SOS Rule Formats for Idempotent Terms and Idempotent Unary Operators*

Luca Aceto[1,2], Eugen-Ioan Goriac[1], and Anna Ingolfsdottir[1]

[1] ICE-TCS, School of Computer Science, Reykjavik University, Iceland
[2] SysMA, IMT Lucca Institute for Advanced Studies, Lucca 55100, Italy

Abstract. A unary operator f is idempotent if the equation $f(x) = f(f(x))$ holds. On the other end, an element a of an algebra is said to be an idempotent for a binary operator $\odot$ if $a = a \odot a$. This paper presents a rule format for Structural Operational Semantics that guarantees that a unary operator be idempotent modulo bisimilarity. The proposed rule format relies on a companion one ensuring that certain terms are idempotent with respect to some binary operator. This study also offers a variety of examples showing the applicability of both formats.

1 Introduction

Over the last three decades, Structural Operational Semantics (SOS) [30] has proven to be a flexible and powerful way to specify the semantics of programming and specification languages. In this approach to semantics, the behaviour of syntactically correct language expressions is given in terms of a collection of state transitions that is specified by means of a set of syntax-driven inference rules. This behavioural description of the semantics of a language essentially tells one how the expressions in the language under definition behave when run on an idealized abstract machine.

Language designers often have expected algebraic properties of language constructs in mind when defining a language. For example, in the field of process algebras such as ACP [10], CCS [26] and CSP [22], operators such as nondeterministic and parallel composition are often meant to be commutative and associative with respect to bisimilarity [29]. Once the semantics of a language has been given in terms of state transitions, a natural question to ask is whether the intended algebraic properties do hold modulo the notion of behavioural semantics of interest. The typical approach to answer this question is to perform an *a posteriori verification*: based on the semantics in terms of state transitions, one proves the validity of the desired algebraic laws, which describe the expected semantic properties of the various operators in the language. An alternative approach is to ensure the validity of algebraic properties *by design*, using the so

* The authors have been partially supported by the project 'Meta-theory of Algebraic Process Theories' (nr. 100014021) of the Icelandic Research Fund. Eugen-Ioan Goriac is also funded by the project 'Extending and Axiomatizing Structural Operational Semantics: Theory and Tools' (nr. 1102940061) of the Icelandic Research Fund.

called *SOS rule formats* [9]. In this approach, one gives *syntactic templates* for the inference rules used in defining the operational semantics for certain operators that guarantee the validity of the desired laws, thus obviating the need for an a posteriori verification. (See [3,5,6,9,16,27] for examples of rule formats for algebraic properties in the literature on SOS.) The definition of SOS rule formats is based on finding a reasonably good trade-off between generality and ease of application. On the one hand, one strives to define a rule format that can capture as many examples from the literature as possible, including ones that may arise in the future. On the other, the rule format should be as easy to apply, and as syntactic, as possible.

The main advantage of the approach based on the development of rule formats is that one is able to verify the desired algebraic properties by syntactic checks that can be mechanized. Moreover, it is interesting to use rule formats for establishing semantic properties since the results so obtained apply to a broad class of languages. Last, but not least, these formats provide one with an understanding of the semantic nature of algebraic properties and of their connection with the syntax of SOS rules. This insight may serve as a guideline for language designers who want to ensure, a priori, that the constructs of a language under design enjoy certain basic algebraic properties.

Contribution The main aim of this paper is to present a format of SOS rules that guarantees that some unary operation f is *idempotent* with respect to any notion of behavioural equivalence that includes bisimilarity. A unary operator f is idempotent if the equation $f(x) = f(f(x))$ holds. Examples of idempotent unary operators from the fields of language theory and process calculi are the unary Kleene star operator [23], the delay operator from SCCS [21,25], the replication operator from the π-calculus [33] and the priority operator from [11].

It turns out that, in order to develop a rule format for unary idempotent operations that can deal with operations such as Kleene star and replication, one needs a companion rule format ensuring that terms of a certain form are idempotent for some binary operator. We recall that an element a of an algebra is said to be an *idempotent* for a binary operator $\odot$ if $a = a \odot a$. For example, the term x^*, where * denotes the Kleene star operation, is an idempotent for the sequential composition operation '$\cdot$' because the equation $x^* = x^* \cdot x^*$ holds. As a second contribution of this paper, we therefore offer an SOS rule format ensuring that certain terms are idempotent with respect to some binary operator. Both the rule formats we present in this paper make an essential use of previously developed formats for algebraic properties such as associativity and commutativity [16,27].

We provide a variety of examples showing that our rule formats can be used to establish the validity of several laws from the literature on process algebras dealing with idempotent unary operators and idempotent terms.

Roadmap of the paper The paper is organized as follows. Section 2 reviews some standard definitions from the theory of SOS that will be used in the remainder of this study. We present our rule format for idempotent terms in Section 3.

That rule format plays an important role in the definition of the rule format for idempotent unary operators that we give in Section 4. We discuss the results of the paper and hint at directions for future work in Section 5.

2 Preliminaries

In this section we review, for the sake of completeness, some standard definitions from process theory and the meta-theory of SOS that will be used in the remainder of the paper. We refer the interested reader to [7,28] for further details.

Transition System Specifications in GSOS Format

Definition 1 (Signature, terms and substitutions). *We let V denote an infinite set of variables with typical members $x, x', x_i, y, y', y_i, \ldots$. A* signature *$\Sigma$ is a set of function symbols, each with a fixed arity. We call these symbols* operators *and usually represent them by $f, g, \ldots$. An operator with arity zero is called a* constant. *We define the set $\mathbb{T}(\Sigma)$ of* terms *over Σ (sometimes referred to as Σ-terms) as the smallest set satisfying the following constraints.*

- *A variable $x \in V$ is a term.*
- *If $f \in \Sigma$ has arity n and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term.*

We use $t, t', t_i, u, \ldots$ to range over terms. We write $t_1 \equiv t_2$ if t_1 and t_2 are syntactically equal. The function $vars : \mathbb{T}(\Sigma) \to 2^V$ gives the set of variables appearing in a term. The set $\mathbb{C}(\Sigma) \subseteq \mathbb{T}(\Sigma)$ is the set of closed terms*, i.e., the set of all terms t such that $vars(t) = \emptyset$. We use $p, p', p_i, q, \ldots$ to range over closed terms. A* context *is a term with an occurrence of a hole [ ] in it.*

A substitution *σ is a function of type $V \to \mathbb{T}(\Sigma)$. We extend the domain of substitutions to terms homomorphically. If the range of a substitution is included in $\mathbb{C}(\Sigma)$, we say that it is a* closed substitution*. For a substitution σ and sequences $x_1, \ldots, x_n$ and $t_1, \ldots, t_n$ of distinct variables and of terms, respectively, we write $\sigma[x_1 \mapsto t_1, \ldots, x_n \mapsto t_n]$ for the substitution that maps each variable x_i to t_i $(1 \leq i \leq n)$ and agrees with σ on all of the other variables. When σ is the identity function over variables, we abbreviate $\sigma[x_1 \mapsto t_1, \ldots, x_n \mapsto t_n]$ to $[x_1 \mapsto t_1, \ldots, x_n \mapsto t_n]$.*

The GSOS format is a widely studied format of deduction rules in transition system specifications proposed by Bloom, Istrail and Meyer [14,15]. Transition system specifications whose rules are in the GSOS format enjoy many desirable properties, and several studies in the literature on the meta-theory of SOS have focused on them—see, for instance, [2,1,4,7,8,12]. In this study we shall also focus on transition system specifications in the GSOS format, which we now proceed to define.

Definition 2 (GSOS Format [15]). *A deduction rule for an operator f of arity n is in the* GSOS format *if and only if it has the following form:*

$$\frac{\{x_i \xrightarrow{l_{ij}} y_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq m_i\} \cup \{x_i \overset{l_{ik}}{\not\to} \mid 1 \leq i \leq n, 1 \leq k \leq n_i\}}{f(\boldsymbol{x}) \xrightarrow{l} C[\boldsymbol{x}, \boldsymbol{y}]} \tag{1}$$

where the x_i's and the y_{ij}'s ($1 \leq i \leq n$ and $1 \leq j \leq m_i$) are all distinct variables, m_i and n_i are natural numbers, $C[\boldsymbol{x}, \boldsymbol{y}]$ is a Σ-term with variables including at most the x_i's and y_{ij}'s, and l_{ij}'s and l are labels. The above rule is said to be f-defining *and* l-emitting.

A transition system specification (TSS) in the GSOS format $\mathcal{T}$ *is a triple (Σ, L, D) where Σ is a finite signature, L is a finite set of labels, and D is a finite set of deduction rules in the GSOS format. The collection of f-defining and l-emitting rules in a set D of GSOS rules is denoted by $D(f, l)$.*

Example 1. An example of a TSS in the GSOS format is the one describing the semantics of BCCSP [20]. The signature for this TSS contains the operators $\mathbf{0}$ (of arity zero), $a.\_$ ($a \in L$) and $\_+\_$. The standard deduction rules for these operators are listed below, where a ranges over L.

$$\frac{}{a.x_1 \xrightarrow{a} x_1} \qquad \frac{x_1 \xrightarrow{a} x_1'}{x_1 + x_2 \xrightarrow{a} x_1'} \qquad \frac{x_2 \xrightarrow{a} x_2'}{x_1 + x_2 \xrightarrow{a} x_1'}$$

Informally, the intent of a GSOS rule of the form (1) is as follows. Suppose that we are wondering whether $f(\boldsymbol{p})$ is capable of taking an l-step. We look at each f-defining and l-emitting rule in turn. We inspect each positive premise $x_i \xrightarrow{l_{ij}} y_{ij}$, checking if p_i is capable of taking an l_{ij}-step for each j and if so calling the l_{ij}-children q_{ij}. We also check the negative premises: if p_i is *in*capable of taking an l_{ik}-step for each k. If so, then the rule *fires* and $f(\boldsymbol{p}) \xrightarrow{l} C[\boldsymbol{p}, \boldsymbol{q}]$. This means that the transition relation $\longrightarrow$ associated with a TSS in the GSOS format is the one defined by the rules using structural induction over closed Σ-terms. This transition relation is the unique sound and supported transition relation. Here *sound* means that whenever a closed substitution σ 'satisfies' the premises of a rule of the form (1), then $\sigma(f(\boldsymbol{x})) \xrightarrow{l} \sigma(C[\boldsymbol{x}, \boldsymbol{y}])$. On the other hand, *supported* means that any transition $p \xrightarrow{l} q$ can be obtained by instantiating the conclusion of a rule of the form (1) with a substitution that satisfies its premises. We refer the interested reader to [14,15] for the precise definition of $\longrightarrow$ and much more information on GSOS languages. The above informal description of the transition relation associated with a TSS in GSOS format suffices to follow the technical developments in the remainder of the paper.

Remark 1. In this paper, we restrict ourselves to TSSs in GSOS format for the sake of simplicity. The rule formats we present in what follows can be extended to arbitrary TSSs at the price of considering the so-called *three-valued stable models*. See [7] for a survey introduction to three-valued stable models.

Bisimilarity Terms built using operators from the signature of a TSS are usually considered modulo some notion of behavioural equivalence, which is used to indicate when two terms describe 'essentially the same behaviour'. The notion of behavioural equivalence that we will use in this paper is the following, classic notion of bisimilarity [26,29].

Definition 3 (Bisimilarity). *Let $\mathcal{T}$ be a TSS in GSOS format with signature Σ. A relation $R \subseteq \mathbb{C}(\Sigma) \times \mathbb{C}(\Sigma)$ is a* bisimulation *if and only if R is symmetric and, for all $p_0, p_1, p_0' \in \mathbb{C}(\Sigma)$ and $l \in L$,*

$$(p_0 \ R \ p_1 \wedge p_0 \xrightarrow{l} p_0') \Rightarrow \exists p_1' \in \mathbb{C}(\Sigma).\ (p_1 \xrightarrow{l} p_1' \wedge p_0' \ R \ p_1').$$

Two terms $p_0, p_1 \in \mathbb{C}(\Sigma)$ are called bisimilar*, denoted by $\mathcal{T} \vdash p_0 \leftrightarroweq p_1$ (or simply by $p_0 \leftrightarroweq p_1$ when $\mathcal{T}$ is clear from the context), when there exists a bisimulation R such that $p_0 \ R \ p_1$. We refer to the relation $\leftrightarroweq$ as* bisimilarity*.*

It is well known that $\leftrightarroweq$ is an equivalence relation over $\mathbb{C}(\Sigma)$. Any equivalence relation $\sim$ over closed terms in a TSS $\mathcal{T}$ is extended to open terms in the standard fashion, i.e., for all $t_0, t_1 \in \mathbb{T}(\Sigma)$, the equation $t_0 = t_1$ holds over $\mathcal{T}$ modulo $\sim$ (sometimes abbreviated to $t_0 \sim t_1$) if, and only if, $\mathcal{T} \vdash \sigma(t_0) \sim \sigma(t_1)$ for each closed substitution σ.

Definition 4. *Let Σ be a signature. An equivalence relation $\sim$ over Σ-terms is a* congruence *if, for all $f \in \Sigma$ and closed terms $p_1, \ldots, p_n, q_1, \ldots, q_n$, where n is the arity of f, if $p_i \sim q_i$ for each $i \in \{1, \ldots, n\}$ then $f(p_1, \ldots, p_n) \sim f(q_1, \ldots, q_n)$.*

Remark 2. Let Σ be a signature and let $\sim$ be a congruence. It is easy to see that, for all $f \in \Sigma$ and terms $t_1, \ldots, t_n, u_1, \ldots, u_n$, where n is the arity of f, if $t_i \sim u_i$ for each $i \in \{1, \ldots, n\}$ then $f(t_1, \ldots, t_n) \sim f(u_1, \ldots, u_n)$.

The following result is well known [14].

Proposition 1. *$\leftrightarroweq$ is a congruence for any TSS in GSOS format.*

The above proposition is a typical example of a result in the meta-theory of SOS: it states that if the rules in a TSS satisfy some syntactic constraint, then some semantic result is guaranteed to hold. In the remainder of this paper, following the work presented in, e.g., [3,5,6,9,16,27], we shall present a rule format ensuring that certain unary operations are idempotent. This rule format will rely on one yielding that terms of a certain form are idempotent for some binary operator. For this reason, we present first the latter rule format in the subsequent section.

3 A rule format for idempotent terms

Definition 5 (Idempotent term). *Let Σ be a signature. Let f and $\odot$ be, respectively, a unary and a binary operator in Σ. We say that $f(x)$ is an* idempotent term *for $\odot$ with respect to an equivalence relation $\sim$ over $\mathbb{T}(\Sigma)$ if the following equation holds:*

$$f(x) \sim f(x) \odot f(x). \tag{2}$$

In what follows, we shall present some syntactic requirements on the SOS rules defining the operators f and $\odot$ that guarantee the validity of equation (2). In order to motivate the syntactic constraints of the rule format, let us consider

the unary replication operator '!', which is familiar from the theory of the π-calculus (see, e.g., [33]), and the binary interleaving parallel composition '||', which appears in, amongst others, ACP [13], CCS [24], and CSP [22,31]. The rules for these operators are given below, where a ranges over the set of action labels L.

$$\frac{x \stackrel{a}{\longrightarrow} x'}{!x \stackrel{a}{\longrightarrow} x' \,||!x} \qquad \frac{x \stackrel{a}{\longrightarrow} x'}{x \,||\, y \stackrel{a}{\longrightarrow} x' \,||\, y} \qquad \frac{y \stackrel{a}{\longrightarrow} y'}{x \,||\, y \stackrel{a}{\longrightarrow} x \,||\, y'} \tag{3}$$

It is well known that $!x$ is an idempotent term for $||$ modulo any notion of equivalence that includes bisimilarity. Indeed, the equation

$$!x = (!x) \,||(!x)$$

is one of the laws for the structural congruence over the π-calculus with replication considered in, e.g., [17].

It is instructive to try and find out why the above law holds by considering the interplay between the transition rules for '!' and those for '||', rather than considering the transitions that are possible for all the closed instantiations of the terms $!x$ and $(!x) \,||(!x)$. To this end, consider the rule for replication for some action a. The effect of this rule can be mimicked by the term $(!x) \,||(!x)$ by means of a combination of the instance of the first rule for $||$ in (3) for action a and of the rule for replication. When we do so, the appropriate instantiation of the target of the conclusion of the first rule for $||$ is the term

$$(x' \,||\, y)[x' \mapsto x' \,||!x,\ y \mapsto !x] = (x' \,||!x) \,||!x.$$

Note that the target of the conclusion of the rule for replication, namely $x' \,||!x$, and the above term can be proved equal using associativity of $||$, which is well known, and the version of axiom (2) for replication and parallel composition, as follows:

$$(x' \,||!x) \,||!x = x' \,||(!x \,||!x) = x' \,||!x.$$

The validity of the associativity law for $||$ is guaranteed by the rule format for associativity given in [16, Definition 8]. On the other hand, the soundness of the use of equation (2) can be justified using coinduction [32].

Consider instead the combination of the instance of the second rule for $||$ in (3) for action a and of the rule for replication. When we do so, the appropriate instantiation of the target of the conclusion of the second rule for $||$ is the term

$$(x \,||\, y')[x \mapsto !x,\ y \mapsto x' \,||!x] = !x \,||(x' \,||!x).$$

Note that the target of the conclusion of the rule for replication, namely $x' \,||!x$, and the above term can be proved equal using commutativity and associativity of $||$, which are well known, and the version of axiom (2) for replication and parallel composition, as follows:

$$!x \,||(x' \,||!x) = (x' \,||!x) \,||!x = x' \,||(!x \,||!x) = x' \,||!x.$$

The validity of the commutativity law for $||$ is guaranteed by the rule format for commutativity given in [27].

The above discussion hints at the possibility of defining an SOS rule format guaranteeing the validity of equation (2) building on SOS rule formats for algebraic properties like associativity and commutativity of operators [9], and on a coinductive use of equation (2) itself. The technical developments to follow will offer a formalization of this observation.

Our definition of the rule format is based on a syntactically defined equivalence relation over terms that is sufficient to handle the examples from the literature we have met so far.

Definition 6 (The relation $\leftrightsquigarrow$). *Let $\mathcal{T} = (\Sigma, L, D)$ be a TSS in GSOS format.*

1. *The relation $\leftrightsquigarrow$ is the least equivalence relation over $\mathbb{T}(\Sigma)$ that satisfies the following clauses:*
   - *$f(t,u) \leftrightsquigarrow f(u,t)$, if f is a binary operator in Σ and the commutativity rule format from [27] applies to f,*
   - *$f(t, f(t',u)) \leftrightsquigarrow f(f(t,t'),u)$, if f is a binary operator in Σ and one of the associativity rule formats from [16] applies to f, and*
   - *$C[t] \leftrightsquigarrow C[t']$, if $t \leftrightsquigarrow t'$, for each context $C[\ ]$.*
2. *Let f and $\odot$ be, respectively, a unary and a binary operator in Σ. We write $t{\downarrow_{f,\odot}}u$ if, and only if, there are some t' and u' such that $t \leftrightsquigarrow t'$, $u \leftrightsquigarrow u'$, and $t' = u'$ can be proved by possibly using one application of an instantiation of axiom (2) in a context—that is, either $t' \equiv u'$, or $t' = C[f(t'')]$ and $u' = C[f(t'') \odot f(t'')]$, for some context $C[\ ]$ and term t'', or vice versa.*

Example 2. Consider the terms $!x\,||(x'\,||!x)$ and $x'\,||!x$. Then

$$x'\,||!x \ \downarrow_{!,||}\ !x\,||(x'\,||!x).$$

Indeed, $!x\,||(x'\,||!x) \leftrightsquigarrow x'\,||(!x\,||!x)$, because the rules for $||$ are in the associativity and commutativity rule formats from [16,27], and $x'\,||!x = x'\,||(!x\,||!x)$ can be proved using one application of the relevant instance of axiom (2) in the context $x'\,||[\ ]$.

Lemma 1. *Let $\mathcal{T} = (\Sigma, L, D)$ be a TSS in the GSOS format, and let $t, t' \in \mathbb{T}(\Sigma)$. If $t \leftrightsquigarrow t'$ then*

1. *$t \underline{\leftrightarrow} t'$,*
2. *$vars(t) = vars(t')$ and*
3. *$\sigma(t) \leftrightsquigarrow \sigma(t')$, for each substitution σ.*

Proof. All the claims can be shown by induction on the definition of $\leftrightsquigarrow$. The soundness modulo $\underline{\leftrightarrow}$ of the rewrite rules in Definition 6(1) is guaranteed by results in [16,27] and by Proposition 1. □

Remark 3. The definition of the relation $\leftrightsquigarrow$ can be easily strengthened by adding more clauses, provided their soundness with respect to bisimilarity can be 'justified syntactically'.

We are now ready to present an SOS rule format guaranteeing the validity of equation (2).

Definition 7 (Rule format for idempotent terms). *Let $\mathcal{T} = (\Sigma, L, D)$ be a TSS in the GSOS format. Let f and $\odot$ be, respectively, a unary and a binary operator in Σ. We say that the rules for f and $\odot$ in $\mathcal{T}$ are in the* rule format for idempotent terms *if either $\odot$ is in the rule format for idempotence from [3] or the following conditions are met:*

1. *For each f-defining rule in D, say*

$$\frac{H}{f(x) \xrightarrow{a} t},$$

there is some $\odot$-defining rule

$$\frac{H'}{x_1 \odot x_2 \xrightarrow{a} u},$$

such that
(a) $H' \subseteq \{x_1 \xrightarrow{a} y_1,\ x_2 \xrightarrow{a} y_2\}$, and
(b) $t \downarrow_{f,\odot} u[x_1 \mapsto f(x), x_2 \mapsto f(x), y_1 \mapsto t, y_2 \mapsto t]$.

2. *Each $\odot$-defining rule has the form*

$$\frac{\{x_i \xrightarrow{a} y_i\} \cup \{x_1 \xrightarrow{a_j} y_j \mid j \in J\} \cup \{x_2 \xrightarrow{b_k} z_k \mid k \in K\}}{x_1 \odot x_2 \xrightarrow{a} u}, \tag{4}$$

where $i \in \{1, 2\}$, J and K are index sets, and $vars(u) \subseteq \{x_1, x_2, y_i\}$.

3. *Let r be an $\odot$-defining rule of the form (4) such that $D(f, a_j) \neq \emptyset$, for each $j \in J$, and $D(f, b_k) \neq \emptyset$, for each $k \in K$. Let*

$$\frac{H}{f(x) \xrightarrow{a} t}$$

be a rule for f. Then

$$t \downarrow_{f,\odot} u[x_1 \mapsto f(x), x_2 \mapsto f(x), y_i \mapsto t].$$

Theorem 1. *Let $\mathcal{T} = (\Sigma, L, D)$ be a TSS in the GSOS format. Let f and $\odot$ be, respectively, a unary and a binary operator in Σ. Assume that the rules for f and $\odot$ in $\mathcal{T}$ are in the rule format for idempotent terms. Then equation (2) holds over $\mathcal{T}$ modulo bisimilarity.*

The proof of this result may be found in Appendix A.

Example 3 (Replication and parallel composition). The unary replication operator and the parallel composition operator, whose rules we presented in (3), are in the rule format for idempotent terms. Indeed, we essentially carried out the verification of the conditions in Definition 7 when motivating the constraints of the rule format and the relation $\leftrightsquigarrow$. Therefore, Theorem 1 yields the soundness, modulo bisimilarity, of the well-known equation

$$!x = (!x) \,||\, (!x).$$

Example 4 (Kleene star and sequential composition). Assume that the set of action labels L contains a distinguished action $\checkmark$, which is used to signal the successful termination of a process.

Consider the unary Kleene star operator '$\_^*$' [23] and the binary sequential composition operator '$\cdot$' given by the rules

$$\frac{}{x^* \xrightarrow{\checkmark} \delta} \qquad \frac{x \xrightarrow{a} x'}{x^* \xrightarrow{a} x' \cdot (x^*)} \; (a \neq \checkmark)$$

$$\frac{x \xrightarrow{a} x'}{x \cdot y \xrightarrow{a} x' \cdot y} \; (a \neq \checkmark) \qquad \frac{x \xrightarrow{\checkmark} x', \;\; y \xrightarrow{a} y'}{x \cdot y \xrightarrow{a} y'} \; (a \in L),$$

where δ is a constant. These rules satisfy the requirements of the rule format for idempotent terms. Indeed, to verify condition 1 in Definition 7, observe that both rules for the Kleene star operator can be 'matched' by the equally-labelled instance of the second rule schema for $\cdot$, because

$$t \equiv y'[x \mapsto x^*, y \mapsto x^*, x' \mapsto t, y' \mapsto t],$$

where t is the target of the conclusion of the rule for the Kleene star operator.

To check conditions 2–3, we examine all pairs of equally-labelled rules for $\cdot$ and the Kleene star operator. By way of example, let us look the pair consisting of the rules

$$\frac{x \xrightarrow{a} x'}{x \cdot y \xrightarrow{a} x' \cdot y} \; (a \neq \checkmark) \qquad \frac{x \xrightarrow{a} x'}{x^* \xrightarrow{a} x' \cdot (x^*)} \; (a \neq \checkmark).$$

Then

$$\begin{aligned} (x' \cdot y)[x \mapsto x^*, y \mapsto x^*, x' \mapsto x' \cdot (x^*), y' \mapsto x' \cdot (x^*)] \;&=\; (x' \cdot (x^*)) \cdot x^* \\ &\leftrightsquigarrow\; x' \cdot (x^* \cdot x^*) \\ &=\; x' \cdot (x^*) \qquad \text{by (2).} \end{aligned}$$

(The second step in the above argument is justified since the associativity rule format from [16, Definition 10] applies to $\cdot$.) Therefore, Theorem 1 yields the soundness, modulo bisimilarity, of the well-known equation

$$x^* = x^* \cdot x^*.$$

Example 5 (Perpetual loop and sequential composition). Consider the unary perpetual loop operator '$\_^\omega$' from [19] given by the rules

$$\frac{x \xrightarrow{a} x'}{x^\omega \xrightarrow{a} x' \cdot (x^\omega)} \ (a \neq \checkmark)$$

and the binary sequential composition operator '$\cdot$' given by the rules in Example 4. These rules satisfy the requirements of the rule format for idempotent terms. The conditions in Definition 7 can be checked along the lines of Example 4. Note that, since there is no $\checkmark$-emitting rule for the perpetual loop operator, the premises of rules of the form

$$\frac{x \xrightarrow{\checkmark} x',\ y \xrightarrow{a} y'}{x \cdot y \xrightarrow{a} y'} \ (a \in L)$$

cannot be all met when x is instantiated with x^ω and therefore condition 3 in Definition 7 holds vacuously for those rules. Thus, Theorem 1 yields the soundness, modulo bisimilarity, of the well-known equation

$$x^\omega = x^\omega \cdot x^\omega.$$

4 A rule format for idempotent unary operators

Definition 8 (Idempotent unary operator). *Let Σ be a signature. Let f be a unary operator in Σ. We say that $f(x)$ is* idempotent *with respect to an equivalence relation $\sim$ over $\mathbb{T}(\Sigma)$ if the following equation holds:*

$$f(x) \sim f(f(x)). \tag{5}$$

Example 6 (Delay operator). The rules for the unary delay operator δ from SCCS [21,25] are as follows, where 1 is a distinguished action symbol in L that is used to denote a delay of one time unit:

$$\frac{}{\delta x \xrightarrow{1} \delta x} \qquad \frac{x \xrightarrow{a} x'}{\delta x \xrightarrow{a} x'} \ (a \in L).$$

It is well known that δ is idempotent modulo bisimilarity.

Example 7 (Prefix iteration). The delay operator we presented in Example 6 is a special case of the unary prefix iteration operator $a^*\_$ from [18]. The rules for this operator are as follows:

$$\frac{}{a^* x \xrightarrow{a} a^* x} \qquad \frac{x \xrightarrow{b} x'}{a^* x \xrightarrow{b} x'} \ (b \in L).$$

It is well known that $a^*\_$ is idempotent modulo bisimilarity.

In what follows, we shall present some syntactic requirements on the SOS rules defining a unary operator f that guarantee the validity of equation (5). The rule format for idempotent unary operators will rely on the one for idempotent terms given in Definition 7.

In order to motivate the use of the rule format for idempotent terms in the definition of the one for idempotent unary operators, consider the replication operator whose rules were introduced in (3). As is well known, the equation

$$!x =!(!x)$$

holds modulo bisimilarity. The validity of this equation can be 'justified' using the transition rules for '!' as follows. Consider the rule for replication for some action a, namely

$$\frac{x \stackrel{a}{\longrightarrow} x'}{!x \stackrel{a}{\longrightarrow} x' \,||!x}.$$

The effect of this rule can be mimicked by the term $!(!x)$ by using the same rule twice. When we do so, the appropriate instantiation of the target of the conclusion of the rule for '!' is the term

$$(x' \,||!x)[x' \mapsto x' \,||!x,\ x \mapsto !x] = (x' \,||!x) \,||!(!x).$$

Note that the target of the conclusion of the rule for replication, namely $x' \,||!x$, and the above term can be proved equal using associativity of $||$, the version of axiom (2) for replication and parallel composition, and the version of axiom (5) for replication, as follows:

$$(x' \,||!x) \,||!(!x) = (x' \,||!x) \,||!x = x' \,||(!x \,||!x) = x' \,||!x.$$

As mentioned in Example 3, the validity of the version of axiom (2) for replication and parallel composition is guaranteed by Theorem 1. On the other hand, the soundness of the use of equation (5) can be justified using coinduction [32].

As we did in the definition of the rule format for idempotent terms presented in Definition 7, in stating the requirements of the rule format for idempotent unary operators, we shall employ a syntactically defined equivalence relation over terms that is sufficient to handle the examples from the literature we have met so far.

Definition 9 (The relation $\leftrightarrow$). *Let $\mathcal{T} = (\Sigma, L, D)$ be a TSS in the GSOS format.*

1. *The relation $\leftrightarrow$ is the least equivalence relation over $\mathbb{T}(\Sigma)$ that satisfies the following clauses:*
   - *$f(t) \leftrightarrow f(t) \odot f(t)$, if the rules for f and $\odot$ in $\mathcal{T}$ are in the rule format for idempotent terms from Definition 7,*
   - *$f(t, u) \leftrightarrow f(u, t)$, if f is a binary operator in Σ and the commutativity rule format from [27] applies to f,*

- $f(t, f(t', u)) \leftrightarrow f(f(t, t'), u)$, *if* f *is a binary operator in* Σ *and one of the associativity rule formats from [16] applies to* f, *and*
- $C[t] \leftrightarrow C[t']$, *if* $t \leftrightarrow t'$, *for each context* $C[\ ]$.

2. *Let* f *be a unary operator in* Σ. *We write* $t \Downarrow_f u$ *if, and only if, there are some* t' *and* u' *such that* $t \leftrightarrow t'$, $u \leftrightarrow u'$, *and* $t' = u'$ *can be proved by possibly using one application of an instantiation of axiom (5) in a context—that is, either* $t' \equiv u'$, *or* $t' = C[f(t'')]$ *and* $u' = C[f(f(t''))]$, *for some context* $C[\ ]$ *and term* t'', *or vice versa.*

Example 8. Consider the terms $(x' \,||\, !x) \,||\, !(!x)$ and $x' \,||\, !x$. Then

$$x' \,||\, !x \Downarrow_! (x' \,||\, !x) \,||\, !(!x).$$

Indeed, $x' \,||\, !x \leftrightarrow (x' \,||\, !x) \,||\, !x$, using the relevant instance of axiom (2) and associativity of $||$, and $(x' \,||\, !x) \,||\, !x = (x' \,||\, !x) \,||\, !(!x)$ can be proved using one application of the relevant instance of axiom (5) in the context $(x' \,||\, !x) \,||\, [\ ]$.

Lemma 2. *Let* $\mathcal{T} = (\Sigma, L, D)$ *be a TSS in the GSOS format, and let* $t, t' \in \mathbb{T}(\Sigma)$. *If* $t \leftrightarrow t'$ *then*

1. $t \leftrightarrows t'$,
2. $vars(t) = vars(t')$ *and*
3. $\sigma(t) \leftrightarrow \sigma(t')$, *for each substitution* σ.

Proof. The lemma can be shown along the lines of the proof of Lemma 1, using Theorem 1 to justify the soundness modulo bisimilarity of applications of the clause $f(t) \leftrightarrow f(t) \odot f(t)$, when the rules for f and $\odot$ in $\mathcal{T}$ are in the rule format for idempotent terms from Definition 7. □

Definition 10 (Rule format for idempotent unary operators). *Let* $\mathcal{T} = (\Sigma, L, D)$ *be a TSS in the GSOS format. Let* f *be a unary operator in* Σ. *We say that the rules for* f *are in the rule format for* idempotent unary operators *if the following conditions are met:*

1. *Each rule for* f *in* D *has the form*

$$\frac{H \cup \{x \overset{b_j}{\nrightarrow} \mid j \in J\}}{f(x) \xrightarrow{a} t}, \tag{6}$$

   where
   (a) $H \subseteq \{x \xrightarrow{a} x'\}$ *and*
   (b) $H = \{x \xrightarrow{a} x'\}$ *if* J *is non-empty.*
2. *If some rule for* f *of the form (6) has a premise of the form* $x \overset{b}{\nrightarrow}$, *then each* b*-emitting and* f*-defining rule has a positive premise of the form* $x \xrightarrow{b} x'$.
3. *Consider a rule for* f *of the form (6). Then*
   (a) *either* H *is empty and* $t \Downarrow_f t[x \mapsto f(x)]$

(b) or $H = \{x \xrightarrow{a} x'\}$ *and, for each* a*-emitting rule for* f

$$\frac{H'}{f(x) \xrightarrow{a} t'},$$

we have that

$$t' \Downarrow_f t[x \mapsto f(x), x' \mapsto t'].$$

Theorem 2. *Let* $\mathcal{T} = (\Sigma, L, D)$ *be a TSS in the GSOS format. Let* f *be a unary operator in* Σ*. Assume that the rules for* f *in* $\mathcal{T}$ *are in the rule format for idempotent unary operators. Then equation (5) holds over* $\mathcal{T}$ *modulo bisimilarity.*

The proof of this result may be found in Appendix B.

Remark 4. Condition 1b in Definition 10 requires that, in rules of the form (6), $H = \{x \xrightarrow{a} x'\}$ if J is non-empty. This requirement is necessary for the validity of Theorem 2. To see this, consider the unary operator f with rules

$$\frac{x \overset{b}{\not\to}}{f(x) \xrightarrow{a} \mathbf{0}} \qquad \frac{x \xrightarrow{b} x',\ x \overset{c}{\not\to}}{f(x) \xrightarrow{b} \mathbf{0}} \qquad \frac{x \xrightarrow{c} x',\ x \overset{b}{\not\to}}{f(x) \xrightarrow{c} \mathbf{0}}.$$

The rules for f satisfy all the conditions in Definition 10 apart from the requirement in condition 1b that all rules have positive premises when they have negative ones.

It is not hard to see that $f(b.\mathbf{0} + c.\mathbf{0})$ has no outgoing transitions. On the other hand, using the a-emitting rule for f, we have that

$$f(f(b.\mathbf{0} + c.\mathbf{0})) \xrightarrow{a} \mathbf{0}.$$

Therefore, $f(b.\mathbf{0}+c.\mathbf{0}) \not\leftrightarrows f(f(b.\mathbf{0}+c.\mathbf{0}))$, and the equation $f(x) \leftrightarrows f(f(x))$ does not hold.

Example 9 (Delay operator). Consider the delay operator δ introduced in Example 6. Each rule for δ is of the form (6), meeting condition 1 in Definition 10. To see that condition 3 is also met, observe that

- $\delta x \Downarrow_f \delta(\delta x) = (\delta x)[x \mapsto \delta x, x' \mapsto \delta x]$,
- $x' \Downarrow_f x' = x'[x \mapsto \delta x, x' \mapsto x']$, and
- $\delta x \Downarrow_f x'[x \mapsto \delta x, x' \mapsto \delta x]$.

Therefore, Theorem 2 yields the soundness, modulo bisimilarity, of the well-known equation

$$\delta x = \delta(\delta x).$$

The prefix iteration operator discussed in Example 7 is handled in similar fashion.

Example 10 (Encapsulation). Consider the classic unary encapsulation operators ∂_H from ACP [10], where $H \subseteq L$, with rules

$$\frac{x \xrightarrow{a} x'}{\partial_H(x) \xrightarrow{a} \partial_H(x')} \quad a \notin H.$$

It is a simple matter to check that the above rules meet all the conditions in Definition 10. In particular,

$$\partial_H(x') \Downarrow_{\partial_H} \partial_H(\partial_H(x')) = \partial_H(x')[x \mapsto \partial_H(x), x' \mapsto \partial_H(x')].$$

Therefore, Theorem 2 yields the soundness, modulo bisimilarity, of the well-known equation

$$\partial_H(x) = \partial_H(\partial_H(x)).$$

Example 11 (Priority). Assume that $<$ is an irreflexive partial ordering over L. The priority operator θ from [11] has rules

$$\frac{x \xrightarrow{a} x',\ (x \overset{b}{\not\rightarrow} \text{ for each } b \text{ such that } a < b)}{\theta(x) \xrightarrow{a} \theta(x')} \qquad (a \in L).$$

It is not hard to see that the rules for θ satisfy the conditions in Definition 10. In particular, for each $a \in L$, the only a-emitting rule for θ has a positive premise of the form $x \xrightarrow{a} x'$. Hence, condition 1b in Definition 10 is met.

Therefore, Theorem 2 yields the soundness, modulo bisimilarity, of the well-known equation

$$\theta(x) = \theta(\theta(x)).$$

Example 12 (Replication). Consider the replication operator '!' whose rules were given in (3). We claim that the rules for '!' satisfy the conditions in Definition 10. Indeed, the rules for '!' are of the form (6) and, as we observed earlier in Example 8,

$$x' \,||!x \Downarrow_! (x' \,||!x) \,||!(!x) = (x' \,||!x)[x \mapsto !x, x' \mapsto x' \,||!x].$$

Therefore, Theorem 2 yields the soundness, modulo bisimilarity, of the well-known equation

$$!x = !(!x).$$

Example 13 (Kleene star). Consider the unary Kleene star operator '$\_^*$' whose rules were given in Example 4. We claim that the rules for '$\_^*$' satisfy the conditions in Definition 10. Indeed, observe, first of all, that the rules for '$\_^*$' are of the form (6). Moreover,

- $\delta \Downarrow_* \delta[x \mapsto x^*, x' \mapsto \delta]$, and
- $x' \cdot (x^*) \Downarrow_* (x' \cdot (x^*)) \cdot (x^*)^* = (x' \cdot (x^*))[x \mapsto x^*, x' \mapsto x' \cdot (x^*)]$.

The proof of the latter claim uses that the rules for '$\_^*$' and '$\cdot$' satisfy the requirements of the rule format for idempotent terms. Therefore, Theorem 2 yields the soundness, modulo bisimilarity, of the well-known equation

$$x^* = (x^*)^*.$$

The perpetual loop operator discussed in Example 5 is handled in similar fashion.

5 Conclusions

In this study, we have presented an SOS rule format that guarantees that a unary operator is idempotent modulo bisimilarity. In order to achieve a sufficient degree of generality, that rule format relies on a companion one ensuring that certain terms are idempotent with respect to some binary operator. In addition, both rule formats make use of existing formats for other algebraic properties such as associativity [16], commutativity [27] and idempotence for binary operators [3]. In this paper, we have restricted ourselves to TSSs in GSOS format [14,15] for the sake of simplicity. The rule formats we offered in this study can be extended to arbitrary TSSs in standard fashion, provided one gives the semantics of such TSSs in terms of three-valued stable models.

The auxiliary rule format ensuring that certain terms are idempotent with respect to some binary operator may be seen as a refinement of the one from [3]. That paper offered a rule format guaranteeing that certain binary operators are idempotent. We recall that a binary operator $\odot$ is idempotent if the equation $x \odot x = x$ holds. Of course, if a binary operation is idempotent, then any term is an idempotent for it. However, the sequential composition operator '$\cdot$' is *not* idempotent, but the term x^* is an idempotent for it. Similarly, the parallel composition operator '$||$' is *not* idempotent, but the term $!x$ is an idempotent for $||$. Since the laws $x^* \cdot x^* = x^*$ and $!x \,||\, !x = !x$ play an important role in establishing, via syntactic means, that the unary Kleene star and replication operators are idempotent, we needed to develop a novel rule format for idempotent terms in order to obtain a powerful rule format for idempotent unary operations.

To our mind, idempotence of unary operators is the last 'typical' algebraic law for which it is worth developing a specialized rule format. An interesting, long-term research goal is to develop a general approach for synthesizing rule formats for algebraic properties from the algebraic law itself and some assumption on the format of the rules used to give the semantics for the language constructs in the style of SOS. We believe that this is a hard research problem. Indeed, the development of the formats for algebraic properties surveyed in [9] has so far relied on ad-hoc ingenuity and it is hard to discern some common principles that could guide the algorithmic synthesis of such formats.

References

1. L. Aceto. Deriving complete inference systems for a class of GSOS languages generating regular behaviours. In B. Jonsson and J. Parrow, editors, *Proceedings of the fifth International Conference on Concurrency Theory (CONCUR'94)*, volume 836 of *Lecture Notes in Computer Science*, pages 449–464. Springer-Verlag, Berlin, Germany, 1994.
2. L. Aceto. GSOS and finite labelled transition systems. *Theoretical Computer Science*, 131:181–195, 1994.
3. L. Aceto, A. Birgisson, A. Ingólfsdóttir, M. R. Mousavi, and M. A. Reniers. Rule formats for determinism and idempotence. *Science of Computer Programming*, 77(7–8):889–907, 2012.

4. L. Aceto, B. Bloom, and F. W. Vaandrager. Turning SOS rules into equations. *Information and Computation*, 111:1–52, 1994.
5. L. Aceto, M. Cimini, A. Ingólfsdóttir, M. R. Mousavi, and M. A. Reniers. Rule formats for distributivity. In A. H. Dediu, S. Inenaga, and C. Martín-Vide, editors, *Language and Automata Theory and Applications - 5th International Conference, LATA 2011, Tarragona, Spain, May 26–31, 2011. Proceedings*, volume 6638 of *Lecture Notes in Computer Science*, pages 80–91. Springer, 2011.
6. L. Aceto, M. Cimini, A. Ingólfsdóttir, M. R. Mousavi, and M. A. Reniers. SOS rule formats for zero and unit elements. *Theoretical Computer Science*, 412(28):3045–3071, 2011.
7. L. Aceto, W. Fokkink, and C. Verhoef. Structural operational semantics. In J. A. Bergstra, A. Ponse, and S. A. Smolka, editors, *Handbook of Process Algebra, Chapter 3*, pages 197–292. Elsevier Science, Dordrecht, The Netherlands, 2001.
8. L. Aceto and A. Ingolfsdottir. CPO models for compact GSOS languages. *Information and Computation*, 129(2):107–141, 1996.
9. L. Aceto, A. Ingólfsdóttir, M. R. Mousavi, and M. A. Reniers. Algebraic properties for free! *Bulletin of the European Association for Theoretical Computer Science*, 99:81–104, 2009.
10. J. Baeten, T. Basten, and M. A. Reniers. *Process Algebra: Equational Theories of Communicating Processes*, volume 50 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2009.
11. J. Baeten, J. Bergstra, and J. W. Klop. Syntax and defining equations for an interrupt mechanism in process algebra. *Fundamenta Informaticae*, IX(2):127–168, 1986.
12. J. Baeten and E. P. de Vink. Axiomatizing GSOS with termination. *Journal of Logic and Algebraic Programming*, 60-61:323–351, 2004.
13. J. Bergstra and J. W. Klop. Process algebra for synchronous communication. *Information and Control*, 60(1-3):109–137, 1984.
14. B. Bloom. *Ready Simulation, Bisimulation, and the Semantics of CCS-like Languages*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1989.
15. B. Bloom, S. Istrail, and A. R. Meyer. Bisimulation can't be traced. *Journal of the ACM*, 42(1):232–268, Jan. 1995.
16. S. Cranen, M. R. Mousavi, and M. A. Reniers. A rule format for associativity. In F. van Breugel and M. Chechik, editors, *Proceedings of the 19th International Conference on Concurrency Theory (CONCUR'08)*, volume 5201 of *Lecture Notes in Computer Science*, pages 447–461. Springer-Verlag, 2008.
17. J. Engelfriet and T. Gelsema. Multisets and structural congruence of the pi-calculus with replication. *Theoretical Computer Science*, 211(1–2):311–337, 1999.
18. W. Fokkink. A complete equational axiomatization for prefix iteration. *Information Processing Letters*, 52(6):333–337, 1994.
19. W. Fokkink. Axiomatizations for the perpetual loop in process algebra. In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, *Automata, Languages and Programming, 24th International Colloquium, ICALP'97, Bologna, Italy, 7–11 July 1997, Proceedings*, volume 1256 of *Lecture Notes in Computer Science*, pages 571–581. Springer, 1997.
20. R. J. v. Glabbeek. The linear time - branching time spectrum I. In J. A. Bergstra, A. Ponse, and S. A. Smolka, editors, *Handbook of Process Algebra, Chapter 1*, pages 3–100. Elsevier Science, Dordrecht, The Netherlands, 2001.
21. M. Hennessy. A term model for synchronous processes. *Information and Control*, 51(1):58–75, 1981.

22. C. A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.
23. S. Kleene. Representation of events in nerve nets and finite automata. In C. Shannon and J. McCarthy, editors, *Automata Studies*, pages 3–41. Princeton University Press, 1956.
24. R. Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer-Verlag, 1980.
25. R. Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25:267–310, 1983.
26. R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
27. M. R. Mousavi, M. A. Reniers, and J. F. Groote. A syntactic commutativity format for SOS. *Information Processing Letters*, 93:217–223, Mar. 2005.
28. M. R. Mousavi, M. A. Reniers, and J. F. Groote. SOS formats and meta-theory: 20 years after. *Theoretical Computer Science*, 373:238–272, 2007.
29. D. M. Park. Concurrency and automata on infinite sequences. In P. Duessen, editor, *Proceedings of the 5th GI Conference*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, Berlin, Germany, 2001.
30. G. D. Plotkin. A structural approach to operational semantics. *Journal of Logic and Algebraic Programming*, 60:17–139, 2004.
31. A. W. Roscoe, C. A. R. Hoare, and R. Bird. *The Theory and Practice of Concurrency*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1997.
32. D. Sangiorgi. *Introduction to Bisimulation and Coinduction*. Cambridge University Press, 2011.
33. D. Sangiorgi and D. Walker. *The π-Calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.

A Proof of Theorem 1

Let R be the least reflexive relation over $\mathbb{C}(\Sigma)$ such that

- $f(p)\ R\ f(p) \odot f(p)$ and $f(p) \odot f(p)\ R\ f(p)$, for each $p \in \mathbb{C}(\Sigma)$,
- if $p \leftrightarrows p'$, $p'\ R\ q'$ and $q' \leftrightarrows q$, then $p\ R\ q$, and
- if $g \in \Sigma$ is an n-ary operator and $p_i\ R\ q_i$ for each $i \in \{1, \ldots, n\}$, then $g(p_1, \ldots, p_n)\ R\ g(q_1, \ldots, q_n)$.

In order to prove the theorem, it suffices to show that R is a bisimulation. To this end, note, first of all, that the relation R defined above is symmetric. Assume now that $p\,R\,q$ and $p \xrightarrow{a} p'$ for some p'. Our aim is to prove that there is some q' such that $q \xrightarrow{a} q'$ and $p'\,R\,q'$. This we show by an induction on the definition of R. Below we limit ourselves to detailing the proof for the cases when, for some $p_1 \in \mathbb{C}(\Sigma)$,

- $p = f(p_1)$ and $q = f(p_1) \odot f(p_1)$, and
- $p = f(p_1) \odot f(p_1)$ and $q = f(p_1)$.

Suppose that $f(p_1) \xrightarrow{a} p'$. Then there are a rule

$$\frac{H}{f(x) \xrightarrow{a} t}$$

and a closed substitution σ such that σ satisfies H, $\sigma(x) = p_1$ and $\sigma(t) = p'$. By condition 1 in Definition 7, there is some $\odot$-defining rule

$$\frac{H'}{x_1 \odot x_2 \xrightarrow{a} u},$$

such that

1. $H' \subseteq \{x_1 \xrightarrow{a} y_1,\ x_2 \xrightarrow{a} y_2\}$, with x_1, x_2, y_1, y_2 pairwise distinct, and
2. $t \downarrow_{f,\odot} u[x_1 \mapsto f(x), x_2 \mapsto f(x), y_1 \mapsto t, y_2 \mapsto t]$.

Consider now the closed substitution

$$\sigma' = \sigma[x_1 \mapsto f(p_1), x_2 \mapsto f(p_1), y_1 \mapsto p', y_2 \mapsto p'].$$

Since $f(p_1) \xrightarrow{a} p'$, we have that σ' satisfies H'. Therefore, using the above-mentioned rule for $\odot$, we may conclude that

$$f(p_1) \odot f(p_1) \xrightarrow{a} \sigma'(u).$$

As $t \downarrow_{f,\odot} u[x_1 \mapsto f(x), x_2 \mapsto f(x), y_1 \mapsto t, y_2 \mapsto t]$, we have that there are some t' and u' such that

- $t \leftrightsquigarrow t'$,
- $u[x_1 \mapsto f(x), x_2 \mapsto f(x), y_1 \mapsto t, y_2 \mapsto t] \leftrightsquigarrow u'$, and

– either $t' \equiv u'$, or without loss of generality $t' = C[f(t'')]$ and $u' = C[f(t'') \odot f(t'')]$, for some context $C[\,]$ and term t''.

By Lemma 1 and the definition of R,

$$p' = \sigma(t) \leftrightarrows \sigma(t')\, R\, \sigma(u') \leftrightarrows \sigma(u[x_1 \mapsto f(x), x_2 \mapsto f(x), y_1 \mapsto t, y_2 \mapsto t]).$$

Moreover, it is easy to see

$$\sigma(u[x_1 \mapsto f(x), x_2 \mapsto f(x), y_1 \mapsto t, y_2 \mapsto t]) = \sigma'(u).$$

Therefore, by the definition of R, we conclude that

$$p' = \sigma(t)\, R\, \sigma'(u),$$

which was to be shown.

Assume now that $p = f(p_1) \odot f(p_1) \xrightarrow{a} p'$. Then, by condition 2 in Definition 7, there are a rule

$$\frac{\{x_i \xrightarrow{a} y_i\} \cup \{x_1 \xrightarrow{a_j} y_j \mid j \in J\} \cup \{x_2 \xrightarrow{b_k} z_k \mid k \in K\}}{x_1 \odot x_2 \xrightarrow{a} u},$$

and a closed substitution σ such that σ satisfies the premises of the rule, $\sigma(x_1) = \sigma(x_2) = f(p_1)$ and $\sigma(u) = p'$. By condition 2 in Definition 7, we have that

1. $i \in \{1, 2\}$, and
2. $vars(u) \subseteq \{x_1, x_2, y_i\}$.

Suppose, without loss of generality, that $i = 1$. Then

$$\sigma(x_1) = f(p_1) \xrightarrow{a} \sigma(y_1).$$

Therefore there are a rule

$$\frac{H}{f(x) \xrightarrow{a} t}$$

for f and a closed substitution σ' such that σ' satisfies H, $\sigma'(x) = p_1$ and $\sigma'(t) = \sigma(y_1)$. We claim that

$$\sigma'(t) = \sigma(y_1)\, R\, \sigma(u) = p'.$$

To see this, observe that, by condition 3 in Definition 7,

$$t \downarrow_{f,\odot} u[x_1 \mapsto f(x), x_2 \mapsto f(x), y_1 \mapsto t].$$

Therefore, we have that there are some t' and u' such that

– $t \leftrightsquigarrow t'$,
– $u[x_1 \mapsto f(x), x_2 \mapsto f(x), y_1 \mapsto t] \leftrightsquigarrow u'$, and

– either $t' \equiv u'$, or without loss of generality $t' = C[f(t'')]$ and $u' = C[f(t'') \odot f(t'')]$, for some context $C[\,]$ and term t''.

By Lemma 1 and the definition of R,

$$\sigma'(t) \leftrightarrows \sigma'(t')\ R\ \sigma'(u') \leftrightarrows \sigma'(u[x_1 \mapsto f(x), x_2 \mapsto f(x), y_1 \mapsto t]).$$

Moreover, it is easy to see

$$\sigma'(u[x_1 \mapsto f(x), x_2 \mapsto f(x), y_1 \mapsto t]) = \sigma(u).$$

Therefore, by the definition of R, we conclude that

$$\sigma(y_1) = \sigma'(t)\ R\ \sigma(u) = p',$$

which was to be shown. Since R is symmetric, we are done.

B Proof of Theorem 2

Let R be the least reflexive relation over $\mathbb{C}(\Sigma)$ such that

– $f(p)\ R\ f(f(p))$ and $f(f(p))\ R\ f(p)$, for each $p \in \mathbb{C}(\Sigma)$,
– if $p \leftrightarrows p'$, $p'\ R\ q'$ and $q' \leftrightarrows q$, then $p\ R\ q$, and
– if $g \in \Sigma$ is an n-ary operator and $p_i\ R\ q_i$ for each $i \in \{1, \ldots, n\}$, then $g(p_1, \ldots, p_n)\ R\ g(q_1, \ldots, q_n)$.

In order to prove the theorem, it suffices to show that R is a bisimulation. To this end, note, first of all, that the relation R defined above is symmetric. Assume now that $p\,R\,q$ and $p \xrightarrow{a} p'$ for some p'. Our aim is to prove that there is some q' such that $q \xrightarrow{a} q'$ and $p'\,R\,q'$. This we show by an induction on the definition of R. Below we limit ourselves to detailing the proof for the cases when, for some $p_1 \in \mathbb{C}(\Sigma)$,

– $p = f(p_1)$ and $q = f(f(p_1))$, and
– $p = f(f(p_1))$ and $q = f(p_1)$.

Suppose that $f(p_1) \xrightarrow{a} p'$. Then, using condition 1 in Definition 10, there are a rule

$$r = \frac{H \cup \{x \xrightarrow{b_j} \mid j \in J\}}{f(x) \xrightarrow{a} t},$$

with $H \subseteq \{x \xrightarrow{a} x'\}$, and a closed substitution σ such that σ satisfies H, $\sigma(x) = p_1$ and $\sigma(t) = p'$. By conditions 3a and 3b in Definition 10,

$$t \Downarrow_f t[x \mapsto f(x), x' \mapsto t].$$

(Note that, if $H = \emptyset$, then x is the only variable that might possibly occur in t, and therefore $t[x \mapsto f(x), x' \mapsto t] = t[x \mapsto f(x)]$.)

Our goal is to use the rule r to infer a matching transition from the term $f(f(p_1))$. To this end, consider the substitution $\sigma[x \mapsto f(p_1), x' \mapsto p']$. Since $f(p_1) \xrightarrow{a} p'$, that substitution satisfies H. We claim that $\sigma[x \mapsto f(p_1), x' \mapsto p']$ also satisfies the negative premises of r. Indeed, let $j \in J$. Since σ satisfies $x \overset{b_j}{\not\to}$, we have that $\sigma(x) = p_1 \overset{b_j}{\not\to}$. By condition 2 in Definition 10, each b_j-emitting and f-defining rule has a positive premise of the form $x \xrightarrow{b_j} x'$. As $p_1 \overset{b_j}{\not\to}$, it follows that $f(p_1) \overset{b_j}{\not\to}$, as claimed. Therefore, the above rule yields

$$f(f(p_1)) \xrightarrow{a} \sigma[x \mapsto f(p_1), x' \mapsto p'](t).$$

As $t \Downarrow_f t[x \mapsto f(x), x' \mapsto t]$, we have that there are some t' and u' such that

- $t \leftrightarrow t'$,
- $t[x \mapsto f(x), x' \mapsto t] \leftrightarrow u'$, and
- either $t' \equiv u'$, or $t' = C[f(t'')]$ and $u' = C[f(f(t''))]$, for some context $C[\ ]$ and term t''.

By Lemma 2 and the definition of R,

$$p' = \sigma(t) \underline{\leftrightarrow} \sigma(t')\ R\ \sigma(u') \underline{\leftrightarrow} \sigma(t[x \mapsto f(x), x' \mapsto t]).$$

It is easy to see that

$$\sigma[x \mapsto f(p_1), x' \mapsto p'](t) = \sigma(t[x \mapsto f(x), x' \mapsto t]).$$

Therefore, again by the definition of R, we conclude that

$$p' = \sigma(t)\ R\ \sigma[x \mapsto f(p_1), x' \mapsto p'](t),$$

which was to be shown.

Assume now that $p = f(f(p_1)) \xrightarrow{a} p'$. Then, by condition 1 in Definition 10, there are a rule

$$\frac{H \cup \{x \overset{b_j}{\not\to} \mid j \in J\}}{f(x) \xrightarrow{a} t}$$

with $H \subseteq \{x \xrightarrow{a} x'\}$, and a closed substitution σ such that σ satisfies H, $\sigma(x) = f(p_1)$ and $\sigma(t) = p'$.

If H is empty, then condition 1b in Definition 10 ensures that J is also empty. Thus the above rule yields the transition $f(p_1) \xrightarrow{a} \sigma[x \mapsto p_1](t)$. Moreover, by condition 3a in Definition 10,

$$t \Downarrow_f t[x \mapsto f(x)].$$

Therefore, reasoning as above,

$$\sigma[x \mapsto p_1](t)\ R\ \sigma[x \mapsto p_1](t[x \mapsto f(x)]) = \sigma(t) = p',$$

and we are done.

If $H = \{x \xrightarrow{a} x'\}$ then $f(p_1) \xrightarrow{a} \sigma(x')$. Therefore, there are some a-emitting rule for f

$$\frac{H'}{f(x) \xrightarrow{a} t'},$$

and some closed substitution σ' such that σ' satisfies H', $\sigma'(x) = p_1$ and $\sigma'(t') = \sigma(x')$. Moreover, by condition 3b in Definition 10, we have that

$$t' \Downarrow_f t[x \mapsto f(x), x' \mapsto t'].$$

Now, reasoning as above,

$$\sigma'(t') \; R \; \sigma'(t[x \mapsto f(x), x' \mapsto t']) = \sigma(t) = p'.$$

Since R is symmetric, we are done.