Synthesis with Mandatory Stop Actions

Giuseppe De Giacomo, Antonio Di Stasio, Giuseppe Perelli, Shufang Zhu

Sapienza Università di Roma
{degiacomo,distasio,perelli.zhu}@diag.uniroma1.it

ERC Advanced Grant
WhiteMech:
White-box Self Programming Mechanisms

Synthesis under Environment Specifications

- Automatically synthesize an agent strategy with respect to a priori knowledge of how the environment works [1].

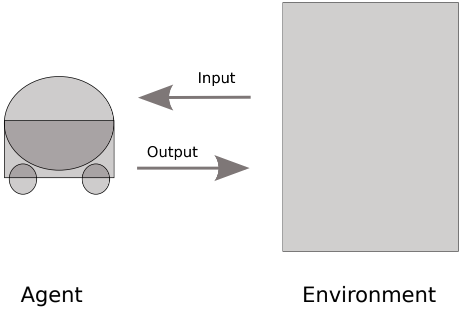

Figure: Reactive System

[1] Pnueli and Rosner, "On the Synthesis of a Reactive Module", 1989

LTL$_f$ Synthesis under LTL Specifications[2]

Given:

- Environment variables $\mathcal{X}$, Agent variables $\mathcal{Y}$
- Agent task φ^a_{task} in LTL$_f$, Environment specification φ^e in LTL

Obtain:

Agent strategy $\sigma_{ag} : (2^{\mathcal{X}})^+ \rightarrow 2^{\mathcal{Y}}$, a function from past history of environment behaviors to agent actions

$$\forall \sigma_{env} \ \rhd \varphi^e, play(\sigma_{ag}, \sigma_{env})^k \models \varphi^a_{task} \text{ for some } k \in \mathbb{N}$$

φ^a_{task} describes the desired goal/task when the environment behaviors satisfy the specification φ^e.

[2] Aminof et al., "Planning and Synthesis Under Assumptions", 2018

Challenges and Successes

- Environment specification φ^e in LTL
  - Büchi determinization, intractable in practice
- Restrictions on the form of the specification φ^e LTL
  - Safe LTL, co-Safe LTL[3], φ^e still contributes to game arena construction
  - Simple Fairness (*infinitely-often*) and Stability (*eventually-always*), GR(1)[4,5], restricted expressiveness

[3] Camacho, Bienvenu, and McIlraith, "Finite LTL Synthesis with Environment Assumptions and Quality Measures", 2018

[4] Zhu et al., "LTL_f Synthesis with Fairness and Stability Assumptions", 2020

[5] Giacomo et al., "Finite-Trace and Generalized-Reactivity Specifications in Temporal Synthesis", 2021

Synthesis with Mandatory Stop Actions

- NO restrictions on environment specification φ^e
- A fundamental requirement on agent strategies
  - Obligatorily conduct the **stop action** in her strategies
  - **Stop action**: the agent cannot do anything anymore

Impacts of Mandatory Stop Action

- Agent cannot wait till the environment spontaneously brings favorable conditions

Example - Shared kitchen

φ^{a}_{task} = "dish cleaned"
φ^{e} = "eventually somebody else will do the dishes"
σ_{ag} = "waiting until somebody does the dishes" is not a winning strategy

This is because when the dishes are done, and the agent can stop, would **not** be controlled by the agent

Synthesis with Mandatory Stop Actions

- Simpler synthesis algorithms, sidestep Büchi determinization from φ^e
- Every LTL formula formed by a safety part and a liveness part[6]

$$\varphi^e = \varphi^e_{safe} \wedge \varphi^e_{live}$$

- Safety “bad" things never happen
- Liveness “good" things eventually happen

[6] Alpern and Schneider, “Recognizing Safety and Liveness”, 1987

Synthesis with Mandatory Stop Actions

$$\varphi^e = \varphi^e_{safe} \wedge \varphi^e_{live}$$

- Safety “bad" things never happen
  - The environment has to maintain the safety for every finite prefix
  - No matter when *stop* is performed
  - Agent can exploit φ^e_{safe} to achieve φ^a_{task}

Synthesis with Mandatory Stop Actions

$$\varphi^e = \varphi^e_{safe} \wedge \varphi^e_{live}$$

- Safety "bad" things never happen
- Liveness "good" things eventually happen
  - Maintain safety, and satisfy liveness in the future
  - φ^e_{live} can be satisfied after *stop* is performed
  - Agent cannot exploit φ^e_{live} to achieve φ^a_{task}

Synthesis with Mandatory Stop Actions

Theorem

Let $\mathcal{P} = \langle \mathcal{X}, \mathcal{Y}, \varphi^e, \varphi^a_{task} \rangle$ be the synthesis problem with mandatory stop actions and σ_{ag} an agent strategy.

$$\sigma_{ag} \text{ realizes } \mathcal{P} \text{ iff } \sigma_{ag} \text{ realizes } \hat{\mathcal{P}} = \langle \mathcal{X}, \mathcal{Y}, \varphi^e_{safe}, \varphi^a_{task} \rangle.$$

Synthesis Technique

Given:

- Synthesis problem $\mathcal{P} = \langle \mathcal{X}, \mathcal{Y}, \varphi^e, \varphi^a_{task} \rangle$, agent task φ^a_{task} in LTL$_f$, environment specification φ^e in LTL

Solution:

1. Abstract φ^e_{safe} such that $\hat{\mathcal{P}} = \langle \mathcal{X}, \mathcal{Y}, \varphi^e_{safe}, \varphi^a_{task} \rangle$

2. Solve $\hat{P}$

Synthesis Technique

Key step: Abstract φ^e_{safe} such that $\hat{\mathcal{P}} = \langle \mathcal{X}, \mathcal{Y}, \varphi^e_{safe}, \varphi^a_{task} \rangle$

1. Build NBA $\mathcal{N}^e$ of φ^e
2. Build NBA $\mathcal{N}^e_s$ of φ^e_{safe} by marking all states of $\mathcal{N}^e$ accepting
3. Build DA $\mathcal{D}^e_s$ of φ^e_{safe} by **subset construction**

Synthesis Technique

We now have:

- Synthesis problem $\hat{P} = \langle \mathcal{X}, \mathcal{Y}, \mathcal{D}^e_s, \varphi^a_{task} \rangle$, $\mathcal{L}(\varphi^e_{safe}) = \mathcal{L}(\mathcal{D}^e_s)$

1. Restrict the environment to considering $\sigma_{env} \rhd \varphi^e_{safe}$
2. Build the DA of φ^a_{task}
3. Combine the two DAs and solve the reachability game over the resulting automaton