

Phaver

Laboratorio di Sistemi in Tempo Reale

Corso di Laurea in Informatica Multimediale

20 Novembre 2008

- 1 Una breve introduzione a PhaVer
- 2 Un primo esempio
- 3 Esercizio

PhaVer è un pacchetto per la verifica di automi ibridi

- sviluppato da Goran Frehse come successore di HyTech
- modella automi con dinamica lineare
- interfaccia di tipo testuale, simile ad HyTech
- supporta la composizione di più automi
- verifica di proprietà di sicurezza e raggiungibilità

Programma e documentazione:

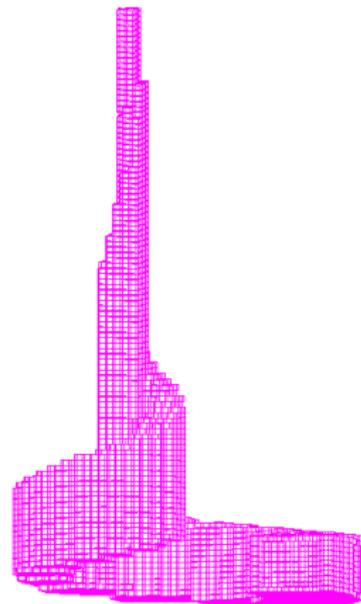
www-verimag.imag.fr/~frehse/phaver_web/index.html

Il risultato può essere molto complesso!

$$\dot{x}_1 = 1.0x_1 + 4.0x_2$$

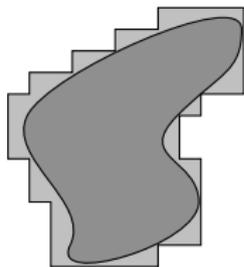
$$\dot{x}_2 = 4.0x_1 + 1.0x_2$$

$$\dot{x}_3 = 0.5x_3$$

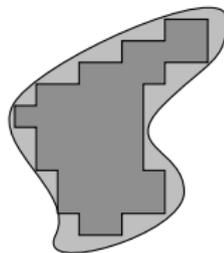


Approssimazione mediante poliedri

- Le regioni di spazio sono rappresentate come unione di poliedri;
- Efficienti da rappresentare e manipolare;
- Permettono di garantire la terminazione;



Sovra approssimazione



Sotto approssimazione

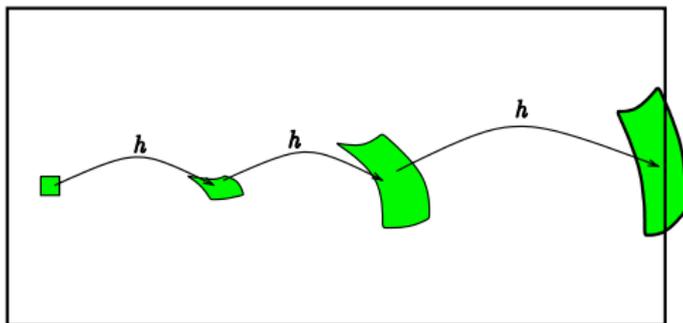
L'algoritmo di Phaver: descrizione generale

- inizia dalla regione degli stati iniziali
- finché si aggiungono nuovi stati:
 - ▶ Calcola l'**evoluzione continua** del sistema;
 - ▶ Calcola l'**evoluzione discreta** del sistema;

Domande:

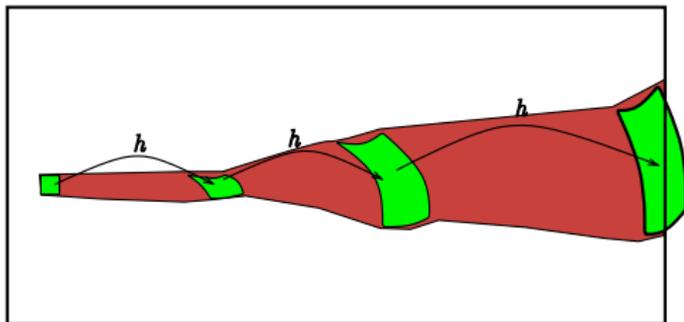
- Come si calcola l'evoluzione continua del sistema?
- Come si calcola l'evoluzione discreta del sistema?
- Come si garantisce che l'algoritmo termina?

Calcolo dell'evoluzione continua del sistema



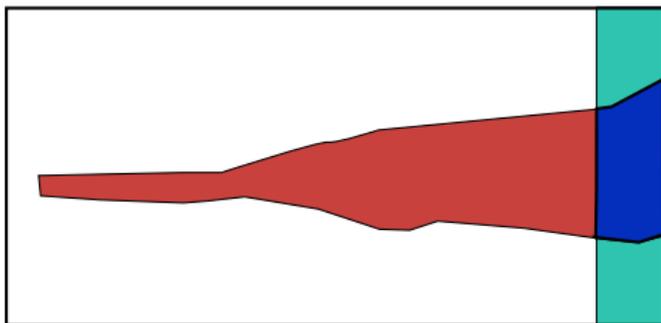
- 1 Finché sei all'interno dell'invariante della locazione:
 - ▶ calcola un passo di integrazione di ampiezza h ;
- 2 Calcola il tubo di flusso a partire dalle singole istantanee.

Calcolo dell'evoluzione continua del sistema



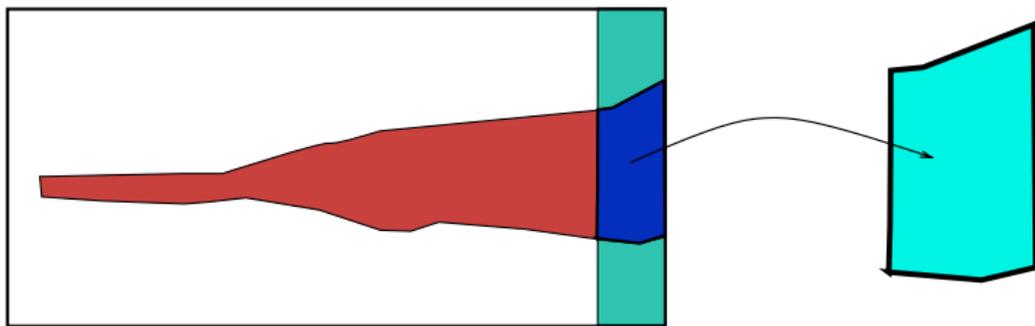
- 1 Finché sei all'interno dell'invariante della locazione:
 - ▶ calcola un passo di integrazione di ampiezza h ;
- 2 Calcola il tubo di flusso a partire dalle singole istantanee.

Calcolo dell'evoluzione discreta del sistema



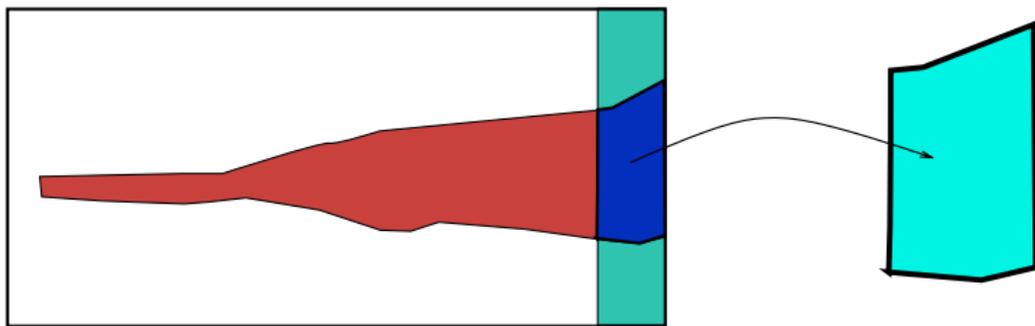
- 1 Calcola l'intersezione della regione raggiunta con le guardie delle transizioni uscenti;
- 2 Se non è vuota, applica i reset all'intersezione;
- 3 Ricomincia il calcolo dell'evoluzione continua dalla nuova regione.

Calcolo dell'evoluzione discreta del sistema



- 1 Calcola l'intersezione della regione raggiunta con le guardie delle transizioni uscenti;
- 2 Se non è vuota, applica i reset all'intersezione;
- 3 Ricomincia il calcolo dell'evoluzione continua dalla nuova regione.

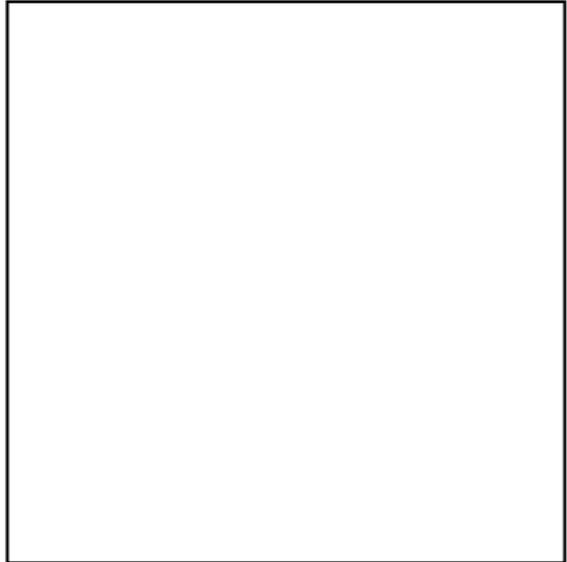
Calcolo dell'evoluzione discreta del sistema



- 1 Calcola l'intersezione della regione raggiunta con le guardie delle transizioni uscenti;
- 2 Se non è vuota, applica i reset all'intersezione;
- 3 Ricomincia il calcolo dell'evoluzione continua dalla nuova regione.

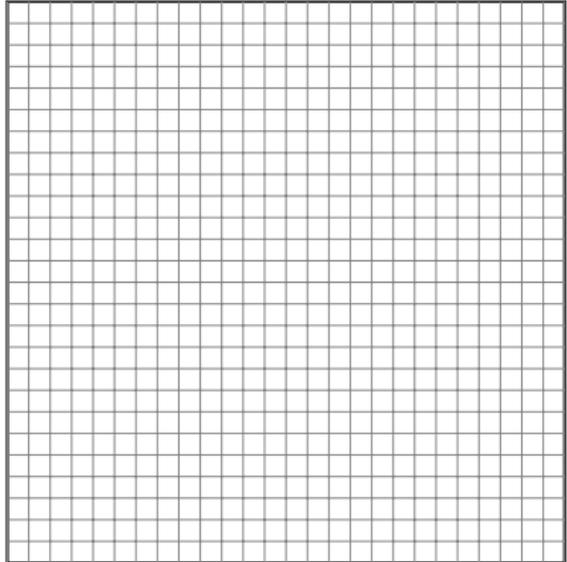
L'algoritmo di Phaver: terminazione

- Si restringe lo spazio di ricerca ad una **regione limitata**
- Divisione mediante una griglia in un numero **finito** di celle
- Le regioni sono rappresentate usando i punti della griglia



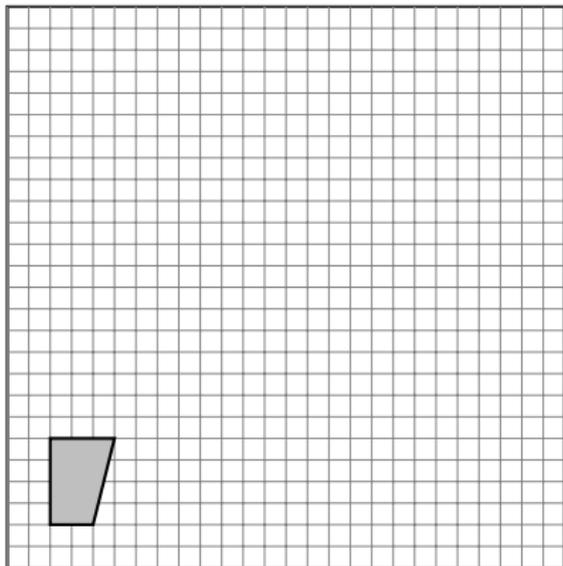
L'algoritmo di Phaver: terminazione

- Si restringe lo spazio di ricerca ad una **regione limitata**
- Divisione mediante una griglia in un numero **finito** di celle
- Le regioni sono rappresentate usando i punti della griglia



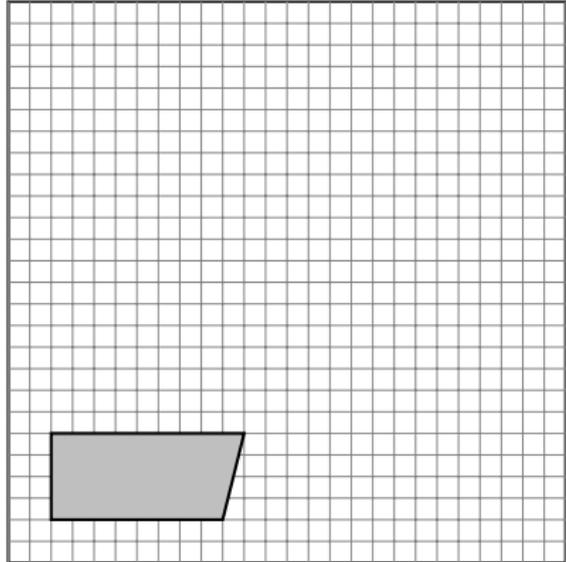
L'algoritmo di Phaver: terminazione

- Si restringe lo spazio di ricerca ad una **regione limitata**
- Divisione mediante una griglia in un numero **finito** di celle
- Le regioni sono rappresentate usando i punti della griglia



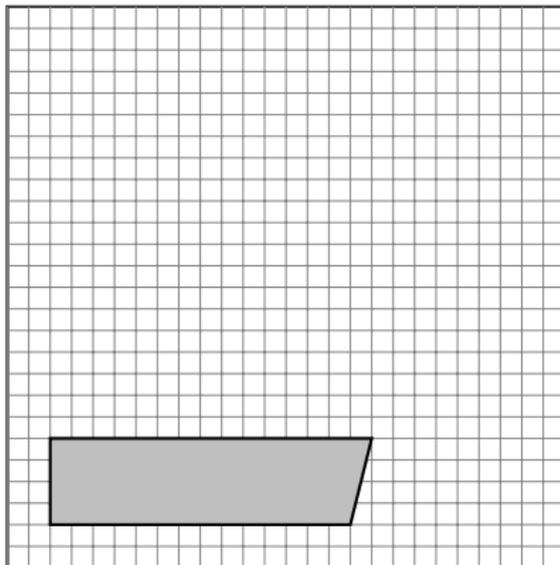
L'algoritmo di Phaver: terminazione

- Si restringe lo spazio di ricerca ad una **regione limitata**
- Divisione mediante una griglia in un numero **finito** di celle
- Le regioni sono rappresentate usando i punti della griglia



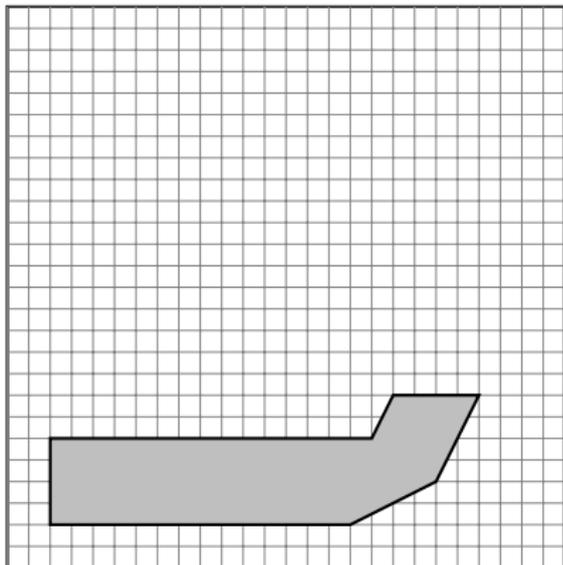
L'algoritmo di Phaver: terminazione

- Si restringe lo spazio di ricerca ad una **regione limitata**
- Divisione mediante una griglia in un numero **finito** di celle
- Le regioni sono rappresentate usando i punti della griglia



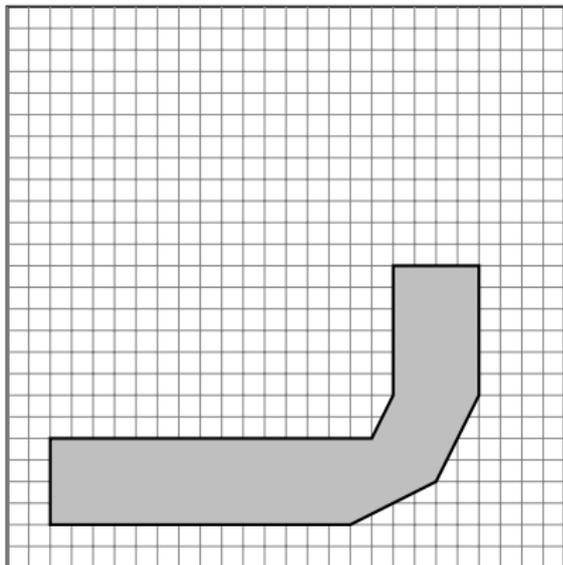
L'algoritmo di Phaver: terminazione

- Si restringe lo spazio di ricerca ad una **regione limitata**
- Divisione mediante una griglia in un numero **finito** di celle
- Le regioni sono rappresentate usando i punti della griglia



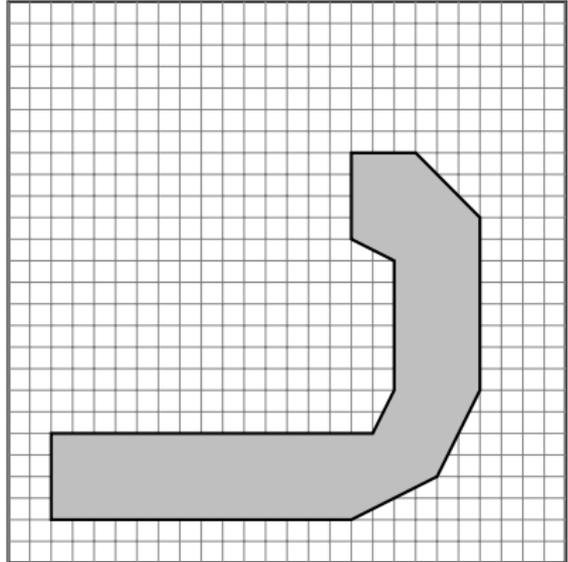
L'algoritmo di Phaver: terminazione

- Si restringe lo spazio di ricerca ad una **regione limitata**
- Divisione mediante una griglia in un numero **finito** di celle
- Le regioni sono rappresentate usando i punti della griglia



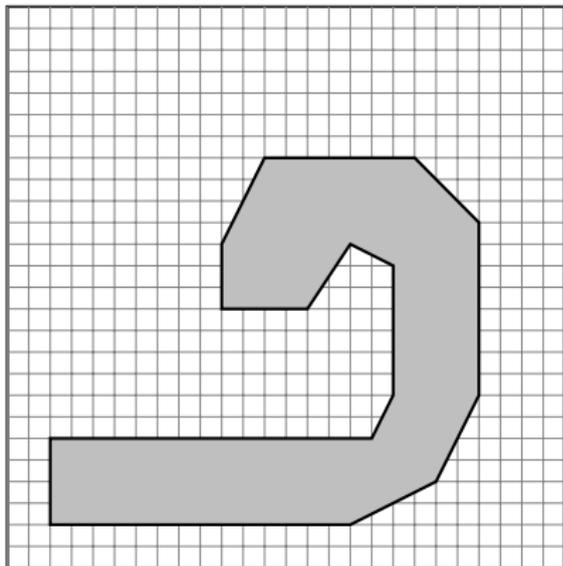
L'algoritmo di Phaver: terminazione

- Si restringe lo spazio di ricerca ad una **regione limitata**
- Divisione mediante una griglia in un numero **finito** di celle
- Le regioni sono rappresentate usando i punti della griglia



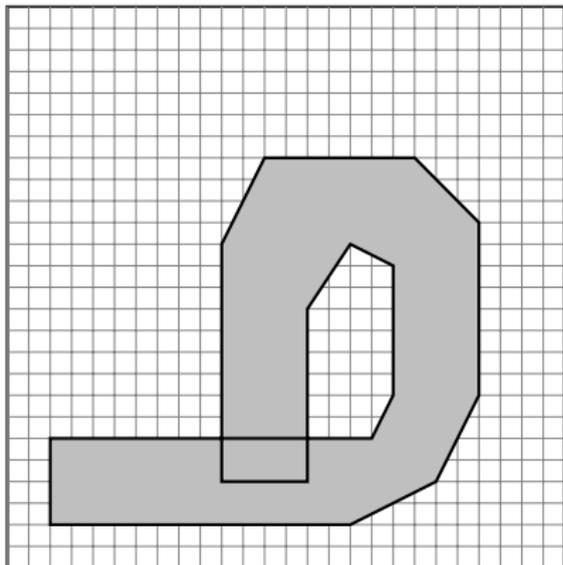
L'algoritmo di Phaver: terminazione

- Si restringe lo spazio di ricerca ad una **regione limitata**
- Divisione mediante una griglia in un numero **finito** di celle
- Le regioni sono rappresentate usando i punti della griglia



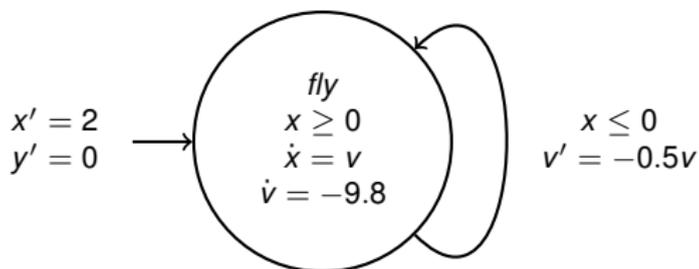
L'algoritmo di Phaver: terminazione

- Si restringe lo spazio di ricerca ad una **regione limitata**
- Divisione mediante una griglia in un numero **finito** di celle
- Le regioni sono rappresentate usando i punti della griglia



- 1 Una breve introduzione a PhaVer
- 2 Un primo esempio**
- 3 Esercizio

Esempio: una palla che rimbalza



- La palla cade da un'altezza iniziale di $2m$;
- ad ogni rimbalzo, la sua velocità viene dimezzata;
- v è la velocità della palla;
- x è la posizione della palla.

Modellare la palla che rimbalza in PhaVer (bouncing.pha)

```
// costante di gravita'  
//  
g:=9.8;  
  
// descrizione dell'automa  
//  
automaton bouncing_ball  
state_var: v, x;  
synclabs: jump;  
loc state: while x>=0  
    wait {x'==v & v'==-g}  
    when x<=0 sync jump do {v'==-v*0.5 & x'==x}  
    goto state;  
initially: state & x==2 & v==0;  
end
```

Modellare la palla che rimbalza in PhaVer (2)

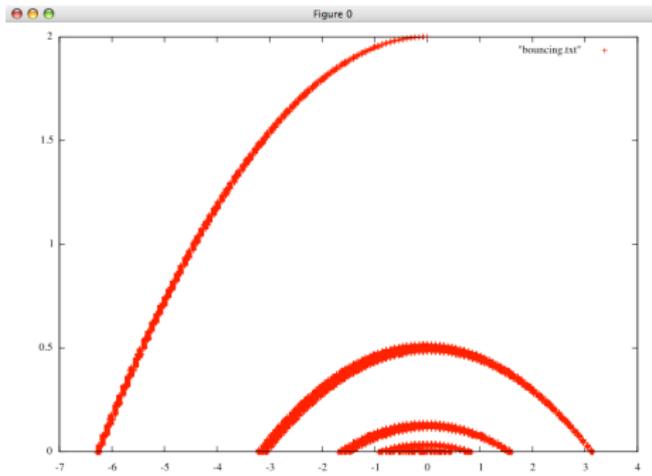
```
// Definizione della griglia
//
bouncing_ball.add_label(tau);
bouncing_ball.set_refine_constraints
    ((x, 0.1), (v, 0.1), tau);

//
// Comandi di analisi
//
reg=bouncing_ball.reachable;

//
// Salvataggio dell'output
//
reg.print("bouncing.txt", 2);
```

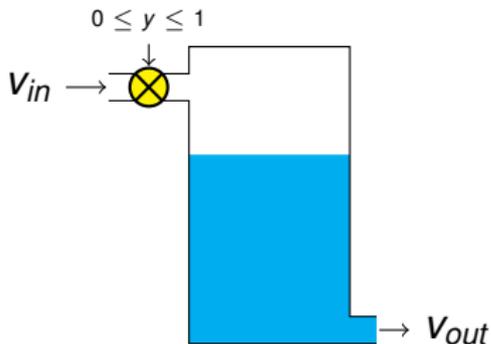
PhaVer: visualizzare le regioni raggiunte

- Lanciamo PhaVer sull'esempio della palla:
`phaver bouncing.pha;`
- Il file `.txt` di output può essere visualizzato con `gnuplot`:
 - ▶ lanciare `gnuplot`
 - ▶ al prompt, digitare il comando `plot "bouncing.txt"`



- 1 Una breve introduzione a PhaVer
- 2 Un primo esempio
- 3 Esercizio**

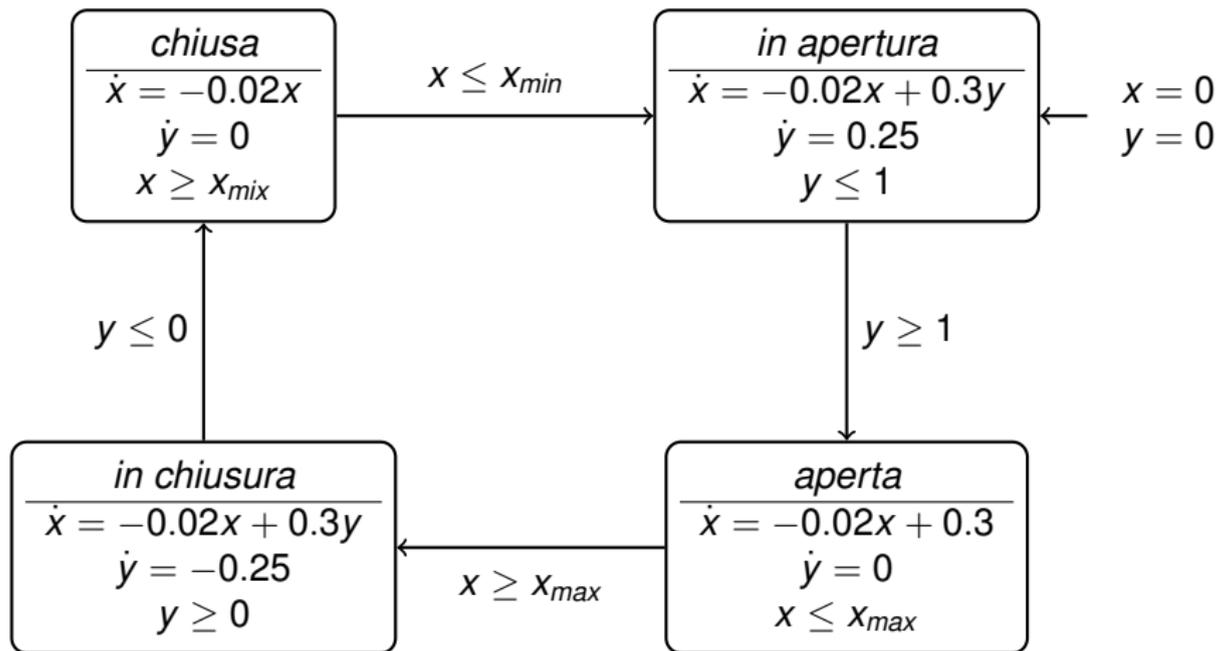
La cisterna: complichiamo le dinamiche



- y è il grado di apertura della valvola e varia tra 0 e 1
- l'acqua esce dalla cisterna con $v_{out} = -0.02x$;
- l'acqua entra nella cisterna con $v_{in} = 0.3y$;
- quando riceve un segnale, la valvola impiega 4s per aprirsi/chiudersi;
- all'inizio la cisterna è vuota, e la valvola si sta aprendo;

Simulare il sistema con $x_{min} = 5.5$ e $x_{max} = 8.5$

L'automa del sistema cisterna/controllore



- Implementare il sistema cisterna / controllore in PhaVer;
- Calcolare la regione raggiunta e verificare se il livello dell'acqua si stabilizza tra 5 e 9.

<http://profs.sci.univr.it/~bresolin/lab03.pdf>