Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Electronic Notes in Theoretical Computer Science 128 (2005) 21–36
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Regularity Results for FIFO Channels[1]

Nils Klarlund

Bell Labs, Lucent Technologies
Murray Hill, New Jersey

Richard Trefler

School of Computer Science
University of Waterloo, Waterloo, Ontario

Abstract

FIFO channel systems, in which messages between processes are cached in queues, are fundamental to the modeling of concurrency. A great deal of effort has gone into identifying scenarios where reasoning about such systems is decidable, often through establishing that the language of all channel contents is regular. Most prior results in this area focus on the effect of repetitions of individual operations sequences or they constrain the channels either to be lossy or to be polynomially bounded (that is, the number of words of a given length describing channel contents is bounded by a polynomial).

We focus on *piecewise* languages for both describing operations and channel contents. Piecewise languages restrict the Kleene star operation to be applied to sets of letters only. For example, $a(b+c)^*$ is piecewise (but not polynomially bounded). These languages correspond to the Σ_2 class of the first-order quantifier hierarchy. It is already known that piecewiseness plays a key role in establishing regularity results about parameterized systems subjected to rewritings according to semi-commutation rules.

In this paper, we show that piecewiseness is central to the understanding of FIFO channel systems. Our contribution is to study the effect of iterating sets of operations, while extending and unifying previous work on both lossy and perfect FIFO systems. In particular, we show that well-quasi-orderings are important to Σ_2, not only to the lossy systems of Π_1. Moreover, we show that Σ_2 also describes limits in a class of FIFO systems that include iterations of arbitrary sets of simultaneous read and write operations.

Keywords: Regularity, FIFO channels, queues, concurrency.

---

[1] The authors are supported in part by grant CCR-0341658 from the National Science Foundation and by grants from NSERC, CITO, and Nortel.

[2] Email: `klarlund@research.bell-labs.com`

[3] Email: `trefler@cs.uwaterloo.ca`

doi:10.1016/j.entcs.2005.04.002

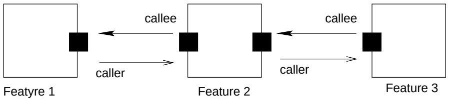

Fig. 1. ECLIPSE call structure

1 Introduction

We show that the class of *piecewise languages* are important to the understanding of finite-state systems that communicate over unbounded channels. Piecewise languages are regular languages that are finite unions of *simply piecewise* languages of the form $M_1^* a_1 \cdots M_n^* a_n M_{n+1}^*$, where the M_i's are subsets of a finite alphabet Σ of symbols and the a_i's are elements of Σ. To express the language consisting of ϵ (the empty word), we allow $n = 0$. Note that if $M_i = \{\}$, then M_i^* is ϵ; also, it can be seen that replacing the a_i's by finite strings of u_i's does not affect the class of languages defined.

Piecewise languages can also be characterized as Σ_2-languages of the quantifier alternation hierarchy for first-order logic on words over the signature $(<, a(\cdot)_{a\in\Sigma})$ or as the level $1\frac{1}{2}$ of the concatenation hierarchy of Straubing-Thérien, see [19]. More simply, piecewise languages are those recognized by nondeterministic automata whose only nontrivial strongly connected components are states with self-loops.

1.1 Motivating example

Our investigations have a practical background. ECLIPSE (now called BoxOS) is a next-generation telephony service over IP infrastructure developed at AT&T Labs; see [8] for our earlier work on model-checking ECLIPSE. Telephone calls are structured as in Figure 1. Boxes at the end points represent telephones, intermediate boxes represent call features, for example call-forwarding-on-busy, and the arrows represent, possibly unbounded, perfect communication channels, or queues, that pass messages from endpoint to endpoint. However, at a sufficiently high level of granularity, each box represents a finite state transducer.

Our focus is on the problems inherent in checking properties of systems composed of several boxes. Consider the *transparent box* described on the left of Figure 2. This transparent box represents a communication template that all system boxes must implement. Figure 2 has been simplified for this presentation in several important ways: the feature box in the picture communicates with only one neighbor, in general, communication may be n-way, and is usually two-way. Simple replication of the one-way communication functionality to create a two-way machine results in a feature with 17 states — the neighbors are not symmetric, one will be the initiating, upstream feature, the other the receiving, downstream feature.