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 Theory of computation → Distributed computing models

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

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.

2 System topology

2.1 Model

We work with a topology which we describe using the tuple $(P, C, \mathit{Reader}, \mathit{Writer})$ 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 [1]

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 [1]

▶ **Lemma 1.** *Given a directed tree topology with r=0 as the root of the tree* $w \in L_r^e \implies (s_0, s_1, ..., s_n, w\downarrow_{r(0)}, \epsilon, ..., \epsilon) \rightarrow_G^* (f_0, f_1, ..., f_n, \epsilon, \epsilon, ...\epsilon)$

Proof. Induction on the no. of nodes in the directed tree topology

Base case: We have only one node r, $L_r^e = \epsilon \implies \epsilon \in L_r \implies s_r = f_r$, clearly since we start in s_r it is reachable

Induction: Let r be a non-leaf node and let the children be $k_1, k_2, ..., k_m$, $w \in L_r^e \implies w \in L_i \cap shuffle(L_{k_1}^e\downarrow_r, L_{k_2}^e\downarrow_r, ..., L_{k_m}^e\downarrow_r, M_{r(i)}^*)$ w is of the form $shuffle(w_1, w_2, ..., w_m)$ where $w_i \in L_{k_i}^e\downarrow_{r(i)}$

$w_i \downarrow_{r(k_i)} = w_i' \downarrow r(k_i)$ where $w_i' \in L_{k_i}^e$

Let π_i be the tree that is rooted by k_i By induction hypothesis, since $w_i' \in L_{k_i}^e$ we have run from nodes in π are in start state, and reader channel of k_i has $w_i' \downarrow r(k_i)$ to a final configuration where every node is in the final state and all channels are empty.

Since w is in L_r^e it means it is also in L_r we can take this run and convert to the corresponding run in the configuration graph where only r is moving, and this w can generate $w_i' \downarrow r(k_i)$ in each k_i's reader channel, so we can start from $(s_0, s_1, ..., s_n, \epsilon...)->$ (only r moves and fills up all the channels)$->$ $(f_0, s_1, ..., s_n, \epsilon, w_i \downarrow_{r(k_i)}, ...w_i \downarrow_{r(k_i)})->$ $stitchthek_irunoneaftertheother$

◀

▶ **Lemma 2.** *Given a directed tree topology with i=0 as the root of the tree.*

$(s_0, s_1, ..., s_n, \alpha, \epsilon, ..., \epsilon) \rightarrow_G^* (f_0, f_1, ..., f_n, \epsilon, \epsilon, ...\epsilon) => \exists w \in L_i^e, \alpha \downarrow_{r(0)} = w \downarrow_{r(0)}$

Proof. Base case : There is only one node α has to be epsilon

Induction: Let r be a non-leaf node, and $k_1, k_2, ..., k_n$ be the children of the r and let the tree rooted by them be $\pi_1, \pi_2, .., \pi_n$

Let r run and fill all the reader channels with α_i For each k_i we form a run from where only nodes in π_i are in start state and α_i is in the reader channel of k_i, there and final state is reached. Now we use induction hypothesis to say that there is a word in $w_i \in L^e_{k_i}$ and $\alpha_i \downarrow_{r(0)} = w_i \downarrow_{r(0)}$

So we find a w_i for each k_i

so we have $w_i \downarrow_{r(i)} \in L^e_i$ so we have $\alpha_i \in L^e_i$

Consider the word $w \in shuffle(\alpha_1, \alpha_2, ...\alpha_n, \alpha)$ $walsobelongstoL_r$

$w \in L^e_r$ and $w \downarrow_{r(r)} = \alpha$

◀

▶ **Theorem 3.** *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

◀

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 La Torre, S., Madhusudan, P., Parlato, G. (2008). Context-Bounded Analysis of Concurrent Queue Systems. In: Ramakrishnan, C.R., Rehof, J. (eds) Tools and Algorithms for the Construction and Analysis of Systems. TACAS 2008. Lecture Notes in Computer Science, vol 4963. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-78800-3_21