Synchronizability of Communicating Finite State Machines is not Decidable

Alain Finkel, Etienne Lozes

Synchronizability of Communicating Finite State Machines is not Decidable

Alain Finkel[1] **and Etienne Lozes**[1]

1 **LSV, ENS Cachan, CNRS, France**
{finkel,lozes}@lsv.fr

Abstract

A system of communicating finite state machines is *synchronizable* [1, 4] if its send trace semantics, i.e. the set of sequences of sendings it can perform, is the same when its communications are FIFO asynchronous and when they are just rendez-vous synchronizations. This property was claimed to be decidable in several conference and journal papers [1, 4, 3, 2]. In this paper, we show that synchronizability is actually undecidable. We show that synchronizability is decidable if the topology of communications is an oriented ring. We also show that, in this case, synchronizability implies the absence of unspecified receptions and orphan messages, and the channel-recognizability of the reachability set.

Keywords and phrases verification, distributed system, asynchronous communications, choreographies

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

Asynchronous distributed systems are error prone not only because they are difficult to program, but also because they are difficult to execute in a reproducible way. The slack of communications, measured by the number of messages that can be buffered in a same communication channel, is not always under the control of the programmer, and even when it is, it may be delicate to to choose the right size of the communication buffers.

Slack elasticity of a distributed system with asynchronous communications is the property that the "observable behaviour" of the system is the same whatever the slack of communications is. There are actually as many notions of slack elasticity as there are notions of observable behaviours (and of distributed systems). Slack elasticity has been studied in various contexts: for hardware design [15], with the goal of ensuring that some code transformations are semantic-preserving, for parallel programming in MPI [17, 18], for ensuring the absence of deadlocks and other bugs, or more recently for web services and choreographies [1, 4, 2], for verifying various properties, among which choreography realizability [3].

This paper focuses on *synchronizability* [1], a special form of slack elasticity that was defined by Basu and Bultan for analyzing choreographies. Synchronizability is the slack elasticity of the send trace semantics of the system: a system of communicating finite state machines is synchronizable if any asynchronous trace can be mimicked by a synchronous one that contains the same send actions in the same order. Synchronizability was claimed decidable [4, 2], by contrast with many other properties of systems of communicating finite state machines (including deadlock-freedom, absence of orphan messages, boundedness, etc) that are undecidable for systems of just two machines [6]. The proof relied on the claim that synchronizability would be the same as 1-synchronizability, which states that any 1-bounded trace can be mimicked by a synchronous trace.