Incremental Synthesis of Switching Protocols via Abstraction Refinement

Petter Nilsson and Necmiye Ozay

December 17, 2014

UNIVERSITY OF MICHIGAN

Outline

Outline

Motivation

- Provably-correct controller synthesis using discrete abstractions is a hot topic.
- Typical workflow: system $+$ spec $\rightarrow$ discrete abstraction $\rightarrow$ discrete synthesis $\rightarrow$ continuous implementation.
- Two issues
  - Bottlenecks: large discrete abstractions, discrete synthesis expensive.
  - More focus on finding a controller, also valuable to find counter-examples.

Motivation

- Provably-correct controller synthesis using discrete abstractions is a hot topic.
- Typical workflow: system $+$ spec $\rightarrow$ discrete abstraction $\rightarrow$ discrete synthesis $\rightarrow$ continuous implementation.
- Two issues
  - Bottlenecks: large discrete abstractions, discrete synthesis expensive.
  - More focus on finding a controller, also valuable to find counter-examples.
- This paper: do adaptive abstractions and search for both control protocols and certificates of non-realizability.

Switched system

Continuous-time switched system:

$$\mathcal{S} = (X, \mathcal{A}, \{f_a\}_{a\in\mathcal{A}}, D). \tag{1}$$

- $X \subset \mathbb{R}^n$: Domain
- $\mathcal{A} = \{a_1, \ldots, a_s\}$: Modes
- $\{f_a\}_{a\in A}$: Vector fields
- $D \subset \mathbb{R}^d$: Disturbance set

Evolution of the state is governed by

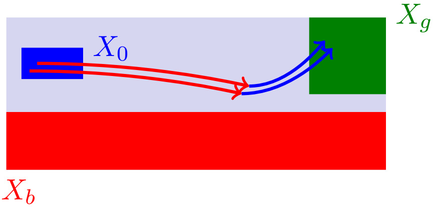
$$\dot{x}(t) = f_{\sigma(t)}\left(x(t), \delta(t)\right), \quad \delta(t) \in D. \tag{2}$$

Problem

Problem

Given an initial set X_0, a goal set X_g, and a bad (unsafe) set X_b, synthesize a switching protocol $\sigma : X \to \mathcal{A}$ such that all trajectories starting in X_0

1. *remain in $X \setminus X_b$;*
2. *reach X_g in finite time and remain there.*



Approach

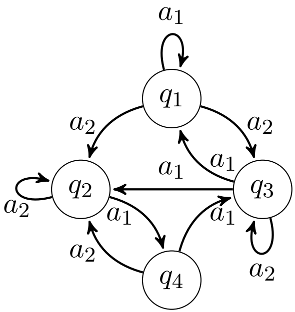
- Abstract the switched system using *augmented* finite transition systems $\mathcal{T}^t$.
- Solve the problem on the discrete state space of $\mathcal{T}^t$.
- If no solution could be found, refine the abstraction to $\mathcal{T}^{t+1}$.
- When solution found, implement discrete controller as a switching protocol.

Augmented finite transition systems

Finite transition system (FTS):

$$\mathcal{T} = (Q, \mathcal{A}, \rightarrow_{\mathcal{T}})$$

- $Q = \{q_1, \ldots, q_N\}$: state space
- $\mathcal{A} = \{a_1, \ldots, a_s\}$: actions
- $\rightarrow_{\mathcal{T}} \subset Q \times A \times Q$: transitions

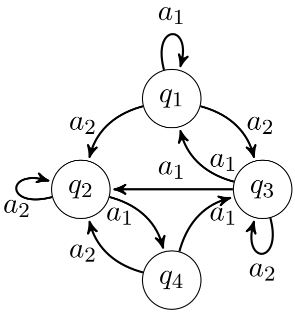


Augmented finite transition systems

Finite transition system (FTS):

$$\mathcal{T} = (Q, \mathcal{A}, \rightarrow_{\mathcal{T}})$$

- $Q = \{q_1, \ldots, q_N\}$: state space
- $\mathcal{A} = \{a_1, \ldots, a_s\}$: actions
- $\rightarrow_{\mathcal{T}} \subset Q \times A \times Q$: transitions



Augmented FTS:

$$\mathcal{T} = (Q, \mathcal{A}, \rightarrow_{\mathcal{T}}, \mathcal{G})$$

- $\mathcal{G} : \mathcal{A} \rightarrow 2^{2^Q}$: progress group map.

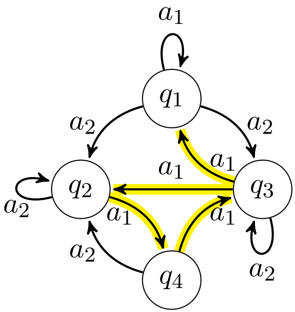
For $G \in \mathcal{G}(a)$, the system can not remain indefinitely in the progress group G by using the action a only.

Augmented finite transition systems

Finite transition system (FTS):

$$\mathcal{T} = (Q, \mathcal{A}, \rightarrow_{\mathcal{T}})$$

- $Q = \{q_1, \ldots, q_N\}$: state space
- $\mathcal{A} = \{a_1, \ldots, a_s\}$: actions
- $\rightarrow_{\mathcal{T}} \subset Q \times A \times Q$: transitions



Augmented FTS:

$$\mathcal{T} = (Q, \mathcal{A}, \rightarrow_{\mathcal{T}}, \mathcal{G})$$

- $\mathcal{G} : \mathcal{A} \rightarrow 2^{2^Q}$: progress group map.

For $G \in \mathcal{G}(a)$, the system can not remain indefinitely in the progress group G by using the action a only.

Augmented finite transition systems

Finite transition system (FTS):

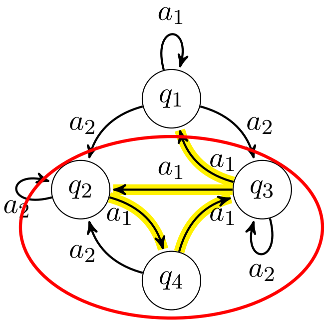
$$\mathcal{T} = (Q, \mathcal{A}, \rightarrow_{\mathcal{T}})$$

- $Q = \{q_1, \ldots, q_N\}$: state space
- $\mathcal{A} = \{a_1, \ldots, a_s\}$: actions
- $\rightarrow_{\mathcal{T}} \subset Q \times A \times Q$: transitions

Augmented FTS:

$$\mathcal{T} = (Q, \mathcal{A}, \rightarrow_{\mathcal{T}}, \mathcal{G})$$

- $\mathcal{G} : \mathcal{A} \rightarrow 2^{2^Q}$: progress group map.



$\in \mathcal{G}(a_1) \implies$

$(\Box a_1 \implies \Diamond q_1)$

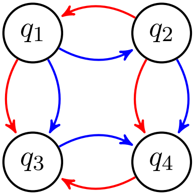
For $G \in \mathcal{G}(a)$, the system can not remain indefinitely in the progress group G by using the action a only.

Simulation relation

Definition

An augmented finite transition system $\mathcal{T} = (Q, \mathcal{A}, \to_{\mathcal{T}}, \mathcal{G})$ *over-approximates* a switched system $\mathcal{S} = (X, \mathcal{A}, \{f_a\}_{a \in \mathcal{A}}, D)$ (denoted $\mathcal{T} \underset{\text{O.A.}}{\succeq} \mathcal{S}$) if there exists a mapping $\alpha : X \to Q$ s.t.

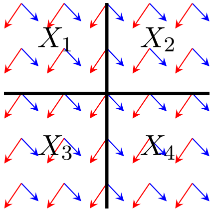
- $\mathcal{T}$ captures all transitions in $\mathcal{S}$.
- For each $G \in \mathcal{G}(a)$, $\alpha^{-1}(G)$ is transient[1] in $\mathcal{S}$ under mode a.



$\mathcal{T}$

$$X_i \ni x \mapsto q_i$$

$$\underset{\text{O.A.}}{\succeq}$$



$\mathcal{S}$

[1]transient: finite exit time

Outline

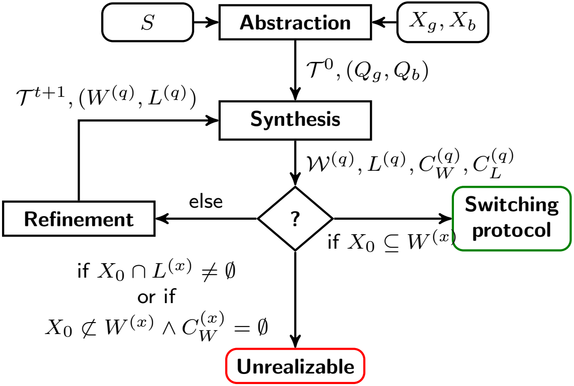
Introduction

Abstraction-refinement loop

Examples

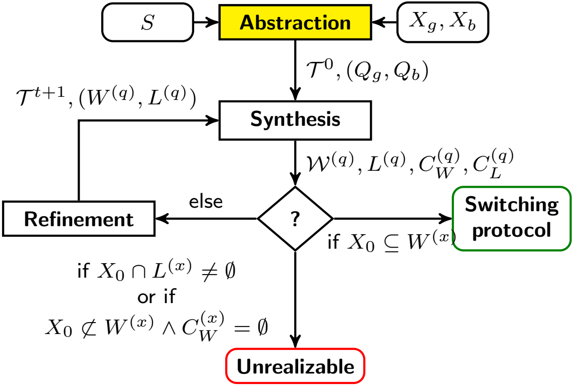
Overview

Idea: incrementally refine the partition where there is potential to expand the winning or losing set.



Overview

Idea: incrementally refine the partition where there is potential to expand the winning or losing set.



Abstraction I

Purpose: Construct an augmented finite transition system that simulates the switched system.
Method: Partition continuous state space, record transitions for each switched mode.
Two things are required:

Abstraction I

Purpose: Construct an augmented finite transition system that simulates the switched system.
Method: Partition continuous state space, record transitions for each switched mode.
Two things are required:

- Way to represent sets.
    1. Hyperboxes
    2. Polyhedra
    3. Semi-algebraic sets

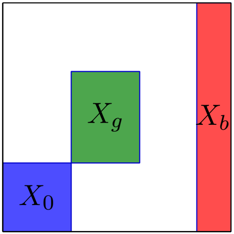
Abstraction I

Purpose: Construct an augmented finite transition system that simulates the switched system.
Method: Partition continuous state space, record transitions for each switched mode.
Two things are required:

- Way to represent sets.
  1. Hyperboxes
  2. Polyhedra
  3. Semi-algebraic sets
- A way (given dynamics and set representation) to determine if there is a trajectory between adjacent regions
  1. Linear vector fields + hyper boxes: Corner check
  2. Linear vector fields + polyhedra: Linear programming
  3. Polynomial vector fields + semi-algebraic sets: Positive polynomial optimization

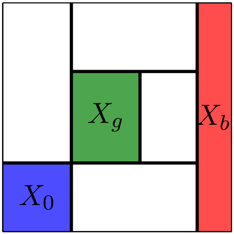
Abstraction II

1. Given X_0, X_g, X_b, create canonical partition of X. Assign a discrete state to each cell.



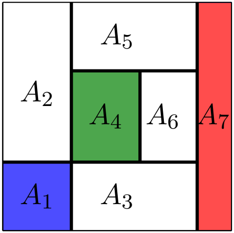
Abstraction II

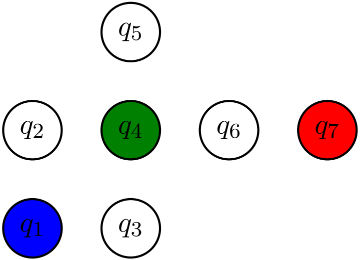
1. Given X_0, X_g, X_b, create canonical partition of X. Assign a discrete state to each cell.



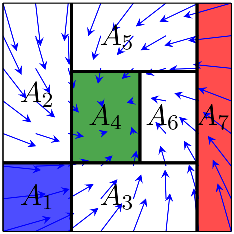
Abstraction II

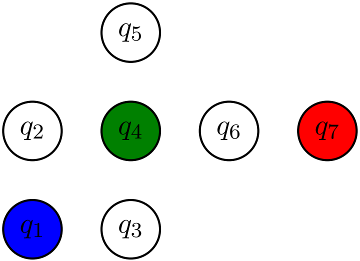
1. Given X_0, X_g, X_b, create canonical partition of X. Assign a discrete state to each cell.





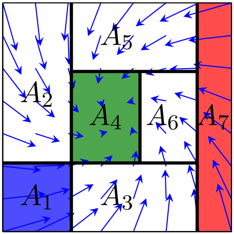
1. Given X_0, X_g, X_b, create canonical partition of X. Assign a discrete state to each cell.
2. For each mode, determine transitions between neighboring cells.

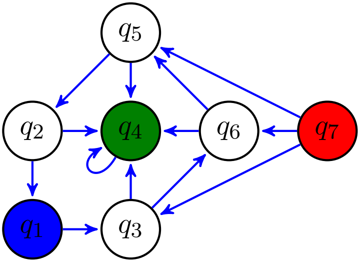




Abstraction II

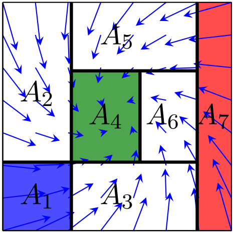
1. Given X_0, X_g, X_b, create canonical partition of X. Assign a discrete state to each cell.
2. For each mode, determine transitions between neighboring cells.

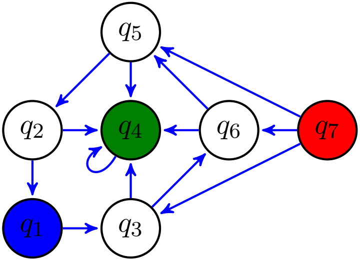




Abstraction II

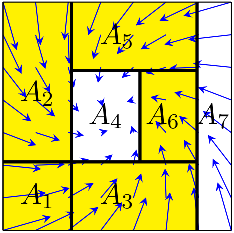
1. Given X_0, X_g, X_b, create canonical partition of X. Assign a discrete state to each cell.
2. For each mode, determine transitions between neighboring cells.
3. Add progress groups to the progress group map.

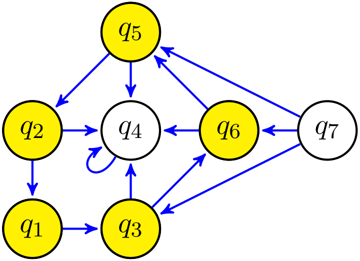




Abstraction II

1. Given X_0, X_g, X_b, create canonical partition of X. Assign a discrete state to each cell.
2. For each mode, determine transitions between neighboring cells.
3. Add progress groups to the progress group map.

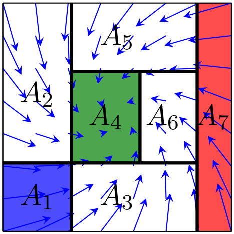




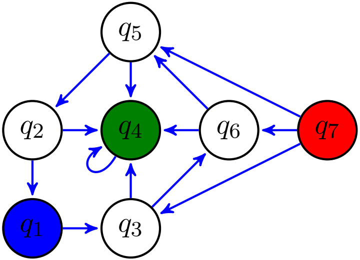
$$\{q_1, q_3, q_6, q_5, q_2\} \in \mathcal{G}(a_1)$$

Abstraction II

1. Given X_0, X_g, X_b, create canonical partition of X. Assign a discrete state to each cell.
2. For each mode, determine transitions between neighboring cells.
3. Add progress groups to the progress group map.

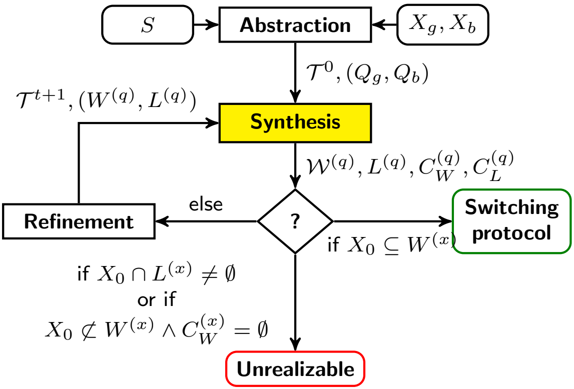


$\preceq_{\text{O.A.}}$



$\{q_1, q_3, q_6, q_5, q_2\} \in \mathcal{G}(a_1)$

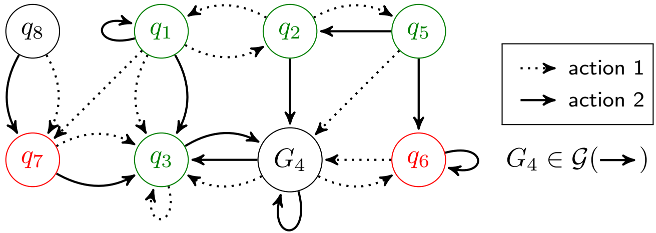
Overview



S
Abstraction
X_g, X_b
$\mathcal{T}^0, (Q_g, Q_b)$
$\mathcal{T}^{t+1}, (W^{(q)}, L^{(q)})$
Synthesis
$\mathcal{W}^{(q)}, L^{(q)}, C_W^{(q)}, C_L^{(q)}$
Refinement
else
?
Switching protocol
if $X_0 \subseteq W^{(x)}$
if $X_0 \cap L^{(x)} \neq \emptyset$
or if
$X_0 \not\subset W^{(x)} \wedge C_W^{(x)} = \emptyset$
Unrealizable

Synthesis: Solve reach-stay-avoid game on augmented FTS

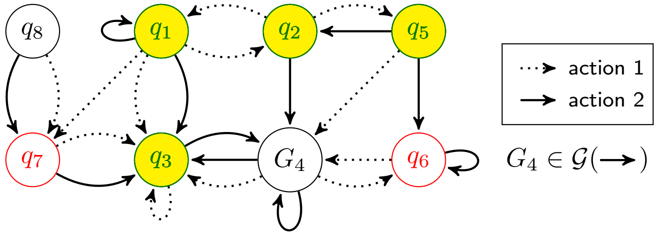
Given: Goal set $Q_g = \{q_1, q_2, q_3, q_5\}$, bad set $Q_b = \{q_6, q_7\}$.



Synthesis: Solve reach-stay-avoid game on augmented FTS

Given: Goal set $Q_g = \{q_1, q_2, q_3, q_5\}$, bad set $Q_b = \{q_6, q_7\}$.

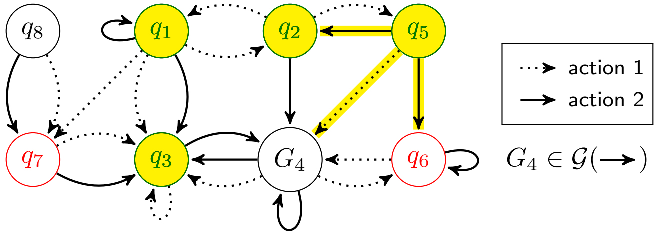
1. Find a *controlled-invariant* set $C \subset Q_g$ by iteratively removing states in Q_g.



Synthesis: Solve reach-stay-avoid game on augmented FTS

Given: Goal set $Q_g = \{q_1, q_2, q_3, q_5\}$, bad set $Q_b = \{q_6, q_7\}$.

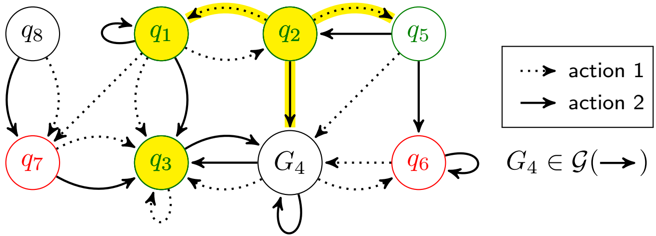
1. Find a *controlled-invariant* set $C \subset Q_g$ by iteratively removing states in Q_g.



Synthesis: Solve reach-stay-avoid game on augmented FTS

Given: Goal set $Q_g = \{q_1, q_2, q_3, q_5\}$, bad set $Q_b = \{q_6, q_7\}$.

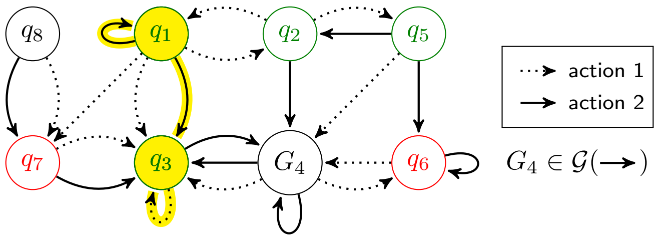
1. Find a *controlled-invariant* set $C \subset Q_g$ by iteratively removing states in Q_g.



Synthesis: Solve reach-stay-avoid game on augmented FTS

Given: Goal set $Q_g = \{q_1, q_2, q_3, q_5\}$, bad set $Q_b = \{q_6, q_7\}$.

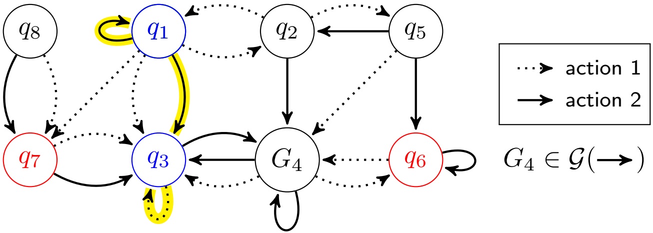
1. Find a *controlled-invariant* set $C \subset Q_g$ by iteratively removing states in Q_g.



Synthesis: Solve reach-stay-avoid game on augmented FTS

Given: Goal set $Q_g = \{q_1, q_2, q_3, q_5\}$, bad set $Q_b = \{q_6, q_7\}$.

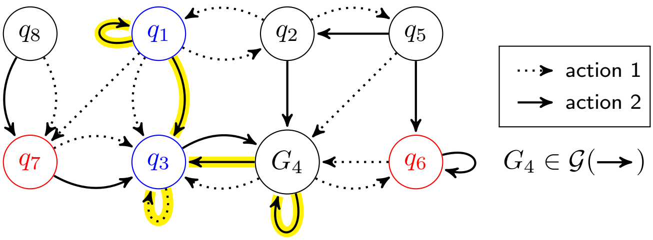
1. Find a *controlled-invariant* set $C = \{q_1, q_3\} \subset Q_g$ by iteratively removing states in Q_g.



Synthesis: Solve reach-stay-avoid game on augmented FTS

Given: Goal set $Q_g = \{q_1, q_2, q_3, q_5\}$, bad set $Q_b = \{q_6, q_7\}$.

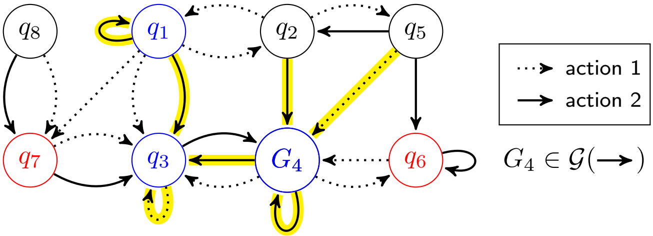
1. Find a *controlled-invariant* set $C = \{q_1, q_3\} \subset Q_g$ by iteratively removing states in Q_g.
2. Compute the *controllable predecessor set* [Sun et al. (2013)] of C (takes advantage of progress groups). Gives *winning set* $W^{(q)} = C$.



Synthesis: Solve reach-stay-avoid game on augmented FTS

Given: Goal set $Q_g = \{q_1, q_2, q_3, q_5\}$, bad set $Q_b = \{q_6, q_7\}$.

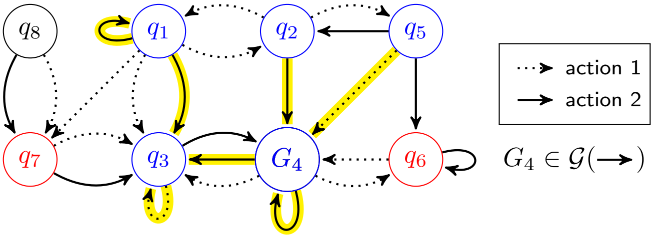
1. Find a *controlled-invariant* set $C = \{q_1, q_3\} \subset Q_g$ by iteratively removing states in Q_g.
2. Compute the *controllable predecessor set* [Sun et al. (2013)] of C (takes advantage of progress groups). Gives *winning set* $W^{(q)} = C \cup G_4$.



Synthesis: Solve reach-stay-avoid game on augmented FTS

Given: Goal set $Q_g = \{q_1, q_2, q_3, q_5\}$, bad set $Q_b = \{q_6, q_7\}$.

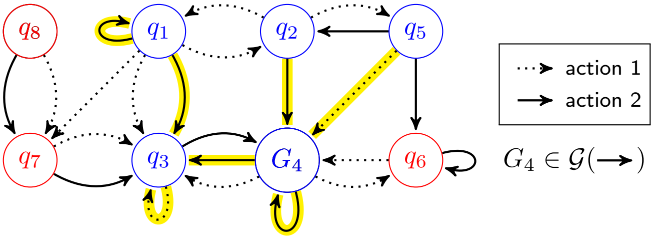
1. Find a *controlled-invariant* set $C = \{q_1, q_3\} \subset Q_g$ by iteratively removing states in Q_g.
2. Compute the *controllable predecessor set* [Sun et al. (2013)] of C (takes advantage of progress groups). Gives *winning set* $W^{(q)} = C \cup G_4 \cup \{q_2, q_5\}$.



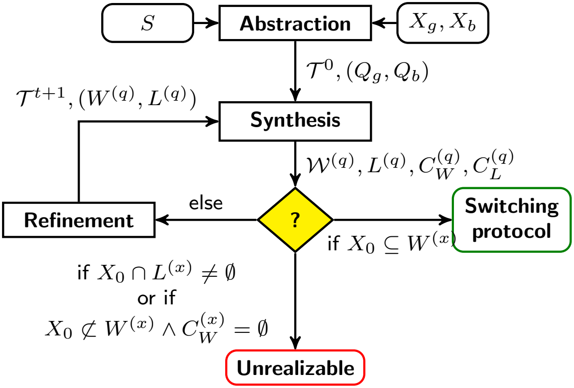
Synthesis: Solve reach-stay-avoid game on augmented FTS

Given: Goal set $Q_g = \{q_1, q_2, q_3, q_5\}$, bad set $Q_b = \{q_6, q_7\}$.

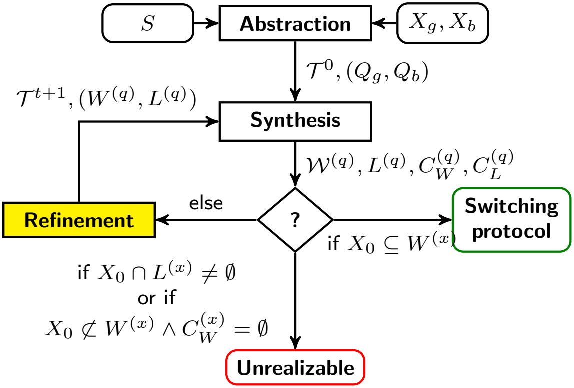
1. Find a *controlled-invariant* set $C = \{q_1, q_3\} \subset Q_g$ by iteratively removing states in Q_g.
2. Compute the *controllable predecessor set* [Sun et al. (2013)] of C (takes advantage of progress groups). Gives *winning set* $W^{(q)} = C \cup G_4 \cup \{q_2, q_5\}$.
3. Also extract *losing set* $Ł^{(q)} = Q_b \cup \{q_8\}$, states from where there is no chance of avoiding Q_b.



Overview

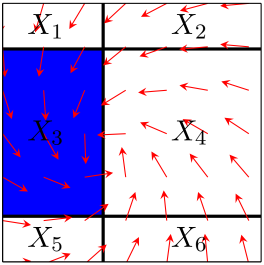


- Initial set in winning set $\rightarrow$ done.
- Losing states in initial set $\rightarrow$ unrealizable.
- Refinement meaningless $\rightarrow$ unrealizable.

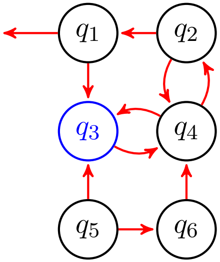


- Initial set in winning set $\rightarrow$ done.
- Losing states in initial set $\rightarrow$ unrealizable.
- Refinement meaningless $\rightarrow$ unrealizable.
- Refine in *potential* winning and losing sets $\alpha^{-1}(C_W^{(q)})$ and $\alpha^{-1}(C_L^{(q)})$.

Refinement I

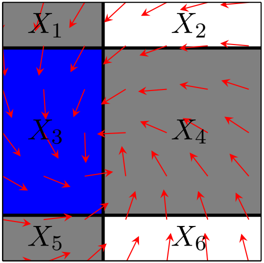


$$\underset{\text{O.A.}}{\preceq}$$

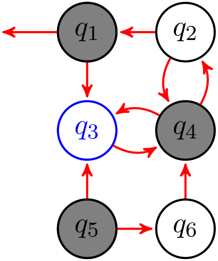


- Winning set: $W^{(q)} = \{q_3\}$.

Refinement I

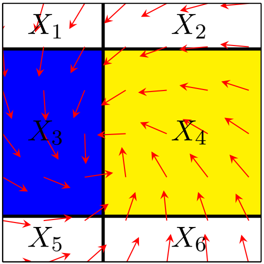


$\preceq_{\text{O.A.}}$

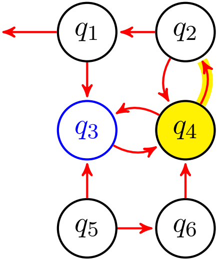


- Winning set: $W^{(q)} = \{q_3\}$.
- Potential winning set $C_W^{(q)} = \{q_1, q_4, q_5\}$: possibility (but no guarantee) to reach $W^{(q)}$.

Refinement I

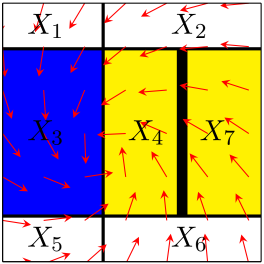


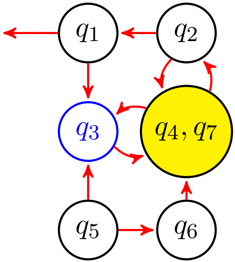
$\preceq_{\text{O.A.}}$



- Winning set: $W^{(q)} = \{q_3\}$.
- Potential winning set $C_W^{(q)} = \{q_1, q_4, q_5\}$: possibility (but no guarantee) to reach $W^{(q)}$.
- Select a cell in potential winning set $C_W^{(x)} = \alpha^{-1}\left(C_W^{(q)}\right)$.

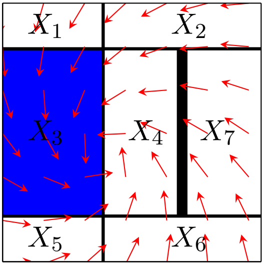
Refinement II



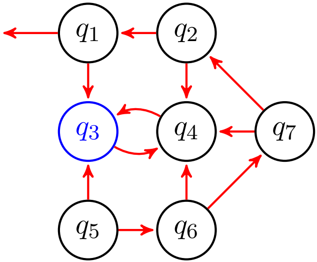


- Split cell, update transitions and progress group map.

Refinement II



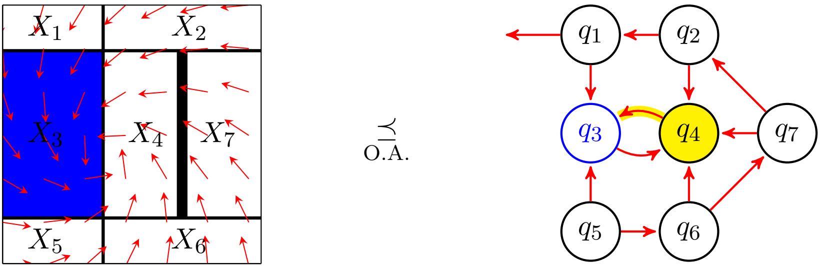
$\underset{\text{O.A.}}{\preceq}$



- Split cell, update transitions and progress group map.
- Results in new abstraction $\mathcal{T}^{t+1}$ s.t

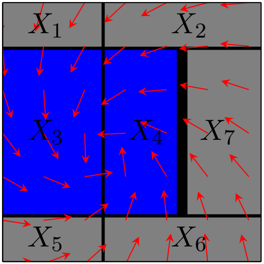
$$S \underset{\text{O.A.}}{\preceq} \mathcal{T}^{t+1} \preceq \mathcal{T}^{t} \tag{3}$$

Refinement III: Next synthesis iteration

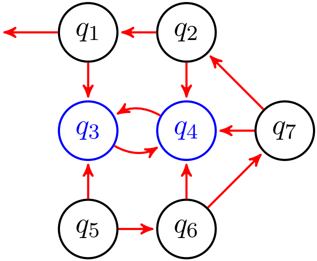


- Expand sets from previous synthesis (complete re-synthesis not necessary).
- Winning set in $\mathcal{T}^{t+1}$ can be expanded due to refinement.

Refinement III: Next synthesis iteration

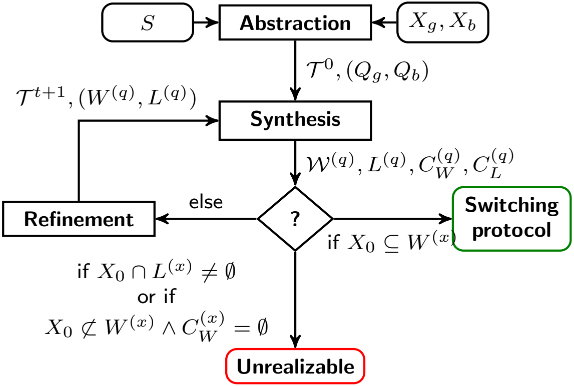


$\underset{\text{O.A.}}{\preceq}$



- Expand sets from previous synthesis (complete re-synthesis not necessary).
- Winning set in $\mathcal{T}^{t+1}$ can be expanded due to refinement.
- New winning set: $W^{(q)} = \{q_3, q_4\}$, new potential winning set $C_W^{(q)} = \{q_1, q_2, q_5, q_6, q_7\}$.

Overview

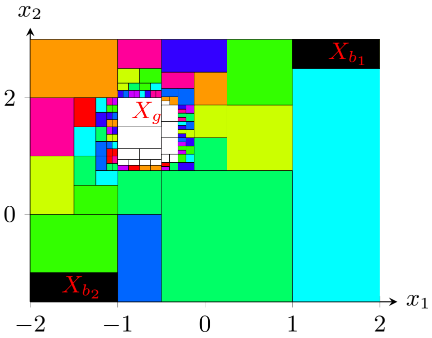


- Iterate until switching protocol or certificate of unrealizability obtained, or maximal number of iterations reached.

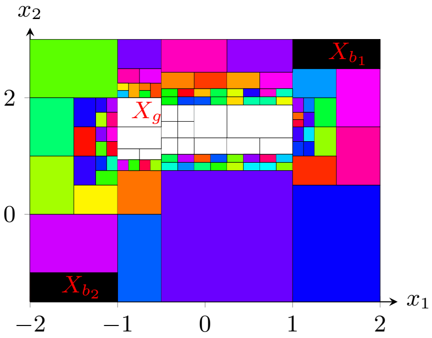
Outline

Example 1: Augmented vs traditional transition system

Polynomial dynamics, 3 modes.



With finite transition systems.



With augmented finite transition systems.

Winning sets in white.

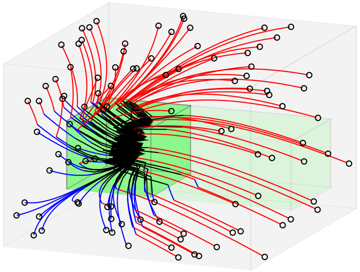
Example 2: Hydronic radiant system for buildings

Hot or chilled supply water is pumped through tubes in order to adjust the temperature of a room.

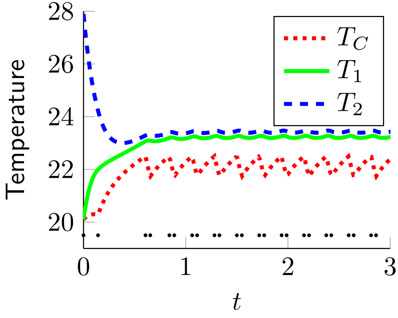
$$
\begin{aligned}
C_r\dot{T}_c &= \sum_{i=1}^{2} K_{r,i}(T_i - T_c) + K(T_w - T_c),\\
C_i\dot{T}_i &= K_{r,i}(T_c - T_i) + K_i(T_a - T_i) + \sum_{j\neq i} K_{ij}(T_j - T_i) + q_i,
\end{aligned}
$$

- Two modes: $K = K_w$ and $K = 0$.
- Three states: T_c, T_1, T_2.
- Goal: steer to goal set $T_c \in [21, 27]$ and $T_{1,2} \in [22, 25]$.
- Fixed points of modes are outside goal set.

Example 2: Result



Winning set consists of 705 discrete states



Simulation

Summary

- Proposed a method for switching protocol synthesis based on incremental refinement.
- Augmented finite transition systems enables encoding of additional properties of the underlying switched systems.
- Possibility to obtain certificates of unrealizability (losing set intersects initial set).

Future work:

- Parallel implementations.
- Explore trade offs between set representations.
- Identify problem classes with termination guarantees.

Thank you for your attention.

References

F. Sun, N. Ozay. E. M. Wolff, J. Liu and R. M. Murray.
Efficient Control Synthesis for Augmented Finite Transition Systems with an Application to Switching Protocols.
In *Proc. of the ACM/IEEE International Conference on Cyber-Physical Systems*, 2013.