Static Analysis for State-Space Reduction of Polygonal Hybrid Systems

Gordon Pace[1] Gerardo Schneider[2]

[1]Dept. of Computer Science and AI – University of Malta

[2]Dept. of Informatics – University of Oslo

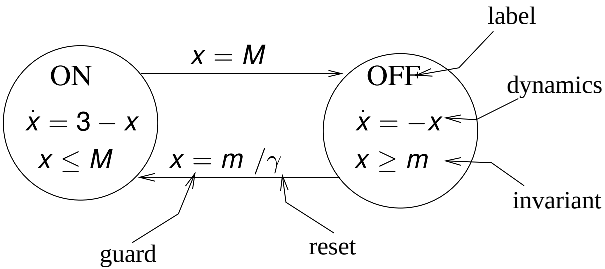
FORMATS'06
25-27 September 2006, Paris

Outline

Outline

Hybrid Systems

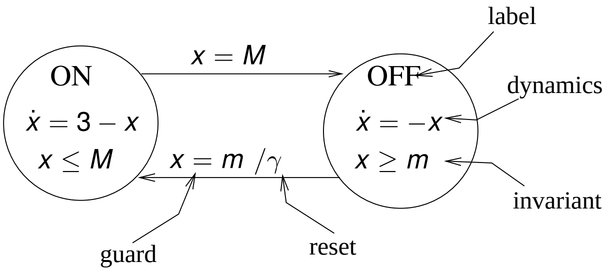
- Usual representation: Hybrid Automata



In general, we can have *differential inclusions* instead of differential equations

Hybrid Systems

- Usual representation: Hybrid Automata



In general, we can have *differential inclusions* instead of differential equations

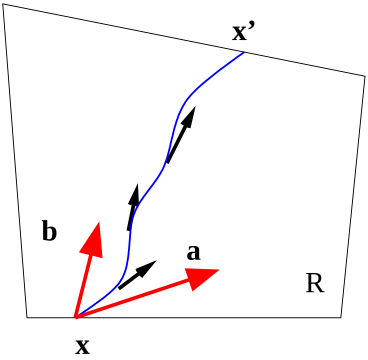
Outline

Polygonal Hybrid Systems (SPDIs)

- A finite partition of the plane into convex polygonal sets
- Dynamics given by the angle determined by two vectors: $\dot{x} \in \angle_{\mathbf{a}}^{\mathbf{b}}$

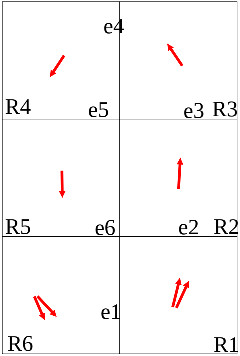
Polygonal Hybrid Systems (SPDIs)

- A finite partition of the plane into convex polygonal sets
- Dynamics given by the angle determined by two vectors: $\dot{x} \in \angle_{\mathbf{a}}^{\mathbf{b}}$



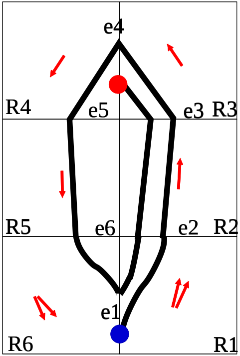
Polygonal Hybrid Systems (SPDIs)

- A finite partition of the plane into convex polygonal sets
- Dynamics given by the angle determined by two vectors: $\dot{x} \in \angle_{\mathbf{a}}^{\mathbf{b}}$



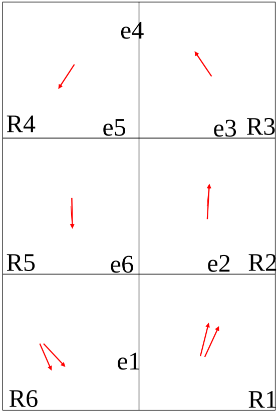
Polygonal Hybrid Systems (SPDIs)

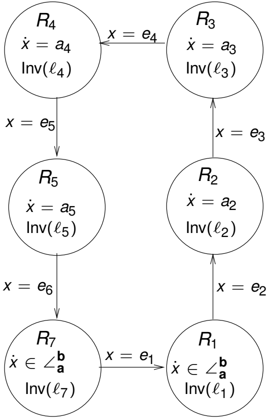
- A finite partition of the plane into convex polygonal sets
- Dynamics given by the angle determined by two vectors: $\dot{x} \in \angle_{\mathbf{a}}^{\mathbf{b}}$



Polygonal Hybrid Systems (SPDIs)

- An SPDI can be seen as a hybrid automaton





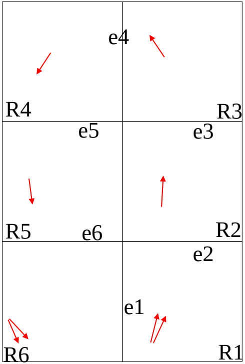
Polygonal Hybrid Systems (SPDIs)
Underlying Graph

- The reachability algorithm operates on a graph representation, not on the automaton

Polygonal Hybrid Systems (SPDIs)

Underlying Graph

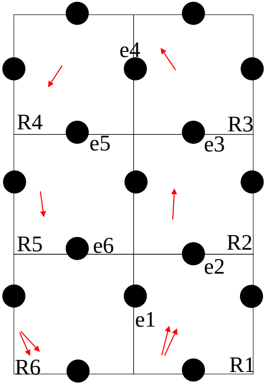
- The reachability algorithm operates on a graph representation, not on the automaton



Polygonal Hybrid Systems (SPDIs)

Underlying Graph

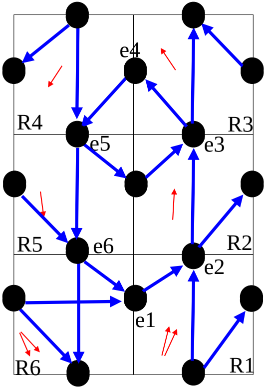
- The reachability algorithm operates on a graph representation, not on the automaton



Polygonal Hybrid Systems (SPDIs)

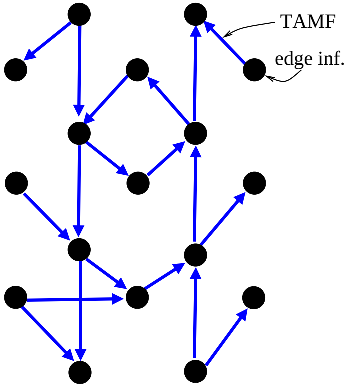
Underlying Graph

- The reachability algorithm operates on a graph representation, not on the automaton



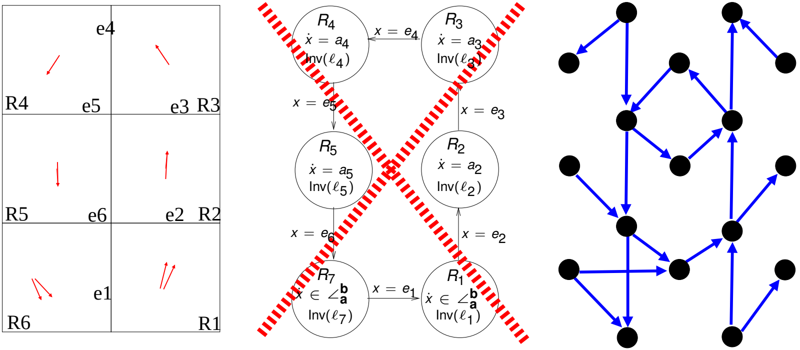
Polygonal Hybrid Systems (SPDIs)
Underlying Graph

- The reachability algorithm operates on a graph representation, not on the automaton



Polygonal Hybrid Systems (SPDIs)

Three Views



- We will, however, use the geometrical representation in what follows instead for clarity of presentation

Known Results about SPDIs

- Reachability is decidable –in the plane
  (based on Poincaré maps, finite characterization of simple cycles, acceleration, ...)
  - DFS algorithm (HSCC'01)
  - BFS algorithm (VMCAI'04)
  - Tool: SPeeDI (CAV'02)
- Reachability is undecidable –3-dim and higher (ICALP'94)
- For slights extensions in 2-dim reachability is an open question, for others is undecidable (CONCUR'02, FSTTCS'05)
- Phase portrait computation
  - Viability and controllability kernels (HSCC'02)
  - Invariance kernels (NJC'04)

Contributors: E. Asarin, O. Maler, V. Mysore, G. Pace, A. Pnueli, G. Schneider, S. Yovine

Outline

Motivation

- Application: Use of SPDIs for approximating non-linear differential equations
  - Triangulation of the plane: Huge number of regions
- Need to reduce the state space (for reachability analysis)...
  ... without too much overhead

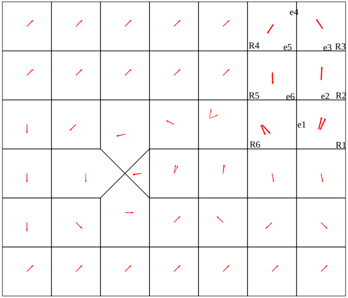
Motivation

- Application: Use of SPDIs for approximating non-linear differential equations
  - Triangulation of the plane: Huge number of regions
- Need to reduce the state space (for reachability analysis)...
  ... without too much overhead

Outline

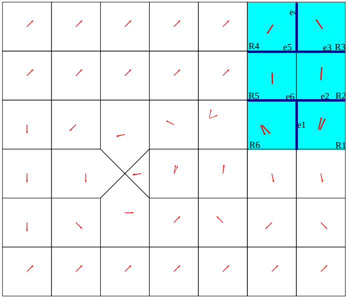
Few Preliminaries

- We only need to consider simple cycles
  - Given a sequence of non-repeating edges (except for the first and last edge) – e.g., $\sigma = e_1, \cdots, e_k, e_1$
  - Consider the polygonal subset of the SPDI determined by such sequence (denoted K_σ)



Few Preliminaries

- We only need to consider simple cycles
  - Given a sequence of non-repeating edges (except for the first and last edge) – e.g., $\sigma = e_1, \cdots, e_k, e_1$
  - Consider the polygonal subset of the SPDI determined by such sequence (denoted K_σ)

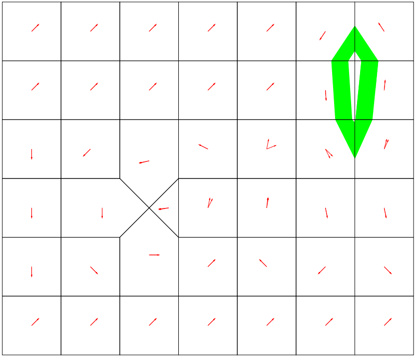


Controllability Kernels

- Given K_σ, its controllability kernel is the largest subset such that any two points are reachable from each other

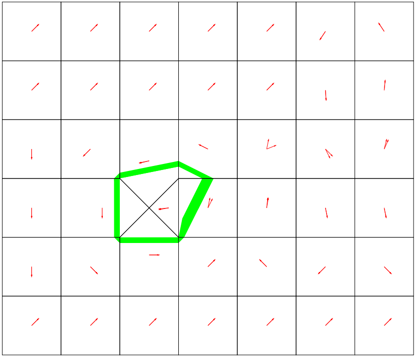
Controllability Kernels

- Given K_σ, its controllability kernel is the largest subset such that any two points are reachable from each other



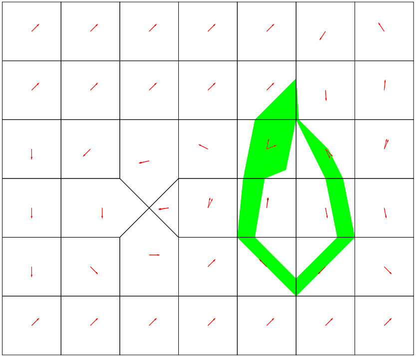
Controllability Kernels

- Given K_σ, its controllability kernel is the largest subset such that any two points are reachable from each other



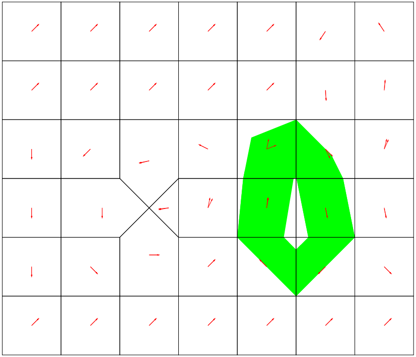
Controllability Kernels

- Given K_σ, its controllability kernel is the largest subset such that any two points are reachable from each other



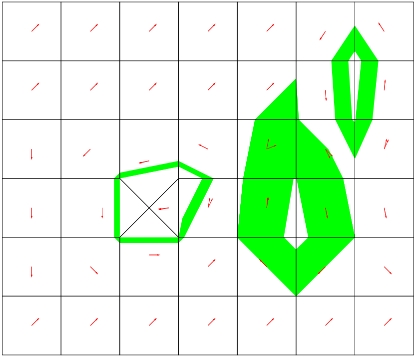
Controllability Kernels

- Given K_σ, its controllability kernel is the largest subset such that any two points are reachable from each other



Controllability Kernels

- Given K_σ, its controllability kernel is the largest subset such that any two points are reachable from each other

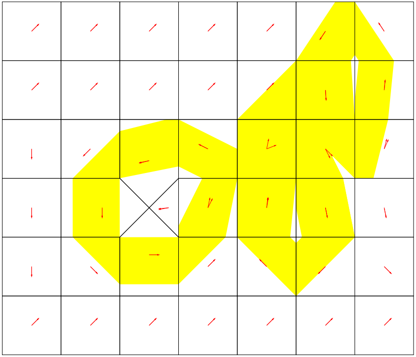


Viability Kernels

- Given K_σ, its viability kernel is the largest subset such that for any point in the set, there is at least one trajectory which remains in the set forever

Viability Kernels

- Given K_σ, its viability kernel is the largest subset such that for any point in the set, there is at least one trajectory which remains in the set forever

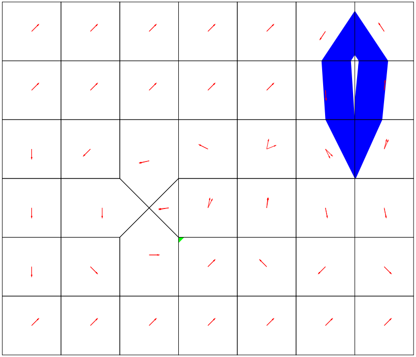


Invariance Kernels

- Given K_σ, its invariance kernel is the largest subset such that for any point x in the set, there is at least one trajectory starting in it and every trajectory starting in x is viable

Invariance Kernels

- Given K_σ, its invariance kernel is the largest subset such that for any point x in the set, there is at least one trajectory starting in it and every trajectory starting in x is viable



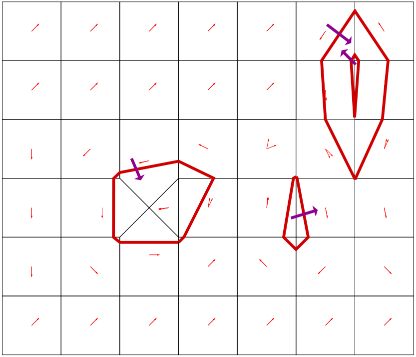
Outline

Semi-Separatrices

- A semi-separatrix is a closed curve dissecting the state space into two subsets such that one is reachable from the other but not vice-versa

Semi-Separatrices

- A semi-separatrix is a closed curve dissecting the state space into two subsets such that one is reachable from the other but not vice-versa



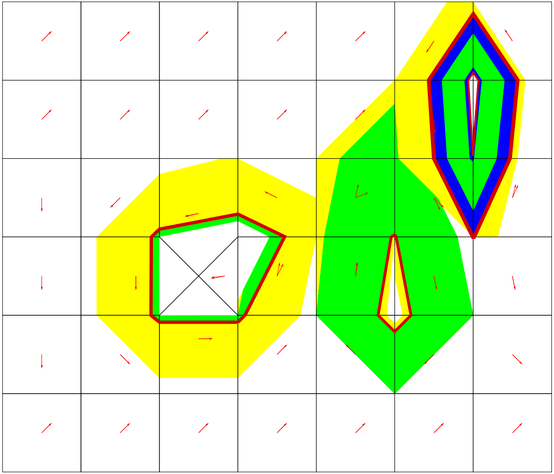
Semi-Separatrices

Based on properties of limit trajectories on simple cycles and the invariance kernel we have an algorithm for computing semi-separatrices

Theorem

The computation of semi-separatrices for SPDIs is decidable

Phase Portrait



Outline

1. Introduction
   - Hybrid Systems
   - Polygonal Hybrid Systems
   - Motivation
2. Phase Portrait of SPDIs
   - Kernels
   - Semi-Separatrices
3. State-Space Reduction
   - Using Semi-Separatrices
   - Using Kernels

State-Space Reduction using Semi-Separatrices

Let e be a source edge and e' a target edge

- Identification of *inert* edges
  - Given a semi-separatrix γ, e_I is inert if it lies outside γ while e lies inside, or it lies inside, while e' lies outside

Theorem

Given an SPDI $\mathcal{S}$, a semi-separatrix γ, and edges e and e', then, e' is reachable from e in $\mathcal{S}$ if and only if e' is reachable from e in $\mathcal{S}$ without the inert edges

State-Space Reduction using Semi-Separatrices

Let e be a source edge and e' a target edge

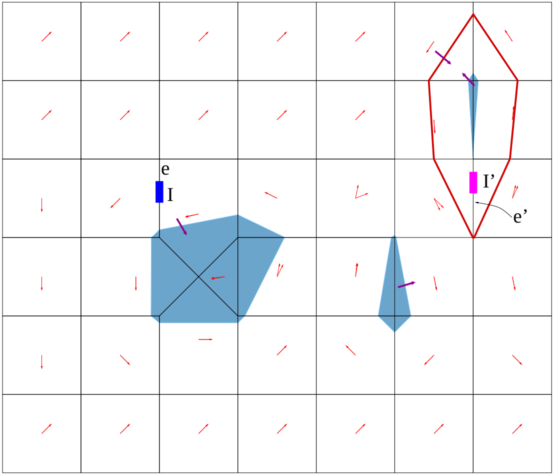
- Identification of *inert* edges
  - Given a semi-separatrix γ, e_I is inert if it lies outside γ while e lies inside, or it lies inside, while e' lies outside

Theorem

Given an SPDI $\mathcal{S}$, a semi-separatrix γ, and edges e and e', then, e' is reachable from e in $\mathcal{S}$ if and only if e' is reachable from e in $\mathcal{S}$ without the inert edges

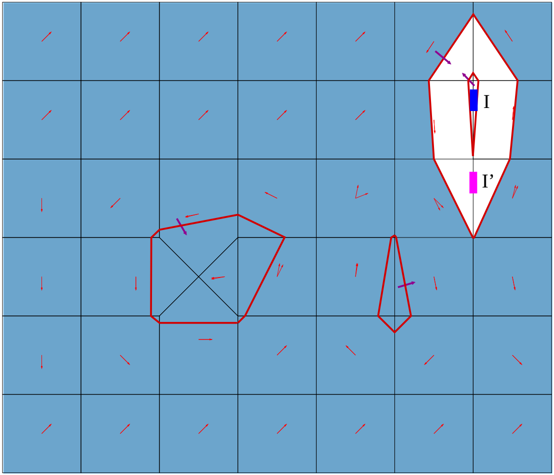
State-Space Reduction using Semi-Separatrices

Example 1



State-Space Reduction using Semi-Separatrices

Example 2



Outline

State-Space Reduction using Kernels

Let e be a source edge and e' a target edge

- Identification of *redundant* edges
  - e_R is redundant if it lies on an opposite side of a controllability kernel as both e and e'

Theorem

Given an SPDI $\mathcal{S}$, a cycle σ, edges e and e', then e' is reachable from e in $\mathcal{S}$ if and only if e' is reachable from e in $\mathcal{S}$ without the redundant edges

State-Space Reduction using Kernels

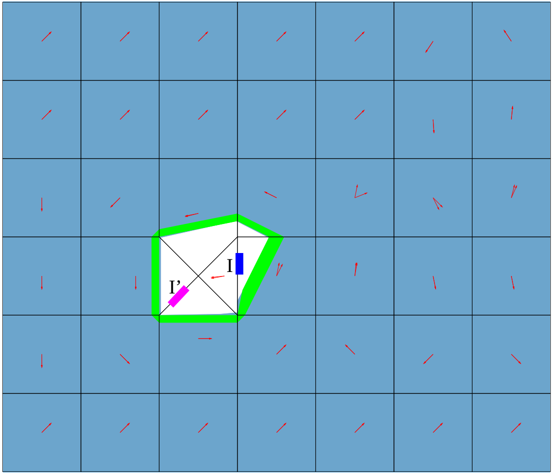
Let e be a source edge and e' a target edge

- Identification of *redundant* edges
  - e_R is redundant if it lies on an opposite side of a controllability kernel as both e and e'

Theorem

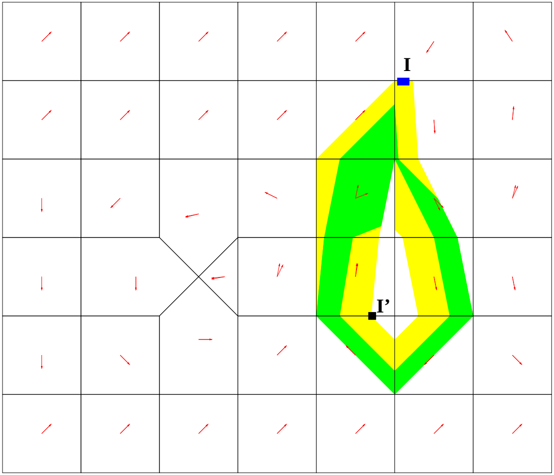
Given an SPDI $\mathcal{S}$, a cycle σ, edges e and e', then e' is reachable from e in $\mathcal{S}$ if and only if e' is reachable from e in $\mathcal{S}$ without the redundant edges

State-Space Reduction using Kernels
Example



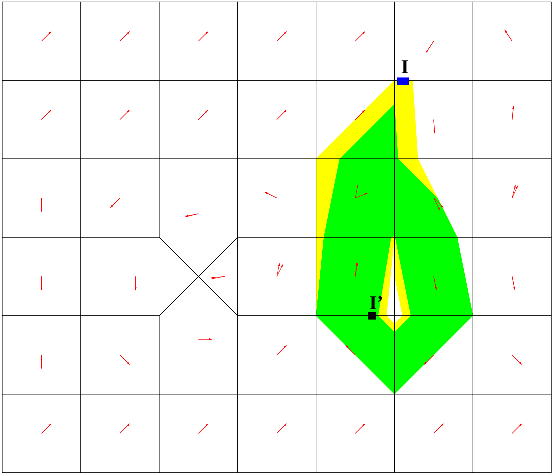
State-Space Reduction using Kernels

Immediate Answer



State-Space Reduction using Kernels

Immediate Answer



Final Remarks
Contributions

- Computation of Semi-Separatrices
- Use of the phase portrait objects to reduce the state-space (for reachability analysis)
  - No extra work needed to perform the optimization: identification and analysis of loops is performed in the first part of the reachability algorithm
- Combination of techniques
  - The detection of *inert* and *redundant* edges may be done by using standard geometrical test (odd-parity test, used in computer graphics)
  - The reduction is then performed on the graph

Final Remarks
Further Work

Extensions and Applications

- Not exact extensions to higher dimensions (undecidable)
  - Maybe use the idea for approximations
- Use of SPDIs for approximating non-linear differential equations on the plane
  - Approximation of phase portrait objects

Implementation

- Implementation in SPeeDI^{+}