Sep. 25, 2006

as a technique for random sampling

Random sampling via Markov chain

マルコフ連鎖を用いたランダムサンプリング法

*Shuji Kijima (来嶋 秀治)[1], Tomomi Matsui (松井 知己)[2]

[1]The University of Tokyo

(東京大学 大学院情報理工学系研究科 数理情報学専攻)

[2]Chuo University

(中央大学 理工学部 情報工学科)

In this talk,
`sampling' = `random sampling'

-7. Sampling via Markov chain

key word

Markov chain Monte Carlo (MCMC)

<narrowly defined as>

Monte Carlo method with sampling via Markov chain

<comprehensively intend>

sampling via Markov chain

Markov chain

- Markov chain M (ergodic) defined by
  - state space: $\Omega=\{s_1,s_2,s_3\}$ (finite)
  - transition: transition probability matrix P

$$P_{ij} = \Pr(\, i \to j \,) =: P(\, i, j)$$

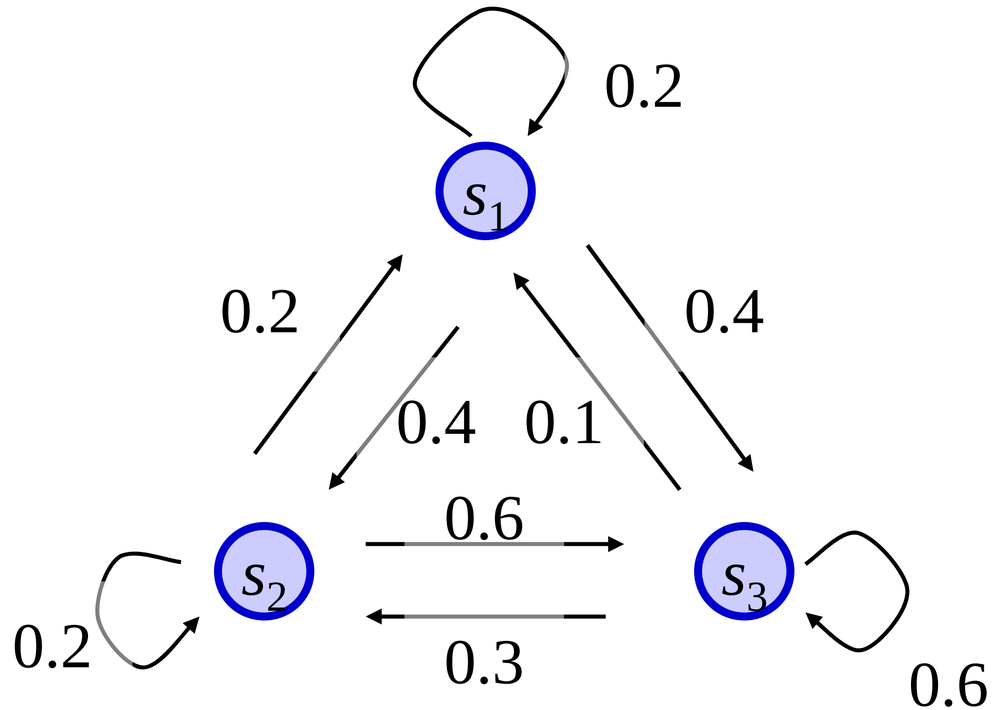


figure shows an ex. of
prob. trans. diagram of Markov chain

Markov chain

- Markov chain M (ergodic)
  - state space: $\Omega=\{s_1,s_2,s_3\}$ (finite)
  - transition: transition probability matrix P

$$P_{ij} = \Pr(\, i \to j \,) =: P(\, i, j)$$

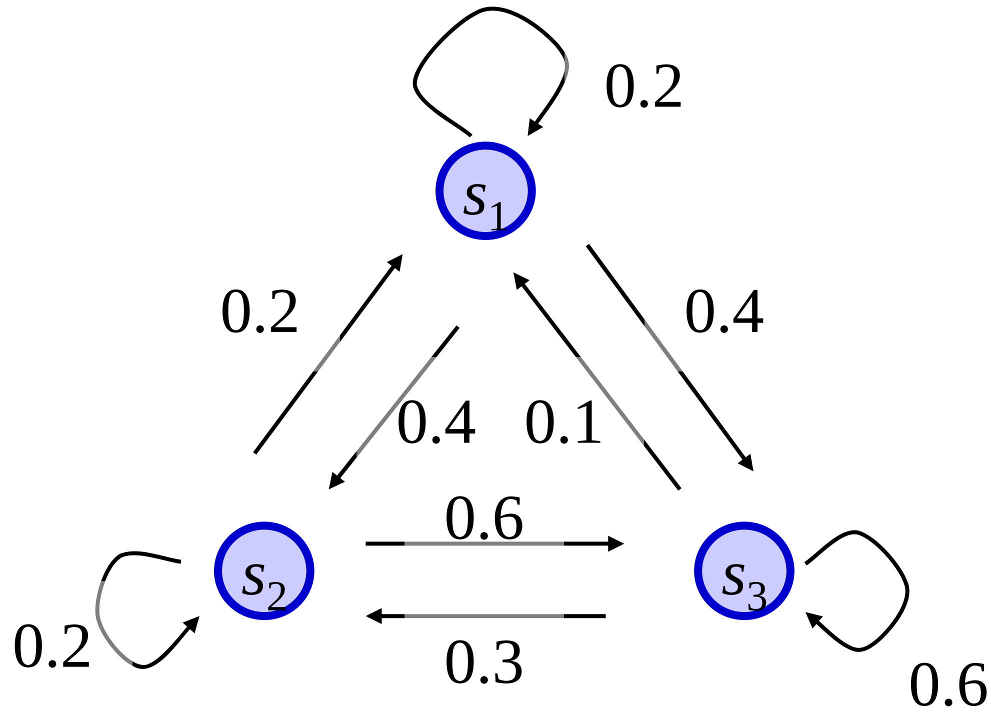
next state

current state

$$P = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{ccc} s_1 & s_2 & s_3 \\ \end{array}$$

$$P = \begin{pmatrix} 0.2 & 0.4 & 0.4 \\ 0.2 & 0.2 & 0.6 \\ 0.1 & 0.3 & 0.6 \end{pmatrix}$$

	s_1	s_2	s_3
s_1	0.2	0.4	0.4
s_2	0.2	0.2	0.6
s_3	0.1	0.3	0.6



underlying graph

Stationary distribution

- Markov chian M (ergodic)
  - state space: $\Omega=\{s_1,s_2,s_3\}$ (finite)
  - transition: transition probability matrix P

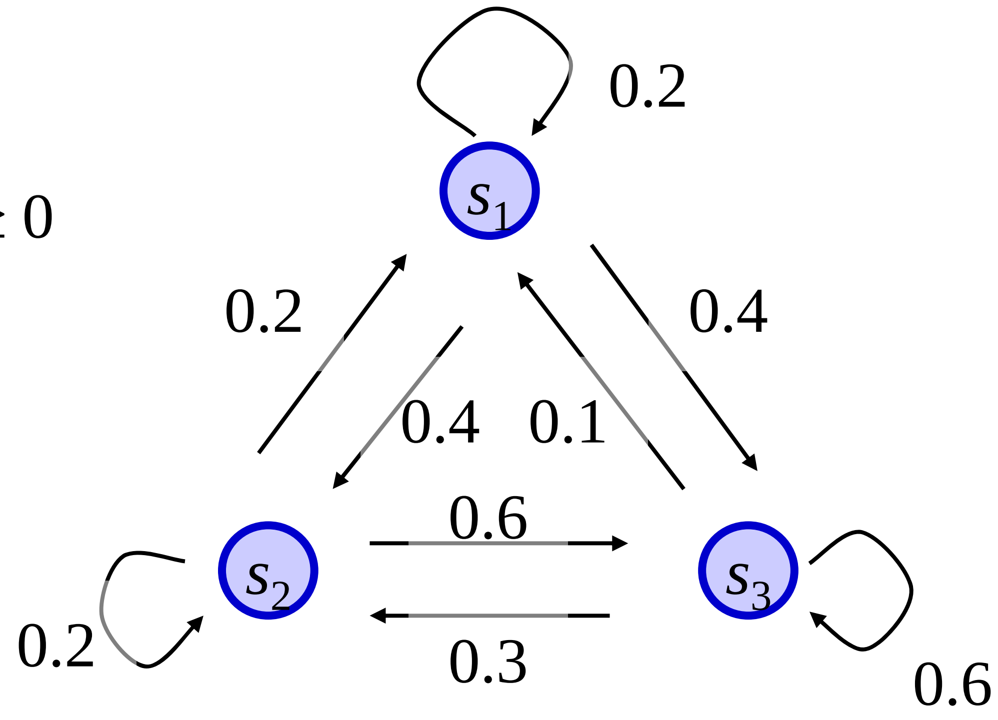
    $$P_{ij} = \Pr(\, i \to j \,) =: P(\, i, j)$$

  - stationary distribution: π

    $$\Leftrightarrow \pi\, P = \pi, \;\; |\pi| = 1, \pi \geq 0$$

$$P = \begin{pmatrix} 0.2 & 0.4 & 0.4 \\ 0.2 & 0.2 & 0.6 \\ 0.1 & 0.3 & 0.6 \end{pmatrix}$$

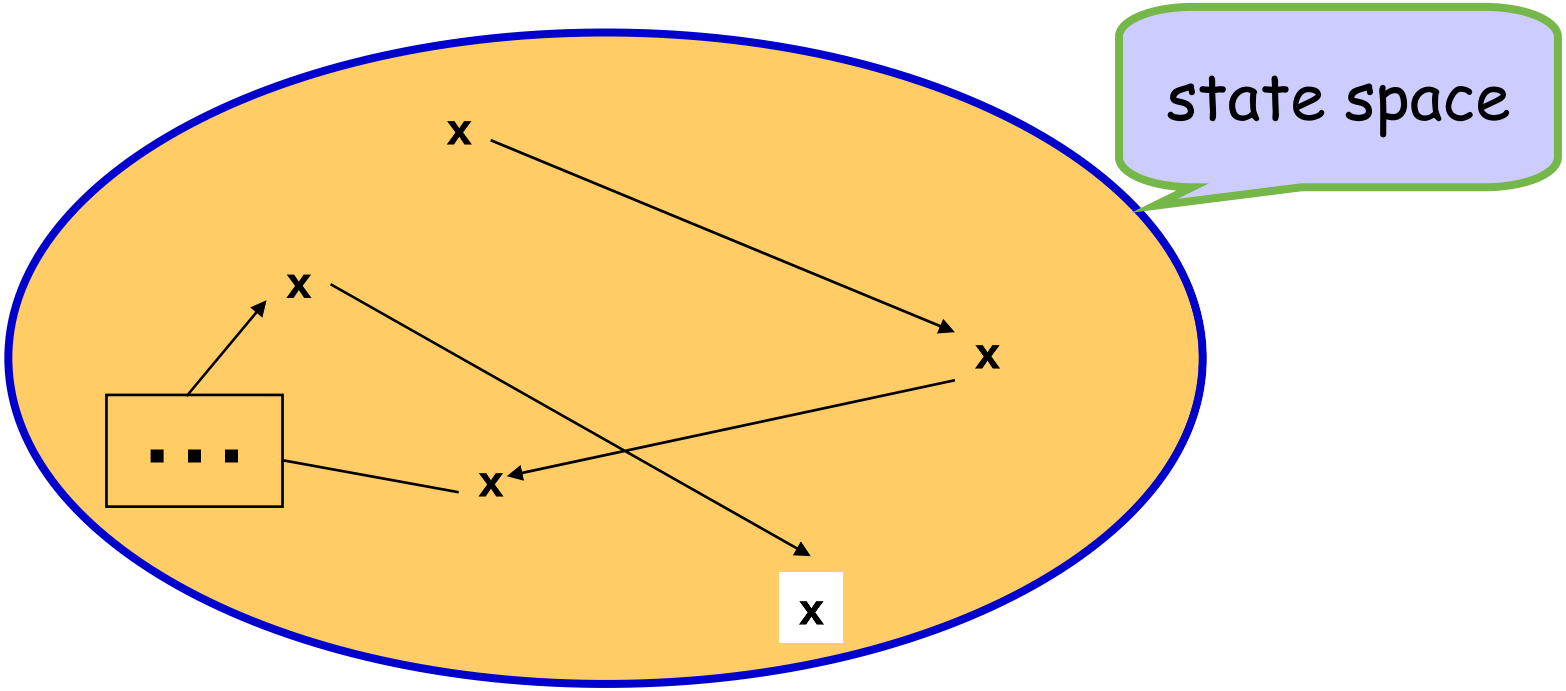
$$\pi = (1/7, 2/7, 4/7)$$



underlying graph

Basic idea of "sampling via Markov chain"

1. Design a Markov chain whose stat. dist = aiming dist.
2. Generate a sample from stat. dist. after many tran.s.



outputs after many transitions according to asymptotically stationary distribution

Applications of MCMC

- counting-hard
- huge state space

MCMC works powerfully for sampling-hard objects

- ➢ Easy to design a sampler for an objective distribution

  cf) simulated annealing ← Pr[global opt.] = 1, Pr[other sol.s] = 0

- ➢ Many applications
  - Contingency table
  - Ising model
  - Permanent
  - Spanning tree
  - Numerical integration (Monte Carlo integration)
  - Optimization

in this talk, deal with

Ex. 1. Contingency table

as an ex. of sampling via Markov chain

- ✓ matrix of non-negative integers
- ✓ satisfies (given) marginal sums

						12
						18
5	4	3	7	5	6	30

Problem

Given: marginal sums

Output: a contingency table u.a.r.

Ex. 1. Contingency table

- ✓ matrix of non-negative integers
- ✓ satisfies (given) marginal sums

						12
						18
5	4	3	7	5	6	30

5	4	3	0	0	0	12
0	0	0	7	5	6	18
5	4	3	7	5	6	30

table A

4	3	1	3	1	0	12
1	1	2	4	4	6	18
5	4	3	7	5	6	30

table B

0	0	0	1	5	6	12
5	4	3	6	0	0	18
5	4	3	7	5	6	30

table C

Problem

Given: marginal sums

Output: a contingency table u.a.r.

Ex. 1. Contingency table

- ✓ matrix of non-negative integers
- ✓ satisfies (given) marginal sums

						12
						18
5	4	3	7	5	6	30

counting # of tables satisfying given marginal sums

$\Rightarrow$ #P-complete (NP-hard), even when size of table is $2 \times n$

[Dyer, Kannan, and Mount '97]

Problem

Given: marginal sums

Output: a contingency table u.a.r.

sampling via Markov chain

Previous works of contingency tables

1985, Diaconis and Effron, exact test with uniform sampler ,

1995, Diaconis and Saloff-Coste, approx. sampler for $m^* \times n^*$ table,

1997, Dyer, Kannan and Mount, **#P-completeness**,

2000, Dyer and Greenhill, approx. sampler for $2 \times n$ table,

2002, Cryan et al., approx. sampler for $m^* \times n$ table

2003, Kijima and Matsui, perfect sampler for $2 \times n$ table

Open problem

Is there a poly-time (approx. or perfect) sampler for $m \times n$ table?

Basic idea of "sampling via Markov chain"

1. Design a Markov chain whose stat. dist = aiming dist.
2. Generate a sample from stat. dist. after many tran.s.

-6: Design of Stationary Distribution

Design of Markov chain with aiming stat. dist.

A Markov chain M is ergodic, iff

limit dist. = stat. dist.

1. the state space is finite,
2. trans. matrix is irreducible, and
3. apperiodic.

every pair is mutually reachable

Them

For a positive function f and transition prob. matrix P,

if detailed balance equations

$$f(x) \cdot P(x, y) = f(y) \cdot P(y, x) \qquad \forall x, y \in \Omega$$

hold, then the stationary distribution

$$\pi(x) = c \cdot f(x)$$

c is the normalizing constant

Metropolis-Hastings, Gibbs sampler, heat-bath chain, etc.

Ex. 1. Markov chain for contingency tables [KM '06]

transition rule is defined as follows

1. choose a consecutive pair of columns $(j, j+1)$ u.a.r. (prob. $1/(n-1)$)

2. change the values of cells in $(j, j+1)$-th columns u.a.r. on possible states

2	3	5
1	4	5
3	7	10

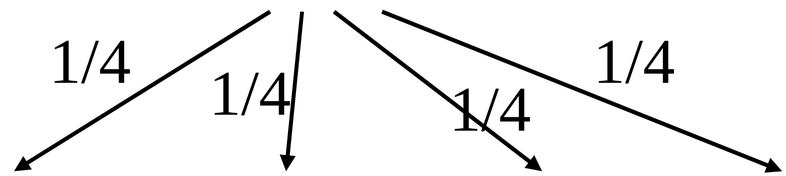
+

$+k$	$-k$
$-k$	$+k$

=> preserve marginal sums

j $j+1$

4	3	2	3	0	0	12
1	1	1	4	5	6	18
5	4	3	7	5	6	30



3	2
0	5

2	3
1	4

1	4
2	3

0	5
3	2

4 possible states

(requirement on non-negativity)

Our Markov chain for contingency tables [KM '06]

Them

Our Markov chain is ergodic, and

the unique stat. dist. of the chain is uniform dist.

proof for the latter claim: detailed balance equations

$$1 \cdot P(x,y) = 1 \cdot P(y,x) \qquad \forall x, y \in \Omega$$ hold.

with the following ex. of a pair (x,y)

X

4	3	2	3	0	0	12
1	1	1	4	5	6	18
5	4	3	7	5	6	30

$\frac{1}{5} \cdot \frac{1}{4}$ (X → Y)

$\frac{1}{5} \cdot \frac{1}{4}$ (Y → X)

Y

4	3	0	5	0	0	12
1	1	3	2	5	6	18
5	4	3	7	5	6	30

transition prob. from X to Y

transition prob. from Y to X

since ...

Our Markov chain for contingency tables [KM '06]

Them

Our Markov chain is ergodic, and

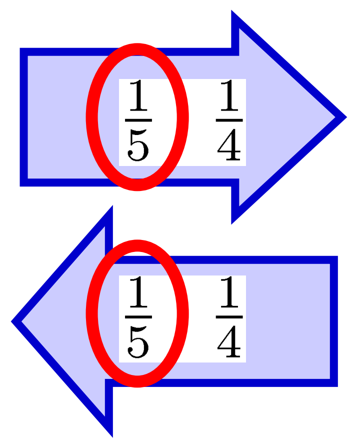
the unique stat. dist. of the chain is uniform dist.

proof for the latter claim: detailed balance equations

$$1 \cdot P(x,y) = 1 \cdot P(y,x) \qquad \forall x, y \in \Omega$$ hold.

X

4	3	2	3	0	0	12
1	1	1	4	5	6	18
5	4	3	7	5	6	30



Y

4	3	0	5	0	0	12
1	1	3	2	5	6	18
5	4	3	7	5	6	30

choose a consecutive pair of indices u.a.r. (w.p. 1/(6-1))

Our Markov chain for contingency tables [KM '06]

Thm

Our Markov chain is ergodic, and

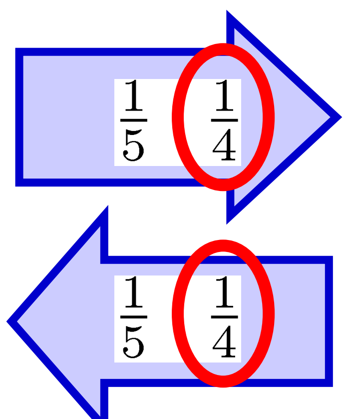
the unique stat. dist. of the chain is uniform dist.

proof for the latter claim: detailed balance equations

$$1 \cdot P(x,y) = 1 \cdot P(y,x) \qquad \forall x,y \in \Omega$$ hold.

X

4	3	2	3	0	0	12
1	1	1	4	5	6	18
5	4	3	7	5	6	30



Y

4	3	0	5	0	0	12
1	1	3	2	5	6	18
5	4	3	7	5	6	30

on the condition (3,4) columns are chosen,
there are common 4 possible states,
since values in other columns are same.

3	2
0	5

2	3
1	4

1	4
2	3

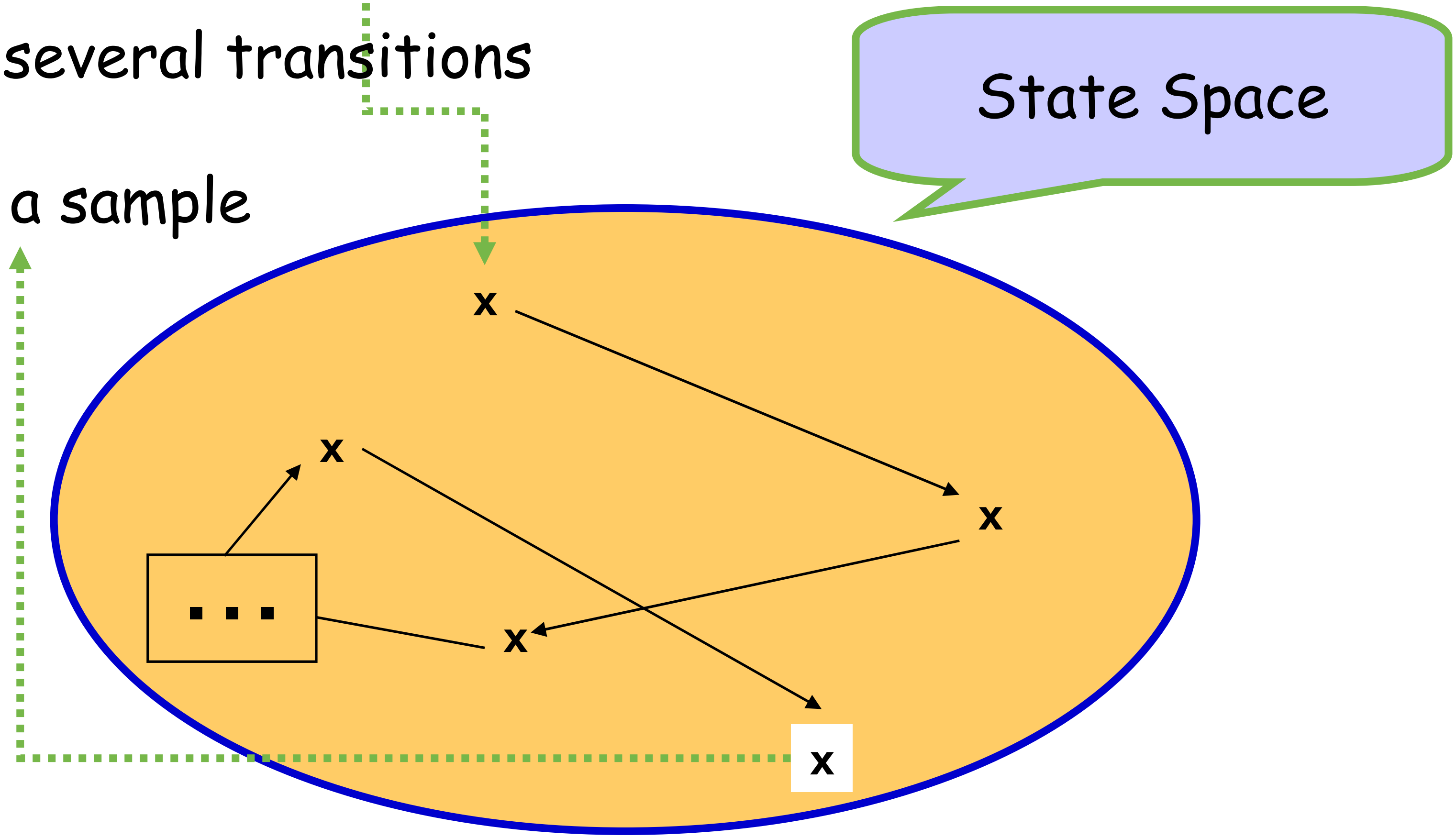
0	5
3	2

Basic idea of sampling via Markov chain

Start from arbitrary initial state

Make several transitions

Output a sample



The output is 'approximately' according to the stat. dist.

Basic idea of "sampling via Markov chain"

1. Design a Markov chain whose stat. dist = aiming dist.
2. Generate a sample from stat. dist. after many tran.s.

-5: Convergence Speed of Markov chain

frontier of MCMC

"How many transitions do we need?"

make the error sufficiently small

- If we have an **approximate sampler**,
  - we have to estimate the mixing time, and
  - bound the total variation distance.
- If we have a **perfect sampler**,
  - we can output a sample exactly according to the stationary distribution.

No error

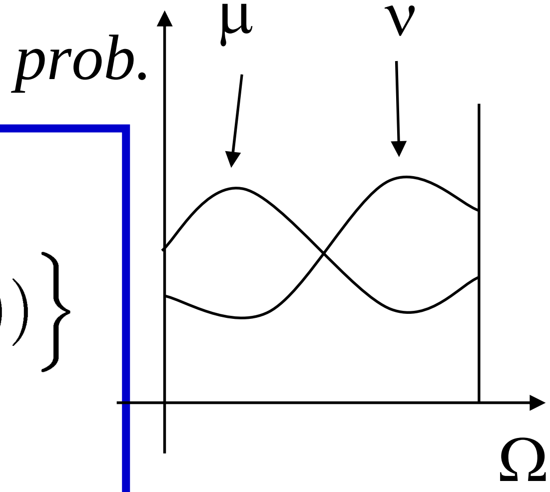
We need not to decide the error rate.

- CFTP (Coupling From The Past) realizes a perfect sampler

Mixing time of a Markov chain is defined as follows

μ, ν: dist. on Ω

Error



Total variation distance

$$d_{\mathrm{TV}}(\mu,\nu) \overset{\text{def.}}{=} \max_{Q\subseteq\Omega}\left\{\sum_{x\in Q}(\mu(x)-\nu(x))\right\}$$

$$\equiv \frac{1}{2}\sum_{x\in\Omega}|\mu(x)-\nu(x)|$$

Ergodic chain M (stat. space Ω, trans. matrix P, stat. dist. π)

t.v.d. is proved sufficiently small in mixing time

Mixing time

$$\tau(\varepsilon) \overset{\text{def.}}{=} \max_{x\in\Omega}\{\min\{t \mid \forall s \geq t,\ d_{\mathrm{TV}}(P_x^t,\pi) \leq \varepsilon\}\}$$

- rapidly mixing if $\tau(\varepsilon) \leq \mathrm{poly.}(\log \Omega, \varepsilon^{-1})$

"How many transitions do we need?"

> make the error sufficiently small

- If we have an **approximate sampler**,
  - we have to estimate the mixing time, and
  - bound the total variation distance.
- If we have a **perfect sampler**,
  - we can output a sample exactly according to the stationary distribution.

> NO ERROR

We need not to decide the error rate.

- CFTP (Coupling From The Past) realizes a perfect sampler review of

INTERMEZZO

-3. PERFECT SIMULATION

Propp and Wilson [1996]

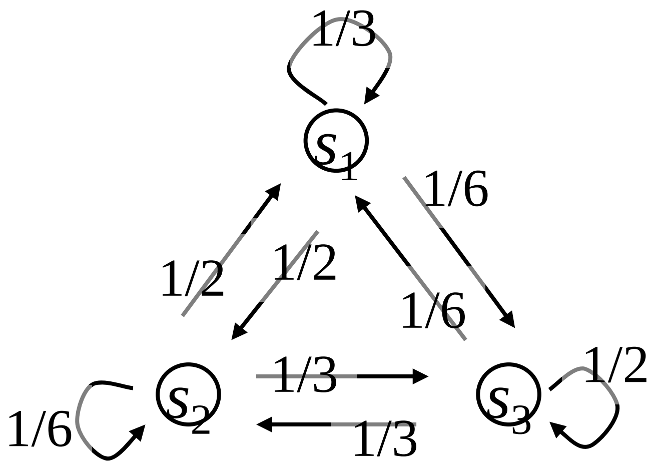
The coupling from the past (CFTP)

- is an ingenious simulation of Markov chain,
- which realize perfect sampling.

sampling from EXACTLY limit distribution

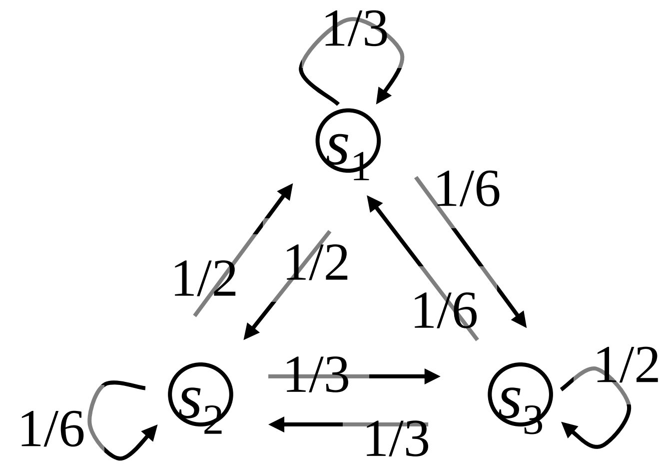
Update function -- with an example

- An ergodic Markov chain *MC*
  - finite state space: s_1, s_2, s_3 ;
  - Transition



Update function -- with an example

- An ergodic Markov chain *MC*



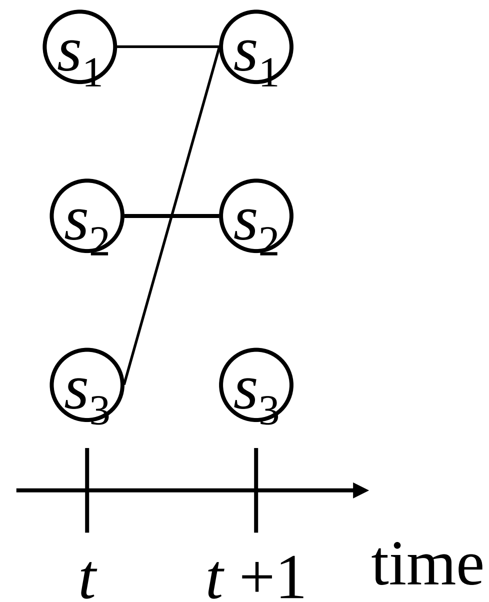
We consider to determine the next state with
- a random number $\lambda \in \{1,\ldots,6\}$ (u.a.r.), and
- an Update function.

current state

λ	s_1	s_2	s_3
1	s_3	s_1	s_2
2	s_2	s_3	s_2
3	s_2	s_1	s_3
4	s_1	s_1	s_3
5	s_1	s_2	s_1
6	s_2	s_3	s_3

This table shows an update function

e.g., 5 ← λ



This is an illustration of a transition.

CFTP Algorithm and Theorem.

Markov chain *MC*:
- Ω: finite state space
- $\Phi_s^t(x, \lambda)$: transition rule
- ergodic

consists of
- 3 steps and
- a stopping condition

CFTP Algorithm

1. Set $T = -1$; set λ: empty;
2. Generate $\lambda[T]$: random number;

   Put $\lambda := (\lambda[T], \lambda[T+1],\ldots,\lambda[-1])$;
3. Start a chain from every element in Ω at period T, run *MC* with λ to period 0.
   a. if coalesce $(\exists y\in\Omega, \forall x\in\Omega, y = \Phi_T^0(x, \lambda)) \Rightarrow$ return y;
   b. otherwise, set $T := T$-1 ; go to step 2.;

CFTP Theorem

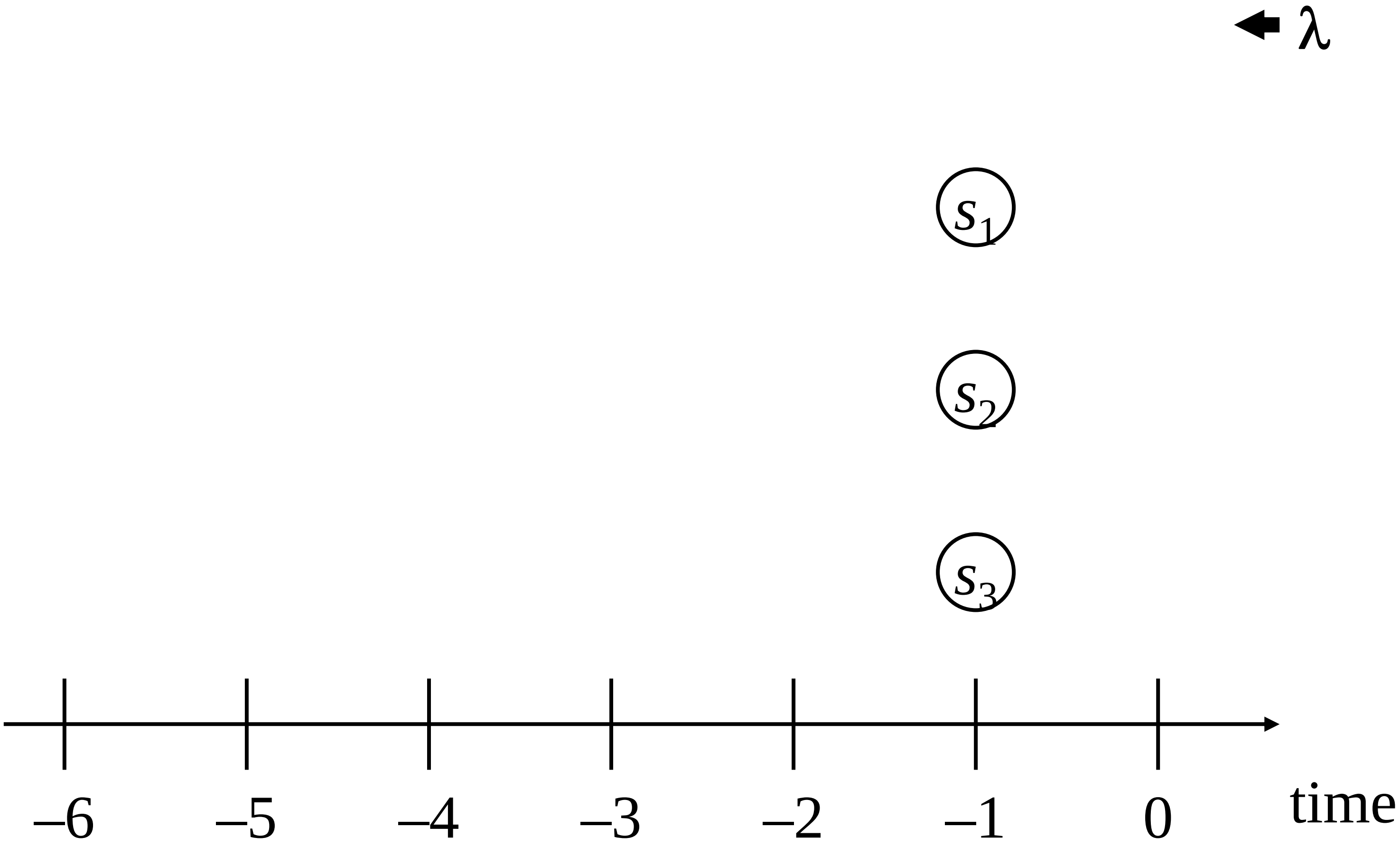
When CFTP Algorithm terminates, the returned value realizes the random variable from stationary distribution, exactly.

In the following slides, I will illustrate the algorithm precisely.

Simulation from *T* = -1

Step 1 is an initializing step

1. set $T = -1$; set λ: empty sequence;

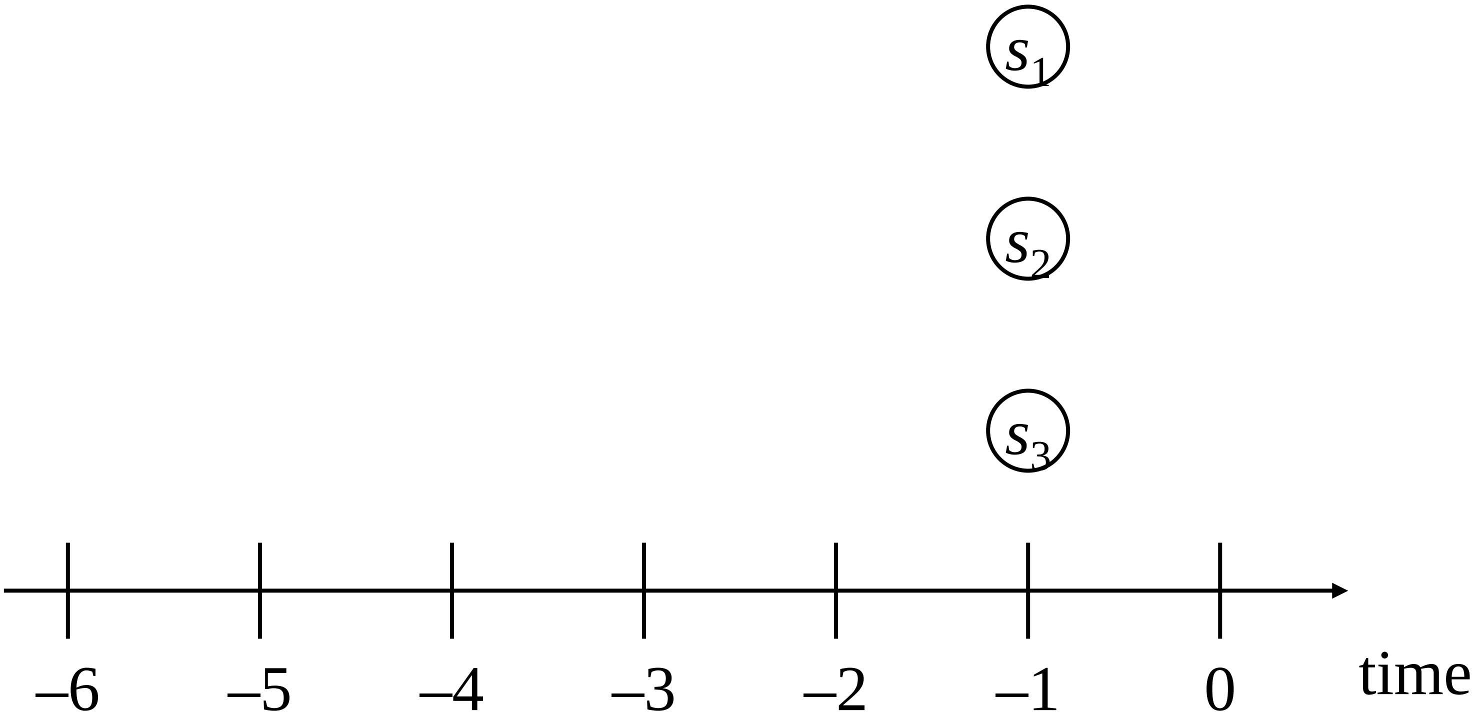


Simulation from T = -1

2. Generate $\lambda[T]$: random number;

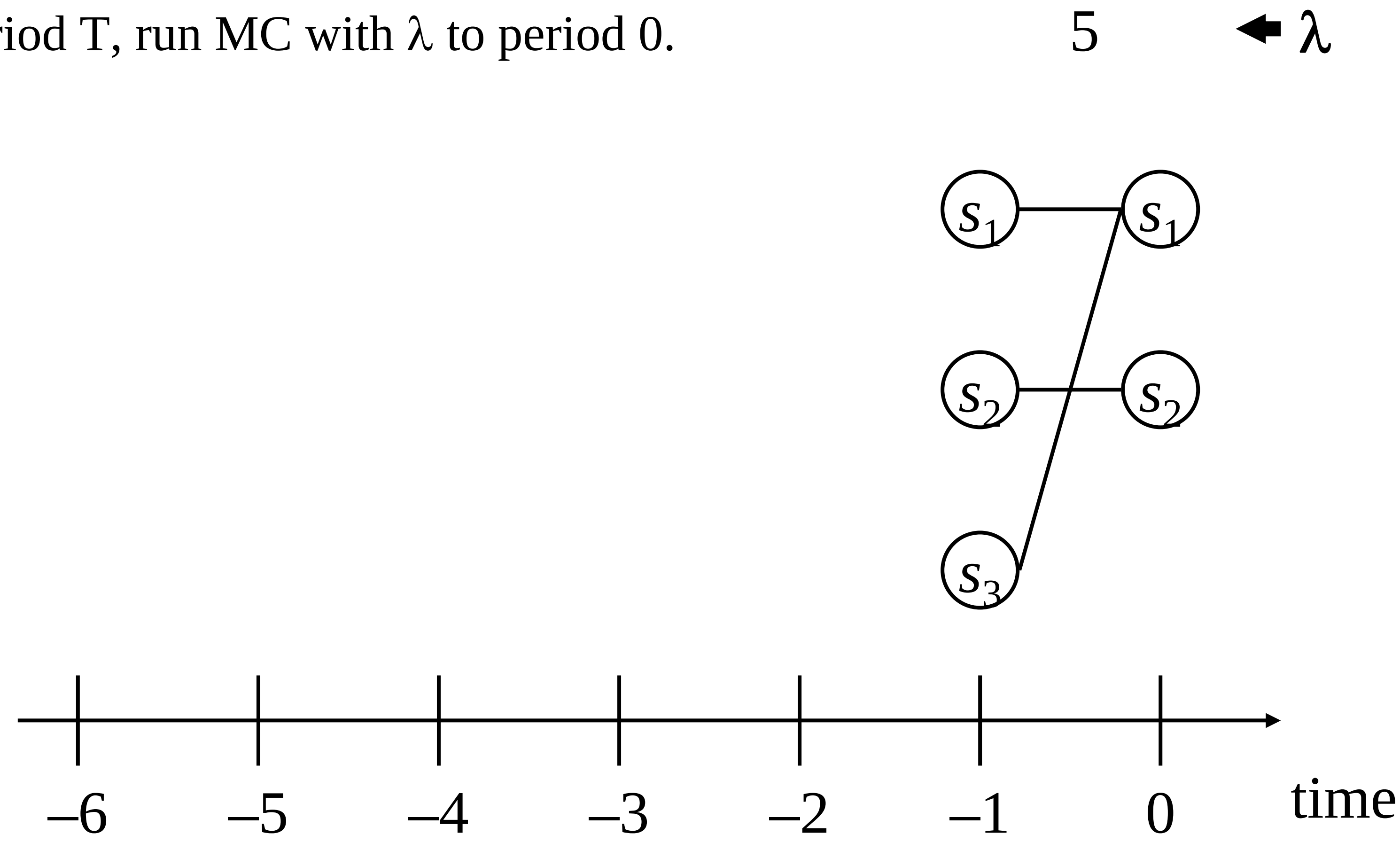
   Put $\lambda := (\lambda[T], \lambda[T+1],\ldots,\lambda[-1])$;

$\lambda(-1) = 5 \leftarrow \lambda$



Simulation from $T = -1$

3. Start a chain from every element in Ω at period T, run MC with λ to period 0.



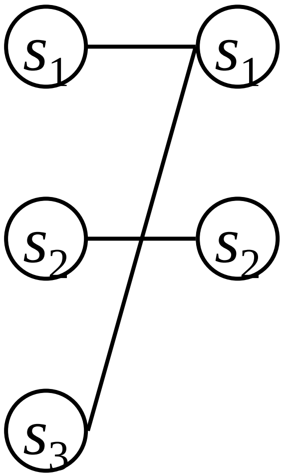
Simulation from T = -1

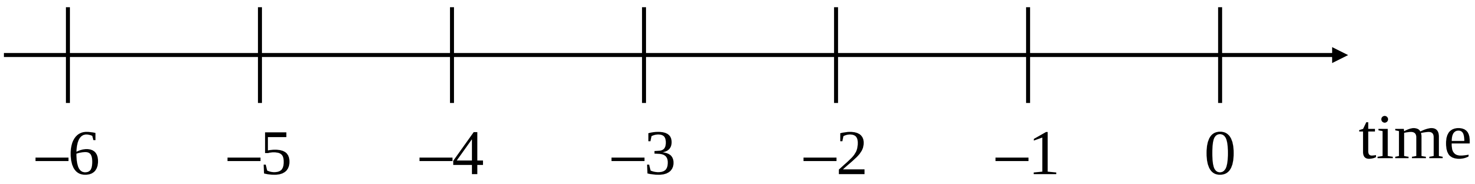
coalesce means
a state at 0 is unique.

3. Start a chain from every element in Ω
   at period T, run MC with λ to period 0.

5 ← λ

   a. if coalesce
      $(\exists y \in \Omega, \forall x \in \Omega, y = \Phi_T^0(x, \lambda))$
      ⇒ return y;
   b. otherwise, set $T := T - 1$;
      go to step 2.;

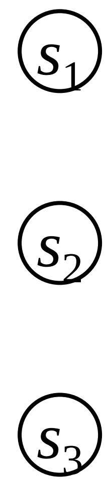


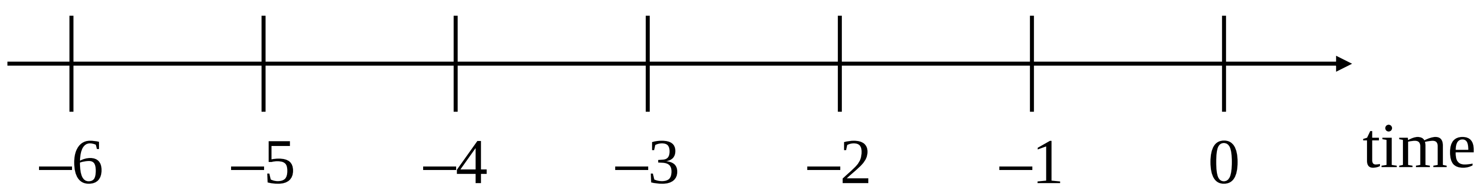


In this case, the states do not coalesce, thus…

Simulation from *T* = -2

3. Start a chain from every element in Ω at period T, run MC with λ to period 0. 5 ← λ
   a. if coalesce
      $(\exists y \in \Omega, \forall x \in \Omega, y = \Phi_T^0(x, \lambda))$
      $\Rightarrow$ return y;
   b. otherwise, set $T := T - 1$;
      go to step 2.;





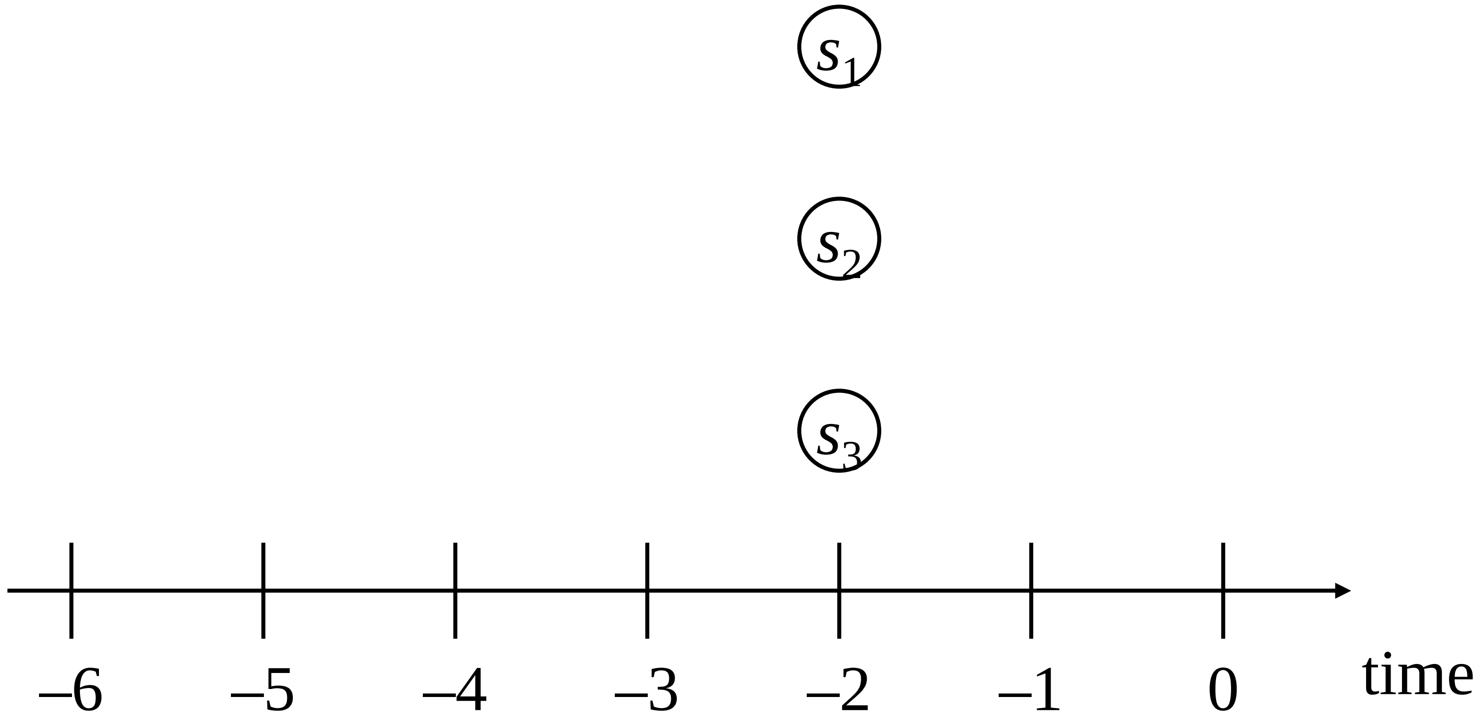
Simulation from T= -2

We will use "5" from -1 to 0 again, generated in the previous iteration.

2. Generate $\lambda[T]$: random number;

Put λ := ($\lambda[T]$, $\lambda[T+1]$,…,$\lambda[-1]$);

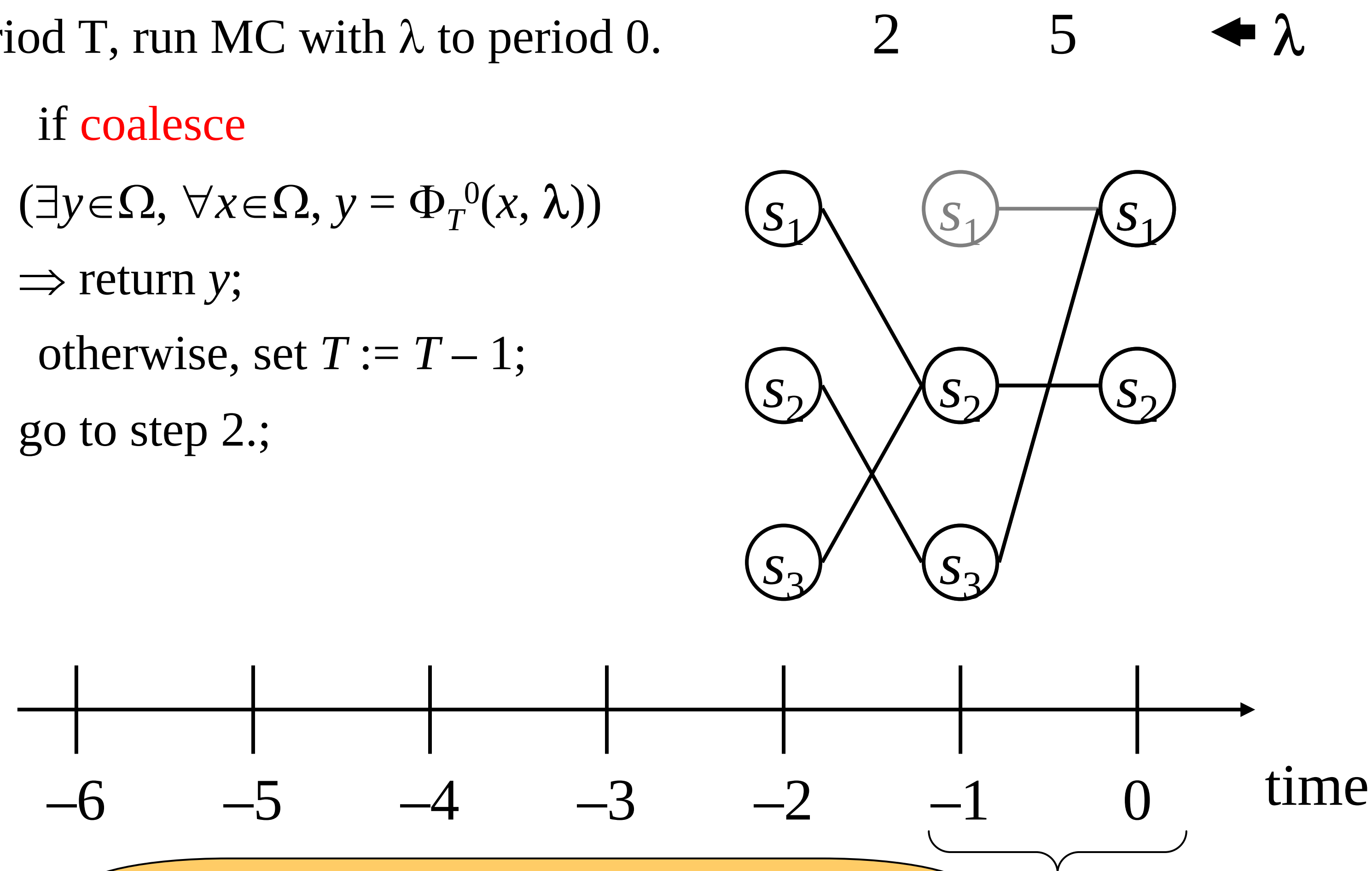
$\lambda(-2) = 2 \quad 5$ ← λ



Simulation from T= -2

We will use "5" from -1 to 0 again, generated in the previous iteration.

3. Start a chain from every element in Ω at period T, run MC with λ to period 0.
   a. if coalesce
   $(\exists y \in \Omega, \forall x \in \Omega, y = \Phi_T^0(x, \lambda))$
   $\Rightarrow$ return y;
   b. otherwise, set $T := T - 1$;
   go to step 2.;



Note
transitions from -1 to 0
= transitions in the previous iteration

Simulation from $T = -3$

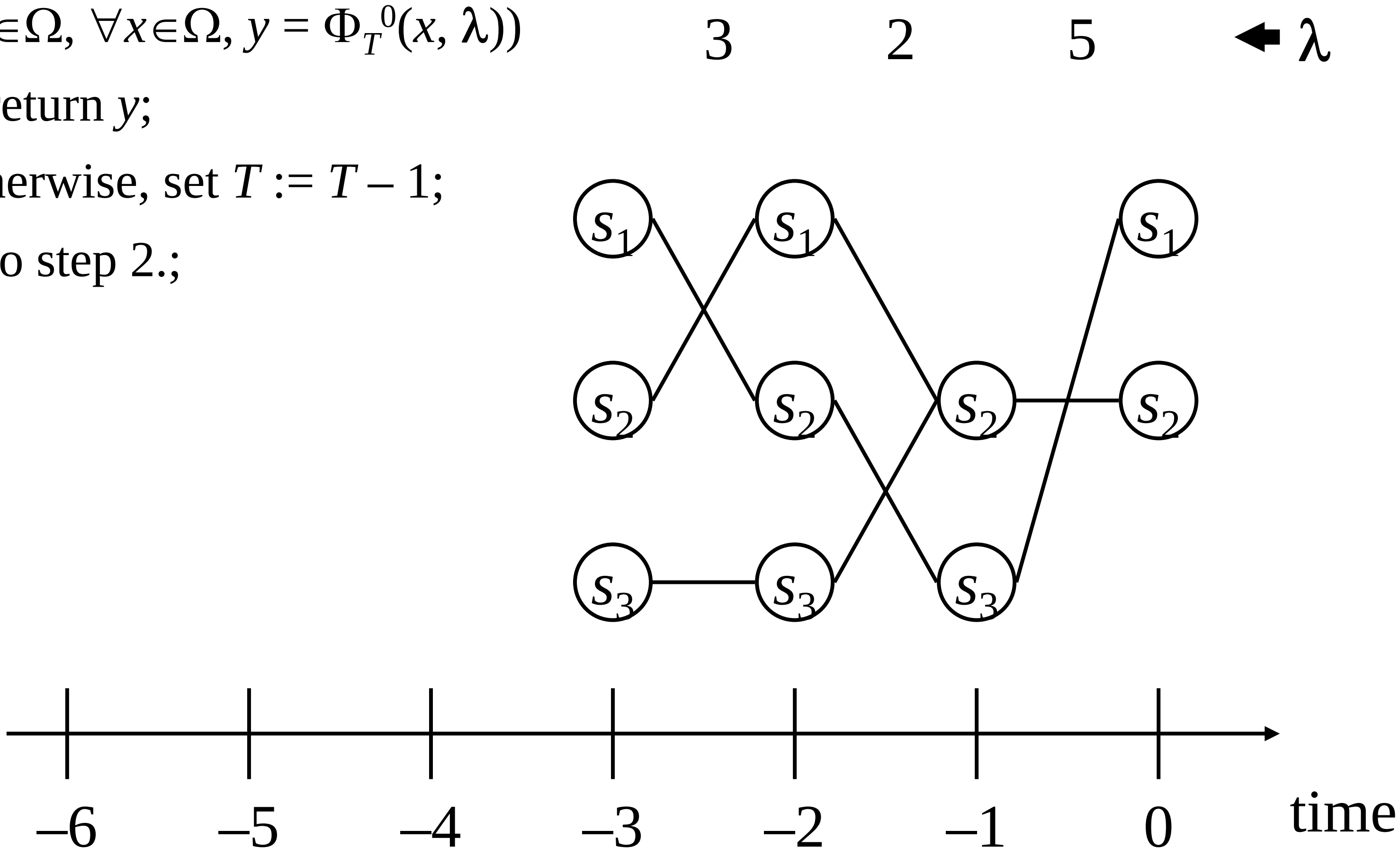
a. if coalesce

$(\exists y \in \Omega, \forall x \in \Omega, y = \Phi_T^0(x, \lambda))$

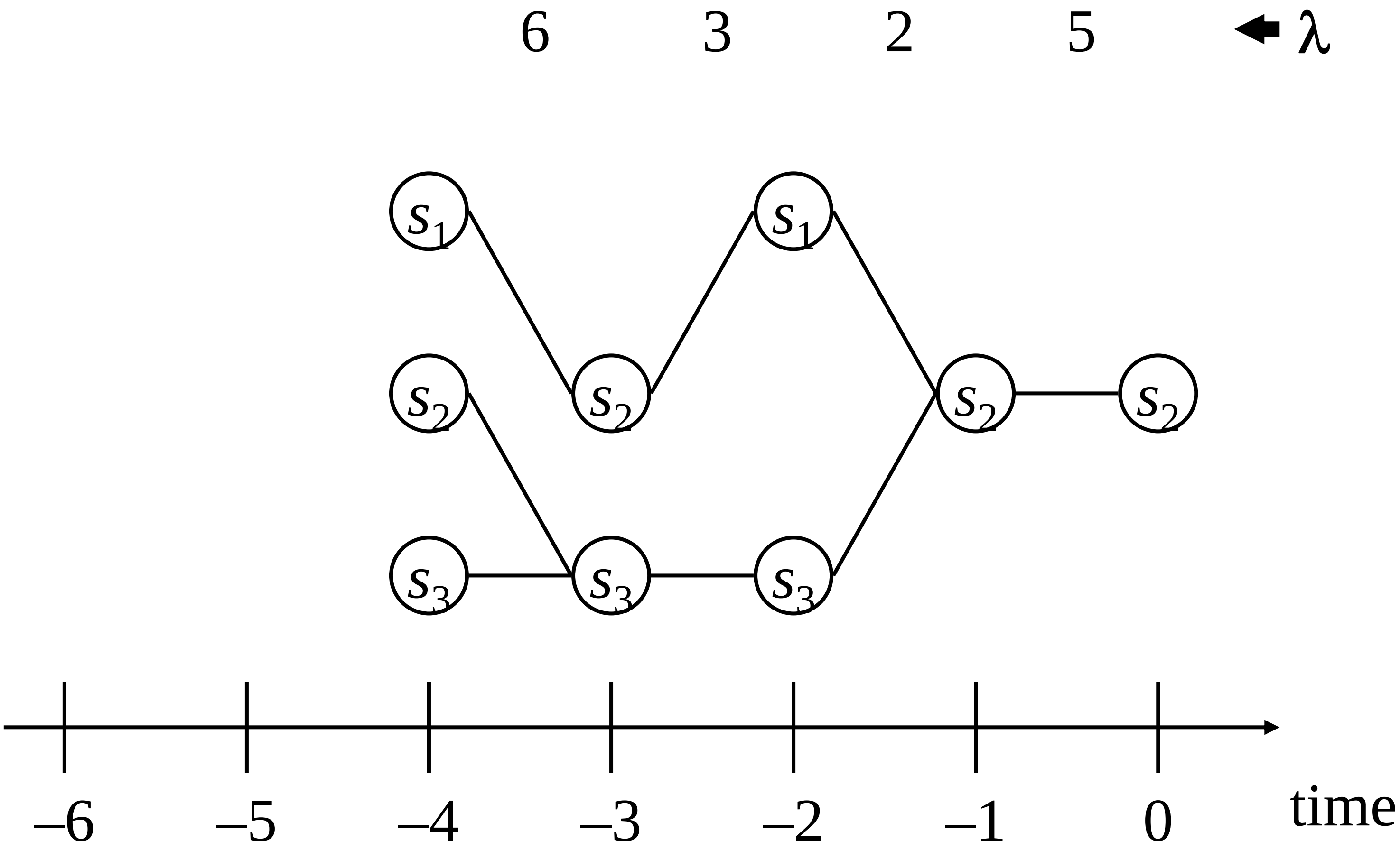
$\Rightarrow$ return y;

b. otherwise, set $T := T - 1$;

go to step 2.;



Simulation from T= -4

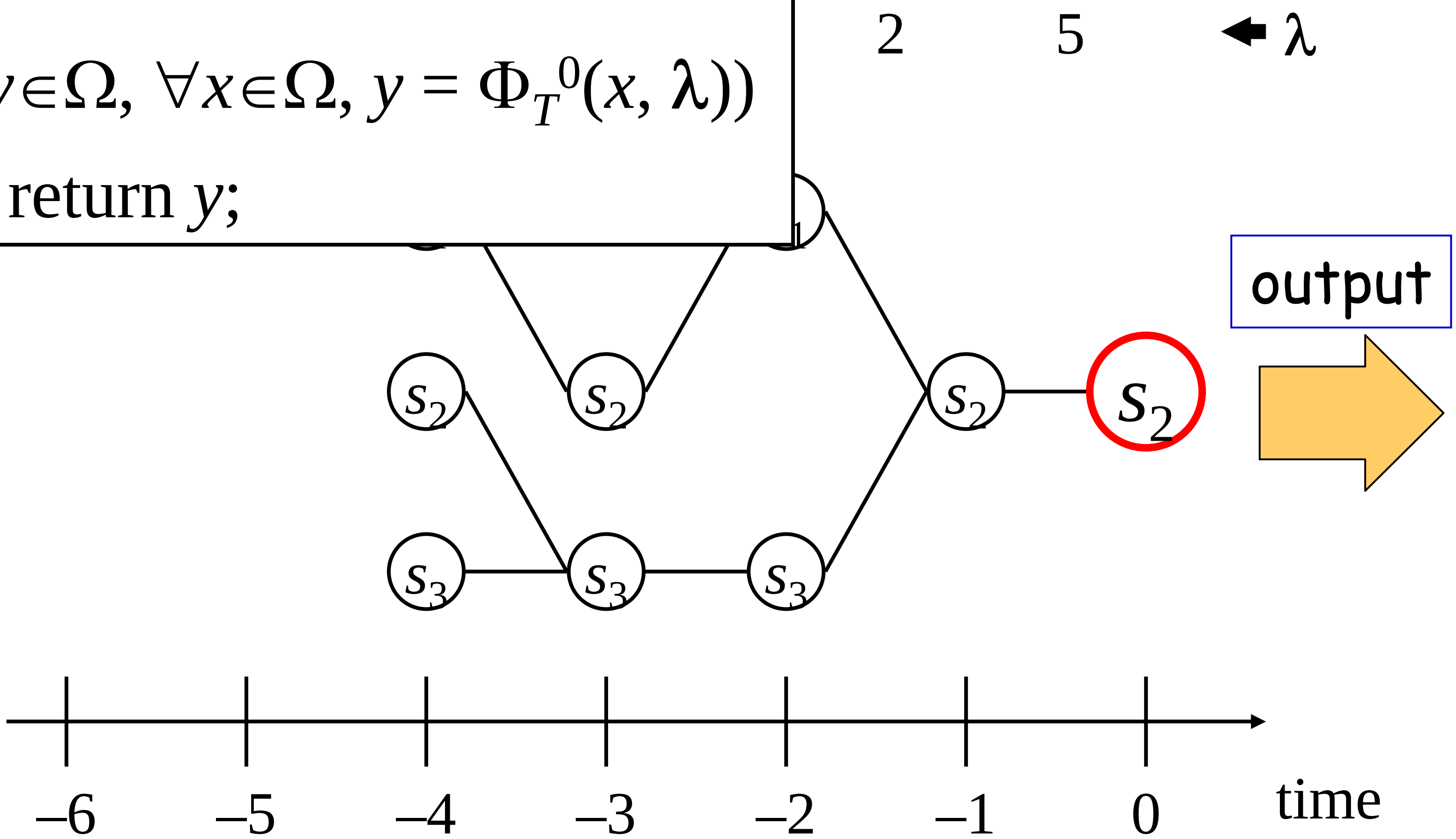


Simulation from $T = -4$

a. if coalesce

$(\exists y \in \Omega, \forall x \in \Omega, y = \Phi_T^0(x, \lambda))$

$\Rightarrow$ return y;



CFTP Algorithm and Theorem.

Markov chain *MC*:
- Ω: finite state space
- $\Phi_s^t(x, \lambda)$: transition rule
- ergodic

CFTP Algorithm

1. Set $T = -1$; set λ: empty;
2. Generate $\lambda[T]$,... , $\lambda[T/2 - 1]$: random number;

   Put $\lambda := (\lambda[T],\dots,\lambda[T/2-1], \lambda[T/2],\dots,\lambda[-1])$;
3. Start a chain from every element in Ω at period T, run *MC* with λ to period 0.
   - a. if coalesce ($\exists y\in\Omega, \forall x\in\Omega, y = \Phi_T^0(x, \lambda)$) $\Rightarrow$ return y;
   - b. otherwise, set $T := T-1$; go to step 2.;

CFTP Theorem

When CFTP Algorithm terminates, the returned value realizes the random variable from stationary distribution, exactly.

what is the idea of CFTP?

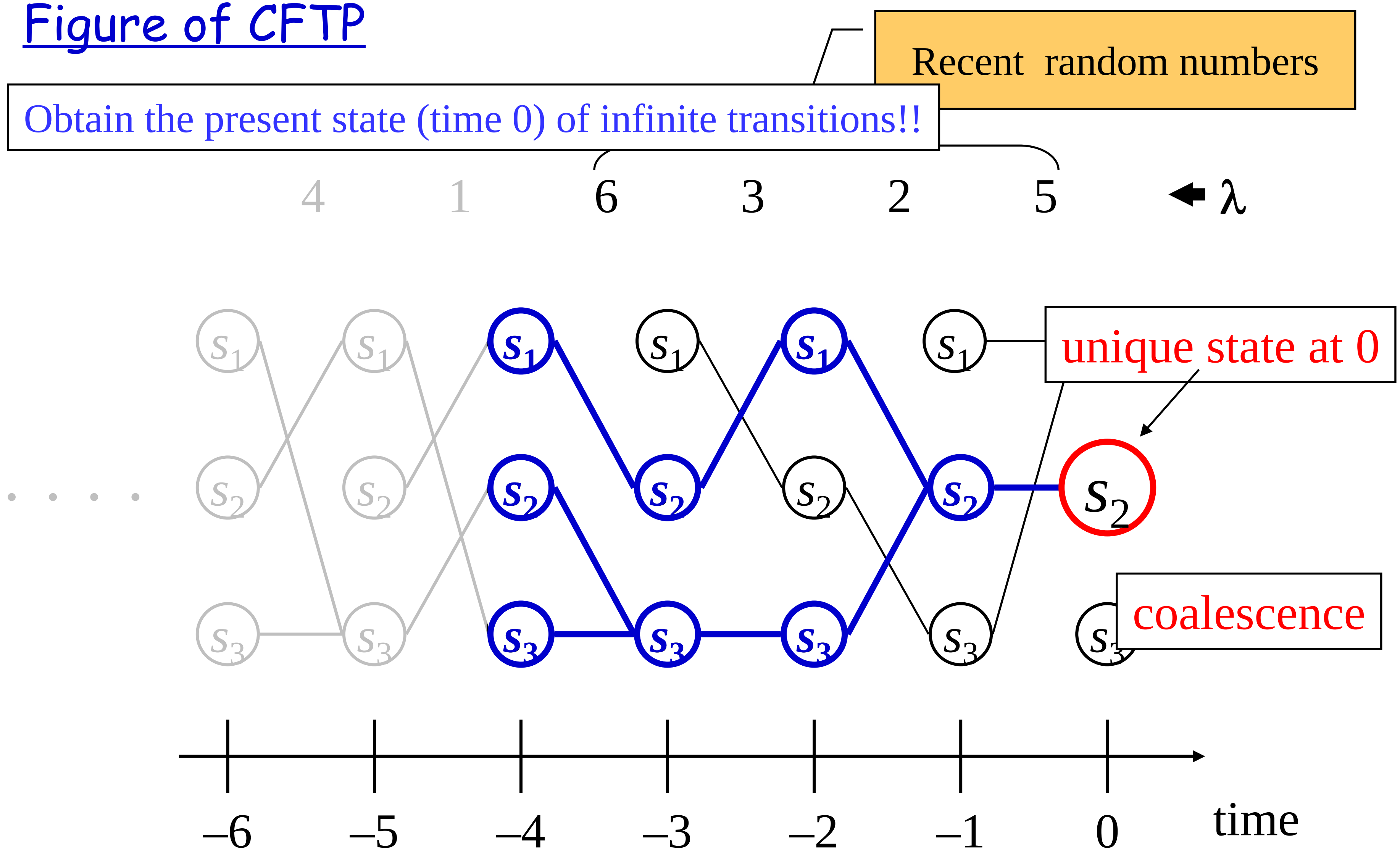
Idea of CFTP (Coupling From the Past)

- Suppose an ergodic chain from infinite past, imaginarily.
  - Present state (state at time 0) is EXACTLY according to the stat. dist.
- What is the present state?
  - Guess from the recent random transitions.

    ⇒ Find the evidence of the present state.

obtained by considering random numbers and transitions with an update function.

Figure of CFTP



Then we can start chains at time -4 from all states with recent random numbers.

Fortunately, we obtain a unique state at time 0. we call this situation 'coalescence'

CFTP Algorithm and Theorem.

Markov chain MC:
- Ω: finite state space
- $\Phi_s^t(x, \lambda)$: transition rule
- ergodic

However it is hard to start …

since we concern with huge state space

CFTP Algorithm

1. Set $T = -1$; set λ: empty;
2. Generate $\lambda[T]$,... , $\lambda[T/2-1]$: random number;

   Put $\lambda := (\lambda[T],\dots,\lambda[T/2-1], \lambda[T/2],\dots,\lambda[-1])$;
3. Start a chain from every element in Ω at period T, run MC with λ to period 0.
   a. if coalesce ($\exists y\in\Omega, \forall x\in\Omega, y = \Phi_T^0(x, \lambda)$) $\Rightarrow$ return y;
   b. otherwise, set $T := T-1$; go to step 2.;

CFTP Theorem

When CFTP Algorithm terminates, the returned value realizes the random variable from stationary distribution, exactly.

We cannot apply this algorithm, directly

-2: Perfect Sampler for two-rowed contingency tables

[KM '06]

Rem. Markov chain for contingency tables [KM '06]

1. choose a consecutive pair of columns $(j, j+1)$ u.a.r. (prob. $1/(n-1)$)

2. change the values of cells in $(j, j+1)$-th columns u.a.r. on possible states

2	3	5
1	4	5
3	7	10

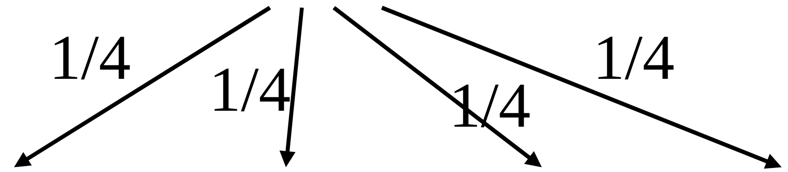
+

$+k$	$-k$
$-k$	$+k$

=> preserve marginal sums

j $j+1$

4	3	2	3	0	0	12
1	1	1	4	5	6	18
5	4	3	7	5	6	30



3	2
0	5

2	3
1	4

1	4
2	3

0	5
3	2

4 possible states

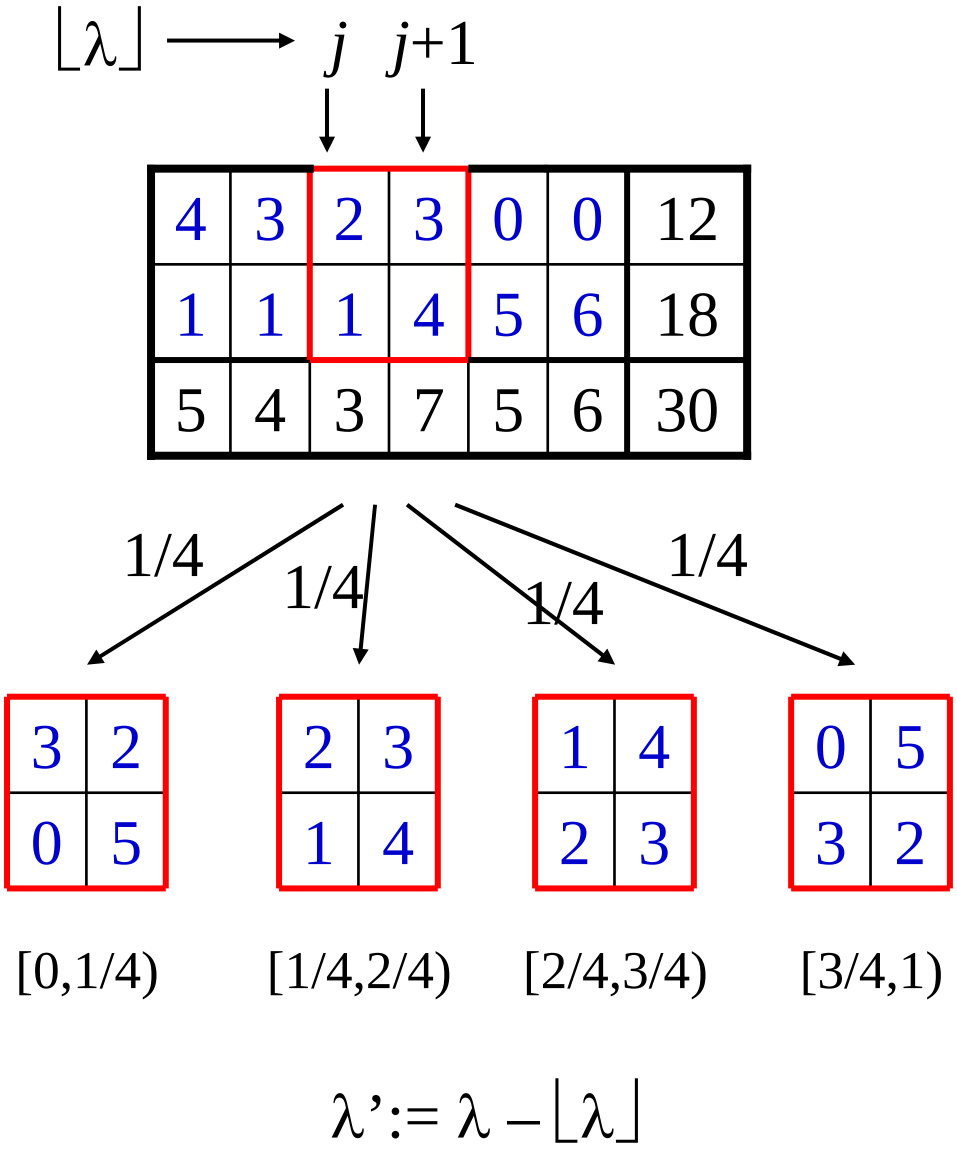
(requirement on non-negativity)

Update function [KM '06]

- generate random real $\lambda \in [1, n)$
- set $j = \lfloor \lambda \rfloor$
- set

$$X(1, j) = \max\{X'[1, j]\} - \lfloor \theta \lambda' \rfloor$$

  - ➢ θ: #of possible states
  - ➢ $\lambda' := \lambda - \lfloor \lambda \rfloor$



Sampling algorithm (monotone CFTP)

1. Set $T = -1$; Set λ: empty sequence;
2. Generate $\lambda[T], \ldots, \lambda$[illegible]
   random number;

   put $\lambda := (\lambda[T], \ldots, \lambda[T/2$ [illegible] $\lambda[T/2], \ldots, \lambda[-1])$;
3. Start chains from x_U, x_L at period T and simulate with λ until period 0;
   a. if coalesce on $Y \Rightarrow$ return Y;
   b. otherwise, set $T := 2T$; go to 2;

We simulate just 2 chains

5	4	3	0	0	0	12
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	18
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	30

X_U : N-W rule

0	0	0	1	5	6	12
5	4	3	6	0	0	18
5	4	3	7	5	6	30

X_L : N-E rule

Them.
The algorithm returns a random vector EXACTLY according to a product form solution.

Key points of the theorem is …

Claim

Coalescence from x_U and x_L $\Leftrightarrow$ Coalescence from all states

- Introduce a partial order on the state space.
- X_U and X_L are the max. and the min., respectively.
- Any transition keeps the partial order.

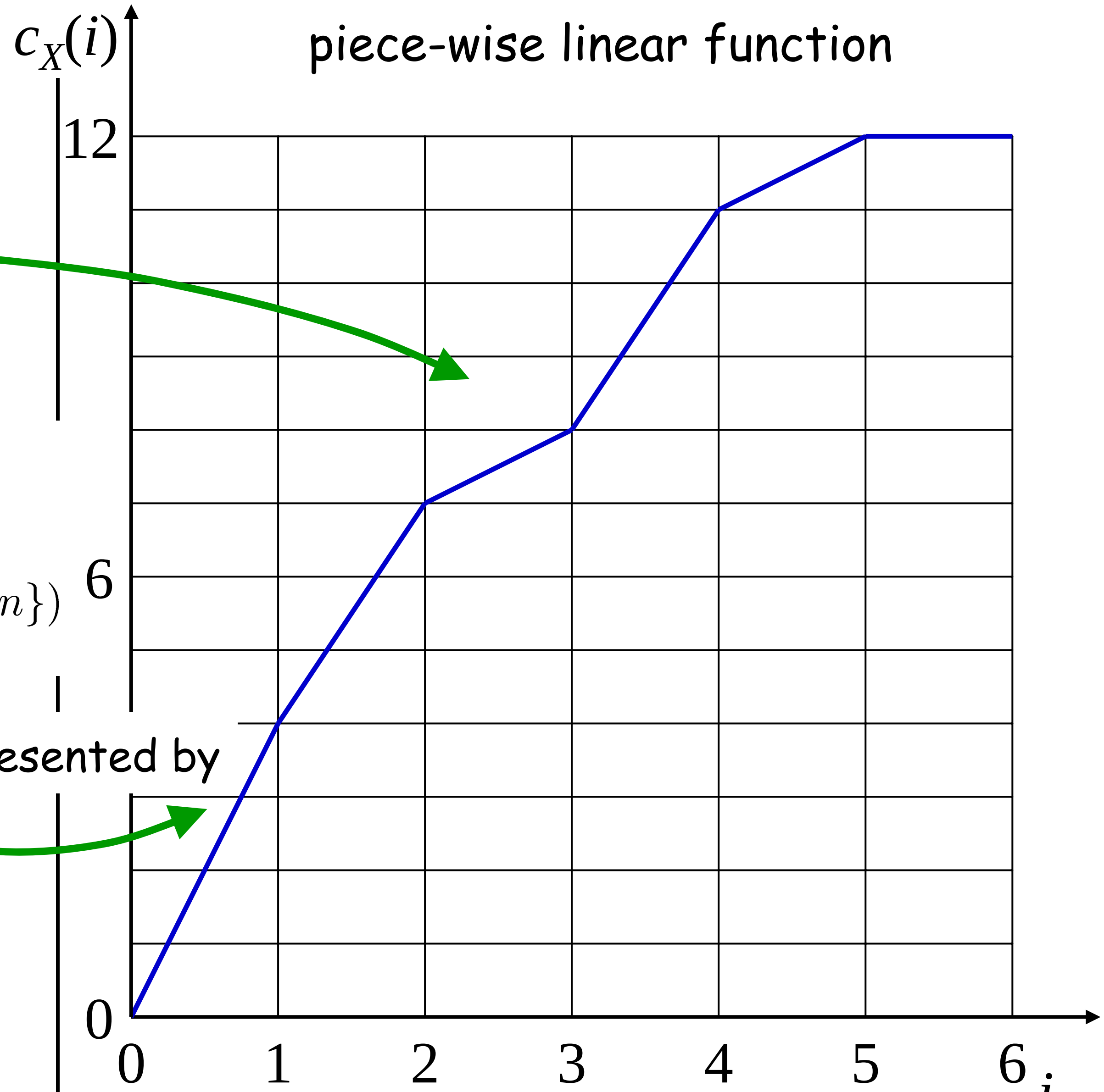
Our Markov chain is a monotone Markov chain

[Propp and Wilson 1996]

Def. cumulative sum vector

Consider to represent a $2{\times}n$ table $\boldsymbol{X}$ by a line, which is a piece-wise linear function of cumulative sum vector c_X defined by

$$c_X(i) \stackrel{\text{def.}}{=} \begin{cases} 0 & (i=0) \\ \sum_{j=1}^{i} X[1,j] & (i \in \{1,\ldots,n\}) \end{cases}$$



e.g.,

4	3	1	3	1	0	12
1	1	2	4	4	6	18
5	4	3	7	5	6	30

represented by

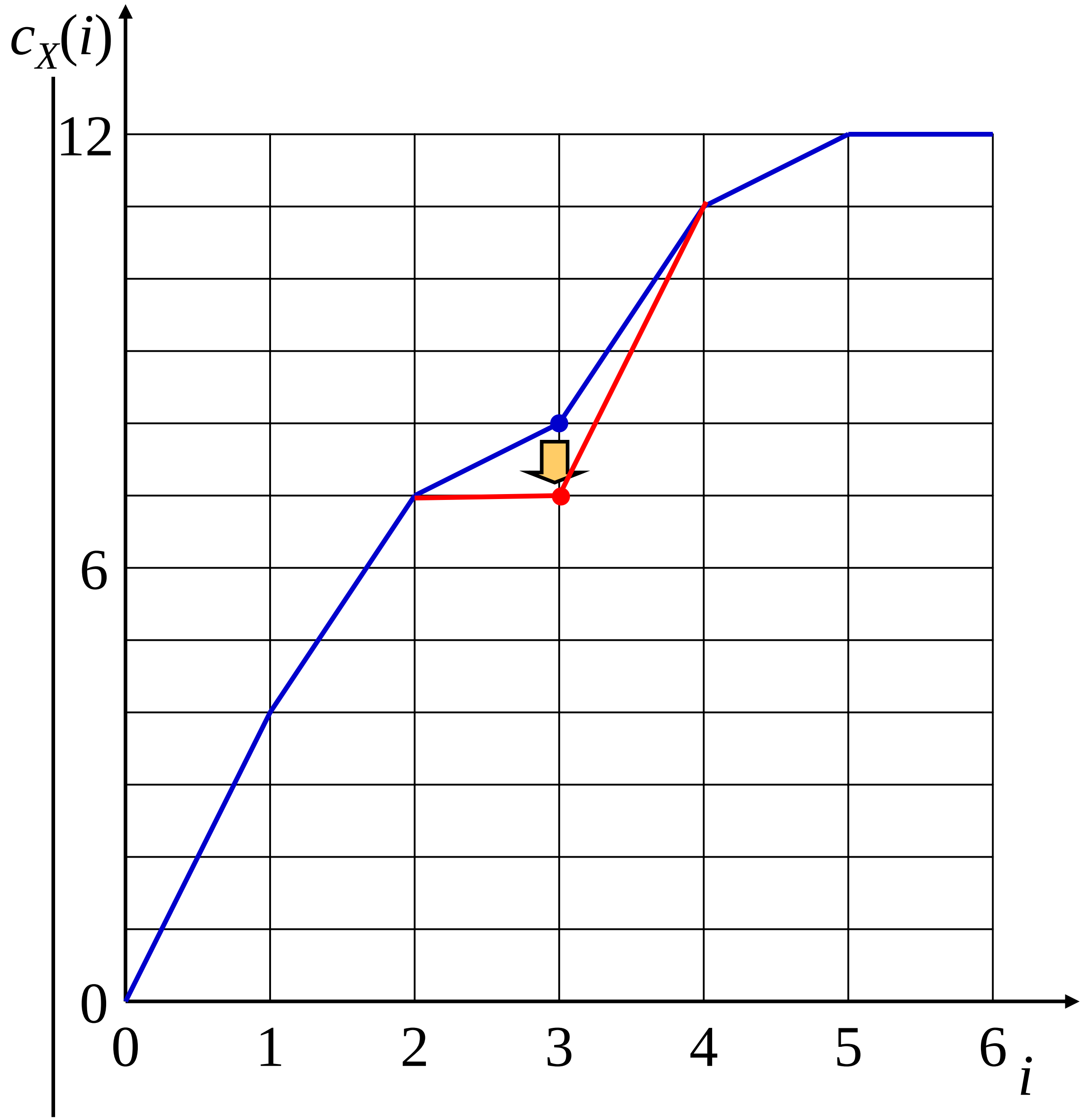
- bijection: $X \to c_X$

transition (represented by a line)

e.g., $\lambda = 3.1$

4	3	1	3	1	0	12
1	1	2	4	4	6	18
5	4	3	7	5	6	30

4	3	0	4	1	0	12
1	1	3	3	4	6	18
5	4	3	7	5	6	30



in the line representation

Choose an index, and
change the point
only at the index.

Def. Partial order

$X \geq Y$

$\Leftrightarrow \ c_X(i) \geq c_Y(i)$

$(\forall i \in \{0,\ldots,n\})$

X

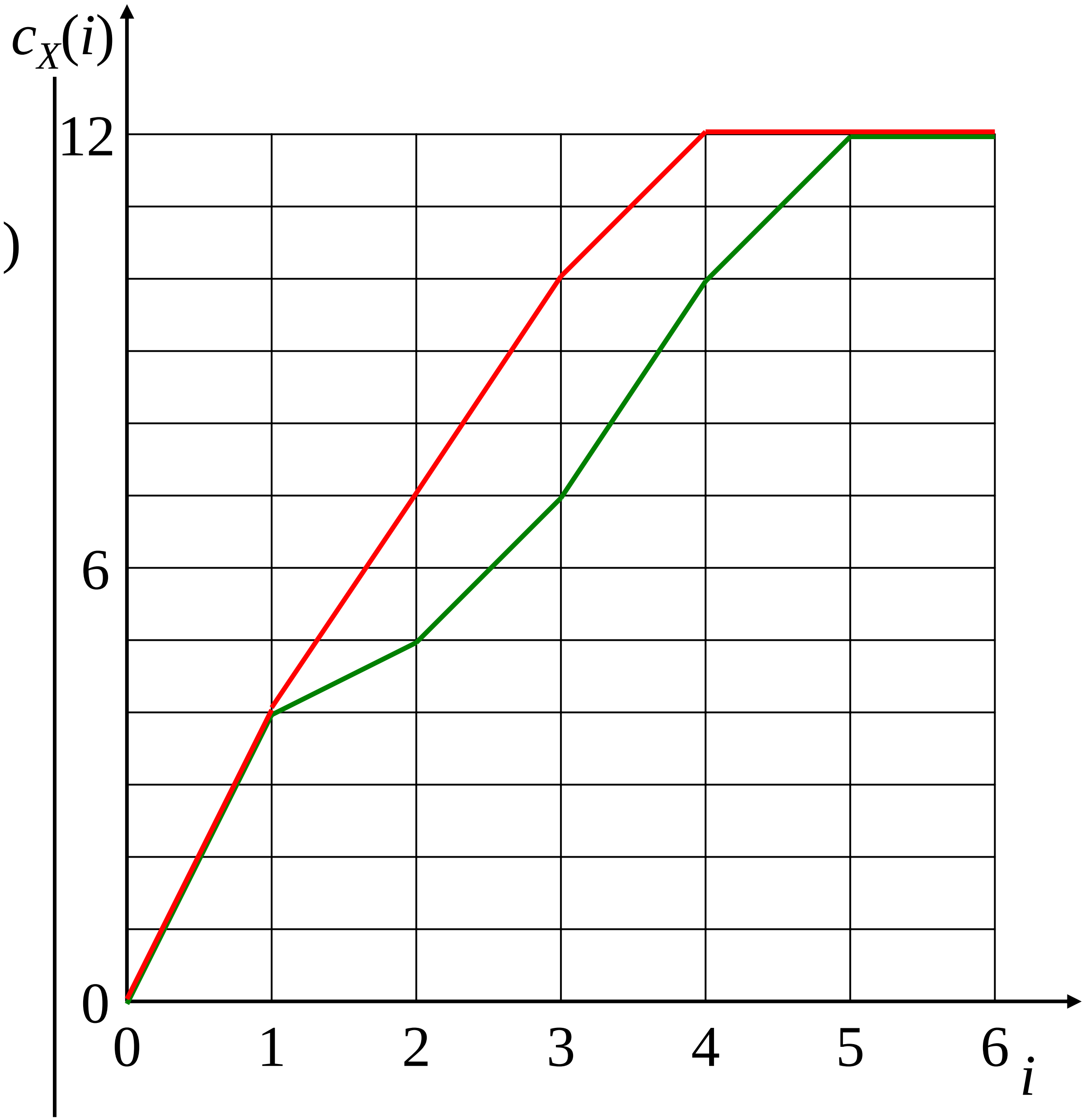
4	3	3	2	0	0	12
1	1	0	5	5	6	18
5	4	3	7	5	6	30

Y

4	1	2	3	2	0	12
1	3	1	4	3	6	18
5	4	3	7	5	6	30

$X \geq Y$

It means that red line is upper than green.



Max, Min on poset

5	4	3	0	0	0	12
0	0	0	7	5	6	18
5	4	3	7	5	6	30

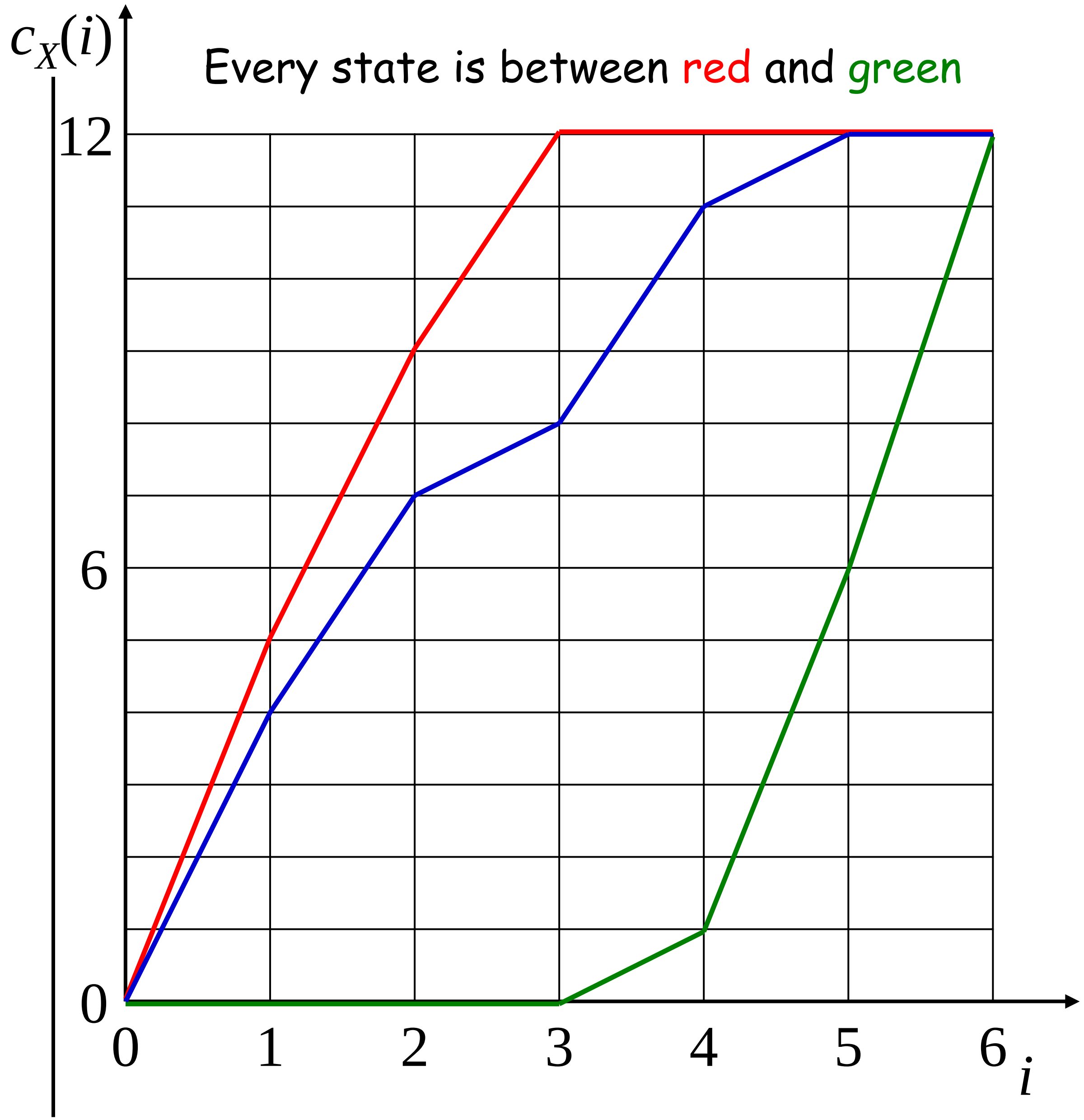
X_{U} : N-W rule

0	0	0	1	5	6	12
5	4	3	6	0	0	18
5	4	3	7	5	6	30

X_{L} : N-E rule

Lemma

$$X_{\mathrm{U}} \geq \forall X \geq X_{\mathrm{L}}$$



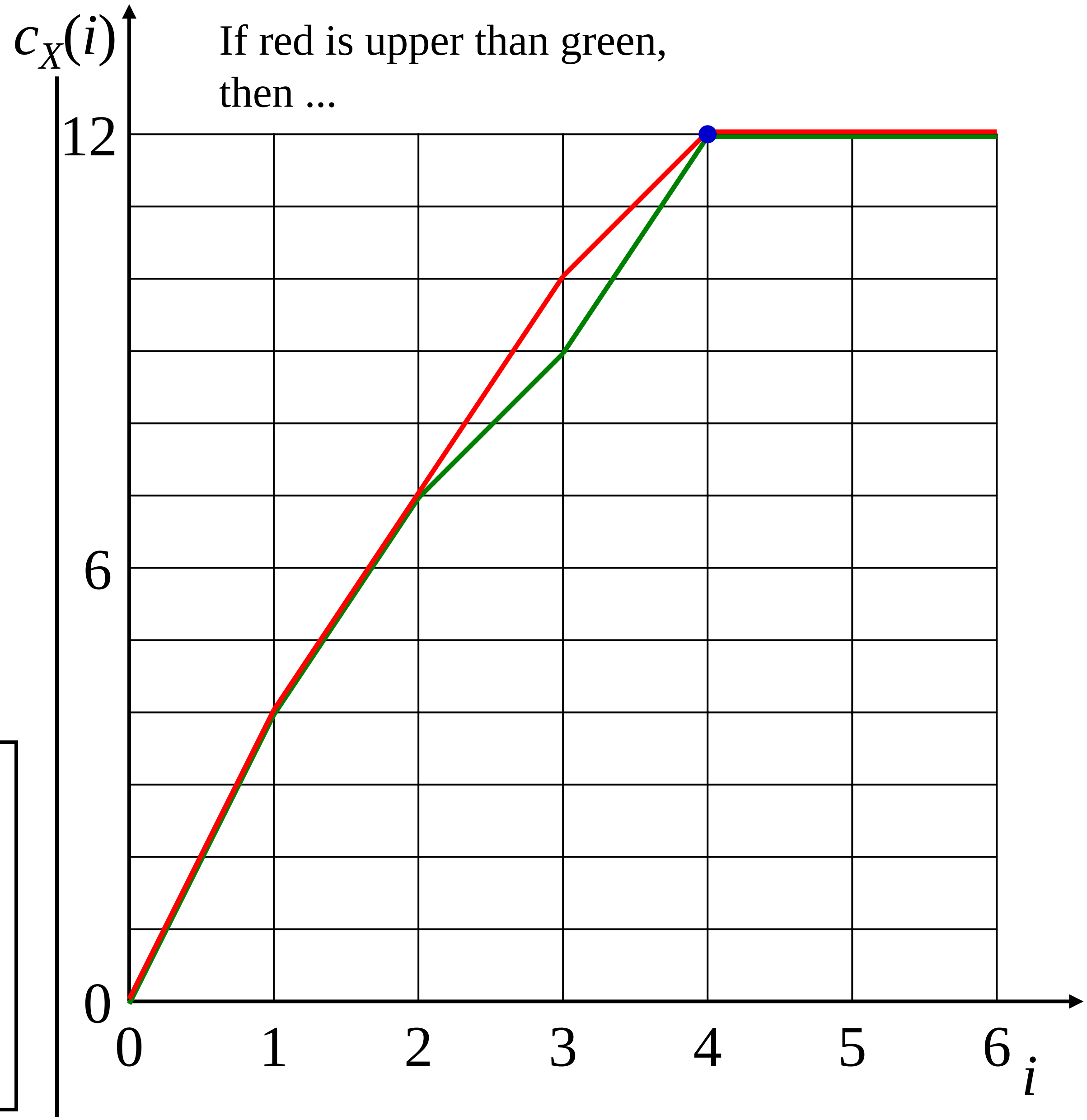
Key lemma

X'

4	3	3	0	2	0	12
1	1	0	7	3	6	18
5	4	3	7	5	6	30

Y'

4	3	2	2	1	0	12
1	1	1	5	4	6	18
5	4	3	7	5	6	30



Lemma

Any transition keeps partial order i.e.,

$\forall(X, Y)$ s.t. $X \geq Y$,

$\phi(X, \lambda) \geq \phi(Y, \lambda)$

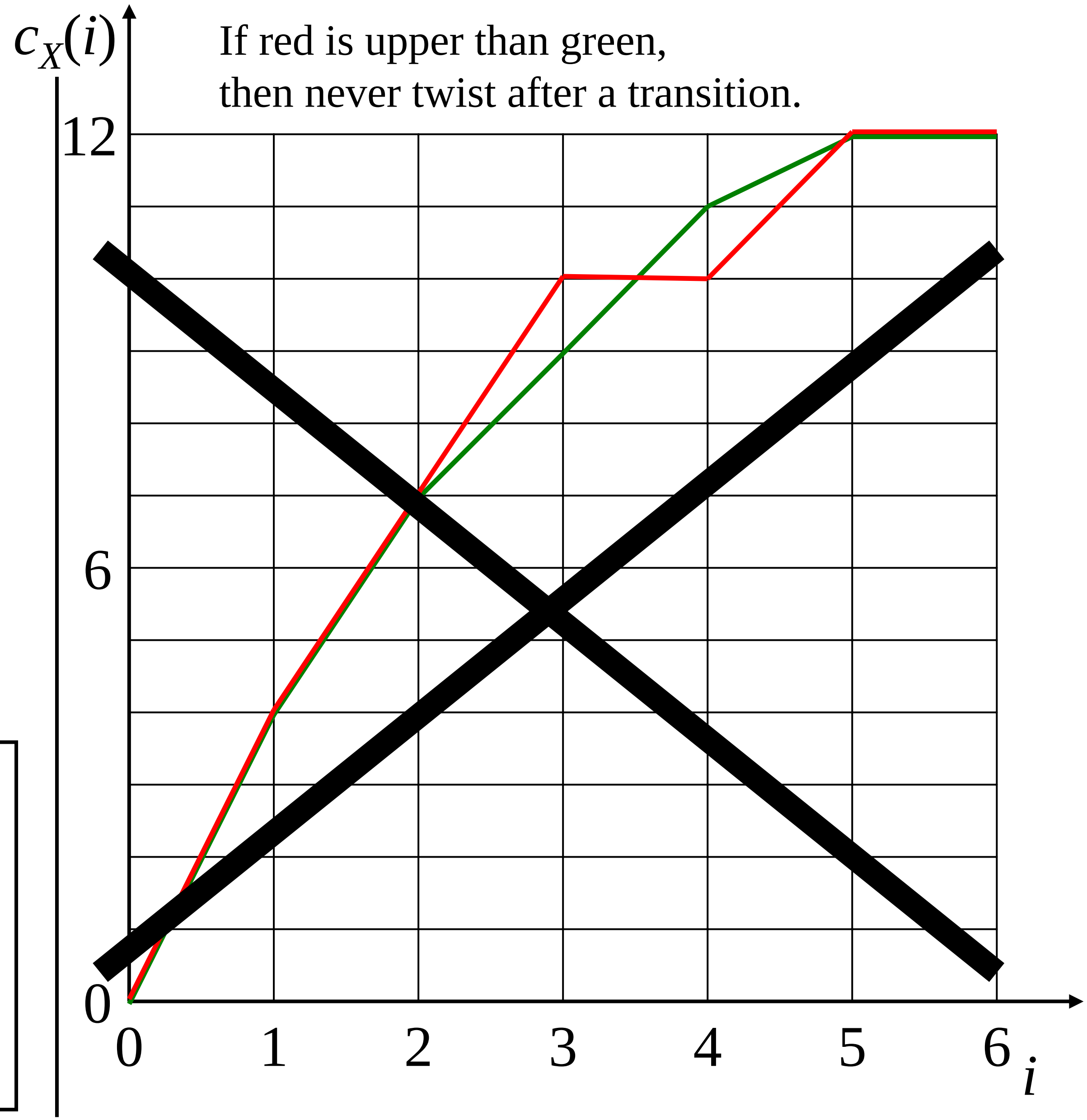
Key lemma

X'

4	3	3	0	2	0	12
1	1	0	7	3	6	18
5	4	3	7	5	6	30

Y'

4	3	2	2	1	0	12
1	1	1	5	4	6	18
5	4	3	7	5	6	30



Lemma

Any transition keeps partial order i.e.,

$\forall(X, Y)$ s.t. $X \geq Y$,

$\phi(X, \lambda) \geq \phi(Y, \lambda)$

Key points of the theorem is …

Claim

Coalescence from x_U and x_L $\Leftrightarrow$ Coalescence from all states

- Introduce a partial order on the state space.
- X_U and X_L are the max. and the min., respectively.
- Any transition keeps the partial order.

Our Markov chain is a monotone Markov chain

[Propp and Wilson 1996]

fig. of coalescence

every state is between red line (= X_U) and green line (= X_L)

5	4	3	0	0	0	12
0	0	0	7	5	6	18
5	4	3	7	5	6	30

4	3	1	3	1	0	12
1	1	2	4	4	6	18
5	4	3	7	5	6	30

0	0	0	1	5	6	12
5	4	3	6	0	0	18
5	4	3	7	5	6	30

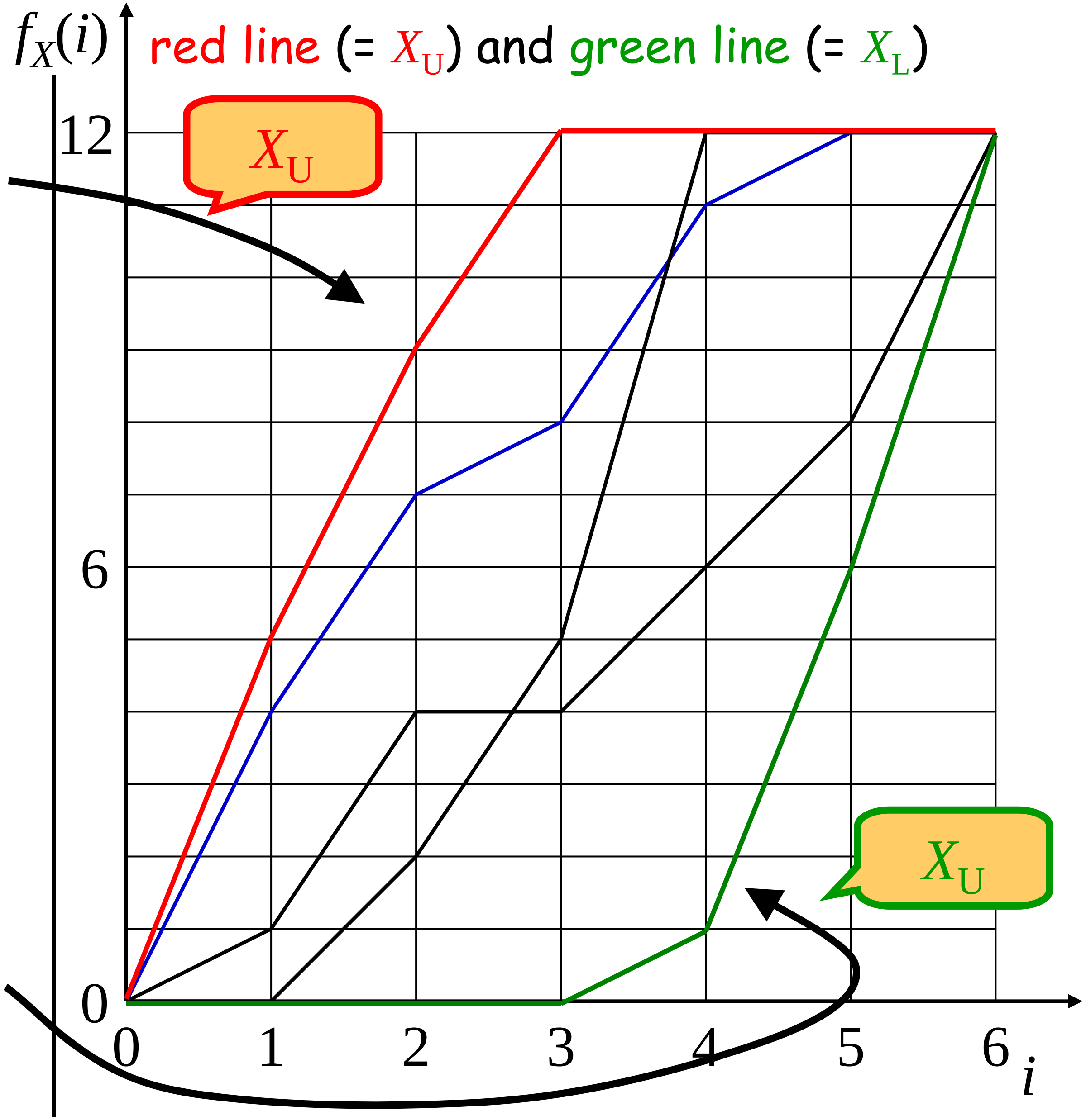


fig. of coalescence

any transition preserve part. order

=> every state is between red & green

5	4	0	3	0	0	12
0	0	3	4	5	6	18
5	4	3	7	5	6	30

4	3	0	4	1	0	12
1	1	3	3	4	6	18
5	4	3	7	5	6	30

0	0	0	1	5	6	12
5	4	3	6	0	0	18
5	4	3	7	5	6	30

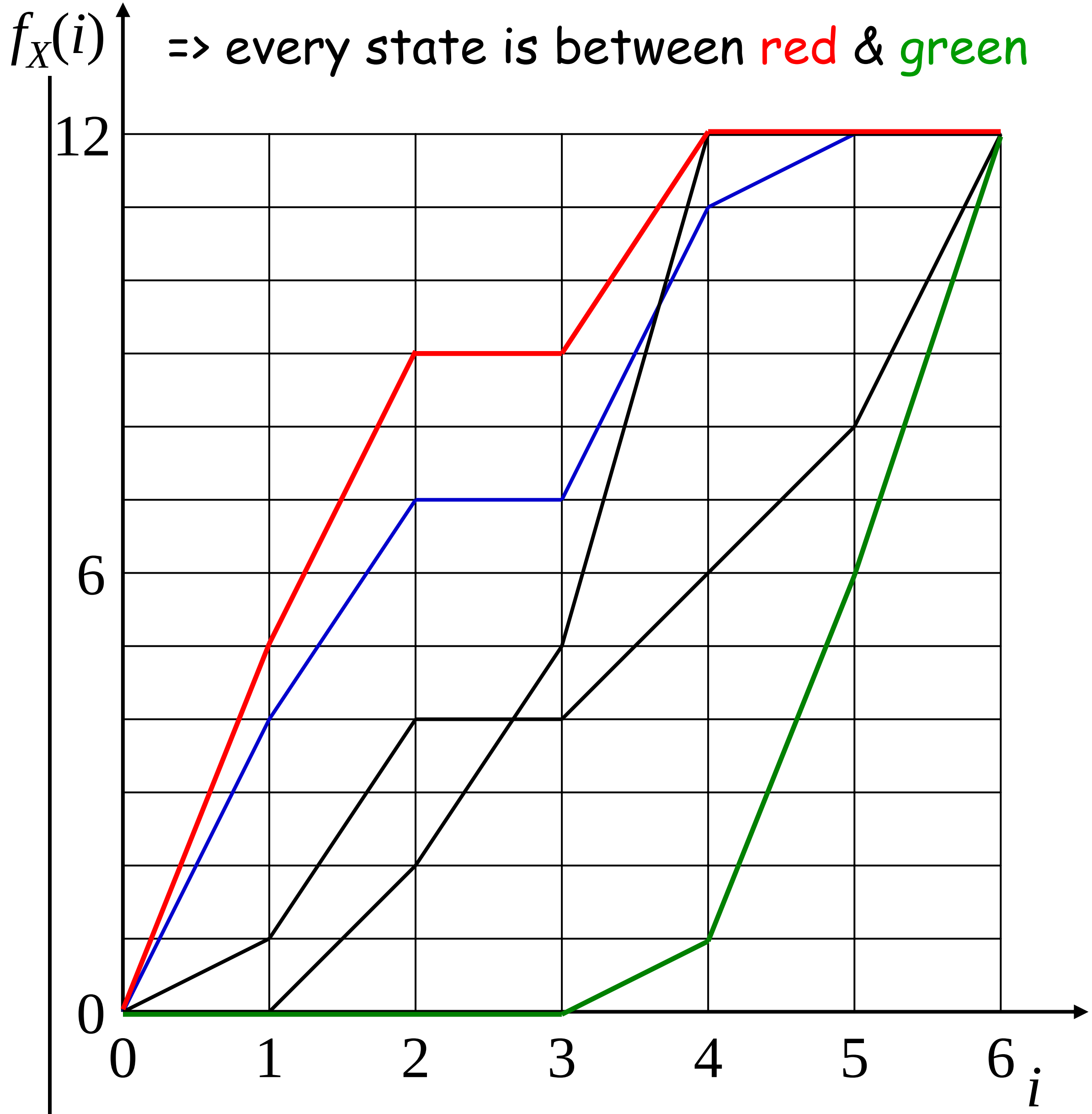


fig. of coalescence

any transition preserve part. order

=> every state is between red & green

5	4	0	2	1	0	12
0	0	3	5	4	6	18
5	4	3	7	5	6	30

4	3	0	3	2	0	12
1	1	3	4	3	6	18
5	4	3	7	5	6	30

0	0	0	4	2	6	12
5	4	3	3	3	0	18
5	4	3	7	5	6	30

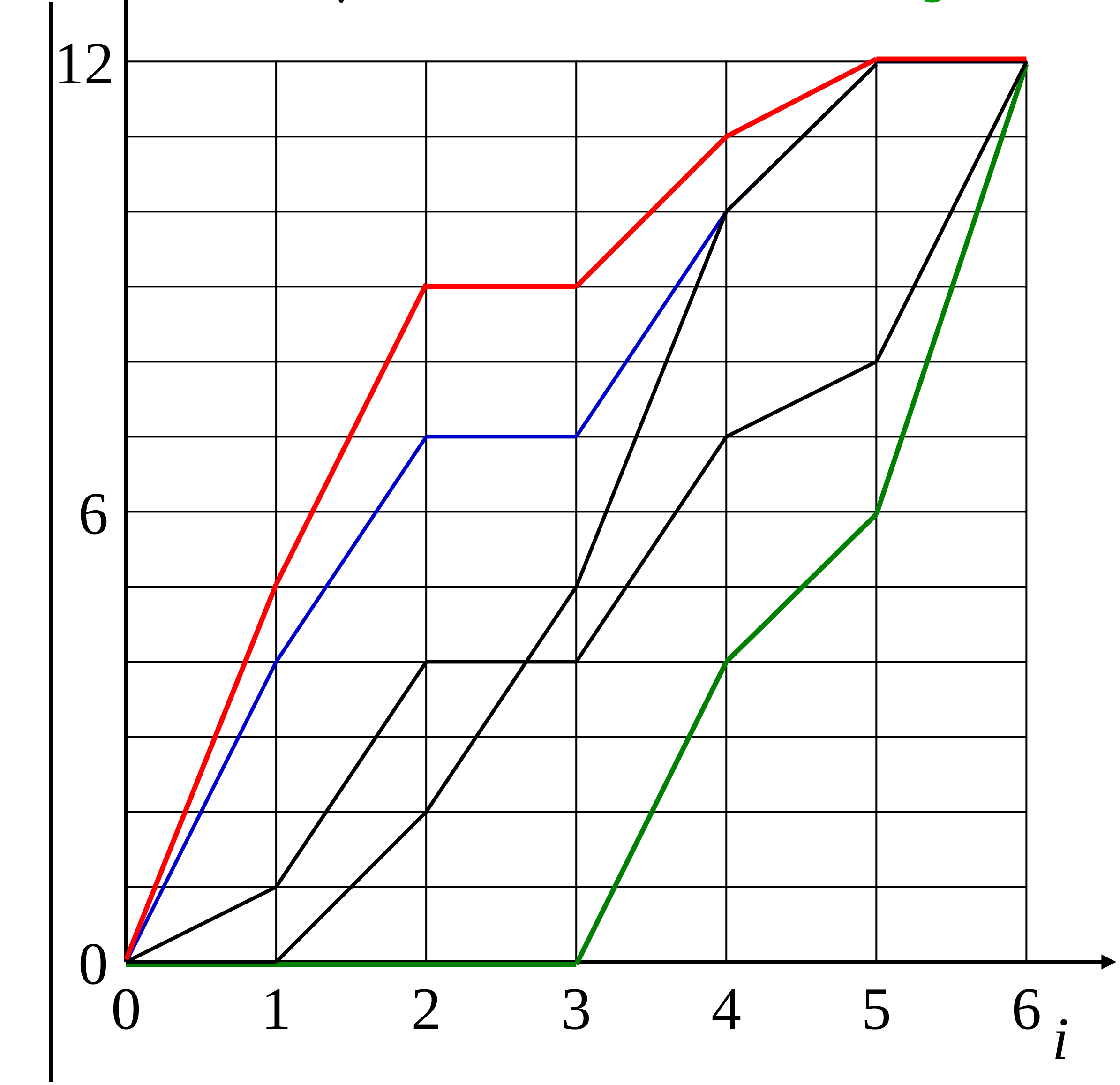


fig. of coalescence

any transition preserve part. order

=> every state is between red & green

3	3	2	1	2	1	12
2	1	1	6	3	5	18
5	4	3	7	5	6	30

3	3	1	2	2	1	12
2	1	2	5	3	5	18
5	4	3	7	5	6	30

1	2	2	3	2	2	12
4	2	1	4	3	4	18
5	4	3	7	5	6	30

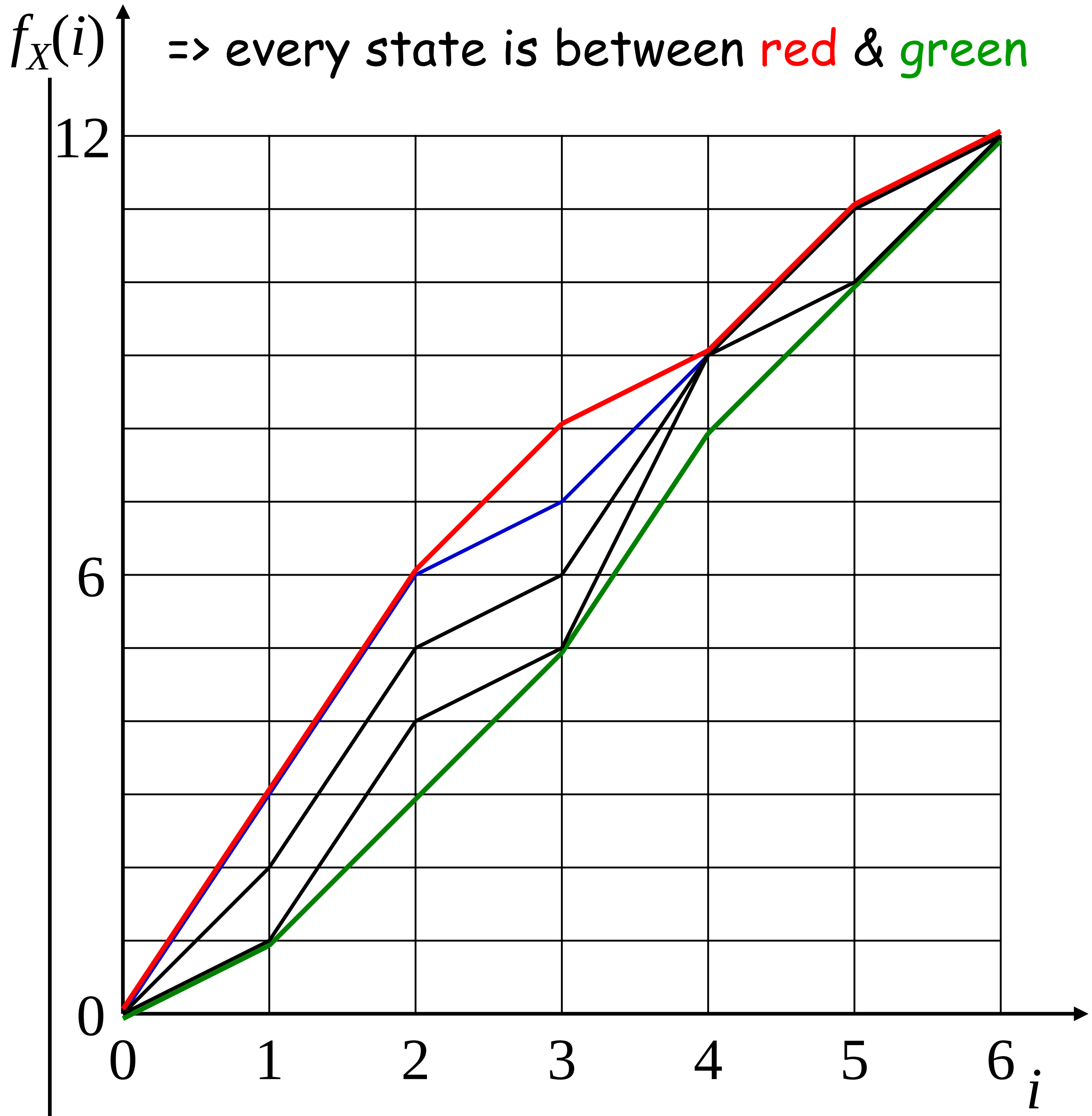
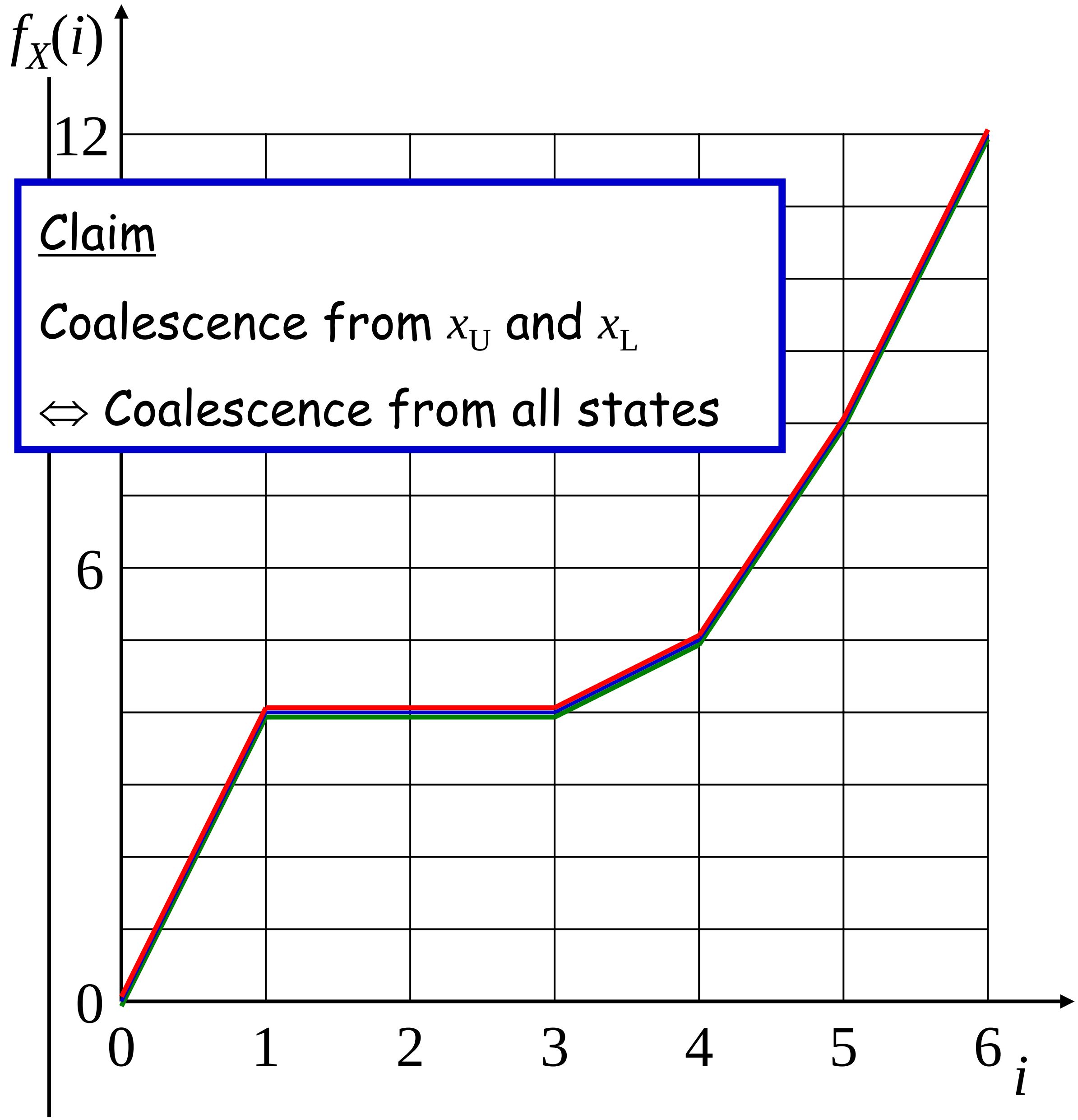


fig. of coalescence

4	0	0	1	3	4	12
1	4	3	6	2	2	18
5	4	3	7	5	6	30

4	0	0	1	3	4	12
1	4	3	6	2	2	18
5	4	3	7	5	6	30

4	0	0	1	3	4	12
1	4	3	6	2	2	18
5	4	3	7	5	6	30



Claim

Coalescence from x_U and x_L

$\Leftrightarrow$ Coalescence from all states

-1: Concluding Remark

Expected running time of our perfect sampler

Condition

column sum vector ***s*** satisfies

$$s_1 \geq s_2 \geq \cdots \geq s_n$$

Claim

Expected coalescence time = O($n^3 \ln (n \cdot K)$).

n: # of rows

K: sum total in table

- Omit the proof

Single server model

- Expected running time of CFTP algorithm =
  (T_* : coalescence time)
- Coalescence time of monotone CFTP (Propp and Wilson '96)
  - E [T_*] $\leq 2\tau(1 + \ln D)$
    (τ : mixing rate, D : the distance of max. and min.)
- Mixing rate of our chain
  - $\tau = n^2(n-1)\ln(n \cdot K)$, $D \leq (n \cdot K)$ (by path coupling)
    - ✧ **special distance**

Discussion

monotone Markov chain

- ✓ Ising model
  - restoration of mono-chromatic pictures
  - Potts model Hard-core model
- ✓ tiling
- ✓ 2-rowed contingency tables
- ✓ queueing network

Another perfect sampling algorithm

- ✓ Rooted spanning tree

Improvement of memory space

- ➢ Read once algorithm [Wilson 2000]

Reference

- O. Haeggstroem,
  “Finite Markov Chains and Algorithmic Application,“
  London Mathematical Society, Student Texts, 52,
  Cambridge University Press, 2002.

- 来嶋秀治, 松井知己, “完璧にサンプリングしよう！”
  オペレーションズ・リサーチ, 50 (2005),
  第一話「遥かなる過去から」, 169--174 (no. 3),
  第二話「天と地の狭間で」, 264--269 (no. 4),
  第三話「終りある未来」, 329--334 (no. 5).

http://www.simplex.t.u-tokyo.ac.jp/~kijima/

（来嶋のHPの“資料”からダウンロード可能）

Future works

- To apply the monotone CFTP to sampling-hard objects

  $\Rightarrow$ Design a Markov chain

  $\Rightarrow$ Design a perfect sampler

  $\Rightarrow$ Estimate the coalescence time

- $m \times n$ contingency tables
- New algorithm for Perfect Sampling

0. The end

— all of your views coalesce.

Thank you for the attention.