, E.P. de Vink: *Axiomatizing GSOS with Termination*, Proc. STACS 2002, H. Alt and A. Ferreira (eds.), LNCS, to appear.

[6] J.C.M.Baeten, W.P. Weijland: *Process Algebra*, Cambridge University Press, 1990.

[7] J.W. de Bakker and E.P. de Vink: *Control Flow Semantics*, Foundations of Computing Series, The MIT Press, 1996.

[8] G. Birkhoff: *On the structure of abstract algebras*, Proceedings of the Cambridge Philosophical Society, 31: 433-454, 1935.

[9] B. Bloom, S. Istrail, A.R. Meyer: *Bisimulation can't be traced*, Journal of the ACM, 42: 232-268, 1995.

[10] S. Burris: *Free Algebras as subdirect products*, Algebra Universalis 11, 113-134, 1980.

[11] P.M. Cohn: *Universal Algebra*, D. Reidel Publishing Company, New York, 1980.

[12] R. De Simone: *Higher level synchronizing devices in MEIJE-SCCS*, Theoretical Computer Science, 37: 245-267, 1985

[13] W.J. Fokkink: *The tyft/tyxt format reduces to tree rules,* in (M. Hagiya and J.C. Mitchell, eds), Proc. TACS'94, LNCS 789, pp. 440-453, 1994.

[14] Wan Fokkink and Rob van Glabbeek: *Ntyft/ntyxt rules reduce to ntree rules*, Information and Computation, 126(1):1-10, 10 April 1996.

[15] J.F. Groote, F.W. Vaandrager: *Structured operational semantics and bisimulation as a congruence*, Information and Computation, 100(2): 202-260, 1992.

[16] G. Huet:*Confluent Reductions: Abstract Properties and Applications to Term rewritting*, Journal of the ACM, Vol.27, 803-821, 1980.

[17] J.Jezek, *Free Groupoids in Varieties Determined By a Short Equation*, Acta Universitatis Carolinae - Math. et Phys. Vol.23 $N_0$1, 1982.

[18] S. Markovski, A. Sokolova: *Free Basic Process Algebra*, Contributions to General Algebra 11, Klagenfurt 1999.

[19] S. Markovski, A. Sokolova: *Free Steiner Loops*, Glasnik Matematički 36:85-93, 2001.

[20] R. Milner: *A Calculus of Communicating Systems*, LNCS 92, 1980.

[21] R. Milner: *Communication and Concurrency,* Prentice Hall International, 1989.

[22] R. N. McKenzie, W. F. Taylor, G. F. McNulty: *Algebras, Lattices, Varieties*, Wadsworth & Brooks/Cole Advanced Books & Software, 1987.

[23] G. Plotkin: *A structural approach to operational semantics*, Technical Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

[24] D. M. Smirnov: *Mnogoobraziya Algebr*, Nauka, Novosibirsk, 1992.

[25] I. Ulidowski: *Finite axiom systems for testing preorder and De Simone process languages,* Theoretical Computer Science 239:97–139, 2000.

[26] C. Verhoef: *A congruence theorem for structured operational semantics with predicates and negative premisses*, Nordic Journal of Computing 2:274-302, 1995.