Theoretical Computer Science 604 (2015) 102–126

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Killing epsilons with a dagger: A coalgebraic study of systems with algebraic label structure

Filippo Bonchi [a,*], Stefan Milius [b,*,1], Alexandra Silva [c,d,e,*], Fabio Zanasi [a,*]

[a] ENS Lyon, U. de Lyon, CNRS, INRIA, UCBL, France
[b] Lehrstuhl für Theoretische Informatik, FAU Erlangen-Nürnberg, Germany
[c] Institute for Computing and Information Sciences, Radboud University Nijmegen, The Netherlands
[d] Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
[e] HASLab/INESC TEC, Universidade do Minho, Braga, Portugal

ARTICLE INFO

Article history:
Received 10 October 2014
Received in revised form 7 March 2015
Accepted 14 March 2015
Available online 21 March 2015

Keywords:
Coalgebras on Kleisli categories
Parametrized fixpoint operator
Trace semantics
Epsilon transitions
Mazurkiewicz traces
Non-deterministic transducers

ABSTRACT

We propose an abstract framework for modelling state-based systems with internal behaviour as e.g. given by silent or ϵ-transitions. Our approach employs monads with a parametrized fixpoint operator $\dagger$ to give a semantics to those systems and implement a sound procedure of abstraction of the internal transitions, whose labels are seen as the unit of a free monoid. More broadly, our approach extends the standard coalgebraic framework for state-based systems by taking into account the algebraic structure of the labels of their transitions. This allows to consider a wide range of other examples, including Mazurkiewicz traces for concurrent systems and non-deterministic transducers.

1. Introduction

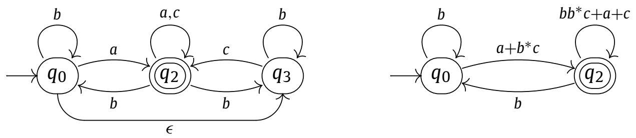
The theory of coalgebras provides an elegant mathematical framework to express the semantics of computing devices: the operational semantics, which is usually given as a state machine, is modelled as a coalgebra for a functor; the denotational semantics as the unique map into the final coalgebra of that functor. While the denotational semantics is often *compositional* (as, for instance, ensured by the bialgebraic approach of [34]), it is sometimes not *fully-abstract*, i.e., it discriminates systems that are equal from the point of view of an external observer. This is due to the presence of internal transitions (also called ϵ-transitions) that are not observable but that are not abstracted away by the usual coalgebraic semantics using the unique homomorphism into the final coalgebra.

In this paper, we focus on the problem of giving trace semantics to systems with internal transitions. Our approach stems from an elementary observation (pointed out in previous work, e.g. [39]): the labels of transitions form a monoid and the internal transitions are those labelled by the unit of the monoid. Thus, there is an *algebraic structure* on the labels that needs to be taken into account when modelling the denotational semantics of those systems. To illustrate this point, consider the following two non-deterministic automata (NDA).

* Corresponding authors.
E-mail addresses: filippo.bonchi@ens-lyon.fr (F. Bonchi), mail@stefan-milius.eu (S. Milius), alexandra@cs.ru.nl (A. Silva), fabio.zanasi@ens-lyon.fr (F. Zanasi).

[1] Stefan Milius acknowledges support by the Deutsche Forschungsgemeinschaft (DFG) under project MI 717/5-1.

http://dx.doi.org/10.1016/j.tcs.2015.03.024
0304-3975/



The one on the left (that we call $\mathbb{A}$) is an NDA with ϵ-transitions: its transitions are labelled either by the symbols of the alphabet $A = \{a, b, c\}$ or by the empty word $\epsilon \in A^*$. The one on the right (that we call $\mathbb{B}$) has transitions labelled by languages in $\mathcal{P}(A^*)$, here represented as regular expressions. The monoid structure on the labels is explicit on $\mathbb{B}$, while it is less evident in $\mathbb{A}$ since the set of labels $A \cup \{\epsilon\}$ does not form a monoid. However, this set can be trivially embedded into $\mathcal{P}(A^*)$ by looking at each symbols as the corresponding singleton language. For this reason each automaton with ϵ-transitions, like $\mathbb{A}$, can be regarded as an automaton with transitions labelled by languages, like $\mathbb{B}$. Furthermore, we can define the semantics of NDA with ϵ-transitions by defining the semantics of NDA with transitions labelled by languages: a word w is accepted by a state q if there is a path $q \xrightarrow{L_1} \cdots \xrightarrow{L_n} p$ where p is a final state, and there exist a decomposition $w = w_1 \cdots w_n$ such that $w_i \in L_i$. Observe that, with this definition, $\mathbb{A}$ and $\mathbb{B}$ accept the same language: all words over A that end with a or c. In fact, $\mathbb{B}$ was obtained from $\mathbb{A}$ in a well-known process to compute the regular expression denoting the language accepted by a given automaton [25].

We propose to define the semantics of systems with internal transitions following the same idea as in the above example. Given some transition type (i.e. an endofunctor) F, one first defines an embedding of F-systems with internal transitions into F^*-system, where F^* has been derived from F by making explicit the algebraic structure on the labels. Next one models the semantics of an F-system as the one of the corresponding F^*-system e. Naively, one could think of defining the semantics of e as the unique map $!_e$ into the final coalgebra for F^*. However, this approach turns out to be too fine grained, essentially because it ignores the underlying algebraic structure on the labels of e. The same problem can be observed in the example above: $\mathbb{B}$ and the representation of $\mathbb{A}$ as an automaton with languages as labels have different final semantics—they accept the same language only modulo the equations of monoids.

Thus we need to extend the standard coalgebraic framework by taking into account the algebraic structure on labels. To this end, we develop our theory for systems whose transition type F^* has a *canonical fixpoint*, i.e. its initial algebra and final coalgebra coincide. This is the case for many relevant examples, as observed in [23]. Our *canonical fixpoint semantics* will be given as the composite $¡ \circ !_e$, where $!_e$ is a coalgebra morphism given by finality and $¡$ is an algebra morphism given by initiality. The target of $¡$ will be an algebra for F^* encoding the equational theory associated with the labels of F^*-systems. Intuitively, $¡$ being an *algebra* morphism, will take the quotient of the semantics given by $!_e$ modulo those equations. Therefore the extension provided by $¡$ is the technical feature allowing us to take into account the algebraic structure on labels.

It were Simpson and Plotkin [38, Section 6] who realized that the above composites $¡ \circ !_e$ yield a *parametrized fixpoint operator* $e \mapsto e^\dagger$. This operator can be understood as assigning to systems of mutually recursive equations a certain *solution*, and the properties of $e \mapsto e^\dagger$ will be crucial for our canonical fixpoint semantics.

Within the same perspective we also consider a different fixpoint operator $e \mapsto e^\ddagger$ turning any system e with internal transitions into one $e^\ddagger$ where those have been abstracted away. By comparing the operators $e \mapsto e^\dagger$ and $e \mapsto e^\ddagger$, we are then able to show that such a procedure (also called ϵ*-elimination*) is sound with respect to the canonical fixpoint semantics.

We further explore the flexibility of our framework by modelling the case in which the algebraic structure of the labels is quotiented under some equations, resulting in a coarser equivalence than the one given by the canonical fixpoint semantics. As a relevant example of this phenomenon, we give the first coalgebraic account of Mazurkiewicz traces.

As our last application, we model non-deterministic transducers (with and without ϵ-transitions). This is a pleasing case study: on the one hand, it was known to be a hard problem to solve in the coalgebraic framework [21]; on the other hand, it follows as a simple application of our approach, thereby illustrating its power. In fact, as we observe, the only difference between transducers and non-deterministic automata is a change in the monad capturing the branching structure. In the NDA case, this is just non-determinism ($\mathcal{P}$, the powerset monad) whereas in the transducer case the monad needs to also capture the fact that transitions can output words ($\mathcal{P}(B^* \times \mathrm{Id})$, composition of the powerset and monoid action monads).

This paper is an extended and improved version of our CMCS'14 paper [10]. We have included all the proofs and the new example of non-deterministic transducers. We were also able to weaken the assumptions of our framework. In the conference version, Assumption 5.1 required the base category **C** to be **Cppo**-enriched and the monad T to be locally continuous. These assumptions ensure (a) initial algebra-final coalgebra coincidence for the functors $T(\mathrm{Id} + Y)$ and (b) that the canonical fixpoint operator $e \mapsto e^\dagger$ satisfies the so-called *double dagger law*. The latter is instrumental in our framework to correctly capture the semantics of systems with internal behaviour. Fortunately, it follows from the results of Simpson and Plotkin [38] that (a) and (b) hold whenever T has *enough canonical fixpoints*, in particular, no **Cppo**-enrichment and local continuity of T is needed.

Synopsis After recalling the necessary background in Section 2, we discuss our motivating examples—automata with ϵ-transitions and automata on words—in Section 3. Sections 4 and 5 are devoted to present the canonical fixpoint semantics and the sound procedure of ϵ-elimination. This framework is then instantiated to the examples of Section 3. In Section 6 we show how a quotient of the algebra on labels induces a coarser canonical fixpoint semantics. We propose