The Journal of Logic and Algebraic Programming 83 (2014) 64–80

Contents lists available at ScienceDirect

The Journal of Logic and Algebraic Programming

www.elsevier.com/locate/jlap

SOS rule formats for idempotent terms and idempotent unary operators [☆]

Luca Aceto [a,b,*], Anna Ingolfsdottir [a], Eugen-Ioan Goriac [a]

[a] ICE-TCS, School of Computer Science, Reykjavik University, Menntavegur 1, IS 101 Reykjavik, Iceland
[b] SysMA, IMT Lucca Institute for Advanced Studies, Lucca 55100, Italy

ARTICLE INFO

Article history:
Received 21 December 2012
Accepted 19 July 2013
Available online 9 August 2013

Keywords:
Structural operational semantics
Rule formats
Concurrency
Process algebra
Bisimilarity
Equational logic

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) [35] 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 [12], CCS [31] and CSP [27], operators such as non-deterministic and parallel composition are often meant to be commutative and associative with respect to bisimilarity [34]. 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-called *SOS rule formats* [11]. 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 [3,5,6,11,21,32] for examples of rule formats for algebraic properties in the literature on SOS.) The definition of SOS rule formats

☆ 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.
* Corresponding author at: ICE-TCS, School of Computer Science, Reykjavik University, Menntavegur 1, IS 101 Reykjavik, Iceland.
E-mail addresses: luca@ru.is (L. Aceto), annai@ru.is (A. Ingolfsdottir), eugen.goriac@me.com (E.-I. Goriac).

1567-8326/$ – see front matter
http://dx.doi.org/10.1016/j.jlap.2013.07.003

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 [28], the delay operator from SCCS [26,30], the replication operator from the π-calculus [38] and the priority operator from [14].

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 [21,32].

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.

This paper is an extended version of [9]. Apart from offering the proofs of results that were announced without proof in that reference and a variety of examples of applications of our rule formats, this version of the paper also presents a generalization of the rule format for idempotent unary operations given in [9].

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 [7,33] for further details.

Transition system specifications in GSOS format.

Definition 2.1 *(Signature, terms and substitutions).* Let V be an infinite set of variables with typical members $x, x', x_i, y, y', y_i, \ldots$. A *signature* Σ is a set of function symbols, each with a fixed arity. We call these symbols *operators* and usually denote 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 $(1 \leqslant i \leqslant n)$ 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 [18,19]. 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, [2,1,4,7,10,15].