Contribution ID: **111** Type: **not specified**

Checking Refinement of Asynchronous Programs against Context-Free Specifications

Thursday, July 13, 2023 4:00 PM (20 minutes)

Pascal Baumann, Moses Ganardi, Rupak Majumdar, Ramanathan Thinniyam Srinivasan, and Georg Zetzsche

Abstract: In the language-theoretic approach to refinement verification, we check that the
language of traces of an implementation all belong to the language of a specification.
We consider the refinement verification problem for asynchronous programs against specifications given by a Dyck
language.
We show that this problem is EXPSPACE-complete—the same complexity as that of language emptiness and for
refinement verification against a regular specification.
Our algorithm uses several novel technical ingredients.
First, we show that checking if the coverability language of a succinctly described
vector addition system with states (VASS) is contained in a Dyck language
is EXPSPACE-complete.
Second, in the more technical part of the proof, we define an ordering on words and show a downward closure construction that
allows replacing the (context-free) language of each task in an asynchronous program by a regular language.
Unlike downward closure operations usually considered in infinite-state verification, our ordering is not a well-quasi-ordering, and we have
to construct the regular language ab initio.
Once the tasks can be replaced, we show a reduction to an appropriate VASS and use our first ingredient.
In addition to the inherent theoretical interest, refinement verification with Dyck specifications captures common
practical resource usage patterns based on reference counting, for which few algorithmic techniques were known.

Presenter: BAUMANN, Pascal

Session Classification: Track B