ACADEMIC PRESS

Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Information and Computation 181 (2003) 1–31

Information and Computation

www.elsevier.com/locate/ic

Well-abstracted transition systems: application to FIFO automata☆

Alain Finkel,[a,*,1] S. Purushothaman Iyer,[b,2] and Grégoire Sutre[a,3]

[a] *LSV, ENS Cachan and CNRS UMR 8643, 61, Avenue du Président Wilson, Cachan Cedex 94235, France*
[b] *Department of Computer Science, NC State University, Raleigh, NC 27615, USA*

Received 6 June 2000; revised 13 April 2001

Abstract

Formal methods based on symbolic representations have been found to be very effective. In the case of infinite state systems, there has been a great deal of interest in *accelerations* – a technique for characterizing the result of iterating an execution sequence an arbitrary number of times, in a sound, but not necessarily complete, way. We propose the use of abstractions as a general framework to design accelerations. We investigate SemiLinear Regular Expressions (SLREs) as symbolic representations for FIFO automata. In particular, we show that: (a) SLREs are *easy to manipulate*, (b) SLREs form the *core* of known FIFO symbolic representations, and (c) SLREs are *sufficient* to represent the effect of arbitrary iterations of a loop for FIFO automata with one channel.

Keywords: Infinite state systems; Abstraction; Symbolic representation; Acceleration; Protocols; FIFO automata; Regular expressions; Flatness

1. Introduction

Formal methods are now routinely applied in design and implementation of finite state systems, such as those that occur in VLSI circuits. It has also been applied fairly regularly in the design and

☆ Extended version of *Well-Abstracted Transition Systems* published in CONCUR'2000 proceedings.

* Corresponding author.
E-mail addresses: finkel@lsv.ens-cachan.fr (A. Finkel), purush@csc.ncsu.edu (S. Purushothaman Iyer), sutre@lsv.ens-cachan.fr (G. Sutre).

[1] Supported in part by FORMA, a project funded by DGA, CNRS, and MENRT.
[2] Supported in part by ARO under Grant DAAG55-98-1-03093.
[3] Partially supported by the NSF Theory Grant CCR-9988172 and the NSF ITR Grant CCR-0085949.

0890-5401/02/$ - see front matter
PII: S0890-5401(02)00027-5

implementation of network protocols. Based on the success of formal methods in reasoning about finite state systems [9] there has been a great deal of interest in reasoning about infinite state systems. Given that programs, as well as network protocols, are infinite state in nature there is a need for automatic techniques to extend the reach of formal methods to a much larger class.

Infinite state systems could, in general, be Turing-powerful. Consequently, in reasoning about any non-trivial property of such systems we will have to contend with incompleteness. At least two approaches have been considered in the literature: (a) semi-computation of the set of reachable states [1,4,5,26], and (b) computation of a superset of reachability set [14,32]. A requirement common to both approaches is that an infinite set of reachable states (from some given initial state) be finitely described. Clearly, the finite description should be such that it admits questions of membership and emptiness to be answered effectively. However, given that the reachability set is explored in an iterative fashion, an even more important question is "how does one infer the existence of an infinite set of states in the reachability set? And how does one calculate it?" Techniques called *accelerations* or *meta*-transitions have been discussed in the literature [1,4,5,10,12,20]. We focus in this paper on symbolic representations for the computation of the reachability set of FIFO automata – a finite control with multiple unbounded FIFO channels. To the best of our knowledge, Pachl uses for the first time regular expressions to represent infinite sets of channel contents [31]. In [17], linear regular expressions have been defined and used. Boigelot et al. chose a deterministic finite automata based representation, namely *Queue-content Decision Diagrams* [4] and afterwards Bouajjani et al. added Pressburger formulas, namely *Constrained QDDs* [5]. Simple regular expressions have been introduced for lossy FIFO automata [1].

We propose to address the issue "what are symbolic representations?" In this paper, we show how symbolic representations and accelerations can be couched in terms of abstract interpretation [14], a powerful semantics-based technique for explaining data flow analysis. We present a generic algorithm which, given an abstraction of a labelled transition system and an acceleration, computes a symbolic tree. We illustrate the usefulness of this approach by exploring *Linear* and *SemiLinear Regular Expressions* (LREs and SLREs) as symbolic representations for FIFO automata. In particular, we show the following about SLREs:

- SLREs are *easy to manipulate*: indeed, SLREs are exactly regular languages of *polynomial density* [35]. This class enjoys good complexity properties. In particular, we prove that inclusion between two SLREs is in $\mathrm{NP} \cap \mathrm{coNP}$.
- SLREs form the *core* of known FIFO symbolic representations: more formally, a set of queue contents is SLRE representable iff it is both CQDD representable and QDD representable ($SLREs = QDDs \cap CQDDs$),
- SLREs are usually *sufficient* since for FIFO automata with one channel, an arbitrary iteration of a loop is SLRE representable. Moreover, several examples in the literature have a SLRE representable reachability set: the alternating bit protocol [31], the bounded retransmission protocol of Philips [1], the producer/consumer described in [4] and the connection/deconnection protocol [28].

We say that a labelled transition system is *flat* when its set of traces is included in a SLRE language. We give an algorithm which computes an exact symbolic reachability set of any *flat* labelled transition system whose abstraction has a sound and complete acceleration.

The road map for the paper is as follows: in Section 2 we introduce labelled transition systems, in Section 3 we discuss FIFO automata and symbolic representations based on SLREs.