5 Formal Controller Synthesis

5.1 Abstraction-based Controller Synthesis

5.2 Control Barrier Functions

5.3 Funnel-based Control

5.1 Abstraction-based Controller Synthesis

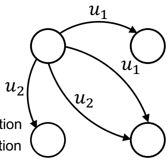
	Continuous	Discrete
Uncontrolled	$\dot{\xi} = f(\xi, \upsilon)$, $\xi(t) \in X \subseteq \mathbb{R}^n$, $\upsilon(t) \in U \subseteq \mathbb{R}^m$	
Controlled		

5.1 Abstraction-based Controller Synthesis

	Continuous	Discrete
Uncontrolled	$\dot{\xi} = f(\xi, \upsilon)$, $\xi(t) \in X \subseteq \mathbb{R}^n$, $\upsilon(t) \in U \subseteq \mathbb{R}^m$	
Controlled		

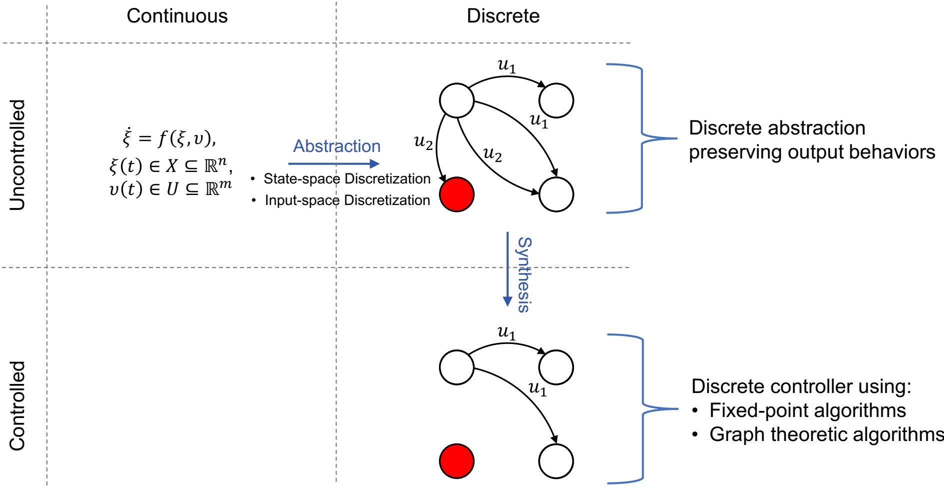
Abstraction

- State-space Discretization
- Input-space Discretization

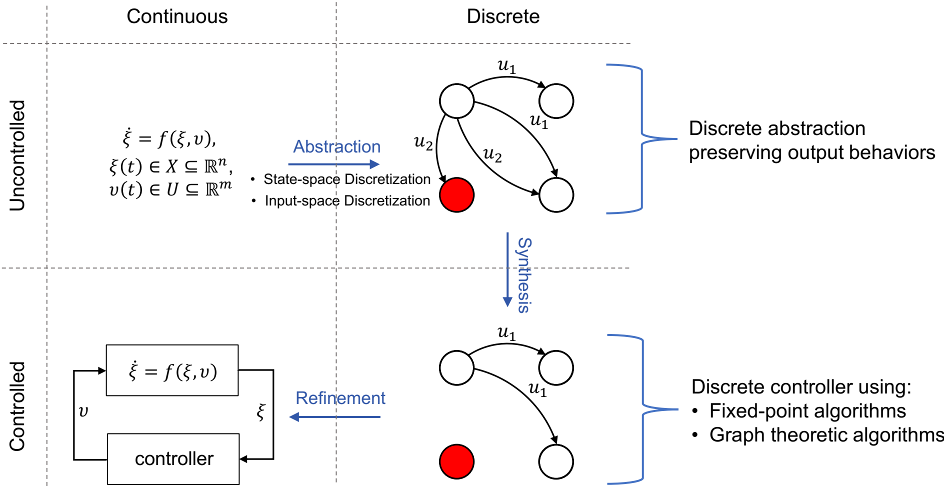


Discrete abstraction preserving output behaviors

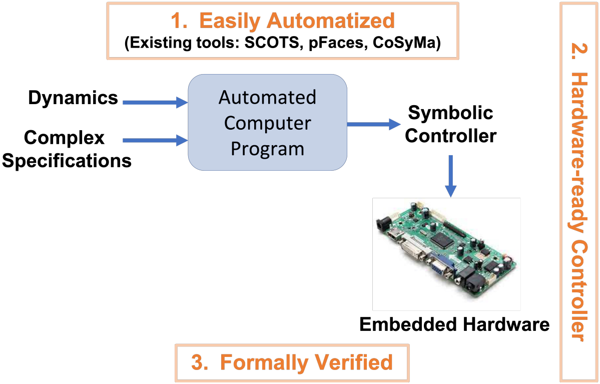
5.1 Abstraction-based Controller Synthesis



5.1 Abstraction-based Controller Synthesis



Advantages of Symbolic Control



- Efficiently deals with state and input constraints
- No restriction on dynamics (only it should be defined over Euclidean space)
- Gives maximally permissible controllers

Definition: Admissible Inputs

Given a system $S = (X, X_0, U, F)$, we define the set of admissible inputs at state $x \in X$ by

$$U_S(x) = \{u \in U \mid F(x, u) \neq \varnothing\}$$

Definition: Feedback Refinement Relations

Let $S_i = (X_i, X_{i,0}, U_i, F_i)$, $i \in \{1, 2\}$ be two simple systems and assume $U_2 \subseteq U_1$.
A strict relation

$$Q \subseteq X_1 \times X_2$$

is a feedback refinement relation from S_1 to S_2, denoted by

$$S_1 \preccurlyeq_Q S_2$$

if the following holds for all $(x_1, x_2) \in Q$:

1. $x_1 \in X_{1,0}$ implies $x_2 \in X_{2,0}$
2. $U_{S_2}(x_2) \subseteq U_{S_1}(x_1)$
3. $u \in U_{S_2}(x_2) \implies Q(F_1(x_1, u)) \subseteq F_2(x_2, u)$

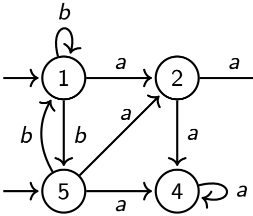
In words for $(x_1, x_2) \in Q$

1. if x_1 is an initial state then x_2 needs to be an initial state
2. every admissible input of S_2 at x_2 is an admissible input of S_1 at x_1
3. every successor $x_1' \in F_1(x_1, u)$ when mapped to X_2 via Q is contained in $F_2(x_2, u)$

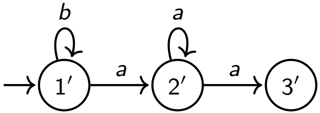
Example: Feedback Refinement Relation

Consider the two simple systems

S_1 :



S_2 :



and the relation $Q = \{(1, 1'), (5, 1'), (2, 2'), (3, 3'), (4, 2')\}$.

Let us verify that $S_1 \preccurlyeq_Q S_2$:

- $\forall_{x_1 \in X_1} : \quad Q(x_1) \neq \varnothing \implies \quad Q$ is strict ✓
- $\forall_{(x_1, x_2) \in Q \wedge x_1 \in X_{1,0}} : \; x_2 \in X_{2,0}$ ✓
- $\forall_{(x_1, x_2) \in Q} : U_{S_2}(x_2) \subseteq U_{S_1}(x_1)$:
  - $U_{S_2}(1') = \{a, b\}, \; U_{S_2}(2') = \{a\}, \; U_{S_2}(3') = \varnothing$
  - $U_{S_1}(1) = U_{S_1}(5) = \{a, b\}, \; U_{S_1}(2) = U_{S_1}(4) = \{a\}, \; U_{S_1}(3) = \varnothing$

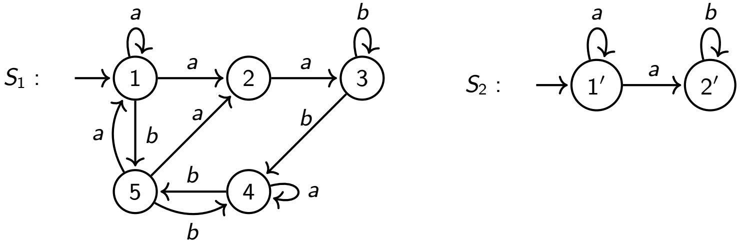
  $U_{S_2}(1') \subseteq U_{S_1}(1), U_{S_2}(1') \subseteq U_{S_1}(5), U_{S_2}(2') \subseteq U_{S_1}(2), U_{S_2}(2') \subseteq U_{S_1}(4), U_{S_2}(3') \subseteq U_{S_1}(3)$✓
- $\forall_{(x_1, x_2) \in Q \wedge u \in U_{S_2}(x_2)} : Q(F_1(x_1, u)) \subseteq F_2(x_2, u)$: ✓

  $Q(F_1(1, a)) = \{2'\} \subseteq F_2(1', a) \quad Q(F_1(1, b)) = \{1'\} \subseteq F_2(1', b) \quad Q(F_1(2, a)) = \{2', 3'\} \subseteq F_2(2', a)$

  $Q(F_1(5, a)) = \{2'\} \subseteq F_2(1', a) \quad Q(F_1(5, b)) = \{1'\} \subseteq F_2(1', b) \quad Q(F_1(4, a)) = \{2'\} \subseteq F_2(2', a)$

Question: Feedback Refinement Relation

Consider the simple systems



Verify that

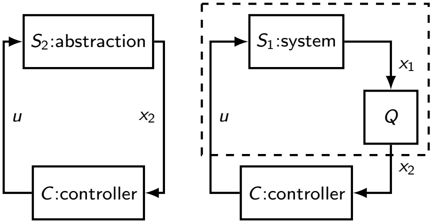
$$Q = \{(1, 1'), (2, 1'), (3, 2'), (4, 2'), (5, 2')\}$$

satisfies $S_1 \preccurlyeq_Q S_2$.

Controller Refinement with Behavioral Inclusion

Consider

- $S_i = (X_i, X_{i,0}, U_i, F_i)$, $i \in \{1, 2\}$, $U_2 \subseteq U_1$
- $C = (X_c, X_{c,0}, U_c, V_c, Y_c, F_c, H_c)$
- strict $Q \subseteq X_1 \times X_2$



and the statements

1. C is feedback composable with S_2
2. C is feedback composable with $Q \circ S_1$
3. $\mathcal{B}(C \times (Q \circ S_1)) \subseteq \mathcal{B}(C \times S_2)$

and the implication

$$(y_c, v_c) \in H_c(x_c, x_2) \wedge F_2(x_2, y_c) = \varnothing \implies F_c(x_c, v_c) = \varnothing \qquad (*)$$

- $(*)$: C non-blocking $\implies$ S_2 non-blocking

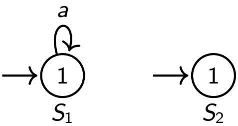
Theorem: FRR and Behavioral Inclusion

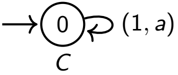
Let $S_1 \preccurlyeq_Q S_2$, then 1. and $(*)$ imply 2. and 3. ■

Example: On the Non-Blocking Condition $(*)$

Consider

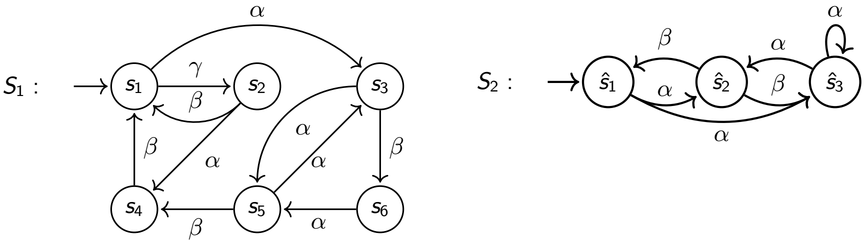
- $S_1 = (\{1\}, \{1\}, \{a\}, F_1)$ with $F_1(1, a) = \{1\}$
- $S_2 = (\{1\}, \{1\}, \{a\}, F_2)$ with $F_2(1, a) = \varnothing$
- $C = (\{0\}, \{0\}, \{1\}, \{1\}, \{a\}, F_q, H_q)$ with $F_q(0, 1) = \{0\}$ and $H_q(0, 1) = \{a\} \times \{1\}$.
- $Q = \{(1, 1)\}$ is FRR from S_1 to S_2
- C is feedb. comp. with S_2
- C is feedb. comp. with S_1
- $(*)$ is not satisfied
- $(a, 1)^\omega \in \mathcal{B}(C \times S_1)$ but $(a, 1)^\omega \notin \mathcal{B}(C \times S_2)$





Exercise 8: FRR and Behavioral Inclusion

Consider the simple systems with input alphabet $U = \{\alpha, \beta, \gamma\}$



and the relation

$$Q = \{(s_1, \hat{s}_1), (s_2, \hat{s}_1), (s_3, \hat{s}_2), (s_4, \hat{s}_2), (s_3, \hat{s}_3), (s_5, \hat{s}_3), (s_6, \hat{s}_3)\}$$

which satisfies $S_1 \preccurlyeq_Q S_2$.
Consider $C = (\{q\}, \{q\}, X_2, X_2, U_2, F_c, H_c)$ with F_c being strict and

$$H_c(q, \hat{s}_1) = \{\alpha\} \times \{\hat{s}_1\}, H_c(q, \hat{s}_2) = \{\beta\} \times \{\hat{s}_2\}, H_c(q, \hat{s}_3) = \{\alpha\} \times \{\hat{s}_3\}.$$

1. Does C and S_2 satisfy the non-blocking condition $(*)$?
2. Is C feedb. composable with $Q \circ S_1$?
3. Verify $\mathcal{B}(C \times (Q \circ S_1)) \subseteq \mathcal{B}(C \times S_2)$.

5.2 Computation of Abstractions

Let

- $S_i = (X_i, X_{i,0}, U_i, F_i)$ be two systems $i \in \{1, 2\}$
- X_2 be a cover by non-empty sets of X_1

Theorem: Computation of Abstractions

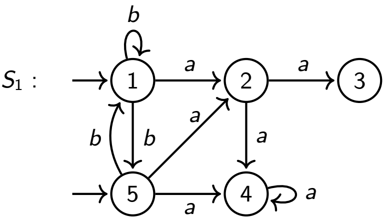
$S_1 \preccurlyeq_{\in} S_2$ if and only if

1. $x_1 \in X_{1,0}$ and $x_1 \in x_2$ implies $x_2 \in X_{2,0}$
2. $U_2 \subseteq U_1$ and $x_1 \in x_2 \in X_2$ implies $U_{S_2}(x_2) \subseteq U_{S_1}(x_1)$
3. $x_2, x_2' \in X_2$, $u \in U_{S_2}(x_2)$ and $x_2' \cap F_1(x_2, u) \neq \varnothing$ implies $x_2' \in F_2(x_2, u)$ ■

Computation of abstraction reduces to computation of reachable sets!

Example: Feedback Refinement Relation

Consider the simple systems



and the sets

$$1' = \{1, 5\},\ 2' = \{2, 4\},\ 3' = \{3\}$$

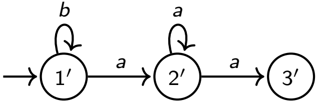
Let us determine $X_{2,0}$, F_2 so that

$$(x_1, x_2) \in Q \iff x_1 \in x_2$$

is a FRR from S_1 to $S_2 = (X_2, X_{2,0}, \{a, b\}, F_2)$.

We look at Theorem: Computation of Abstractions:

- from 1. we get $X_{2,0} = \{1'\}$
- let us determine $F_2(1', a)$
  - we check 2.:
    $1 \in 1'$ and $a \in U_{S_1}(1)$,
    $5 \in 1'$ and $a \in U_{S_1}(5)$,
    hence $a \in U_{S_2}(1')$ is fine
  - we check 3.:
    $F_1(1', a) = \{2, 4\}$ hence $2' = F_2(1', a)$
- let us determine $F_2(3', a)$
  - we check 2.:
    $3 \in 3'$ and $a \notin U_{S_1}(3)$, hence $F_2(3', a) = \varnothing$

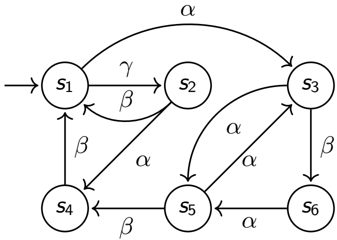


S_2

Question: Construction of an Abstraction

Consider the simple systems

S_1 :



and the set

$$X_2 = \{\underbrace{\{s_1, s_2\}}_{\hat{s}_1}, \underbrace{\{s_3, s_4\}}_{\hat{s}_2}, \underbrace{\{s_3, s_5, s_6\}}_{\hat{s}_3}\}$$

determine $X_{2,0}$, F_2 so that $S_2 = (X_2, X_{2,0}, U, F_2)$ satisfies $S_1 \preccurlyeq_{\in} S_2$.

5.3 Abstractions of Control Systems

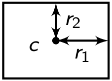
Sample-and-Hold Linear Control System

- continuous dynamics
  - $\dot{\xi}(t) = A\xi(t) + B\nu(t)$
- sample-and-hold behavior $S = (X, X_{1,0}, U, F)$
  - $X = \mathbb{R}^n$
  - $X_0 \subseteq \mathbb{R}^n$
  - $U \subseteq \mathbb{R}^m$
  - $F(x, u) = \{A_d x + B_d u\}$ with
    - $A_d = \mathrm{e}^{A\tau}$
    - $B_d = \int_0^\tau \mathrm{e}^{As} B \mathrm{d}s$

An Abstraction

- $\hat{S} = (\hat{X}, \hat{X}_0, \hat{U}, \hat{F})$
- $\hat{U}$ is some finite subset of U
- $\hat{X} = \hat{X}_b \cup \hat{X}_o$ be a finite cover of $\mathbb{R}^n$
  - every element $\hat{x} \in \hat{X}_b$ is hyper-rectangle with center $c \in \mathbb{R}^n$ and radius $r \in \mathbb{R}^n$

$$\hat{x} = [-r_1 + c_1, c_1 + r_1] \times \cdots \times [-r_n + c_n, c_n + r_n]$$



  - $\hat{X}_o$ contains the "over-flow" symbols, e.g., $\hat{x} \in \hat{X}_o$

$$\hat{x} = \{y \in \mathbb{R}^n \mid y_1 \geq 2\}$$

- $\hat{X}_0 = \{\hat{x} \in \hat{X} \mid \hat{x} \cap X_0 \neq \varnothing\}$

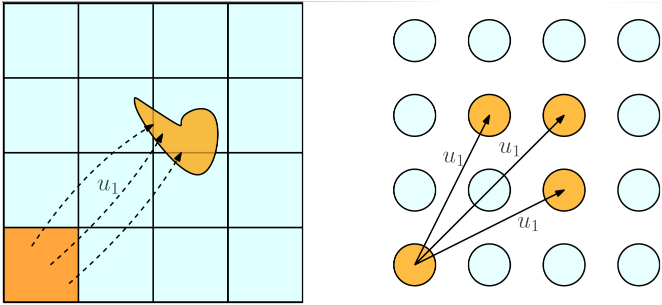
An Abstraction

- we compute $\hat{F}(\hat{x}, u)$ as follows
  - if $\hat{x} \in \hat{X}_o$, then $\hat{F}(\hat{x}, u) = \varnothing$
  - if $\hat{x} = c + [-r, r] \in \hat{X}_b$, compute $c' + [-r', r']$ by:
    $c' = A_d c + B_d u$
    $r' = \mathrm{e}^{\mathrm{metzler}(A)\tau} r$
  - $\hat{x}' \in \hat{F}(\hat{x}, u)$ if and only if $\hat{x}' \cap c' + [-r', r'] \neq \varnothing$

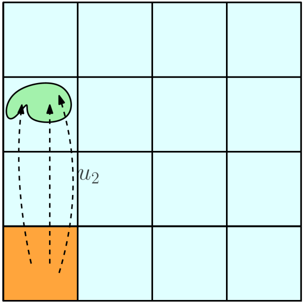
- given $A \in \mathbb{R}^{n \times n}$ the matrix $\mathrm{metzler}(A) \in \mathbb{R}^{n \times n}$ is defined by

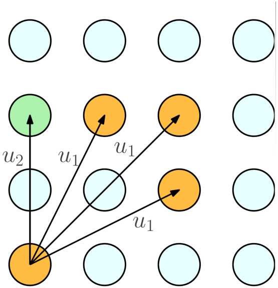
$$\mathrm{metzler}(A)_{ij} = \begin{cases} a_{ij} & \text{if } i = j \\ |a_{ij}| & \text{otherwise} \end{cases}$$

Construction of Abstraction via Reachable set Construction

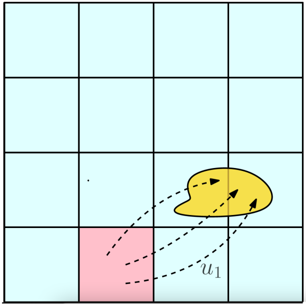


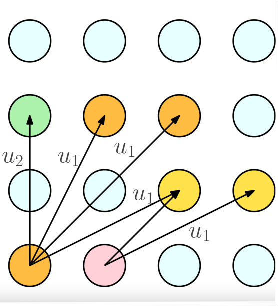
Construction of Abstraction via Reachable set Construction



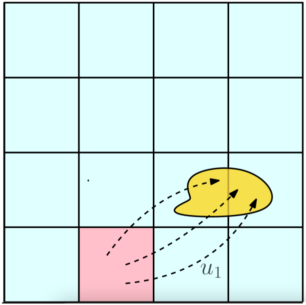


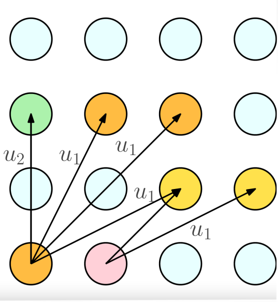
Construction of Abstraction via Reachable set Construction





Construction of Abstraction via Reachable set Construction





Given discrete abstraction we can synthesize a controller for enforcing LTL Specifications via CS algorithm.

Proofs: "Feedback Refinement Relations for the Synthesis of Symbolic Controllers" IEEE TAC 2018