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

Salar Moarref
Joint work with Rajeev Alur & Ufuk Topcu
University of Pennsylvania

June 2013
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

- Specification
  - Generating Candidates
    - Subset of Variables
      - Patterns Synthesis
    - Counter-Strategy
      - Yes
        - Done
      - No
        - Choose & Add
          - Realizable
            - Yes
              - Done
            - No
              - Choose & Add
Abstraction of the Counter-Strategy

<table>
<thead>
<tr>
<th>S0</th>
<th>0,1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>r=True</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S1</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>r=True</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S2</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S3</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>r=False</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>r=True</td>
</tr>
</tbody>
</table>

\[
q_0 \xrightarrow{0} q_1 \xrightarrow{r=True} q_2 \xrightarrow{r=False} q_3
\]
Patterns

- LTL formulas of special 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))

<table>
<thead>
<tr>
<th>Environment assumption, $\mathcal{Q}_{env}$</th>
<th>System requirement, $\mathcal{Q}_{sys}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_{init}^e \land \bigwedge_{i \in I_e} \Box \psi_i^e \land \bigwedge_{k \in K_e} \Diamond J_k^e$</td>
<td>$\theta_{init}^s \land \bigwedge_{i \in I_s} \Box \psi_i^s \land \bigwedge_{k \in K_s} \Diamond J_k^s$</td>
</tr>
</tbody>
</table>

- initial conditions
- safety + transitions
- fairness + goals
Eventually Always Patterns

- Complement of liveness formulas
- $\diamondsuit \square (q_1 \lor q_2 \lor q_3)$
Generating Candidate Assumptions

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

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

![Diagram showing state transitions and state predicate: $c \land r$]
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