Synthesis via Sampling-Based Abstractions
Some Problems and Initial Ideas

Matthias Rungger\textsuperscript{2}  Morteza Lahijanian\textsuperscript{1}
Lydia E. Kavvaki\textsuperscript{1}  Paulo Tabuada\textsuperscript{2}  Moshe Y. Vardi\textsuperscript{1}

\textsuperscript{1}Department of Computer Science, Rice University
\textsuperscript{2}Cyber-Physical Systems Laboratory, UCLA
**Problem statement**

Given a LTL specification $\varphi$ and a control system $S$, find a controller $C$ that enforces $\varphi$ on $S$.
Problem statement
Given a LTL specification $\varphi$ and a control system $S$, find a controller $C$ that enforces $\varphi$ on $S$

Well-known abstraction/refinement approach

1. Compute a finite abstraction $\hat{S}$ of $S$
2. Synthesize controller $\hat{C}$ based on $\hat{S}$
3. Refine solution $\hat{C}$ to $C$
Problem statement
Given a LTL specification $\varphi$ and a control system $S$, find a controller $C$ that enforces $\varphi$ on $S$

Well-known abstraction/refinement approach

1. Compute a finite abstraction $\hat{S}$ of $S$
2. Synthesize controller $\hat{C}$ based on $\hat{S}$
3. Refine solution $\hat{C}$ to $C$

$\implies$ All done ✓
Problem statement
Given a LTL specification $\varphi$ and a control system $S$, find a controller $C$ that enforces $\varphi$ on $S$

Well-known abstraction/refinement approach

1. Compute a finite abstraction $\hat{S}$ of $S$
2. Synthesize controller $\hat{C}$ based on $\hat{S}$
3. Refine solution $\hat{C}$ to $C$

$\Rightarrow$ All done ✓

So... what is the problem?
1D: Temperature

\[ \dot{T} = c(T_{env} - T) \]
Computing Abstractions

1D: Temperature

\[ \dot{T} = c(T_{env} - T) \]
Computing Abstractions

1D: Temperature

\[ \dot{T} = c(T_{env} - T) \]

\[ |\hat{X}| = 100 \checkmark \]
Computing Abstractions

1D: Temperature

\[ \dot{T} = c(T_{env} - T) \]

2D: Pendulum

\[ |\hat{X}| = 100 \]
Computing Abstractions

1D: Temperature

\[ \dot{T} = c(T_{env} - T) \]

2D: Pendulum

| \hat{X} | = 100 ✓
Computing Abstractions

1D: Temperature

\[ \dot{T} = c(T_{env} - T) \]

2D: Pendulum

3D: Unicycle Robot

4D: Pendulum on a cart

\[ |\hat{X}| = 100 \checkmark \]
\[ |\hat{X}| = 100^2 \checkmark \]
Computing Abstractions

1D: Temperature

\[ \dot{T} = c(T_{env} - T) \]

2D: Pendulum

3D: Unicycle Robot

\[ |\hat{X}| = 100 \quad \checkmark \quad |\hat{X}| = 100^2 \quad \checkmark \]
Computing Abstractions

1D: Temperature
\[ \dot{T} = c(T_{env} - T) \]

2D: Pendulum

3D: Unicycle Robot

3/6
Computing Abstractions

1D: Temperature

\[ \dot{T} = c(T_{env} - T) \]

2D: Pendulum

3D: Unicycle Robot

\[ |\hat{X}| = 100 \checkmark \]

\[ |\hat{X}| = 100^2 \checkmark \]

\[ |\hat{X}| = 100^3 ?？ \]
Computing Abstractions

1D: Temperature
\[ \dot{T} = c(T_{env} - T) \]

2D: Pendulum

3D: Unicycle Robot

4D: Pendulum on a cart

| \hat{X} | = 100 ✓ | \hat{X} | = 100^2 ✓ | \hat{X} | = 100^3 ?? |
Computing Abstractions

1D: Temperature

\[ \dot{T} = c(T_{\text{env}} - T) \]

2D: Pendulum

3D: Unicycle Robot

4D: Pendulum on a cart

\[ |\hat{X}| = 100 \quad \checkmark \]
\[ |\hat{X}| = 100^2 \quad \checkmark \]
\[ |\hat{X}| = 100^3 \quad ?? \]
\[ |\hat{X}| = 100^4 \]
Sampling-based Ideas to Compute Abstractions

Synergistic approach for syntactically co-safe LTL

- Lower layer:
  use sampling-based methods to grow the abstraction

- Higher layer:
  use Büchi automaton (from $\varphi$) and environment geometry to guide the expansion

- Use “synergistic” layer to alternate between layers

A. Bhatia, L. E. Kavraki, and M. Y. Vardi. “Sampling-based motion planning with temporal goals”. In: ICRA. IEEE, 2010


http://msl.cs.uiuc.edu/~lavalle/
Sampling-based Ideas to Compute Abstractions

**Synergistic approach for syntactically co-safe LTL**

- Lower layer:
  use sampling-based methods to grow the abstraction

- Higher layer:
  use Büchi automaton (from \(\varphi\)) and environment geometry to guide the expansion

- Use “synergistic” layer to alternate between layers

**Solution (point-to-point)**

http://msl.cs.uiuc.edu/~lavalle/

curse of dimensionality is no problem
Sampling-based Ideas to Compute Abstractions

**Synergistic approach for syntactically co-safe LTL**

- Lower layer: use sampling-based methods to grow the abstraction
- Higher layer: use Büchi automaton (from $\varphi$) and environment geometry to guide the expansion
- Use “synergistic” layer to alternate between layers

**Solution (point-to-point)**

The curse of dimensionality is no problem

Problem solved?
What if we have... a set of initial states?

$X_{init}$
What if we have...

a set of initial states?

- Solve problem for some samples of $X_{init}$
What if we have...

**a set of initial states?**

- Solve problem for some samples of $X_{init}$
- Can we use local controllers to enlarge/robustify solutions?
What if we have...

a set of initial states?

| Solve problem for some samples of $X_{\text{init}}$ |
| Can we use local controllers to enlarge/robustify solutions? |

safety specifications? (infinite behavior)
What if we have...

a set of initial states?

- Solve problem for some samples of $X_{init}$
- Can we use local controllers to enlarge/robustify solutions?

safety specifications? (infinite behavior)

- What are good heuristics to grow the abstraction?
What if we have...

a set of initial states?

\[ X_{init} \]

- Solve problem for some samples of \( X_{init} \)
- Can we use local controllers to enlarge/robustify solutions?

safety specifications? (infinite behavior)

- What are good heuristics to grow the abstraction?
- How to find loops?
What if we have...

A set of initial states?

- Solve problem for some samples of $X_{\text{init}}$
- Can we use local controllers to enlarge/robustify solutions?

Safety specifications? (infinite behavior)

- What are good heuristics to grow the abstraction?
- How to find loops?
- Can we merge close-by samples?
To answer those questions we combine

**Sampling-based planning (Rice)**

Morteza Lahijanian

Lydia Kavraki
To answer those questions we combine

**Sampling-based planning (Rice)**

Morteza Lahijanian

Lydia Kavraki

**Control theory (UCLA)**

Matthias Rungger

Paulo Tabuada
To answer those questions we combine

**Sampling-based planning (Rice)**

- Morteza Lahijanian
- Lydia Kavraki

**Control theory (UCLA)**

- Matthias Rungger
- Paulo Tabuada

Are we satisfied?
To answer those questions we combine:

**Sampling-based planning (Rice)**

- Morteza Lahijanian
- Lydia Kavraki

**Control theory (UCLA)**

- Matthias Rungger
- Paulo Tabuada

**Reactive synthesis (Rice)**

- Moshe Vardi