Succinct Approximations of Distributed Hybrid Behaviors

P.S. Thiagarajan

School of Computing, National University of Singapore

Joint Work with: Yang Shaofa

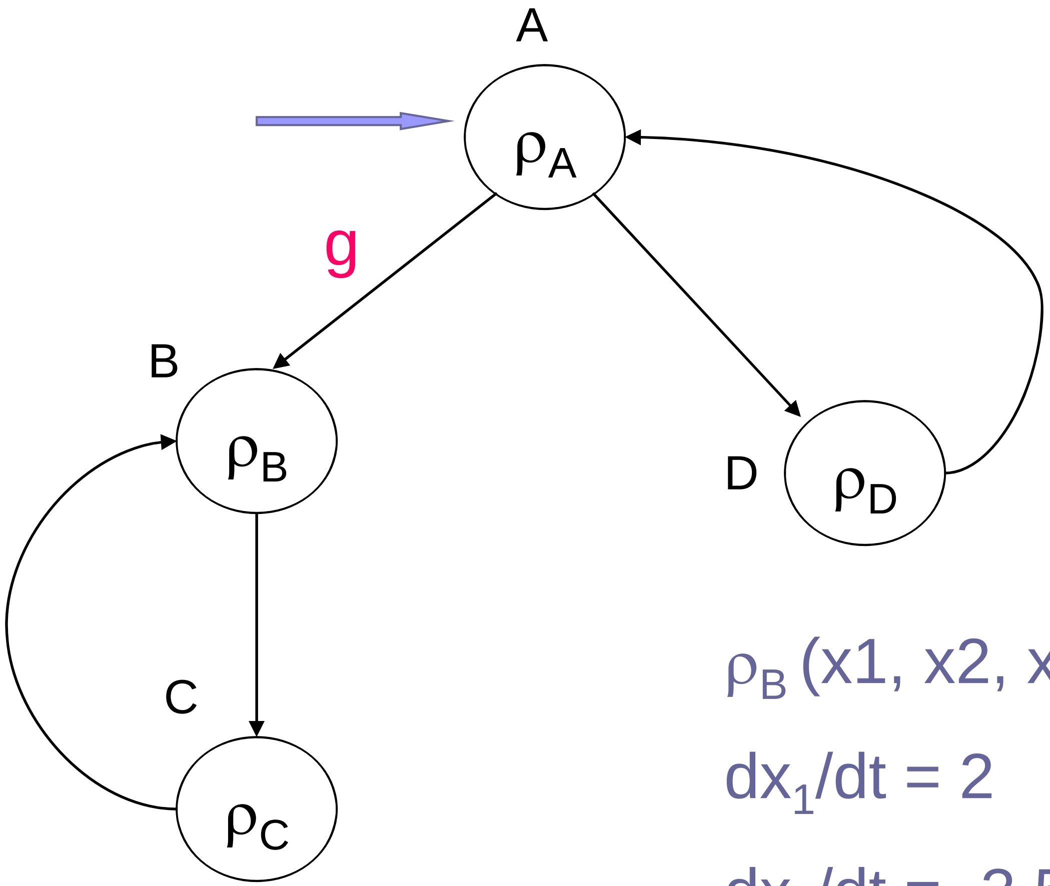
IIST, UNU, Macau

(To be presented at HSCC 2010)

Hybrid Automata

- Hybrid behaviors:
- Mode-specific continuous dynamics + discrete mode changes
- Standard model: ***Hybrid Automata***
  - Piecewise constant rates
  - Rectangular guards

Piecewise Constant Rates



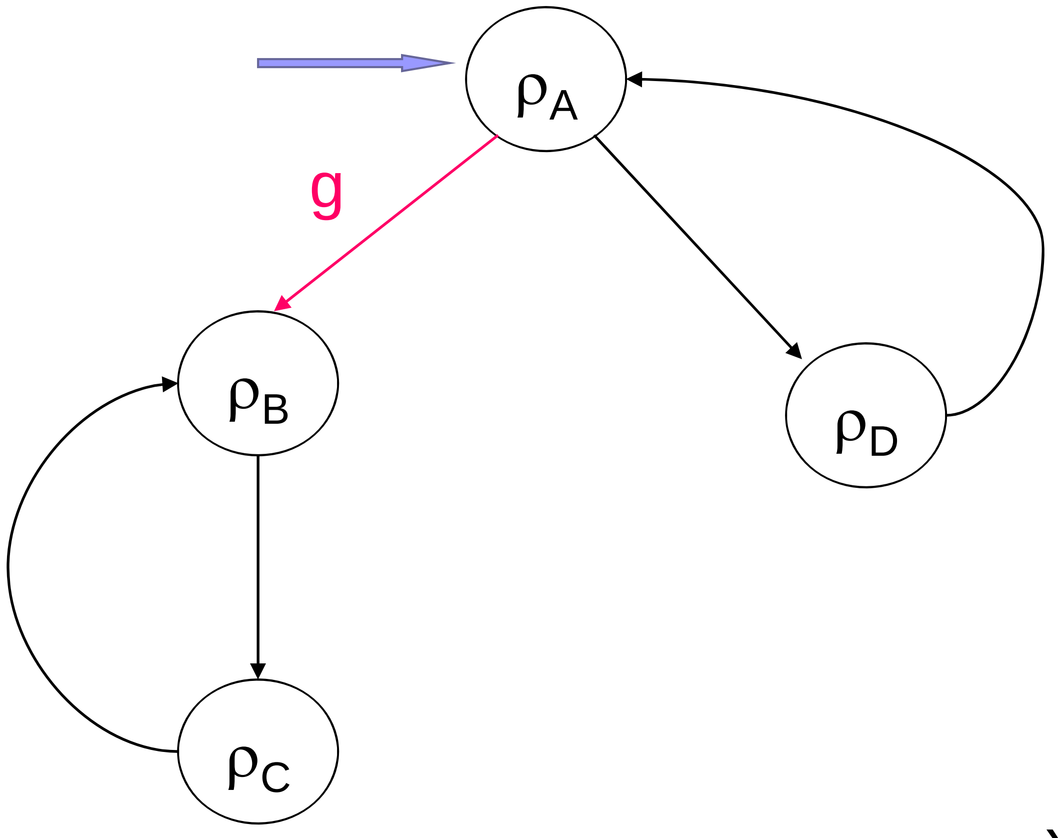
$\{x_1, x_2, x_3\}$

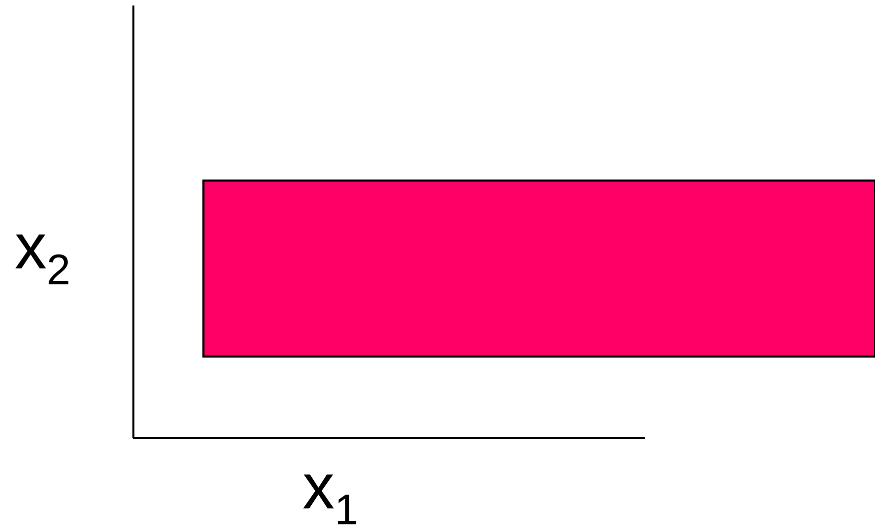
$\rho_B (x1, x2, x3) = (2, -3.5, 1)$

$dx_1/dt = 2 \quad x_1(t) = 2t + x_1(0)$

$dx_2/dt = -3.5 \quad dx_3/dt = 1$

Rectangular Guards



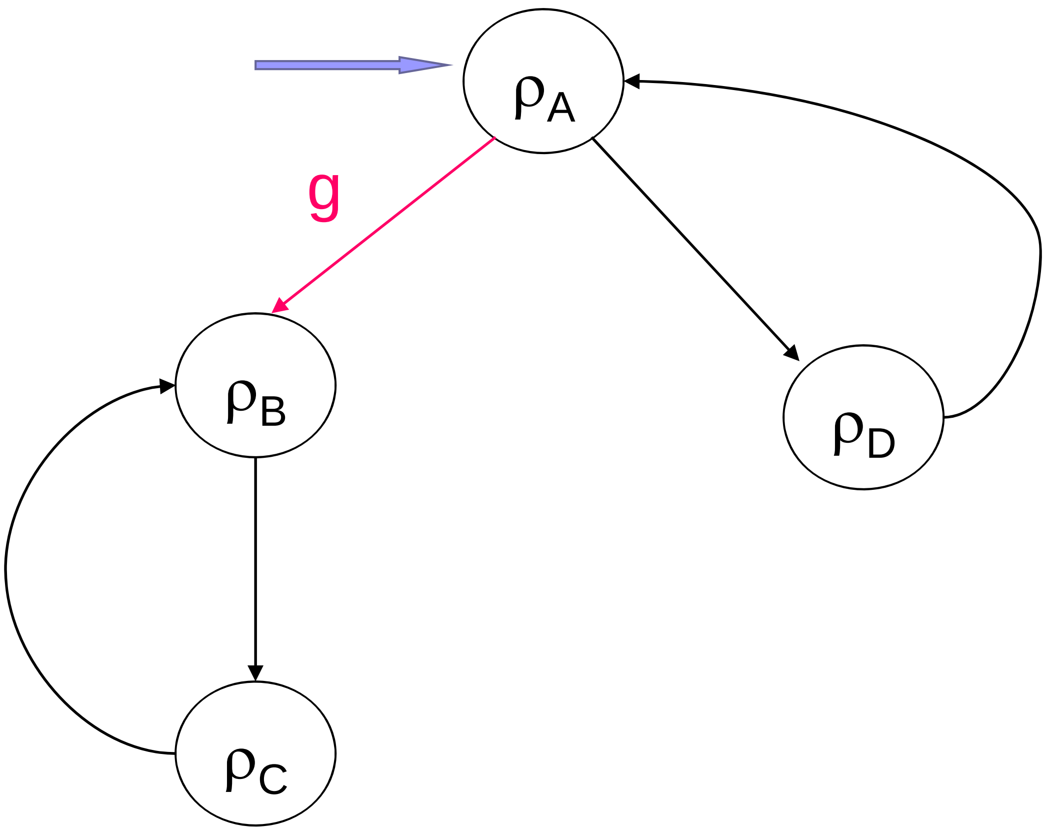


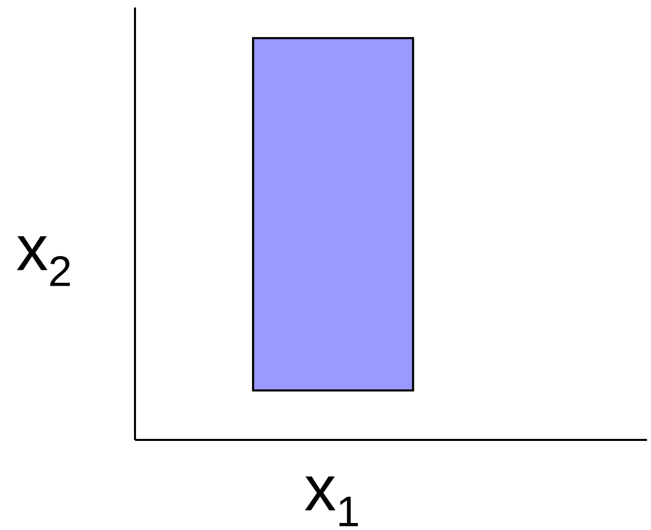
$x \leq c \;|\; x \geq c \;|\; \varphi \wedge \varphi'$

Initial Regions

$(x_1 \leq 2 \;\wedge\; x_1 \geq 1) \;\wedge$

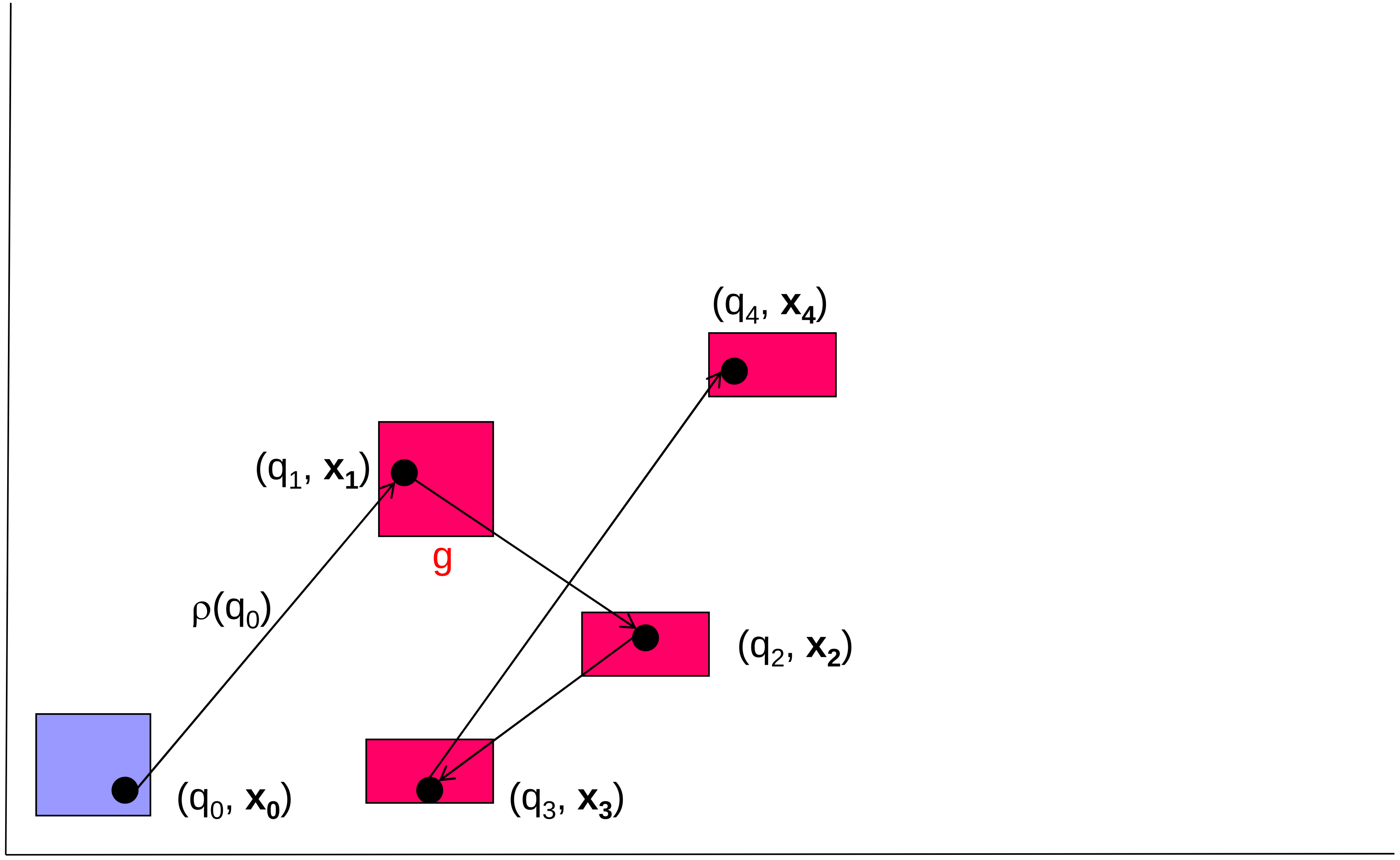
$(x_2 \leq 3.5 \;\wedge\; x_2 \geq 0.5)$





Highly Expressive

- Piecewise constant rates; rectangular guards
- The control state (mode) reachability problem is undecidable.
  - Given q_f, Whether there exists a trajectory $(q_0, \mathbf{x_0})\ (q_1, \mathbf{x_1})\ \ldots\ldots\ (q_m, \mathbf{x_m})$ such that $q_m = q_f$
  - q_0 = initial mode; $\mathbf{x_0}$ in initial region.
- ***HKPV'95***

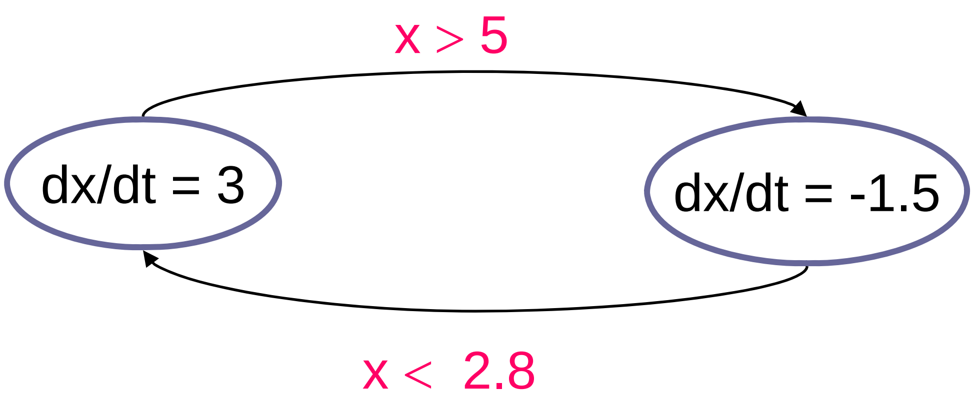


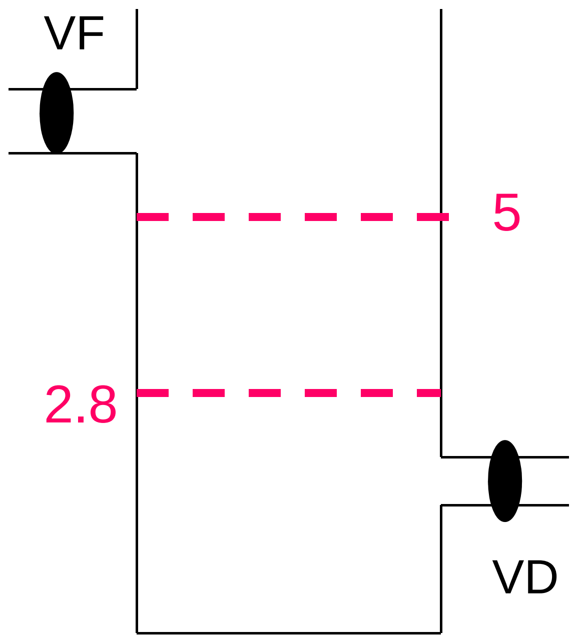
(q4, x4)
(q1, x1)
g
ρ(q0)
(q2, x2)
(q0, x0)
(q3, x3)

Two main ways to circumvent undecidability

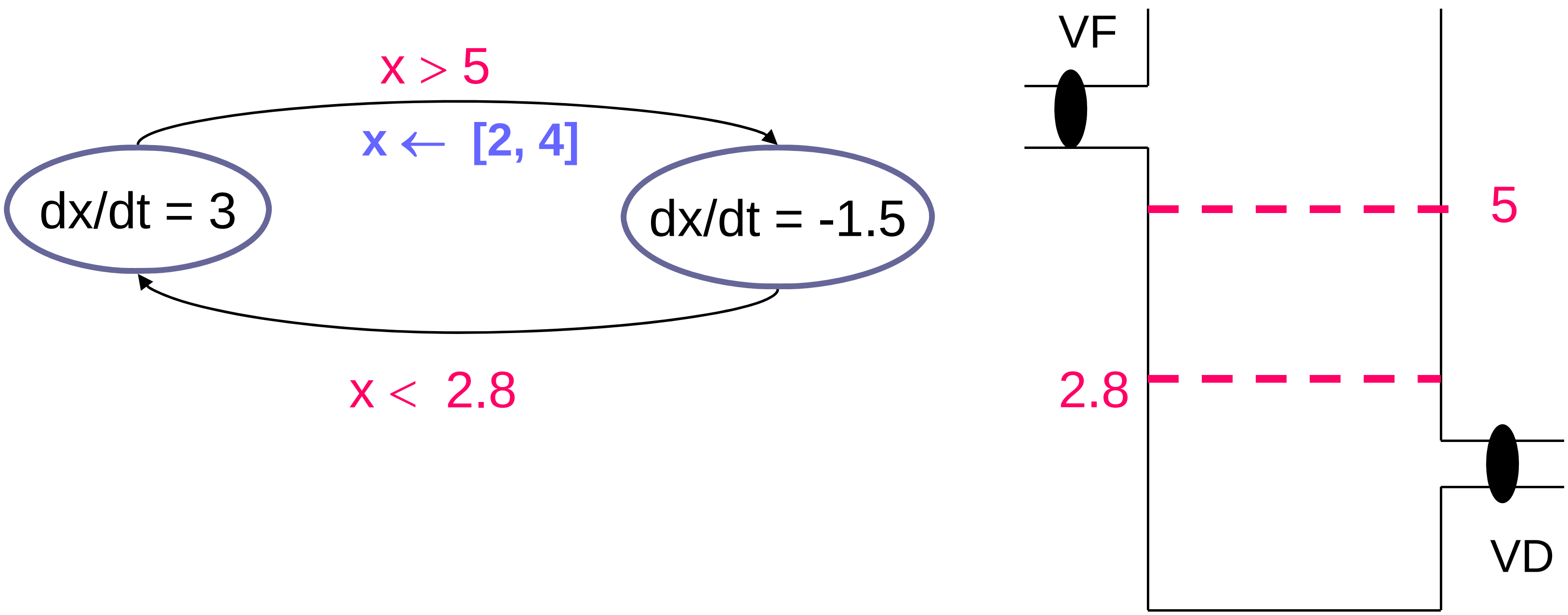
- If its rate changes as the result of a mode change,
  - Reset the value of a variable to a pre-determined region

Hybrid Automata with resets.

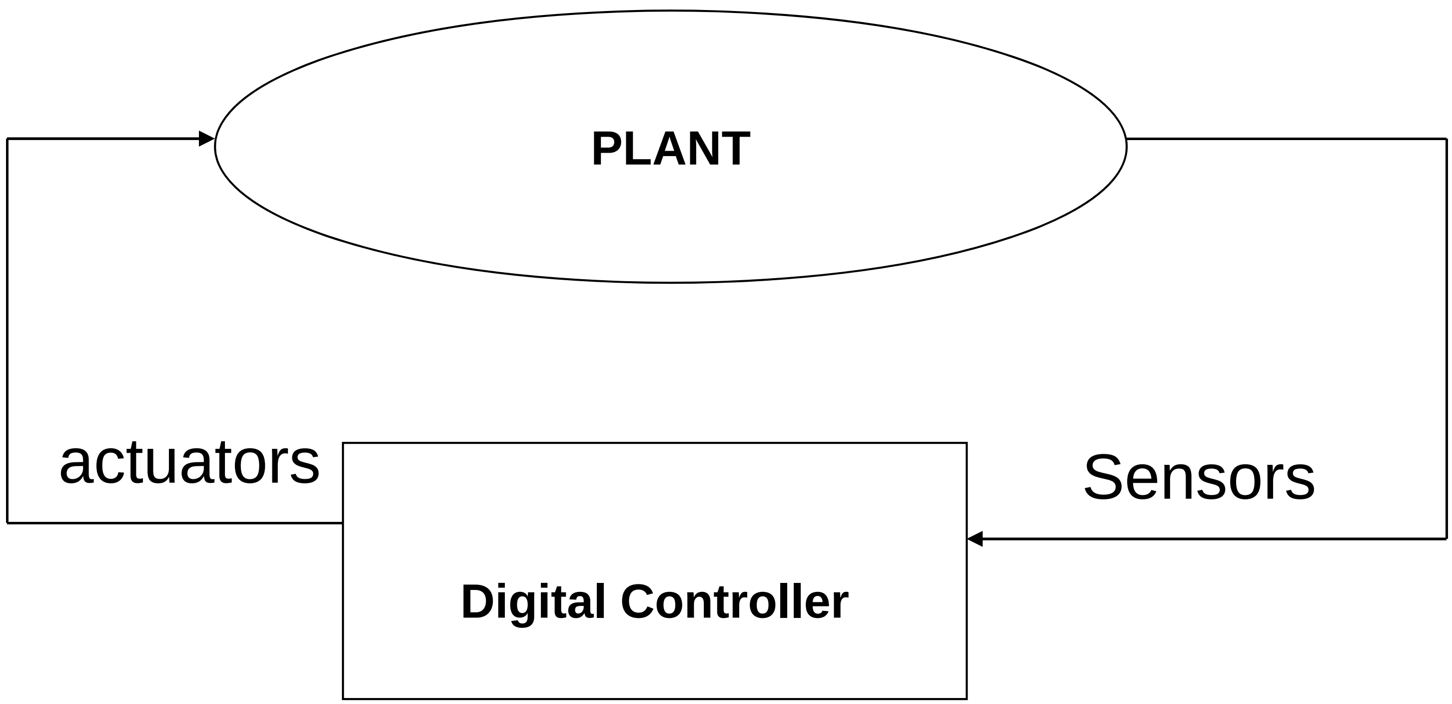




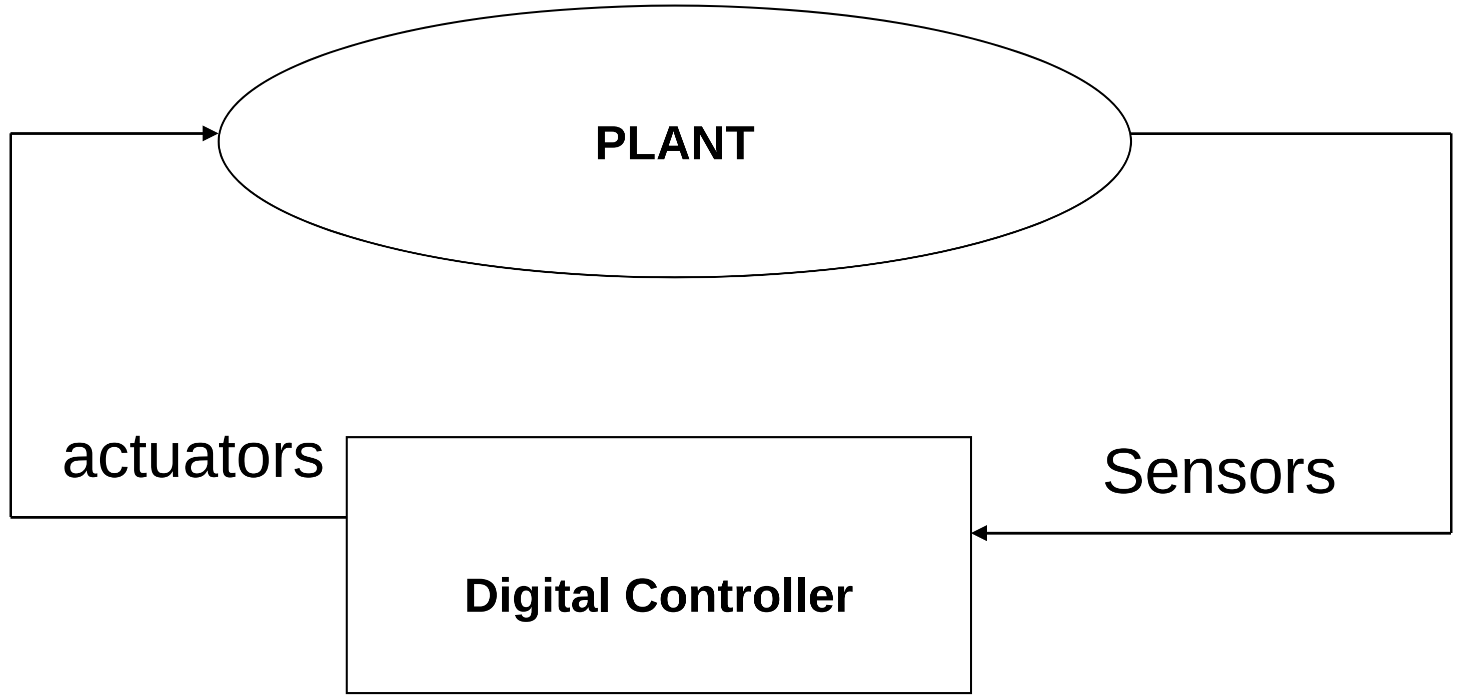
Hybrid Automata with resets.



Control Applications



The reset assumption is untenable.



[HK'97]: Discrete time assumption.

The plant state is observed only at (periodic) discrete time points T_0 T_1 T_2 …..

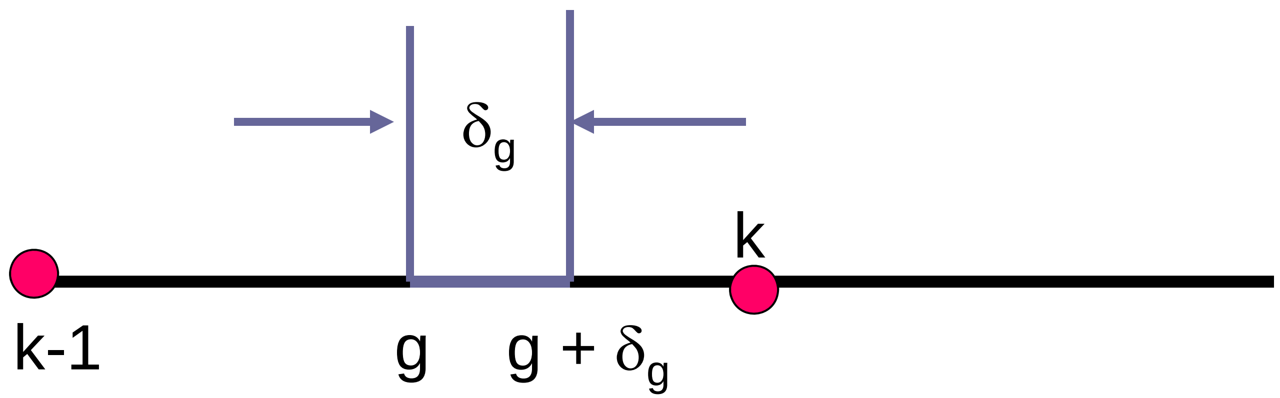
$T_{i+1} - T_i = \Delta$

Discrete time behaviors

- ***The discrete time behavior of a hybrid automaton:***
  - Q : The set of modes
  - $q_0\ q_1 \ldots q_m$ is a ***state sequence*** iff there exists a run $(q_0, v_0)\ (q_1, v_1) \ldots (q_m, v_m)$ of the automaton.
  - ***The discrete time behavior of Aut*** is
    - $L(Aut) \subseteq Q^*$
    - the set of state sequences of Aut.

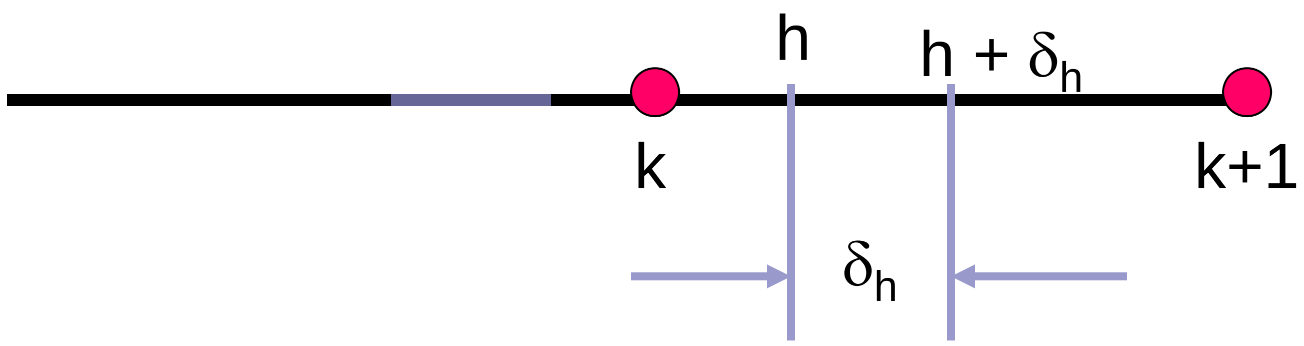
- [HK'97]:
  - The discrete time behavior of (piecewise constant + rectangular guards) an hybrid automaton is regular.
  - A finite state automaton representing this language can be effectively constructed.
- Discrete time behavior is an approximation. With fast enough sampling, it is a good approximation.

- [AT'04]: The discrete time behavior of an hybrid automaton is regular even with delays in sensing and actuating (laziness)



The value of x_i reported at t = k is the value at some t' in [(k-1)+ g, (k-1)+g +δ_g]

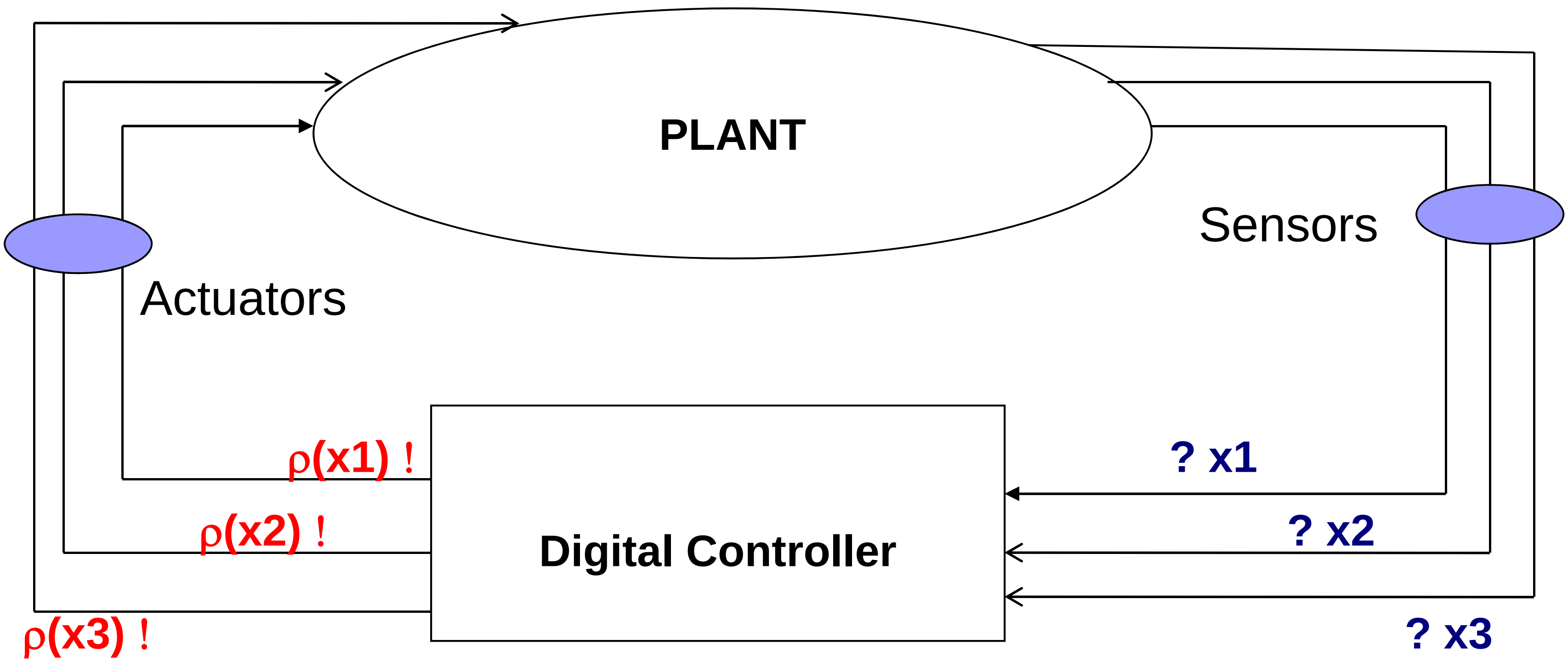
g and δ_g are fixed rationals



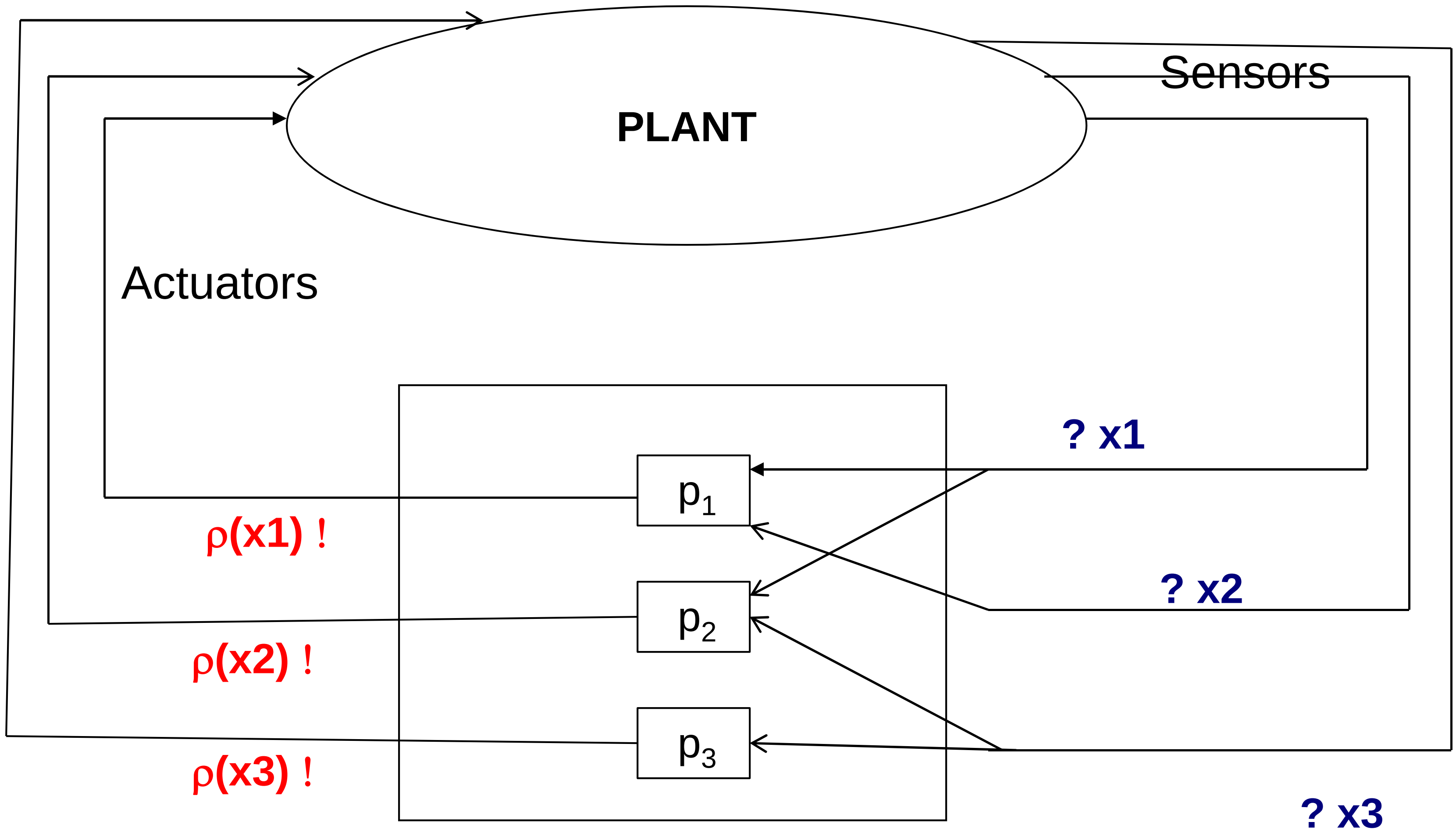
If a mode change takes place at t = k is then x_i starts evolving at $\rho'(x_i)$ at some t' in [k+h, k+h + δ_h]

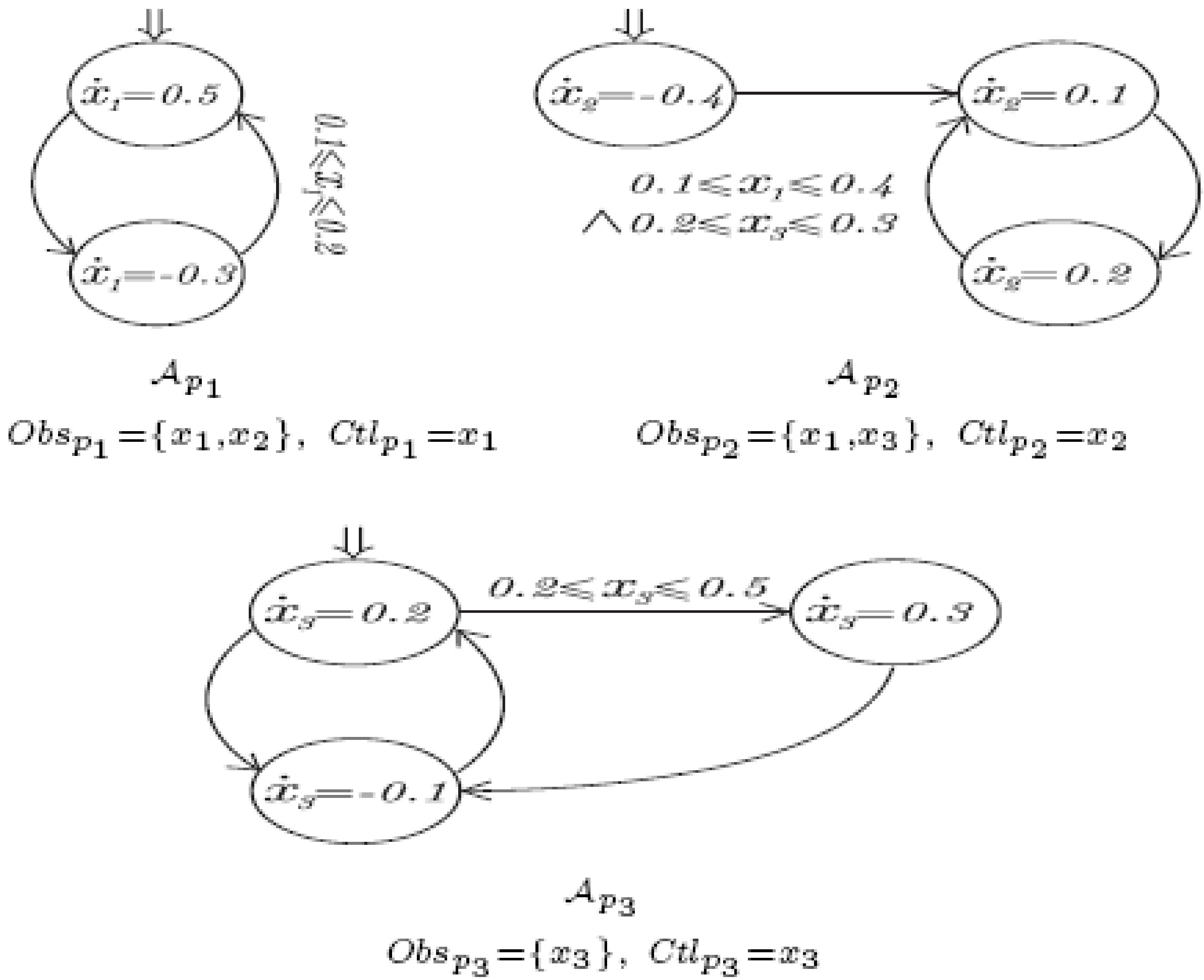
h and δ_h are fixed rationals.

Global Hybrid Automata

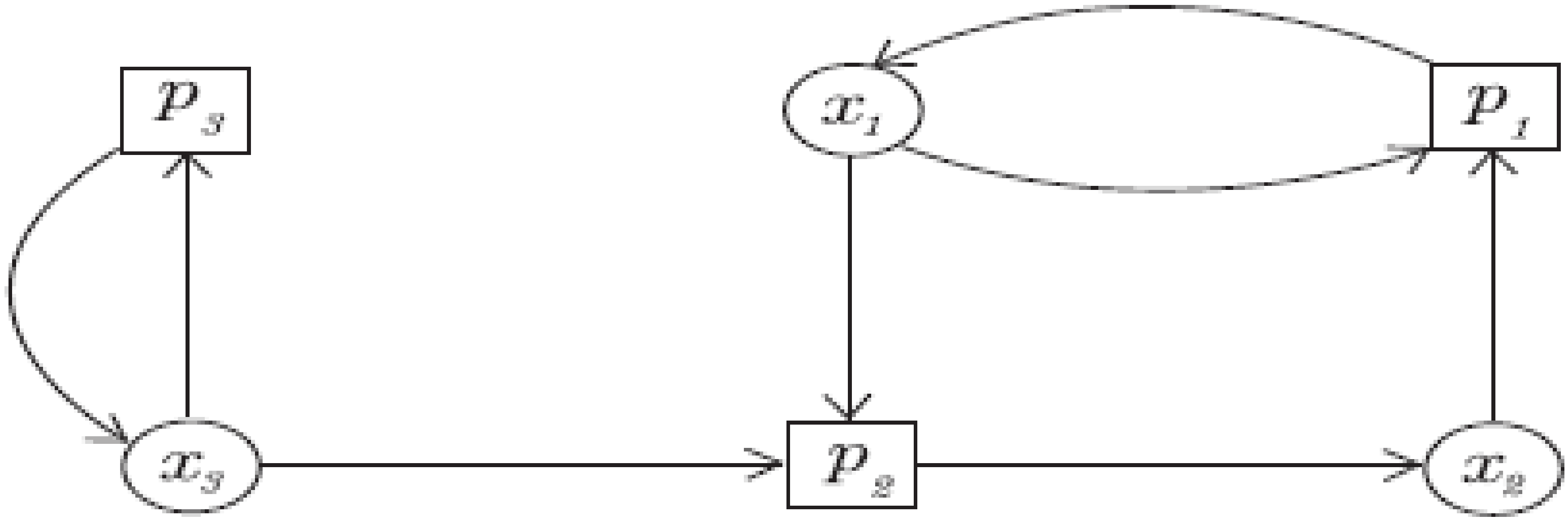


Distributed Hybrid Automata





No explicit communication between the automata.. However, coordination through the shared memory of the plant's state space.



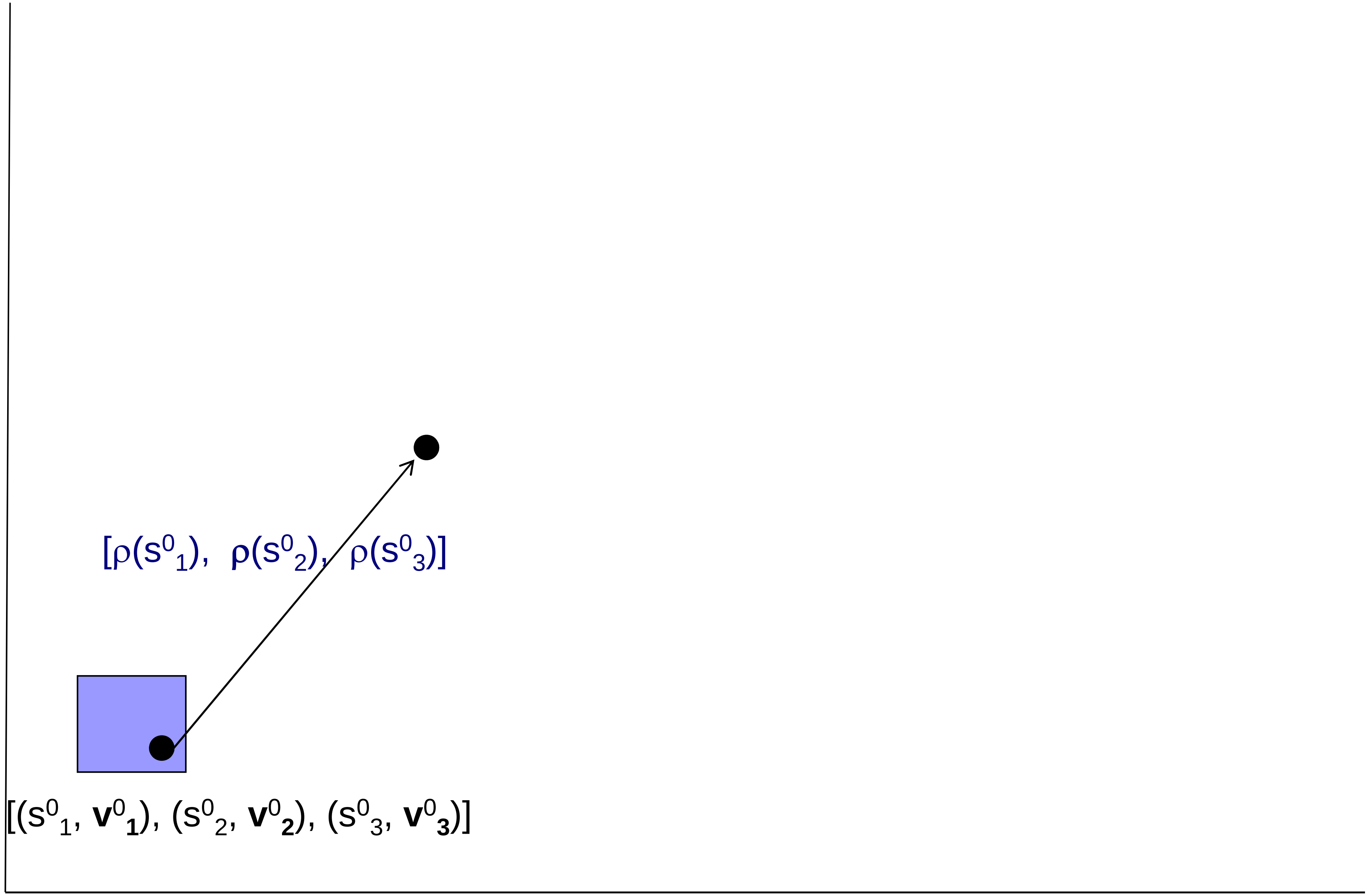
The Communictaion graph of DHA

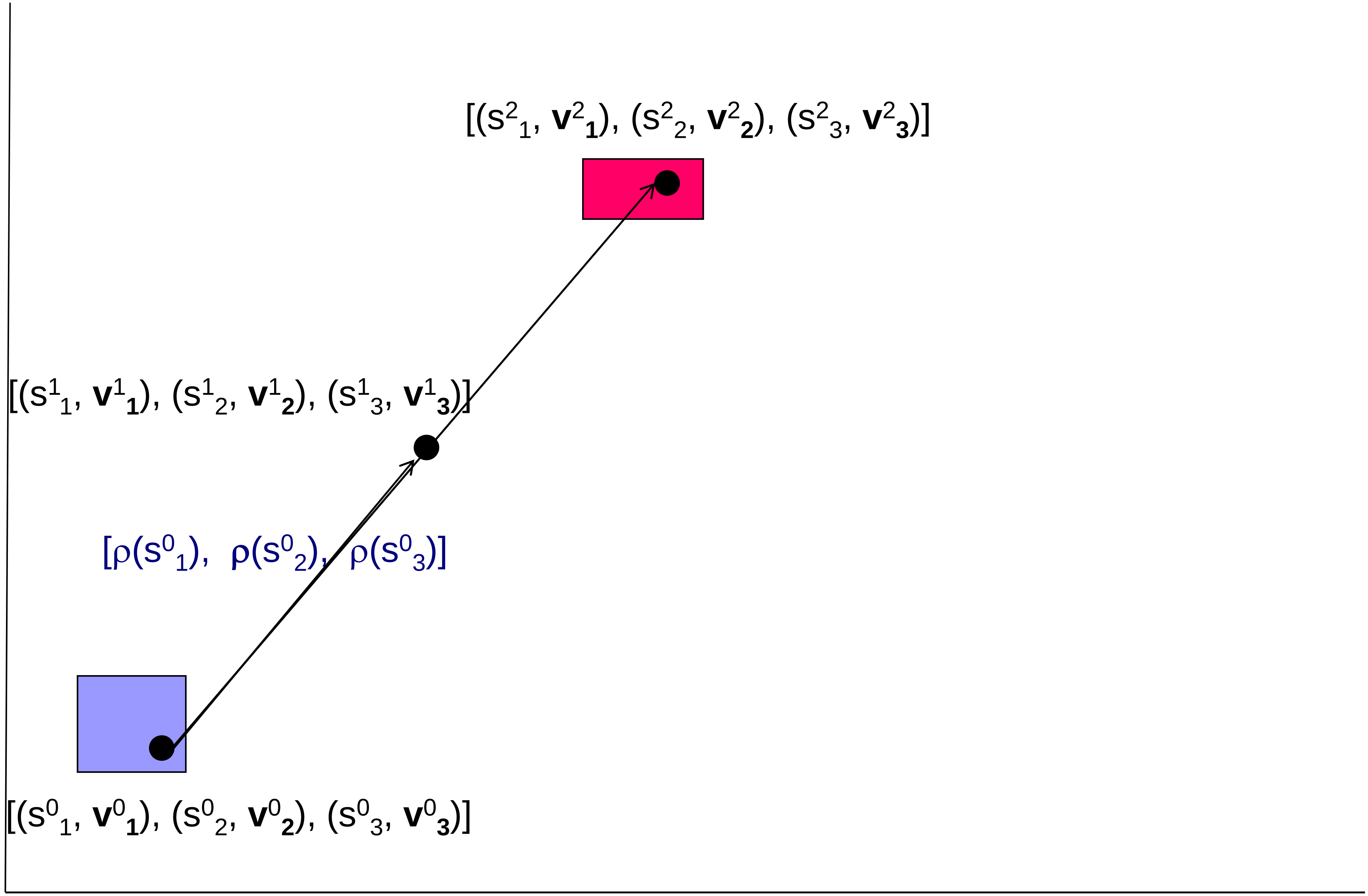
Obs(p) --- The set of variables observed by p

Ctl(p) --- The set of variables controlled by p

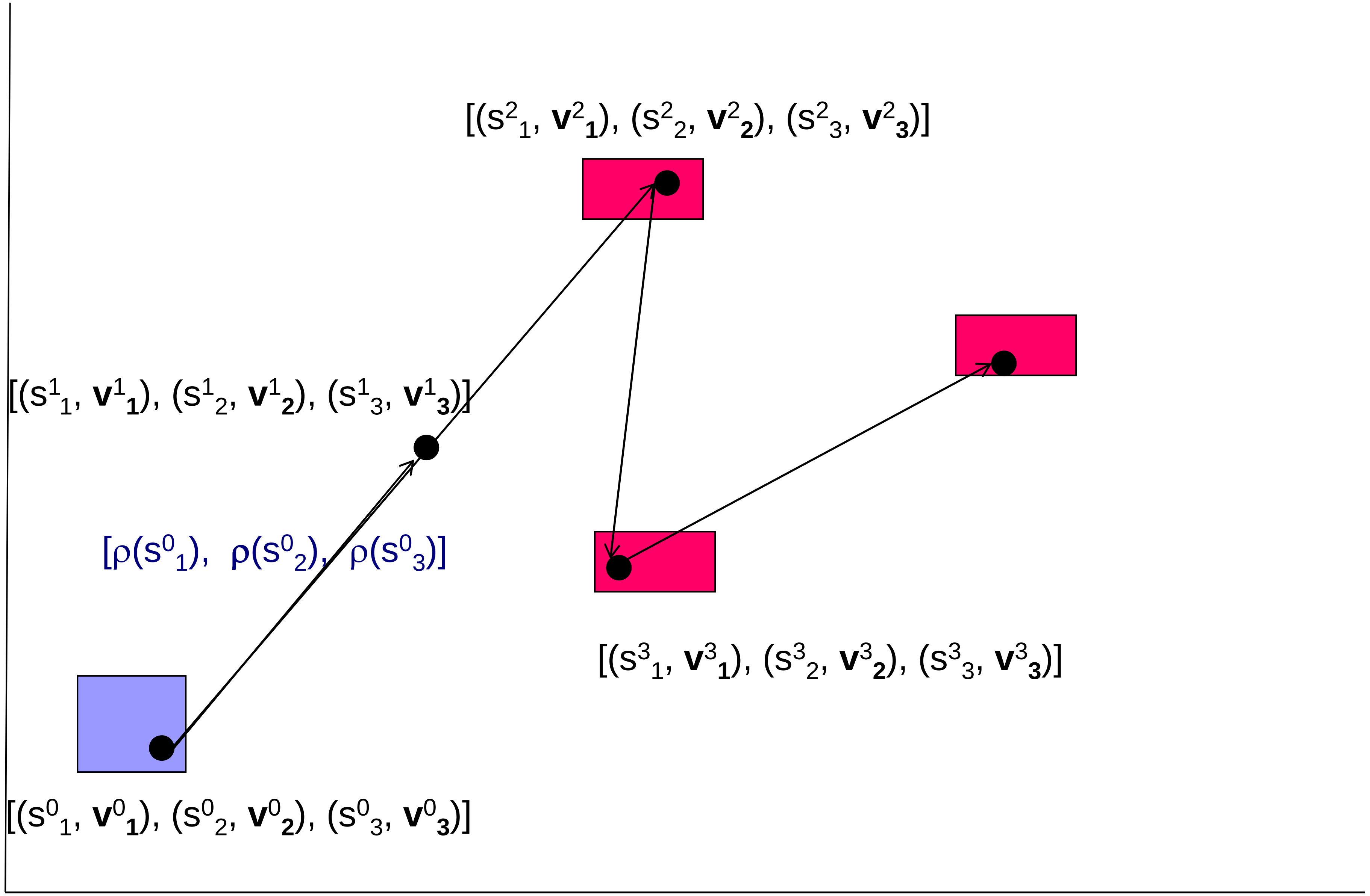
Ctl(p) $\cap$ Ctl(q) = $\varnothing$

Nbr(p) = Obs(p) $\cup$ Ctl(p)

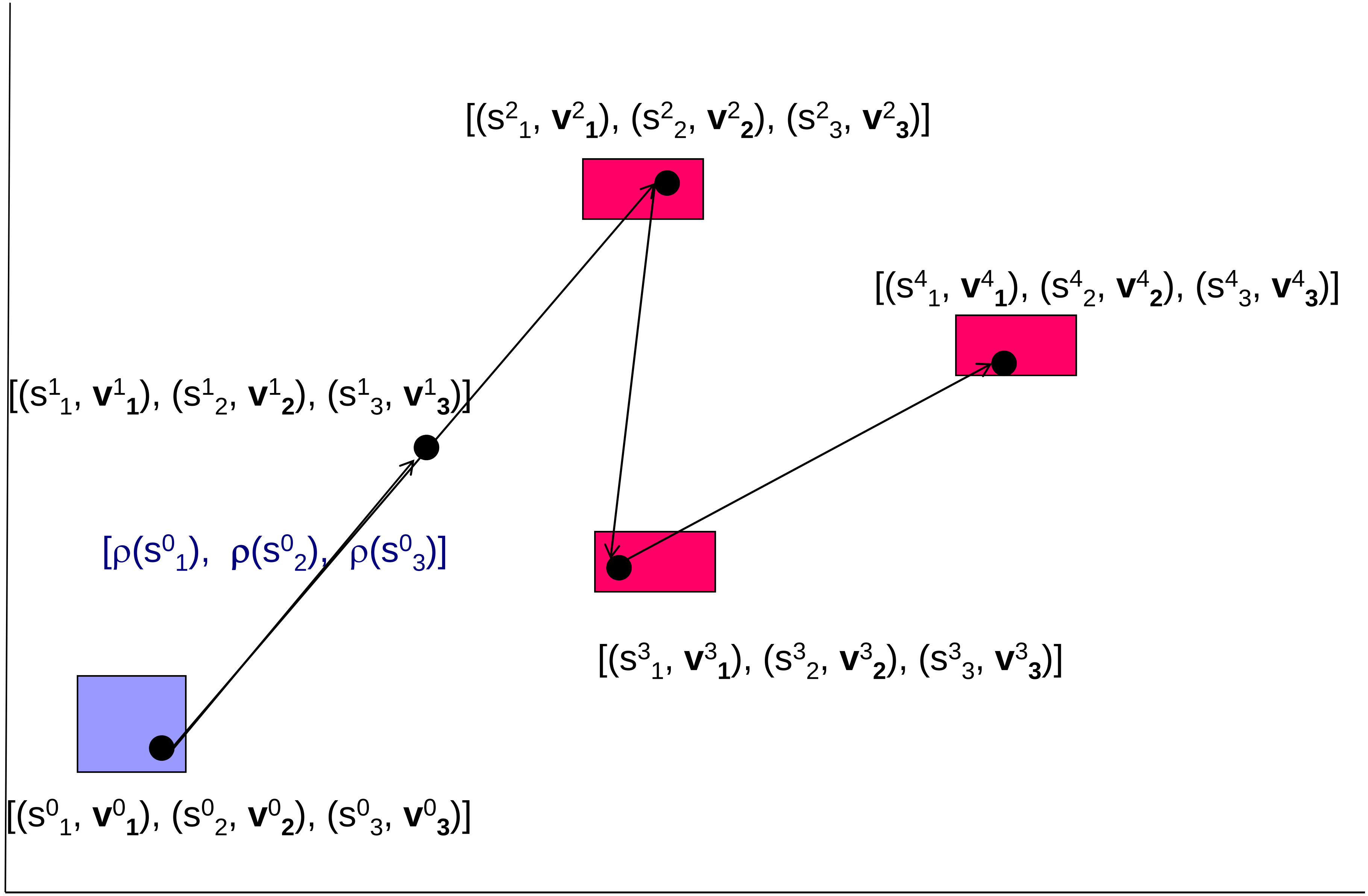

$[\rho(s^0_1),\ \rho(s^0_2),\ \rho(s^0_3)]$
$[(s^0_1, \mathbf{v}^0_{\mathbf{1}}), (s^0_2, \mathbf{v}^0_{\mathbf{2}}), (s^0_3, \mathbf{v}^0_{\mathbf{3}})]$

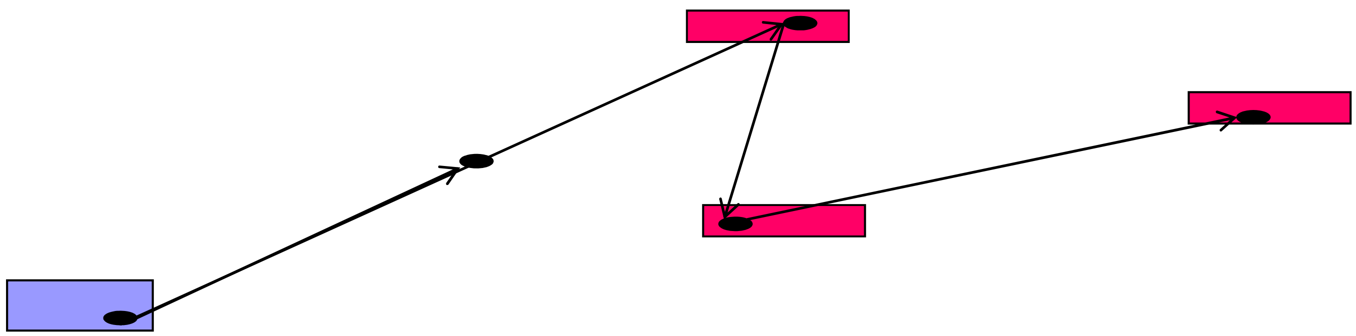


$[(s^2_1, \mathbf{v}^2_{\mathbf{1}}), (s^2_2, \mathbf{v}^2_{\mathbf{2}}), (s^2_3, \mathbf{v}^2_{\mathbf{3}})]$
$[(s^1_1, \mathbf{v}^1_{\mathbf{1}}), (s^1_2, \mathbf{v}^1_{\mathbf{2}}), (s^1_3, \mathbf{v}^1_{\mathbf{3}})]$
$[\rho(s^0_1), \rho(s^0_2), \rho(s^0_3)]$
$[(s^0_1, \mathbf{v}^0_{\mathbf{1}}), (s^0_2, \mathbf{v}^0_{\mathbf{2}}), (s^0_3, \mathbf{v}^0_{\mathbf{3}})]$



$[(s^2_1, \mathbf{v}^2_{\mathbf{1}}), (s^2_2, \mathbf{v}^2_{\mathbf{2}}), (s^2_3, \mathbf{v}^2_{\mathbf{3}})]$
$[(s^1_1, \mathbf{v}^1_{\mathbf{1}}), (s^1_2, \mathbf{v}^1_{\mathbf{2}}), (s^1_3, \mathbf{v}^1_{\mathbf{3}})]$
$[\rho(s^0_1), \rho(s^0_2), \rho(s^0_3)]$
$[(s^3_1, \mathbf{v}^3_{\mathbf{1}}), (s^3_2, \mathbf{v}^3_{\mathbf{2}}), (s^3_3, \mathbf{v}^3_{\mathbf{3}})]$
$[(s^0_1, \mathbf{v}^0_{\mathbf{1}}), (s^0_2, \mathbf{v}^0_{\mathbf{2}}), (s^0_3, \mathbf{v}^0_{\mathbf{3}})]$


$[(s^2_1, \mathbf{v}^2_{\mathbf{1}}), (s^2_2, \mathbf{v}^2_{\mathbf{2}}), (s^2_3, \mathbf{v}^2_{\mathbf{3}})]$
$[(s^4_1, \mathbf{v}^4_{\mathbf{1}}), (s^4_2, \mathbf{v}^4_{\mathbf{2}}), (s^4_3, \mathbf{v}^4_{\mathbf{3}})]$
$[(s^1_1, \mathbf{v}^1_{\mathbf{1}}), (s^1_2, \mathbf{v}^1_{\mathbf{2}}), (s^1_3, \mathbf{v}^1_{\mathbf{3}})]$
$[\rho(s^0_1), \rho(s^0_2), \rho(s^0_3)]$
$[(s^3_1, \mathbf{v}^3_{\mathbf{1}}), (s^3_2, \mathbf{v}^3_{\mathbf{2}}), (s^3_3, \mathbf{v}^3_{\mathbf{3}})]$
$[(s^0_1, \mathbf{v}^0_{\mathbf{1}}), (s^0_2, \mathbf{v}^0_{\mathbf{2}}), (s^0_3, \mathbf{v}^0_{\mathbf{3}})]$



Discrete time behavior:
(Global) state sequences

$[s^0_1, s^0_2, s^0_3]$ $[s^1_1, s^1_2, s^1_3]$ $[s^2_1, s^2_2, s^2_3]$ $[s^3_1, s^3_2, s^3_3]$ . . . .

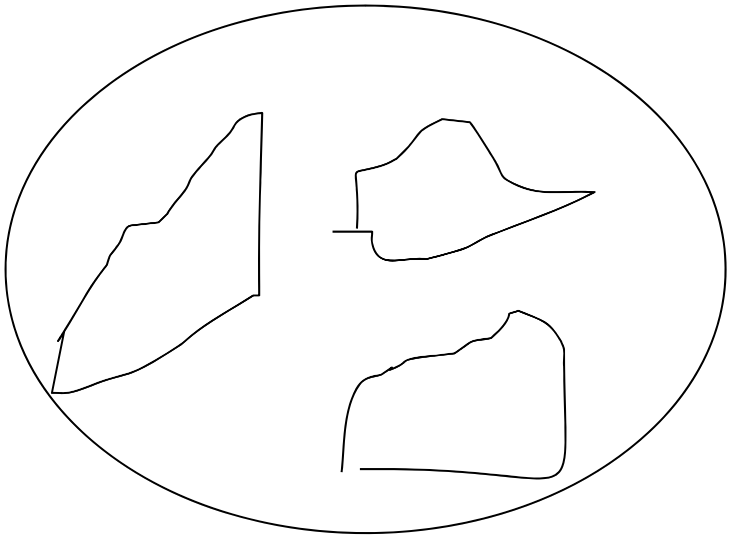
Discrete time behaviors

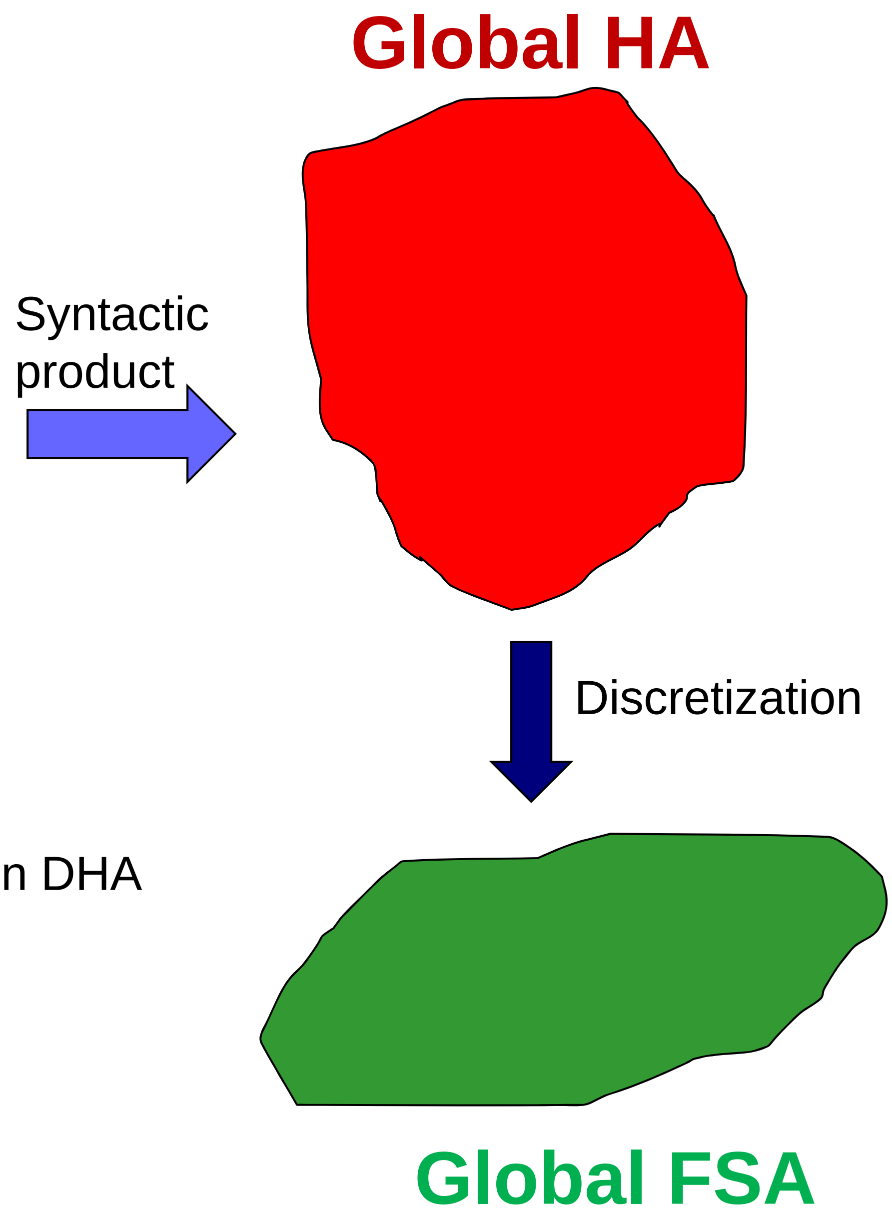
- ***The discrete time behavior of DHA*** is
  - L(*DHA*) $\subseteq$ $(S_{p1} \times S_{p2} \times \ldots\ldots \times S_{pn})^*$
  - the set of global state sequences of DHA.
- L(*DHA*) is regular?

Discrete time behaviors

- ***The discrete time behavior of DHA*** is
  - L(*DHA*) $\subseteq$ $(S_{p1} \times S_{p2} \times \ldots\ldots \times S_{pn})^*$
  - the set of global state sequences of DHA.
- L(*DHA*) is regular?
- Yes. Construct the (syntactic product) AUT of *DHA.*
- AUT will have piecewise constant rates and rectangular guards. Hence…..

Network of HAs



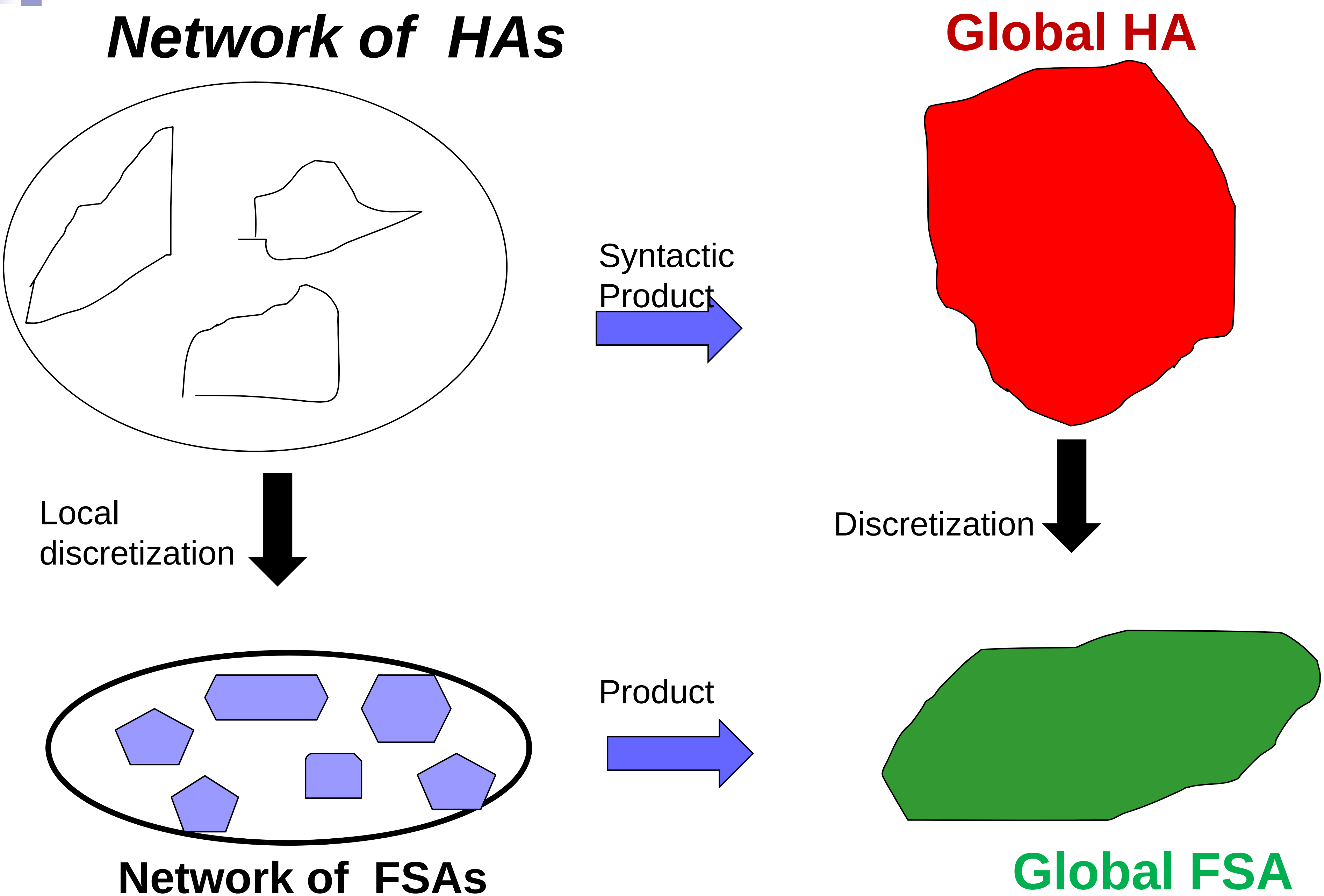


m ---- the number of component automata in DHA

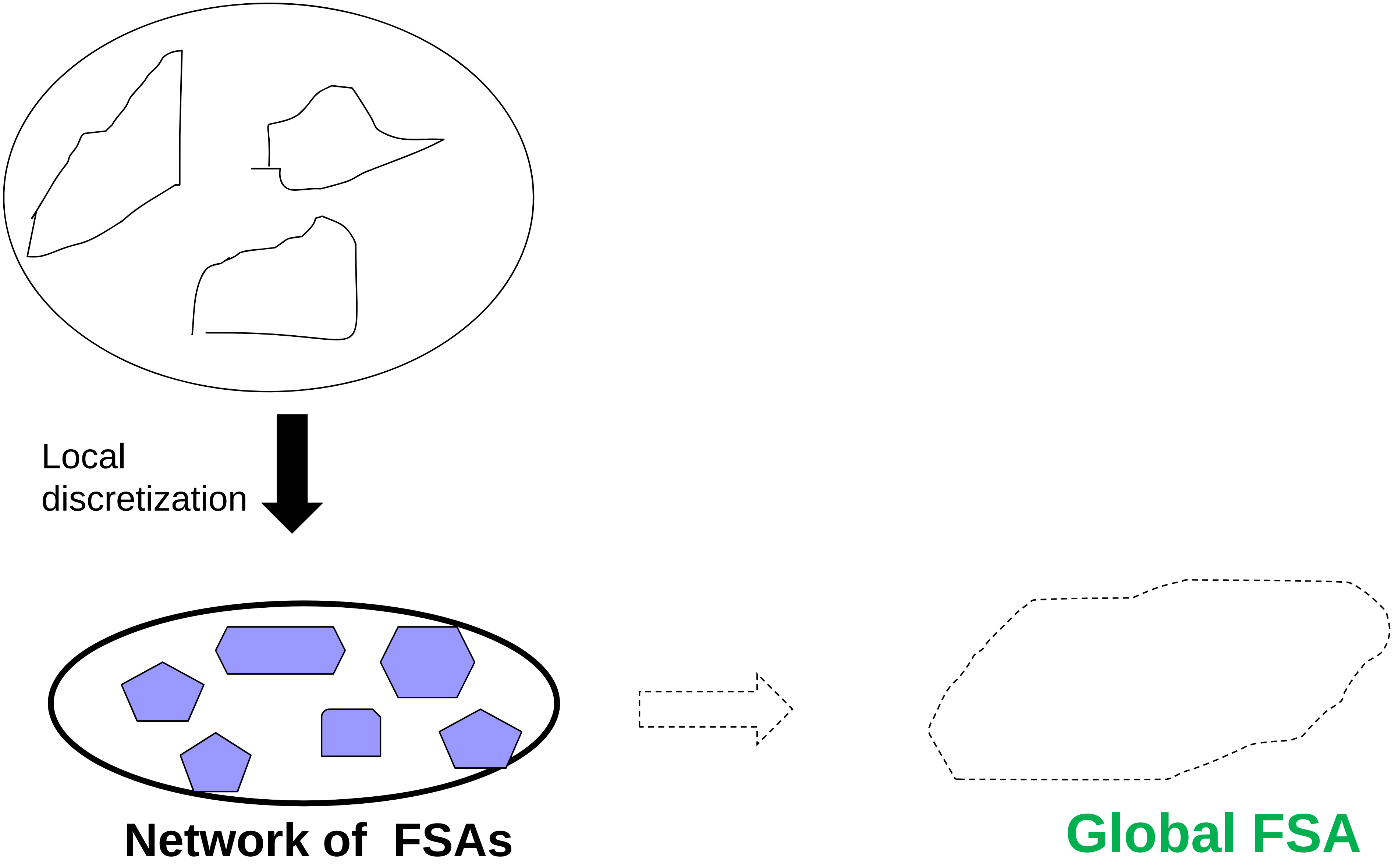
The size of ***DHA*** will be **linear** in m

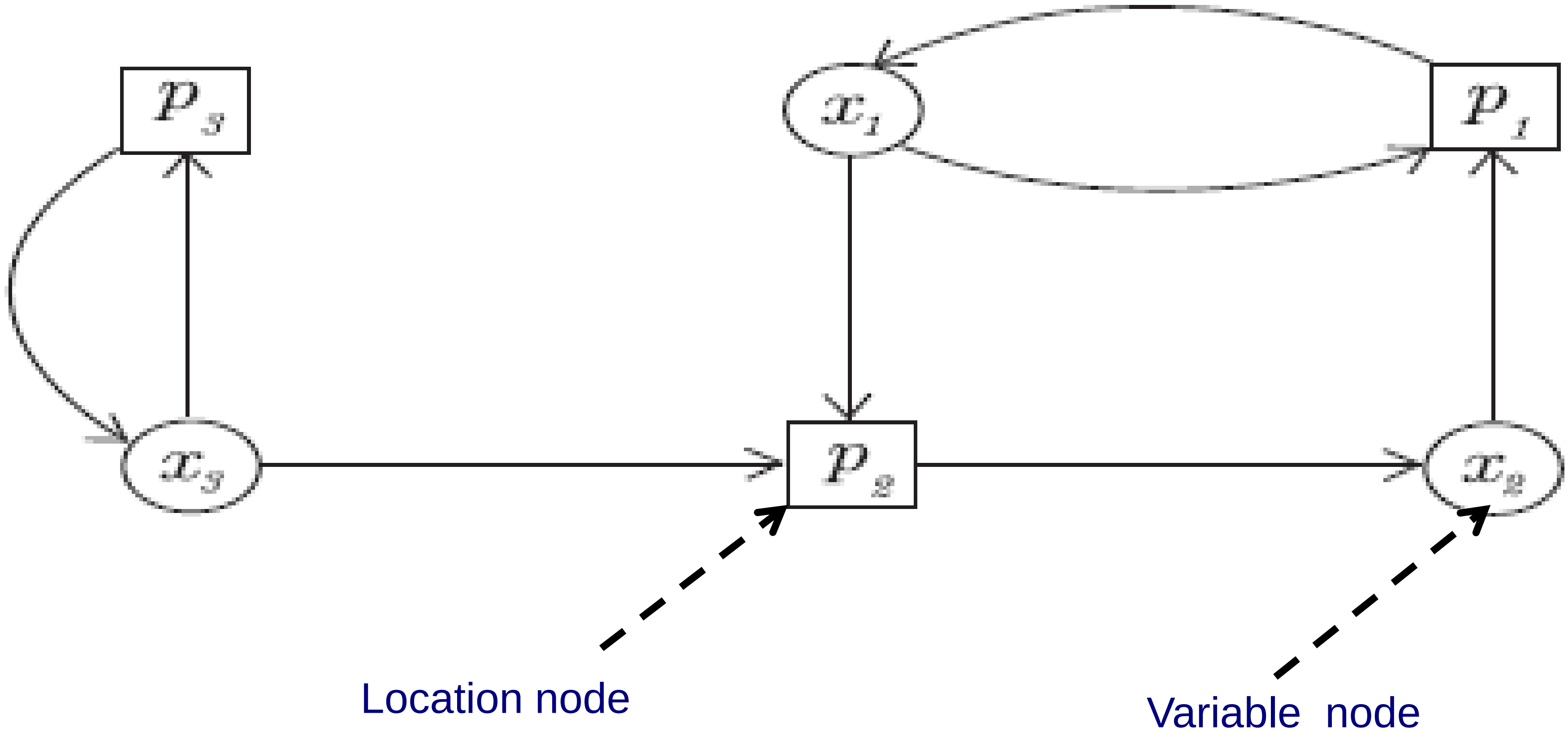
The size of **AUT** will be **exponential** in m.

Can we do better?


Network of HAs
Global HA
Syntactic Product
Local discretization
Discretization
Product
Network of FSAs
Global FSA

Network of HAs





For each node, construct an FSA

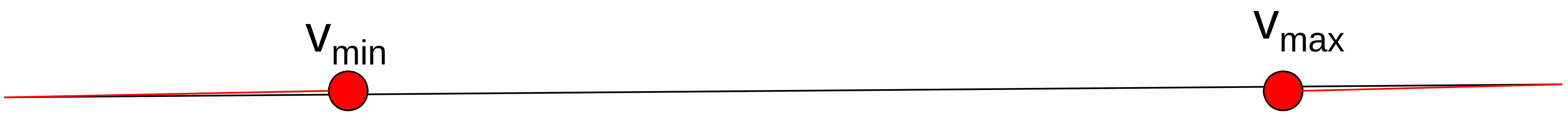
Each FSA will "read" from all its neighbor FSAs to make its moves.

Nbr(p) = Ctl(p) ∪ Obs(p)   Nbr(x) = {p | x ∈ Ctl(p) ∪ Obs(p) }

Aut_x

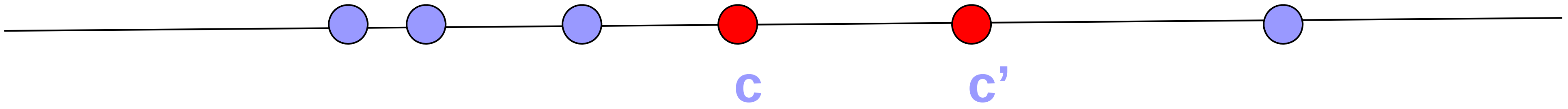
- Aut_x will keep track of the current value of x
- $CTL(x) = p$ if $x \in Ctl(p)$
- A move of Aut_x:
  - read the current rate of x from $Aut_{CTL(x)}$ and update the current value of x
- Can only keep bounded information
- Quotient the value space of x

Quotienting the value space of x



Quotienting the value space of x

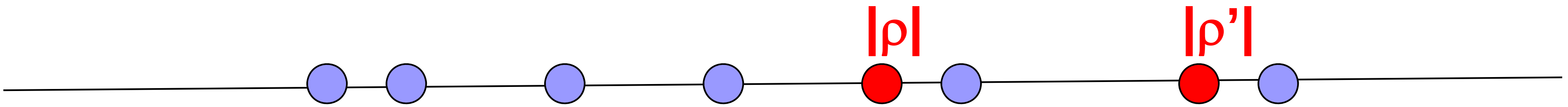
Quotienting the value space of x



c, c' , .., . . ., the constants that appear in some guard

Quotienting the value space of x

ρ, ρ' ….. rates of x associated with modes in $AUT_{CTL(x)}$

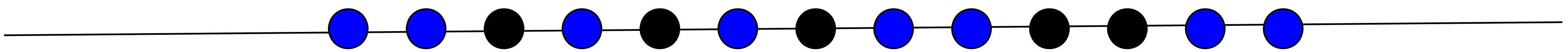


Find the largest positive rational that evenly divides all these rationals

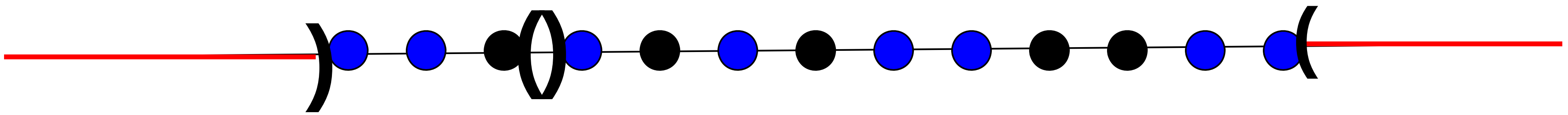
.

Use it to divide $[v_{min}, v_{max}]$ into uniform intervals

Quotienting the value space of x

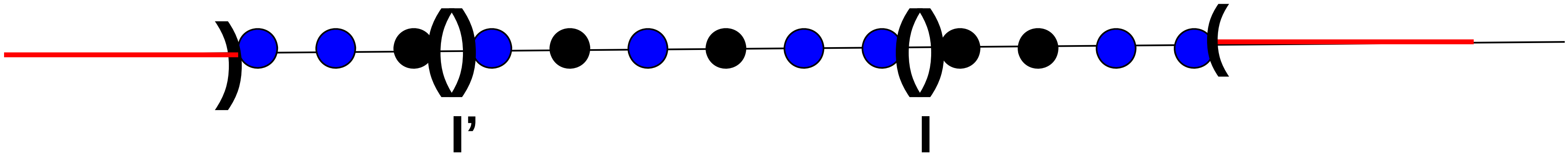


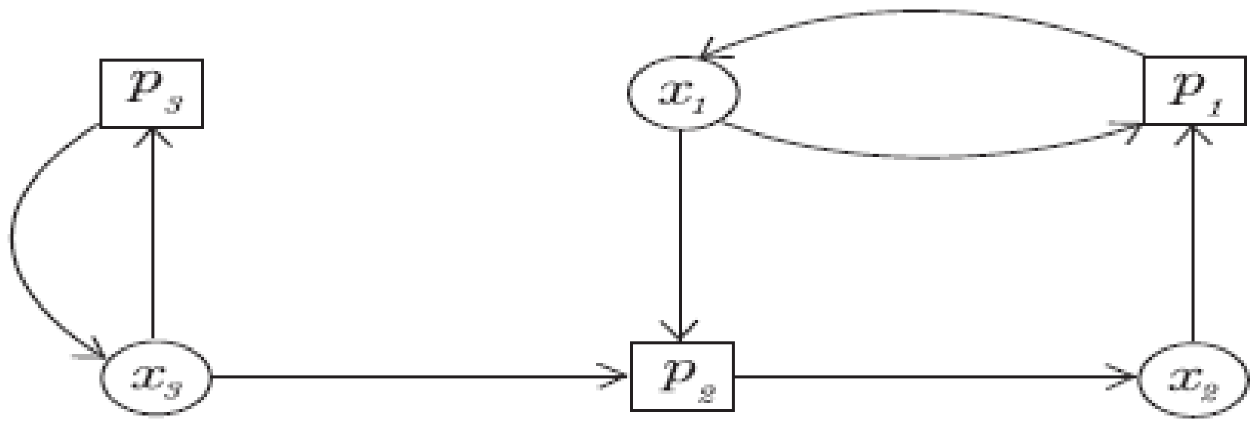
Quotionting the value space of x

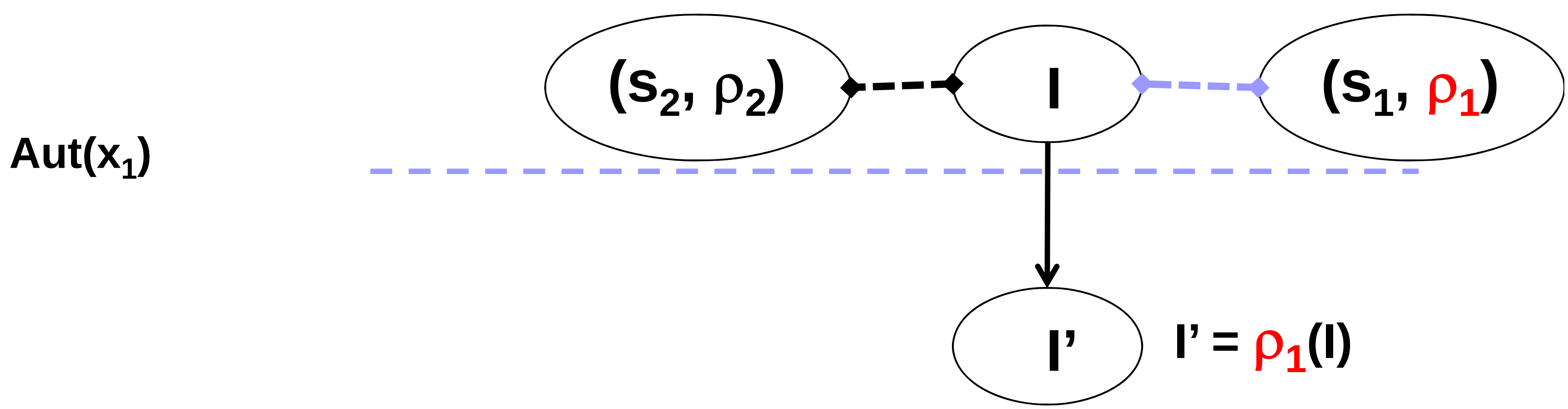


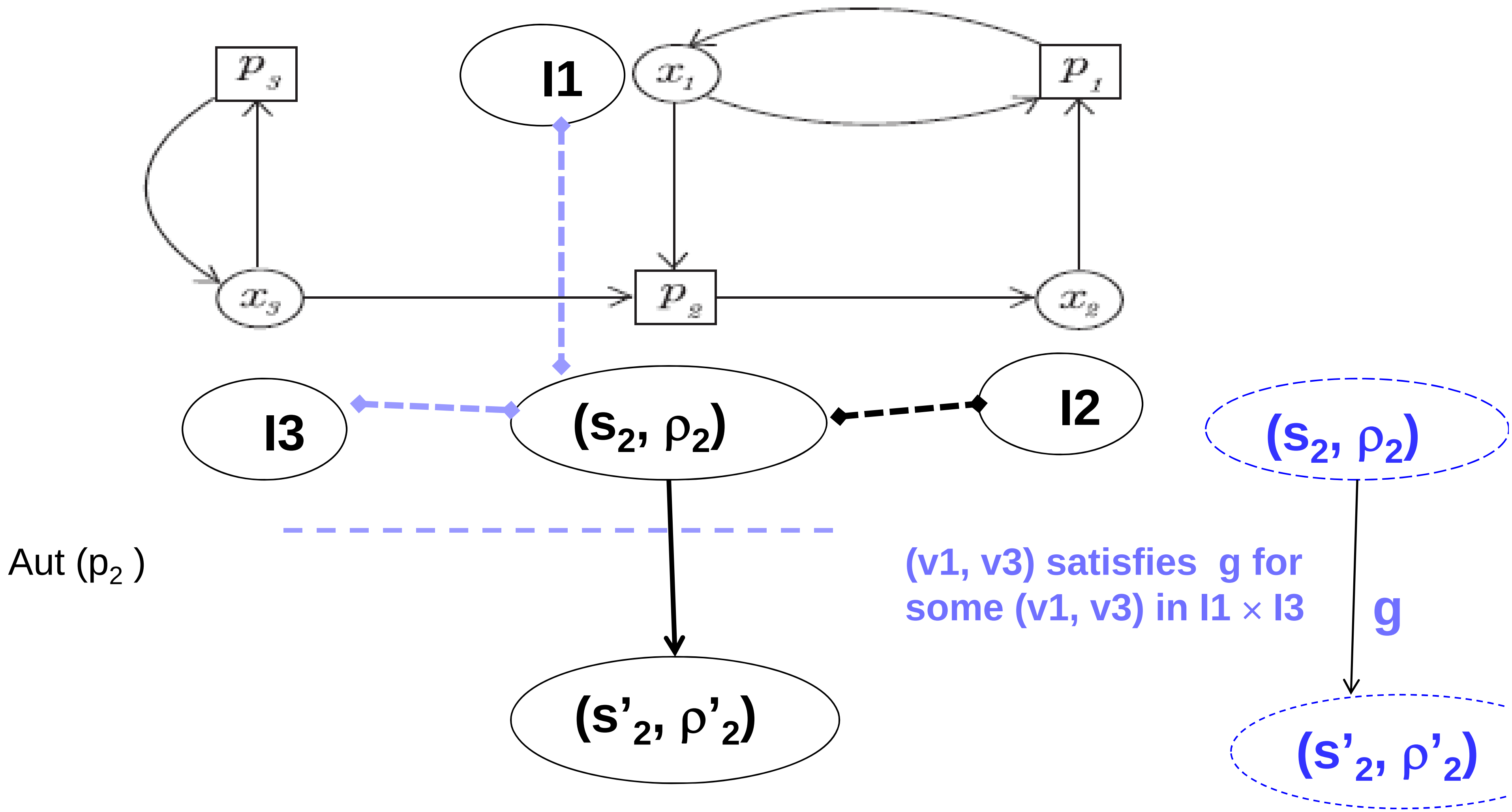
A move of Aut_x:

If Aut_x is in state I and CTL(x) = p and Aut_p's state is ρ then
Aut_x moves from I to I' = ρ(I)

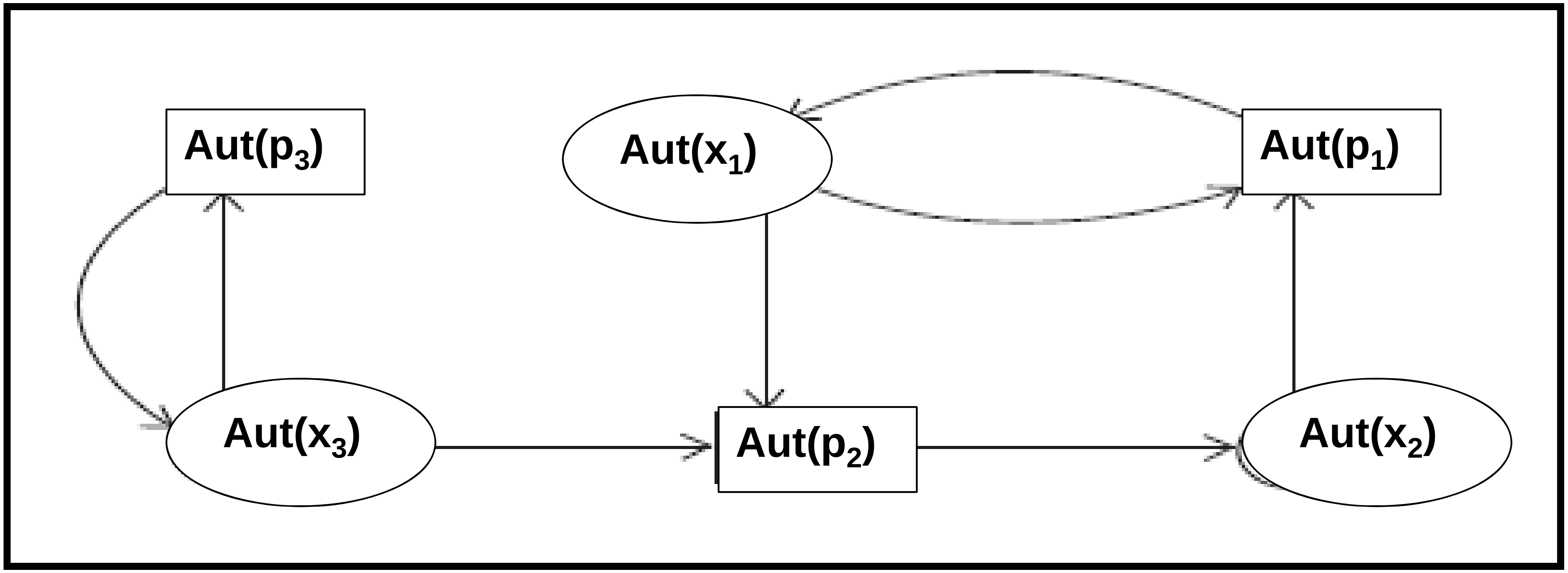

I'
I


p_3
x_1
p_1
x_3
p_2
x_2


(s_2, ρ_2)
I
(s_1, ρ_1)
$Aut(x_1)$
I'
$I' = \rho_1(I)$



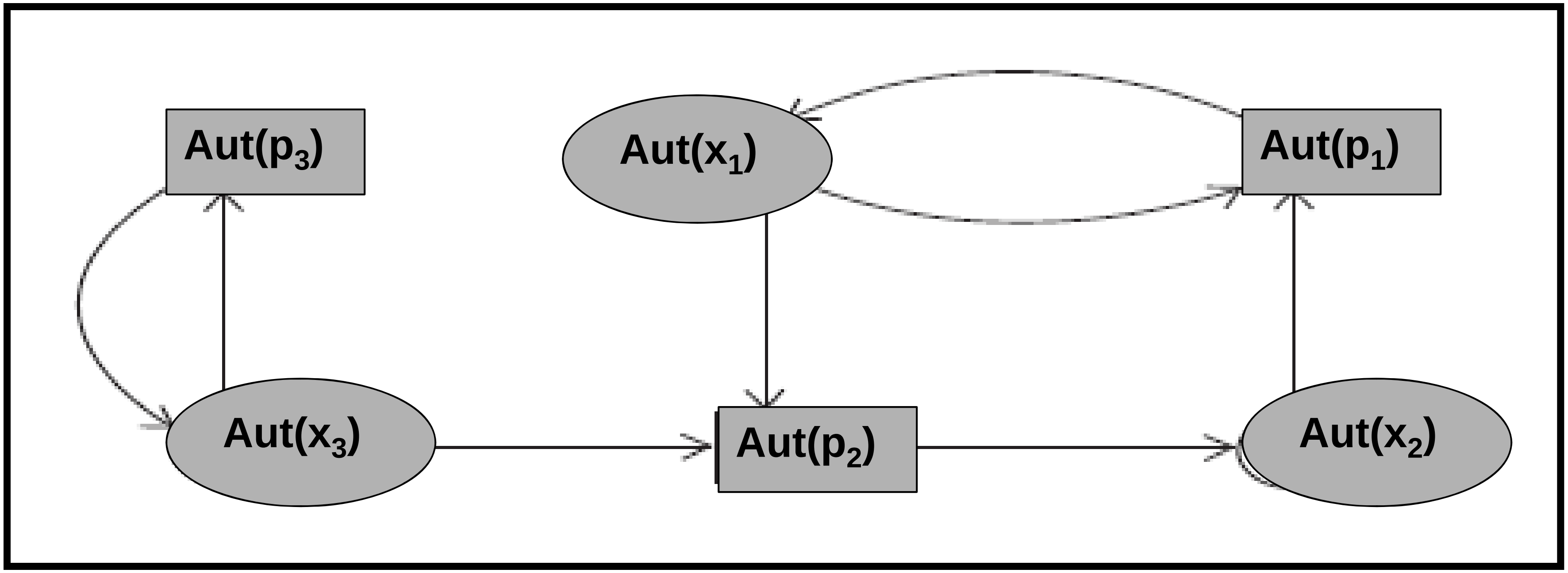
p_3
I1
x_1
p_1
x_3
p_2
x_2
I3
(s_2, ρ_2)
I2
(s_2, ρ_2)
Aut (p_2)
(v1, v3) satisfies g for some (v1, v3) in I1 × I3
g
(s'_2, ρ'_2)
(s'_2, ρ'_2)

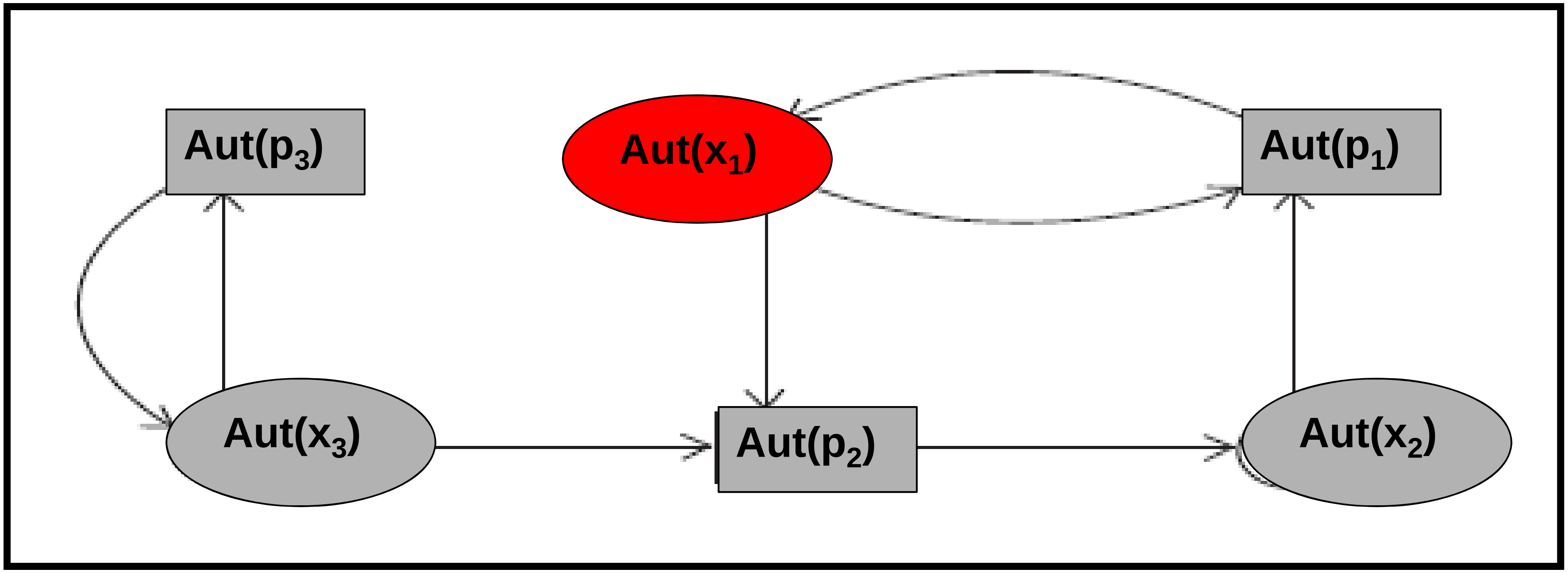


Each automaton will have a parity bit. This bit flips every time the automaton makes a move. Initially all the parities are 0.

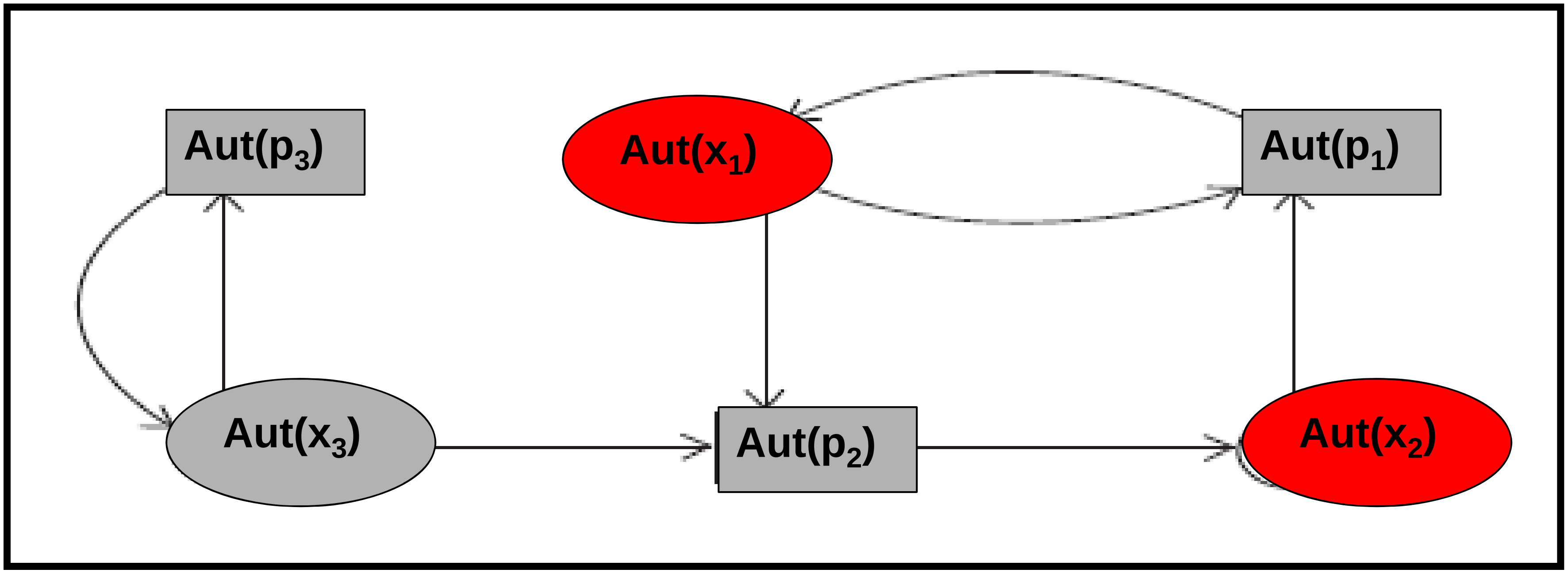
A variable node automaton makes a move only when its parity is the same as all its neighbors'

A location node automaton makes a move only when its parity is different from all its neighbors.

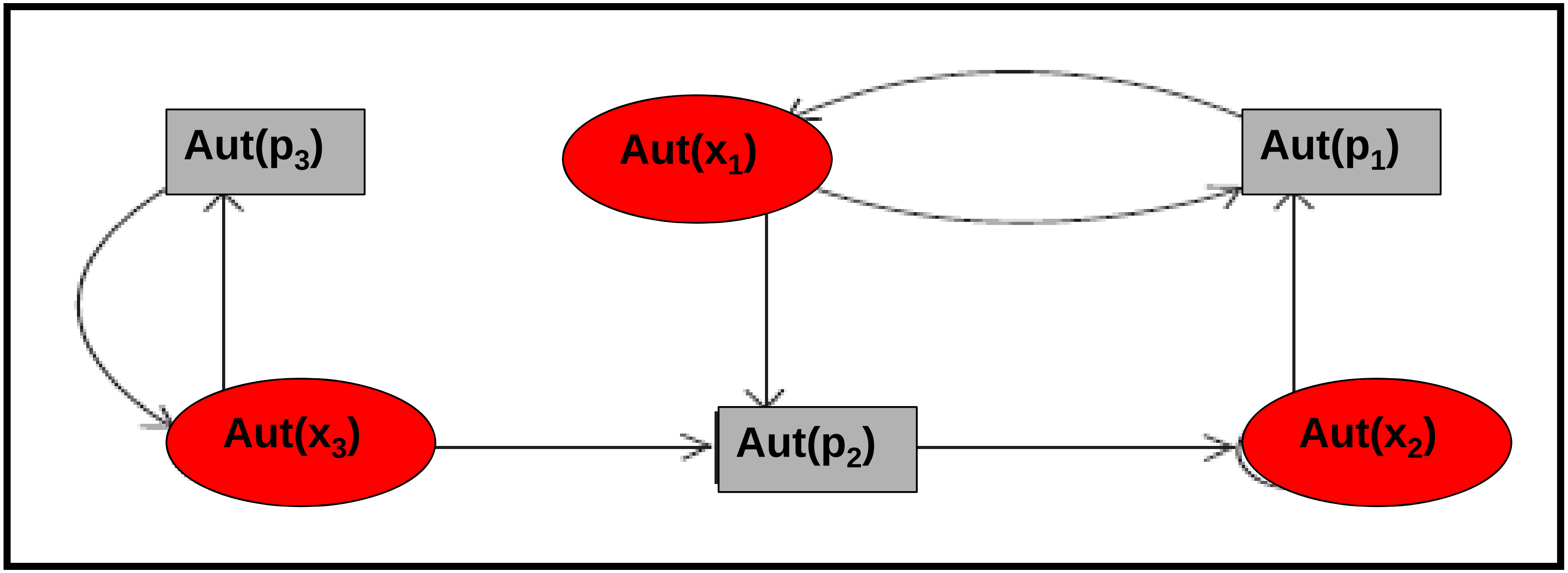

Aut(p_3)
Aut(x_1)
Aut(p_1)
Aut(x_3)
Aut(p_2)
Aut(x_2)



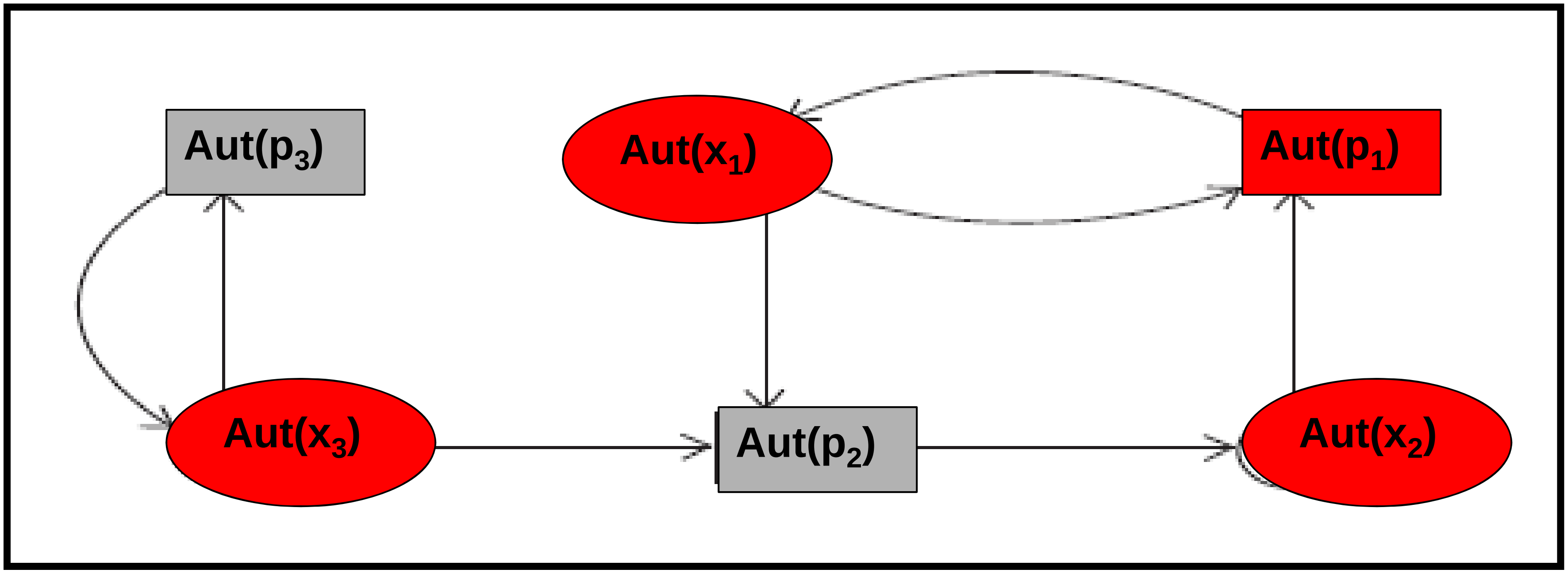
Aut(p3)
Aut(x1)
Aut(p1)
Aut(x3)
Aut(p2)
Aut(x2)



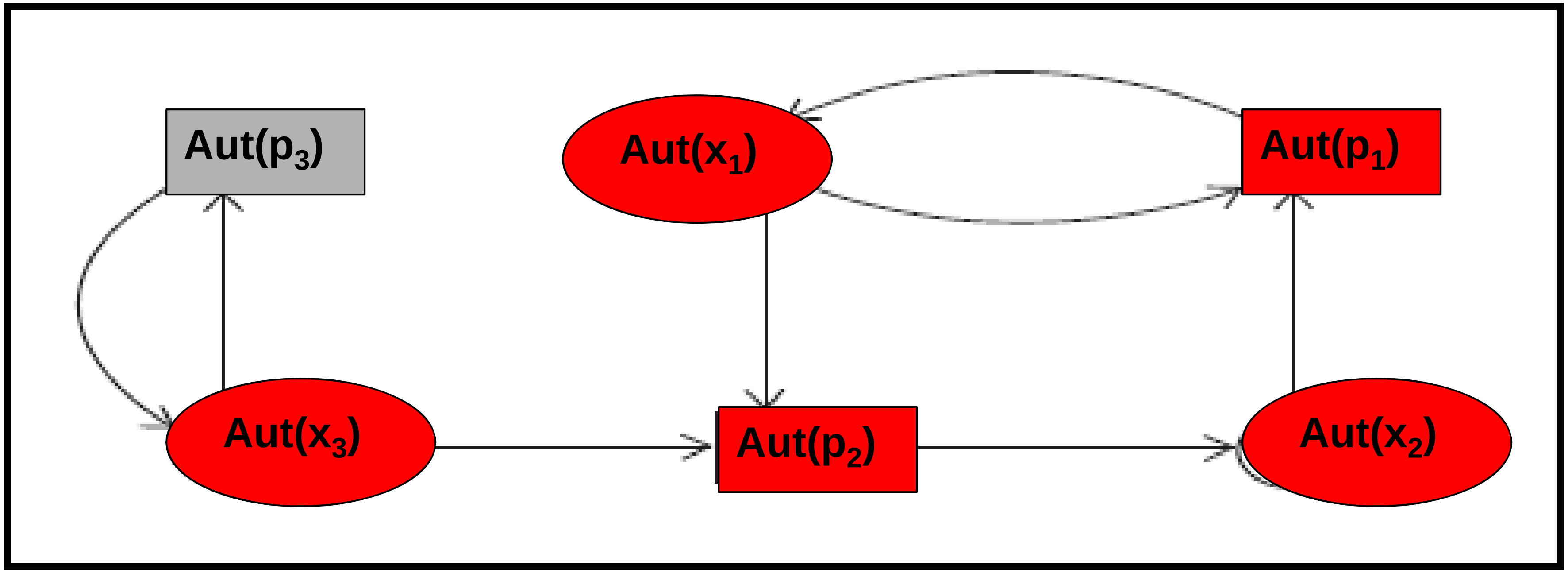
$Aut(p_3)$
$Aut(x_1)$
$Aut(p_1)$
$Aut(x_3)$
$Aut(p_2)$
$Aut(x_2)$



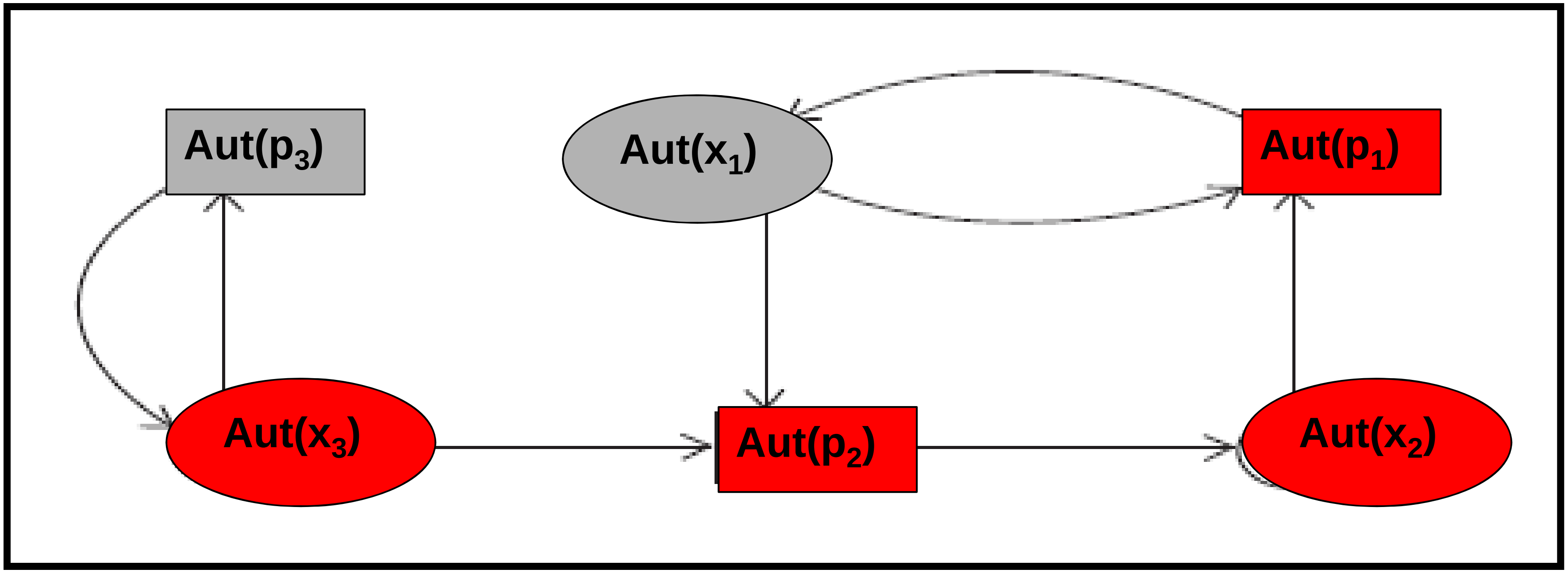
Aut(p3)
Aut(x1)
Aut(p1)
Aut(x3)
Aut(p2)
Aut(x2)



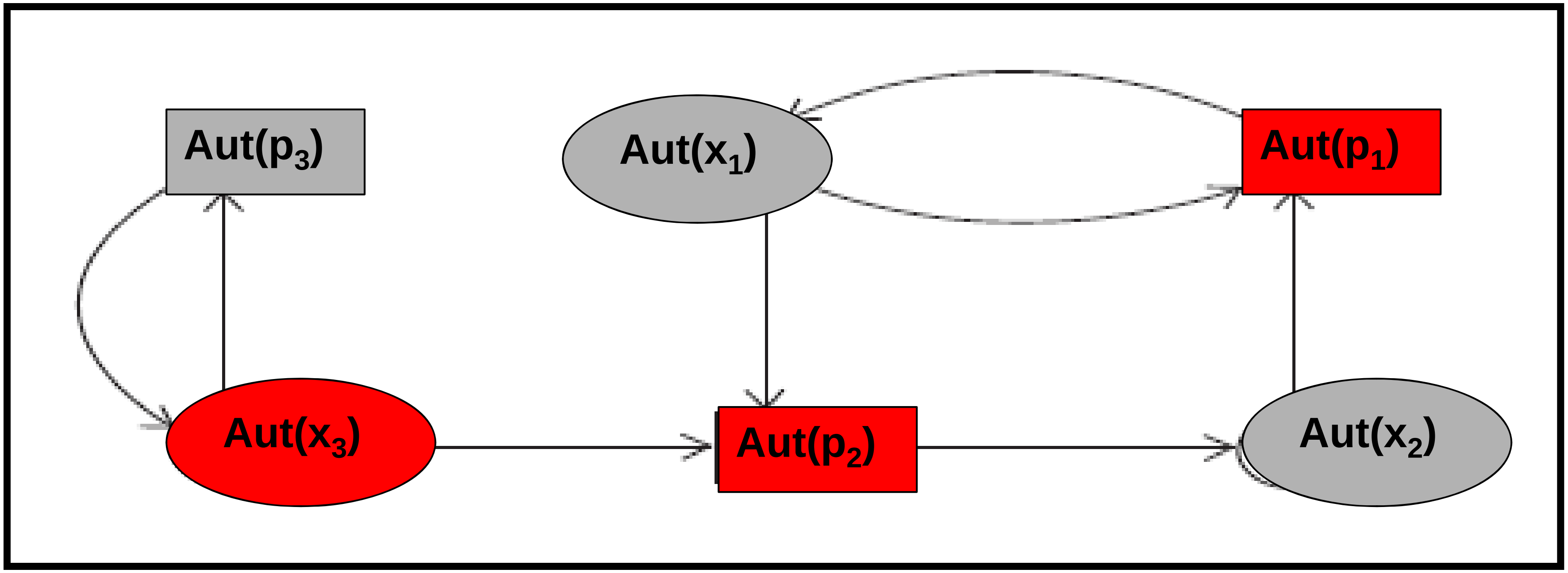
Aut(p3)
Aut(x1)
Aut(p1)
Aut(x3)
Aut(p2)
Aut(x2)

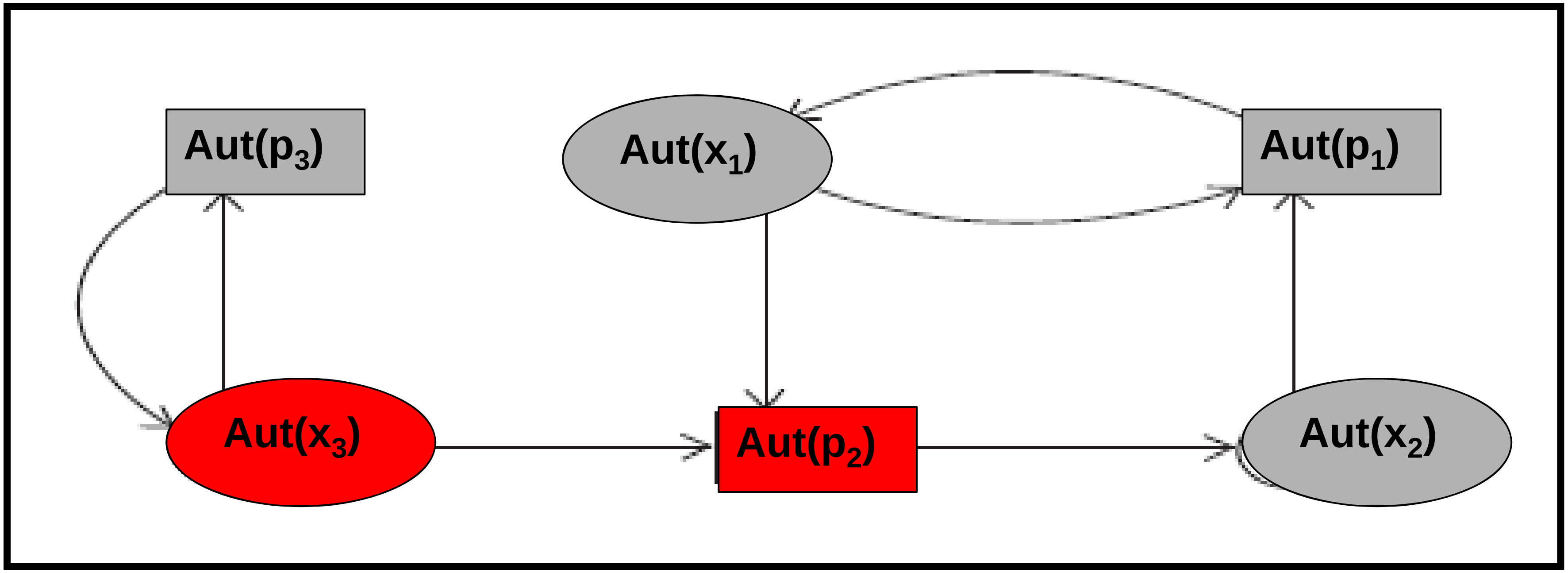


Aut(p_3)
Aut(x_1)
Aut(p_1)
Aut(x_3)
Aut(p_2)
Aut(x_2)

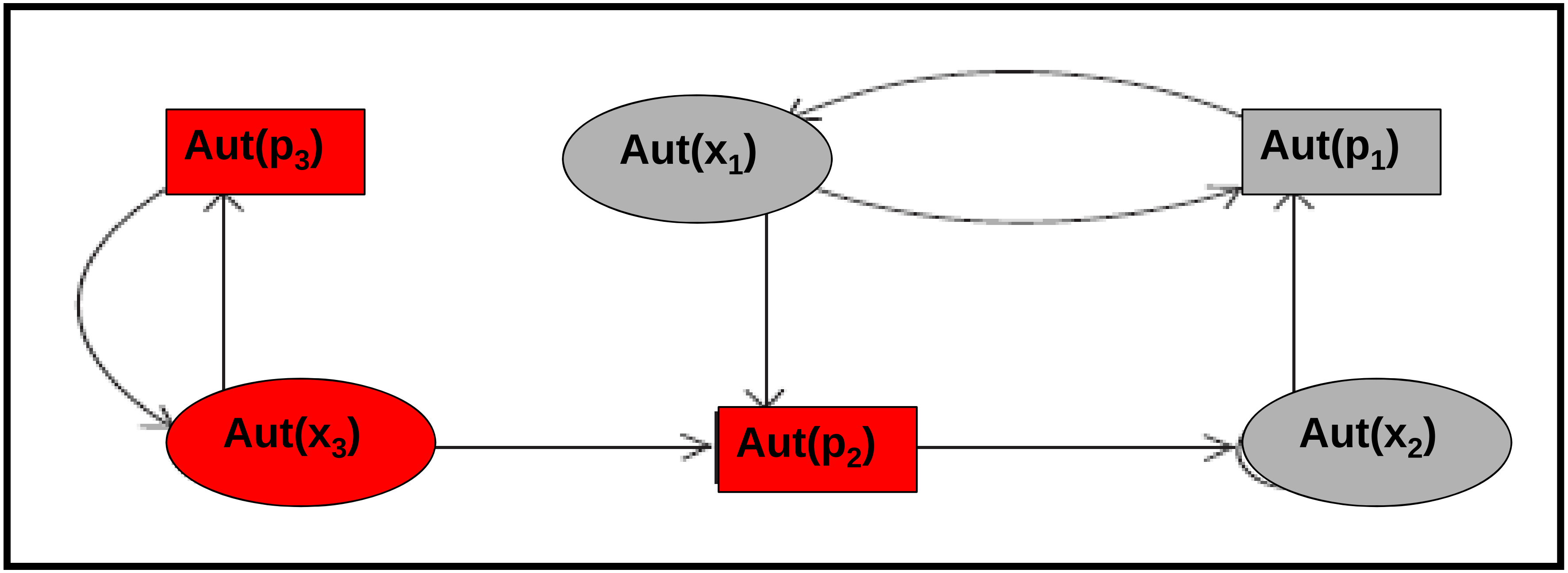


Aut(p_3)
Aut(x_1)
Aut(p_1)
Aut(x_3)
Aut(p_2)
Aut(x_2)

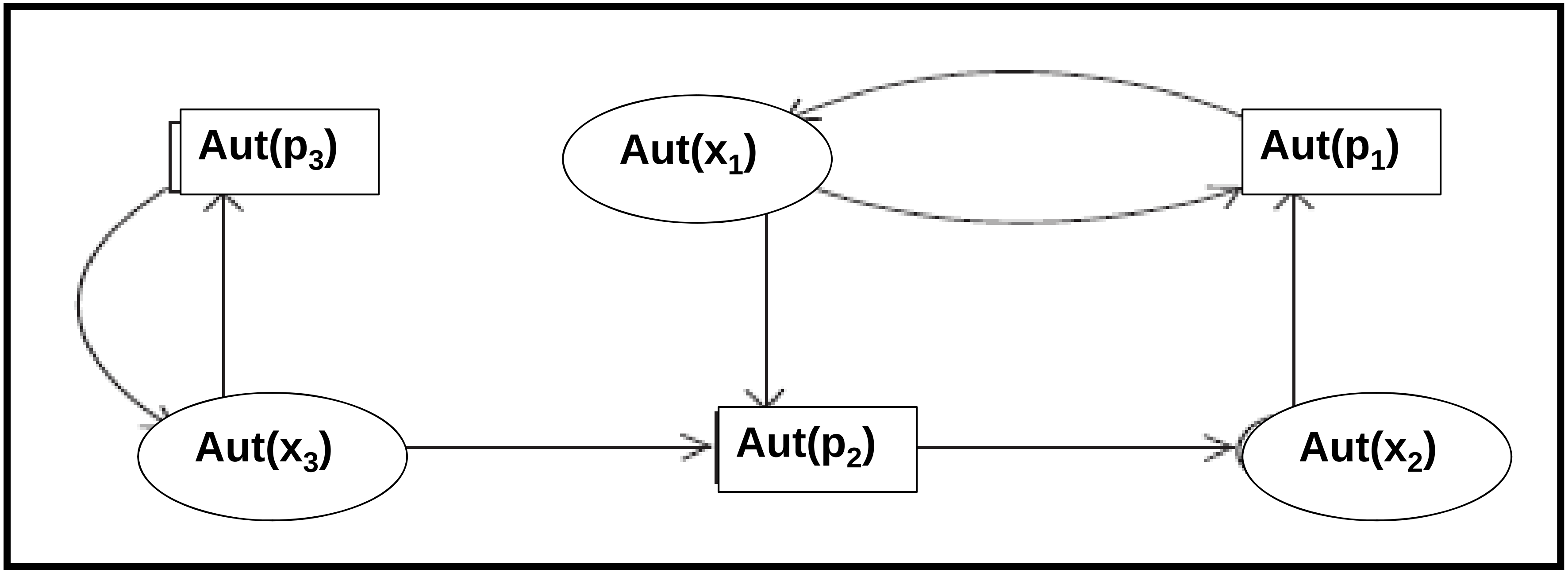

Aut(p3)
Aut(x1)
Aut(p1)
Aut(x3)
Aut(p2)
Aut(x2)



Aut(p_3)
Aut(x_1)
Aut(p_1)
Aut(x_3)
Aut(p_2)
Aut(x_2)



The automata can ‘drift” in time steps.



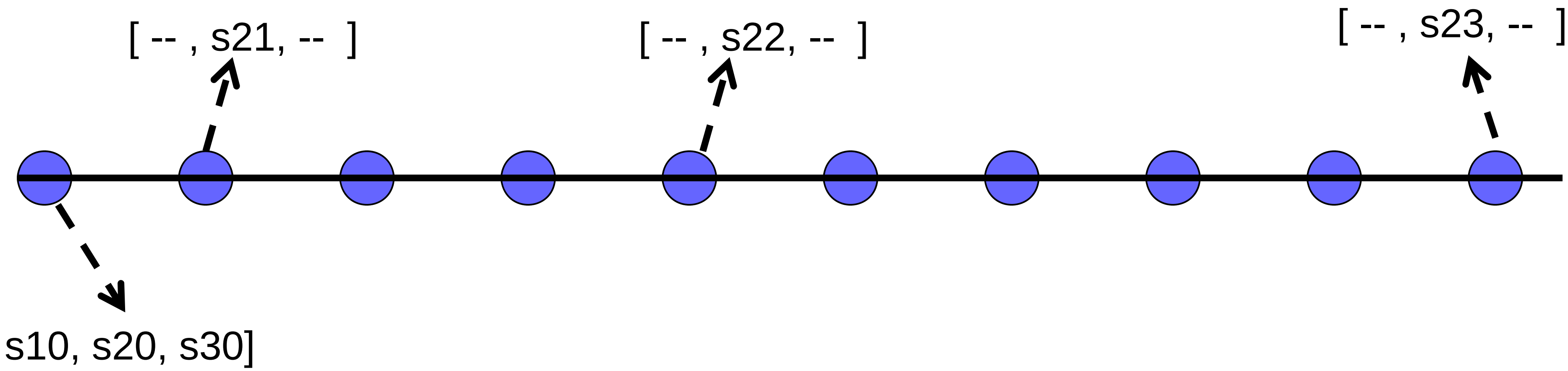
Aut ------ The asynchronous product of
{ **Aut(x_1), Aut(p_1), Aut(x_2), Aut(p_2), Aut(x_3), Aut(p_3) }**

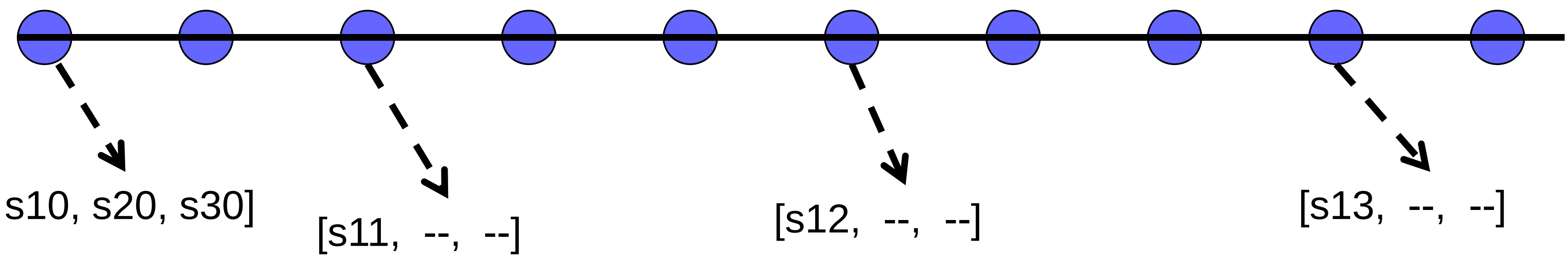
Each global state of Aut will induce a global state of *DHA*.

[(l1, (s1, ρ1), l2, (s2, ρ2), l3, (s3, ρ3)] $\xrightarrow{f}$ **[s1, s2, s3]**

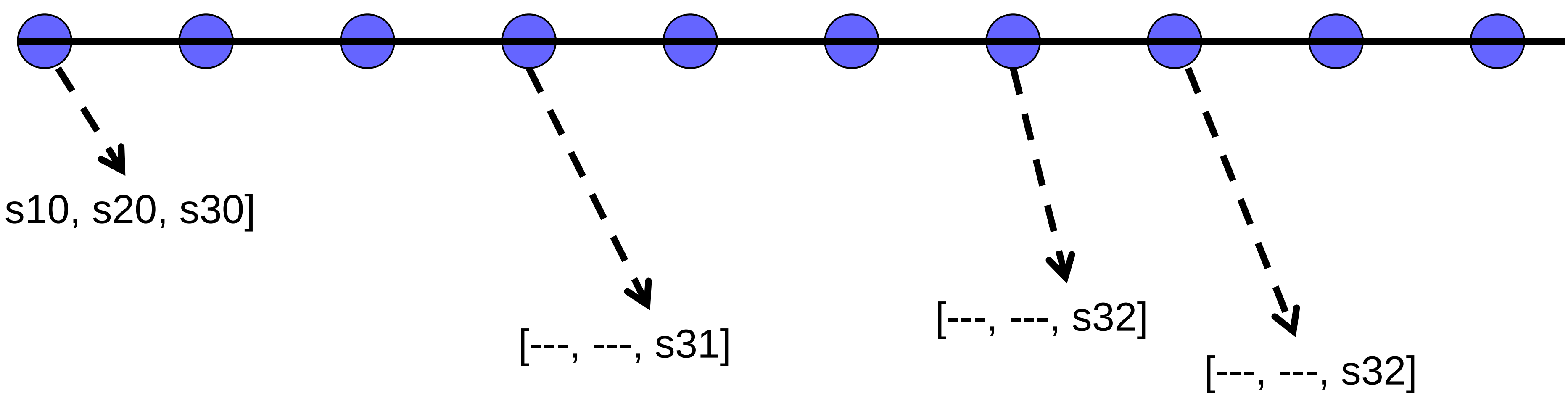
In fact, each *complete state sequence* of Aut will induce a global state sequence of *DHA.*

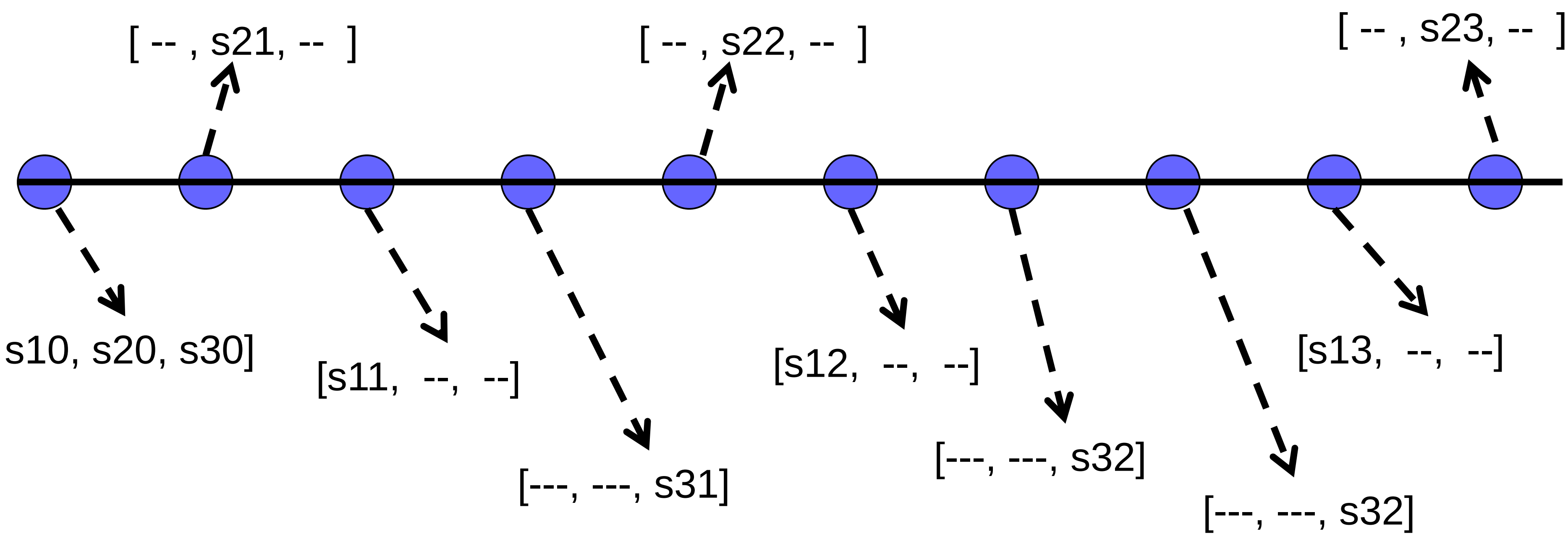
A state sequence τ of Aut is ***complete*** iff all the FSA make an equal number of moves along τ.





s10, s20, s30]
[s11, --, --]
[s12, --, --]
[s13, --, --]


s10, s20, s30]
[---, ---, s31]
[---, ---, s32]
[---, ---, s32]



[s10, s20, s30] [s11, s21, s31] [s12, s22, s32] [s13, s23, s33]

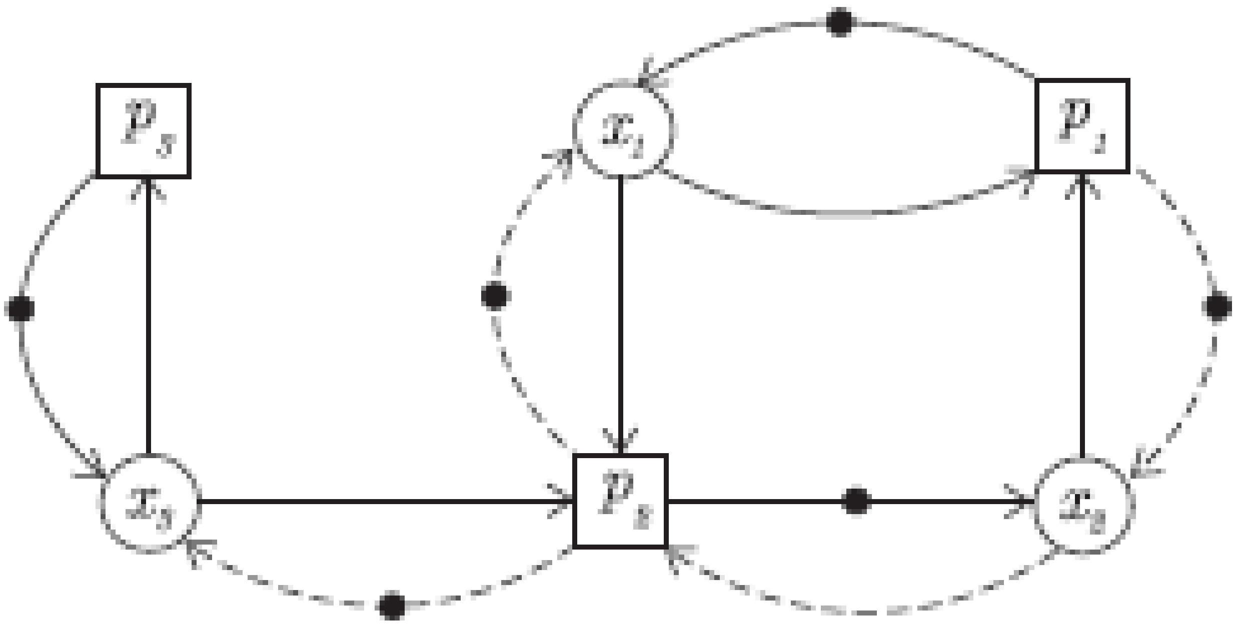
The main results

- Suppose τ is a complete state sequence of Aut. Then f(τ) is a global state sequence of *DHA*.
- If σ is global state sequence of *DHA* then there exists a complete sequence τ of Aut such that f(τ) = σ.
- In the absence of deadlocks, every state sequence of Aut can be extended to a complete state sequence.

Extensions

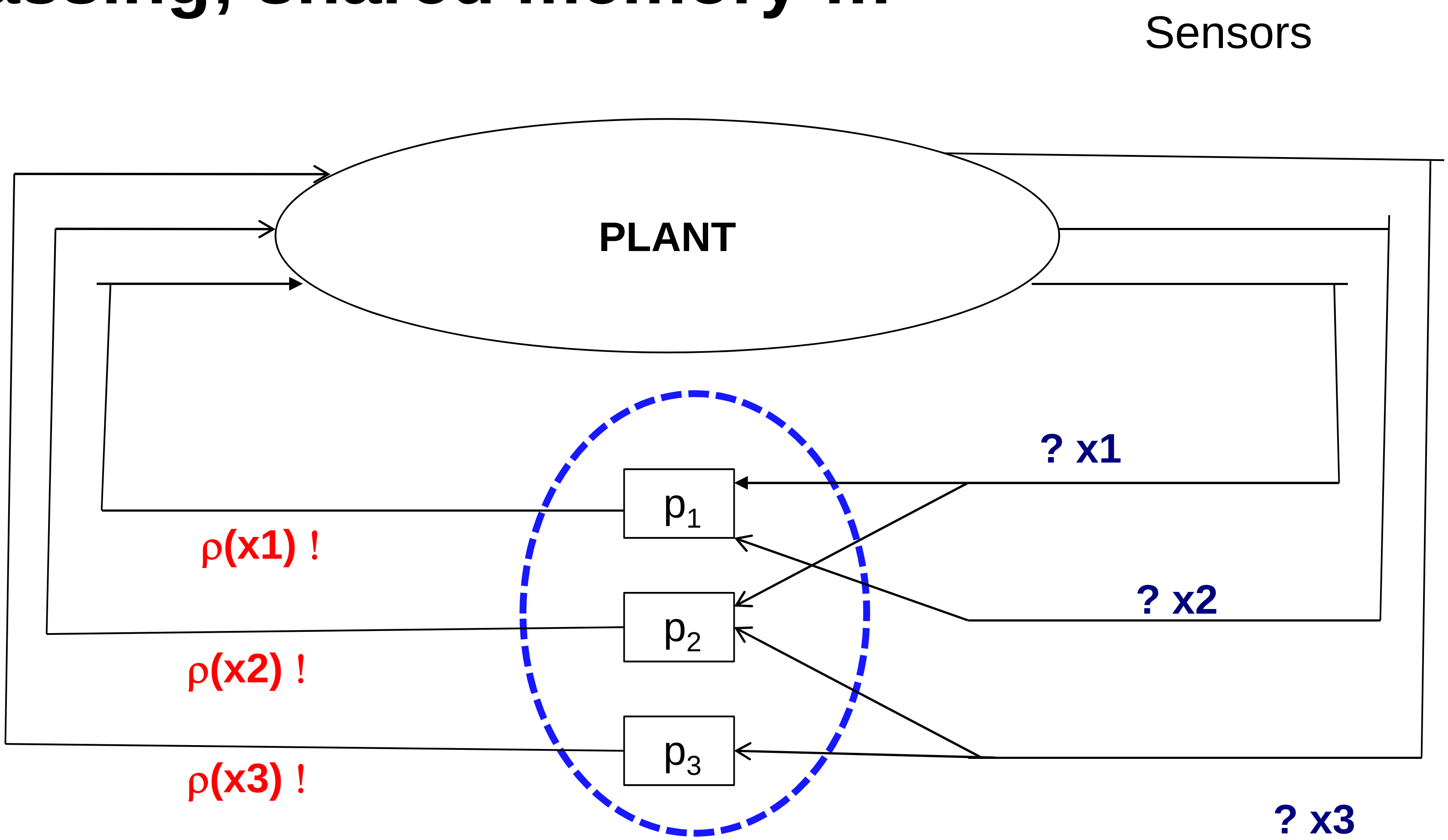
- Laziness
- Delays in communictions between the plant and controllers.
- Different granularities of time for the controllers
- …….

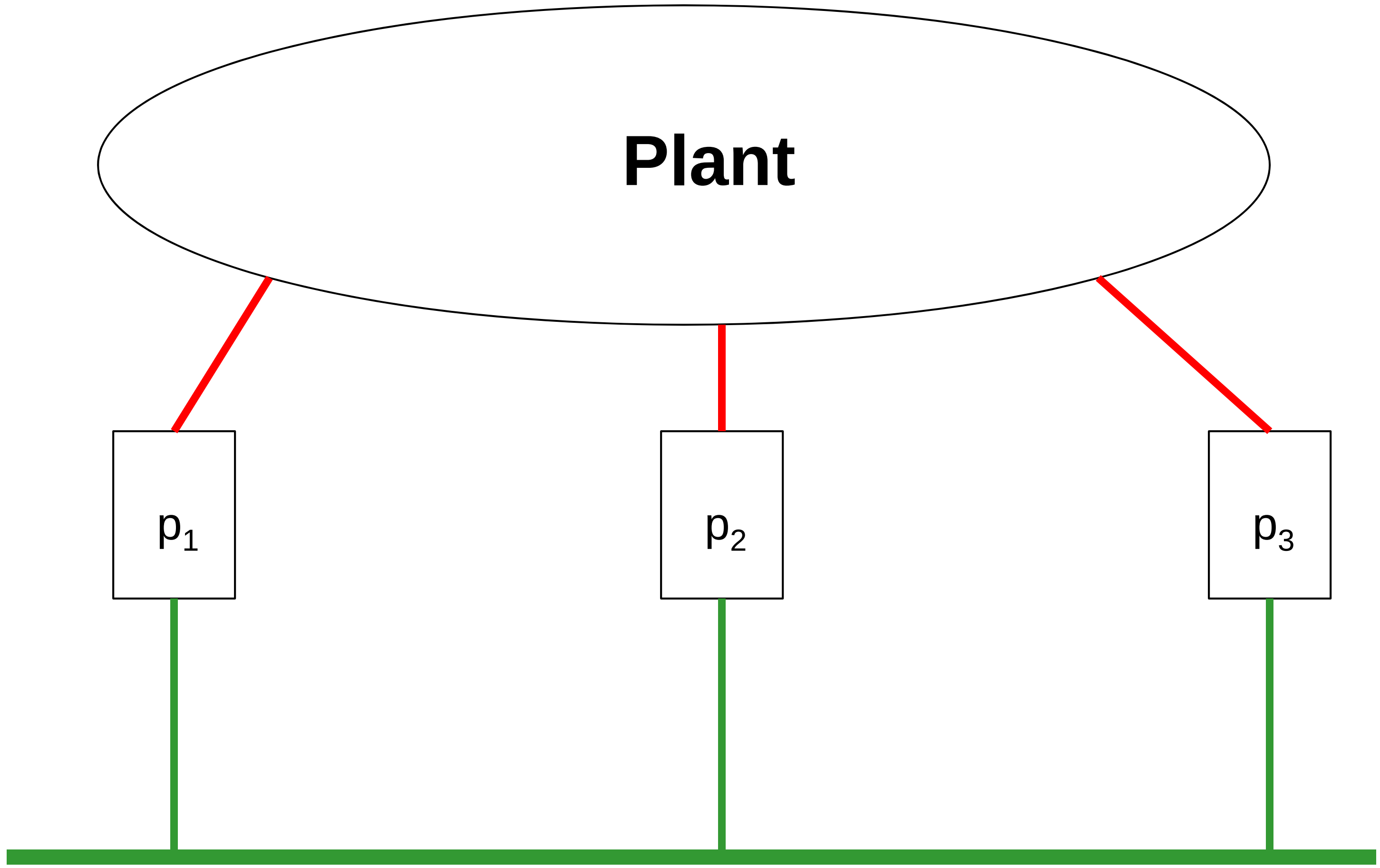
The marked graph connection



Can be used to derive partial order reduction verification algorithms.

Communicating hybrid automata: Synchronize on common actions; message passing; shared memory …





Time-triggered protocol;
Each controller is implemented on an ECU
Study interplay between plant dynamics and the performance of the computational platform

Summary

- The discrete time behavior of distributed hybrid automata can be succinctly represented.
  - as a network of FSA (communicating as in asynchronous cellular automata).
  - many extensions possible.
  - ***Finite precision*** assumption can yield stronger results.