COUNTER-STRATEGY GUIDED REFINEMENT OF GR(1) TEMPORAL LOGIC SPECIFICATIONS

Salar Moarref
Joint work with Rajeev Alur & Ufuk Topcu
University of Pennsylvania
August 2013
Reactive Synthesis

- Specification
  - LTL, CTL, ...
- Game
  - Environment
  - System
- Realizability?
  - Yes: strategy
  - No: counter-strategy
Motivation

- Developing a correct and complete formal specification is challenging and tedious
  - initial specifications are often unrealizable
  - due to inadequate environment assumptions
- Unrealizable specification cannot be executed or simulated
  - Debugging an unrealizable specification is hard
Motivation

- Developing a correct and complete formal specification is challenging and tedious
  - initial specifications are often unrealizable
  - due to inadequate environment assumptions
- Unrealizable specification cannot be executed or simulated
  - Debugging an unrealizable specification is hard

Goal
Automatically refining the constraints over the environment by adding assumptions in order to achieve realizability.
Applications

- Giving the user an insight into the specification
- Correcting the specification
- Constructing interface rules between the components in the context of compositional synthesis
- And more...
Main Flow of the Method

1. Specification
2. Generating Candidates
3. Counter-Strategy
4. Patterns Synthesis
5. Subset of Variables

Choose & Add

Realizable
- Yes: Done
- No:
  - Generating Candidates
  - Counter-Strategy
  - Patterns Synthesis
  - Subset of Variables
Specification

Choose & Add

Generating Candidates

Subset of Variables

Patterns Synthesis

Counter-Strategy

Realizable

Yes

Done

No
Patterns

- Environment behaviors learnt from the counter-strategy as LTL formulas of the form
  - $\Diamond \Box \psi$, $\Diamond \psi$, $\Diamond (\psi_1 \land \Diamond \psi_2)$

- Hold over all runs of the abstraction of the counter-strategy

- Synthesized using simple graph search algorithms

Generalized Reactivity(1) (GR(1))

- Environment assumption, $\varphi_{\text{env}}$

  $$\theta_{\text{init}}^e \land \bigwedge_{i \in I_e} \Box \psi_i^e \land \bigwedge_{k \in K_e} \Diamond J_k^e$$

- System requirement, $\varphi_{\text{sys}}$

  $$\theta_{\text{init}}^s \land \bigwedge_{i \in I_s} \Box \psi_i^s \land \bigwedge_{k \in K_s} \Diamond J_k^s$$

- Initial conditions
- Safety + transitions
- Fairness + goals
Abstraction of the Counter-Strategy
Eventually Always Patterns

- Complement of liveness formulas
- $\Diamond \Box (q_1 \lor q_2 \lor q_3)$

![Diagram showing strongly connected components including cycle]

$q_0$ connected to $q_1$, $q_1$ to $q_2$, $q_2$ to $q_1$, $q_3$ to $q_3$, and $q_3$ to $q_0$.
Generating Candidate Assumptions

- Replace each state in pattern with corresponding state predicate
  - $\Diamond \Box (q_1 \lor q_2 \lor q_3)$ leads to
  - $\Diamond \Box ((c \land r) \lor (c \land \neg r) \lor (c \land r)) = \Diamond \Box c$

- Complement the formula
  - $\Box \Diamond \neg c$

**Diagram:**
- State predicate: $c \land r$
- States S0, S1, S2, S3 with transitions:
  - S0: 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0
  - S1: 0, 0, 0, 0
  - S2: 0, 1, 0, 0, r=False
  - S3: 0, 1, 1, 0, r=True

**Nodes:**
- $q_0$, $q_1$, $q_2$, $q_3$ with transitions:
  - $q_0 \rightarrow q_1 \rightarrow q_2 \rightarrow q_3$
Conclusions & Future Work

- **Summary**
  - Refining the unrealizable specification by adding assumptions
    - Simple GR(1) formulas
    - As weak as possible in the specified structure

- **Future work**
  - Taking advantage of multiplicity of generated candidates
  - Extending the method to more general subsets of LTL
  - Synthesizing interfaces between components