Automi Ibridi

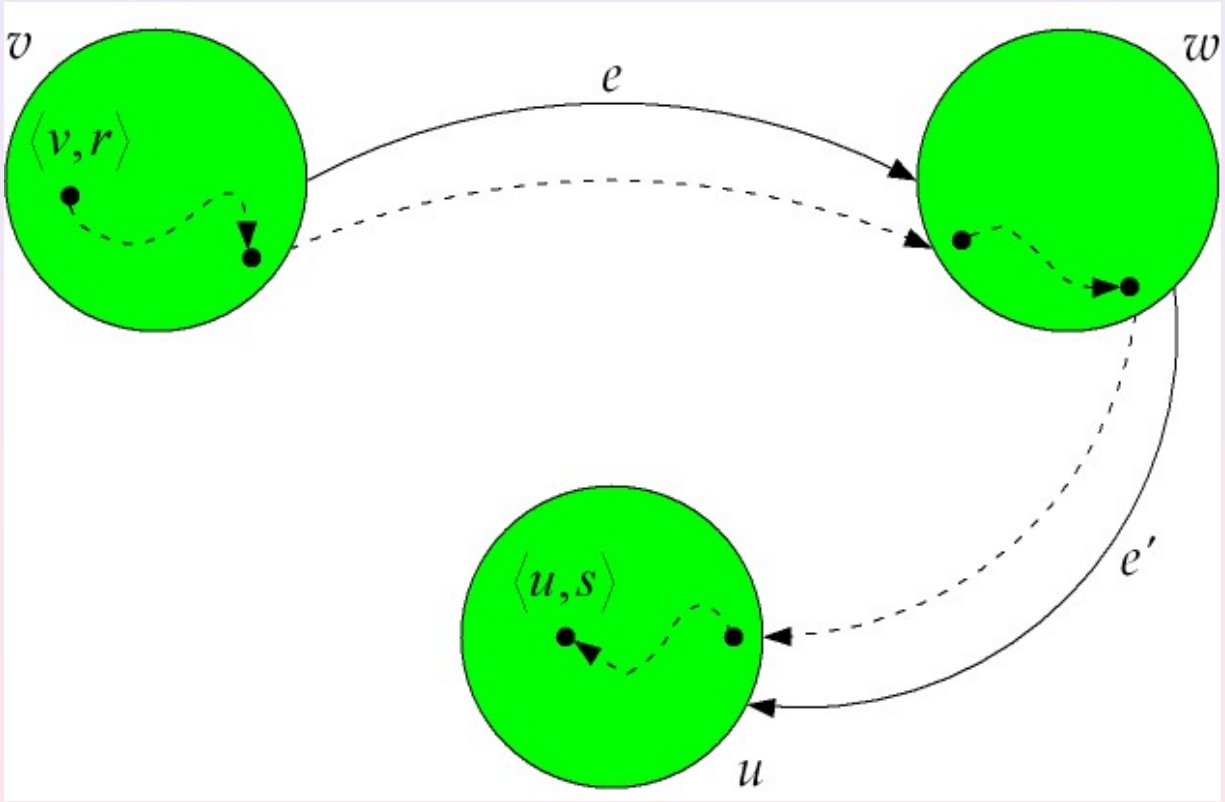
Carla Piazza[1]

[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

Hybrid Automata - Reachability



Siano $I, F \in \mathbb{R}^k$. Possiamo raggiungere F da I?

Modelli ed Indecidibilità

Quale tecnica usereste per dimostrare che un problema è indecidibile?

Modelli ed Indecidibilità

Quale tecnica usereste per dimostrare che un problema è indecidibile?

Cerchiamo un modello semplice su cui esista un problema indecidibile

Modelli ed Indecidibilità

Definition (2CM)

Una 2-counter machine (2CM) è costituita da:

- due registri c_0 ed c_1, detti contatori, adatti a contenere numeri naturali;
- un program counter *pc*
- una lista finita di istruzioni $[\ell_1, \ldots, \ell_n]$ scelte nell'insieme:
  $INCR(i)$ $DECR(i)$ $JZERO(i, j)$ $HALT$

Theorem (Minsky)

Ogni Macchina di Turing può essere simulata con una 2CM, con una opportuna codifica di input e output

Corollary

Il problema dell'arresto per le 2CM è indecidibile

Modelli ed Indecidibilità

Definition (2CM)

Una 2-counter machine (2CM) è costituita da:

- due registri c_0 ed c_1, detti contatori, adatti a contenere numeri naturali;
- un program counter *pc*
- una lista finita di istruzioni $[\ell_1, \ldots, \ell_n]$ scelte nell'insieme:
  $INCR(i)$ $DECR(i)$ $JZERO(i, j)$ $HALT$

Theorem (Minsky)

Ogni Macchina di Turing può essere simulata con una 2CM, con una opportuna codifica di input e output

Corollary

Il problema dell'arresto per le 2CM è indecidibile

Modelli ed Indecidibilità

Definition (2CM)

Una 2-counter machine (2CM) è costituita da:

- due registri c_0 ed c_1, detti contatori, adatti a contenere numeri naturali;
- un program counter *pc*
- una lista finita di istruzioni $[\ell_1, \ldots, \ell_n]$ scelte nell'insieme:
  $INCR(i)$ $DECR(i)$ $JZERO(i, j)$ $HALT$

Theorem (Minsky)

Ogni Macchina di Turing può essere simulata con una 2CM, con una opportuna codifica di input e output

Corollary

Il problema dell'arresto per le 2CM è indecidibile

Primo Tentativo di Codifica: Intuizione

- uso una variabile continua per ogni contatore

$$c_i = n \qquad \text{iff} \qquad Z_i = n$$

- farò in modo che assumano solo valori interi positivi
- uso una locazione discreta per ogni istruzione
- uso gli archi per passare il controllo da un'istruzione all'altra

Primo Tentativo di Codifica: Intuizione

- uso una variabile continua per ogni contatore

$$c_i = n \qquad \text{iff} \qquad Z_i = n$$

- farò in modo che assumano solo valori interi positivi
- uso una locazione discreta per ogni istruzione
- uso gli archi per passare il controllo da un'istruzione all'altra

Primo Tentativo di Codifica: Intuizione

- uso una variabile continua per ogni contatore

$$c_i = n \qquad \text{iff} \qquad Z_i = n$$

- farò in modo che assumano solo valori interi positivi
- uso una locazione discreta per ogni istruzione
- uso gli archi per passare il controllo da un'istruzione all'altra

Primo Tentativo di Codifica

Sia $M = (c_0, c_1, pc, [\ell_1, \ldots, \ell_n])$ una *2CM*

Considero un automa ibrido H_M avente:

- n locazioni discrete $L_1, \ldots, L_n$
- Z_0, Z_1 due variabili continue
- in ogni locazione le dinamiche sono banali

$$Z_0' = Z_0 \wedge Z_1' = Z_1$$

- $Act((L_i, L_j))[Z_0, Z_1, T]$ e $Reset((L_i, L_j))[Z_0, Z_1, Z_0', Z_1', T]$ implementano l'istruzione ℓ_i ed il passaggio all'istruzione ℓ_j

Primo Tentativo di Codifica

Sia $M = (c_0, c_1, pc, [\ell_1, \ldots, \ell_n])$ una *2CM*

Considero un automa ibrido H_M avente:

- n locazioni discrete $L_1, \ldots, L_n$
- Z_0, Z_1 due variabili continue
- in ogni locazione le dinamiche sono banali

$$Z_0' = Z_0 \wedge Z_1' = Z_1$$

- $Act((L_i, L_j))[Z_0, Z_1, T]$ e $Reset((L_i, L_j))[Z_0, Z_1, Z_0', Z_1', T]$ implementano l'istruzione ℓ_i ed il passaggio all'istruzione ℓ_j

Primo Tentativo di Codifica

Sia $M = (c_0, c_1, pc, [\ell_1, \ldots, \ell_n])$ una *2CM*

Considero un automa ibrido H_M avente:

- n locazioni discrete $L_1, \ldots, L_n$
- Z_0, Z_1 due variabili continue
- in ogni locazione le dinamiche sono banali

$$Z_0' = Z_0 \wedge Z_1' = Z_1$$

- $Act((L_i, L_j))[Z_0, Z_1, T]$ e $Reset((L_i, L_j))[Z_0, Z_1, Z_0', Z_1', T]$ implementano l'istruzione ℓ_i ed il passaggio all'istruzione ℓ_j

Primo Tentativo di Codifica

Sia $M = (c_0, c_1, pc, [\ell_1, \ldots, \ell_n])$ una *2CM*

Considero un automa ibrido H_M avente:

- n locazioni discrete $L_1, \ldots, L_n$
- Z_0, Z_1 due variabili continue
- in ogni locazione le dinamiche sono banali

$$Z_0' = Z_0 \wedge Z_1' = Z_1$$

- $Act((L_i, L_j))[Z_0, Z_1, T]$ e $Reset((L_i, L_j))[Z_0, Z_1, Z_0', Z_1', T]$ implementano l'istruzione ℓ_i ed il passaggio all'istruzione ℓ_j

Primo Tentativo di Codifica

Sia $M = (c_0, c_1, pc, [\ell_1, \ldots, \ell_n])$ una *2CM*

Considero un automa ibrido H_M avente:

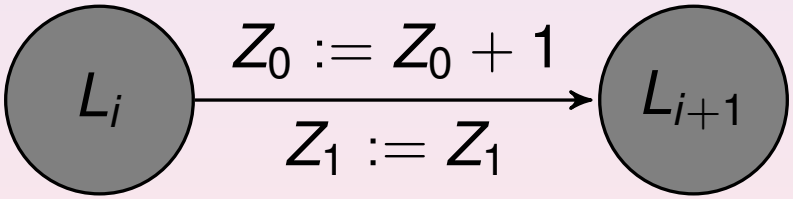
- n locazioni discrete $L_1, \ldots, L_n$
- Z_0, Z_1 due variabili continue
- in ogni locazione le dinamiche sono banali

$$Z_0' = Z_0 \wedge Z_1' = Z_1$$

- $Act((L_i, L_j))[Z_0, Z_1, T]$ e $Reset((L_i, L_j))[Z_0, Z_1, Z_0', Z_1', T]$ implementano l'istruzione ℓ_i ed il passaggio all'istruzione ℓ_j

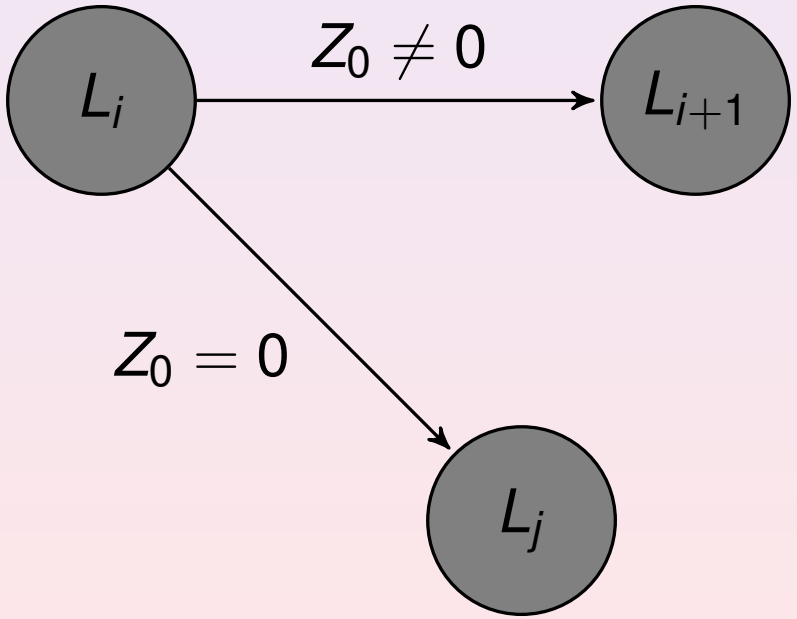
La Codifica delle Istruzioni

$\ell_i : INCR(0)$



La Codifica delle Istruzioni

$\ell_i : JZERO(0, j)$



Il Bicchiere è . . .

- . . . mezzo vuoto: abbiamo usato dinamiche banali. Quindi non possiamo neanche incolparle!
- . . . mezzo pieno:
  - abbiamo usato Invarianti illimitate
  - abbiamo usato Activation e Reset potenti

Domanda:
Cosa succede se ammettiamo dinamiche non banali, ma aggiungiamo vincoli su Invarianti, Activation, Reset?

[Anonimo] Non è questione di bicchiere mezzo pieno o mezzo vuoto. È questione di bicchiere rotto!

Il Bicchiere è . . .

- . . . mezzo vuoto: abbiamo usato dinamiche banali. Quindi non possiamo neanche incolparle!
- . . . mezzo pieno:
  - abbiamo usato Invarianti illimitate
  - abbiamo usato Activation e Reset potenti

Domanda

Cosa succede se ammettiamo dinamiche non banali, ma aggiungiamo vincoli su Invarianti, Activation, Reset?

[Anonimo] Non è questione di bicchiere mezzo pieno o mezzo vuoto. È questione di bicchiere rotto!

Il Bicchiere è . . .

- . . . mezzo vuoto: abbiamo usato dinamiche banali. Quindi non possiamo neanche incolparle!
- . . . mezzo pieno:
  - abbiamo usato Invarianti illimitate
  - abbiamo usato Activation e Reset potenti

Domanda

Cosa succede se ammettiamo dinamiche non banali, ma aggiungiamo vincoli su Invarianti, Activation, Reset?

[Anonimo] Non è questione di bicchiere mezzo pieno o mezzo vuoto. È questione di bicchiere rotto!

Il Bicchiere è . . .

- . . . mezzo vuoto: abbiamo usato dinamiche banali. Quindi non possiamo neanche incolparle!
- . . . mezzo pieno:
  - abbiamo usato Invarianti illimitate
  - abbiamo usato Activation e Reset potenti

Domanda

Cosa succede se ammettiamo dinamiche non banali, ma aggiungiamo vincoli su Invarianti, Activation, Reset?

[Anonimo] Non è questione di bicchiere mezzo pieno o mezzo vuoto. È questione di bicchiere rotto!

Il Bicchiere è . . .

- . . . mezzo vuoto: abbiamo usato dinamiche banali. Quindi non possiamo neanche incolparle!
- . . . mezzo pieno:
  - abbiamo usato Invarianti illimitate
  - abbiamo usato Activation e Reset potenti

Domanda

Cosa succede se ammettiamo dinamiche non banali, ma aggiungiamo vincoli su Invarianti, Activation, Reset?

[Anonimo] Non è questione di bicchiere mezzo pieno o mezzo vuoto. È questione di bicchiere rotto!

Invarianti Limitate

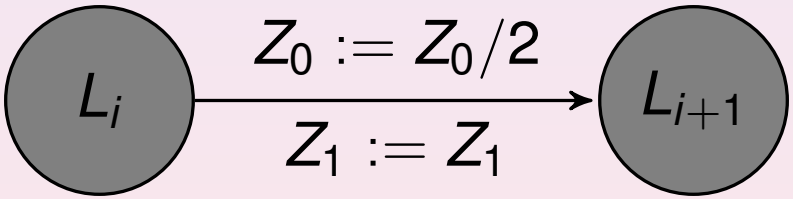
- Non è realistico assumere di poter lavorare con interi di grandezza arbitraria: in ogni sistema fisico, biologico o ingegneristico le grandezze coinvolte sono limitate superiormente
- Purtroppo la situazione non migliora: posso imporre che le variabili continue restino sempre in $[0, 1]$ (oppure $[0, 2]$) e codificare n con $1/2^n$ (oppure $2/2^n$)

Invarianti Limitate

- Non è realistico assumere di poter lavorare con interi di grandezza arbitraria: in ogni sistema fisico, biologico o ingegneristico le grandezze coinvolte sono limitate superiormente
- Purtroppo la situazione non migliora: posso imporre che le variabili continue restino sempre in $[0, 1]$ (oppure $[0, 2]$) e codificare n con $1/2^n$ (oppure $2/2^n$)

L'Incremento con Invarianti Limitate

$\ell_i : INCR(0)$



Più Dinamica, Meno Activation e Reset

Bibliografia

- The Algorithmic Analysis of Hybrid Systems. Alur, Courcobetis, Halbwachs, Henzinger, Ho, Nicolin, Olivero, Sifakis, Yovine. Theoretical Computer Science 138(1), 1995.
- Undecidability Results for Hybrid Systems. Henzinger and Kopke. Unpublished.
- What's Decidable about Hybrid Automata? Henzinger, Kopke, Puri, Varaiya. Journal of Computer and System Sciences 57(1), 1998.

Alcune Definizioni

Consideriamo le possibili Dinamiche

Una variabile continua Z è detta:

- Clock se $Z' = Z + T$ in ogni locazione, ovvero $\dot{Z} = 1$
- Skewed Clock se $Z' = Z + gT$ con $g \notin \{0, 1\}$, ovvero $\dot{Z} = g$
- Memory Cell se $Z' = Z$ con $g \notin \{0, 1\}$, ovvero $\dot{Z} = 0$
- Stopwatch se $Z' = Z$ oppure $Z' = Z + T$, ovvero $\dot{Z} = 1$ oppure $\dot{Z} = 0$
- Two Slope se $Z' = Z + gT$ oppure $Z' = Z + hT$, ovvero $\dot{Z} = g$ oppure $\dot{Z} = h$, con $g, h > 0$
- Linear se per ogni locazione esistono a e b tali che $a \leq \dot{Z} \leq b$

Alcune Definizioni

Consideriamo le possibili Activation

- Weak Predicate ($\mathcal{W}$): congiunzione di vincoli del tipo $Z_i \leq n$ oppure $Z_i \geq n$
- $\mathcal{W}_=$: vincoli di $\mathcal{W}$ e vincoli del tipo $Z_i = Z_j$ (confronto tra variabili)
- $\mathcal{W}_{=2}$: vincoli di $\mathcal{W}_=$ in cui al posto di Z_i può comparire $2 * Z_i$
- $\mathcal{W}_{=+}$: vincoli di $\mathcal{W}_=$ in cui al posto di Z_i può comparire $Z_i + Z_j$

Alcune Definizioni

Consideriamo i possibili Reset

- Pass: $Z_i' := Z_i$ (identità)
- Reset: $Z_i' := 0$
- Switch: $Z_i' := Z_j$ con $i \neq j$

Indichiamo con:

- $\mathcal{R}$: reset Pass e Reset
- $\mathcal{V}$: reset Pass, Reset e Switch

2-rate Timed Automata

Theorem

Il problema della raggiungibilità per automi con:

- *variabili di tipo clock*
- *una variabile di tipo skewed clock*
- *activation in* $\mathcal{W}_{=}$
- *reset in* $\mathcal{R}$

è indecidibile

Alla Base dell'Indecidibilità

Intuitivamente

- codifico n con $1/2^n$ (oppure $2/2^n$)
- uso un clock y per accertarmi di spendere in ogni locazione esattamente un unità di tempo
- uso un clock x_i per ogni contatore
- uso un clock z ed uno skewed clock $\widetilde{z}$ con slope $g = 2$ per incrementare/decrementare i contatori

- è facile implementare $JZERO(i, j)$
- le istruzioni più complicate sono $INCR(i)$ e $DECR(i)$
- per implementare $INCR(0)$ devo tenere ferma x_1 e dimezzare x_0

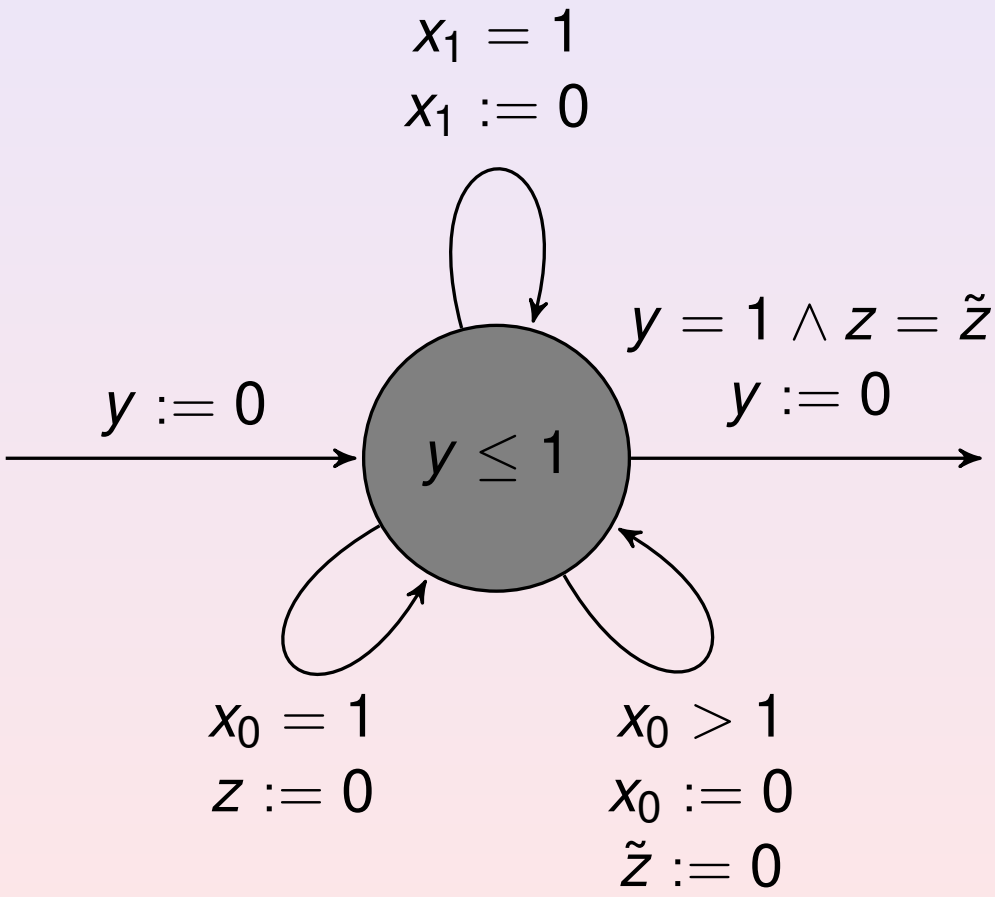
Alla Base dell'Indecidibilità

Intuitivamente

- codifico n con $1/2^n$ (oppure $2/2^n$)
- uso un clock y per accertarmi di spendere in ogni locazione esattamente un unità di tempo
- uso un clock x_i per ogni contatore
- uso un clock z ed uno skewed clock $\widetilde{z}$ con slope $g = 2$ per incrementare/decrementare i contatori

- è facile implementare $JZERO(i, j)$
- le istruzioni più complicate sono $INCR(i)$ e $DECR(i)$
- per implementare $INCR(0)$ devo tenere ferma x_1 e dimezzare x_0

L'Incremento $INCR(0)$



Generalizzazioni

Utilizzando solo vincoli in $\mathcal{W}$ e reset in $\mathcal{R}$ posso:

Lemma (Wrapping)

spendere un'istante temporale in una locazione e lasciare invariati i valori di alcuni (skewed) clocks

Lemma (Equality)

spendere un'istante temporale in una locazione e testare se due (skewed) clocks hanno lo stesso valore (senza farli evolvere)

Lemma (Assignment)

spendere un'istante temporale in una locazione e assegnare ad uno (skewed) clock il valore di un altro

Generalizzazioni

Utilizzando solo vincoli in $\mathcal{W}$ e reset in $\mathcal{R}$ posso:

Lemma (Wrapping)

spendere un'istante temporale in una locazione e lasciare invariati i valori di alcuni (skewed) clocks

Lemma (Equality)

spendere un'istante temporale in una locazione e testare se due (skewed) clocks hanno lo stesso valore (senza farli evolvere)

Lemma (Assignment)

spendere un'istante temporale in una locazione e assegnare ad uno (skewed) clock il valore di un altro

Generalizzazioni

Utilizzando solo vincoli in $\mathcal{W}$ e reset in $\mathcal{R}$ posso:

Lemma (Wrapping)

spendere un'istante temporale in una locazione e lasciare invariati i valori di alcuni (skewed) clocks

Lemma (Equality)

spendere un'istante temporale in una locazione e testare se due (skewed) clocks hanno lo stesso valore (senza farli evolvere)

Lemma (Assignment)

spendere un'istante temporale in una locazione e assegnare ad uno (skewed) clock il valore di un altro

Generalizzazioni

Utilizzando solo vincoli in $\mathcal{W}$ e reset in $\mathcal{R}$ posso:

Lemma (Wrapping)

spendere un'istante temporale in una locazione e lasciare invariati i valori di alcuni (skewed) clocks

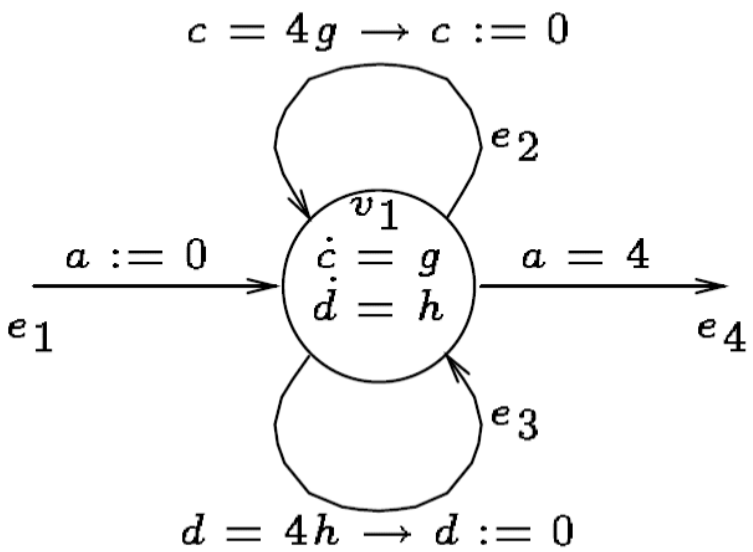
Lemma (Equality)

spendere un'istante temporale in una locazione e testare se due (skewed) clocks hanno lo stesso valore (senza farli evolvere)

Lemma (Assignment)

spendere un'istante temporale in una locazione e assegnare ad uno (skewed) clock il valore di un altro

Wrapping


$c = 4g \rightarrow c := 0$
e_2
v_1
$\dot{c} = g$
$\dot{d} = h$
$a := 0$
$a = 4$
e_1
e_4
e_3
$d = 4h \rightarrow d := 0$

Non bastano i Lemmi

Nota

Nei lemmi non si usano vincoli di confronto
Sono “indispensabili” per codificare incremento/decremento

Indecidibili

Hybrid Variables other than Clocks	Predicates	Assignments
none	$\mathcal{W}_{=2}$	$\mathcal{R}$
none	$\mathcal{W}_{=+}$	$\mathcal{R}$
one skewed clock	$\mathcal{W}_{=}$	$\mathcal{R}$
one skewed clock	$\mathcal{W}$	$\mathcal{V}$
one stopwatch	$\mathcal{W}$	$\mathcal{R}$
one two-slope variable	$\mathcal{W}$	$\mathcal{R}$
one memory cell	$\mathcal{W}$	$\mathcal{V}$
three linear variables with triangular diff. incl.	$\mathcal{W}$	$\mathcal{R}$

Figure 9: Undecidable reachability problems