Partial Order Reduction for Real Time ABS

December 2017

candidate	Martin Mellum Jackson
supervisors	Lars Tveito and Einar Broch Johnsen
group	SIRIUS
type	60ECTS
study program	Informatics: Programming and Networks
planned date of completion	May 2019

Short description

The task is to adapt *dynamic partial order reduction* [**DBLP:journals/jacm/AbdullaAJS17**] to Real-Time ABS [**DBLP:journals/jlp/JohnsenST15**].

Background and motivation

Real-Time ABS is an executable modeling language, which is suitable for modeling distributed systems at a high level of abstraction. The language is object-oriented and features concurrent object groups that communicate via asynchronous method calls and futures. Models in Real-Time ABS are nondeterministic, meaning that executing the same model twice might not yield the same answer. It follows that tests in such a model can sometimes pass and sometimes fail, making testing such models reliably a very challenging task.

To strengthen testing facilities for Real-Time ABS, we want to be able to explore many different runs of a model. The number of different possible runs of a heavily concurrent model can be extremely large. However, many of these runs may be equivalent and therefore unnecessary to explore. Detecting equivalent runs is a crucial prerequisite for any exploration-based testing framework for Real-Time ABS.

The most promising technique for establishing equivalence between different runs is *partial order reduction* (POR) [**DBLP:journals/sttt/ClarkeGMP99**]. In recent years, *dynamic* POR [**DBLP:journals/jacm/AbdullaAJS17**] has been shown to guarantee *optimal* partial order reduction. Furthermore, similar techniques have been adapted to actor systems [**DBLP:conf/forte/AlbertAG14**] similar to Real-Time ABS.

Problem description

The thesis will consist of both conceptual and implementation work. On the conceptual side, the dynamic POR algorithm must be adapted to the semantics of Real-Time ABS. In addition, the algorithm should be optimized for a distributed environment, as the Real-Time ABS simulator is implemented as a distributed system in Erlang. Finally, a prototype of the proposed algorithm will be implemented as a part of the Real-Time ABS simulator.