Axiomatizing an Algebra of Step Reactions for Synchronous Languages

Gerald Lüttgen[1] and Michael Mendler[2]

[1] Department of Computer Science, Sheffield University, 211 Portobello Street, Sheffield S1 4DP, U.K., g.luettgen@dcs.shef.ac.uk
[2] Fakultät für Wirtschaftsinformatik und Angewandte Informatik, Universität Bamberg, D–96045 Bamberg, michael.mendler@wiai.uni-bamberg.de

Abstract. This paper introduces a novel algebra for reasoning about step reactions in synchronous languages, such as macro steps in Harel, Pnueli and Shalev's Statecharts and instantaneous reactions in Berry's Esterel. The algebra describes step reactions in terms of configurations which can both be read in a standard operational as well as in a model–theoretic fashion. The latter arises by viewing configurations as propositional formulas, interpreted intuitionistically over finite linear Kripke structures. Previous work by the authors showed the adequacy of this approach by establishing compositionality and full–abstraction results for Statecharts and Esterel. The present paper generalizes this work in an algebraic setting and, as its main result, provides a sound and complete equational axiomatization of step reactions. This yields, for the first time in the literature, a complete axiomatization of Statecharts macro steps, which can also be applied, modulo encoding, to Esterel reactions.

1 Introduction

Synchronous languages provide a popular framework for designing and programming *event–based reactive systems*. Prominent examples of such languages include Harel's *Statecharts* [5], which is a graphical language that extends finite–state machines by concepts of state hierarchy, concurrency, and event priority, and Berry's *Esterel* [2, 3], which is a textual language having similar features to Statecharts. Today, both languages are supported by commercial tools, including *Statemate* [7] and *Esterel Studio* [4], which mainly focus on generating running code. The development of semantic–based verification tools is still in its infancy, partly due to the lack of sufficiently simple compositional semantics.

The semantics of Statecharts, as conceived by Pnueli and Shalev [20], and of Esterel are based on the idea of *cycle–based reaction*, where first the input events, as defined by a system's environment, are sampled at the beginning of each cycle, then the system's reaction in form of the emission of further events is determined, and finally the generated events are output to the environment. Statecharts and Esterel differ in the details of what exactly constitutes a cycle, which is also called a *macro step* in Statecharts and an *instantaneous reaction* in Esterel. Moreover, Esterel refers to events as signals. Both languages have in common that they obey the semantic principles of *synchrony* and *causality*. The

G. Luettgen, M. Mendler: Axiomatizing an Algebra of Step Reactions for Synchronous Languages.
In L. Brim, P. Jancar, M. Ketinsky, A. Kucera (eds.), Int'l Conference on Concurrency Theory (CONCUR'02), Springer LNCS 2421, 2002, pp. 386-401.

synchrony requirement reflects the mechanism behind cycle–based reaction and is mathematically modeled via the *synchrony hypothesis*. This hypothesis ensures that reactions and the propagations of events are instantaneous, which is an idealized system behavior, practically justified by the observation that reactive systems usually perform much faster than their environments. Causality refers to the requirement that the reason for an event to be generated in a system reaction must be traced back to the input events provided by the environment. Esterel differs from Statecharts in that it further adopts the principles of *reactivity* and *determinism*. Reactivity implies that, in each cycle, a system response in the form of generated events can be constructed, for any inputs an environment may provide. Determinism requires for this response to be unique.

This brief discussion highlights the variety of possible choices when defining a semantics for step reactions, with different choices implying subtly different semantics. Recent research by the authors, aiming at a unifying semantic framework for synchronous languages, has concentrated on employing ideas from *intuitionistic logic* for describing step reactions [12–15]. Intuitionistic logic, in contrast to classical logic, is constructive and thus truly reflects the operational character of step reactions in the light of causality: it rejects the classical principle of the excluded middle, i.e., events are either always present or always absent *throughout* a reaction, which cannot be maintained for a compositional semantics that allows the system environment to inject events *during* a step reaction. Indeed, our intuitionistic setting has lead to compositional and fully–abstract characterizations of Statecharts macro steps and Esterel reactions [12, 14].

This paper introduces a simple yet expressive algebra for describing and reasoning about step reactions in terms of so–called *configurations* and presents an equational axiomatization for it. In particular, this gives for the first time in the literature a sound and complete axiomatization for Statecharts macro steps, which can also be applied, modulo encoding, to Esterel reactions. The step algebra's semantics is inspired by the authors' previous work and reads configurations as propositional formulas, interpreted intuitionistically over finite linear Kripke structures, to which we refer as *sequence structures* (cf. Sec. 2). Our axiomatization is then built on top of this algebra (cf. Sec. 3), and its proof of completeness combines techniques used in process algebras [1] and logics (cf. Sec. 4); it employs a process–algebraic notion of *normal form* that in turn is defined by model–theoretic means. Our axioms have an appealing operational intuition that shades light on the semantics of step reactions. They also provide groundwork for an axiomatic comparison of popular synchronous languages.

2 Step Algebra

This section introduces our step algebra for reasoning about those step reactions that may be specified within *event–based synchronous languages*. Usually, synchronous languages, such as *Statecharts* [5, 20] or *Esterel* [2, 3] (with its graphical front–end *SyncCharts*), enrich the notation of finite state machines by mechanisms for expressing hierarchy, concurrency, and priority. This allows one to

G. Luettgen, M. Mendler: Axiomatizing an Algebra of Step Reactions for Synchronous Languages.
In L. Brim, P. Jancar, M. Ketinsky, A. Kucera (eds.), Int'l Conference on Concurrency Theory (CONCUR'02),
Springer LNCS 2421, 2002, pp. 386-401.