Decidability results of Communicating Finite State Machines over acyclic topology

G. Namratha Reddy
Chennai Mathematical Institute, India
`http://www.cmi.ac.in/~namratha`

Abstract

The paper shows that the reachability problem for concurrent finite state processes which communicate over FIFO queues is decidable over acyclic topologies.

2012 ACM Subject Classification Author: Please fill in 1 or more \ccsdesc macro

Keywords and phrases Distributed Systems, Reachability, Finite state machines with FIFO queues, Network of concurrent processes

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

LIPIcs is a series of open access high-quality conference proceedings across all fields in informatics established in cooperation with Schloss Dagstuhl. In order to do justice to the high scientific quality of the conferences that publish their proceedings in the LIPIcs series, which is ensured by the thorough review process of the respective events, we believe that LIPIcs proceedings must have an attractive and consistent layout matching the standard of the series. Moreover, the quality of the metadata, the typesetting and the layout must also meet the requirements of other external parties such as indexing service, DOI registry, funding agencies, among others. The guidelines contained in this document serve as the baseline for the authors, editors, and the publisher to create documents that meet as many different requirements as possible.

Please comply with the following instructions when preparing your article for a LIPIcs proceedings volume.

Minimum requirements

- Use pdflatex and an up-to-date LaTeX system.
- Use further LaTeX packages and custom made macros carefully and only if required.
- Use the provided sectioning macros: `\section`, `\subsection`, `\subsubsection`, `\paragraph`, `\paragraph*`, and `\subparagraph*`.
- Provide suitable graphics of at least 300dpi (preferably in PDF format).
- Use BibTeX and keep the standard style (`plainurl`) for the bibliography.
- Please try to keep the warnings log as small as possible. Avoid overfull `\hboxes` and any kind of warnings/errors with the referenced BibTeX entries.
- Use a spellchecker to correct typos.

2 System topology

2.1 Model

We work with a topology which we describe using the tuple $(P, C, Reader, Writer)$ where P is the set of processes, C is the set of channels, $Reader$ is a function $C \to P$ and $Writer$ is a function $C \to P$. Also, we assume that no channel has the same reader and writer.

Each process is a finite state transition system $TS_i = (Q_i, \Sigma_i, \delta_i, s_i)$

Since we are interested only in control state reachability and not in language theory, we will ignore the alphabet and final states. However the transitions may also have associated actions, which in our case may be sending a mes- sage to a channel or receiving a message from a channel.

δ_i has q -> q' on c?a if is a reader, c!a if it is a writer or a nop

Each channel holds messages coming from a finite set i.e. $c_i \in C$ can have messages from M_i

2.2 Configuration Graph

A global configuration of the system would consist of the states of each process and the channel contents of each channel. So if we have n processes and m channels, a configuration would be the tuple $(q_0, q_1, ..., q_n, \gamma_1, \gamma_2, ..., \gamma_m)$

that is when a transition from one tuple to the next happens what all should align in the universe.

A run is a path in this graph

3 The Reachability Problem

The reachability problem is to ask whether we can reach a target configuration where one or more processes are in a target state.

We know that the problem is undecidable in general (cite), because if there is a loop in the topology then we can simulate a queue machine and state reachability is undecidable for queue machines.

So we ask the question of whether it is decidable for acyclic topologies.

What we mean by acyclic topologies is consider the network topology and ignore the direction of the edges, so we get an undirected graph and this graph should have no cycles.

We solve this with the help of two reductions

Reduction 1

given such an acyclic topology the reachability problem can be reduced to one in which the target state is reached only if the queue is empty (cite Madhu)

Reduction 2

We reduce it to another isomorphic topology that looks like a tree, where every process has one incoming edge (except the root) i.e every process can read from one channel but write to multiple channels (cite Madhu)

▶ **Theorem 1.** *The Reachability problem is decidable for CFMs with FIFO channels over undirected topology.*

Proof. We use reductions 1 and 2 to get a tree topology ◀

Nec urna malesuada sollicitudin. Nulla facilisi. Vivamus aliquam tempus ligula eget ornare. Praesent eget magna ut turpis mattis cursus. Aliquam vel condimentum orci. Nunc congue, libero in gravida convallis [1], orci nibh sodales quam, id egestas felis mi nec nisi. Suspendisse tincidunt, est ac vestibulum posuere, justo odio bibendum urna, rutrum bibendum dolor sem nec tellus.

▶ **Lemma 2** (Quisque blandit tempus nunc). *Sed interdum nisl pretium non. Mauris sodales consequat risus vel consectetur. Aliquam erat volutpat. Nunc sed sapien ligula. Proin faucibus sapien luctus nisl feugiat convallis faucibus elit cursus. Nunc vestibulum nunc ac massa pretium pharetra. Nulla facilisis turpis id augue venenatis blandit. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.*

Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.

4 Conclusions

Morbi eros magna, vestibulum non posuere non, porta eu quam. Maecenas vitae orci risus, eget imperdiet mauris. Donec massa mauris, pellentesque vel lobortis eu, molestie ac turpis. Sed condimentum convallis dolor, a dignissim est ultrices eu. Donec consectetur volutpat eros, et ornare dui ultricies id. Vivamus eu augue eget dolor euismod ultrices et sit amet nisi. Vivamus malesuada leo ac leo ullamcorper tempor. Donec justo mi, tempor vitae aliquet non, faucibus eu lacus. Donec dictum gravida neque, non porta turpis imperdiet eget. Curabitur quis euismod ligula.

References

1 John E. Hopcroft, Wolfgang J. Paul, and Leslie G. Valiant. On time versus space and related problems. In *16th Annual Symposium on Foundations of Computer Science, Berkeley, California, USA, October 13-15, 1975*, pages 57–64. IEEE Computer Society, 1975. `doi:10.1109/SFCS.1975.23`.

A Styles of lists, enumerations, and descriptions

List of different predefined enumeration styles:

- `\begin{itemize}...\end{itemize}`
- ...
- ...

1. `\begin{enumerate}...\end{enumerate}`
2. ...
3. ...

(a) `\begin{alphaenumerate}...\end{alphaenumerate}`
(b) ...
(c) ...

(i) `\begin{romanenumerate}...\end{romanenumerate}`
(ii) ...
(iii) ...

(1) `\begin{bracketenumerate}...\end{bracketenumerate}`

(2) ...
(3) ...

Description 1 `\begin{description} \item[Description 1]  ...\end{description}`
Description 2 Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.
Description 3 ...

Proposition 6 and Proposition 6 ...

B Theorem-like environments

List of different predefined enumeration styles:

▶ **Theorem 3.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

▶ **Lemma 4.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

▶ **Corollary 5.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

▶ **Proposition 6.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

▶ **Exercise 7.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

▶ **Definition 8.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

▶ **Example 9.** Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.

▶ Note 10. Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.

▶ Note. Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.

▶ Remark 11. Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.

▶ Remark. Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.

▷ Claim 12. Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.

▷ Claim. Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.

Proof. Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque. ◀

Proof. Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque. ◁