NSMC 2003

On the exact simulation of functionals of stationary Markov chains

J-M. Vincent, C. Marchand

Decore-Imag and Apache-Inria Projects

ID-IMAG Laboratory

Universities of Grenoble

http://www-id.imag.fr

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

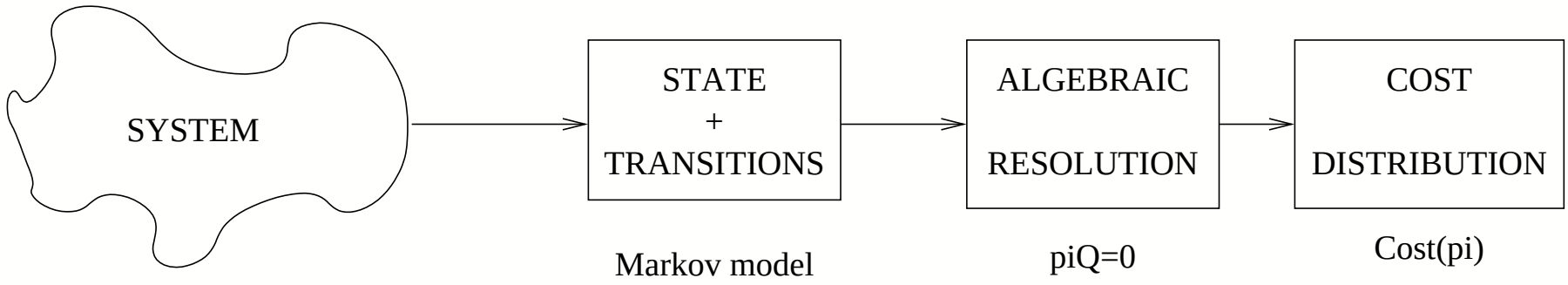
INP Grenoble

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

GRENOBLE 1 UNIVERSITÉ JOSEPH FOURIER SCIENCES,TECHNOLOGIE,MÉDECINE

Modeling discrete event systems

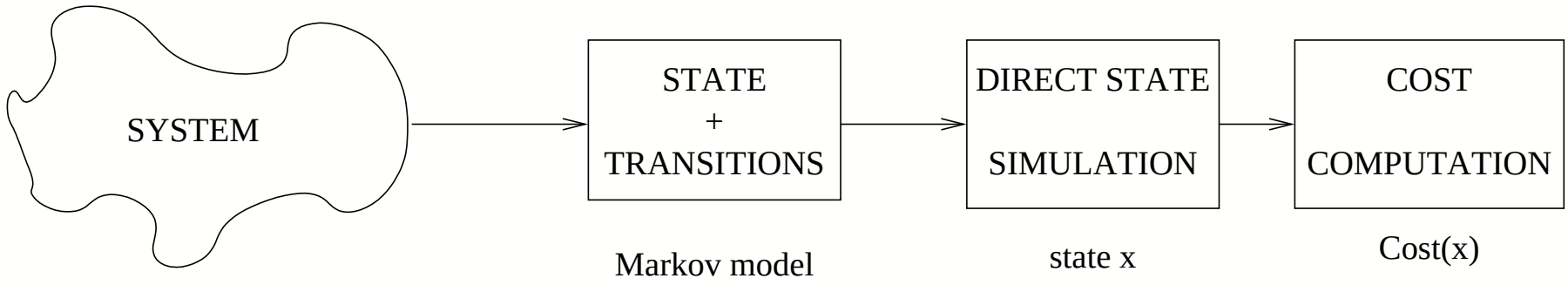


Difficulties:

- complex structure
- large state space
- analytical/numerical method
- approximation/bounding techniques

$\Rightarrow$ reduction of the state space

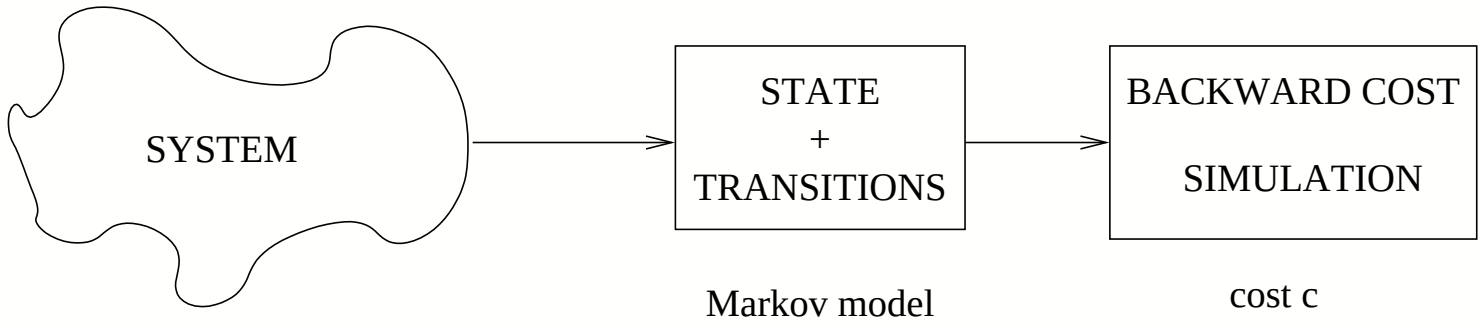
Modeling discrete event systems



Difficulties:

- stopping criteria : burn in time
- simulation biases $||\pi_n - \pi_\infty||$
- estimation biases : confidence intervals $\mathcal{O}(\frac{1}{\sqrt{n}})$

Modeling discrete event systems



Properties:

- Exact stopping criteria

  $\Rightarrow$ no simulation bias

Constraints:

- N parallel trajectories

Iterated system of functions

$\mathcal{X}$ state space (size n); $\mathcal{U}$ set of external input values
Transition function Φ

$$\begin{array}{rccc} \Phi: & \mathcal{X}\times\mathcal{U} & \longrightarrow & \mathcal{X} \\ & (x,u) & \longmapsto & \Phi(x,u) \end{array}$$

If $\{U_n\}_{n\in\mathbb{Z}}$ is IID then

$$X_0 = x_0, \quad X_{n+1} = \Phi(X_n, U_{n+1})$$

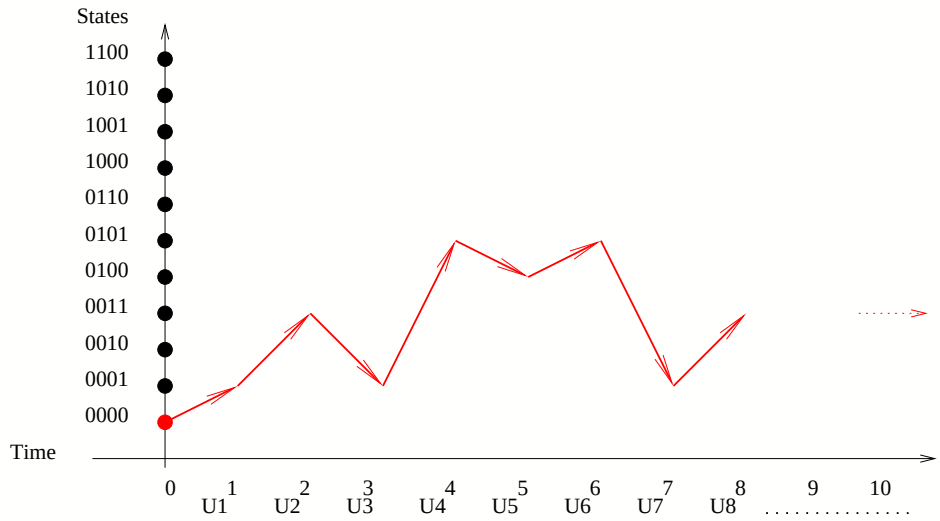
is a Markov chain (stochastic recursive sequence).

$$\{\Phi(.,u)\}_{u\in[0,1[}$$

is an iterated system of function.
Reciprocally, given a transition matrix Q it is possible to build a family of function $\Phi(.,u)$ such that the associated process is a markov chain with transition matrix Q.
$\Longrightarrow$ Simulation kernel

Problem : several representation of Q

Forward simulation



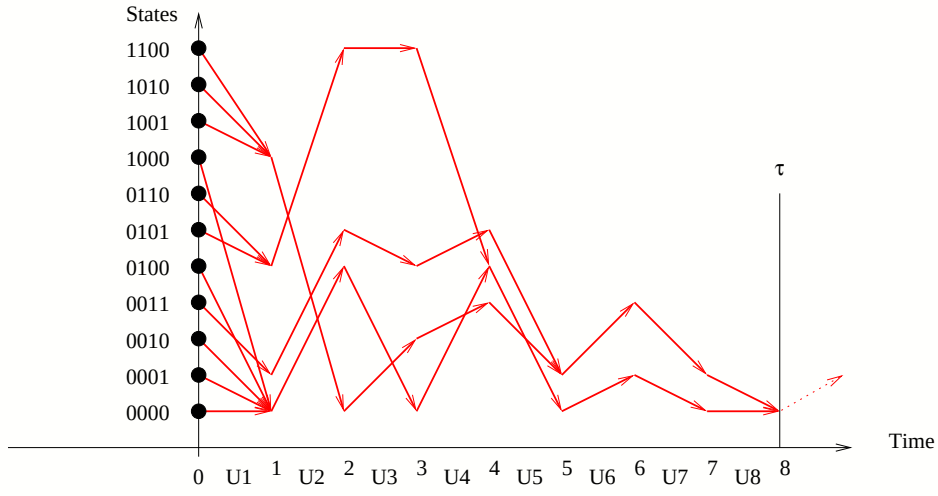
Forward simulation

$x \leftarrow x_0$;
repeat
  $u \leftarrow$ Random;
  $x \leftarrow \Phi(x, u)$;
until stopping criterium
return x
compute statistics

Problems :

- choice of the initial state
- choice of stopping criterium
- error computation (bias)

Forward coupling simulation



Forward coupling algorithm

for all $x \in \mathcal{X}$ **do**
  $y(x) \leftarrow x$
end for
repeat
  u $\leftarrow$ Random;
  for all $x \in \mathcal{X}$ **do**
    $y(x) \leftarrow \Phi(y(x), u)$;
  end for
until All $y(x)$ are equal
return $y(x)$

Problems :

- generated state $y(x)$

does not follow the stationary distribution

Coupling time τ

$$\tau = \min\{n \in \mathbb{N},\ |\Phi(\Phi(\cdots(\Phi(\mathcal{X}, U_1), \cdots), U_{n-1}), U_n)| = 1\}$$

- Is τ almost surely **finite** ? $\mathbb{P}(\tau < +\infty) = 1$?

Backward coupling simulation

Idea :

Propp & Wilson(1996)

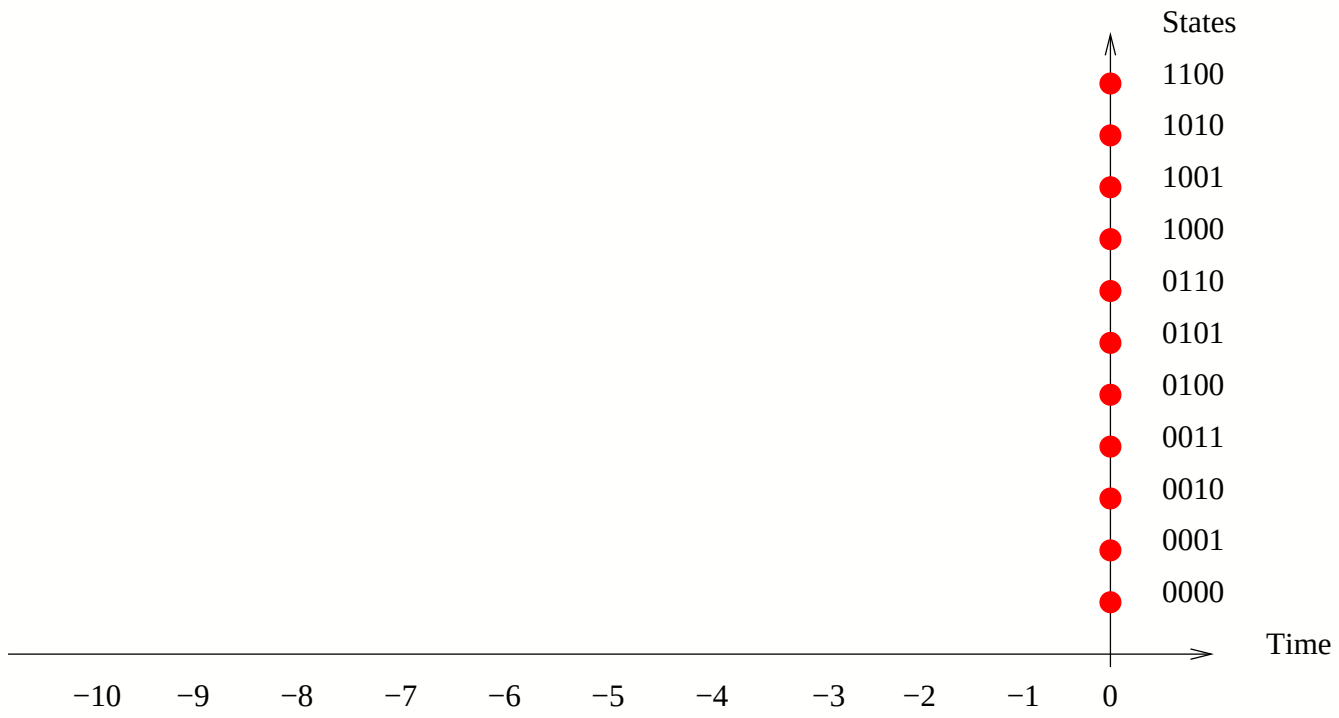
- reverse time
- run N parallel trajectories
- wait for coupling.

$$\mathcal{Z}_n = \Phi(\Phi(\cdots(\Phi(\mathcal{X}, U_{-n+1}), \cdots), U_{-1}), U_0).$$

potential set of reachable states at step n

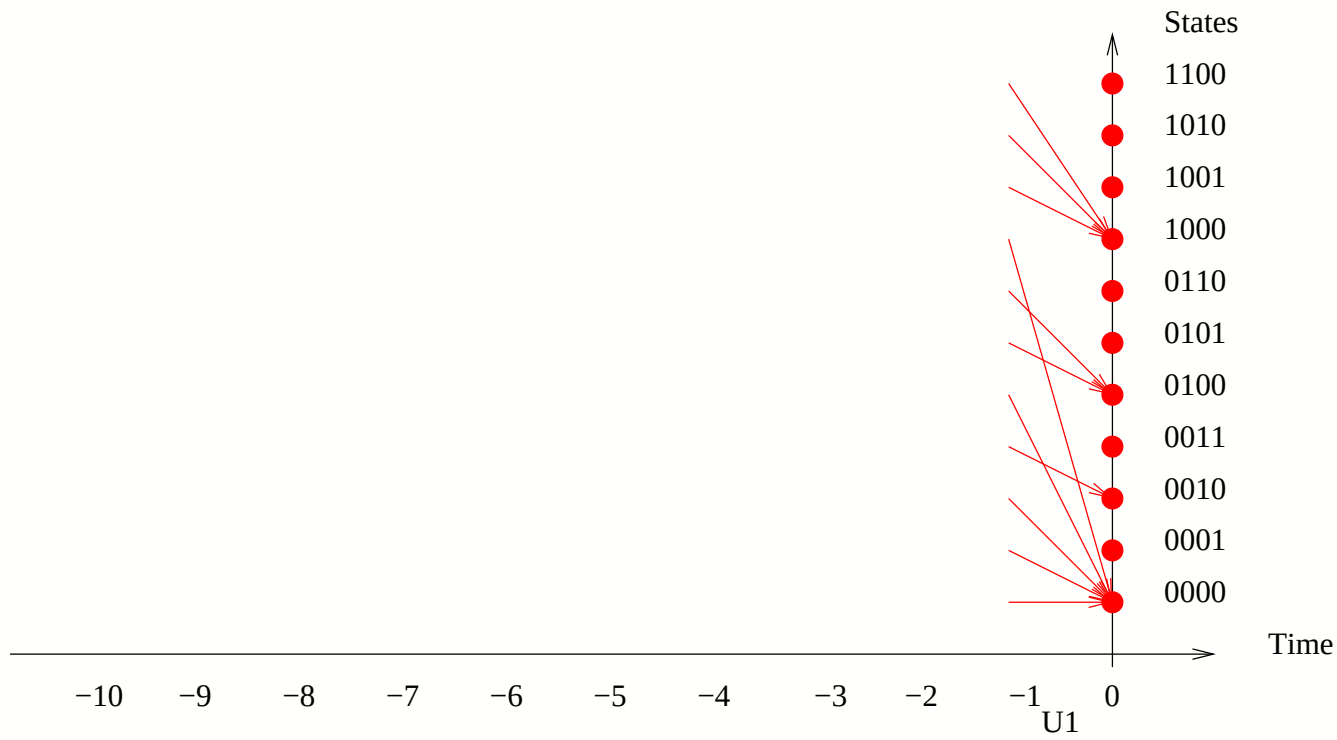
for all $x \in \mathcal{X}$ **do**
  $y(x) \leftarrow x$
end for
repeat
  u $\leftarrow$ Random;
  for all $x \in \mathcal{X}$ **do**
    $y(x) \leftarrow y(\Phi(x, u))$;
  end for
until All $y(x)$ are equal
return $y(x)$

Backward simulation example



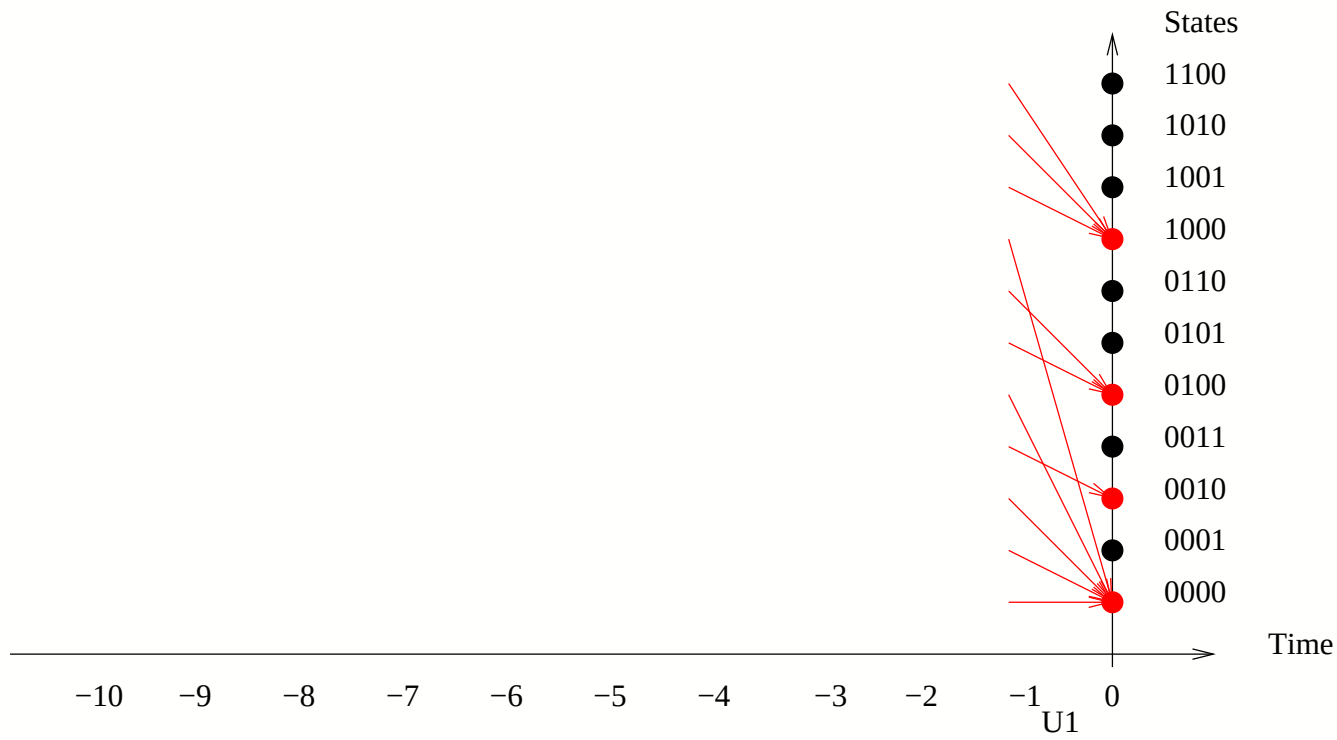
$$\mathcal{Z}_0 = \mathcal{X}$$

Backward simulation example



$$\mathcal{Z}_0 = \mathcal{X}$$

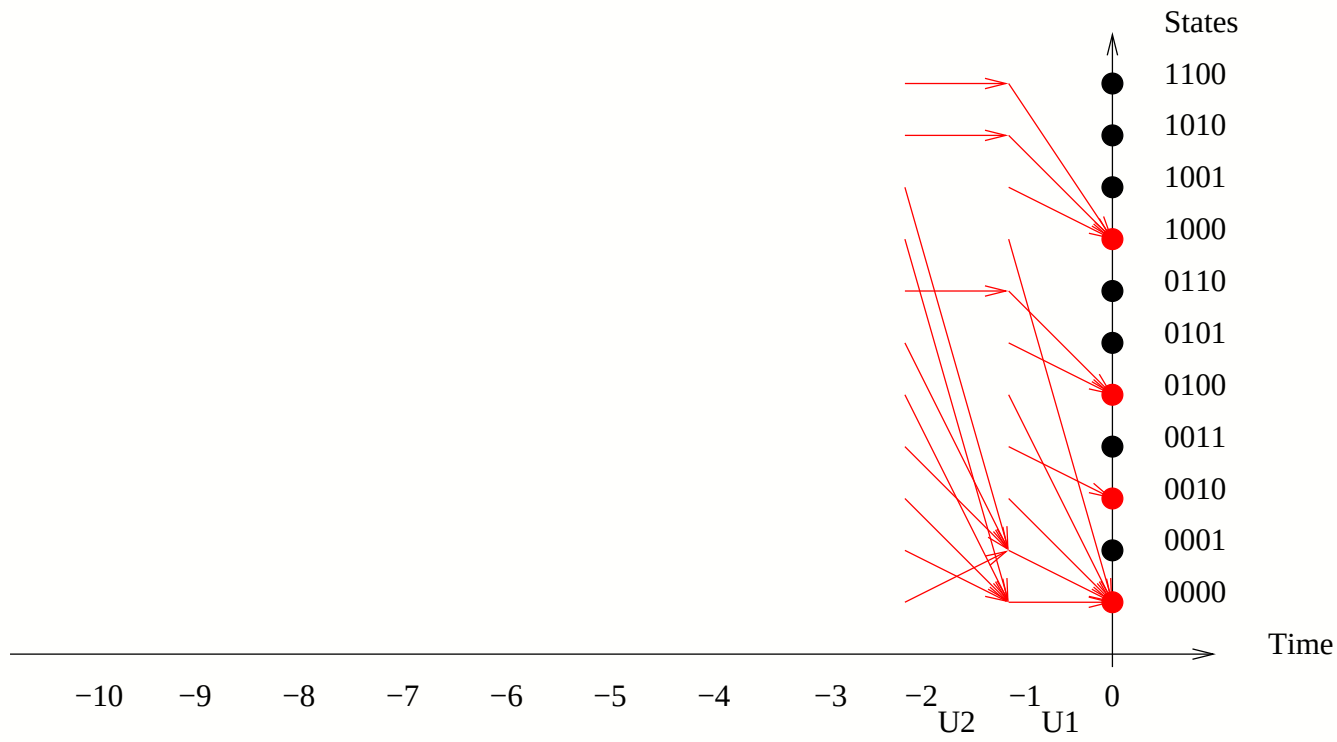
Backward simulation example



$$\mathcal{Z}_0 = \mathcal{X}$$

$$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$$

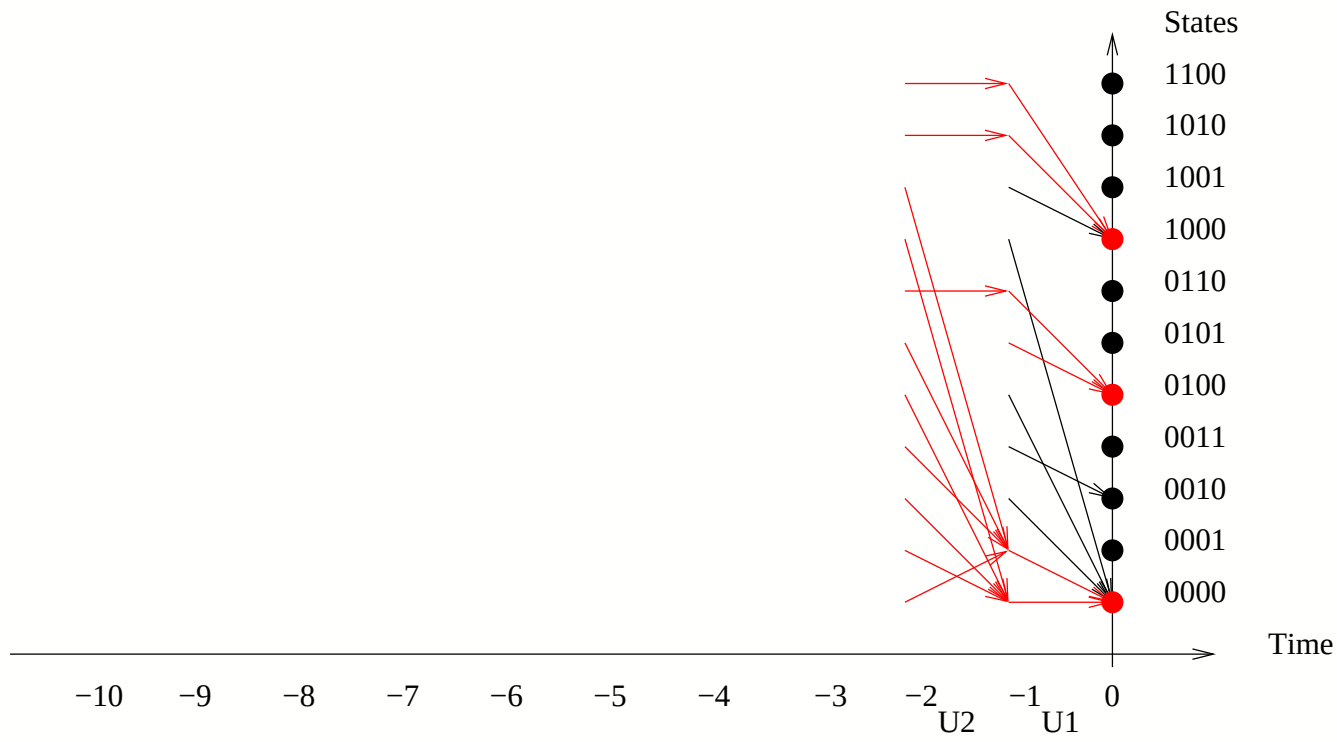
Backward simulation example



$$\mathcal{Z}_0 = \mathcal{X}$$

$$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$$

Backward simulation example

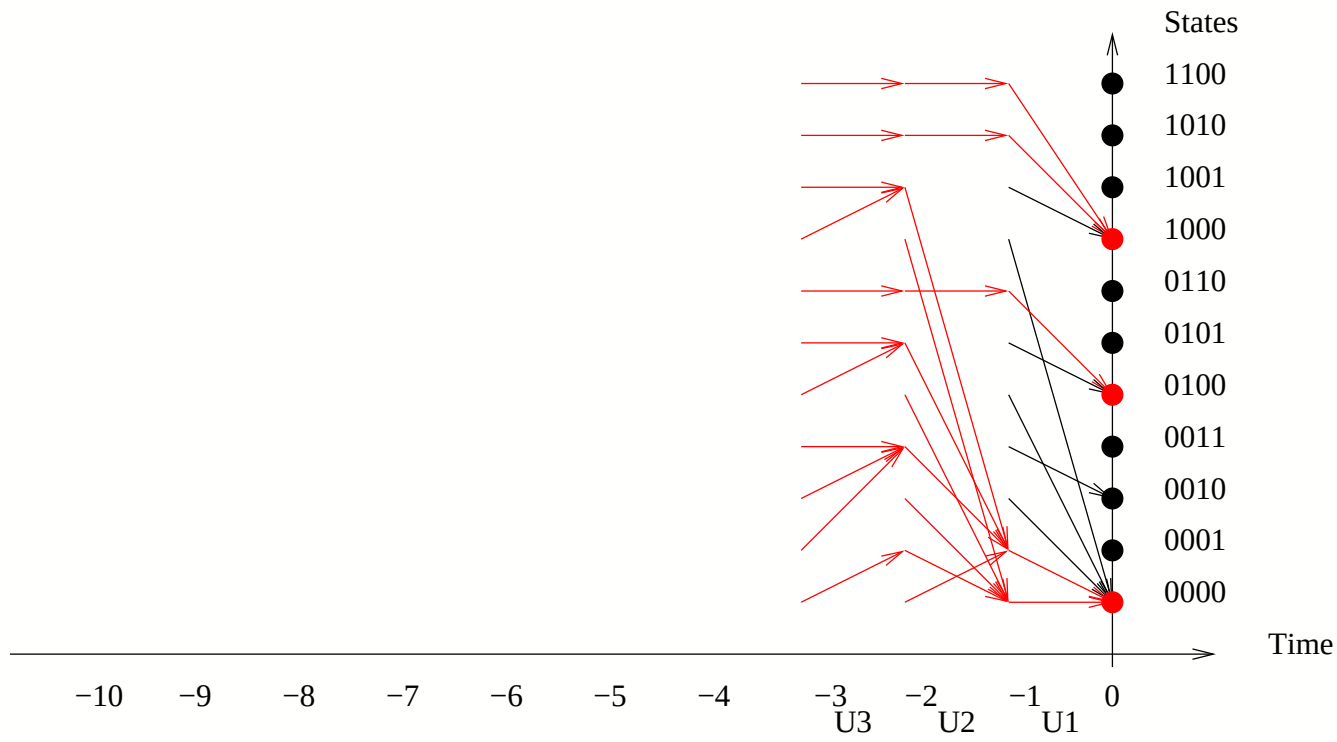


$$\mathcal{Z}_0 = \mathcal{X}$$

$$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$$

$$\mathcal{Z}_2 = \{0000, 0100, 1000\}$$

Backward simulation example

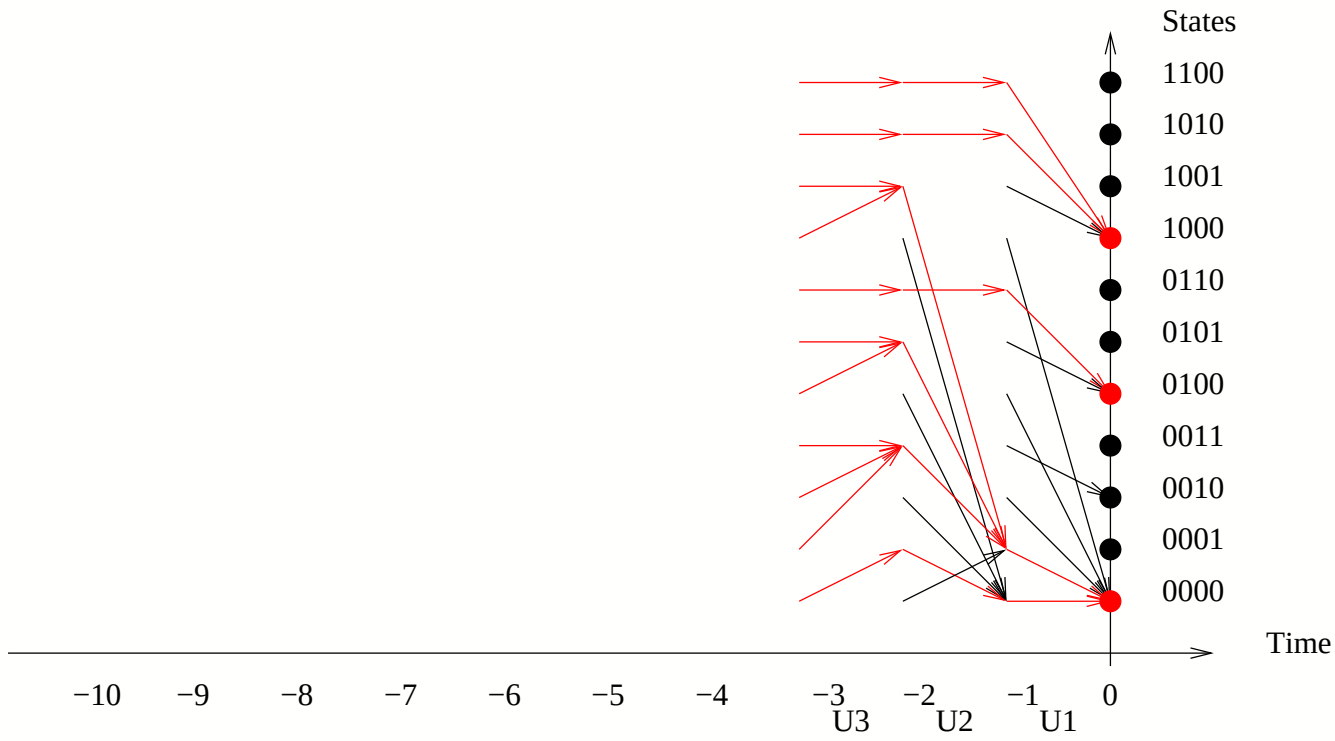


$$\mathcal{Z}_0 = \mathcal{X}$$

$$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$$

$$\mathcal{Z}_2 = \{0000, 0100, 1000\}$$

Backward simulation example



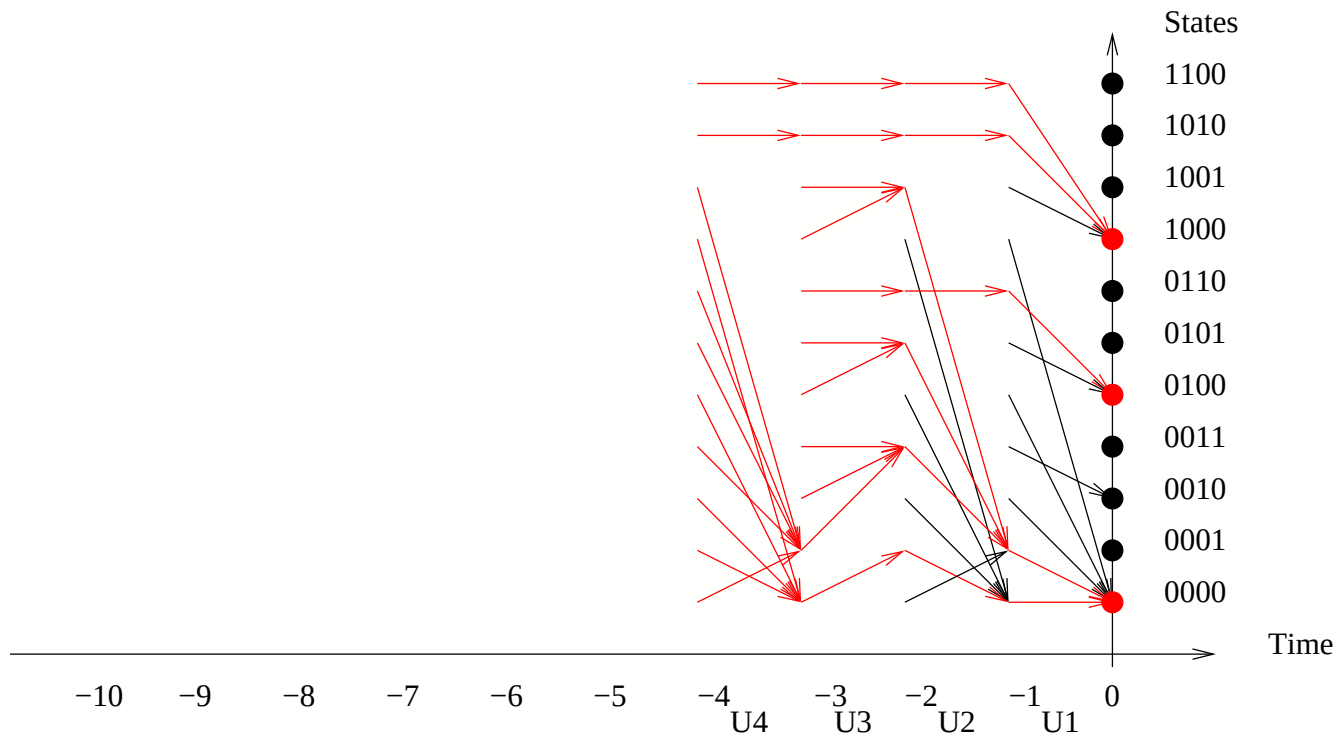
$\mathcal{Z}_0 = \mathcal{X}$

$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$

$\mathcal{Z}_2 = \{0000, 0100, 1000\}$

$\mathcal{Z}_3 = \{0000, 0100, 1000\}$

Backward simulation example



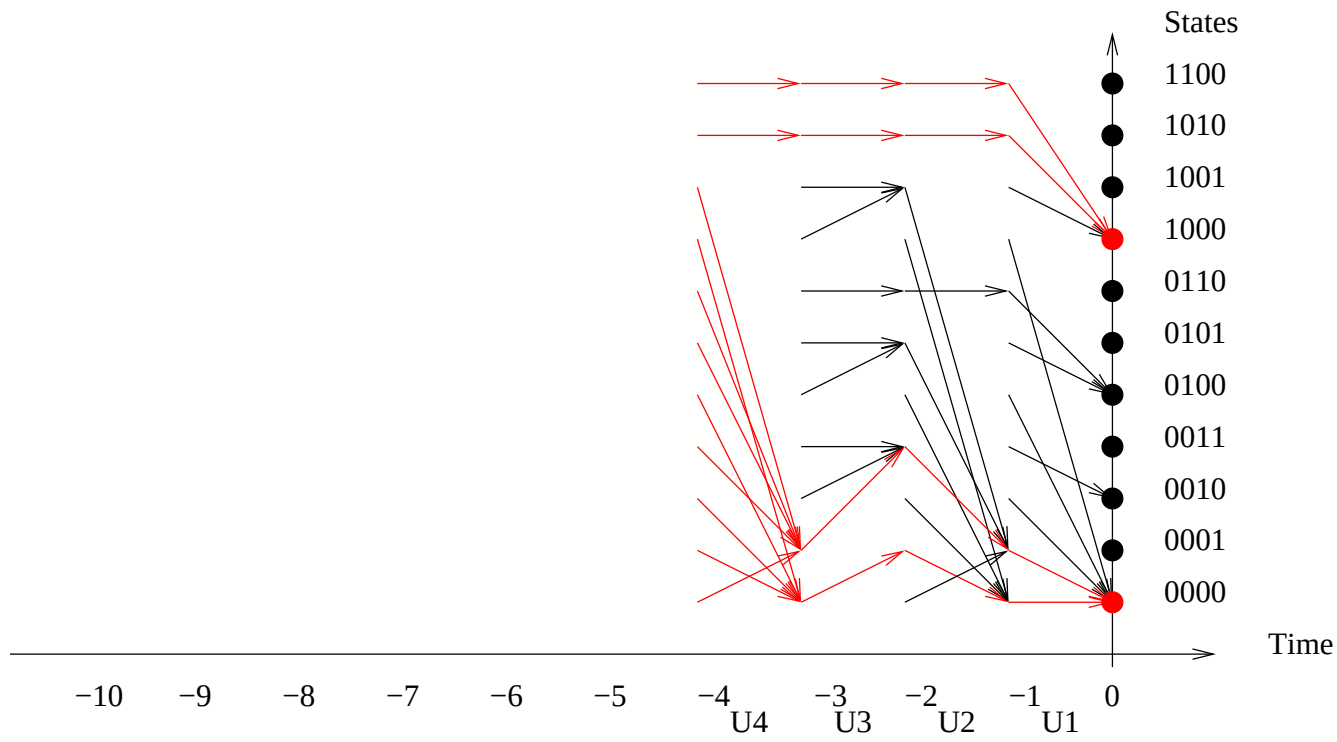
$\mathcal{Z}_0 = \mathcal{X}$

$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$

$\mathcal{Z}_2 = \{0000, 0100, 1000\}$

$\mathcal{Z}_3 = \{0000, 0100, 1000\}$

Backward simulation example



$\mathcal{Z}_0 = \mathcal{X}$

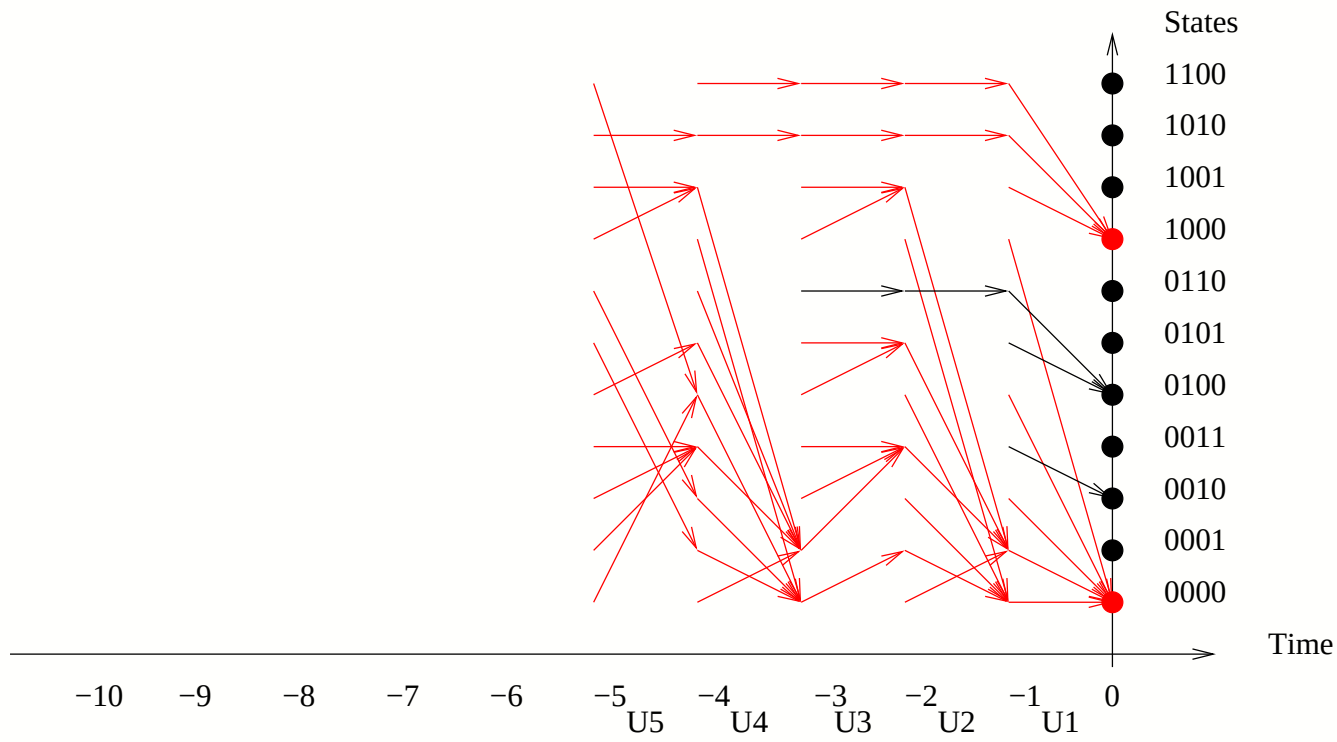
$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$

$\mathcal{Z}_2 = \{0000, 0100, 1000\}$

$\mathcal{Z}_3 = \{0000, 0100, 1000\}$

$\mathcal{Z}_4 = \{0000, 1000\}$

Backward simulation example



$\mathcal{Z}_0 = \mathcal{X}$

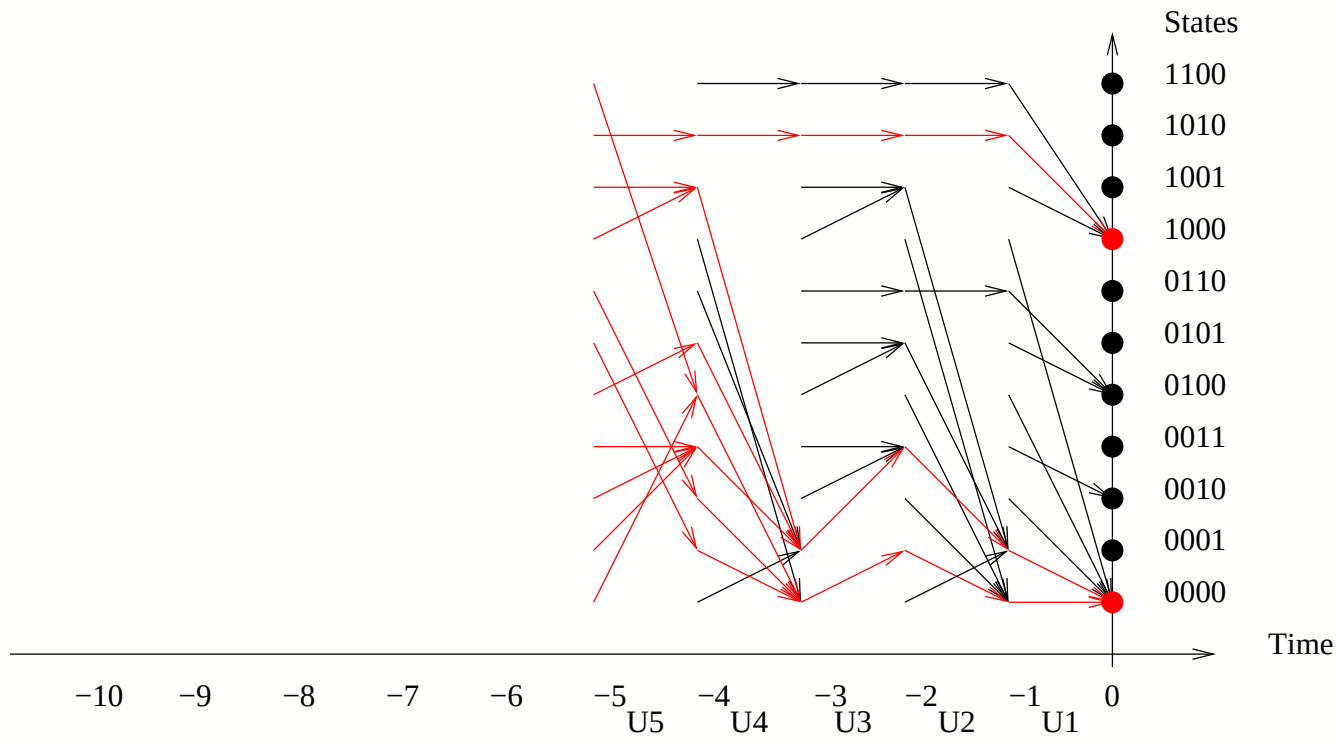
$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$

$\mathcal{Z}_2 = \{0000, 0100, 1000\}$

$\mathcal{Z}_3 = \{0000, 0100, 1000\}$

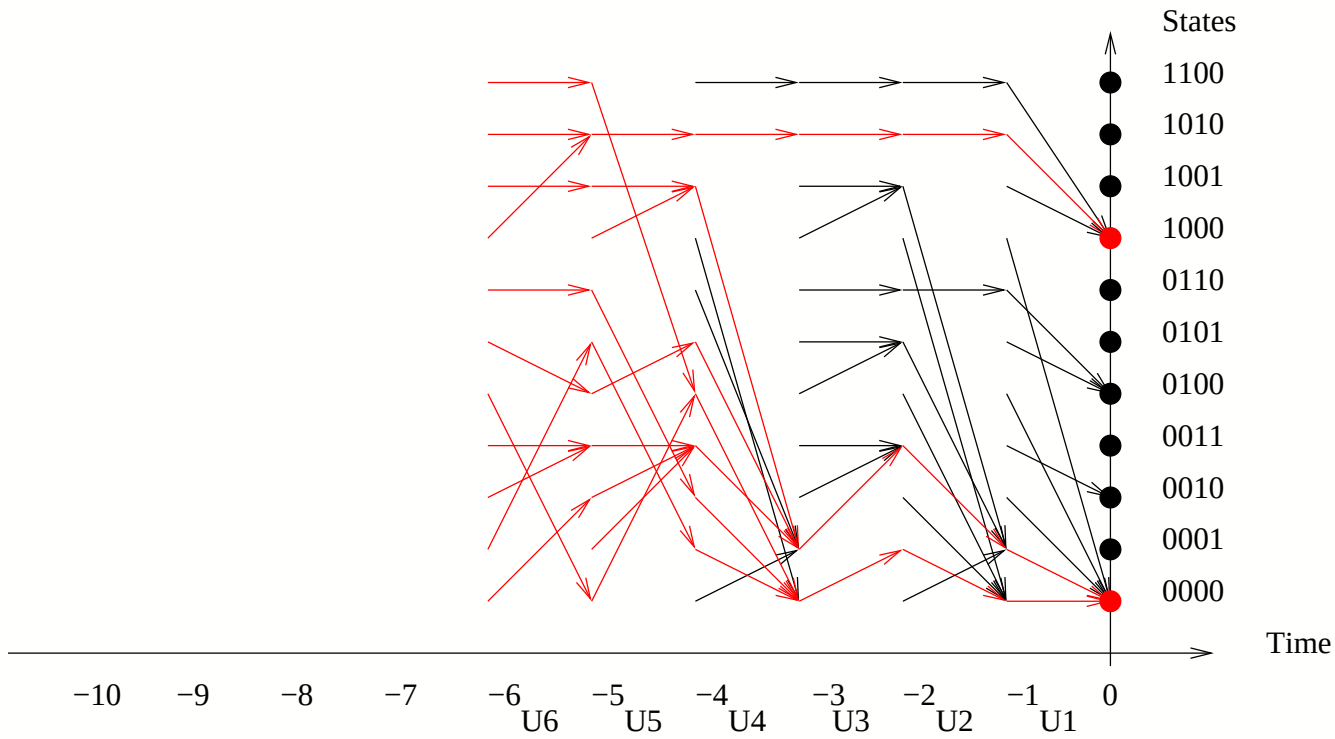
$\mathcal{Z}_4 = \{0000, 1000\}$

Backward simulation example



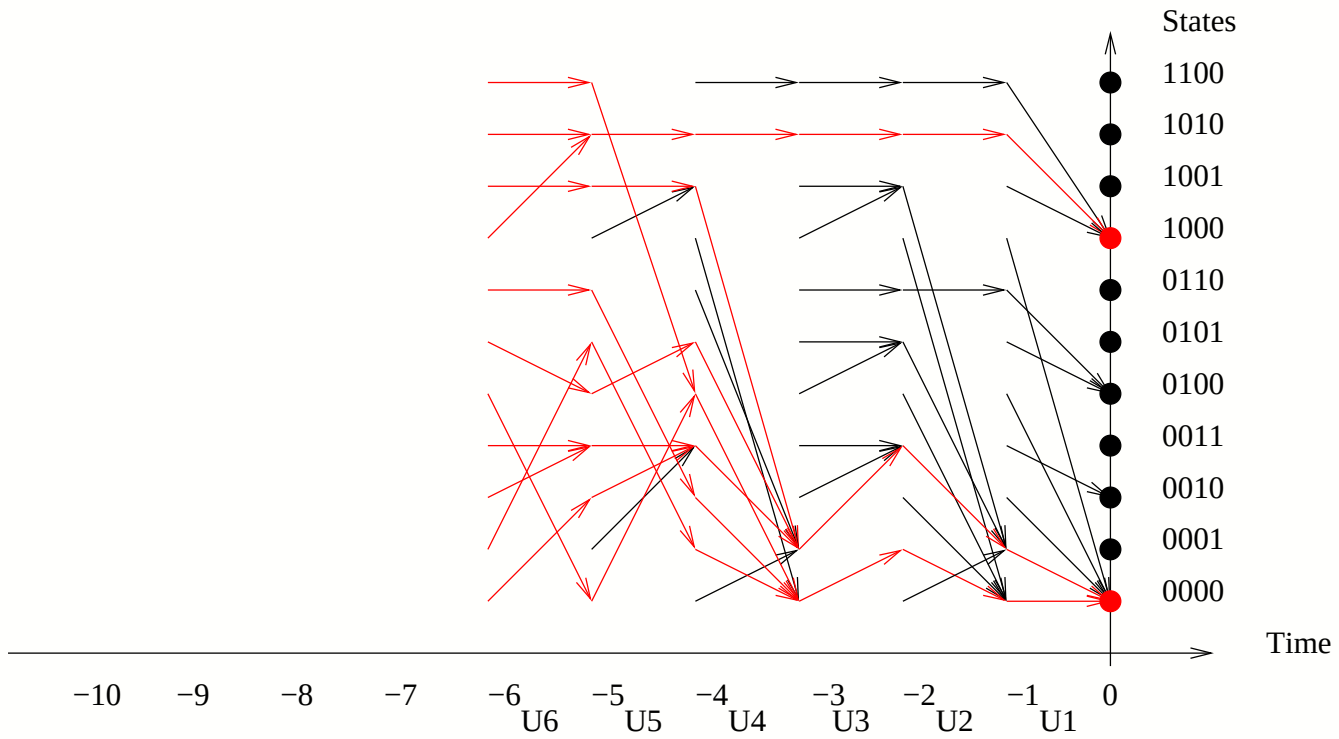
$$\mathcal{Z}_0 = \mathcal{X}$$
$$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$$
$$\mathcal{Z}_2 = \{0000, 0100, 1000\}$$
$$\mathcal{Z}_3 = \{0000, 0100, 1000\}$$
$$\mathcal{Z}_4 = \{0000, 1000\}$$
$$\mathcal{Z}_5 = \{0000, 1000\}$$

Backward simulation example



$$\mathcal{Z}_0 = \mathcal{X}$$
$$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$$
$$\mathcal{Z}_2 = \{0000, 0100, 1000\}$$
$$\mathcal{Z}_3 = \{0000, 0100, 1000\}$$
$$\mathcal{Z}_4 = \{0000, 1000\}$$
$$\mathcal{Z}_5 = \{0000, 1000\}$$

Backward simulation example



$\mathcal{Z}_0 = \mathcal{X}$

$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$

$\mathcal{Z}_2 = \{0000, 0100, 1000\}$

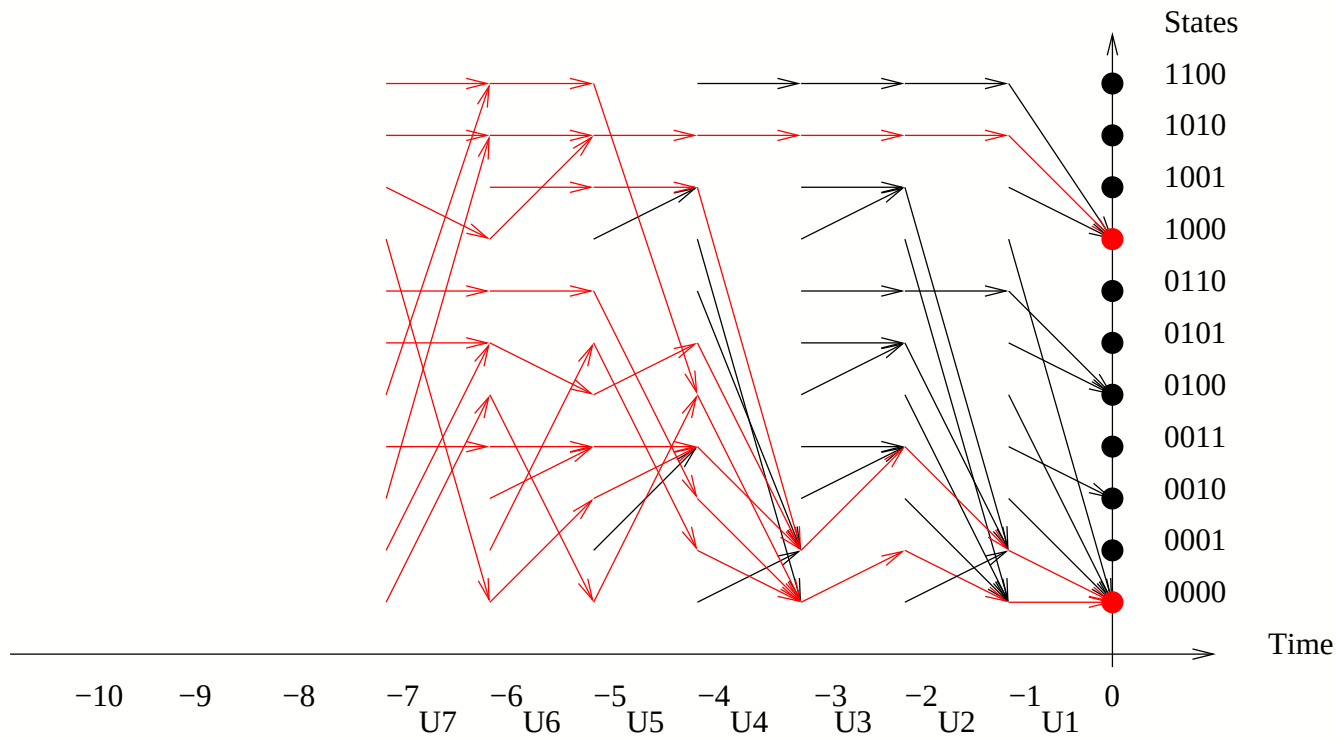
$\mathcal{Z}_3 = \{0000, 0100, 1000\}$

$\mathcal{Z}_4 = \{0000, 1000\}$

$\mathcal{Z}_5 = \{0000, 1000\}$

$\mathcal{Z}_6 = \{0000, 1000\}$

Backward simulation example



$\mathcal{Z}_0 = \mathcal{X}$

$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$

$\mathcal{Z}_2 = \{0000, 0100, 1000\}$

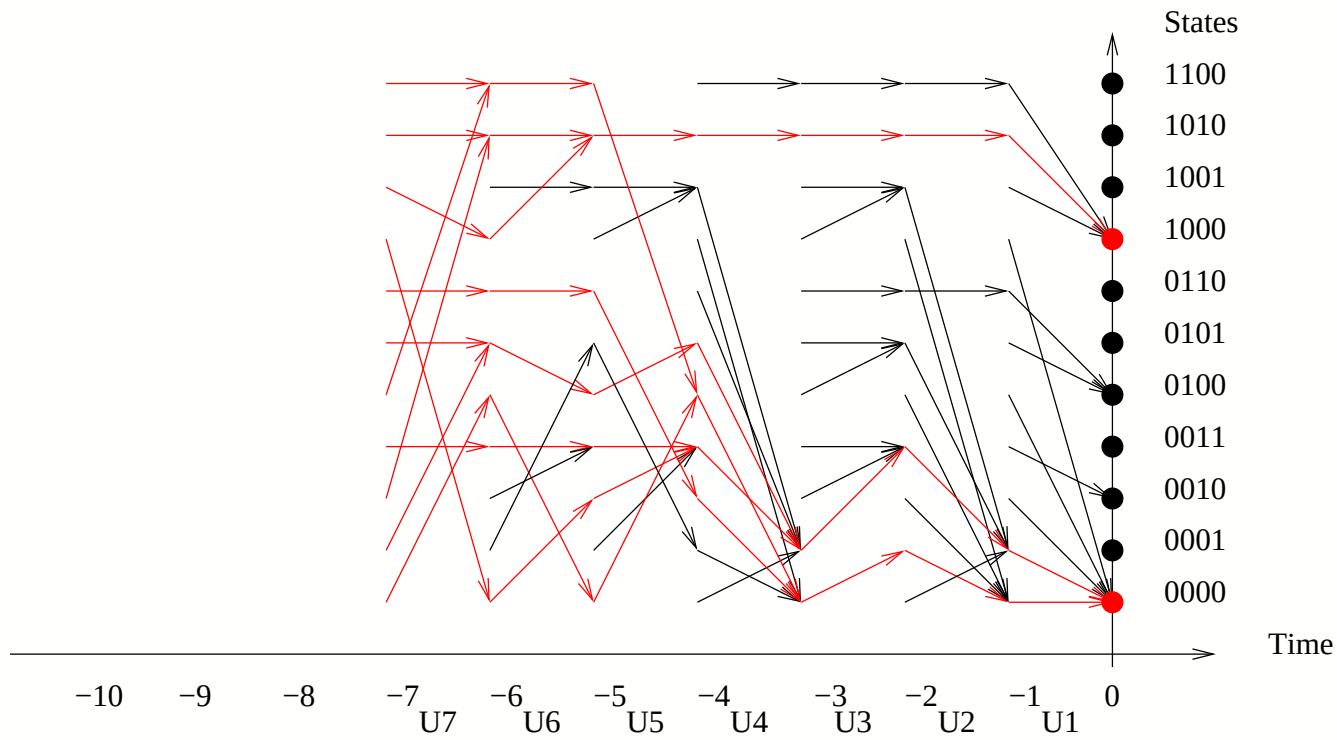
$\mathcal{Z}_3 = \{0000, 0100, 1000\}$

$\mathcal{Z}_4 = \{0000, 1000\}$

$\mathcal{Z}_5 = \{0000, 1000\}$

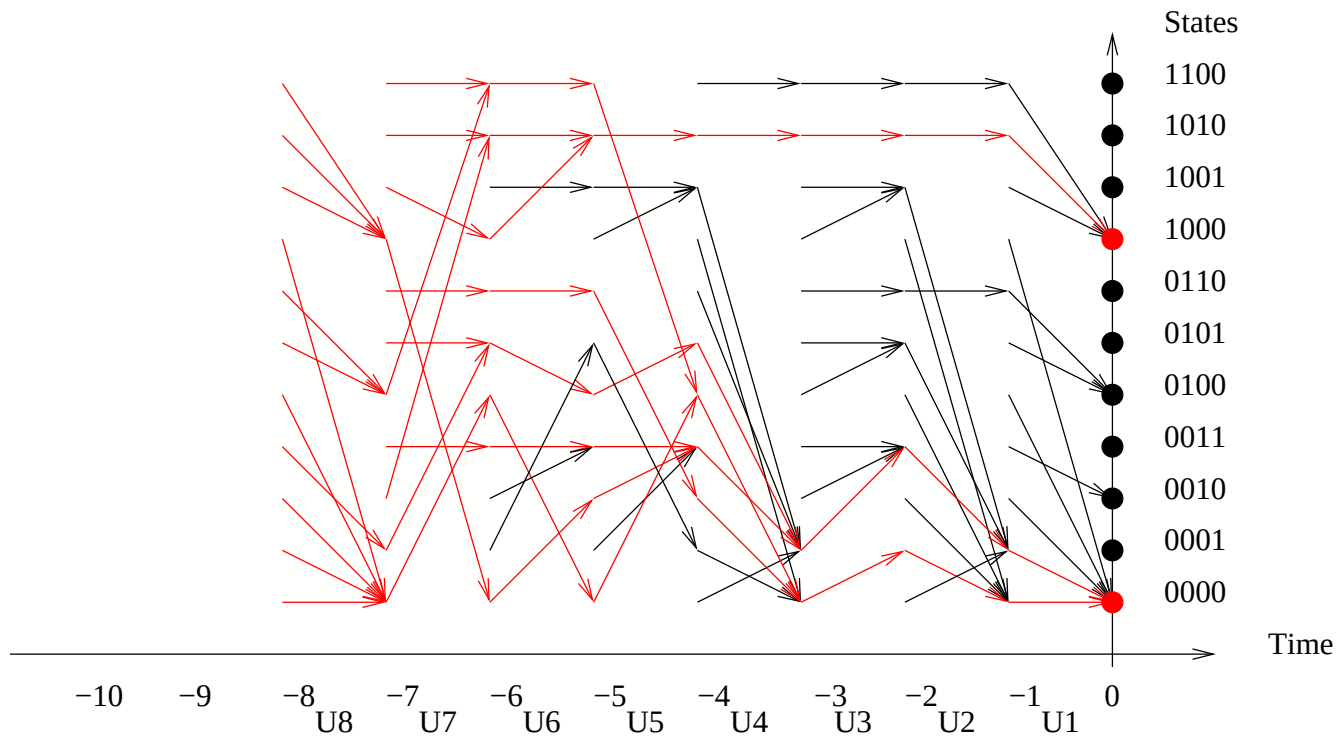
$\mathcal{Z}_6 = \{0000, 1000\}$

Backward simulation example



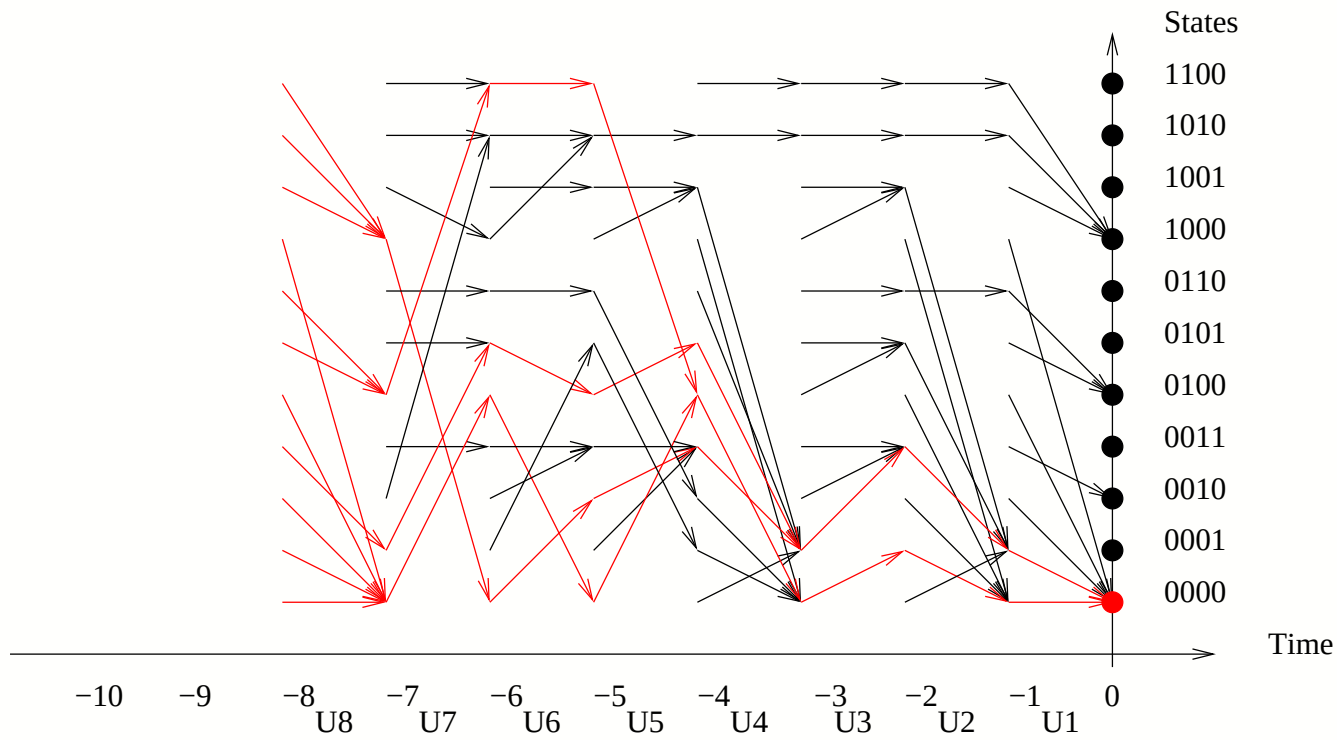
$$\mathcal{Z}_0 = \mathcal{X}$$
$$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$$
$$\mathcal{Z}_2 = \{0000, 0100, 1000\}$$
$$\mathcal{Z}_3 = \{0000, 0100, 1000\}$$
$$\mathcal{Z}_4 = \{0000, 1000\}$$
$$\mathcal{Z}_5 = \{0000, 1000\}$$
$$\mathcal{Z}_6 = \{0000, 1000\}$$
$$\mathcal{Z}_7 = \{0000, 1000\}$$

Backward simulation example



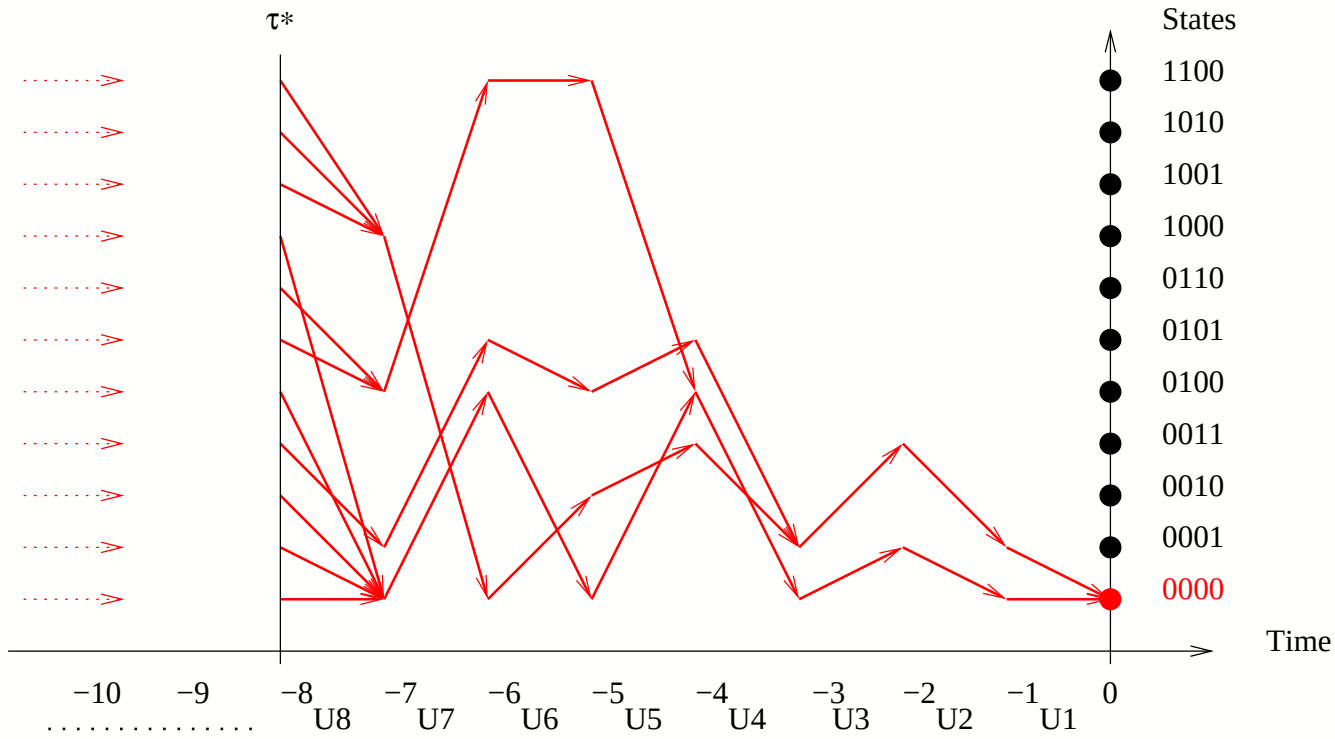
$$\mathcal{Z}_0 = \mathcal{X}$$
$$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$$
$$\mathcal{Z}_2 = \{0000, 0100, 1000\}$$
$$\mathcal{Z}_3 = \{0000, 0100, 1000\}$$
$$\mathcal{Z}_4 = \{0000, 1000\}$$
$$\mathcal{Z}_5 = \{0000, 1000\}$$
$$\mathcal{Z}_6 = \{0000, 1000\}$$
$$\mathcal{Z}_7 = \{0000, 1000\}$$

Backward simulation example



$$\mathcal{Z}_0 = \mathcal{X}$$
$$\mathcal{Z}_1 = \{0000, 0010, 0100, 1000\}$$
$$\mathcal{Z}_2 = \{0000, 0100, 1000\}$$
$$\mathcal{Z}_3 = \{0000, 0100, 1000\}$$
$$\mathcal{Z}_4 = \{0000, 1000\}$$
$$\mathcal{Z}_5 = \{0000, 1000\}$$
$$\mathcal{Z}_6 = \{0000, 1000\}$$
$$\mathcal{Z}_7 = \{0000, 1000\}$$
$$\mathcal{Z}_8 = \{0000\}$$

Backward simulation example



Process stops when $|\mathcal{Z}_n| = 1$

Stopping time $\tau^* = 8$

Number of computation of Φ (complexity) : $n.\tau^*$

Backward coupling simulation

Proposition 1 (Propp & Wilson) *If $\tau^* < +\infty$ a.s. then the returned value is stationary distributed.*

Proof :

$\{\mathcal{Z}_n\}_{n\in\mathbb{N}}$ is non-increasing and constant for n sufficiently large

$\Phi(\Phi(\cdots(\Phi(\mathcal{X}, U_{-n+1}), \cdots), U_{-1}), U_0) \overset{\mathcal{L}}{\sim} \Phi(\Phi(\cdots(\Phi(\mathcal{X}, U_1), \cdots), U_{n-1}), U_n)$

Remarks :

Stopping times τ and τ^* have the same law,

τ depends on Φ coding (not only on the transition matrix !)

$\Rightarrow$ optimization problem

Functional backward coupling

Idea:

- Directly compute the cost function C
- Same backward scheme
- wait for coupling
- stopping time τ_C^*

$$\tau_C^* \leq \tau^* \ \ a.s.$$

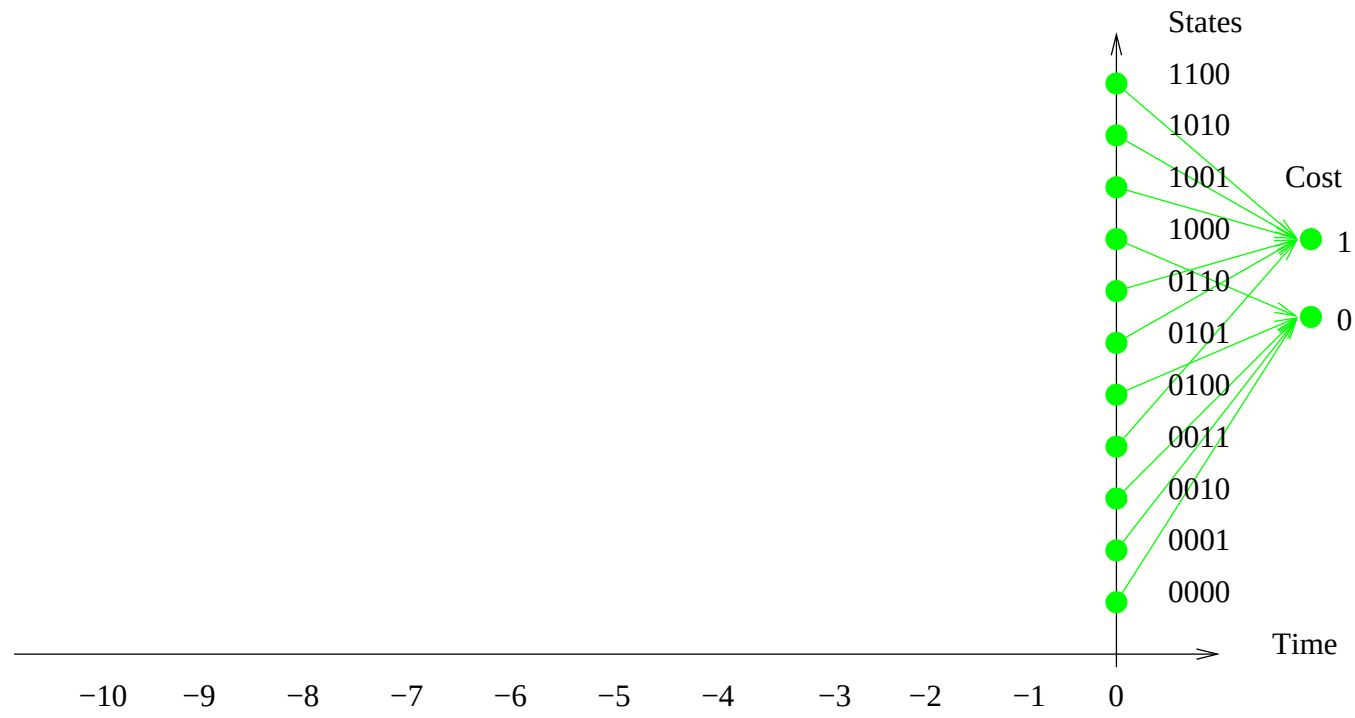
$\Rightarrow$ speedup

$$\mathcal{Z}_n^C = C(\Phi(\Phi(\cdots(\Phi(\mathcal{X}, U_{-n+1}), \cdots), U_{-1}), U_0)).$$

potential set of reachable costs at step n

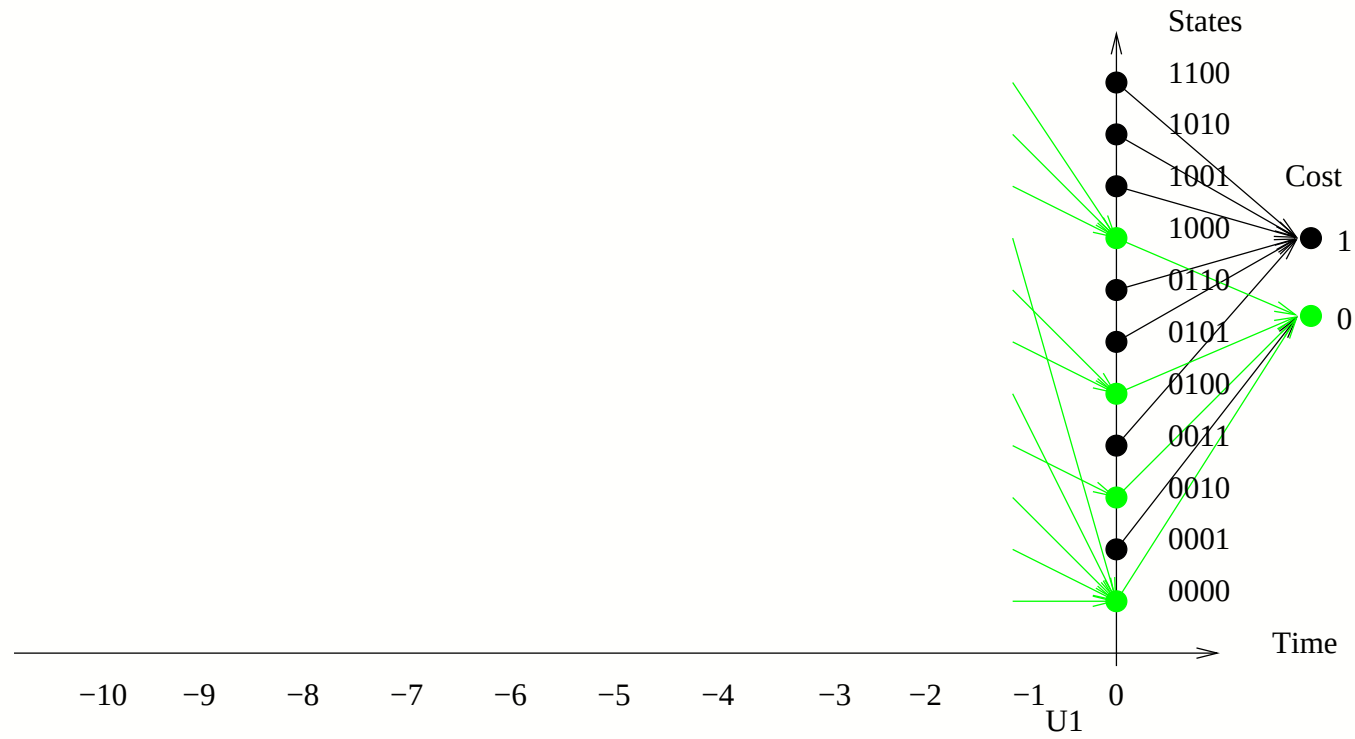
for all $x \in \mathcal{X}$ **do**
  $y(x) \leftarrow C(x)$
end for
repeat
  u $\leftarrow$ Random;
  for all $x \in \mathcal{X}$ **do**
    $y(x) \leftarrow y(\Phi(x, u))$;
  end for
until All $y(x)$ are equal
return $y(x)$

Backward functional simulation



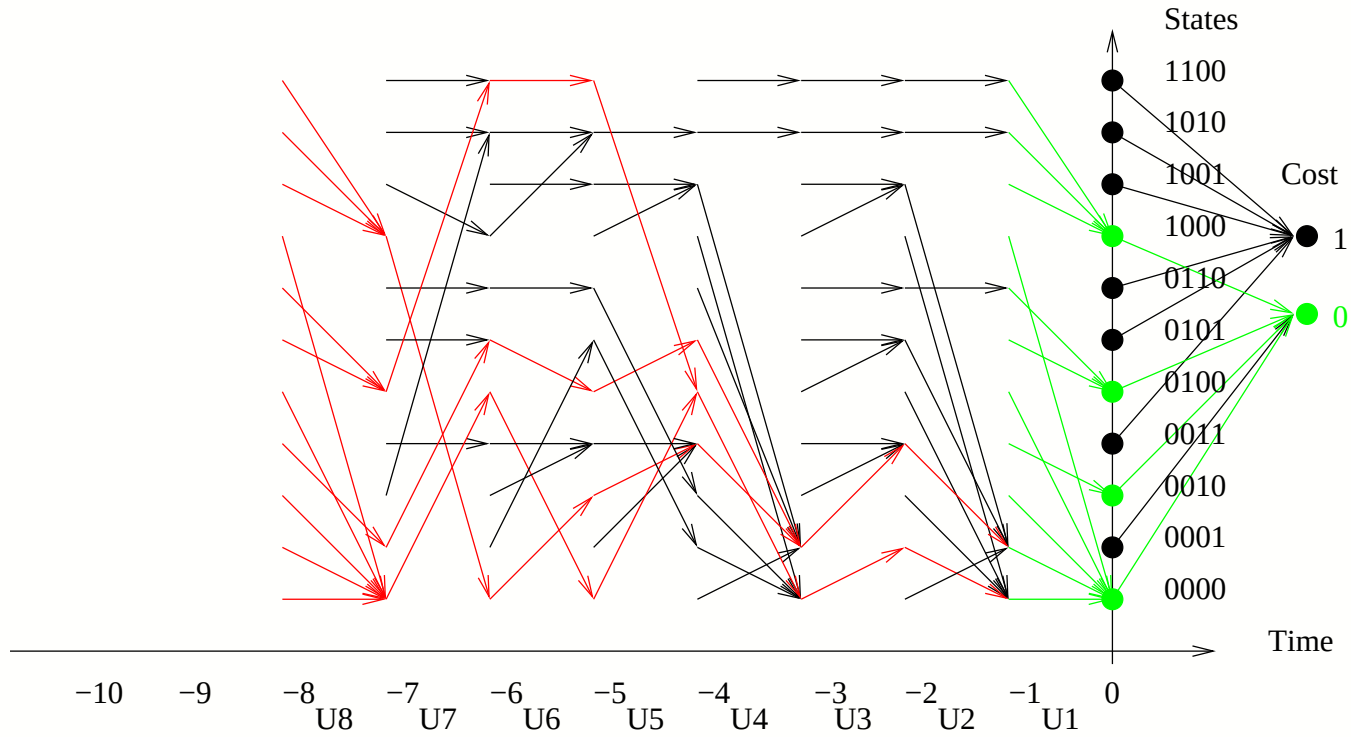
$$\mathcal{Z}_0^C = \{0, 1\}$$

Backward functional simulation



$$\mathcal{Z}_0^C = \{0, 1\}$$
$$\mathcal{Z}_1^C = \{0\}$$

Backward functional simulation



$$\mathcal{Z}_0^C = \{0,1\}$$
$$\mathcal{Z}_1^C = \{0\}$$

$$\tau^* = 8 \quad \tau_C^* = 1$$

Is the coupling time reduction significant ?

Implementation considerations

Φ coding : choice of a representation

- Sparse description of Markov generators (m non negative elements)
- Discretization (Uniformization)
- Aliasing table
- Coupling condition

Examples

- Functionality tests : "random transition matrices"
- Resource sharing model : statistical verification
- Overflow model : sparsity and gain

Example1

Random transition coefficients:

Number of states	10	100	500	1000	3000
Mean coupling time	3.1	4.5	5.3	5.7	6.1
Mean execution time μs	3	17	170	360	1100

Pentium III 700MHz and 256Mb memory. Sample size 10000.

Remarks:

- very small coupling time

- Coefficients : same order of magnitude, aliasing enforces coupling

Comparison with birth and death process :

Number of states	10	100	500	1000	3000
Mean coupling time	41	557	2850	5680	17000
Mean execution time μs	28	1800	88177	366000	3.5s

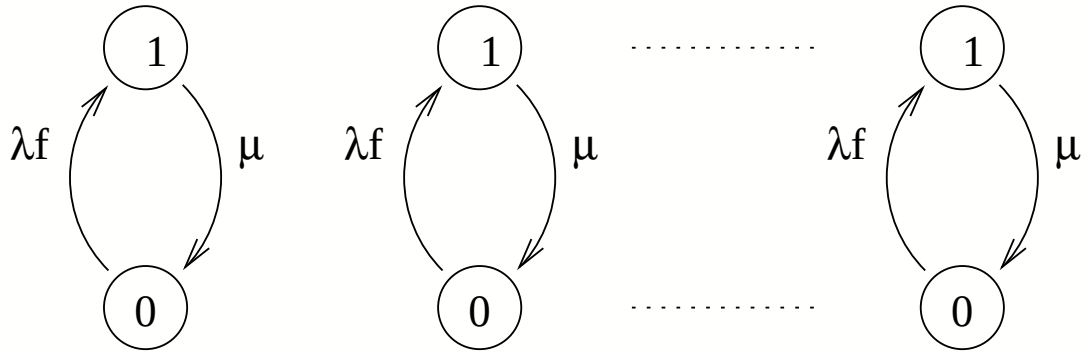
Remarks:

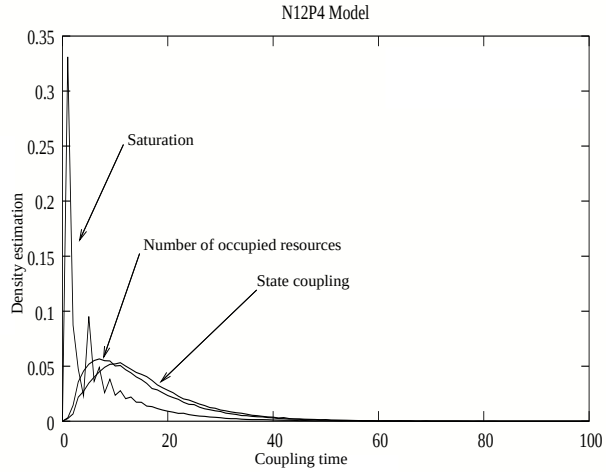
- large coupling time

- sparse matrix, large graph diameter

Resource sharing model

P resources N users; state =$(x_1, \cdots, x_N)$; access constraint $f = (\sum x_i < P)$
product form solution $\Rightarrow$ statistical validation





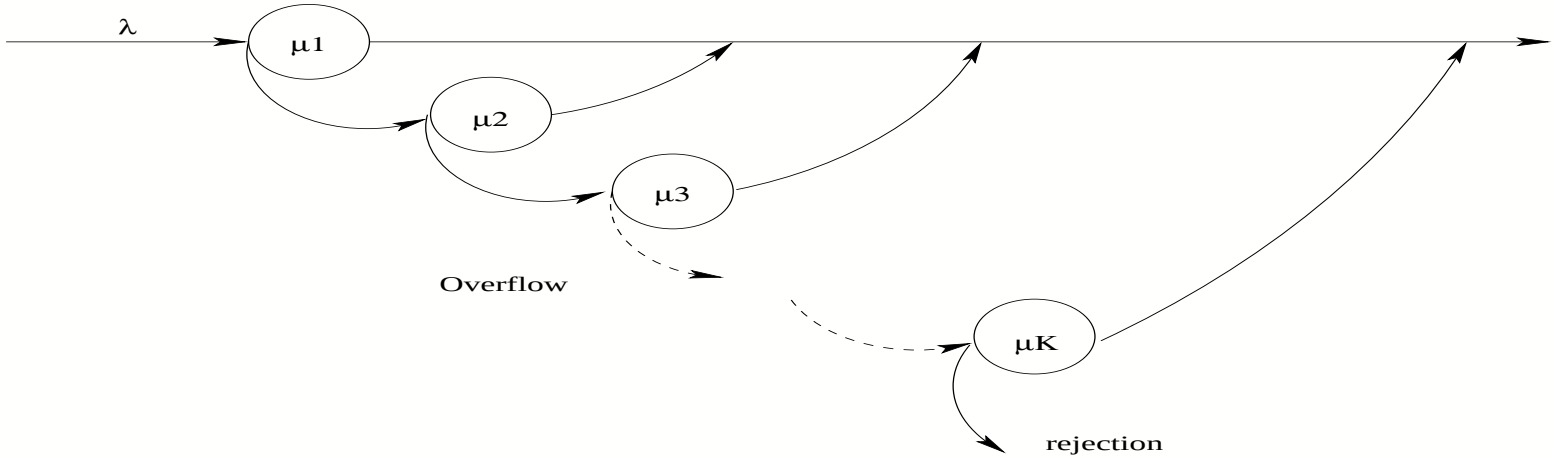
- state coupling
- functional coupling :

number of occupied resources : $\sum_i x_i$,
saturation : $(\sum_i x_i = P)$,

High reduction of the functional coupling time

Erlang model (1905)

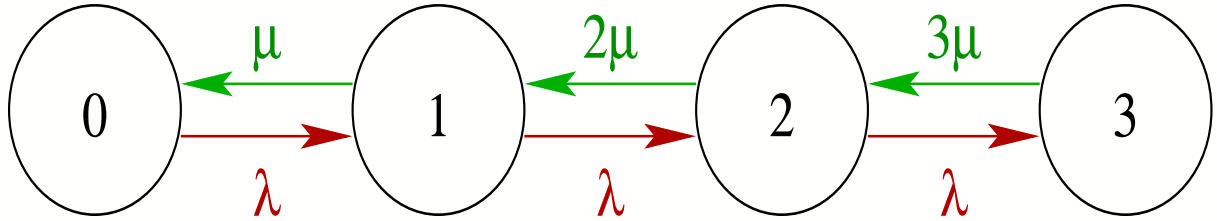
Typical overflow model K servers



State vector = [1,1,0,0,1,0,...]
State space size 2^K
λ input rate
μ_i service rates

Erlang model (1905)

$\mu_i = \mu$ aggregation,



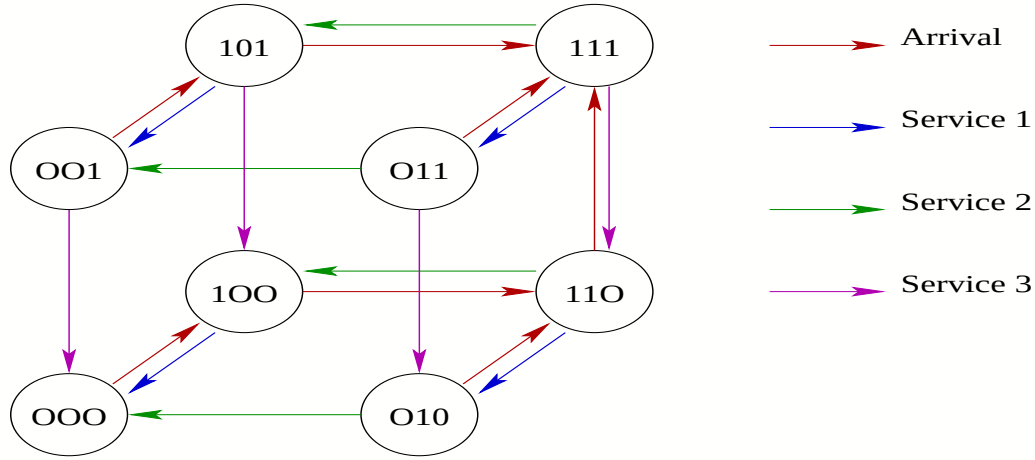
reversibility, product form solution

$$\pi_n = G(K)\frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n$$

$\Rightarrow$ **Erlang formula**

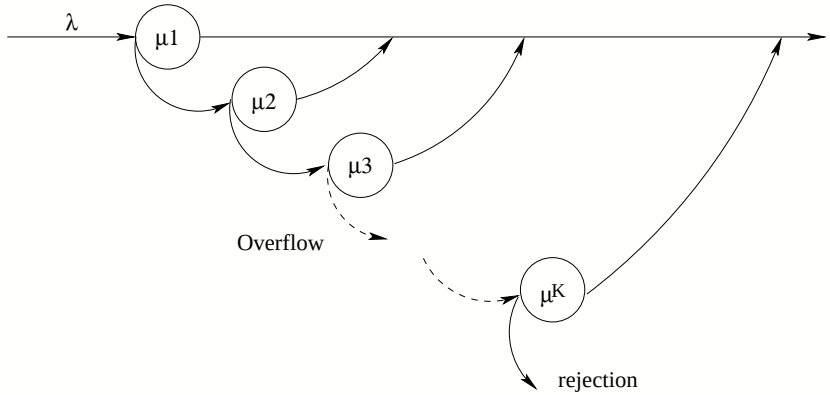
Erlang model (1905)

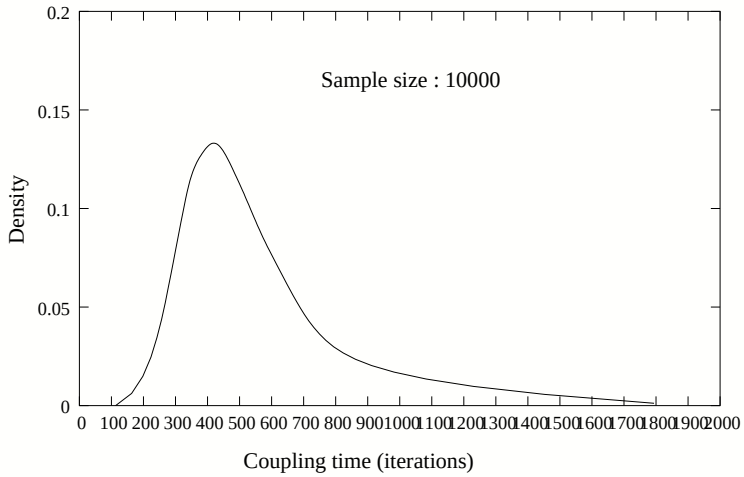
μ_i distincts ???



non monotonic, non reversible !!!

Overflow model



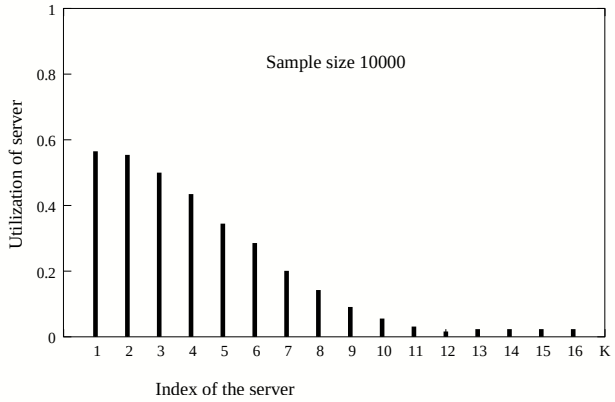


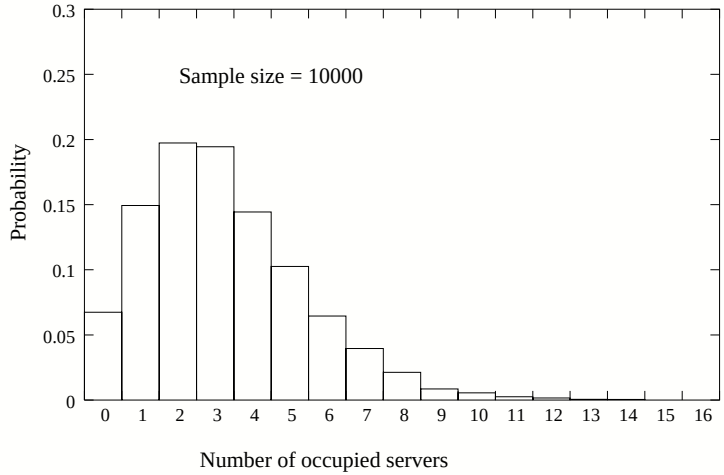
K servers,
priority on overflows
input rate λ,
different service rate
state $(x_1, \cdots, x_K)$, $x_i \in \{0, 1\}$,
size ~ 130000
low diameter
non product-form structure,

Parameter	Value
minimum	113
maximum	1794
median	465
mean	498
Std	180

exponential tail, low mean value

Overflow model (2)



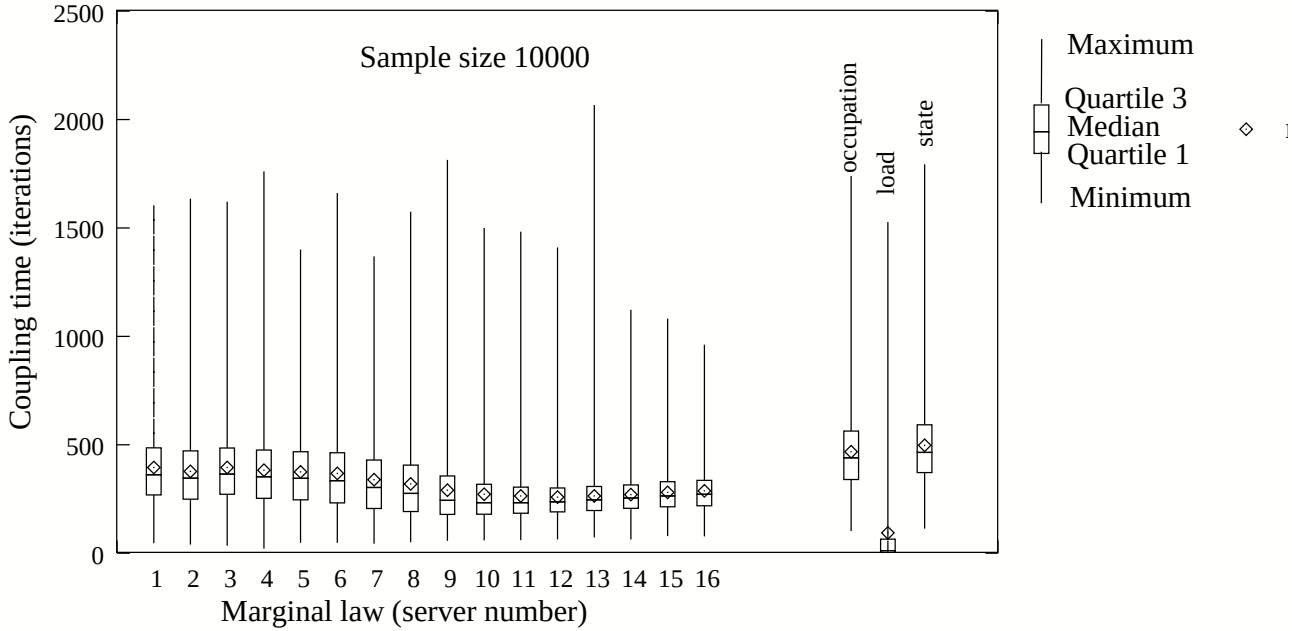


Marginal probabilities estimation

$$\mathbb{P}(X_i = 1)$$

Marginal distribution of the occupied servers

Overflow model (3)

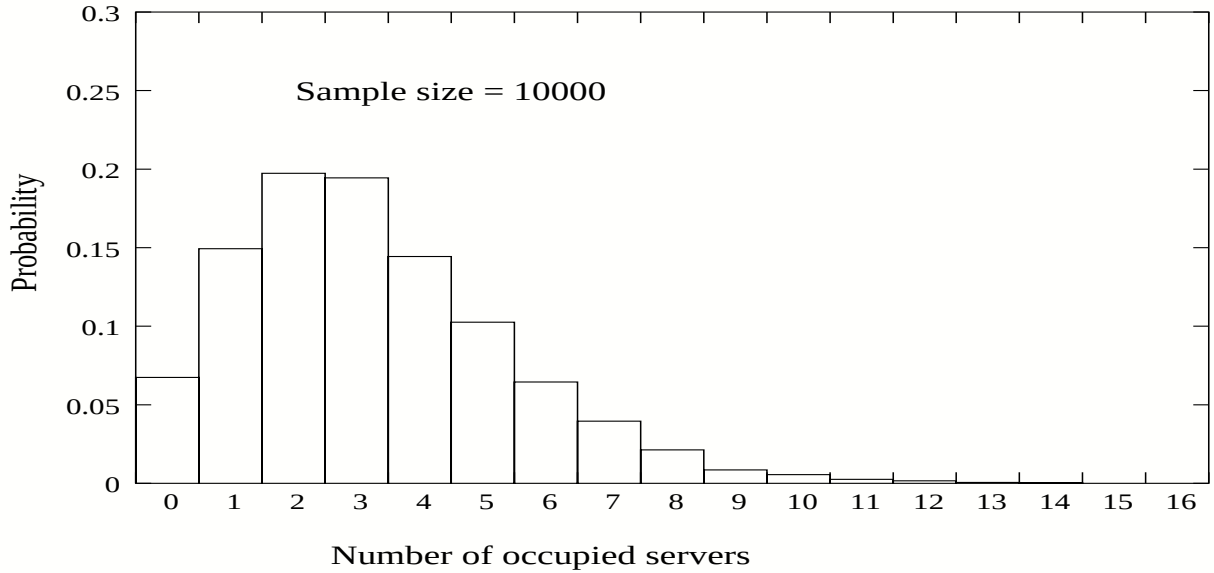


Functional coupling time

- gain 20% for the first marginals
- utilization : best reduction

Performances : service number

Marginal law of occupied servers

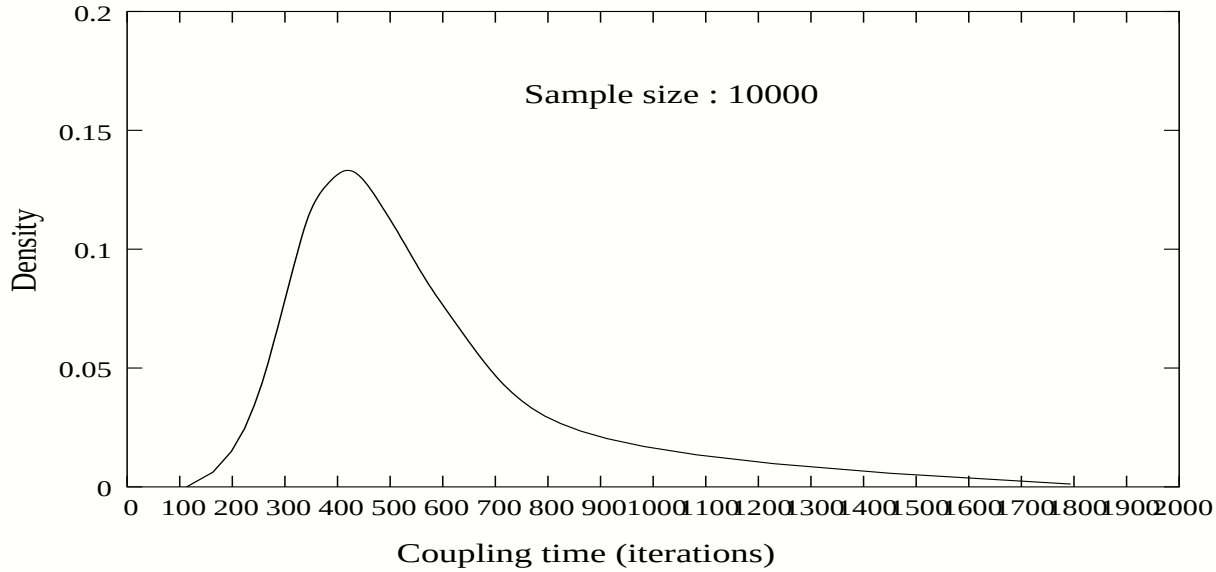


Estimation of the load of the system;

$$\mathbb{P}(N > 13) \leq 10^{-4}$$

Global coupling

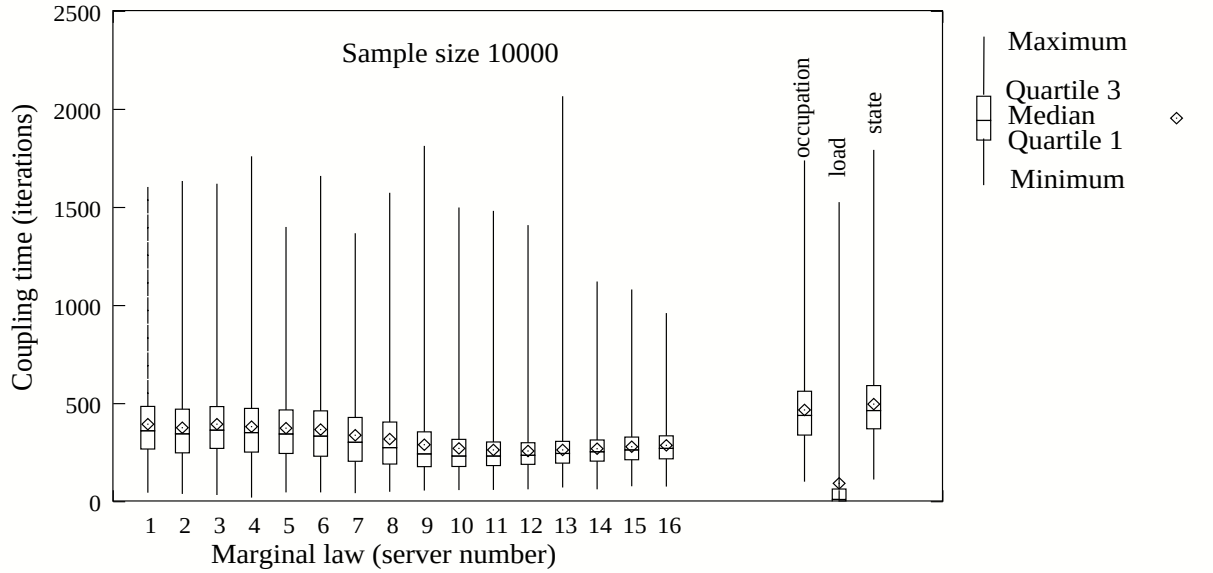
Each iteration operates on 131.072 dimensional vectors



exponential decay

Coupling time : synthesis

Functional coupling :



PSI

Software tool: **PSI : Perfect Simulator**

http://www-id.imag.fr/Software/PSI/

Two versions : unix command / with a simple interface

`psi_alias -i example.marca -o example`

generates alias tables (`example.simu`)

`psi_sample -i example.simu -d sample-size -c example.cost -o example`

`example.cost` associates to each state its cost

generates samples of costs stationary distributed (`example.sample`)

Future works

Open problems

- relation between ϕ and τ
- parallel version of the algorithm : block decomposition
- impact of the cost function and aggregation properties

Technical problems

- interfaces with software modeling tools
- description of the state space (vectors)
- cache optimizations

Stochastic comparison

A Markov chain is said to be monotonic iff it preserves stochastic ordering on distributions.
Question :
could we find a transition coding function ϕ such that

$$x \leq y \ \Rightarrow \forall u \ \ \phi(x,u) \leq_X \phi(y,u)$$

$\rightarrow$ few trajectories are necessary

From continuous to discrete time

Continuous time Markov chain
Generator Q , $\Lambda \geq \max_i q_{i,i}$ then

$$P = Id + \frac{1}{\Lambda} Q$$

is the transition matrix of the markov chain of the embedded process driven by a Poisson process with rate Λ

Transition or events

Example Erlang 3 servers : state $[x_1, x_2, x_3]$,
4 events

departure server 1 :

$$[x_1, x_2, x_3] \longrightarrow [(x_1 - 1)^+, x_2, x_3]$$

departure server 2 :

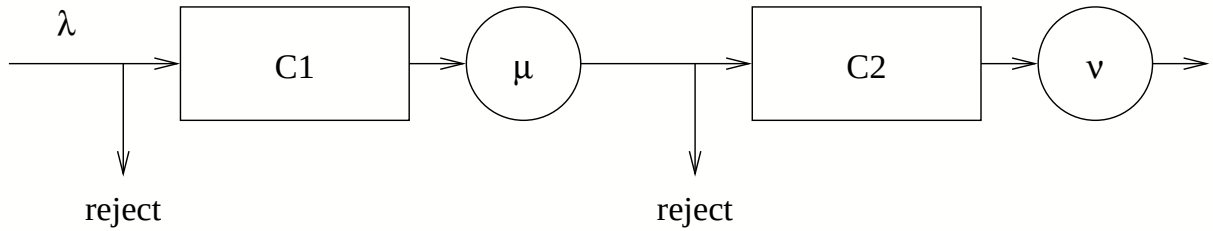
$$[x_1, x_2, x_3] \longrightarrow [x_1, (x_2 - 1)^+, x_3]$$

arrival : place 1 in the first non null place

Uniformization by $\Lambda = \lambda + \mu_1 + \mu_2 + \mu_3$

$\Rightarrow$ Monotonous process !!!!!

Queue or reject



departure server 1 :

$$[x_1, x_2] \longrightarrow [(x_1 - 1)^+, min(x_2 + 1, C2)]$$

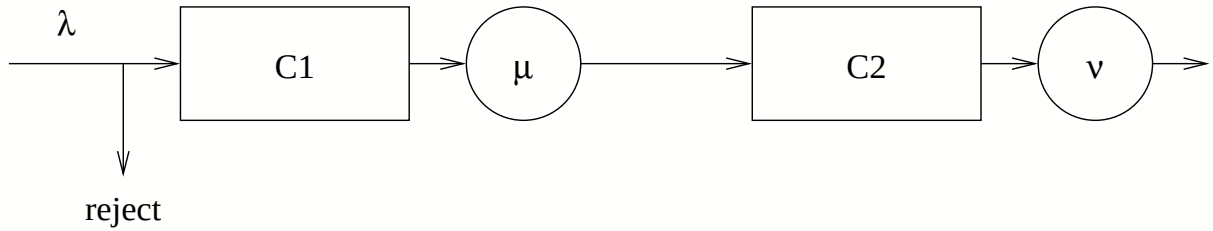
departure server 2 : $[x_1, x_2] \longrightarrow [x_1, (x_2 - 1)^+]$

arrival : $[x_1, x_2] \longrightarrow [min(x_1 + 1, C1), x_2]$

Uniformization by $\Lambda = \lambda + \mu + \nu$

$\Rightarrow$ Monotonous process !!!!!

Queue or block



departure server 1 :

$$[x_1, x_2] \longrightarrow [x_1 - 1, x_2 + 1] \text{ if } x_1 > 0 \text{ and } x_2 < C2$$

departure server 2 : $[x_1, x_2] \longrightarrow [x_1, (x_2 - 1)^+]$

arrival : $[x_1, x_2] \longrightarrow [min(x_1 + 1, C1), x_2]$

Uniformization by $\Lambda = \lambda + \mu + \nu$

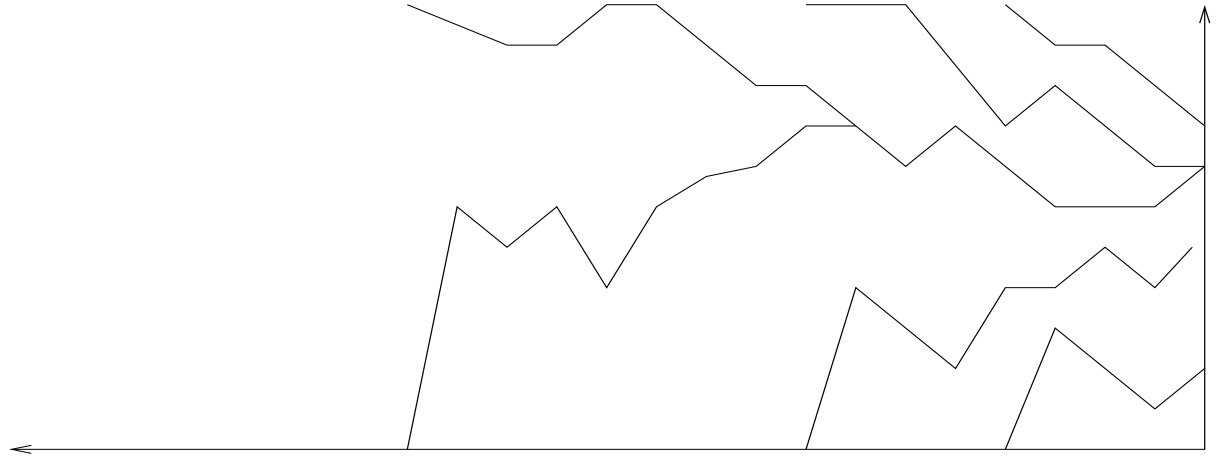
$\Rightarrow$ Monotonous process !!!!!

Queueing networks

Event approach : operates on vectors
Monotonicity with rejection, blocking, priorities
Exercises : find typical non monotonous networks
Generalize to finite number of customers networks (show it)

Implementation considerations

The global scheme is



needs the storage of events
Propp & Wilson : double period of simulation
Fill : interruptible forward algorithm

Future works

- Theoretical improvements:
  - deeper understanding of Φ properties and the spectrum of the transition matrix
  - evaluation or bounds on the coupling time
- Algorithmic perspectives:
  - building of alias table,
  - transform of alias table,
  - parallelization
- Model based approach :
  - structuration of the matrix : adapted strategies (QN, SAN, GSPN, PA,...)
  - model properties : monotonicity, reversibility,...
- Experimental results
  - find limit models : (ex Birth and death)
  - significant results (depending on the diameter and the coding of the chain)
  - huge models (size 2^{22})