(Towards) Translating Temporal Logic to Controller Specifications

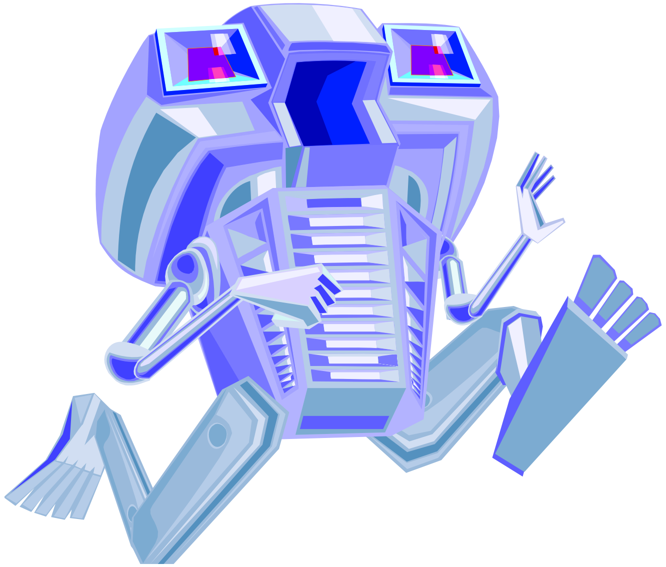
Georgios E. Fainekos, Savvas G. Loizou and George J. Pappas

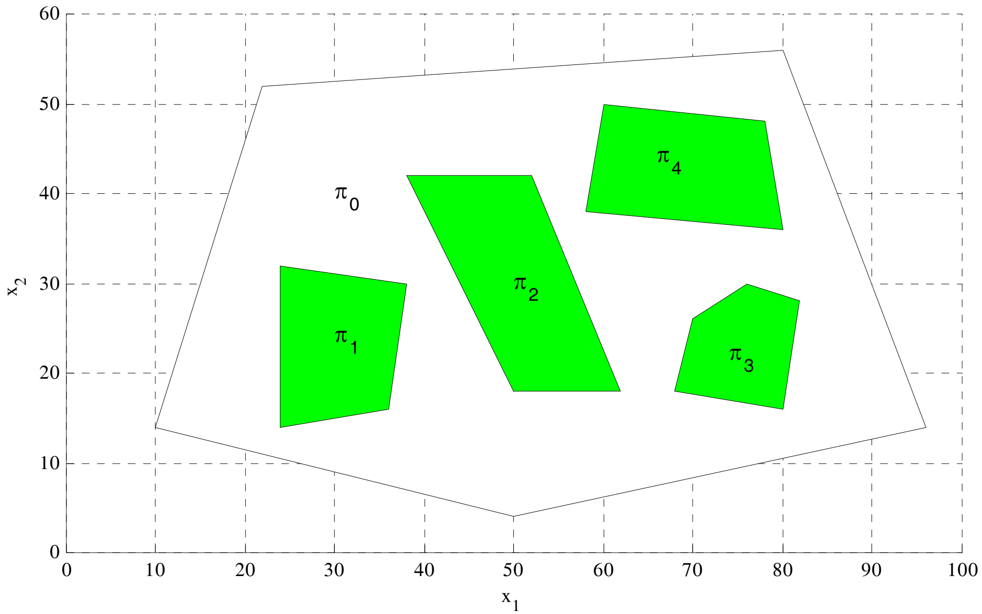
CDC 2006

Math free Presentation!

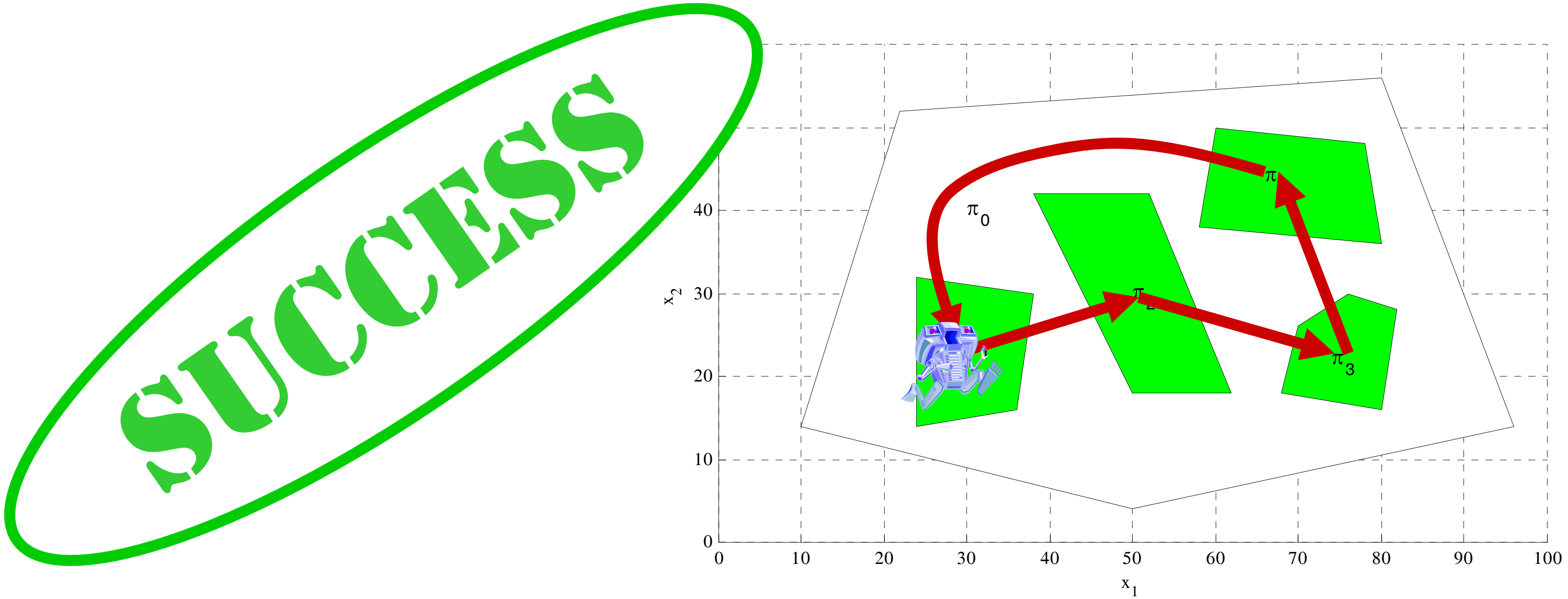
GRASP Lab
Departments of CIS, MEAM and ESE
University of Pennsylvania

Motivation - Motion Planning





Motivation – Motion Planning



Task: "Stay always in π_0 and visit area π_2, then area π_3, then area π_4 and, finally, return to and stay in region π_1 while avoiding areas π_2 and π_3"

Good news

RTL: $\Box\pi_0 \wedge \Diamond(\pi_2 \wedge \Diamond(\pi_3 \wedge \Diamond(\pi_4 \wedge (\neg\pi_2 \wedge \neg\pi_3) U \pi_1)))$

Penn University of Pennsylvania

GRASP

Problem Definition

Temporal Logic Controller Synthesis

Given a dynamical system Σ, a set of initial conditions X_0 and a flat RTL formula φ over Π, construct a closed-loop system in the form of a hybrid automaton H_φ such that the resulting system trajectories $x(t)$ starting at some point $x(0) \in X_0$ satisfy the formula φ.

$$\Sigma: \; \dot{x}(t) = f(x(t), u(t)) \quad x(t) \in X \subseteq \mathbb{R}^n \quad u(t) \in U \subseteq \mathbb{R}^n$$

In the past …

Approaches to synthesis of hybrid systems:

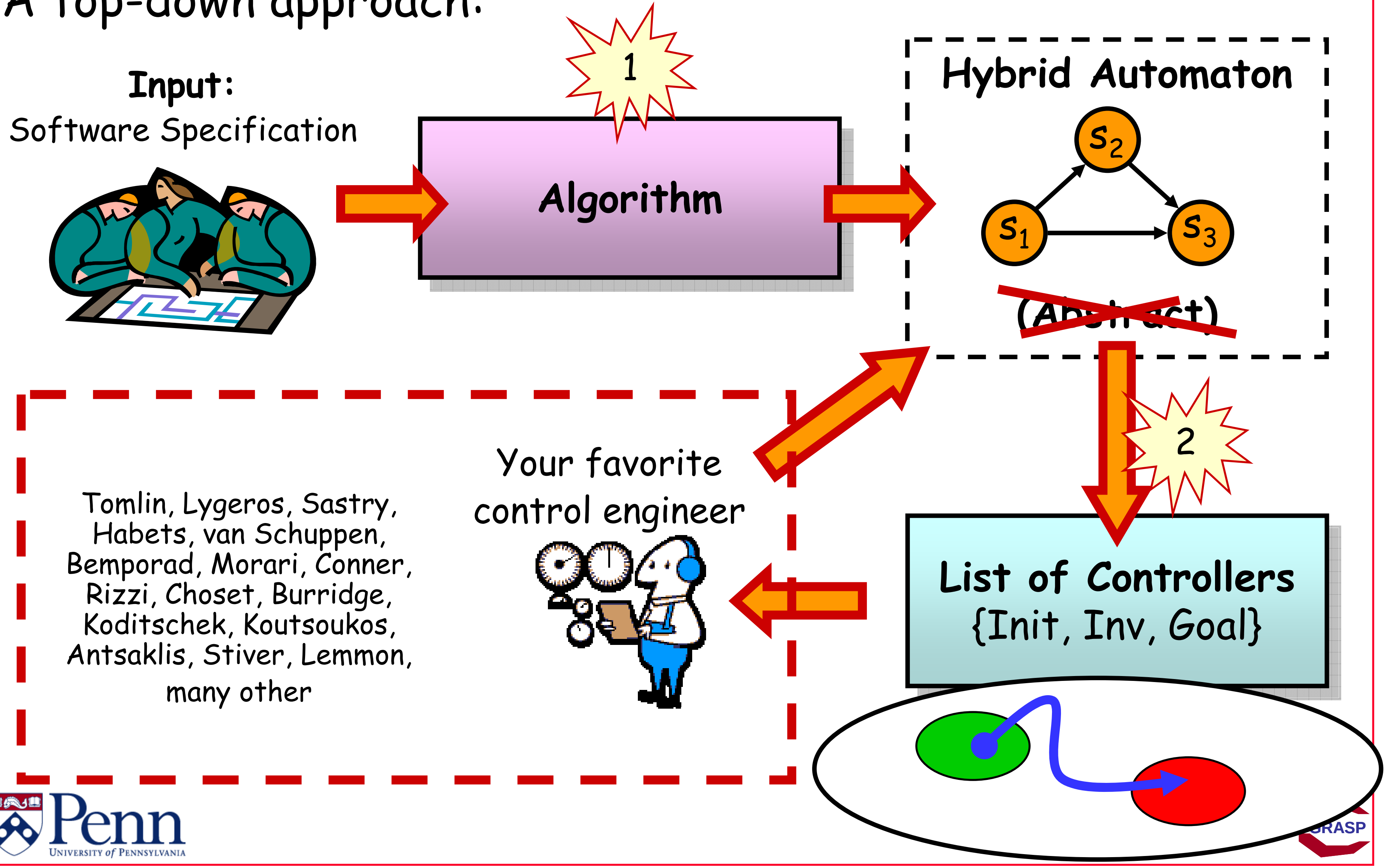
- Given controllers + switching rules $\Longrightarrow$ verify that HS satisfies spec
- Given controllers + spec $\Longrightarrow$ keep necessary controllers
- Given closed-loop dynamics + spec $\Longrightarrow$ design switching rules

Researchers: Alur, Henzinger, Tabuada, Davoren, Moor, Koutsoukos, Antsaklis, Stiver, Lemmon, Tomlin, Lygeros, Sastry, Habets, van Schuppen, Asarin, Bournez, Dang, Maler, Pnueli, Bemporad, Morari, Giorgetti, Lafferriere, Kloetzer, Belta, Kyriakopoulos, Kress-Gazit, Loizou, Pappas, Fainekos and many others …

- **Given spec $\Longrightarrow$ design controllers**

In this presentation …

A top-down approach:



Hybrid Automaton

A hybrid automaton is a tuple

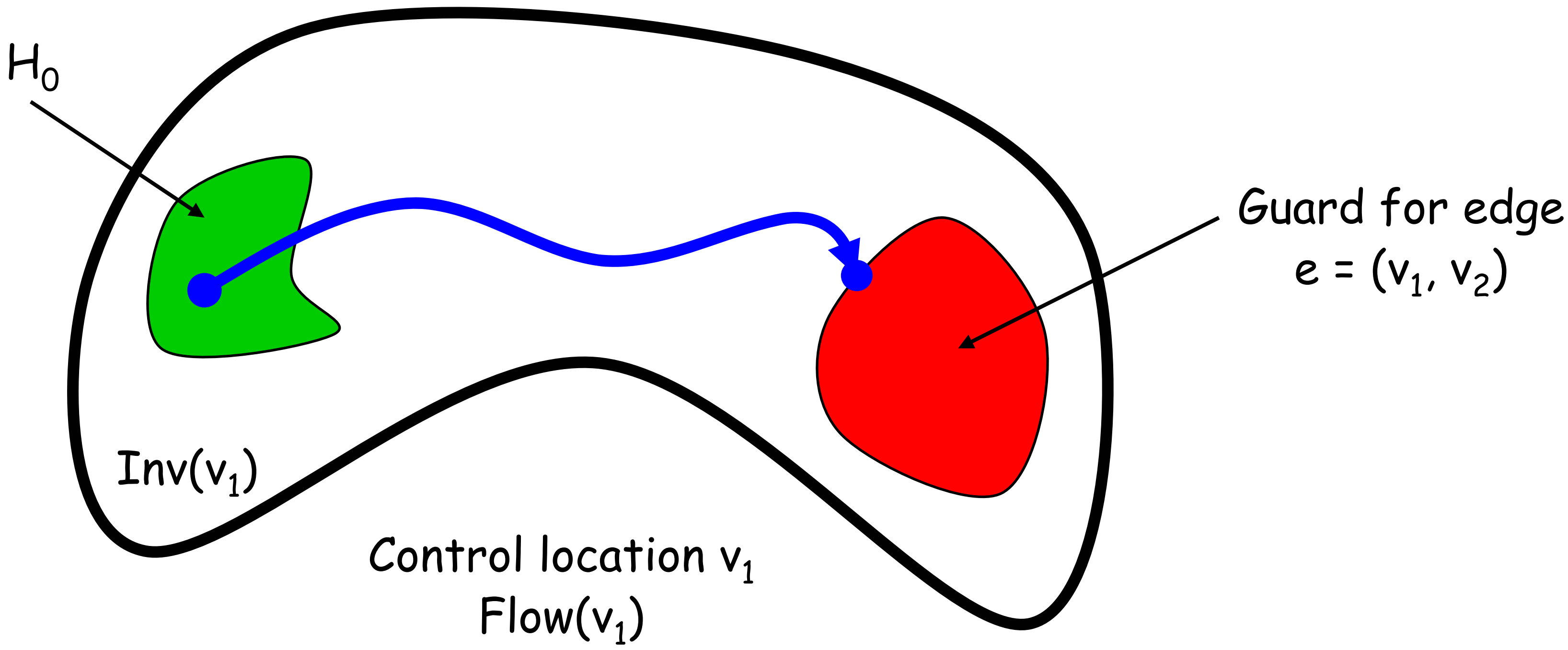
$$H = (X, V, E, Inv, Flow, Init, Guard, F)$$

where

- X is the state space of the system Σ
- V is the set of control locations,
- $E \subseteq V \times V$ is the set of control switches
- $Inv : V \rightarrow P(X)$ assigns an invariant set to each location
- $Flow : V \times X \rightarrow P(\mathbb{R}^n)$ constraints the time derivative of the continuous part of the state
- $Init : V \rightarrow P(X)$ assigns to each control location a set of initial conditions,
- $Guard : E \rightarrow P(X)$ is the guard condition that enables a control switch e in E
- $F \subseteq V$ is the set of final locations

Trajectories of the Hybrid Automaton

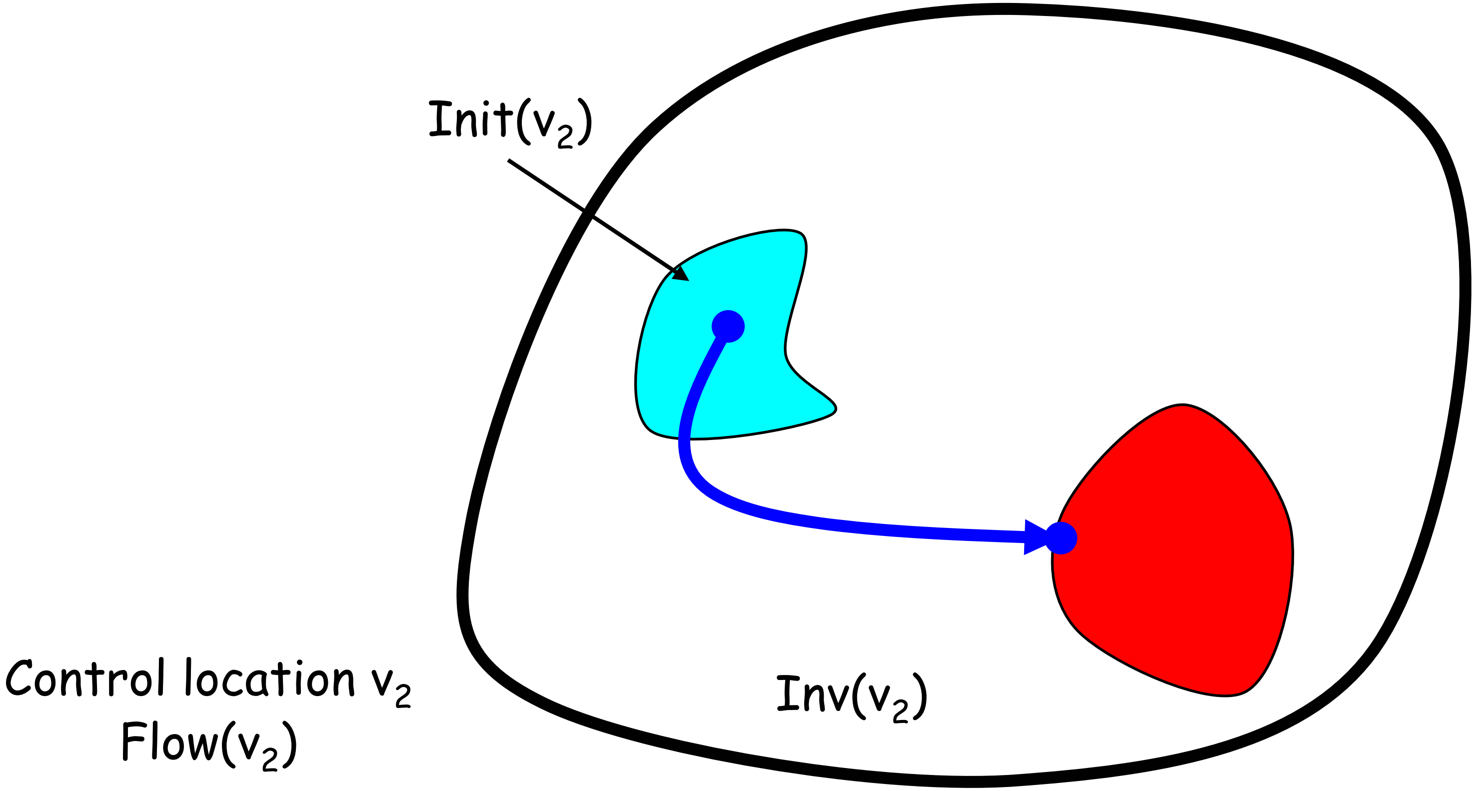
Formally, the semantics of a hybrid automaton are given in terms of timed transition systems $T_H = (H, H_0, \rightarrow)$.



Note: the reset map is the identity, transitions are forced

Trajectories of the Hybrid Automaton

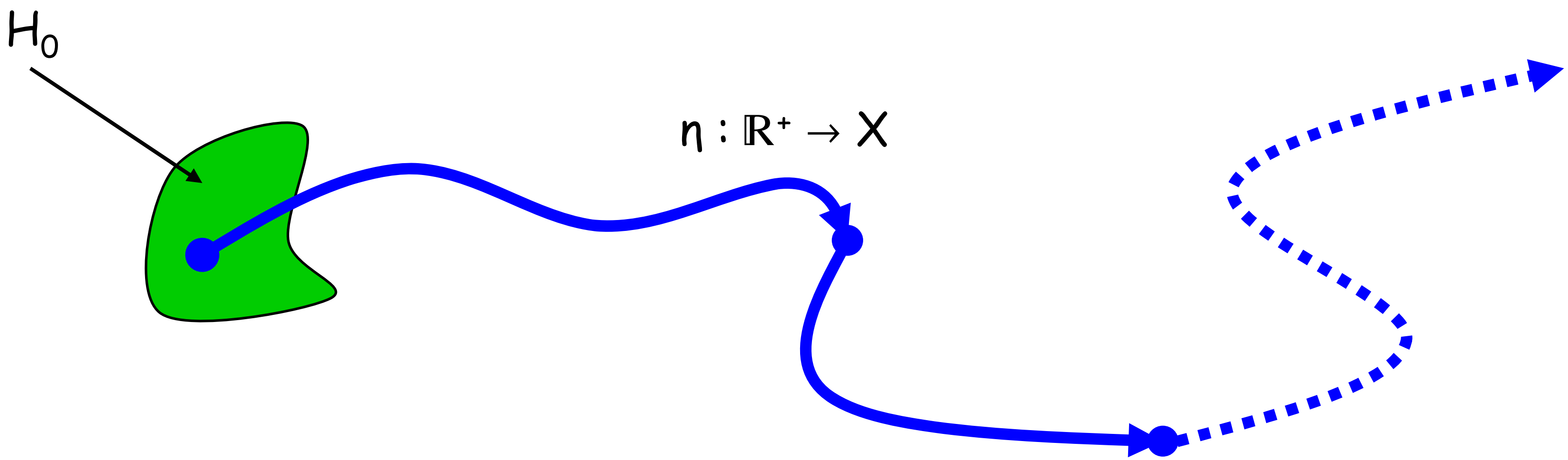
Formally, the semantics of a hybrid automaton are given in terms of timed transition systems $T_H = (H, H_0, \rightarrow)$.



Note: the reset map is the identity, transitions are forced

Language of the timed transition system

The set of all trajectories η of T_H starting from a state in H_0 is the language $L(T_H)$ of the timed transition system T_H.

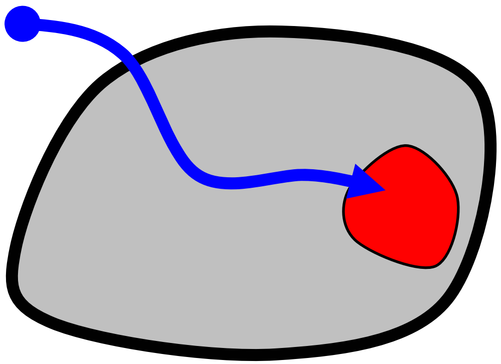


Linear Temporal Logic: Continuous Time

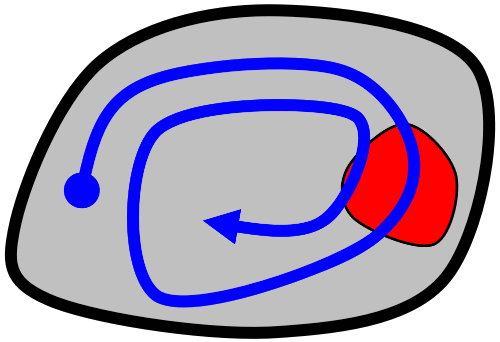
Like propositional logic, but also reasoning wrt time …

Basic operators:
Propositional: ∧, ∨, ¬
Temporal: ◊, □, U, R

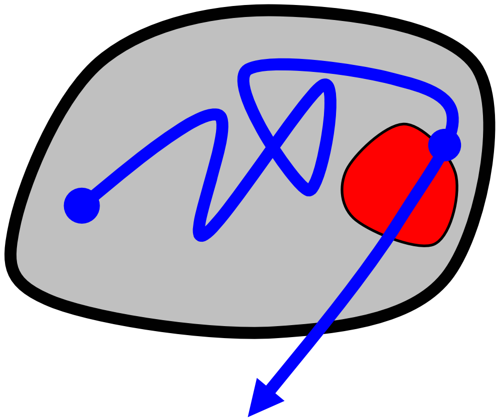
◊(red)
Eventually red



□(gray)
Always gray



(gray) U (red)
gray Until red



Flat LTL with continuous time semantics

Let Π be a set of atomic propositions and Π^* be the set of Boolean combinations of atomic propositions, i.e. $(\pi_1 \wedge \pi_2) \vee \neg \pi_3$.

Define the denotation $[[.]] : \Pi \rightarrow P(X)$ which extends naturally over Π^* as:

$$[[\pi_1 \wedge \pi_2]] = [[\pi_1]] \cap [[\pi_2]] \text{ and } [[\neg\pi]] = [[\pi]]^c$$

Syntax in NNF: $\varphi ::= \pi^* \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \pi^* U \varphi_2 \mid \pi^* R \varphi_2$

Semantics:

$\eta \vDash \pi^*$ iff $\eta(0) \in [[\pi^*]]$

$\eta \vDash \varphi_1 \wedge \varphi_2$ iff $\eta \vDash \varphi_1$ and $\eta \vDash \varphi_2$

$\eta \vDash \varphi_1 \vee \varphi_2$ iff $\eta \vDash \varphi_1$ or $\eta \vDash \varphi_2$

$\eta \vDash \pi^* U \varphi$ iff $\exists t \geq 0$ s.t. $\eta|_t\ \eta \vDash \varphi_2$ and $\forall s \in [0,t]\ \eta(s) \in [[\pi^*]]$

$\eta \vDash \pi^* R \varphi$ iff $\forall t \geq 0\ \eta|_t\ \eta \vDash \varphi_2$ or $\exists s \in [0,t]$ s.t. $\eta(s) \in [[\pi^*]]$

Some notation: $\eta|_t(s) = \eta(t+s)$ $\quad$ $LTL(op_1, \ldots, op_n)$

$\Diamond\varphi = T U \varphi$ $\quad$ $\Box\varphi = F R \varphi$

LTL(U,$\vee$,$\wedge$) to HA

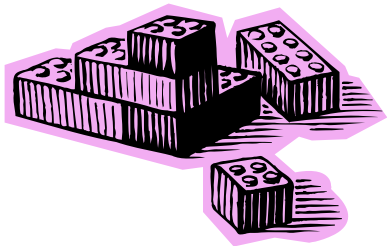
1 **Algorithm**

Modular construction using the structure of φ.
($\equiv$ proof by induction)

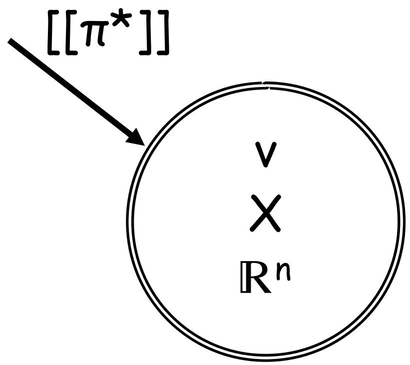
Syntax in NNF: $\varphi ::= \pi^* \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \pi^* U \varphi$

Ex. $\Diamond(\pi_2 \wedge \Diamond(\pi_3 \wedge \Diamond(\pi_4 \wedge (\neg\pi_2 \wedge \neg\pi_3) U \pi_1)))$

LTL(U,$\vee$,$\wedge$) to HA

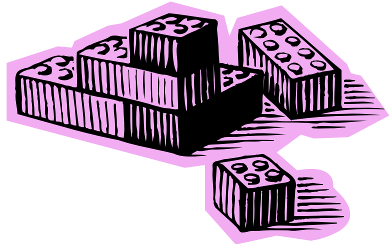


Proposition 2 (base case): Let $\varphi = \pi$, then
$H_{\pi^*} = (X, \{v\}, \emptyset, X, \mathbb{R}^n, [[\pi^*]], \emptyset, \{v\})$



proof immediate from definition: $\eta \models \pi^*$ iff $\eta(0) \in [[\pi^*]]$

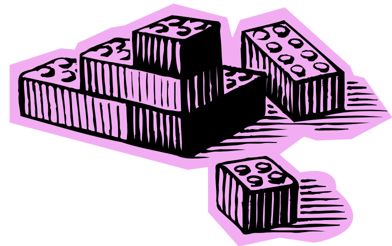
Basic Building Blocks



$\varphi = \varphi_1 \wedge \varphi_2$ or $\varphi = \varphi_1 \vee \varphi_2$

Proposition 1 [Henzinger 95]:
Let H_1 and H_2 be hybrid automata.
If $H = H_1 \cup H_2$, then $L(T_H) = L(T_{H1}) \cup L(T_{H2})$.
If $H = H_1 \cap H_2$, then $L(T_H) = L(T_{H1}) \cap L(T_{H2})$.

LTL(U,∨,∧) to HA

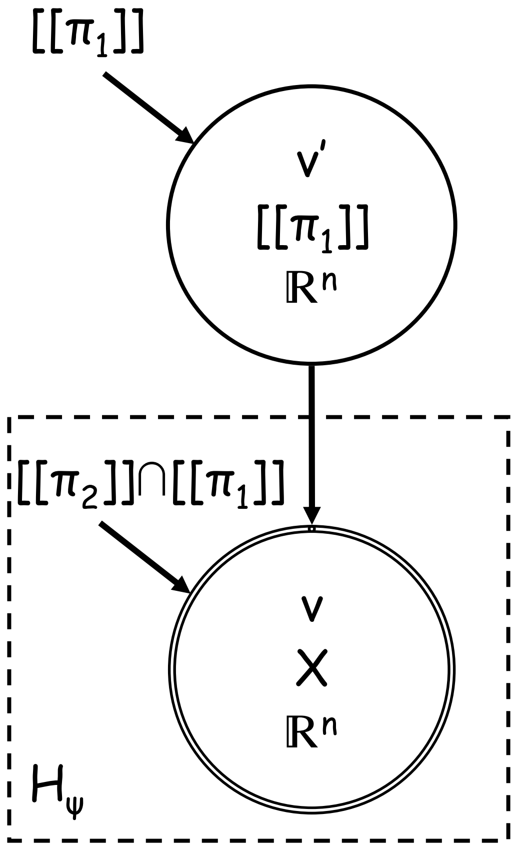


Proposition 3: Let $\varphi = \pi^* U \psi$, then
$H_\varphi = (X, V \cup \{v'\}, E', Inv', Flow', Init', Guard', F)$
Let $H_\psi = (X, V, E, Inv, Flow, Init, Guard, F)$. Then:

- $Inv'(v') = Init'(v') = [[\pi^*]]$ and $Flow(v') = \mathbb{R}^n$
- $Inv'(v) = Inv(v)$ for all $v \in V$
- $Flow'(v) = Flow(v)$ for all $v \in V$
- $Init'(v) = Init(v) \cap [[\pi^*]]$ for all $v \in V$
- $E' = E \cup \{(v', v) \mid v \in V_{in}\}$
- $Guard'(v', v) = Init'(v)$ for all $v \in V_{in}$
- $Guard'(e) = Guard(e)$ for all $e \in E$

where $V_{in} = \{v \in V \mid Init(v) \cap [[\pi^*]] \neq \emptyset\}$

Ex. $\pi_1 U \pi_2$



proof immediate from definition:
$\eta \vDash \pi^* U \psi$ iff $\exists t \geq 0$ s.t. $\eta|_t\ \eta \vDash \psi$ and $\forall s \in [0,t]\ \eta(s) \in [[\pi^*]]$

LTL(U,$\vee$,$\wedge$) to HA

Algorithm 1 The LTL($\underline{\mathcal{U}}, \wedge, \vee$) Fragment

Input: A formula $\phi \in$ LTL($\underline{\mathcal{U}}, \wedge, \vee$)
Output: The hybrid automaton $\mathcal{H}_\phi$

procedure $\mathrm{LTL}_{\mathcal{U}}\mathrm{TOHA}(\phi)$
  if $\phi = \overline{\pi}$ **then**
    return $\mathcal{H}_{\overline{\pi}}$ ▷ Proposition 2
  else if $\phi = \phi_1 \wedge \phi_2$ **then**
    return $\mathrm{LTL}_{\mathcal{U}}\mathrm{TOHA}(\phi_1) \cap \mathrm{LTL}_{\mathcal{U}}\mathrm{TOHA}(\phi_2)$
  else if $\phi = \phi_1 \vee \phi_2$ **then**
    return $\mathrm{LTL}_{\mathcal{U}}\mathrm{TOHA}(\phi_1) \cup \mathrm{LTL}_{\mathcal{U}}\mathrm{TOHA}(\phi_2)$
  else if $\phi = \overline{\pi}\mathcal{U}\psi$ **then**
    $\mathcal{H}_\psi \leftarrow \mathrm{LTL}_{\mathcal{U}}\mathrm{TOHA}(\psi)$
    return $\mathcal{H}_\phi$ ▷ Proposition 3
  end if
end procedure

LTL(U,∨,∧) to HA

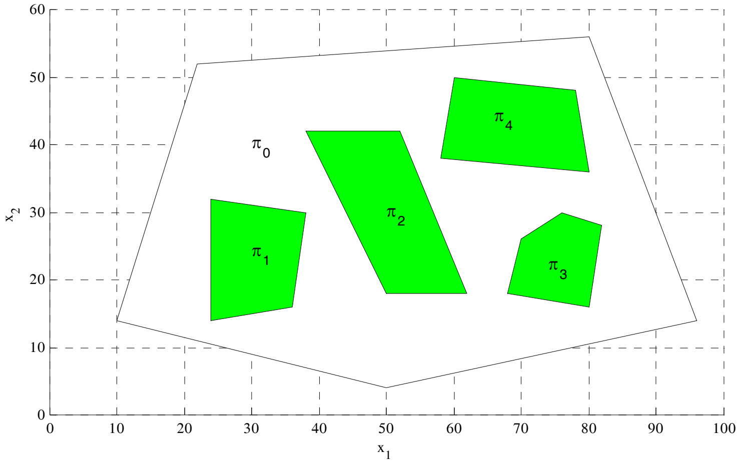
Remarks:

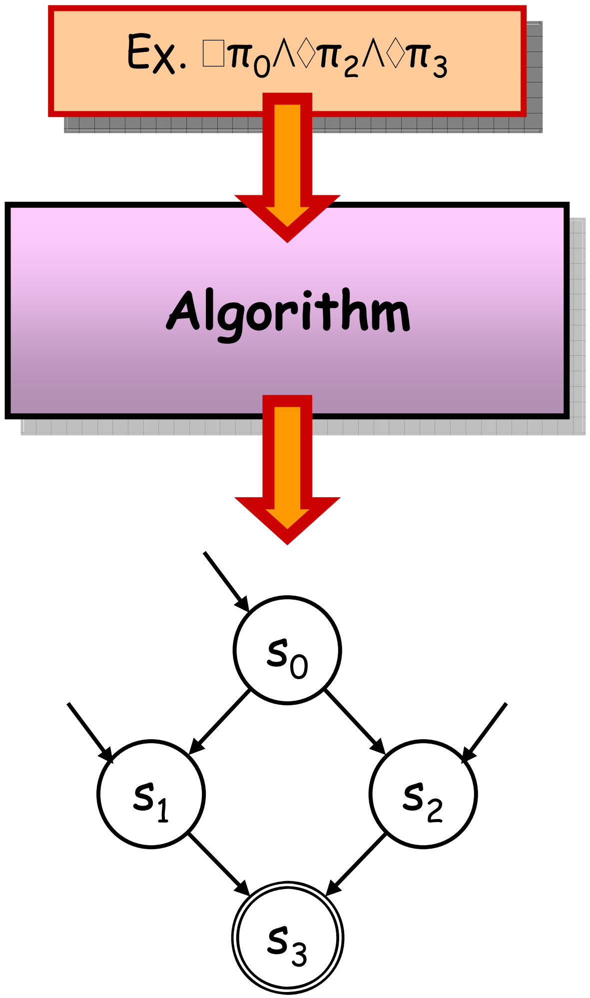
- "Fairness conditions"
- Computational complexity
  - Size of H1∩H2: $O(n_1 n_2)$
  - Size of final automaton $O(\exp(|\varphi|))$
  - Set intersections and complementation
    - set theoretic operations, not reachability etc.
- Note: graph is acyclic
- Final hybrid automaton might have empty guards and invariant sets
  - Then formula might be unsatisfiable

$LTL_{U,\Box}$ fragment

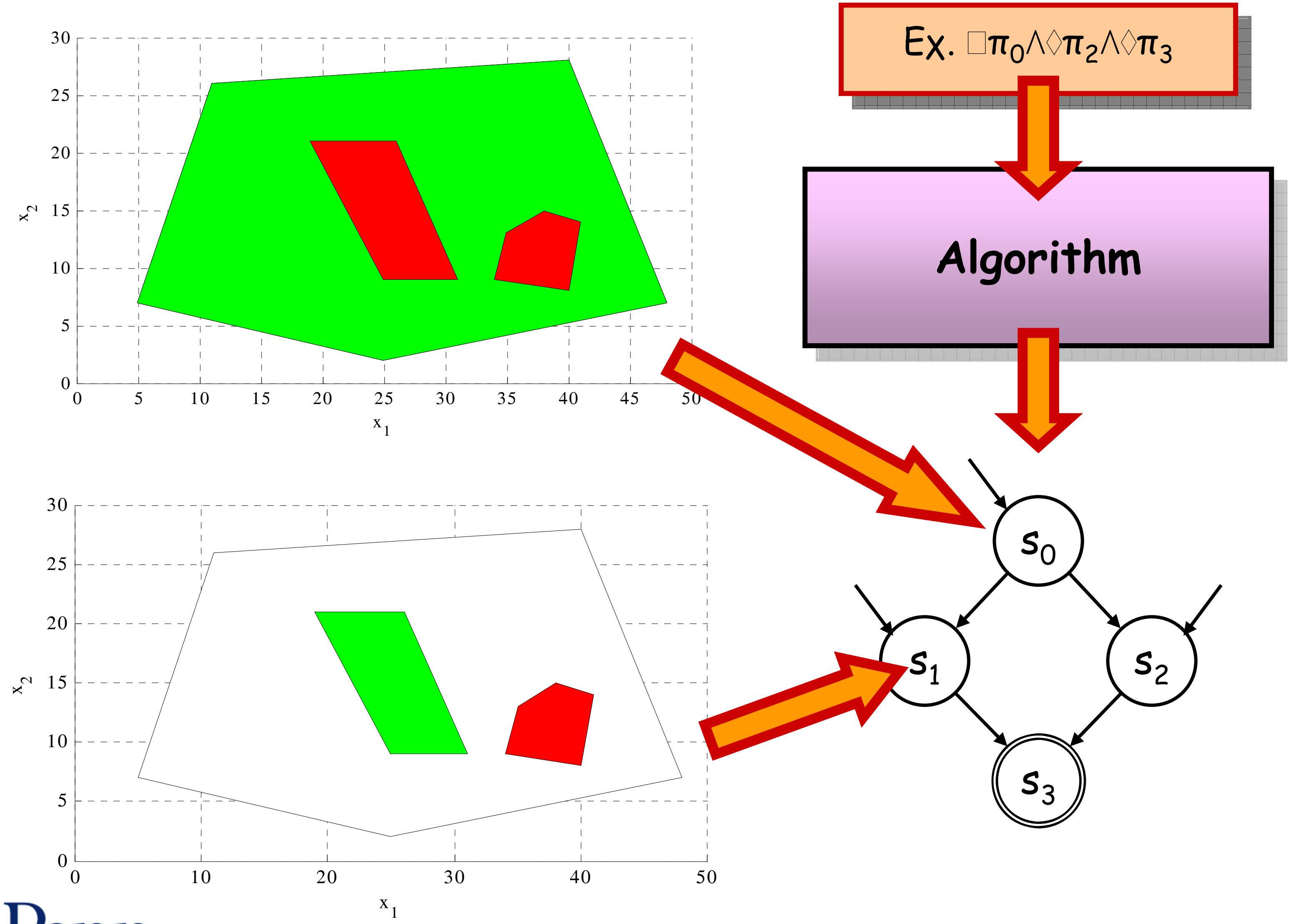
- Similar to $LTL(U,\vee,\wedge)$ we can get an algorithm for $LTL(\Box,\vee,\wedge)$
- General solution for $\varphi_1 \wedge \varphi_2$ or $\varphi_1 \vee \varphi_2$
  - with $\varphi_1 \in LTL(U,\vee,\wedge)$ and $\varphi_2 \in LTL(\Box,\vee,\wedge)$
- More general solution for any formula:
  - $\varphi ::= \pi^* \mid \varphi_{\Box} \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \pi^* U \varphi$
  - $\varphi_{\Box} \in LTL(\Box,\vee,\wedge)$
  - Ex. $\Box\pi_0 \wedge \Diamond(\pi_2 \wedge \Diamond(\pi_3 \wedge \Diamond(\pi_4 \wedge (\neg\pi_2 \wedge \neg\pi_3) U \pi_1)))$

“Abstract” Hybrid Automaton





"Abstract" Hybrid Automaton

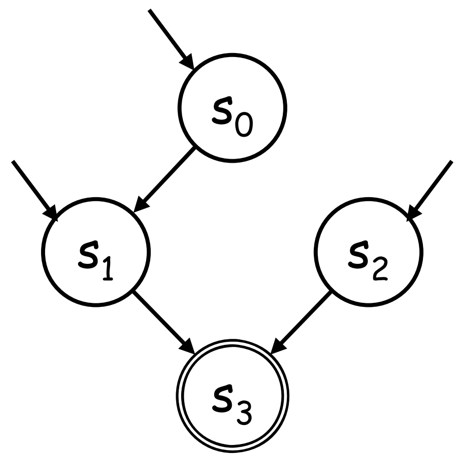


A practical solution

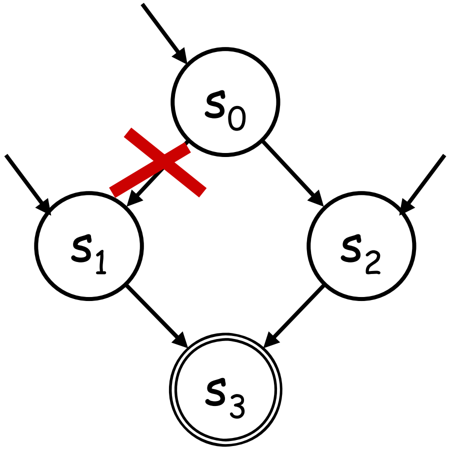
Backward reachability using
Breadth First Search:

✓ fix initial conditions

✓ one outgoing edge from each control location



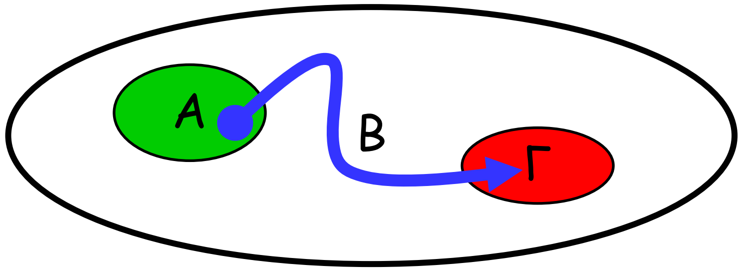
2 Derive controller constraints



Feedback control law $g_c : X \longrightarrow U$

Controller constraints $c = \{A, B, \Gamma\}$

- A initial conditions, i.e. $x(0) \in A$
- Γ final conditions, i.e. $x(t_g) \in \Gamma$
- B invariant, $\forall t \in [0, t_g].x(t) \in B$



$c_3 = \{\cup_{e \in I3} Guard(e), Inv(s_3), \emptyset\} \implies g_{c3}$

$c_1 = \{Guard(e_{I1}) \cup Init(s_1), Inv(s_1), Guard(e_{O1})\} \implies g_{c1}$

$c_2 = \{Init(s_2), Inv(s_2), Guard(e_{O2})\} \implies g_{c2}$

$\cancel{c_0 = \{Init(s_0), Inv(s_0), Guard(e_{O1})\} \implies g_{c0}}$

$c_0 = \{Init(s_0), Inv(s_0), Guard(e_{O1})\} \implies g_{c0}$

Main result

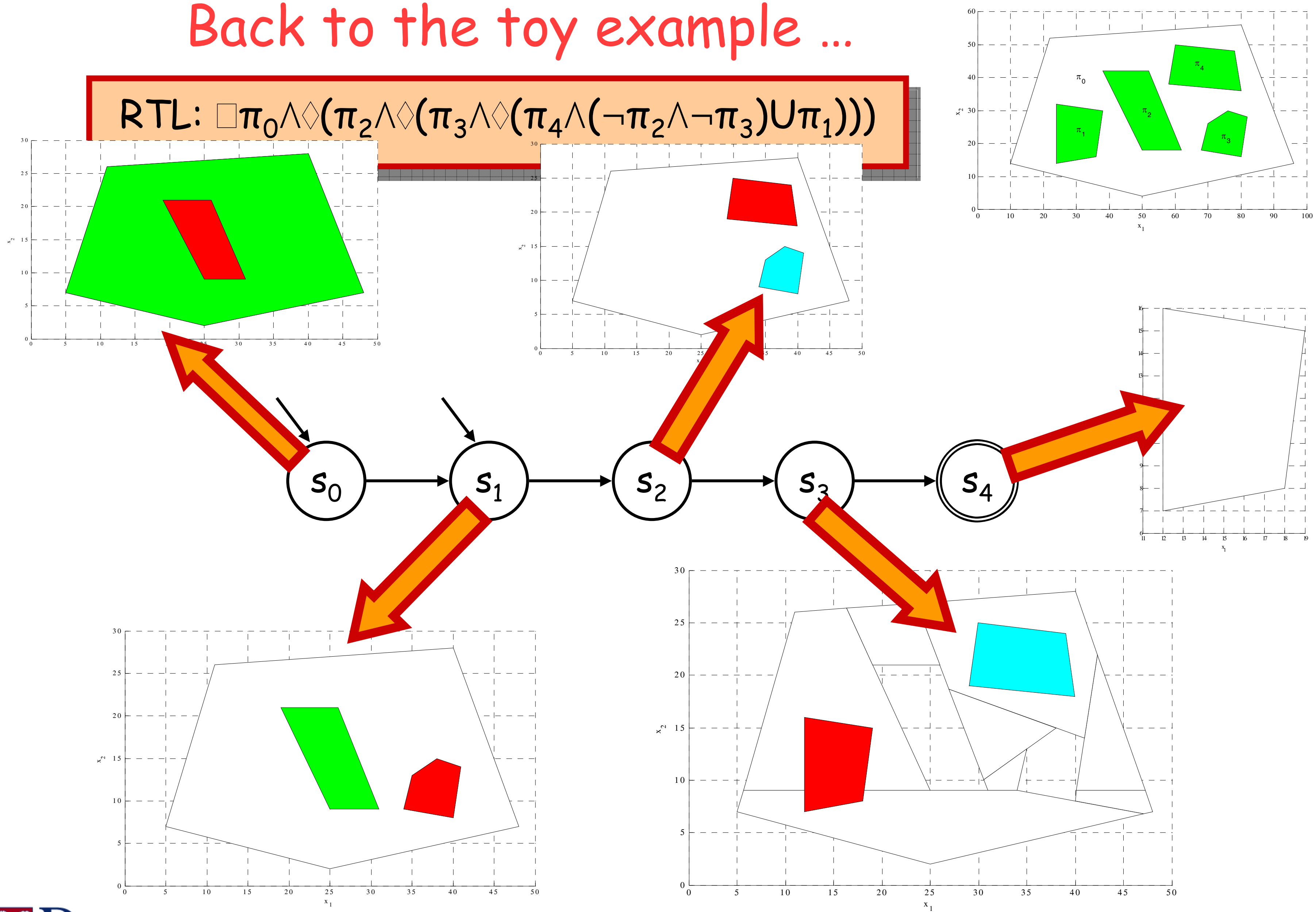
Let $\varphi \in LTL_{U,\square}$ be satisfiable wrt to our model of computation. Then:

(i) φ can be converted into list of controller specifications.

(ii) The hybrid automaton H resulting from the closed-loop dynamics satisfies: $\forall \eta \in L(H) \,.\, \eta \models \varphi$

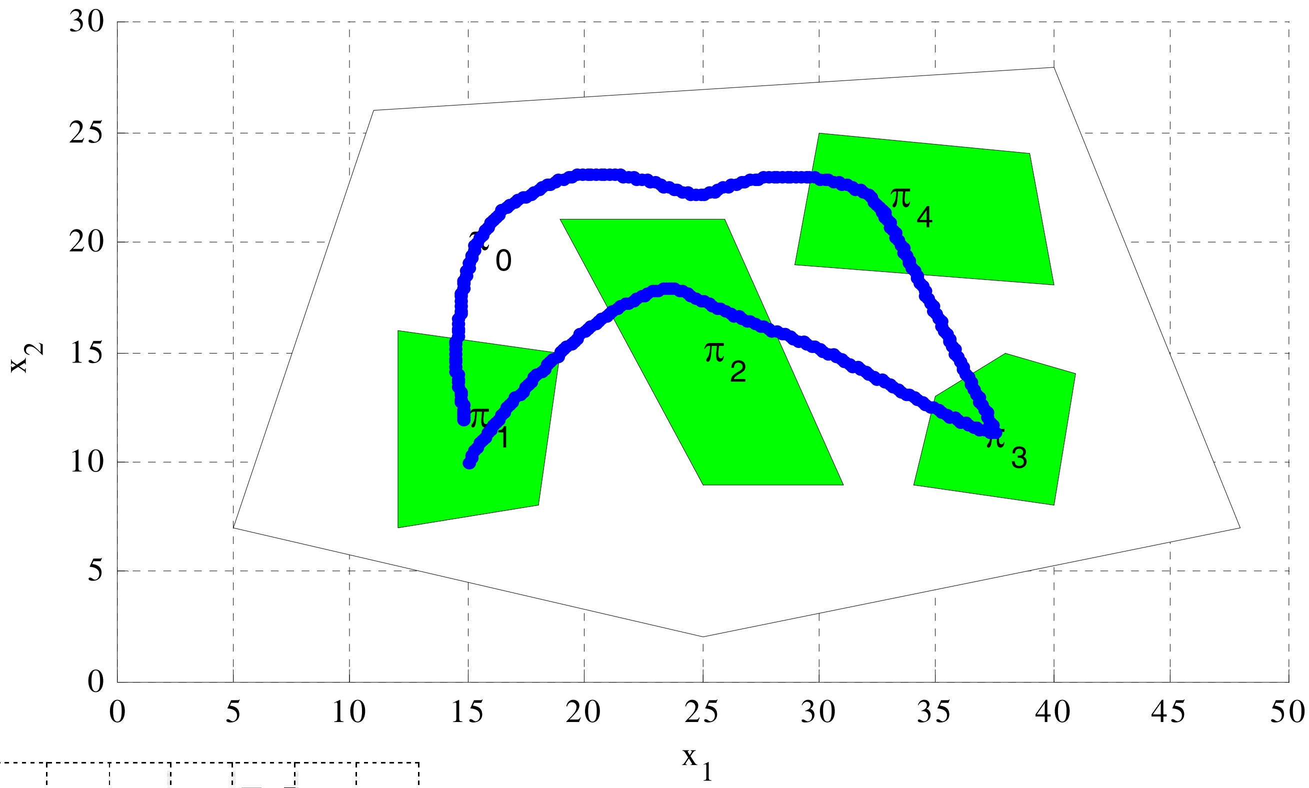
Back to the toy example …

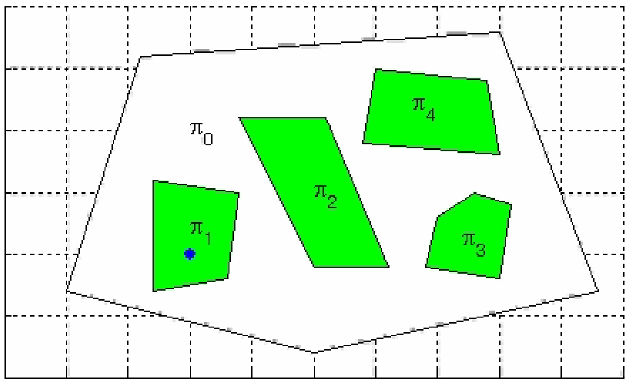
RTL: $\square \pi_0 \wedge \Diamond(\pi_2 \wedge \Diamond(\pi_3 \wedge \Diamond(\pi_4 \wedge (\neg \pi_2 \wedge \neg \pi_3) U \pi_1)))$

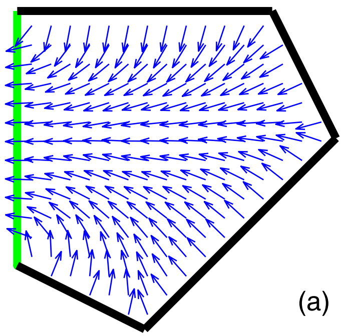


13 Controllers

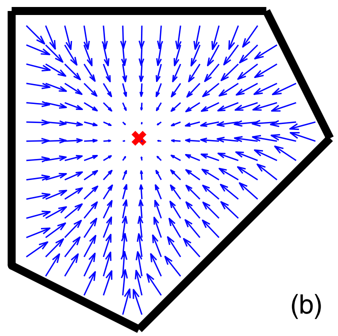
Back to the toy example ...







(a)



(b)

Conner et. al. IROS 2003

Conclusions

- ☑ Top-down approach for the synthesis of HA from temporal logic
  - ☑ identified a fragment of LTL that provides a modular construction
  - ☑ algorithm is based on the closure properties of HA
- ☑ Advantages of the proposed framework:
  - ☑ Clean separation between software specification and controller design
  - ☑ Potentially smaller number of required controllers
  - ☑ The closed-loop dynamics can be designed using different methodology in each control location
  - ☑ "Real" continuous time semantics
    - ☑ Ability to distinguish between events that must hold at a point in time or for an interval

Future work

- ☑ Introduce robustness into the system
  - ☑ Fainekos, Girard, Pappas: *Hierarchical Synthesis of Hybrid Controllers from Temporal Logic Specifications*, HSCC 2007 (to appear)
- ❑ Complete the framework for the full LTL
  - ❑ i.e. add "liveness"
- ❑ Design for distributed specifications
- ❑ Built a complete toolbox with libraries of controllers

Thank You!

Any Question(s) ?