Theoretical Computer Science 412 (2011) 5043–5069

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Bialgebras for structural operational semantics: An introduction

Bartek Klin *

University of Warsaw, Faculty of Mathematics, Informatics and Mechanics, Banacha 2, 02-097 Warsaw, Poland
University of Cambridge, Computer Laboratory, William Gates Bldg., 15 JJ Thomson Avenue, Cambridge CB3 0FD, UK

ARTICLE INFO

Keywords:
Structural operational semantics
Coalgebra
Bialgebra

ABSTRACT

Bialgebras and distributive laws are an abstract, categorical framework to study various flavors of structural operational semantics. This paper aims to introduce the reader to the basics of bialgebras for operational semantics, and to sketch the state of the art in this research area.

1. Introduction

Structural Operational Semantics (SOS) is one of the most popular frameworks for the formal description of programming languages and process calculi. It has become the formalism of choice for a clear and concise presentation of many ideas and formalisms (see [3] for examples), and it is a viable option for the description of fully grown programming languages [38].

In the simplest and most well-studied form of SOS [1], the semantics of processes is described by means of nondeterministic labeled transition systems (LTSs), induced from inference rules following their syntactic structure. For example, the rule

$$\frac{\mathtt{x} \xrightarrow{a} \mathtt{x}' \quad \mathtt{y} \xrightarrow{\bar{a}} \mathtt{y}'}{\mathtt{x}||\mathtt{y} \xrightarrow{\tau} \mathtt{x}'||\mathtt{y}'}$$

used in the definition of the well-known process calculus CCS [37] means that if a process $\mathtt{x}$ can make a transition labeled with an atomic action a, and if $\mathtt{y}$ can make a transition labeled with a corresponding action $\bar{a}$, then the composite process $\mathtt{x}||\mathtt{y}$ can combine the two transitions into one labeled with the label τ.

Already from the original paper on SOS ([40], reprinted as [41]) it was clear that simple LTSs are only one kind of dynamic systems worth considering, and that to model different computational paradigms one needs to study transition systems with state, environments, etc. Later, also probabilistic, stochastic, timed and other kinds of systems were defined in various flavors of SOS. Although each of these flavors is a little different, they all share a common underlying theme: *the interplay between the structure (syntax) and the dynamics (behavior) of systems*.

Although the latter expression might sound a little vague, in the late 1990s it has found an elegant and general formalization with the use of basic category theory. The main conceptual step was made with the development of universal coalgebra, a general categorical approach that described several different kinds of transition systems in a uniform way. Since syntax has traditionally been modeled in the dual framework of universal algebra, with the benefit of hindsight it seems natural that the two theories should somehow be combined to explain the various flavors of SOS. This indeed happened in the seminal paper [54] where, building upon earlier initial ideas of [44], SOS specifications were formalized as *distributive laws* of syntax over behavior, both modeled as endofunctors on the same category, and models of specifications were defined as *bialgebras* for these laws.

* Corresponding author at: University of Warsaw, Faculty of Mathematics, Informatics and Mechanics, Banacha 2, 02-097 Warsaw, Poland. Tel.: +48 225544484.
E-mail addresses: klin@mimuw.edu.pl, bklin@inf.ed.ac.uk.

0304-3975/$ – see front matter © 2011 Published by Elsevier B.V.
doi:10.1016/j.tcs.2011.03.023

More specifically, it was shown in [54] that SOS specifications of LTSs that are in the so-called GSOS format [4], correspond to a certain type of distributive laws of functors that model syntax over a functor that models the behavior of LTSs. The main property of GSOS specifications, that bisimilarity on LTSs induced from them is always a congruence, was formulated and proved at the abstract level of distributive laws. Since the bialgebraic framework is parametrized by the notions of syntax and behavior, this opened a possibility to understand well-behaved SOS formalisms for other kinds of systems in a uniform manner. This has indeed happened since, and several novel, concrete specification formats such as probabilistic or stochastic GSOS have been derived by analysis of the corresponding abstract distributive laws. Although much remains to be done, the bialgebraic framework has a good claim to be the main abstract approach to SOS. Furthermore, bialgebras have been used for an abstract understanding of ideas seemingly unconnected to SOS, such as stream equations or regular languages.

The purpose of this paper is to provide a gentle introduction to the basic framework of distributive laws for SOS, and a survey of the current state of the art in the area. We shall define a few types of distributive laws, from simple distributive laws of endofunctors over endofunctors, to more complex GSOS and coGSOS laws, to the general case of distributing monads over comonads. For concrete kinds of transition systems, most of these types correspond to progressively more permissive formats of well-behaved SOS specifications; to simplify the presentation, we shall concentrate on the very simple stream systems, a kind of automata with deterministic output and no input at all.

After reading this expository paper, the reader should be prepared (and, hopefully, motivated) to study the field of bialgebra in more depth. For further reading, the author recommends to begin with [44,54] and perhaps [52] to see how the ideas originally developed, Chapter 3 of [2] for a thorough but gentle exposition, and [34] for a more abstract categorical perspective.

The structure of the paper is as follows: after Section 2 of preliminaries about algebras and coalgebras, further development is motivated in Section 3 by a few concrete examples related to stream systems. Distributive laws are initially motivated not by a study of inference rules but by well-behaved definitions of operations on infinite streams, but stream SOS rules soon naturally appear as an additional benefit. In Section 4, the examples of the preceding section are cast in a general setting of simple distributive laws of endofunctors over endofunctors, and a basic theory of such laws is developed, up to an abstract formulation of the congruence property of observational equivalence. Section 5 focuses on the world of stream systems again, and provides a concrete, rule-based presentation of simple distributive laws for such systems.

Since simple laws of Section 4 are not expressive enough to cover all examples of interest, in Section 6 more complex types of laws are introduced, motivated by further examples of operations on stream systems. Section 7 presents concrete rule formats obtained so far by analysis of distributive laws for various kinds of systems, and Section 8 lists other relevant work related to bialgebras and their applications to SOS.

2. Algebras and coalgebras

The reader is assumed to be familiar with basic notions of category theory such as categories, functors and natural transformations. A standard reference for these is [35].

2.1. Syntax via algebras

An *algebraic signature* Σ is a collection of operation symbols $\{\mathtt{f}_i \mid i \in I\}$ where each $\mathtt{f}_i$ has an arity $n_i \in \mathbb{N}$. A Σ*-algebra* with carrier set X is a map $\coprod_{i\in I} X^{n_i} \to X$, and therefore a signature Σ shall be identified with the functor $\Sigma X = \coprod_{i\in I} X^{n_i}$ on the category **Set** of sets and functions. In general, given any endofunctor Σ on a category $\mathcal{C}$:

Definition 1. A Σ-algebra is an object X in $\mathcal{C}$ together with a map $g : \Sigma X \to X$. A Σ-algebra morphism from $g : \Sigma X \to X$ to $e : \Sigma Y \to Y$ is a map $f : X \to Y$ such that $e \circ \Sigma f = f \circ g$. Σ-algebras and their morphisms form a category Σ**-alg**.

There is an obvious forgetful functor $U^{\Sigma} : \Sigma\textbf{-alg} \to \mathcal{C}$.

If Σ has an initial algebra $a : \Sigma A \to A$, then for any algebra g, the unique algebra map f from a to g is called the *inductive extension* of g. The principle of defining maps from the carrier of an initial Σ-algebra by providing another Σ-algebra is called *induction*.

For Σ on **Set** arising from a signature, the set Σ^*0 of closed Σ-terms carries an initial Σ-algebra structure, and the inductive extension of an algebra $g : \Sigma X \to X$ is the unique interpretation of closed Σ-terms in g defined by structural induction.

More generally, the set of Σ-terms over a set of variables X is denoted by Σ^*X. It is easy to see that Σ^* extends to a functor on **Set**.

2.2. Behavior via coalgebras

A detailed description of the coalgebraic approach to system dynamics is beyond the scope of this paper; the interested reader can consult [45] for a thorough introduction. This section only briefly recalls basic notions and results that will be useful in the following.

Fix any endofunctor B on a category $\mathcal{C}$, called a *behavior functor* in this context.