Platform-Based Software Synthesis and Verification Using Contracts

Antonio Iannopanllo, Pierluigi Nuzzo, Stavros Tripakis, Alberto Sangiovanni-Vincentelli
University of California, Berkeley, EECS Department

We propose a tool that enables the design of reactive control systems by either using pre-defined library blocks or by defining new blocks. All components are represented as assume/guarantee contracts. The tool seamlessly interacts with other pre-existing software to allow for the verification of a composite model against a formal specification (property) and the synthesis of valid implementations. Operations on contracts are used to improve performance both in property verification and synthesis.

Goals
✓ Provide a unified modeling interface to the designer
✓ Provide a consistent workflow for distributed reactive control synthesis
✓ Enable effective design space exploration

Motivation
➢ Need for correct-by-construction design methodology of critical systems
➢ Need for scalable solutions and compositional approaches
➢ Need to manage system complexity

References

Background
Generalized Reactive (1) LTL
• Subset of LTL, polynomial complexity in the synthesis of reactive controls
• Specifications given in the form:
  \[ \phi_A \Rightarrow \phi_C \]
• Solved as a game between the environment and the system
• The system wins if it can always find a proper state for each environment action or if the environment makes a not allowed action

Assume/Guarantee Contracts

Library Definition
• 10 Contracts (5 refinement relations)
• Library refinement check time: ~ 0.8 sec (NuSMV based)

EPS model
• Without abstraction: 21 Variables
• With max abstraction level: 11 Variables

Results for the EPS example

Property Verification - No parallel between G_A and G_B
If G_A and G_B not in failure, C_I and C_J must be opened

Property Verification - DC bus should always be powered
If at least one generator is healthy, C_I and C_J must always be closed

Conclusion
• Framework for the design of software controllers based on libraries of LTL A/G contracts
• Abstraction, performed by leveraging contract refinement relations, to speed-up the verification of properties of the designed model
• Application of the framework to a concrete example
• Future work includes the realization of a robotic-domain contract library and its application on a related case study