Semantics of Reactive Systems in Abstract Time *

C. Huizing                R. Gerth
Eindhoven University of Technology †

Abstract. We explain that real-time reactive systems pose specific problems in defining languages to specify and program them. Three criteria are formulated, responsiveness, modularity, and causality, that are important to have for a high-level specification language for these systems. We prove that these properties can not be combined in one semantics. Since these properties are mandatory for a structured development of real-time reactive systems, we introduce a two-levelled semantics in which the three properties hold on different levels of the semantics: global events are treated more abstractly with respect to time than local events.

Keywords: Real-time, semantics, specification, Statecharts, reactive systems.

Contents

1 Introduction
2 Framework
   2.1 The language
3 Criteria
   3.1 Responsiveness
   3.2 Modularity
   3.3 Causality
4 Semantics
   4.1 Micro-semantics
   4.2 Further definitions
   4.3 Structure of the step relation
   4.4 Semantics A
   4.5 Semantics B
   4.6 Semantics C
   4.7 Semantics D
   4.8 Semantics E
5 Hybrid semantics

*This research is partially supported by ESPRIT projects 937 (DESCARTES) and 3096.
†Department of Mathematics and Computing Science, Eindhoven University of Technology, P.O.Box 513, 5600 MB Eindhoven, The Netherlands. Electronic mail: keesh@win.tue.nl
A Relation between Esterel and the semantic framework

A.1 Short description of ESTEREL
A.2 Semantics of Esterel
A.3 Relationship
A.4 Conclusion

1 Introduction

There is a fundamental dichotomy in the analysis of computing systems. This dichotomy crosses all borderlines between sequential and parallel systems, central and distributed systems, and between functional and imperative systems. This is the dichotomy between transformational and reactive systems [HP85]. Transformational systems are well described by a relation between input and output value. They read some input value, then produce, perhaps non-deterministically, an output value and terminate. A reactive system, however, maintains a continuous interaction with its environment. Typically, the environment reacts upon the output of the system and in many cases the system is not expected to terminate.

Reactive systems can be found anywhere: they include digital watches, interactive software systems, integrated circuits, real-time embedded systems. Design, programming and verification of reactive systems is an important challenge, since existing techniques for transformational systems are not satisfactory for this purpose [HP85].

Recently, several formalism for the development of reactive systems have been proposed. We mention Esterel [BG88, BC85], Lustre [BCH85b], and Statecharts [Har87]. In the development of these formalisms, serious problems have been encountered. Apparently, it is not so simple to design a high-level language for reactive systems. The central problem is that all these languages try to combine the following three properties, or criteria, in one formalism. These properties are for the first time formally defined in this paper.

The first property is responsiveness, meaning that a system's output comes simultaneously with the input that causes it. This requires an abstract notion of time, since there is always some physical time needed to compute a reaction, ultimately. This property is important for high-level specification where one does not want to bother -yet- with implementation details on the one hand, but on the other hand does want to specify in an accurate, non-fuzzy way. Furthermore, it allows for step-wise refinement, without having to redo the timing over and over again.

The second property, modularity, means that all parts of the system should be treated symmetrically. The interface between the environment and the system should be the same as the interface between the parts of the system itself. Furthermore, every part of the system should have the same view of the events occurring in the total system at any moment. Consequently, in all the formalisms mentioned above, the communication mechanism between the subsystems is the immediate, asynchronous broadcast\(^1\).

The third property, causality, means that for any event generated at a particular moment there must be a causal chain of events leading to the action that generated this

\(^1\)If the travelling time of signals is too high, e.g. widely distributed systems, one has to introduce an explicit delay between the moment that an event can be generated and the moment it will actually be sensed by the other components. This can be done in the current framework.