Automi Ibridi

Carla Piazza[1]

[1]Dipartimento di Matematica ed Informatica
Università di Udine
carla.piazza@dimi.uniud.it

Indice del Corso (Dis)Ordinato

- Automi Ibridi: Sintassi e Semantica
- Sistemi a stati finiti (breve ripasso)
- Il problema della Raggiungibilità
- Risultati di Indecidibilità
- Classi notevoli di Automi Ibridi: timed, rectangular, o-minimal, ...
- Tecniche di Decisione: (Bi)Simulazione, Cylindric Algebraic Decomposition, Teoremi di Selezione, Semantiche approssimate
- ...e tanto altro:
  - Logiche temporali
  - Composizione di Automi
  - Il caso Stocastico
  - Stabilità, Osservabilità, Controllabilità
  - Strumenti Software
  - Applicazioni

Un po' di Storia

C'erano una volta . . .

- Logiche Temporali e (Finite) Model Checking
  erano gli inizi degli '80 . . .
  ma ne parleremo la prossima volta
- Già da tempo ingegneri, matematici e fisici si occupavano di Teoria del Controllo
- Agli inizi degli anni '90 prendendo spunto dalle due discipline (forse) Henzinger, Alur, Courcobetis, Dill introdussero gli Automi Ibridi

Un po' di Storia

C'erano una volta ...

- Logiche Temporali e (Finite) Model Checking
  erano gli inizi degli '80 ...
  ma ne parleremo la prossima volta
- Già da tempo ingegneri, matematici e fisici si occupavano di Teoria del Controllo
- Agli inizi degli anni '90 prendendo spunto dalle due discipline (forse) Henzinger, Alur, Courcobetis, Dill introdussero gli Automi Ibridi

Un po' di Storia

C'erano una volta ...

- Logiche Temporali e (Finite) Model Checking
  erano gli inizi degli '80 ...
  ma ne parleremo la prossima volta
- Già da tempo ingegneri, matematici e fisici si occupavano di Teoria del Controllo
- Agli inizi degli anni '90 prendendo spunto dalle due discipline (forse) Henzinger, Alur, Courcobetis, Dill introdussero gli Automi Ibridi

Un po' di Storia

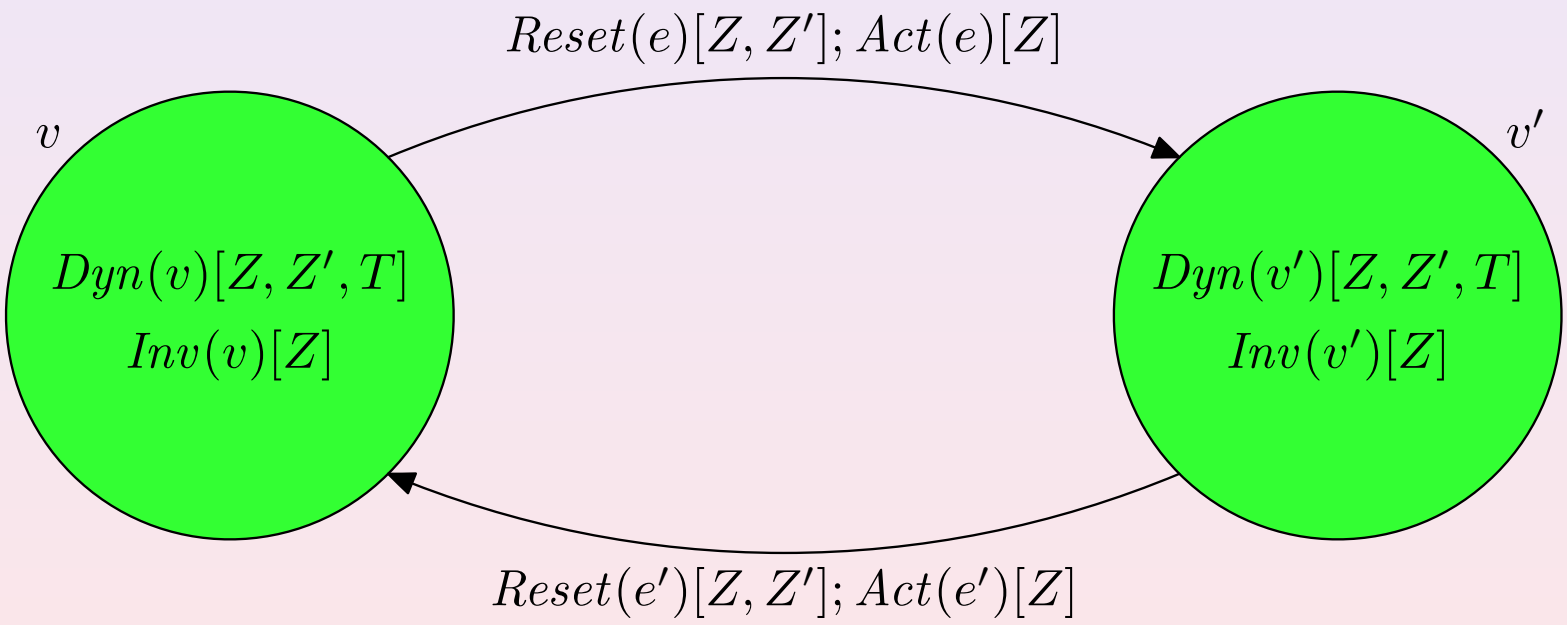
C'erano una volta ...

- Logiche Temporali e (Finite) Model Checking
  erano gli inizi degli '80 ...
  ma ne parleremo la prossima volta
- Già da tempo ingegneri, matematici e fisici si occupavano di Teoria del Controllo
- Agli inizi degli anni '90 prendendo spunto dalle due discipline (forse) Henzinger, Alur, Courcobetis, Dill introdussero gli Automi Ibridi

Hybrid Automata - Intuitivamente

Un automa ibrido H è

un automa a stati finiti con variabili continue Z



Uno stato è una coppia $\langle v, r \rangle$ dove r è una valutazione per Z

Hybrid Automata - Syntax

Definition (Hybrid Automata (Piazza et al.))

A k-hybrid automaton $H = \langle Z, Z', \mathcal{V}, \mathcal{E}, Inv, Dyn, Act, Reset \rangle$ consists of the following components:

1. $Z = (Z_1, \ldots, Z_k)$ and $Z' = (Z'_1, \ldots, Z'_k)$ are two vectors of variables ranging over the reals;
2. $\langle \mathcal{V}, \mathcal{E} \rangle$ is a finite directed graph;
3. Each $v \in \mathcal{V}$ is labeled by the two formulæ $Inv(v)[Z]$ and $Dyn(v)[Z, Z', T]$ such that if $Inv(v)[p]$ holds then $Dyn(v)[p, p, 0]$ holds as well;
4. Each $e \in \mathcal{E}$ is labeled by the formulæ $Act(e)[Z]$ and $Reset(e)[Z, Z']$.

Hybrid Automata - Syntax

Definition (Hybrid Automata (Piazza et al.))

A k-hybrid automaton $H = \langle Z, Z', \mathcal{V}, \mathcal{E}, Inv, Dyn, Act, Reset \rangle$ consists of the following components:

1. $Z = (Z_1, \ldots, Z_k)$ and $Z' = (Z'_1, \ldots, Z'_k)$ are two vectors of variables ranging over the reals;
2. $\langle \mathcal{V}, \mathcal{E} \rangle$ is a finite directed graph;
3. Each $v \in \mathcal{V}$ is labeled by the two formulæ $Inv(v)[Z]$ and $Dyn(v)[Z, Z', T]$ such that if $Inv(v)[p]$ holds then $Dyn(v)[p, p, 0]$ holds as well;
4. Each $e \in \mathcal{E}$ is labeled by the formulæ $Act(e)[Z]$ and $Reset(e)[Z, Z']$.

Hybrid Automata - Syntax

Definition (Hybrid Automata (Piazza et al.))

A k-hybrid automaton $H = \langle Z, Z', \mathcal{V}, \mathcal{E}, Inv, Dyn, Act, Reset \rangle$ consists of the following components:

1. $Z = (Z_1, \ldots, Z_k)$ and $Z' = (Z'_1, \ldots, Z'_k)$ are two vectors of variables ranging over the reals;
2. $\langle \mathcal{V}, \mathcal{E} \rangle$ is a finite directed graph;
3. Each $v \in \mathcal{V}$ is labeled by the two formulæ $Inv(v)[Z]$ and $Dyn(v)[Z, Z', T]$ such that if $Inv(v)[p]$ holds then $Dyn(v)[p, p, 0]$ holds as well;
4. Each $e \in \mathcal{E}$ is labeled by the formulæ $Act(e)[Z]$ and $Reset(e)[Z, Z']$.

Hybrid Automata - Syntax

Definition (Hybrid Automata (Piazza et al.))

A k-hybrid automaton $H = \langle Z, Z', \mathcal{V}, \mathcal{E}, Inv, Dyn, Act, Reset \rangle$ consists of the following components:

1. $Z = (Z_1, \ldots, Z_k)$ and $Z' = (Z'_1, \ldots, Z'_k)$ are two vectors of variables ranging over the reals;
2. $\langle \mathcal{V}, \mathcal{E} \rangle$ is a finite directed graph;
3. Each $v \in \mathcal{V}$ is labeled by the two formulæ $Inv(v)[Z]$ and $Dyn(v)[Z, Z', T]$ such that if $Inv(v)[p]$ holds then $Dyn(v)[p, p, 0]$ holds as well;
4. Each $e \in \mathcal{E}$ is labeled by the formulæ $Act(e)[Z]$ and $Reset(e)[Z, Z']$.

Hybrid Automata - Syntax

Definition (Hybrid Automata (Piazza et al.))

A k-hybrid automaton $H = \langle Z, Z', \mathcal{V}, \mathcal{E}, Inv, Dyn, Act, Reset \rangle$ consists of the following components:

1. $Z = (Z_1, \ldots, Z_k)$ and $Z' = (Z'_1, \ldots, Z'_k)$ are two vectors of variables ranging over the reals;
2. $\langle \mathcal{V}, \mathcal{E} \rangle$ is a finite directed graph;
3. Each $v \in \mathcal{V}$ is labeled by the two formulæ $Inv(v)[Z]$ and $Dyn(v)[Z, Z', T]$ such that if $Inv(v)[p]$ holds then $Dyn(v)[p, p, 0]$ holds as well;
4. Each $e \in \mathcal{E}$ is labeled by the formulæ $Act(e)[Z]$ and $Reset(e)[Z, Z']$.

Note alla Definizione

- *Inv*, *Dyn*, *Act*, *Reset* sono insiemi di formule in un linguaggio (al primo ordine) $\mathcal{L}$
  Ad esempio $\mathcal{L} = (+, *, <, 0, 1)$
- le formule vengono valutate su un modello $\mathcal{M}$ di $\mathcal{L}$ sul dominio $\mathbb{R}$
  Ad esempio $\mathcal{M} = (\mathbb{R}, +, *, <, 0, 1)$
- i nodi $\mathcal{V}$ sono dette locazioni (o *control modes*), gli archi $\mathcal{E}$ sono detti control switches
- la variabile T rappresenta il tempo
- $p \in \mathbb{R}^k$

Note alla Definizione

- *Inv*, *Dyn*, *Act*, *Reset* sono insiemi di formule in un linguaggio (al primo ordine) $\mathcal{L}$
  Ad esempio $\mathcal{L} = (+, *, <, 0, 1)$
- le formule vengono valutate su un modello $\mathcal{M}$ di $\mathcal{L}$ sul dominio $\mathbb{R}$
  Ad esempio $\mathcal{M} = (\mathbb{R}, +, *, <, 0, 1)$
- i nodi $\mathcal{V}$ sono dette locazioni (o *control modes*), gli archi $\mathcal{E}$ sono detti control switches
- la variabile T rappresenta il tempo
- $p \in \mathbb{R}^k$

Note alla Definizione

- *Inv*, *Dyn*, *Act*, *Reset* sono insiemi di formule in un linguaggio (al primo ordine) $\mathcal{L}$
  Ad esempio $\mathcal{L} = (+, *, <, 0, 1)$
- le formule vengono valutate su un modello $\mathcal{M}$ di $\mathcal{L}$ sul dominio $\mathbb{R}$
  Ad esempio $\mathcal{M} = (\mathbb{R}, +, *, <, 0, 1)$
- i nodi $\mathcal{V}$ sono dette locazioni (o *control modes*), gli archi $\mathcal{E}$ sono detti control switches
- la variabile T rappresenta il tempo
- $p \in \mathbb{R}^k$

Note alla Definizione

- *Inv*, *Dyn*, *Act*, *Reset* sono insiemi di formule in un linguaggio (al primo ordine) $\mathcal{L}$
  Ad esempio $\mathcal{L} = (+, *, <, 0, 1)$
- le formule vengono valutate su un modello $\mathcal{M}$ di $\mathcal{L}$ sul dominio $\mathbb{R}$
  Ad esempio $\mathcal{M} = (\mathbb{R}, +, *, <, 0, 1)$
- i nodi $\mathcal{V}$ sono dette locazioni (o *control modes*), gli archi $\mathcal{E}$ sono detti control switches
- la variabile T rappresenta il tempo
- $p \in \mathbb{R}^k$

Note alla Definizione

- *Inv*, *Dyn*, *Act*, *Reset* sono insiemi di formule in un linguaggio (al primo ordine) $\mathcal{L}$
  Ad esempio $\mathcal{L} = (+, *, <, 0, 1)$
- le formule vengono valutate su un modello $\mathcal{M}$ di $\mathcal{L}$ sul dominio $\mathbb{R}$
  Ad esempio $\mathcal{M} = (\mathbb{R}, +, *, <, 0, 1)$
- i nodi $\mathcal{V}$ sono dette locazioni (o *control modes*), gli archi $\mathcal{E}$ sono detti control switches
- la variabile T rappresenta il tempo
- $p \in \mathbb{R}^k$

Esempio: Termostato

Example (Termostato)

Consideriamo una stanza riscaldata da un termosifone controllato da un termostato

- Quando il termosifone è acceso la temperatura cresce esponenzialmente nel tempo
- Qualdo il termosifone è spento la temperatura decresce esponenzialmente nel tempo
- Il termostato accende il termosifone quando la temperatura scende al di sotto dei $19C$
- Il termostato spegne il termosifone quando la temperatura sale al di sopra dei $21C$

Esempio: Termostato

Modelliamo l'andamento della temperatura nel tempo con un automa ibrido H avente:

- 2 locazioni ON e OFF
- 2 archi che collegano le due locazioni
- 1 variabile continua Z che rappresenta la temperatura

Esempio: Termostato

$H = \langle Z, Z', \mathcal{V}, \mathcal{E}, Inv, Dyn, Act, Reset \rangle$ tale che:

- Z e Z' sono due variabili
- $\mathcal{V} = \{ON, OFF\}$ e $\mathcal{E} = \{(ON, OFF), (OFF, ON)\}$
- $Inv(ON)[Z] := Z \leq 22$ e $Dyn(ON)[Z, Z', T] := Z' = Z * e^T$
- $Inv(OFF)[Z] := Z \geq 18$ e
  $Dyn(OFF)[Z, Z', T] := Z' = Z / e^T$
- $Act((ON, OFF))[Z] := Z \geq 21$ e
  $Reset((ON, OFF))[Z, Z'] := Z' = Z$
- $Act((OFF, ON))[Z] := Z \leq 19$ e
  $Reset((OFF, ON))[Z, Z'] := Z' = Z$

. . . meglio disegnarlo alla lavagna

Hybrid Automata - Sintassi in Letteratura

T. A. Henzinger

Definition 1.1 [Hybrid automata] [5, 43, 3] A *hybrid automaton* H consists of the following components.

Variables. A finite set $X = \{x_1, \ldots, x_n\}$ of real-numbered variables. The number n is called the *dimension* of H. We write $\dot{X}$ for the set $\{\dot{x}_1, \ldots, \dot{x}_n\}$ of dotted variables (which represent first derivatives during continuous change), and we write X' for the set $\{x'_1, \ldots, x'_n\}$ of primed variables (which represent values at the conclusion of discrete change).

Control graph. A finite directed multigraph (V, E). The vertices in V are called *control modes*. The edges in E are called *control switches*.

Initial, invariant, and flow conditions. Three vertex labeling functions *init*, *inv*, and *flow* that assign to each control mode $v \in V$ three predicates. Each initial condition $init(v)$ is a predicate whose free variables are from X. Each invariant condition $inv(v)$ is a predicate whose free variables are from X. Each flow condition $flow(v)$ is a predicate whose free variables are from $X \cup \dot{X}$.

Jump conditions. An edge labeling function *jump* that assigns to each control switch $e \in E$ a predicate. Each jump condition $jump(e)$ is a predicate whose free variables are from $X \cup X'$.

Events. A finite set Σ of events, and an edge labeling function $event : E \to \Sigma$ that assigns to each control switch an event. □

Hybrid Automata - Sintassi in Letteratura

J. Lygeros et al.

Definition 3.1 (Hybrid Automaton) *A* hybrid automaton H *is a collection* $H = (Q, X, \mathrm{Init}, f, I, E, G, R)$, *where*

- Q *is a set of discrete variables and* Q *is countable;*
- X *is a set of continuous variables;*
- $\mathrm{Init} \subseteq \mathbf{Q} \times \mathbf{X}$ *is a set of initial states;*
- $f : \mathbf{Q} \times \mathbf{X} \to T\mathbf{X}$ *is a vector field;*
- $\mathrm{Inv} : \mathbf{Q} \to P(X) := 2^{\mathbf{X}}$ *assigns to each* $q \in \mathbf{Q}$ *an invariant set;*
- $E \subset \mathbf{Q} \times \mathbf{Q}$ *is a collection of discrete transitions;*
- $G : E \to P(X)$ *assigns to each* $e = (q, q') \in E$ *a guard; and*
- $R : E \times \mathbf{X} \to P(X)$ *assigns to each* $e = (q, q') \in E$ *and* $x \in \mathbf{X}$ *a reset relation.*

Perché . . .

. . . nella nostra definizione non ci sono le equazioni differenziali?

- perchè fanno tanta paura
- perchè non vogliamo incolparle dell'indecidibilità e complessità
- perchè ci vorrebbero altre 200 ore di lezione per parlarne
- perchè ci servirebbero degli analisti per parlarne
- perchè vogliamo risultati/tecniche generali che funzionino per tutte le equazioni risolvibili/approssimabili

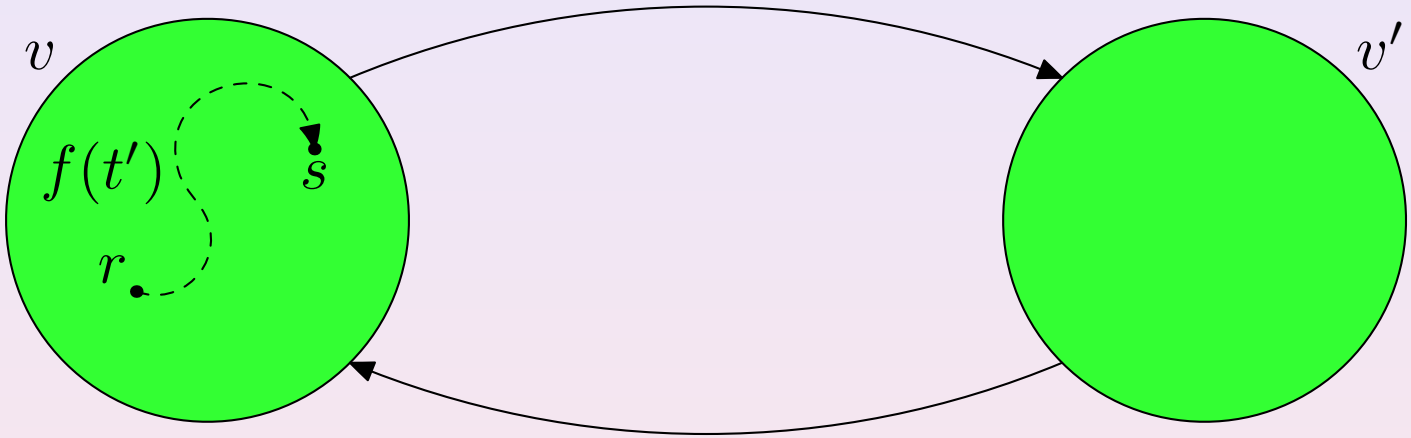
Perché . . .

. . . nella nostra definizione non ci sono le equazioni differenziali?

- perchè fanno tanta paura
- perchè non vogliamo incolparle dell'indecidibilità e complessità
- perchè ci vorrebbero altre 200 ore di lezione per parlarne
- perchè ci servirebbero degli analisti per parlarne
- perchè vogliamo risultati/tecniche generali che funzionino per tutte le equazioni risolvibili/approssimabili

Hybrid Automata - Semantics

$\ell = \langle v, r \rangle$ is *admissible* if $Inv(v)[r]$ holds

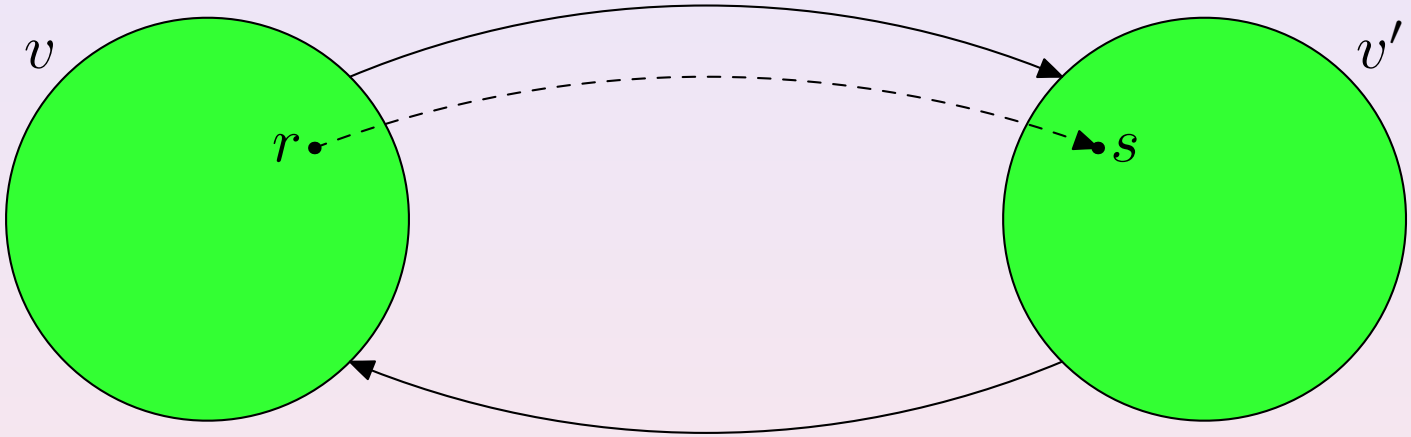


Definition (Continuous Transitions)

$\langle v, r \rangle \xrightarrow{t}_C \langle v, s \rangle \iff$ There exists a continuous function $f : \mathbb{R}^+ \mapsto \mathbb{R}^k$ such that $r = f(0)$, $s = f(t)$ and for each $t' \in [0, t]$ the formulæ $Inv(v)[f(t')]$ and $Dyn(v)[r, f(t'), t']$ hold

Hybrid Automata - Semantics

$\ell = \langle v, r \rangle$ is *admissible* if $Inv(v)[r]$ holds



Definition (Discrete Transitions)

$$\langle v, r \rangle \xrightarrow{\langle v, v' \rangle}_D \langle v', s \rangle \iff \langle v, v' \rangle \in \mathcal{E},\ Inv(v)[r],\ Act(\langle v, v' \rangle)[r],\ Reset(\langle v, v' \rangle)[r, s] \text{ and } Inv(v')[s] \text{ hold}$$

Note alla Definizione

- Abbiamo definito un grafo infinito con due tipi principali di archi

$$(\mathcal{V} \times \mathbb{R}^k, \xrightarrow{\langle \_,\_ \rangle}_D, \stackrel{\_}{\rightarrow}_C)$$

- Potevo essere più precisa?
  Avrei potuto tenere traccia anche della funzione continua f
- Potevo essere meno precisa?
  Avrei potuto considerare un solo tipo di archi

$$\rightarrow = \xrightarrow{\langle \_,\_ \rangle}_D \cup \stackrel{\_}{\rightarrow}_C$$

untimed semantics

Note alla Definizione

- Abbiamo definito un grafo infinito con due tipi principali di archi

$$(\mathcal{V} \times \mathbb{R}^k, \xrightarrow{\langle \_,\_ \rangle}_D, \overset{\_}{\rightarrow}_C)$$

- Potevo essere più precisa?
  Avrei potuto tenere traccia anche della funzione continua f
- Potevo essere meno precisa?
  Avrei potuto considerare un solo tipo di archi

$$\rightarrow = \xrightarrow{\langle \_,\_ \rangle}_D \cup \overset{\_}{\rightarrow}_C$$

untimed semantics

Note alla Definizione

- Abbiamo definito un grafo infinito con due tipi principali di archi

$$(\mathcal{V} \times \mathbb{R}^k, \xrightarrow{\langle \_,\_ \rangle}_D, \overset{-}{\rightarrow}_C)$$

- Potevo essere più precisa?
  Avrei potuto tenere traccia anche della funzione continua f
- Potevo essere meno precisa?
  Avrei potuto considerare un solo tipo di archi

$$\rightarrow = \xrightarrow{\langle \_,\_ \rangle}_D \cup \overset{-}{\rightarrow}_C$$

untimed semantics

Note alla Definizione

- Abbiamo definito un grafo infinito con due tipi principali di archi

$$(\mathcal{V} \times \mathbb{R}^k, \xrightarrow{\langle \_,\_ \rangle}_D, \overset{\_}{\rightarrow}_C)$$

- Potevo essere più precisa?
  Avrei potuto tenere traccia anche della funzione continua f
- Potevo essere meno precisa?
  Avrei potuto considerare un solo tipo di archi

$$\rightarrow = \xrightarrow{\langle \_,\_ \rangle}_D \cup \overset{\_}{\rightarrow}_C$$

untimed semantics

Note alla Definizione

- Abbiamo definito un grafo infinito con due tipi principali di archi

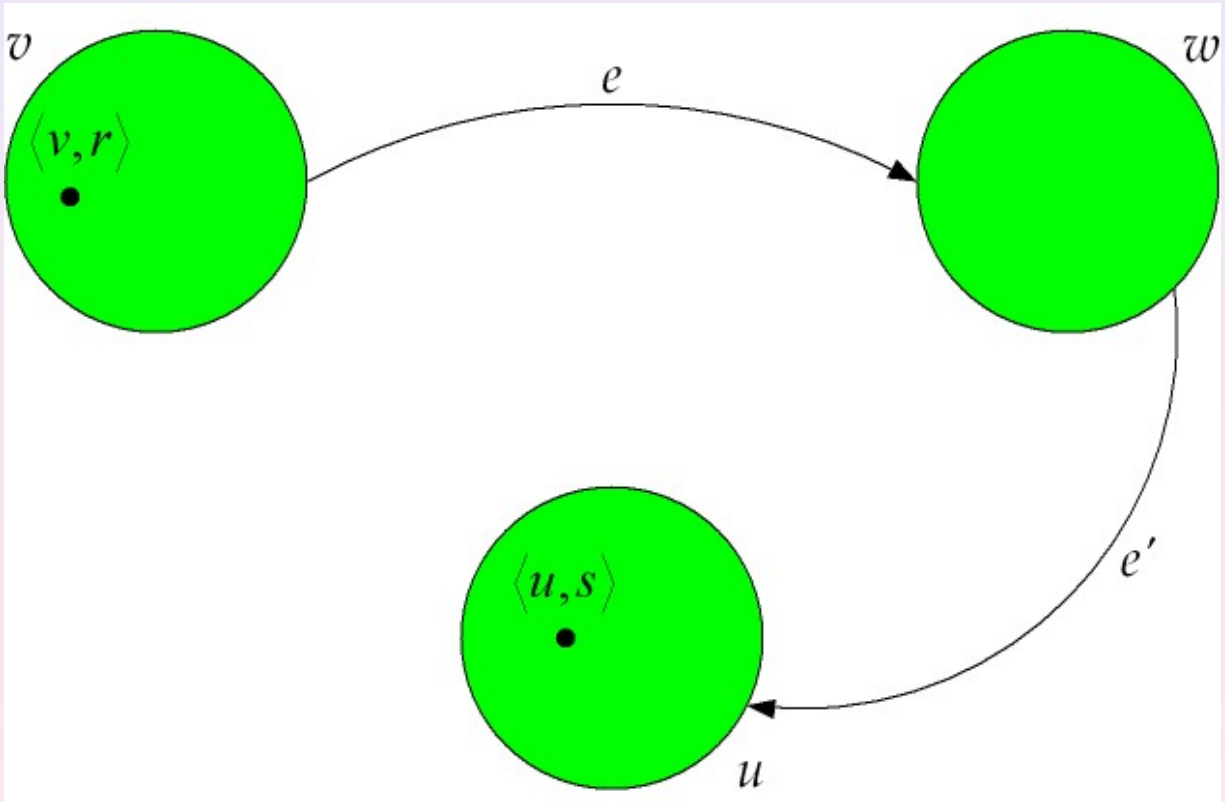
$$(\mathcal{V} \times \mathbb{R}^k, \xrightarrow{\langle \_,\_ \rangle}_D, \overset{\_}{\rightarrow}_C)$$

- Potevo essere più precisa?
  Avrei potuto tenere traccia anche della funzione continua f
- Potevo essere meno precisa?
  Avrei potuto considerare un solo tipo di archi

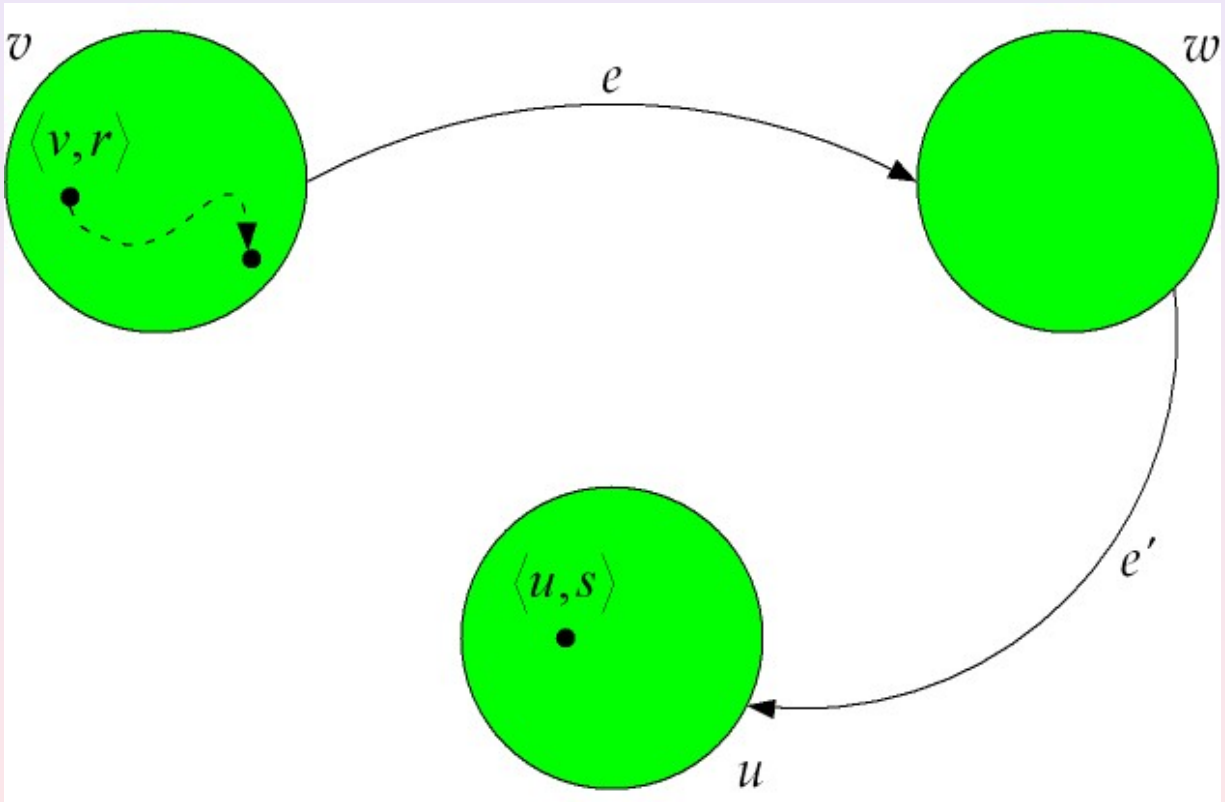
$$\rightarrow = \xrightarrow{\langle \_,\_ \rangle}_D \cup \overset{\_}{\rightarrow}_C$$

untimed semantics

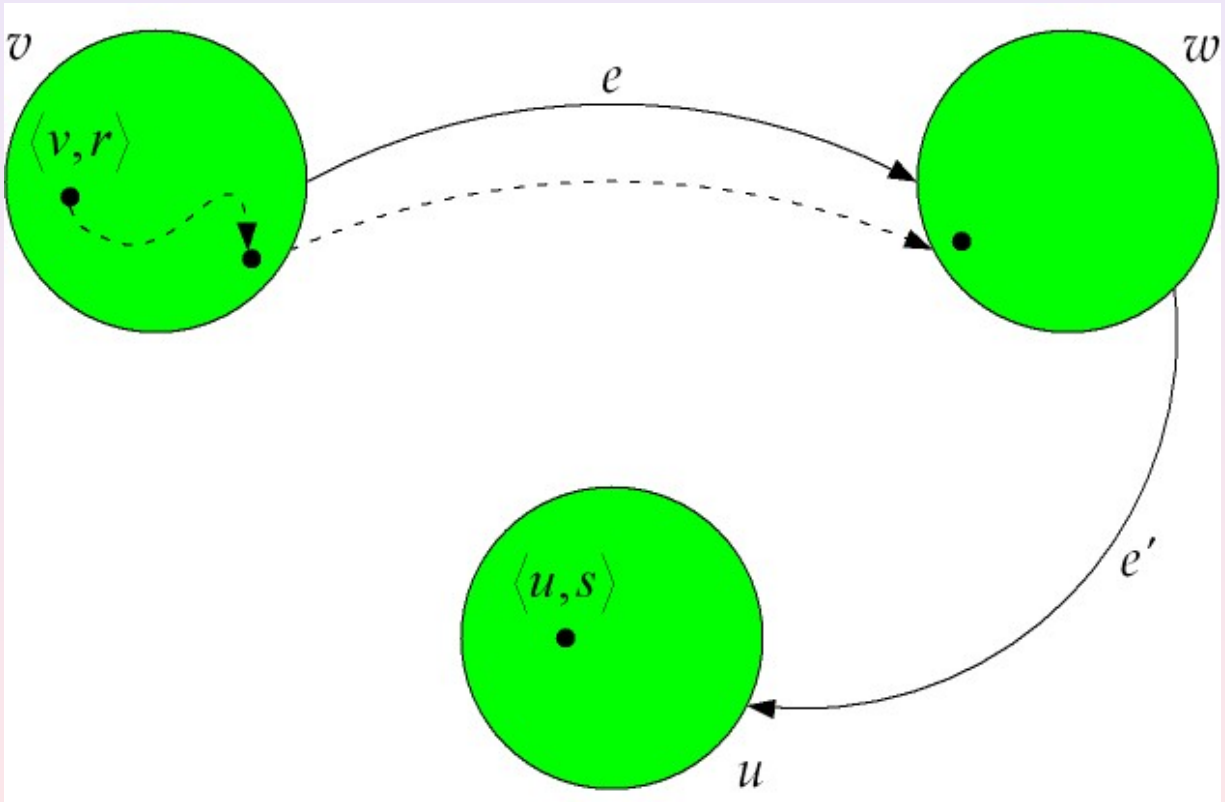
Hybrid Automata - Reachability



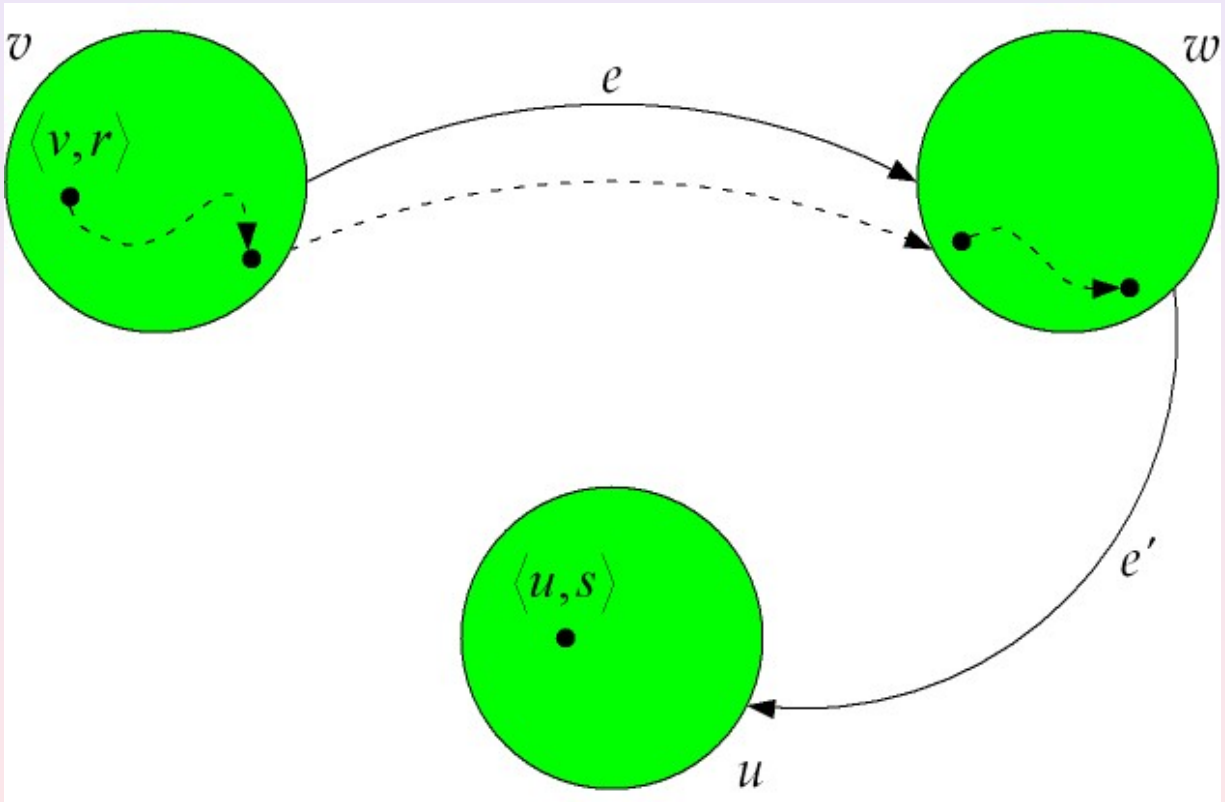
Hybrid Automata - Reachability



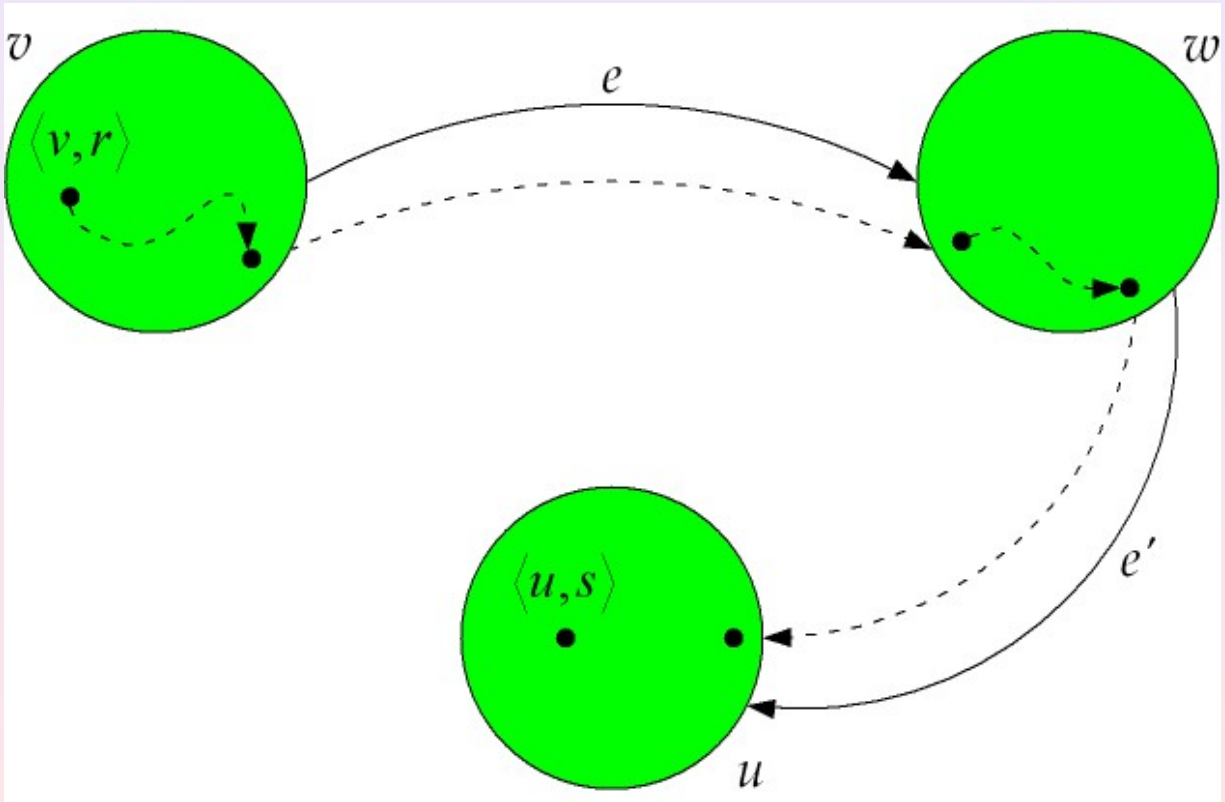
Hybrid Automata - Reachability



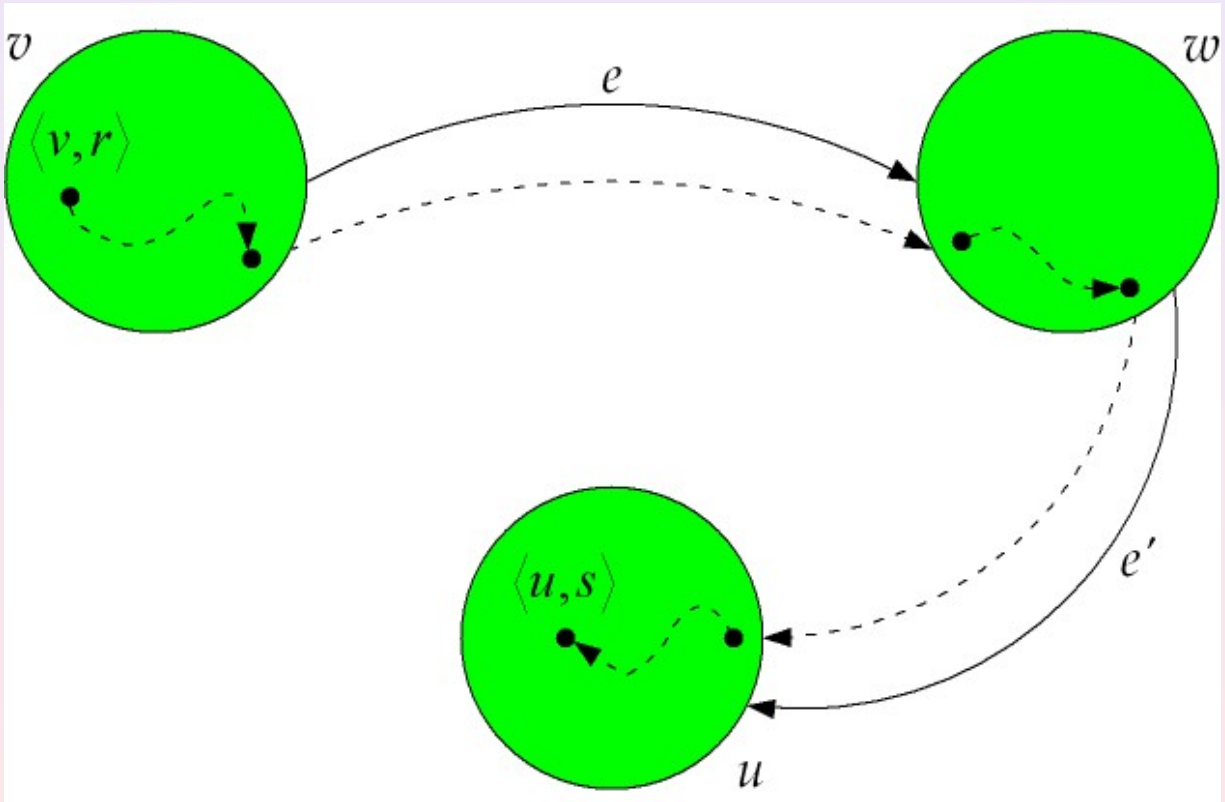
Hybrid Automata - Reachability



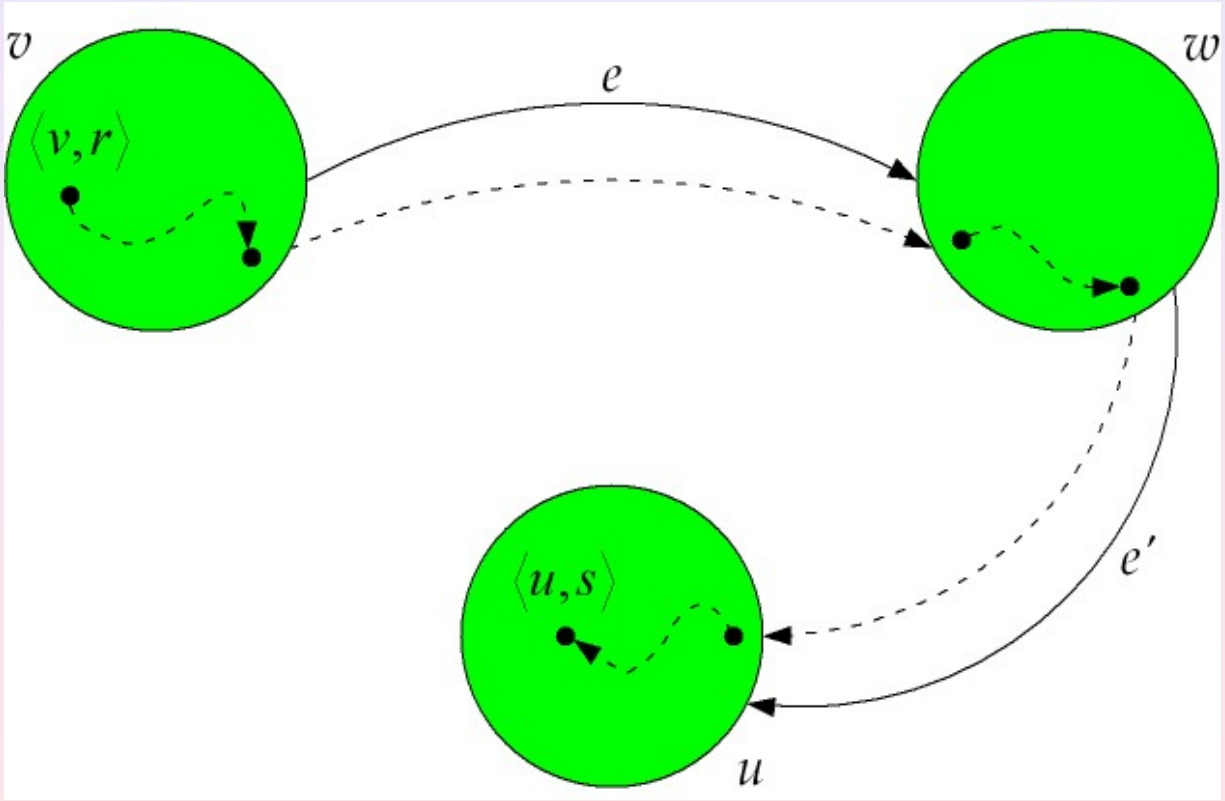
Hybrid Automata - Reachability



Hybrid Automata - Reachability



Hybrid Automata - Reachability



Let $I, F \in \mathbb{R}^k$. Can we reach F from I?

Trace and Reachability

A trace of H is a sequence of admissible states $[\ell_0, \ell_1, \ldots, \ell_i, \ldots, \ell_n]$ such that $\ell_{i-1} \rightarrow \ell_i$ holds $\forall i \in [1, n]$

Definition (Reachability)

The automaton H reaches $s \in \mathbb{R}^k$ from $r \in \mathbb{R}^k$ if there exists a trace $tr = [\ell_0, \ldots, \ell_n]$ of H such that $\ell_0 = \langle v, r \rangle$ and $\ell_n = \langle u, s \rangle$, for some $v, u \in \mathcal{V}$

Definition (Reachability Problem)

Given an automaton H, a set of starting points $I \subseteq \mathbb{R}^k$, and a set of ending points $F \subseteq \mathbb{R}^k$ we wish to decide whether there exists a point in I from which a point in F is reachable

Mille fonti di Non Determinismo

Gli Hybrid Automata sono non-deterministici in quanto:

- Locazioni distinte possono parzialmente condividere gli invarianti
- Da ogni stato ammissibile possono partire diverse traiettorie continue
- Ci possono essere archi che portano in locazioni distinte ma che condividono parzialmente le activation
- Le attivazioni non sono necessariamente sulle frontiere degli invarianti
- I reset non sono necessariamente deterministici

Mille fonti di Non Determinismo

Gli Hybrid Automata sono non-deterministici in quanto:

- Locazioni distinte possono parzialmente condividere gli invarianti
- Da ogni stato ammissibile possono partire diverse traiettorie continue
- Ci possono essere archi che portano in locazioni distinte ma che condividono parzialmente le activation
- Le attivazioni non sono necessariamente sulle frontiere degli invarianti
- I reset non sono necessariamente deterministici

Mille fonti di Non Determinismo

Gli Hybrid Automata sono non-deterministici in quanto:

- Locazioni distinte possono parzialmente condividere gli invarianti
- Da ogni stato ammissibile possono partire diverse traiettorie continue
- Ci possono essere archi che portano in locazioni distinte ma che condividono parzialmente le activation
- Le attivazioni non sono necessariamente sulle frontiere degli invarianti
- I reset non sono necessariamente deterministici

Mille fonti di Non Determinismo

Gli Hybrid Automata sono non-deterministici in quanto:

- Locazioni distinte possono parzialmente condividere gli invarianti
- Da ogni stato ammissibile possono partire diverse traiettorie continue
- Ci possono essere archi che portano in locazioni distinte ma che condividono parzialmente le activation
- Le attivazioni non sono necessariamente sulle frontiere degli invarianti
- I reset non sono necessariamente deterministici

Mille fonti di Non Determinismo

Gli Hybrid Automata sono non-deterministici in quanto:

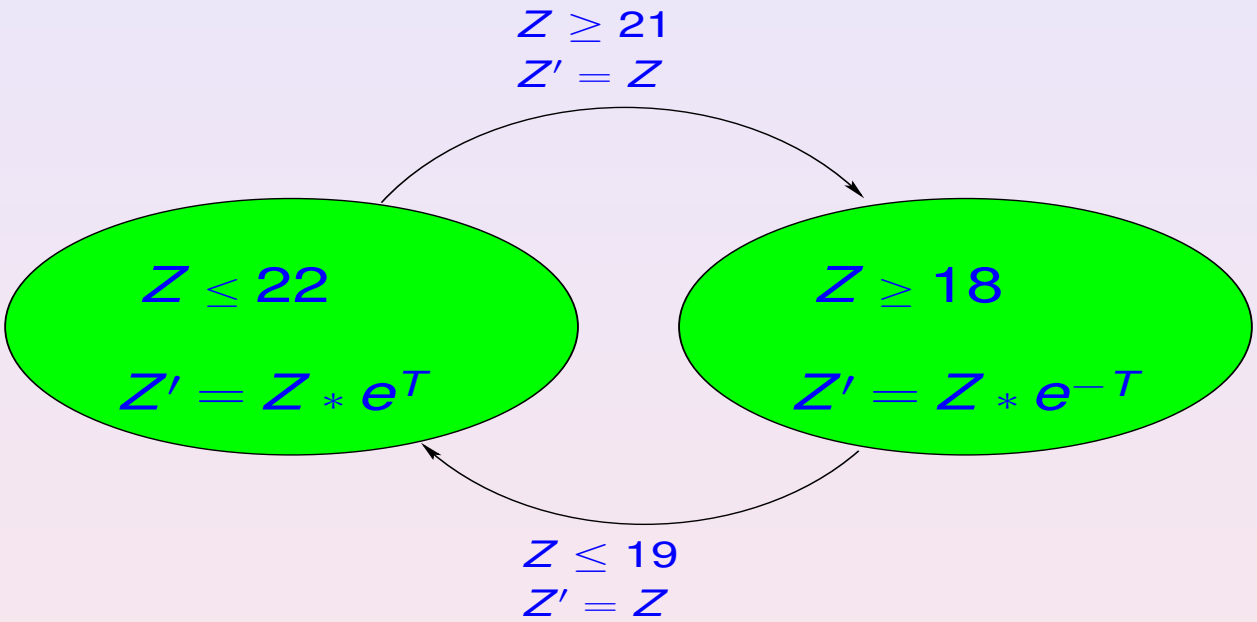
- Locazioni distinte possono parzialmente condividere gli invarianti
- Da ogni stato ammissibile possono partire diverse traiettorie continue
- Ci possono essere archi che portano in locazioni distinte ma che condividono parzialmente le activation
- Le attivazioni non sono necessariamente sulle frontiere degli invarianti
- I reset non sono necessariamente deterministici

Mille fonti di Non Determinismo

Gli Hybrid Automata sono non-deterministici in quanto:

- Locazioni distinte possono parzialmente condividere gli invarianti
- Da ogni stato ammissibile possono partire diverse traiettorie continue
- Ci possono essere archi che portano in locazioni distinte ma che condividono parzialmente le activation
- Le attivazioni non sono necessariamente sulle frontiere degli invarianti
- I reset non sono necessariamente deterministici

Esempio: Termostato



- $\langle ON, 15\rangle \xrightarrow{0.1}_C \langle ON, 16.57\rangle \xrightarrow{0.25}_C \langle ON, 21.28\rangle \xrightarrow{\langle ON,OFF\rangle}_D \langle OFF, 21.28\rangle \ldots$
- $\langle ON, 15\rangle \xrightarrow{0.35}_C \langle ON, 21.28\rangle \xrightarrow{\langle ON,OFF\rangle}_D \langle OFF, 21.28\rangle \ldots$
- $\langle OFF, 18.5\rangle \xrightarrow{\langle OFF,ON\rangle}_D \langle ON, 18.5\rangle \ldots$
- $\langle OFF, 18.5\rangle \xrightarrow{0.01}_C \langle OFF, 18.31\rangle \xrightarrow{\langle OFF,ON\rangle}_D \langle ON, 18.31\rangle \ldots$

Esempio: Termostato

Osserviamo che:

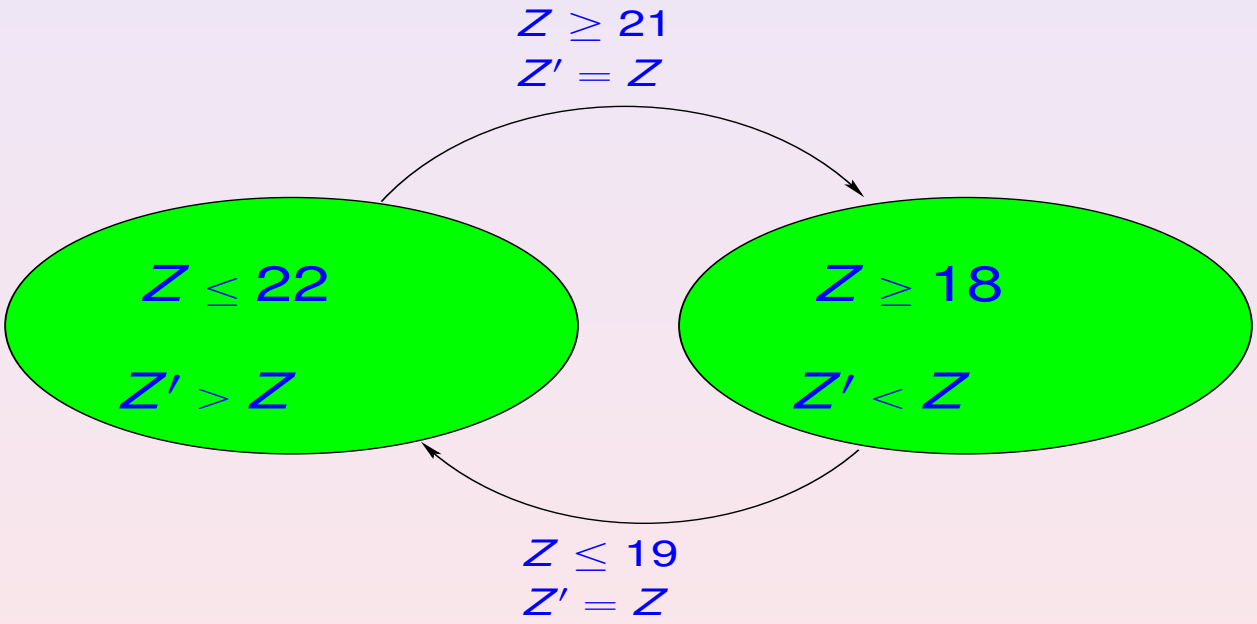
- Da ogni punto partono infinite traiettorie
- Alcune sono sostanzialmente “equivalenti”
- Altre no!

Esempio: Termostato

Che modello avrei potuto costruire con meno informazioni?

Esempio: Termostato

Che modello avrei potuto costruire con meno informazioni?



Questo ha più tracce del precedente!

Hybrid Automata - Semantica in Letteratura

Henzinger:

- Introduce una timed ed una un-timed semantics
- Etichetta le transizioni discrete con gli eventi
- Scarta le zeno-traces
- Parla di *machine-closed* systems e *nonzeno* automata

Hybrid Automata - Semantica in Letteratura

Lygeros et al. pag. 39

- Introducono la nozione di hybrid time set (sequenza di intervalli temporali)
- Su questa nozione basano la definizione di transizioni discrete e continue
- Distinguono tra executions: Finite, Infinite, Zeno, Maximal
- Studiano condizioni al contorno

Altri Problemi

Henzinger introduce altri due problemi:

- emptiness problem: l'automa ha almeno una traccia divergente inizializzata
- (finitary) {timed} trace inclusion problem: dati due automi ibridi determinare se ogni traccia {temporizzata} (finita) del primo è anche una traccia del secondo.

Sono problemi legati all'analisi di proprietà di safety e liveness
Henzinger dice:

- il reachability problem può essere ridotto al trace inclusion problem
- l'emptiness problem può essere ridotto al timed trace inclusion problem

Altri Problemi

Henzinger introduce altri due problemi:

- emptiness problem: l'automa ha almeno una traccia divergente inizializzata
- (finitary) {timed} trace inclusion problem: dati due automi ibridi determinare se ogni traccia {temporizzata} (finita) del primo è anche una traccia del secondo.

Sono problemi legati all'analisi di proprietà di safety e liveness

Henzinger dice:

- il reachability problem può essere ridotto al trace inclusion problem
- l'emptiness problem può essere ridotto al timed trace inclusion problem

Altri Problemi

Henzinger introduce altri due problemi:

- emptiness problem: l'automa ha almeno una traccia divergente inizializzata
- (finitary) {timed} trace inclusion problem: dati due automi ibridi determinare se ogni traccia {temporizzata} (finita) del primo è anche una traccia del secondo.

Sono problemi legati all'analisi di proprietà di safety e liveness Henzinger dice:

- il reachability problem può essere ridotto al trace inclusion problem
- l'emptiness problem può essere ridotto al timed trace inclusion problem

I nostri autori

Thomas A. Henzinger is Professor of Computer and Communication Sciences at EPFL in Lausanne, Switzerland, and Adjunct Professor of Electrical Engineering and Computer Sciences at the University of California, Berkeley. He holds a Dipl. Ing. degree in Computer Science from Kepler University in Linz, Austria, an M.S. degree in Computer and Information Sciences from the University of Delaware, and a Ph.D. degree in Computer Science from Stanford University (1991) with Manna.

I nostri autori

Rajeev Alur is Professor of Computer and Information Science at University of Pennsylvania. He obtained his bachelor's degree in computer science from Indian Institute of Technology at Kanpur in 1987, and PhD in computer science from Stanford University in 1991 with Dill.

I nostri autori

David Dill is a Professor of Computer Science and, by courtesy, Electrical Engineering at Stanford University. He has been on the faculty at Stanford since 1987. He has an S.B. in Electrical Engineering and Computer Science from Massachusetts Institute of Technology (1979), and an M.S and Ph.D. from Carnegie-Mellon University (1982 and 1987) with Clarke.

I nostri autori

John Lygeros has been a Professor of Computation and Control at the Swiss Federal Institute of Technology (ETH) Zurich, Switzerland, since July 2006 and the Head of the Automatic Control Laboratory since January 2009. He completed a B.Eng. degree in electrical engineering in 1990 and an M.Sc. degree in Systems Control in 1991, both at Imperial College of Science Technology and Medicine, London, U.K.. In 1996 he obtained a Ph.D. degree from the Electrical Engineering and Computer Sciences Department, University of California, Berkeley with Sastry.