Oct. 14, 2011

確率と計算

来嶋 秀治 (九州大学)

本講演のねらい

「乱択アルゴリズムにおいて、乱数に真に求める性質は何か？」

本講演のねらい

「乱択アルゴリズムにおいて、乱数に真に求める性質は何か？」

1. 乱択の威力: ストリーム中の頻出アイテム検知
2. 高度な乱択技法: 組合せ的対象のランダム生成
3. 脱乱択化: ランダムウォークの脱乱択化

[ISAAC 2011]

1. ストリーム中の頻出アイテム検知

緒方 正虎，山内 由紀子，来嶋 秀治，山下 雅史

九州大学

ストリームデータ中の頻出アイテム検知

問題: 頻出アイテム検知

Given: $\theta \in (0,1)$ $\boldsymbol{x} = (x_1,\ldots,x_N) \in \Sigma^N$ (順々に)

Find: all $s \in \Sigma$ s.t. $f(s) \geq \theta \cdot N$ ∘

但し $f(s)$ は s が $\boldsymbol{x}$ 中で出現した回数

事前には、
N (or log Nの近似値)も
わからない。

例1. POSデータ

Σ ={お茶, 菓子, 乾電池, …, 弁当}

$\boldsymbol{x}$ = お茶, 菓子, 乾電池, お茶, 弁当, ジュース, お茶, ….

例2. IPアドレス

$\Sigma \subseteq$ {0.0.0.0,…,255.255.255.255}

$\boldsymbol{x}$ = 123.45.67.89, 111.11.1.1., 123.45,67,89, 122.122.12.12…

頻出アイテム検知の領域複雑度

定理 [Karp, Shenker, Papadimitriou '03]

頻出アイテム検知を厳密に行うには,

$\Omega(|\Sigma| \log (N/|\Sigma|))$ bits が必要.

($N \gg |\Sigma| \gg 1/\theta$ とする)

定理 [Karp, Shenker, Papadimitriou '03]

頻出アイテム検知に対する

$O((1/\theta) \log N)$ bits の偽陽性(近似)アルゴリズムが存在.

($N \gg |\Sigma| \gg 1/\theta$ とする)

決定的

$o(\log N)$ bitsアルゴリズム？

- e.g. $O(\log \log N)$ bits?

単純化: o(log N) bitsで要素数を数えられるか？

問題: 要素数え上げ

Given: $x = (a,\ldots,a) \in \Sigma^N$ (順々に)

Find: N

事前には、
N (or log Nの近似値)も
わからない。

アルゴリズム: 数え上げ

0. **Set** n:= 0.
1. **Read** an input. **If** no more input, **goto** 3.
2. n++, **Goto** 1.
3. **Output** n (as N = n).

O(log N) bits

o(log N) bits近似アルゴリズム？

- e.g. O(log log N) bits?

単純化: o(log N) bitsで要素数を数えられるか？

問題: 要素数え上げ

Given: $x = (a,\ldots,a) \in \Sigma^N$ (順々に)

Find: N

事前には、
N (or log Nの近似値)も
わからない。

Remark

- NはO(log N) bitsで表現可能.
  - ✓ N = 1,351,127,649,213
- Nの近似はO(log log N) bitsで表現可能
  - ✓ $N \approx 1.351 \times 10^{12}$

つまり、

指数部(=log N)がo(log N) bits領域で近似計算できるか？ということ。

表現はO(log log N) bitsで可能

単純化: o(log N) bitsで要素数を数えられるか？

問題: 要素数え上げ

Given: $x = (a,\ldots,a) \in \Sigma^N$ (順々に)

Find: N

事前には、
N (or log Nの近似値)も
わからない。

アルゴリズム: 数え上げ

0. **Set** n:=0.
1. **Read** an input. **If** no more input, **goto** 3.
2. n++, **Goto** 1.
3. **Output** n (as N = n).

O(log N) bits

不可能？（決定的アルゴリズムでは）

確率的数え上げ

問題: 要素数え上げ

Given: $x = (a,\ldots,a) \in \Sigma^N$ (順々に)

Find: N

事前には、
N (or log Nの近似値)も
わからない。

アルゴリズム: 確率的数え上げ

0. **Set** k:=0.
1. **Read** an input. **If** no more input, **goto** 3.
2. k++, w.p. $1/2^k$. **Goto** 1.
3. **Output** k (as $N \approx 2^k$).

⇒ ポイント

PTM では w.p. $1/2^k$ を
$O(\log K)$ bitsで実現可能！

$O(\log \log N)$ bits

Thm. [Morris '78, Flajolet '85]

$E[2^{k+1}] \approx N+1$

確率的数え上げ(改良型)

問題: 要素数え上げ

Given: $x = (a,\ldots,a) \in \Sigma^N$ (順々に)

Find: N

事前には、
N (or log Nの近似値)も
わからない。

アルゴリズム: 確率的数え上げ(改良型)

0. **Set** k:=0, l:=0.
1. **Read** an input. **If** no more input, **goto** 4.
2. l++, w.p. $1/2^k$.
3. If l==2^b, k++, and **set** l := l' w.p. $\binom{2^b}{l'} \cdot 2^{-1}$. **Goto** 1.
4. **Output** l and k (as $N \approx l * 2^k$).

O(log log N) bits

Thm. [Ogata, Yamauchi, K., Yamashita '11]

E[$l*2^k$] ≈ N. P

"改良型" を使うと、頻出アイテム検知も可能. (詳細略)

ストリームデータ中の頻出アイテム検知

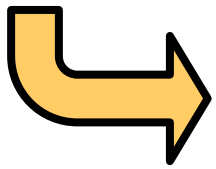
問題: 頻出アイテム検知

Given: $\theta \in (0,1)$ $\boldsymbol{x} = (x_1, \ldots, x_N) \in \Sigma^N$ (順々に)

Find: all $s \in \Sigma$ s.t. $f(s) \geq \theta \cdot N$ ∘

但し $f(s)$ は s が $\boldsymbol{x}$ 中で出現した回数

事前には、
N (or log Nの近似値)も
わからない。



o(log N) bitsアルゴリズム？

- e.g. O(log log N) bits?

ストリームデータ中の頻出アイテム検知

問題: 頻出アイテム検知

Given: $\theta \in (0,1)$ $\boldsymbol{x} = (x_1,\ldots,x_N) \in \Sigma^N$ (順々に)

Find: all $s \in \Sigma$ s.t. $f(s) \geq \theta \cdot N$

但し $f(s)$ は s が $\boldsymbol{x}$ 中で出現した回数

事前には、
N (or log Nの近似値)も
わからない。

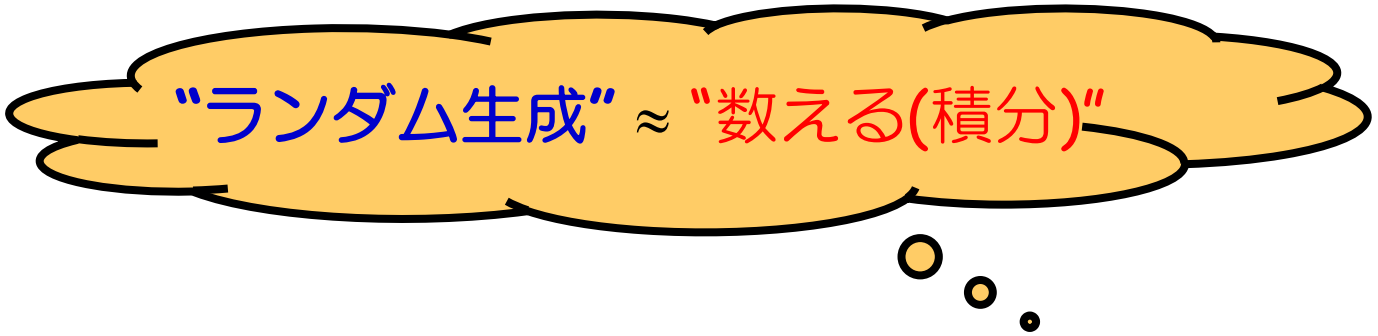
Thm. [Ogata, Yamauchi, K., Yamashita '11]

O(log log N) bits 領域の乱択近似アルゴリズムが存在.

詳細略

o(log N) bitsアルゴリズム？

- e.g. O(log log N) bits?



2. 組合せ的対象のランダム生成

マルコフ連鎖モンテカルロ法

(MCMC: Markov chain Monte Carlo)

joint with 松井知己 (中央大学)

例: 2行分割表のランダム生成

2元分割表

- ✓ 各セルには非負整数が入る
- ✓ （与えられた）周辺和を満たす

						12
						18
5	4	3	7	5	6	30

問題

Given: 周辺和

出力: 分割表の一様ランダム生成

例: 2行分割表のランダム生成

2元分割表

- ✓ 各セルには非負整数が入る
- ✓ （与えられた）周辺和を満たす

						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

状態A

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

状態B

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

状態C

問題

Given: 周辺和

出力: 分割表の一様ランダム生成

例: 2行分割表のランダム生成

2元分割表

- ✓ 各セルには非負整数が入る
- ✓ （与えられた）周辺和を満たす

						12
						18
5	4	3	7	5	6	30

与えられた周辺和を満たす2行分割表の**個数を求める問題**

⇒#P完全 (NP困難) ['97 Dyer, Kannan, & Mount]

問題

Given: 周辺和

出力: 分割表の一様ランダム生成

マルコフ連鎖を用いた

サンプリング法

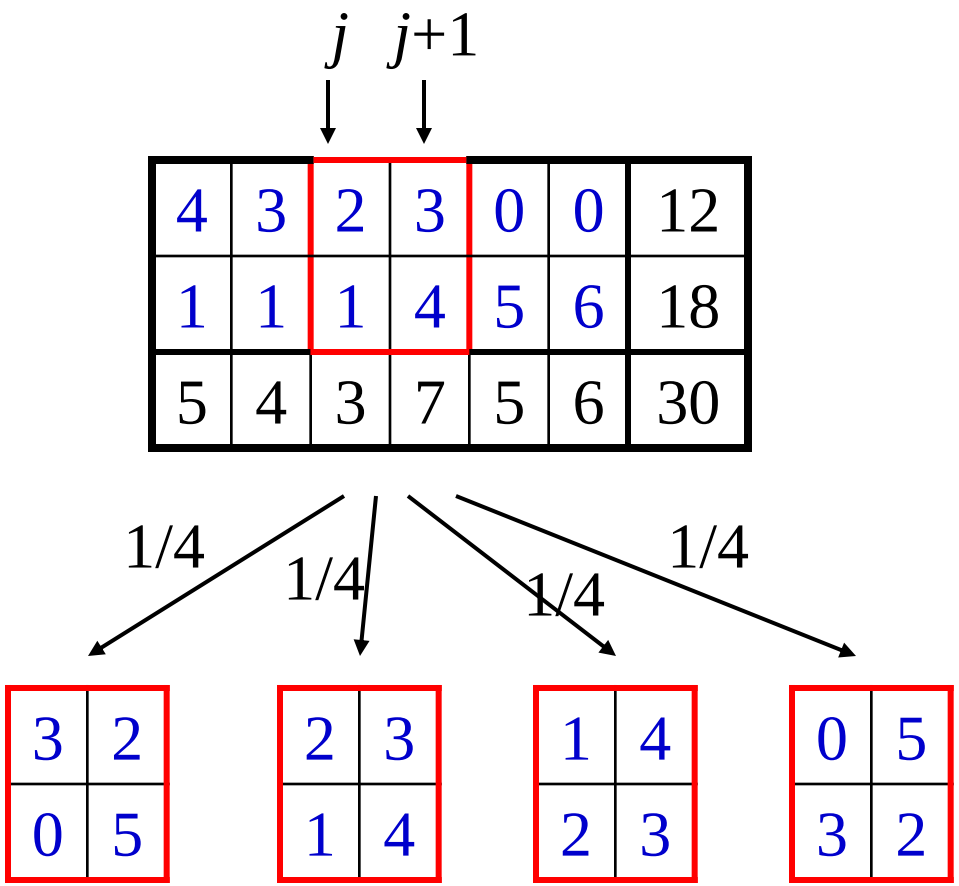
例: 2行分割表に対するマルコフ連鎖 [K & Matsui '06]

- j 列目（とj+1列目）を $1/(n-1)$の確率で選ぶ。
- j 列目と j+1列目に対して 推移可能な状態に 等確率で推移する。

2	3	5
1	4	5
3	7	10

+

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



提案するマルコフ連鎖の特徴

定理

提案したマルコフ連鎖 の定常分布は一様分布である.

略証: $\forall(X, Y),\ P(X, Y) > 0 \Rightarrow P(Y, X) > 0$ かつ $P(X, Y) = P(Y, X)$

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

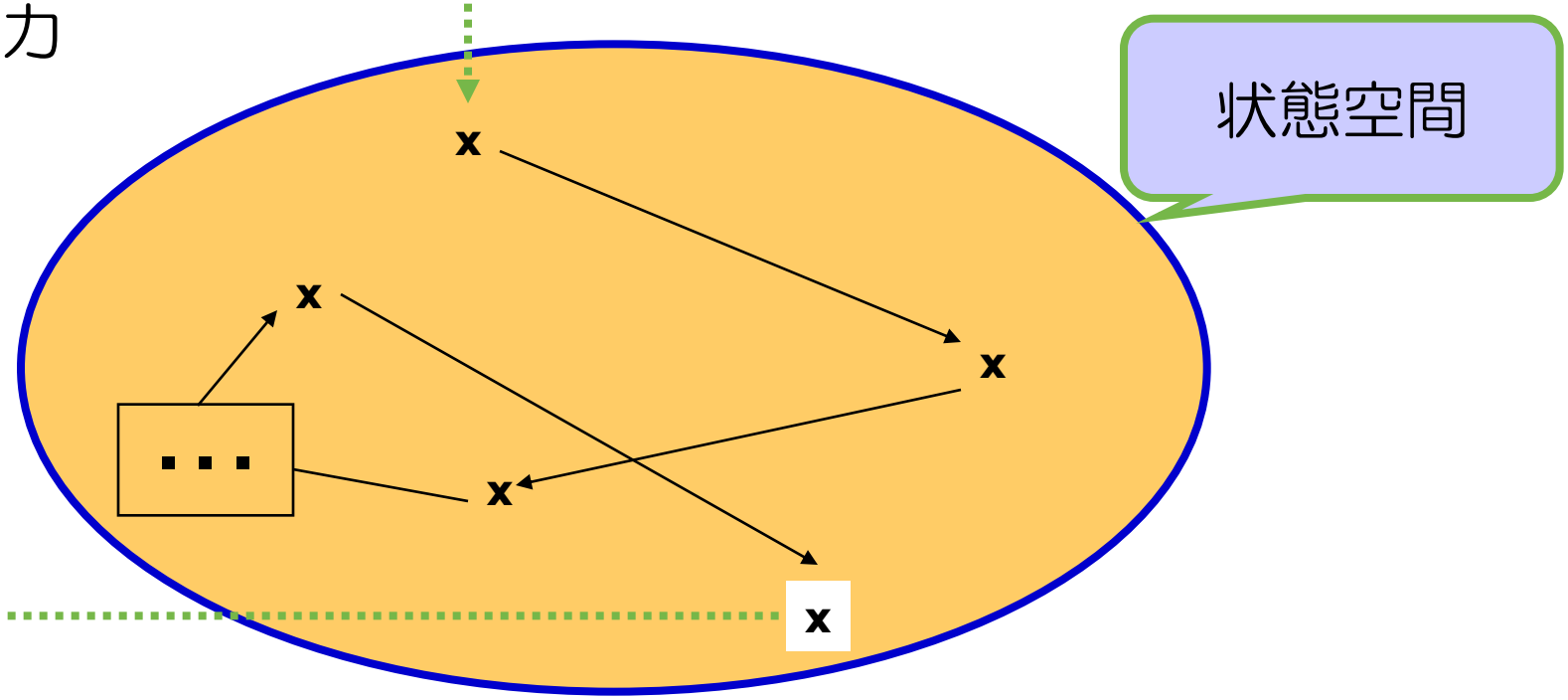
(detailed balance equation $\pi(X)\, P(X, Y) = \pi(Y)\, P(Y, X)$)

マルコフ連鎖の極限分布

任意の初期状態

推移を繰り返す

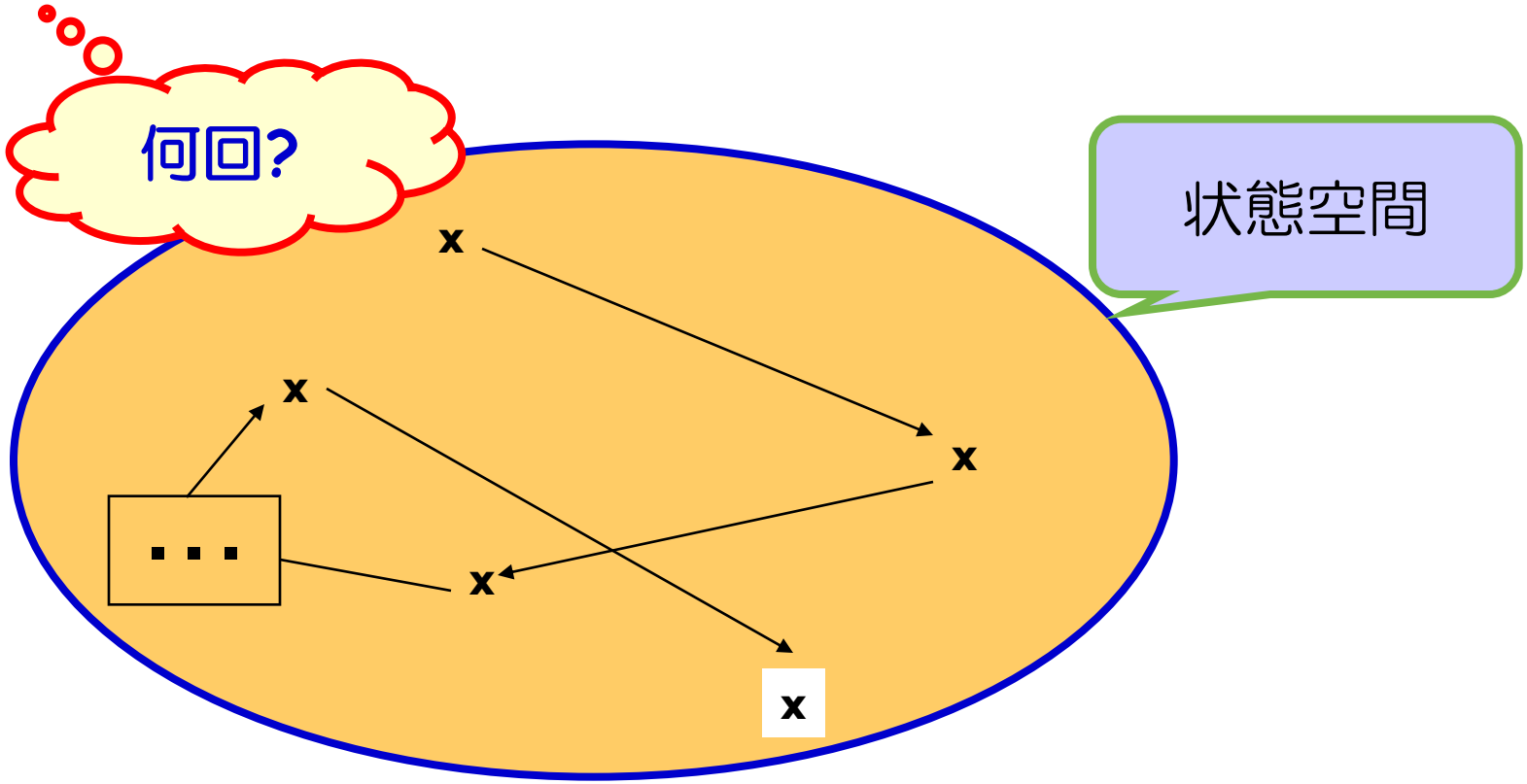
状態を出力



漸近的に定常分布

MCMC法のアイデア

1. 所望の分布を定常分布にもつマルコフ連鎖を設計する.
2. 十分な回数推移させて，定常分布からサンプリングする.



⇒ (漸近的に)所望の分布に従うサンプリングを実現

問題点は何か？

「何回推移させれば十分か？」

近似サンプリング法

- ✓ 収束スピードの算定
  - **mixing time**, total variation distance

完璧サンプリング法

- ✓ マルコフ連鎖の推移シミュレーションを工夫
- ✓ 無限回の推移の結果を出力 (⇒ 定常分布に厳密に従う)
  - **Coupling from the past** [Propp & Wilson 1996]

。。。

単調マルコフ連鎖の設計で効率化

MCMC完璧サンプリングの設計例

単調なマルコフ連鎖

- ✓ Isingモデル
- ✓ Pottsモデル（クラッターモデル）

[Propp &Wilson 96]

- ✓ tiling [Wilson 01]
- ✓ 2行分割表 [K & Matsui 06]
- ✓ 離散化Dirichlet分布 [Matsui & K 07]
- ✓ 待ち行列ネットワーク [K & Matsui 08]
- ✓ Q-Ising モデル [Yamamoto, K & Matsui 08]

それ以外のCFTP

- ✓ 根付き全張木 [Propp &Wilson 98]

乱数ベクトルの記憶容量に関する問題

- ➢ Wilson の read onceアルゴリズム [Wilson 00]
- ➢ Fillのinterruptibleアルゴリズム [Fill 98]

参考文献

- 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).
- 玉木久夫, 乱択アルゴリズム, 共立出版, 2008, 3000円

- ✓ 「ORアーカイブ」でダウンロード可能
  http://www.orsj.or.jp/~archive/menu/03_50.html
- ✓ 来嶋のHP
  http://www.kurims.kyoto-u.ac.jp/~kijima/
  - ➢ “資料”からダウンロード可能
  - ➢ 2行分割表の完璧サンプリング法のデモ (JAVAアプレット)

サンプリングアルゴリズム --最近の動向と今後の課題

解決済み (多項式性)

✓ 高次元凸体上の一様分布/対数凹分布
[Dyer, Frieze, & Kannan 91], [Lovasz & Vempala 07]

✓ パーマネント/2部グラフの完全マッチング
[Jerrum, Sinclair, & Vigoda 04], [Stefankovic, Vempala, & Vigoda 09]

✓Ising モデル/白黒画像 [Jerrum & Sinclair 93], [Propp &Wilson 96]

未解決問題 (多項式性):
- ✓ **2元分割表** (u.a.r.)
- ✓ **順序イデアル** (u.a.r.)
- ✓ **Tutte多項式**
  - ➢ **マトロイド基** (u.a.r.)
  - ➢ **Pottsモデル**

不可能性
一般の**対数劣モジュラ分布**に対して, [K 09+]
(RP ≠ NPの下) 多項式時間サンプリング法は存在しない.

「乱択アルゴリズムにおいて、乱数に真に求める性質は何か？」

脱乱択化 (derandomization)

3. ランダムウォークの脱乱択化

来嶋 秀治 (九州大学)

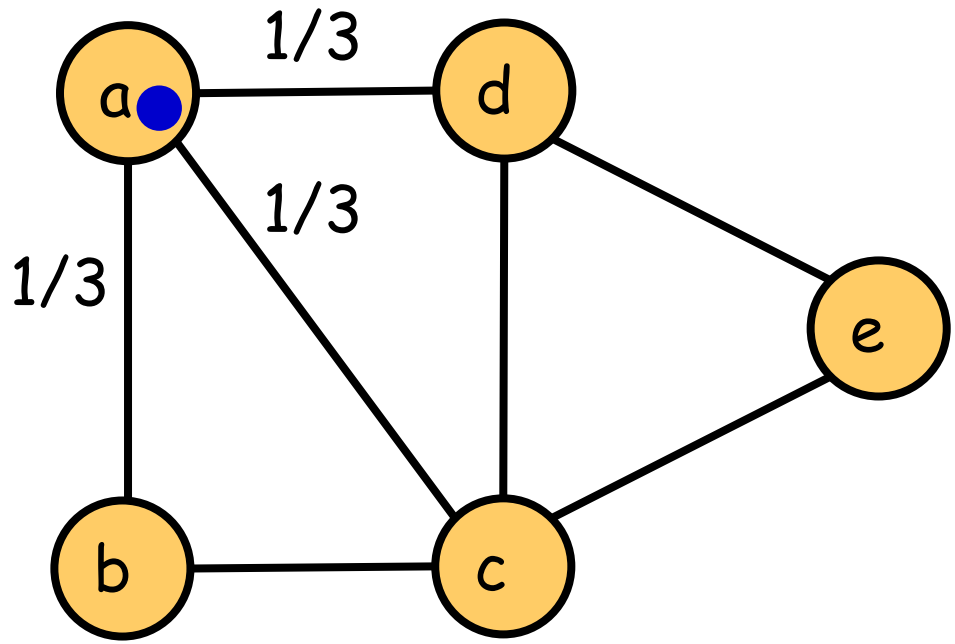
古賀 健太郎 (FANUC)

牧野 和久 (東京大学)

ランダムウォーク

トークンがグラフ上をランダムウォークする.

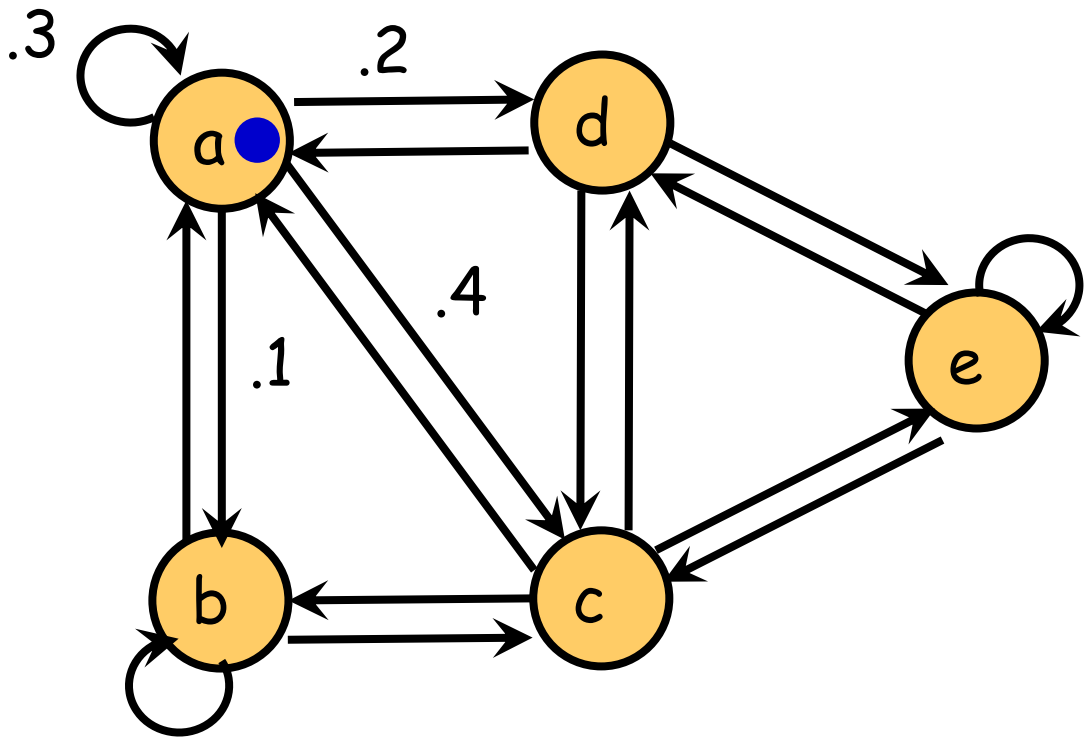
- ✓ π^0: 初期分布 (トークンは確率π^0_vで頂点vに居る)
- ✓ P: 推移確率行列 (確率P_{uv}で頂点u頂点vに移動)
- ✓ 時刻tの確率分布$\pi^t := \pi^0 P^t$ (頂点vに居る確率 $(\pi^t)_v$)



ランダムウォーク

トークンがグラフ上をランダムウォークする.

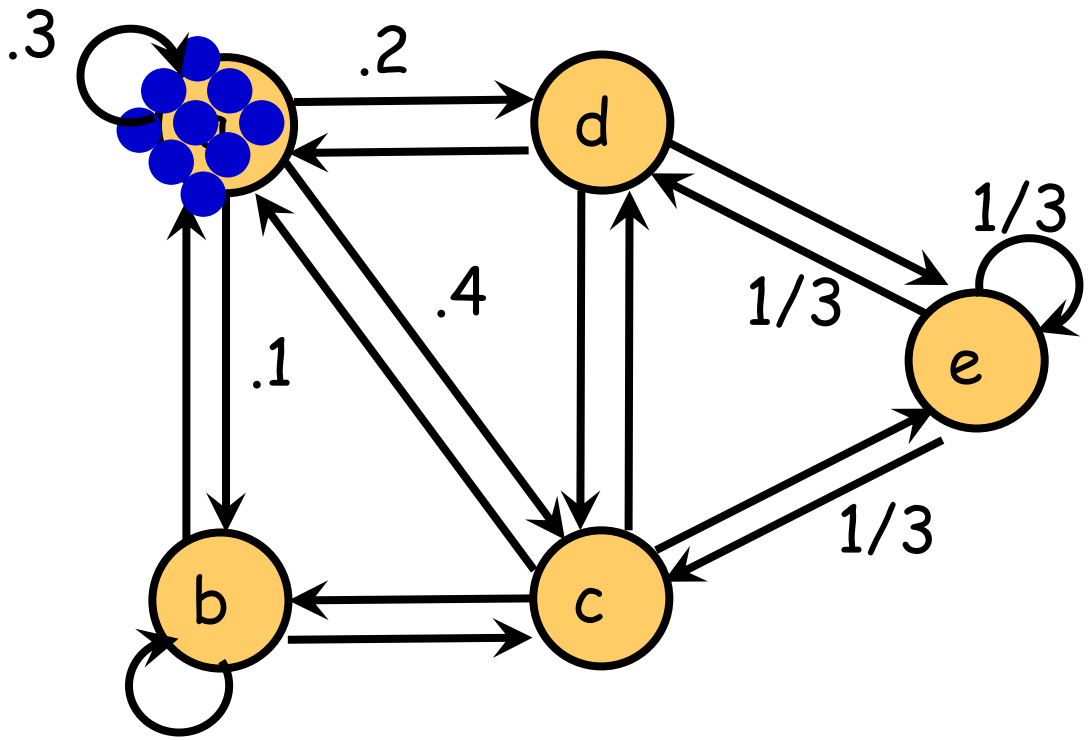
- ✓ π^0: 初期分布 (トークンは確率π^0_vで頂点vに居る)
- ✓ P: 推移確率行列 (確率P_{uv}で頂点u頂点vに移動)
- ✓ 時刻tの確率分布$\pi^t := \pi^0 P^t$ (頂点vに居る確率 $(\pi^t)_v$)



ランダムウォーク (複数トークンによる分布の近似)

N個のトークンがグラフ上を独立にランダムウォークする.

- ✓ μ^0: 初期配置 ($\pi^0 \approx \mu^0/N$)
- ✓ P: 推移確率行列 (確率P_{uv}で頂点u頂点vに移動)
- ✓ 時刻tの期待配置$\mu^t := \mu^0 P^t$ ($\pi^t \approx \mu^t/N$)



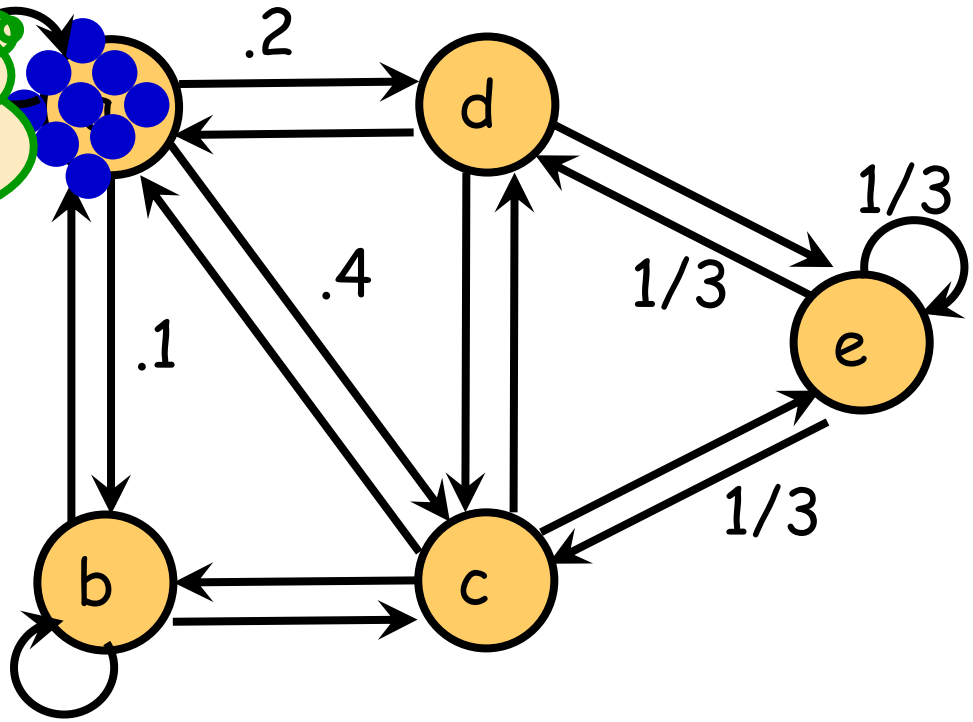
ランダムウォーク (複数トークンによる分布の近似)

N個のトークンがグラフ上を独立にランダムウォークする.

- ✓ μ^0: 初期配置 ($\pi^0 \approx \mu^0/N$)
- ✓ P: 推移確率行列 (確率P_{uv}で頂点u頂点vに移動)
- ✓ 時刻tの期待配置$\mu^t := \mu^0 P^t$ ($\pi^t \approx \mu^t/N$)

(Nが非常に大きければ,)
トークンを約2:4:1:3の割合で
隣接点にばらまいている.

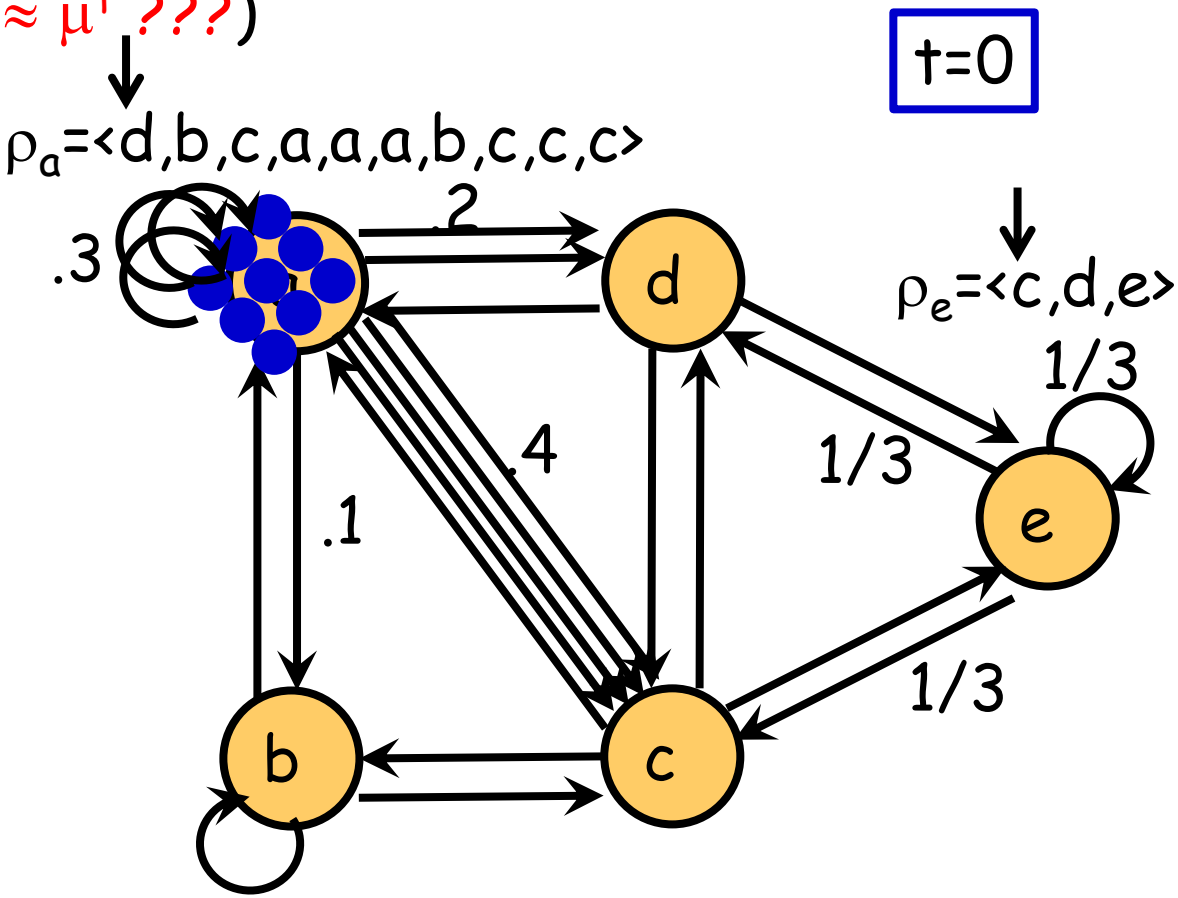
決定的過程でも大差ない?



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

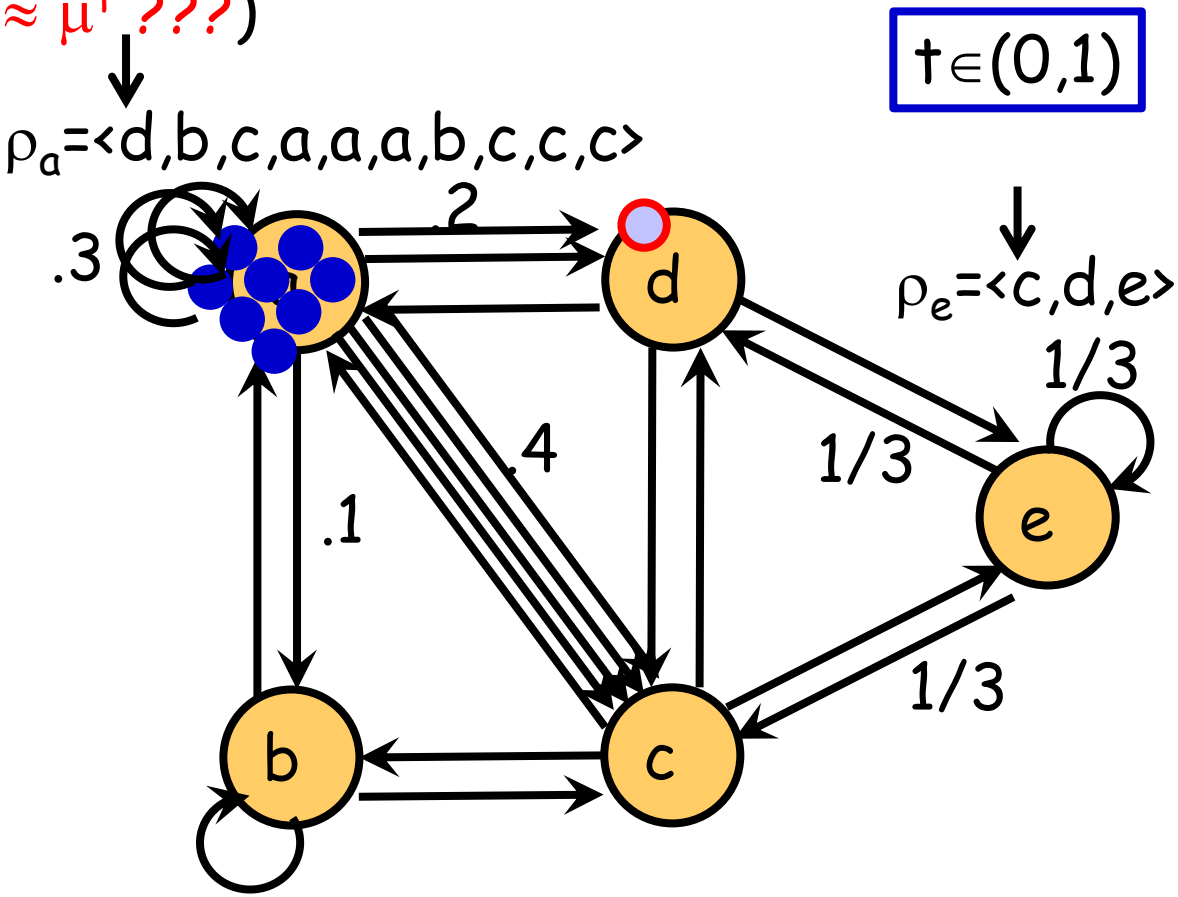
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

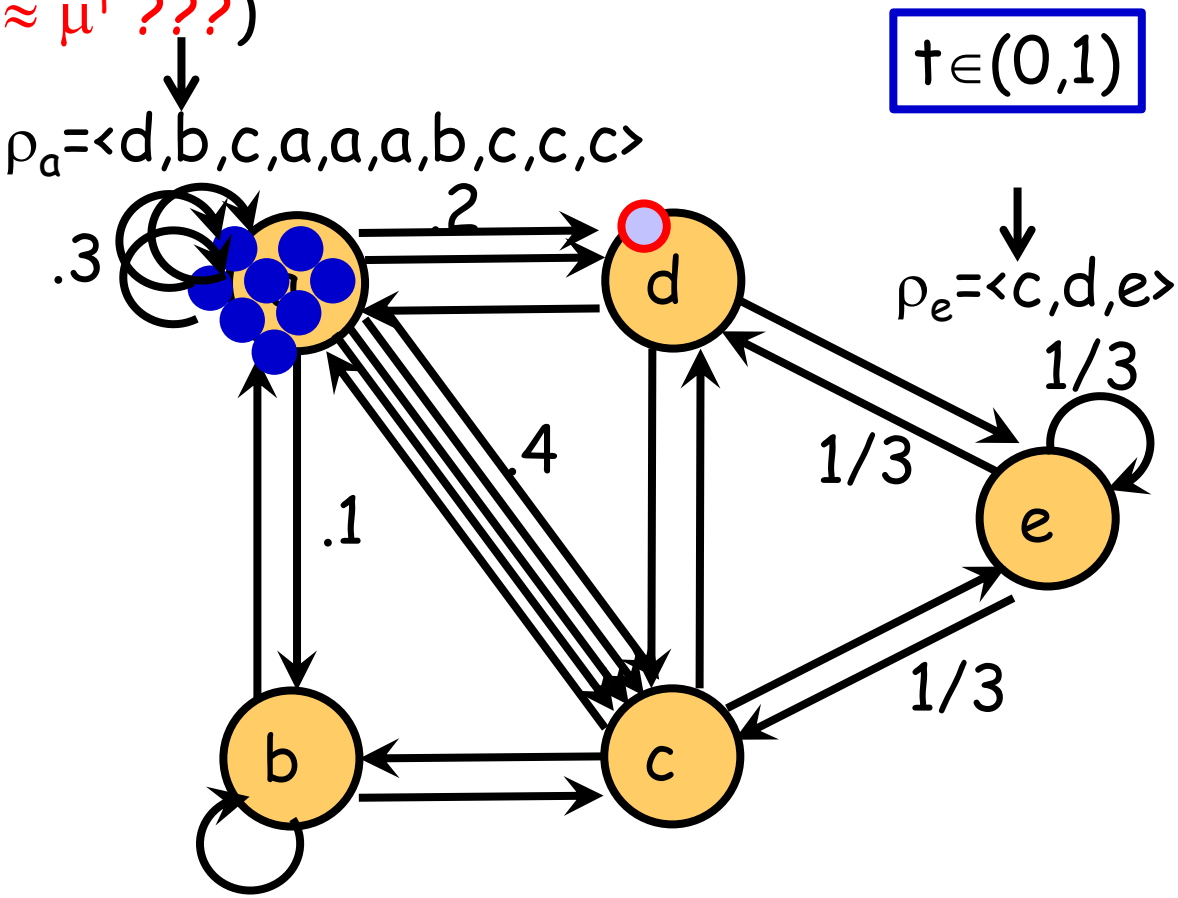
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

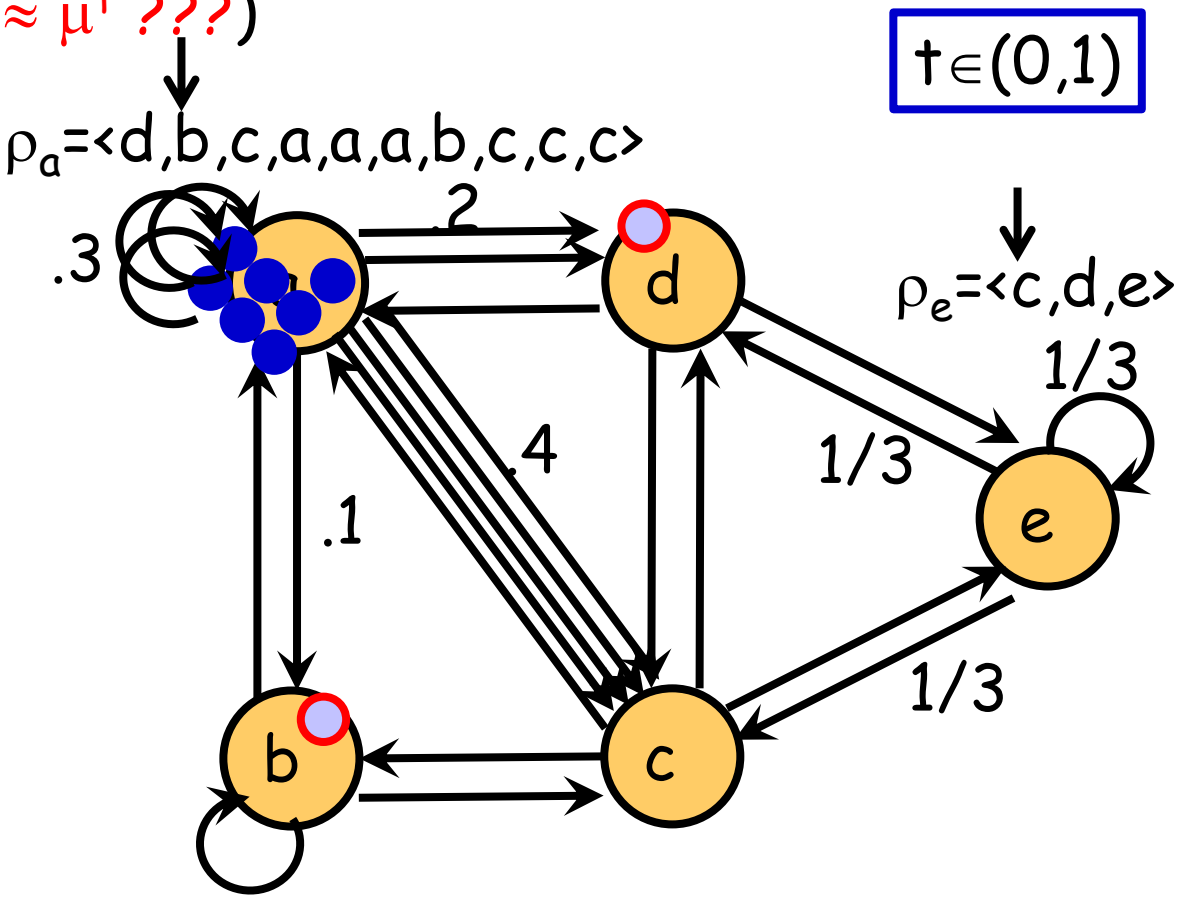
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

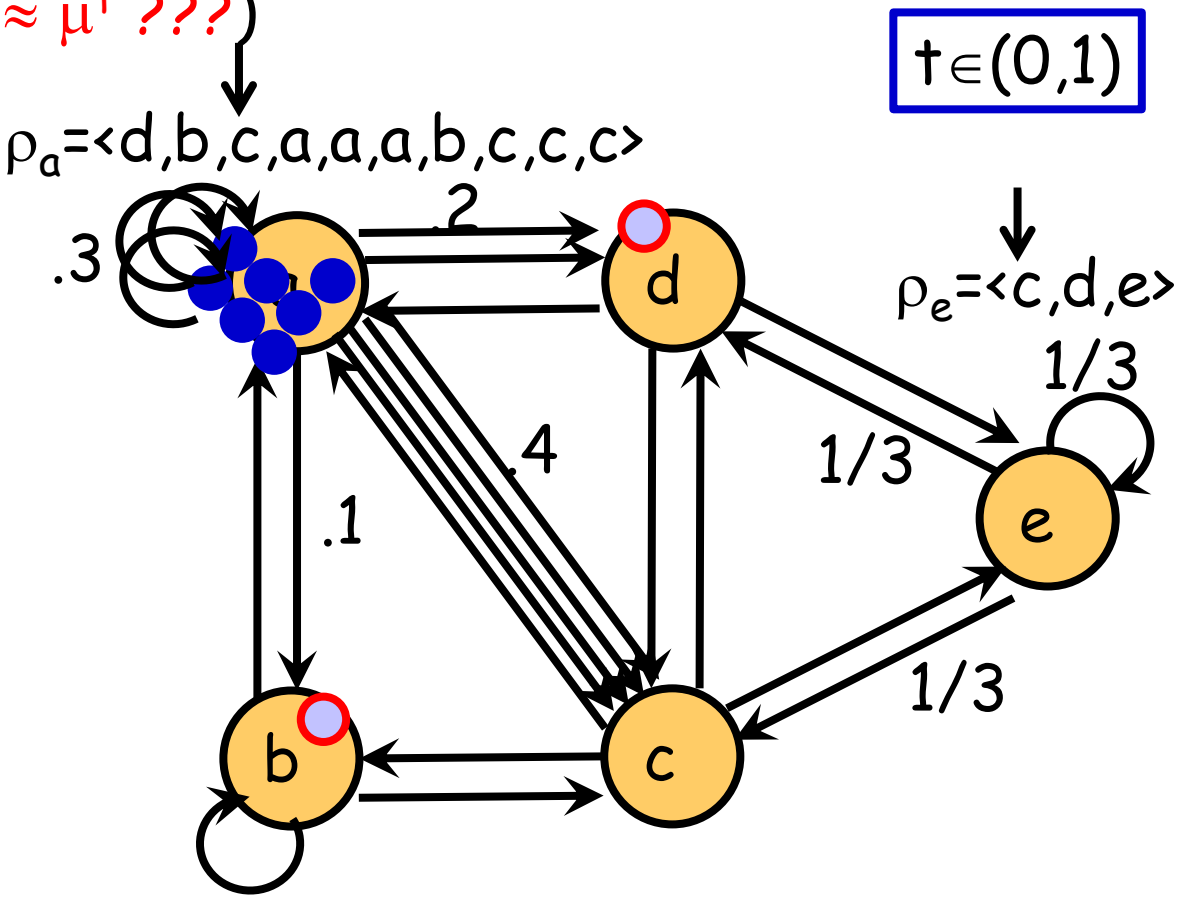
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

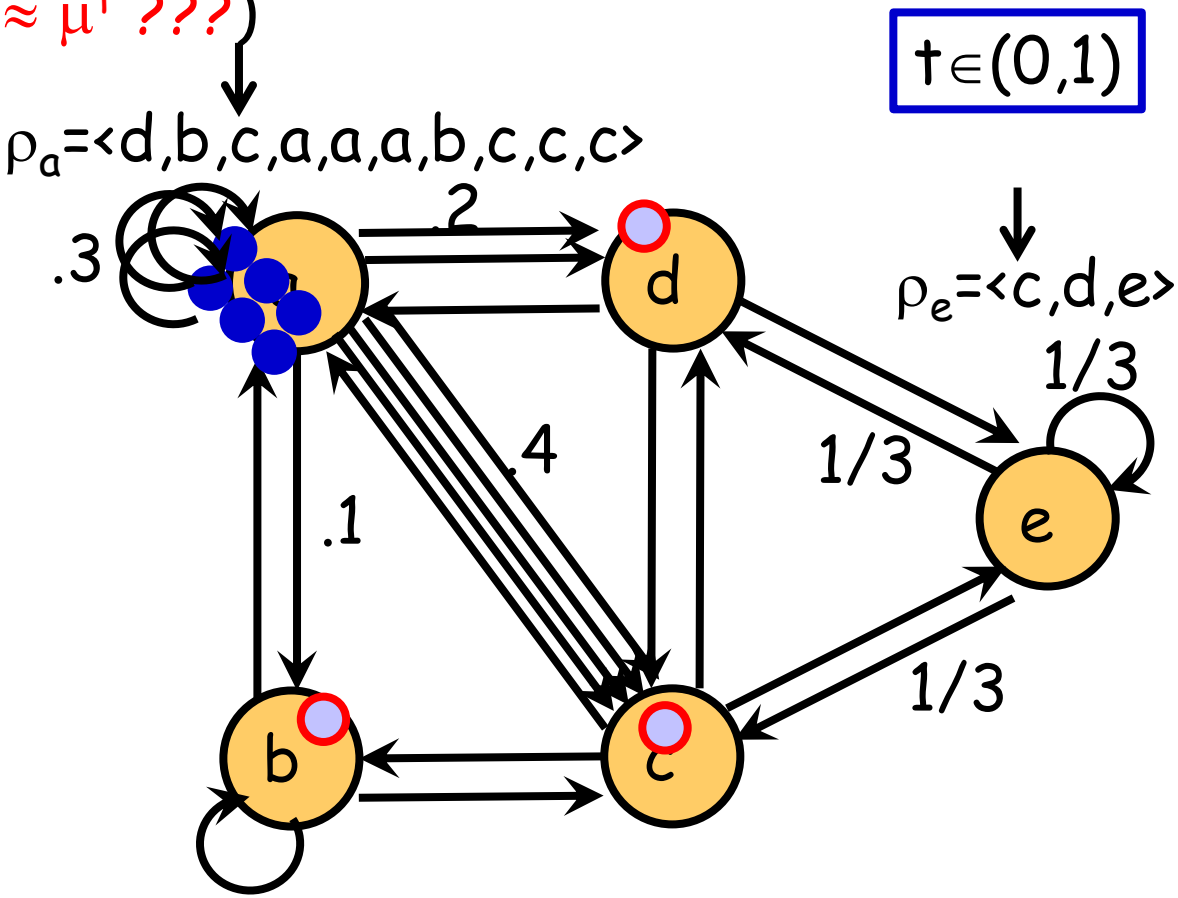
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

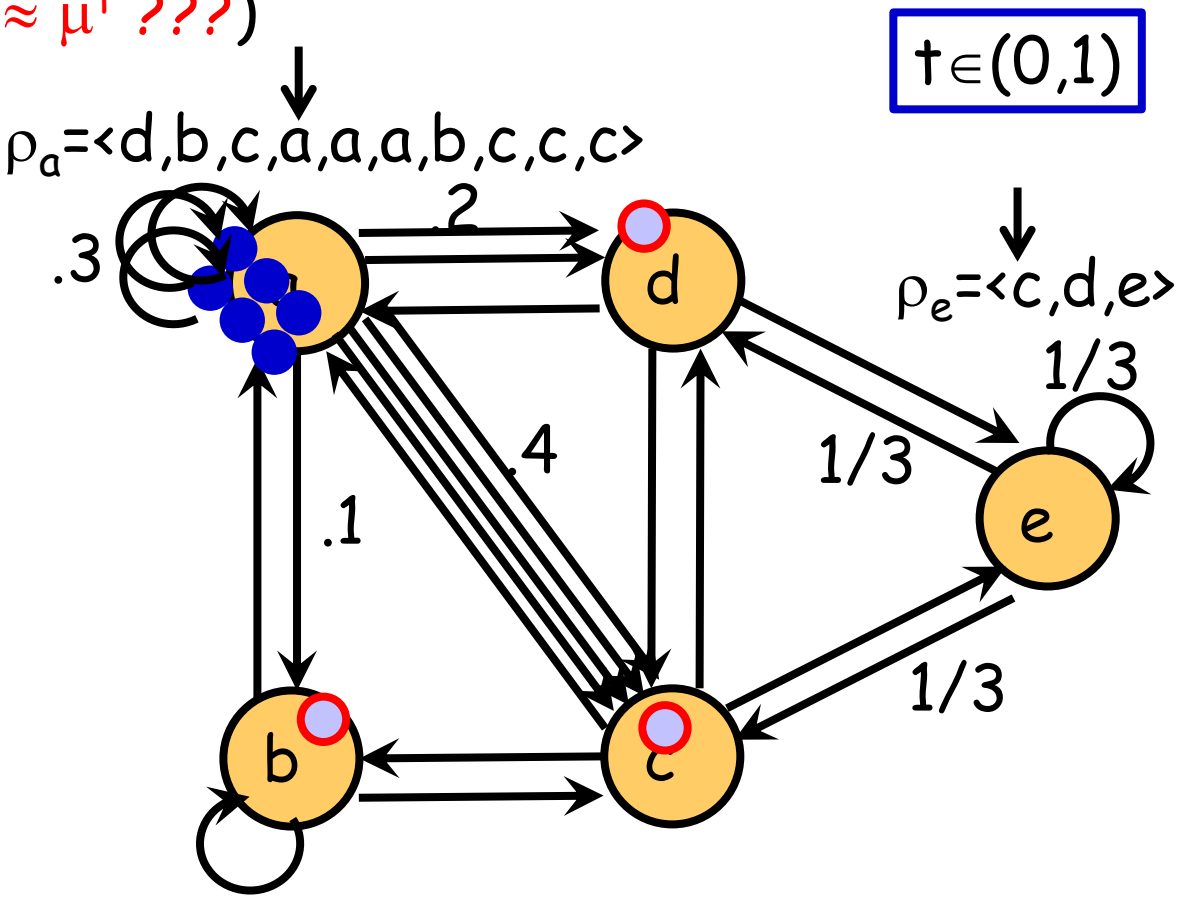
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

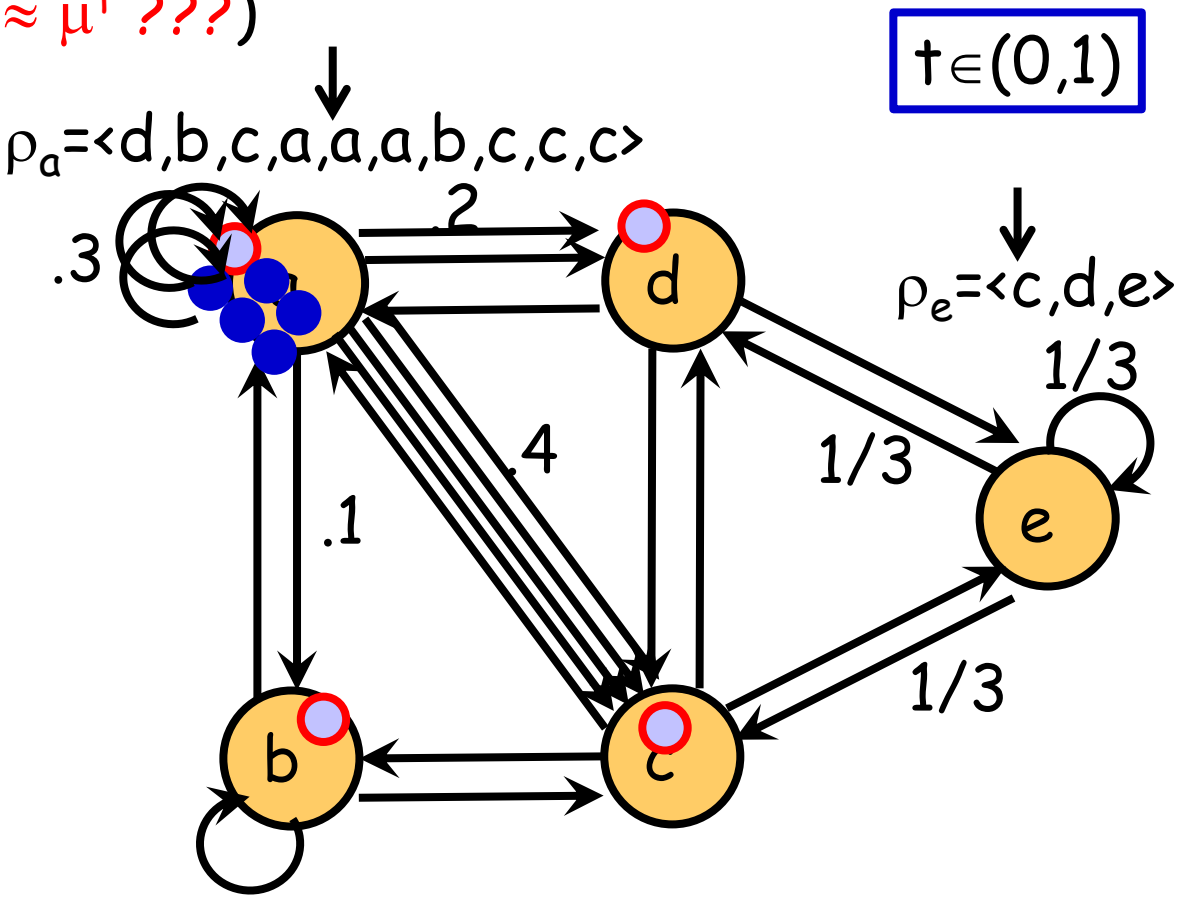
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

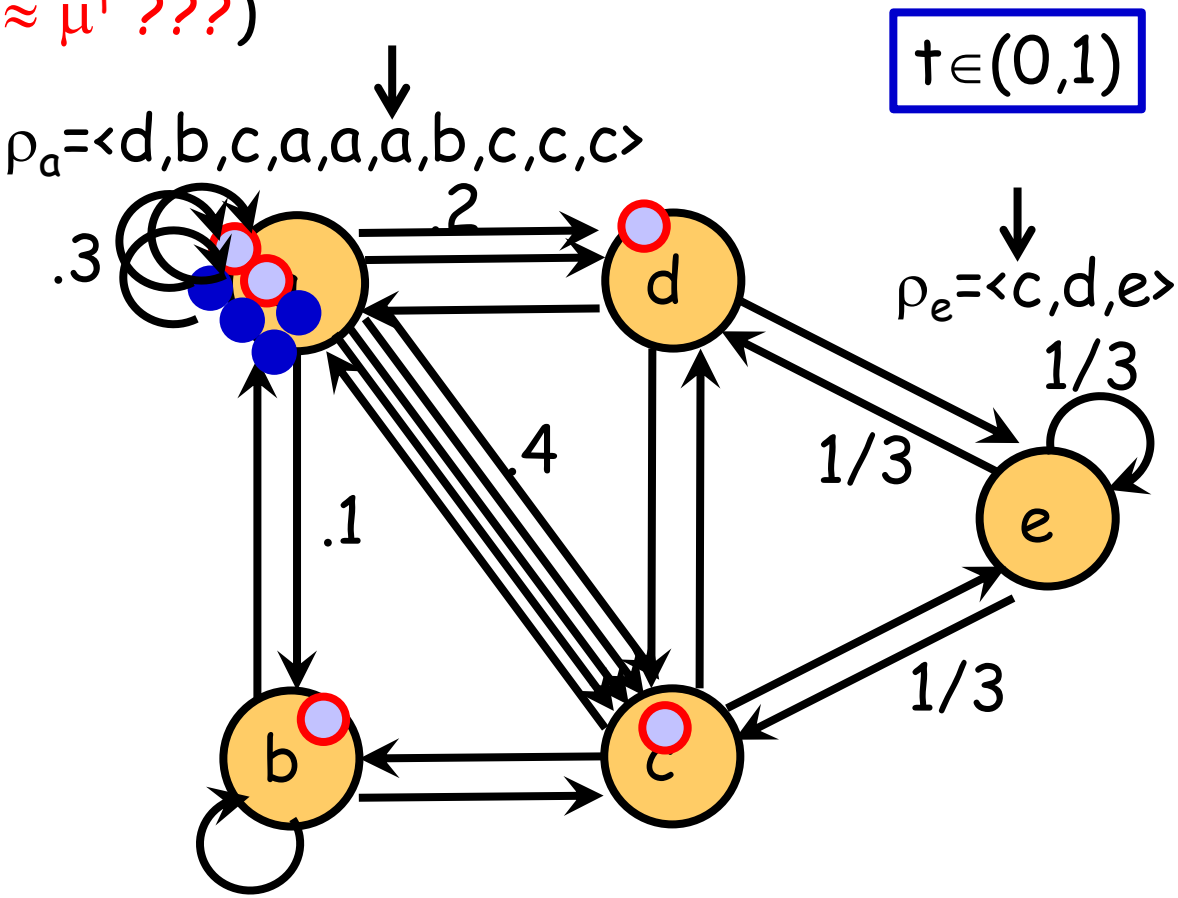
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

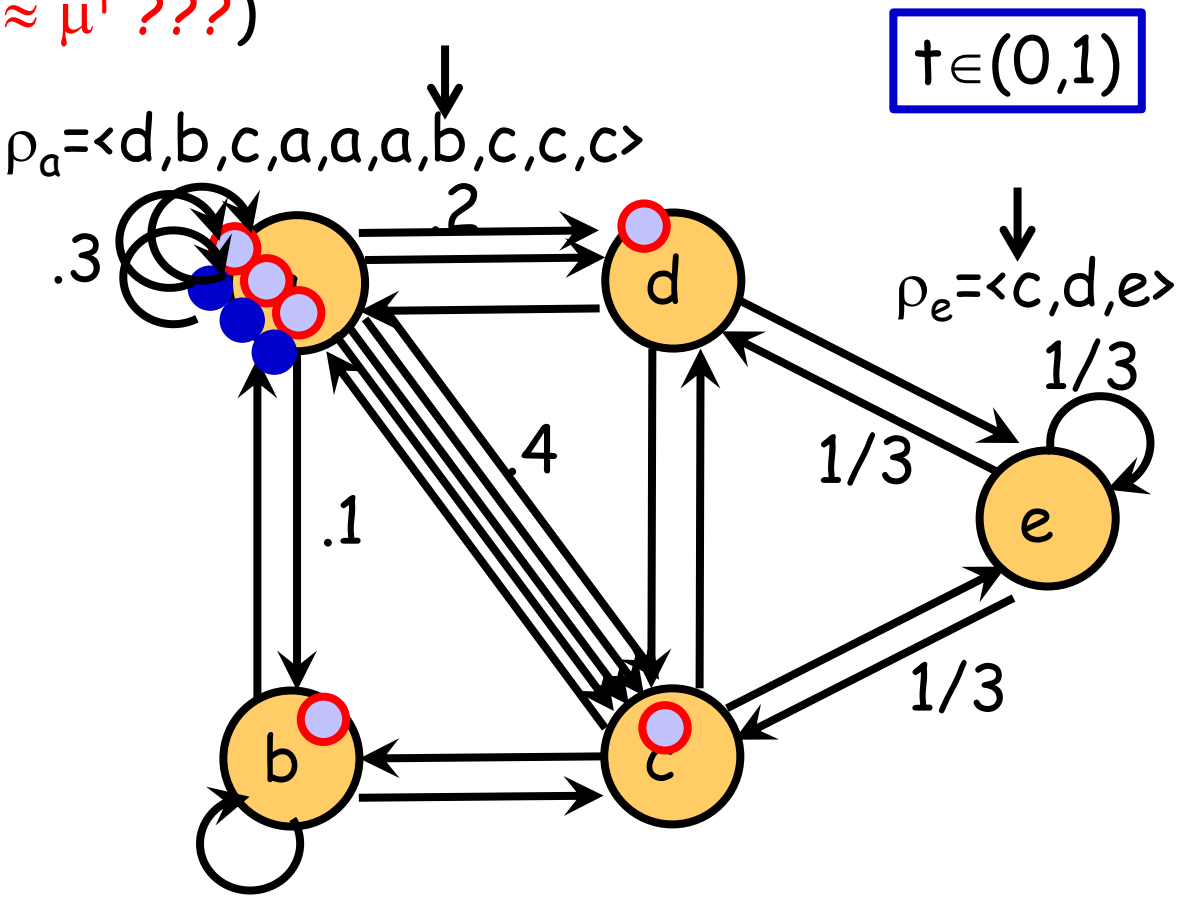
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

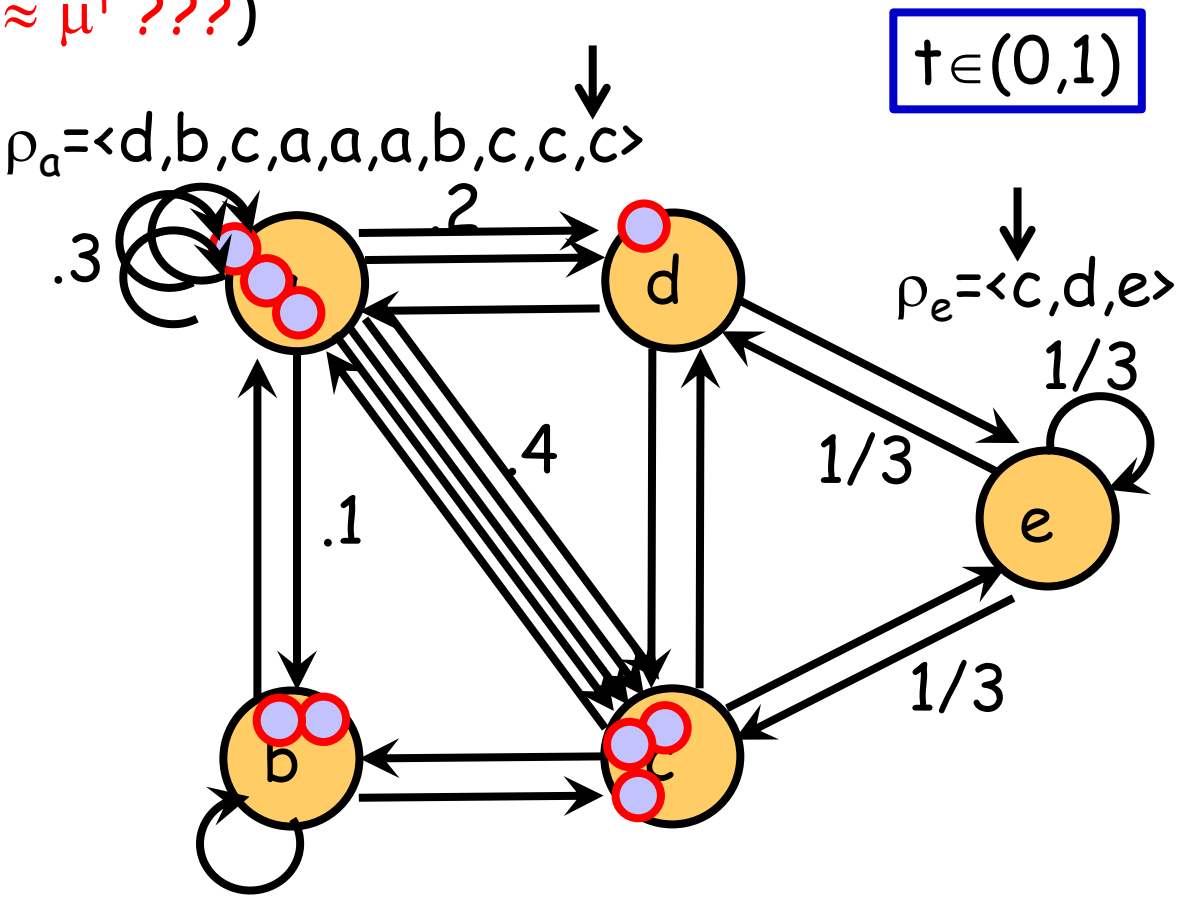
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

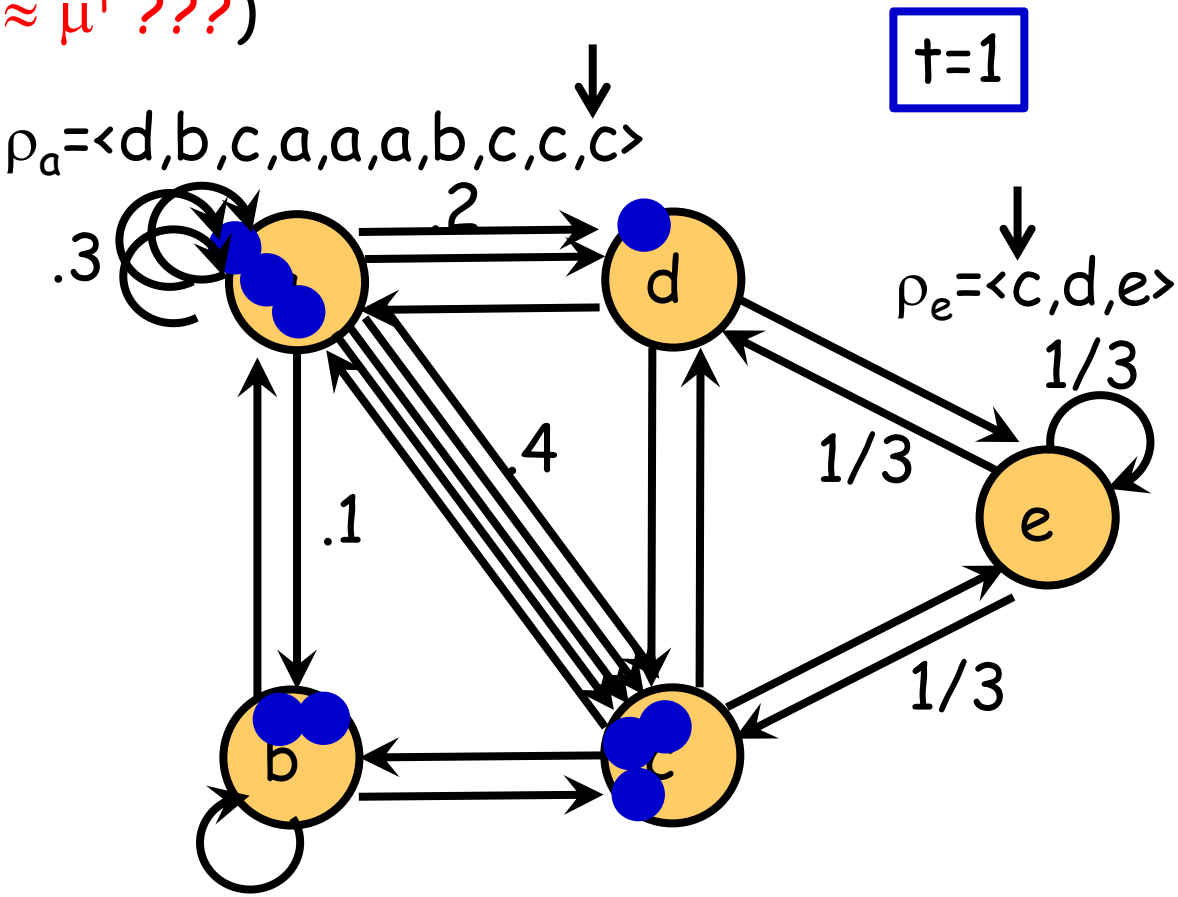
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

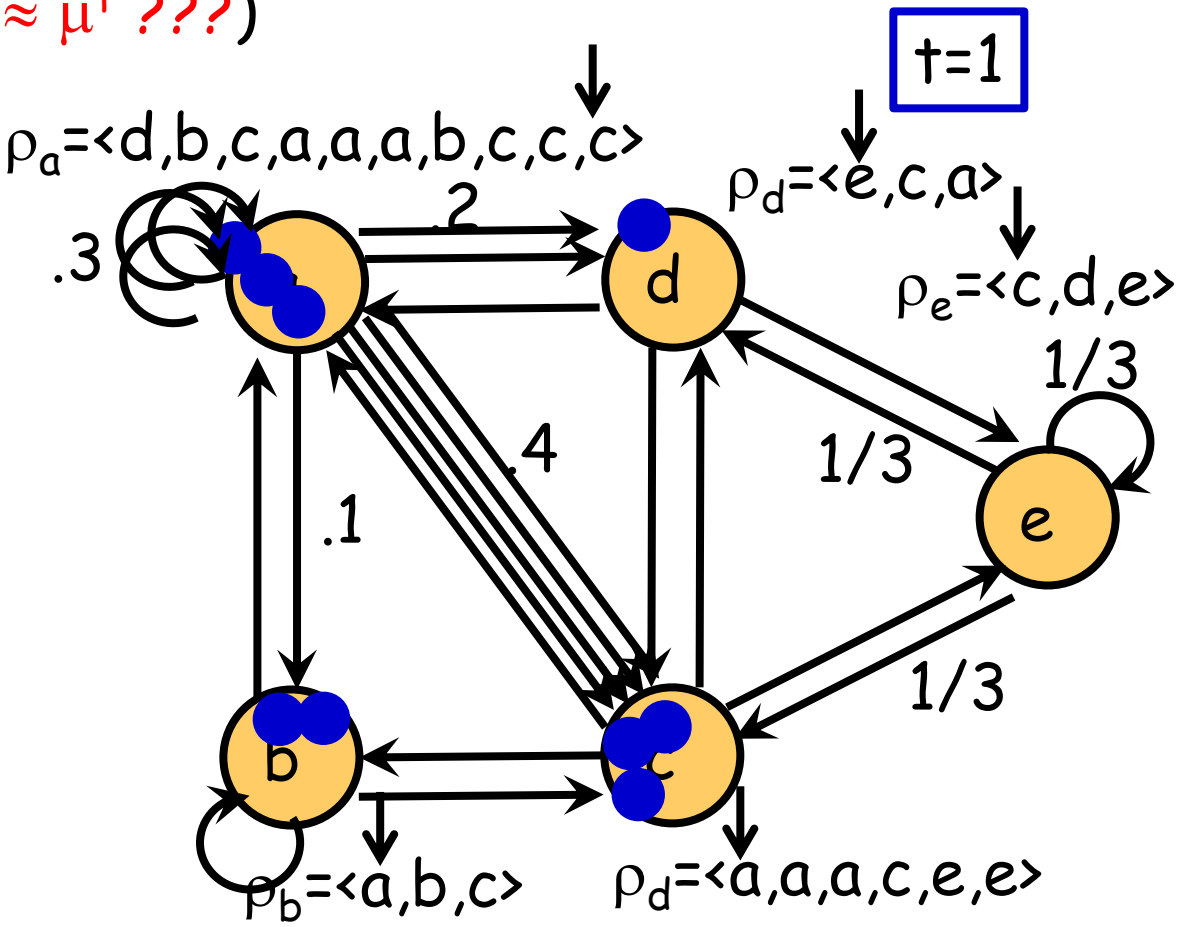
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

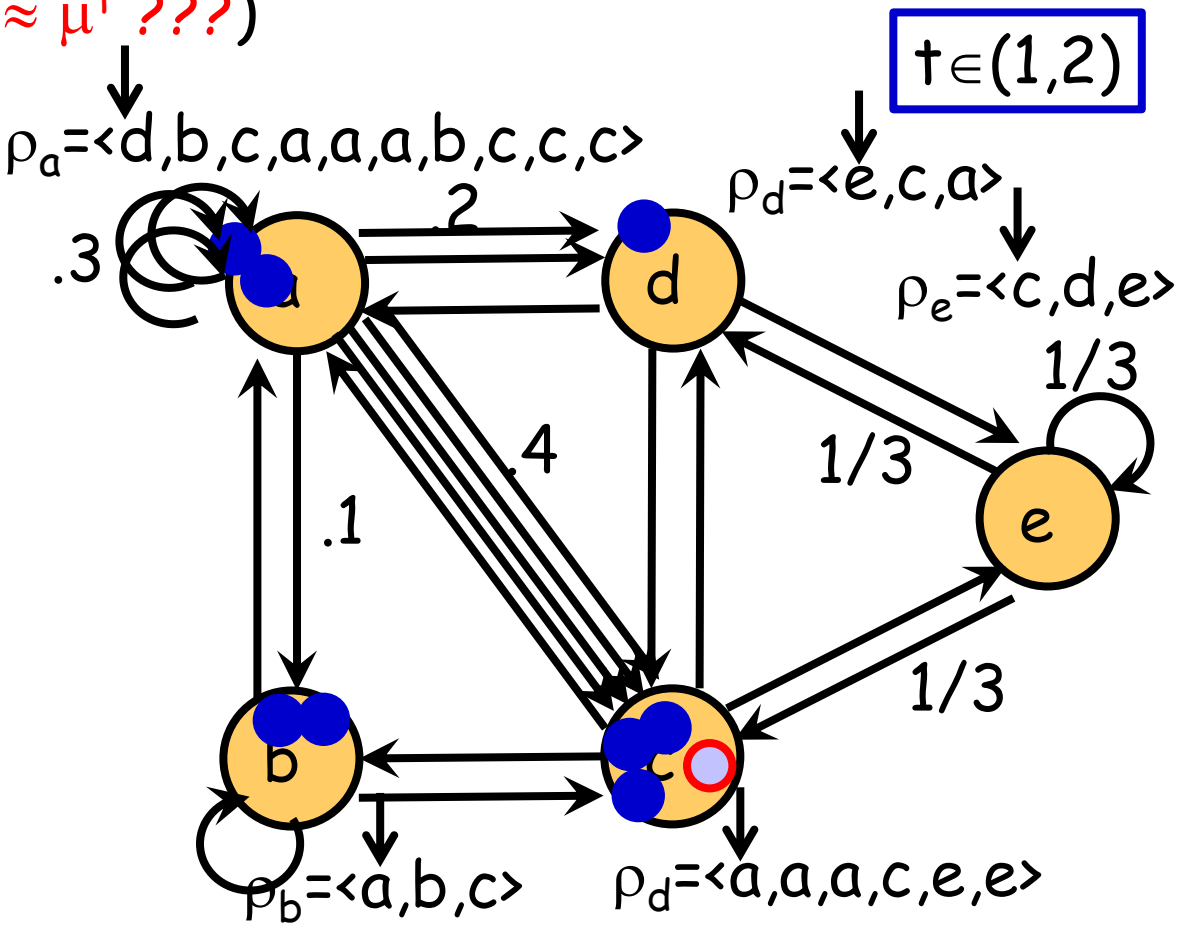
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

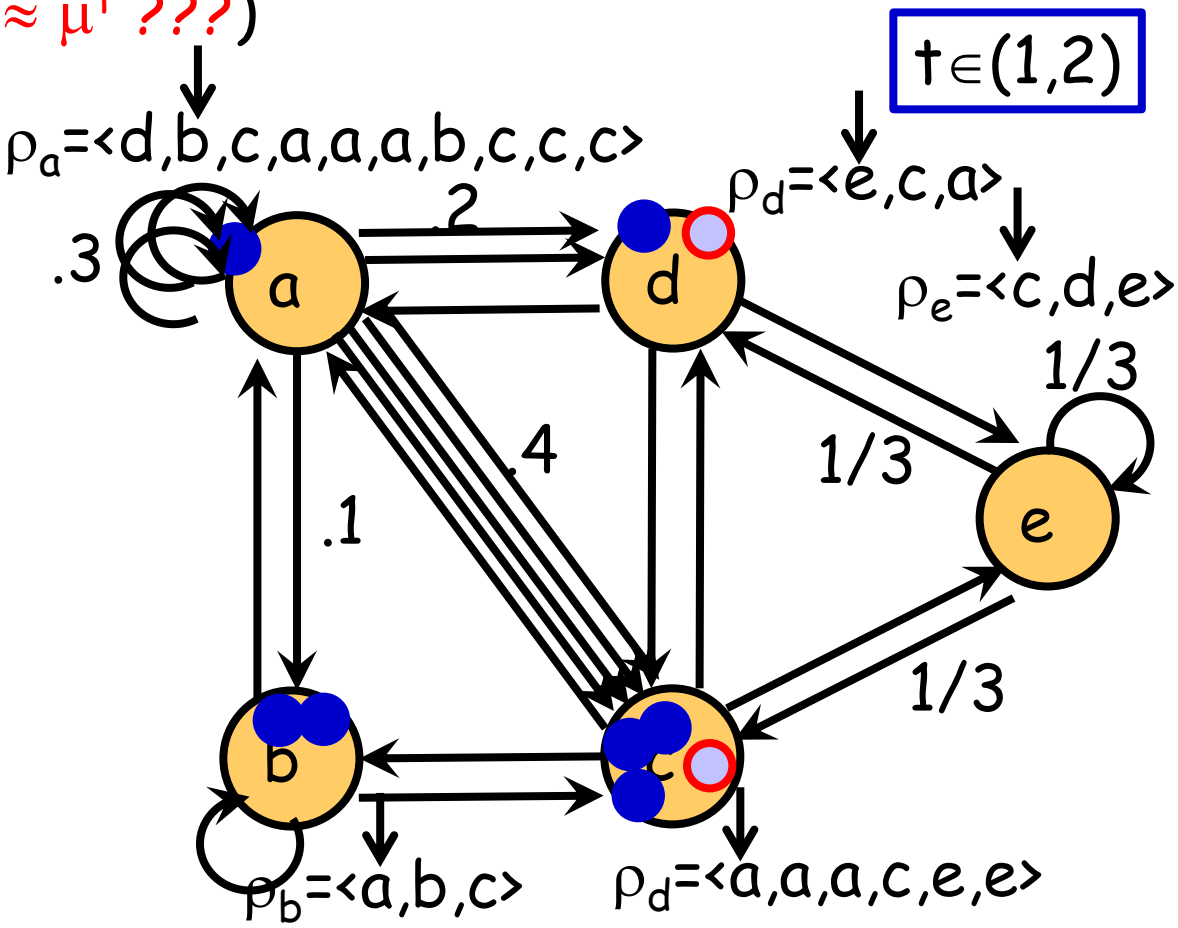
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

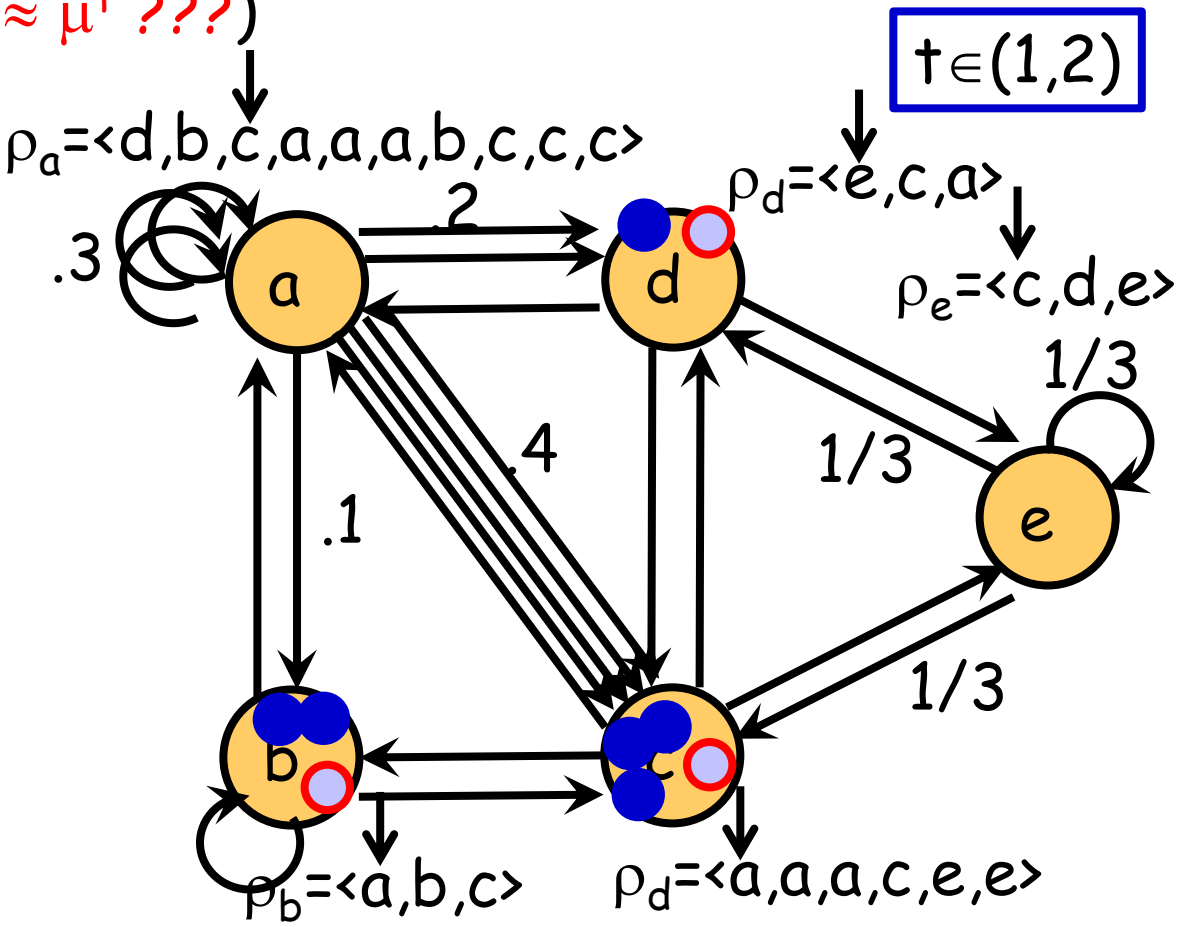
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

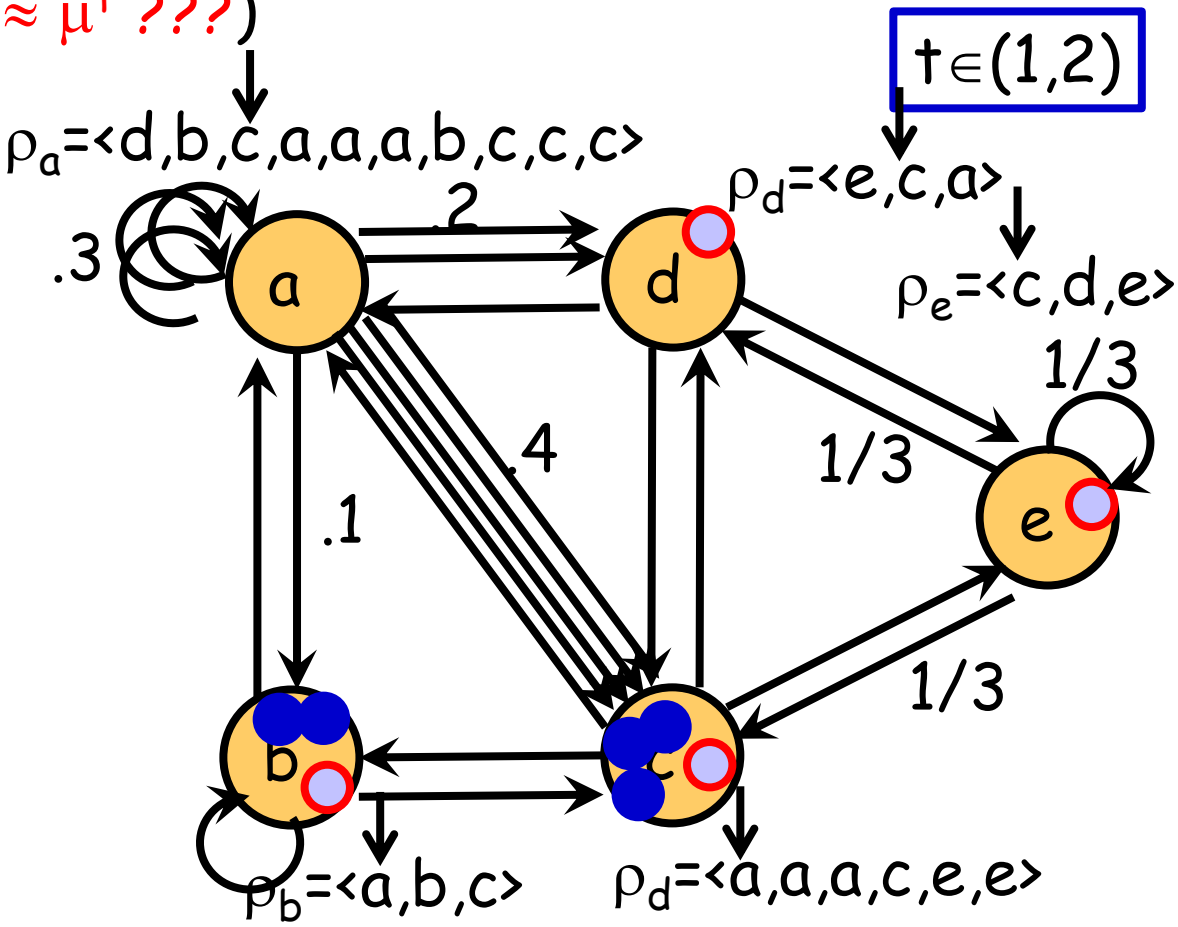
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

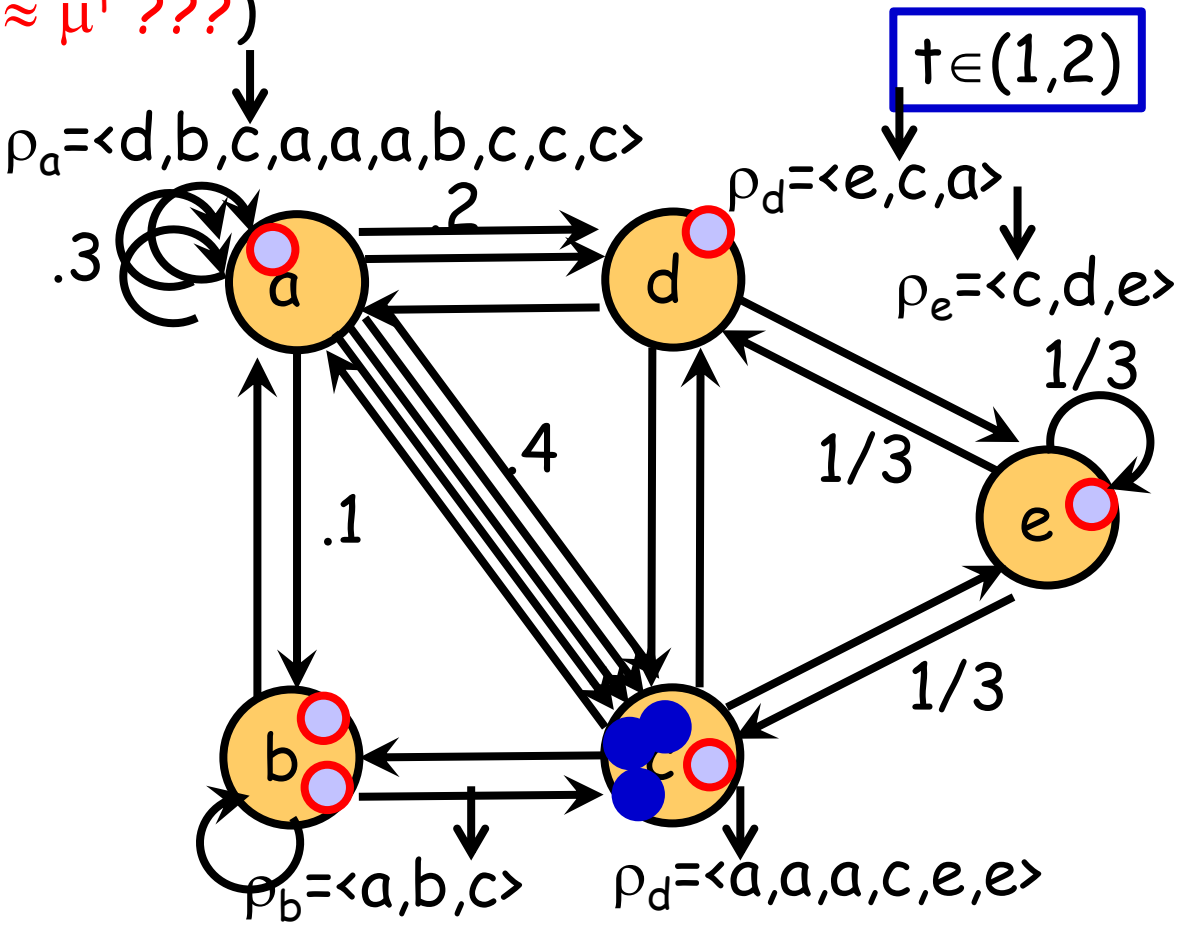
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

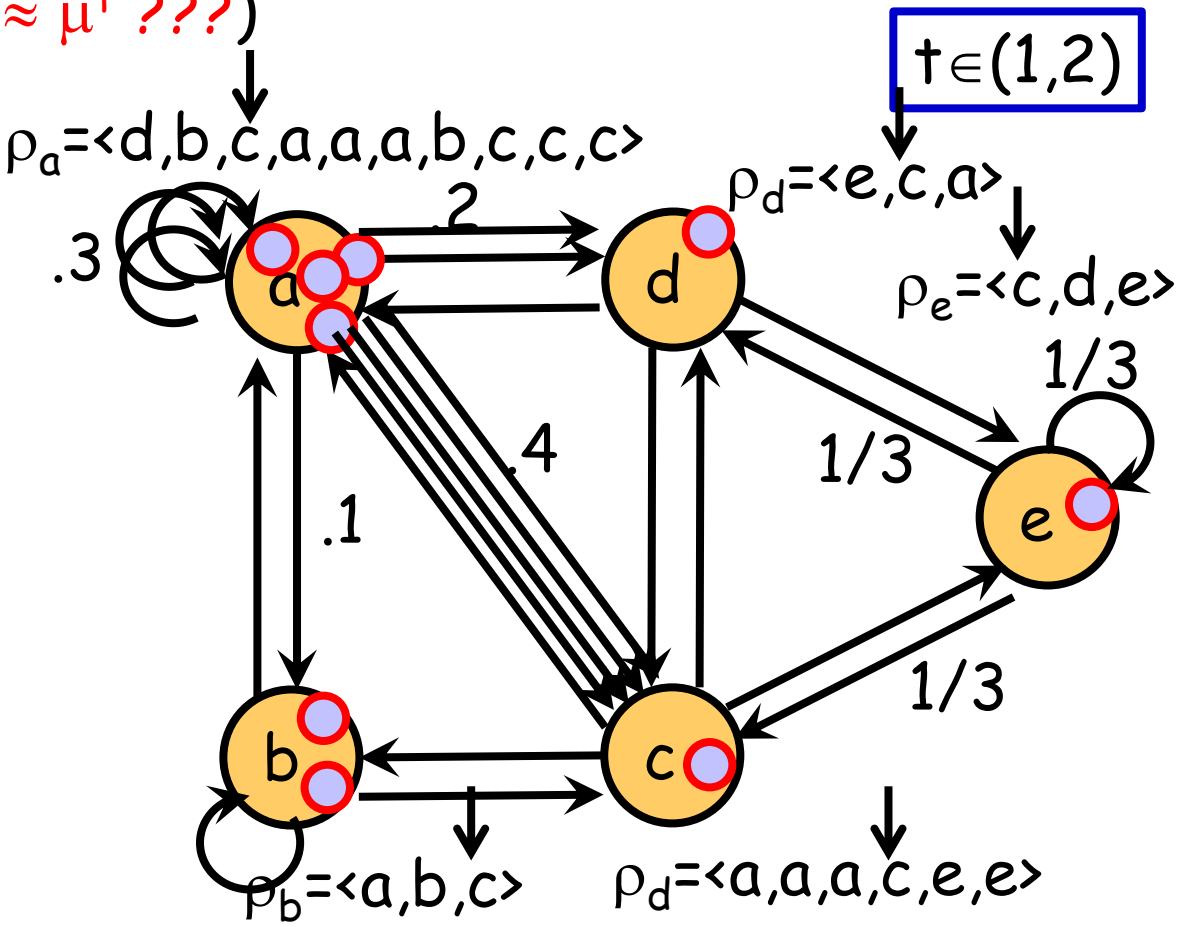
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

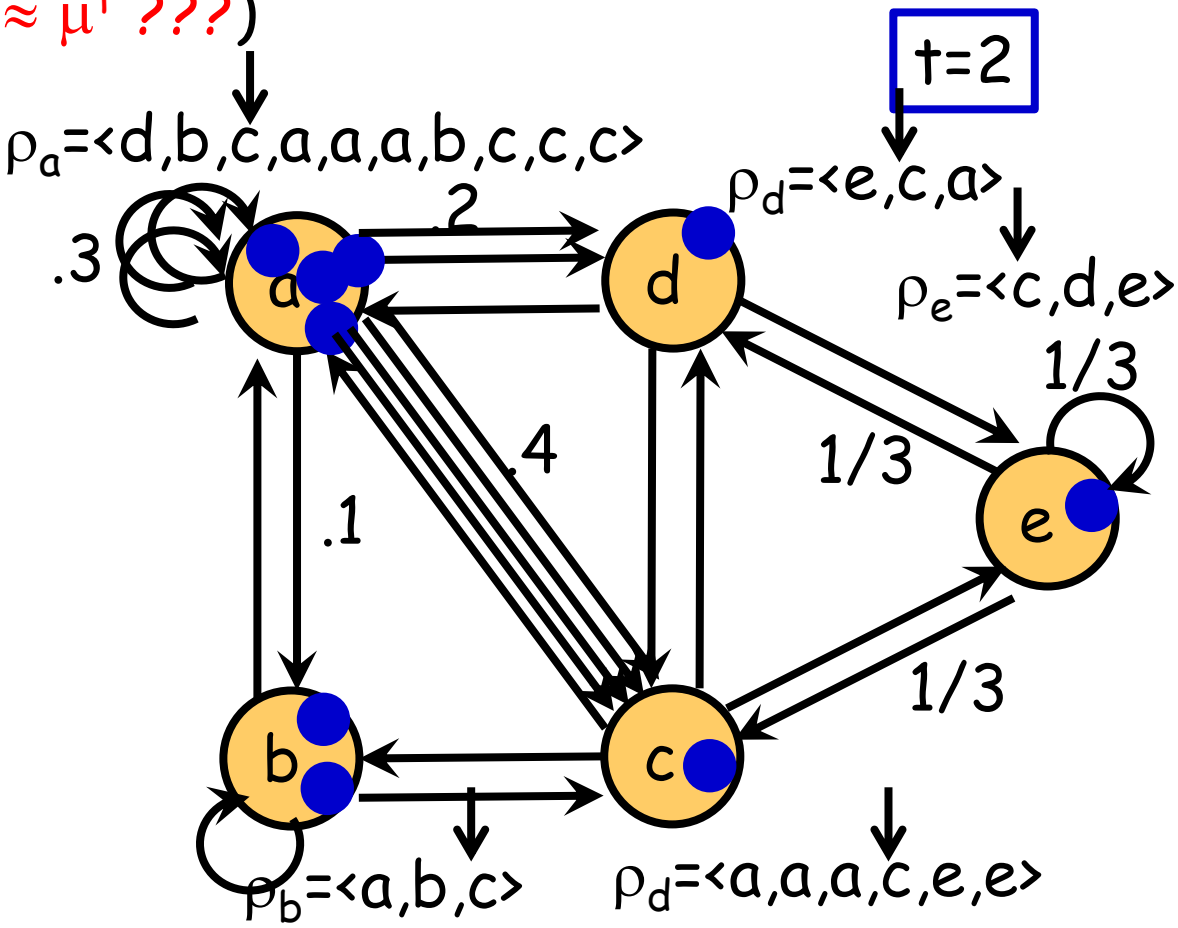
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

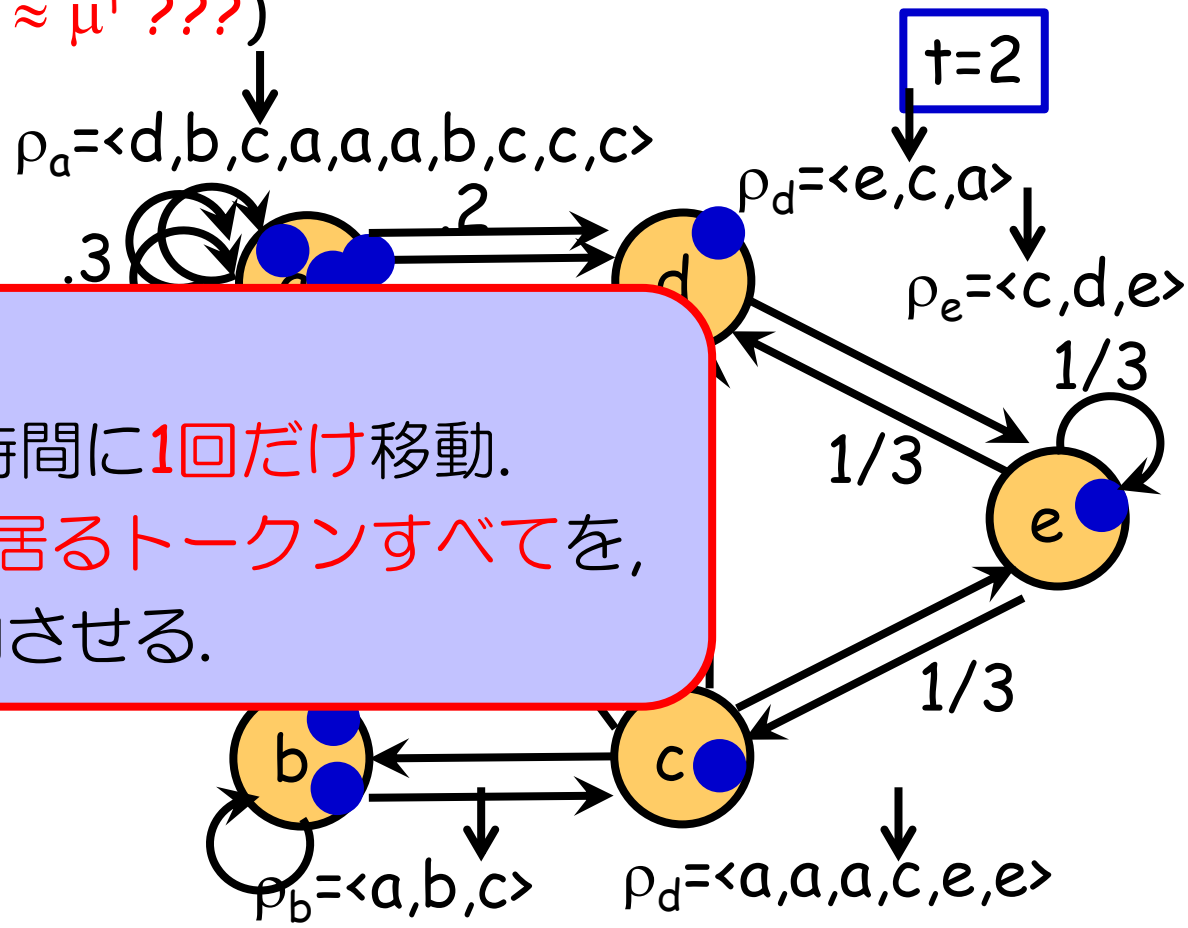
- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



deterministic RW (Propp機械; rotor-router)

N個のトークンがグラフ上を移動する.

- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ ρ: "**rotor router**" (比率P_{uv}で頂点u頂点vに"順次"移動)
- ✓ 時刻tの配置χ^t ($\chi^t \approx \mu^t$???)



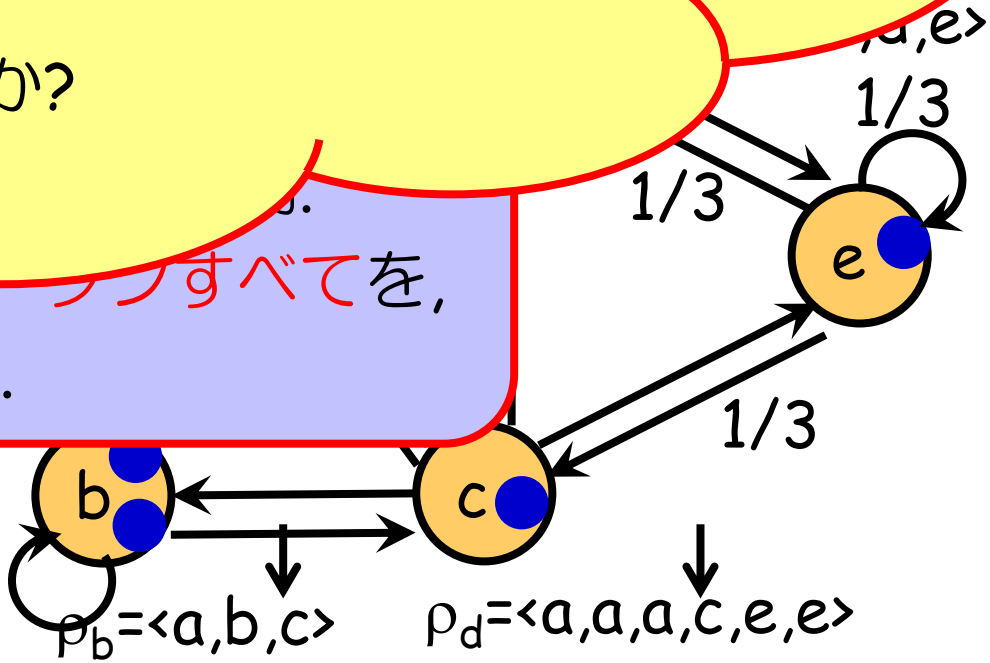
Remark

- ✓ 各トークンは単位時間に1回だけ移動.
- ✓ 各ノードは時刻tに居るトークンすべてを, 時刻t+1までに移動させる.

Propp機械はランダムウォークを模倣できるか?

N個のトークンがグラフ上を移動する.

- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ (エルゴード)マルコフ連鎖は定常分布に収束する
  - ➢ "平滑化"される.
- ✓ Propp機械はランダムウォークを模倣できるか?
  - ➢ "平滑化"できるか?
  - ➢ "だま"はできないのか?
- ✓ 各
- ✓ 各ノードは時刻tに居るトークンすべてを,
  時刻t+1までに移動させる.



誤差の下界

定理 [K, Koga, Makino 10+]

ある多重有向グラフ$\mathcal{G}$=(V,$\mathcal{E}$),

ある初期状態, あるrotor-routerが存在して,

$$\left|\chi_w^{(T)} - \mu_w^{(T)}\right| \geq \Omega(m)$$

但し, mは多重グラフの頂点数,枝数.

だまのできる例

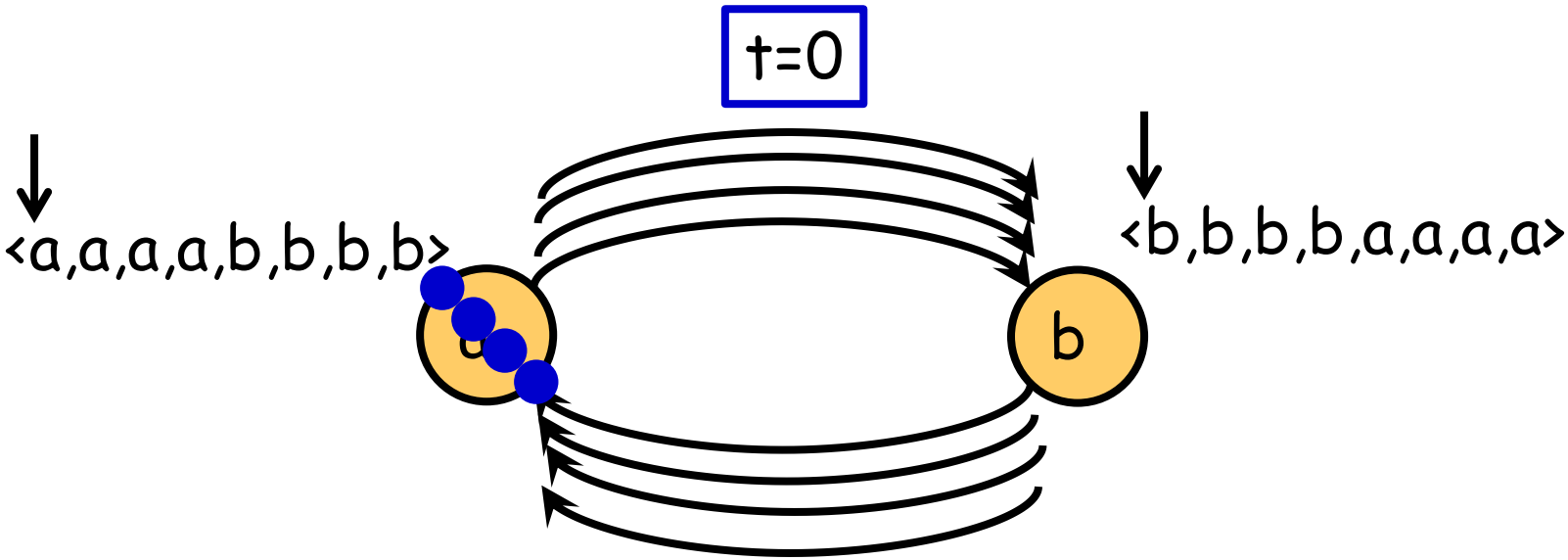
定理 [K, Koga, Makino 10+]

ある多重有向グラフ$\mathcal{G}$=(V,$\mathcal{E}$),

ある初期状態, あるrotor-routerが存在して,

$$\left|\chi_w^{(T)} - \mu_w^{(T)}\right| \geq \Omega(m)$$

但し, mは多重グラフの頂点数,枝数.



だまのできる例

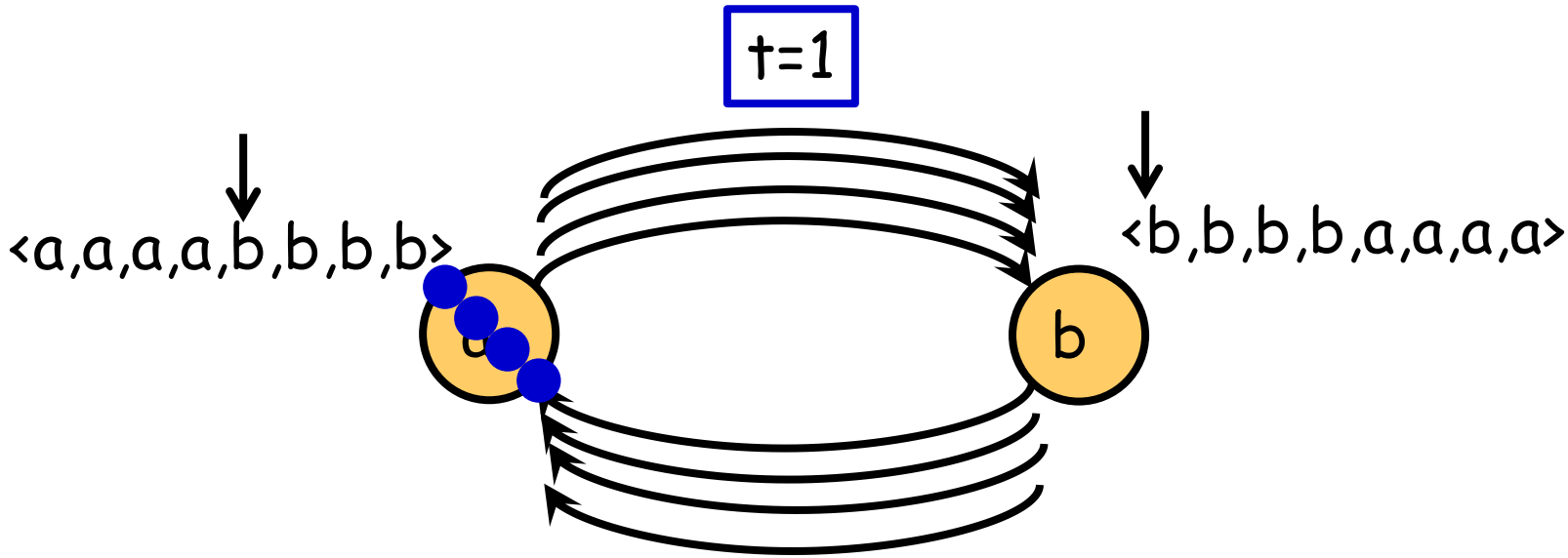
定理 [K, Koga, Makino 10+]

ある多重有向グラフ$G=(V,E)$,

ある初期状態, あるrotor-routerが存在して,

$$\left|\chi_w^{(T)} - \mu_w^{(T)}\right| \geq \Omega(m)$$

但し, mは多重グラフの頂点数,枝数.



だまのできる例

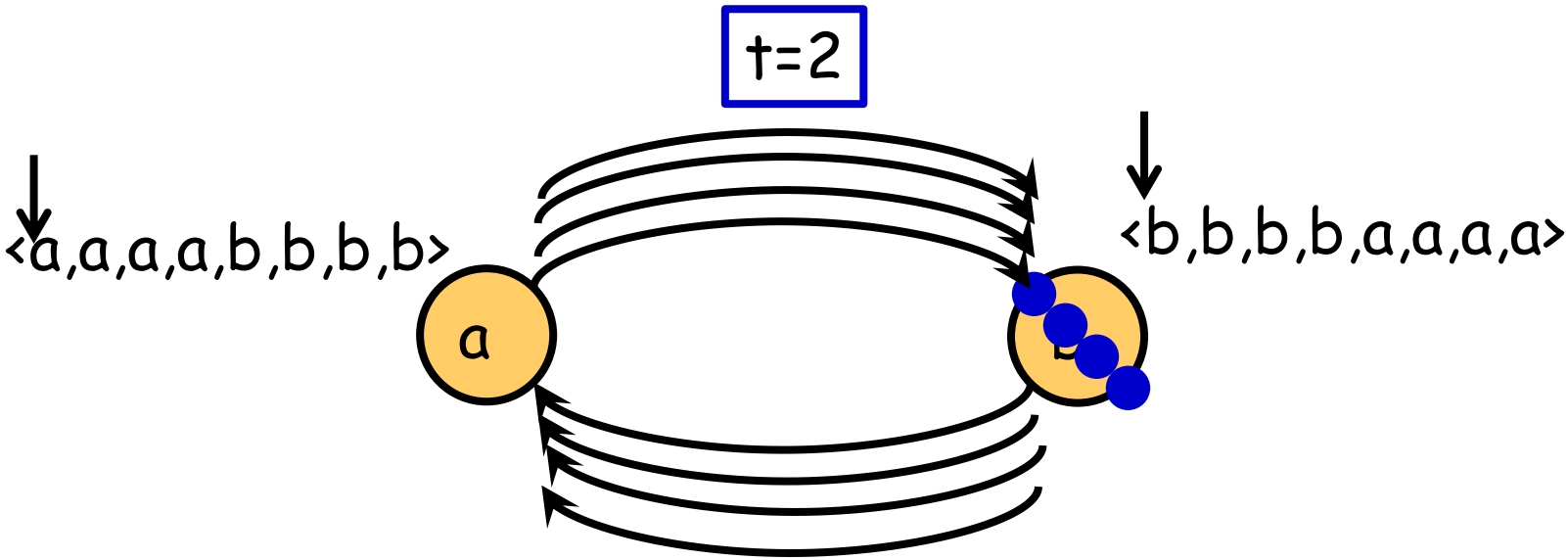
定理 [K, Koga, Makino 10+]

ある多重有向グラフ$G=(V,E)$,

ある初期状態, あるrotor-routerが存在して,

$$\left|\chi_w^{(T)} - \mu_w^{(T)}\right| \geq \Omega(m)$$

但し, mは多重グラフの頂点数,枝数.



だまのできる例

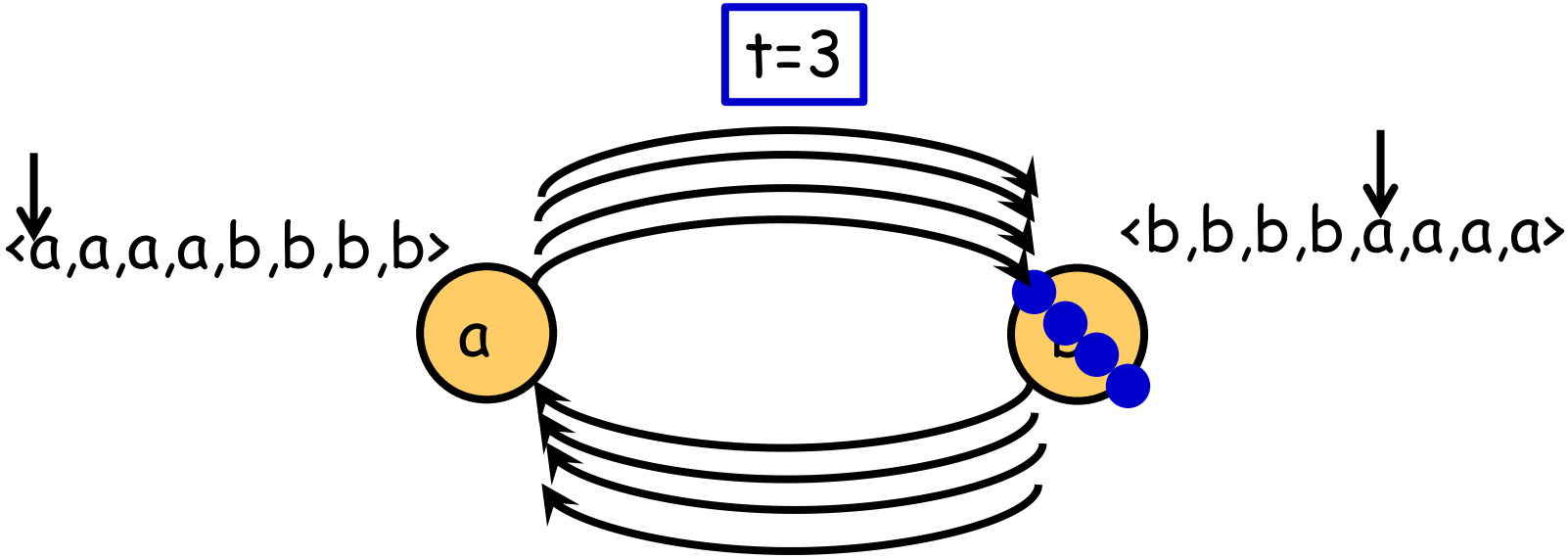
定理 [K, Koga, Makino 10+]

ある多重有向グラフ$G=(V,E)$,

ある初期状態, あるrotor-routerが存在して,

$$\left|\chi_w^{(T)} - \mu_w^{(T)}\right| \geq \Omega(m)$$

但し, mは多重グラフの頂点数,枝数.



だまのできる例

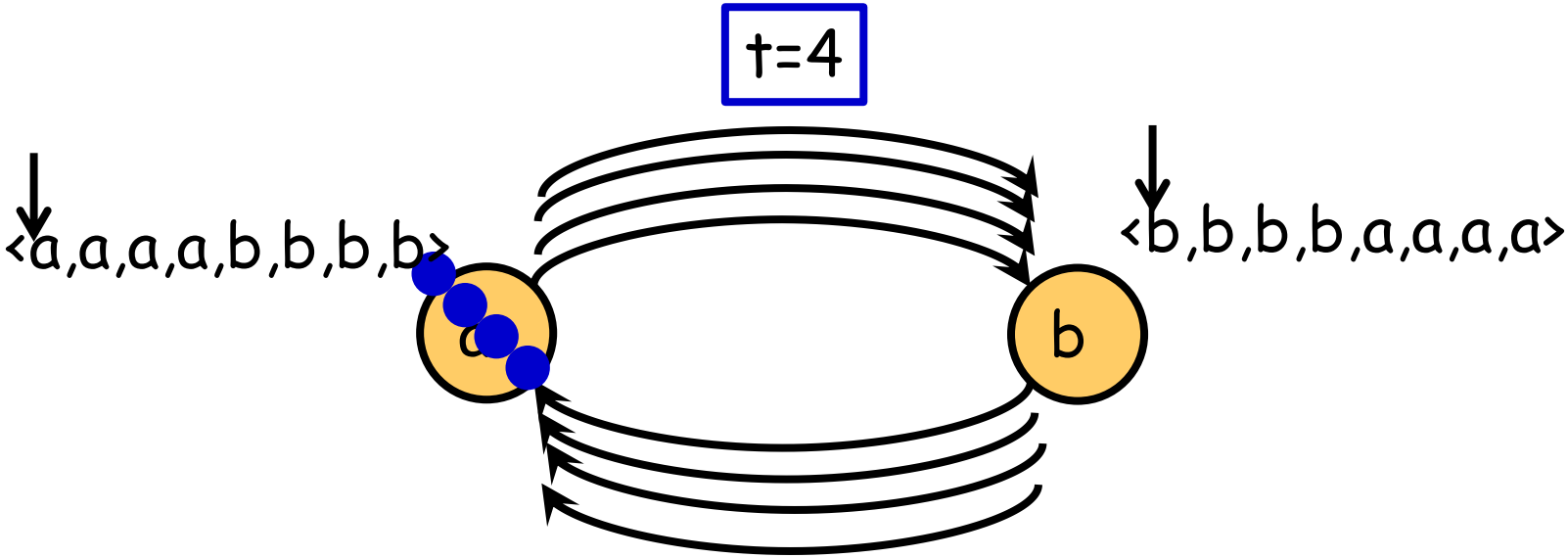
定理 [K, Koga, Makino 10+]

ある多重有向グラフ$G=(V,E)$,

ある初期状態, あるrotor-routerが存在して,

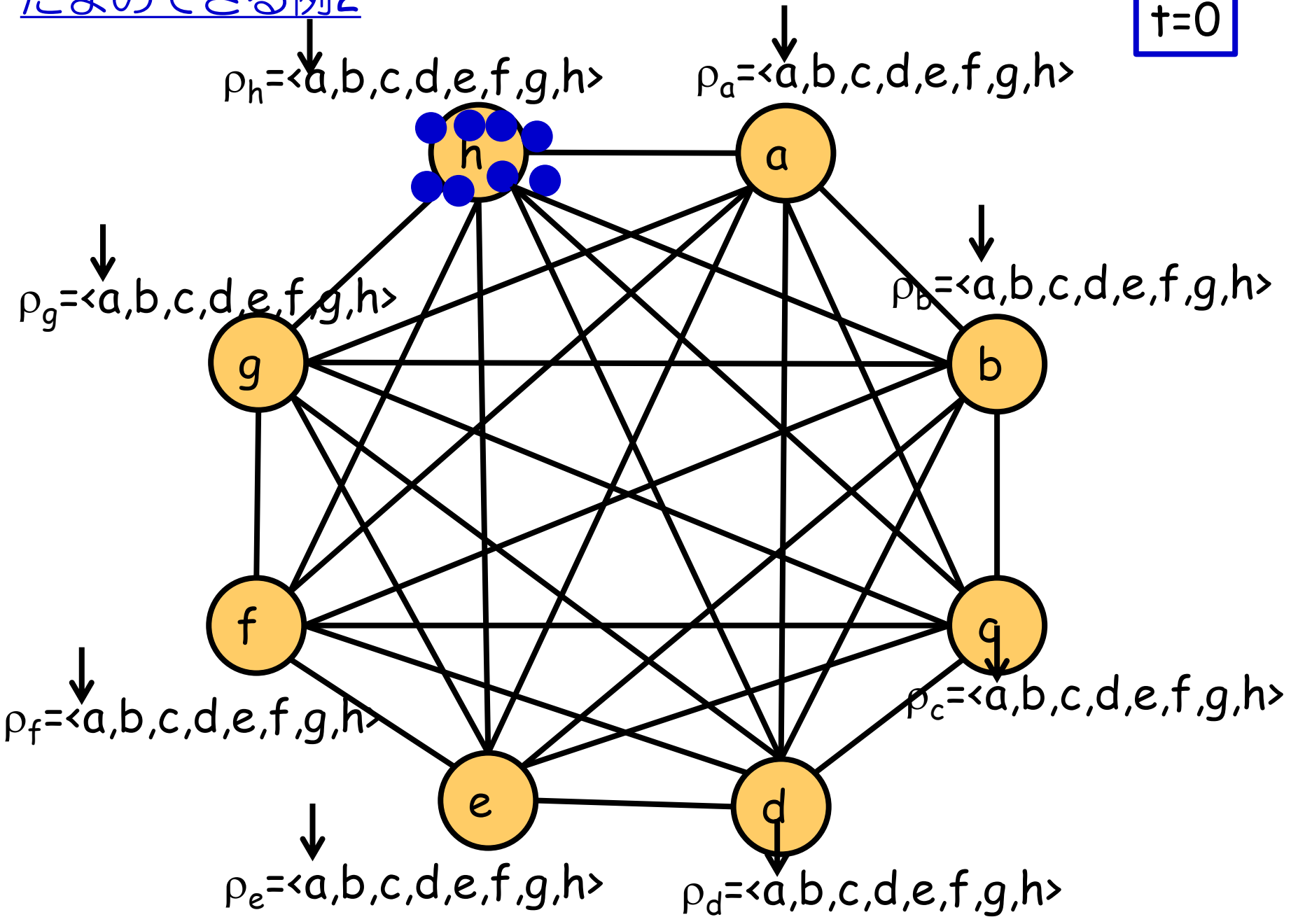
$$\left|\chi_w^{(T)} - \mu_w^{(T)}\right| \geq \Omega(m)$$

但し, mは多重グラフの頂点数,枝数.



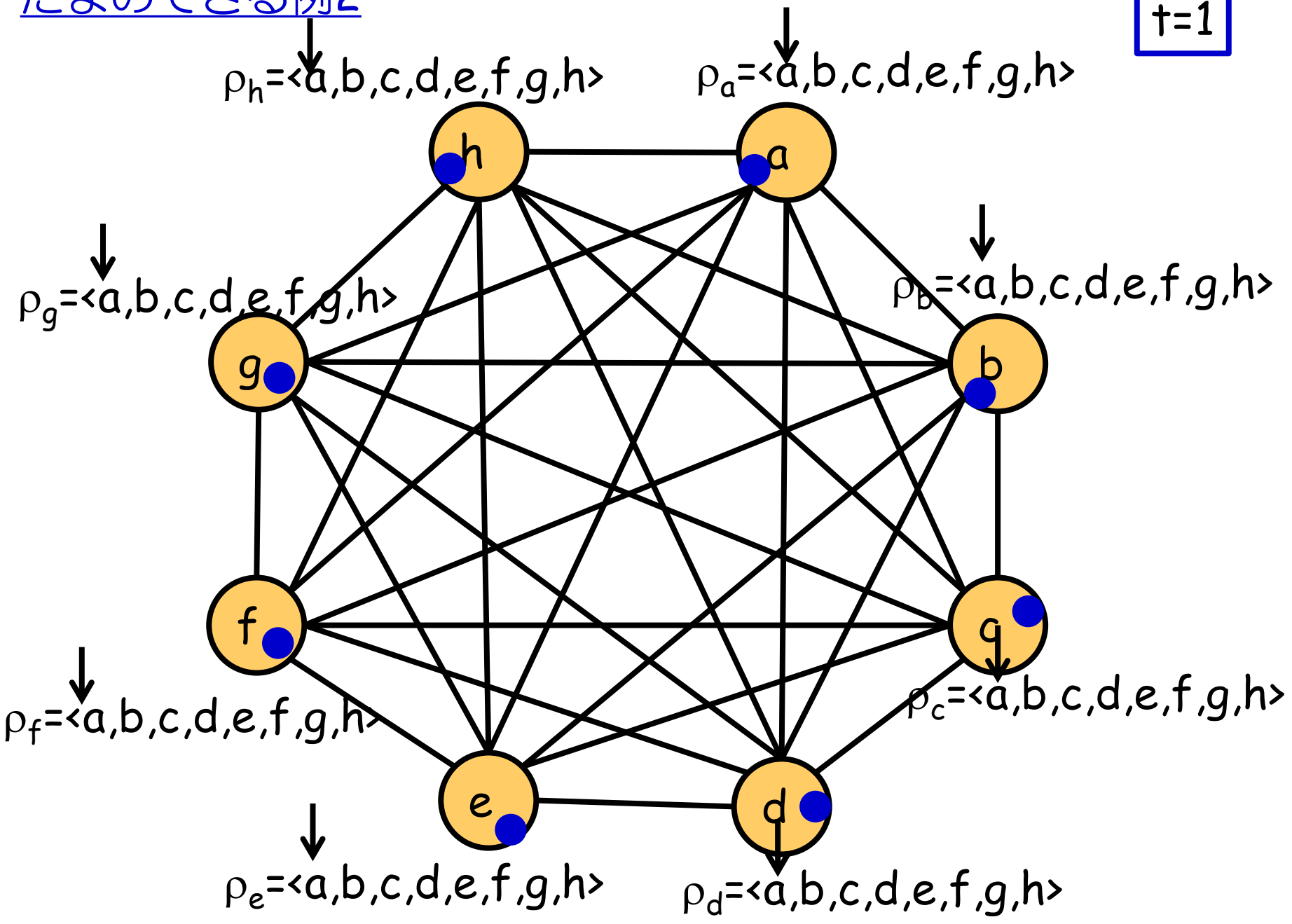
だまのできる例2

t=0


ρh=<a,b,c,d,e,f,g,h>
ρa=<a,b,c,d,e,f,g,h>
h
a
ρg=<a,b,c,d,e,f,g,h>
ρb=<a,b,c,d,e,f,g,h>
g
b
f
c
ρf=<a,b,c,d,e,f,g,h>
ρc=<a,b,c,d,e,f,g,h>
e
d
ρe=<a,b,c,d,e,f,g,h>
ρd=<a,b,c,d,e,f,g,h>

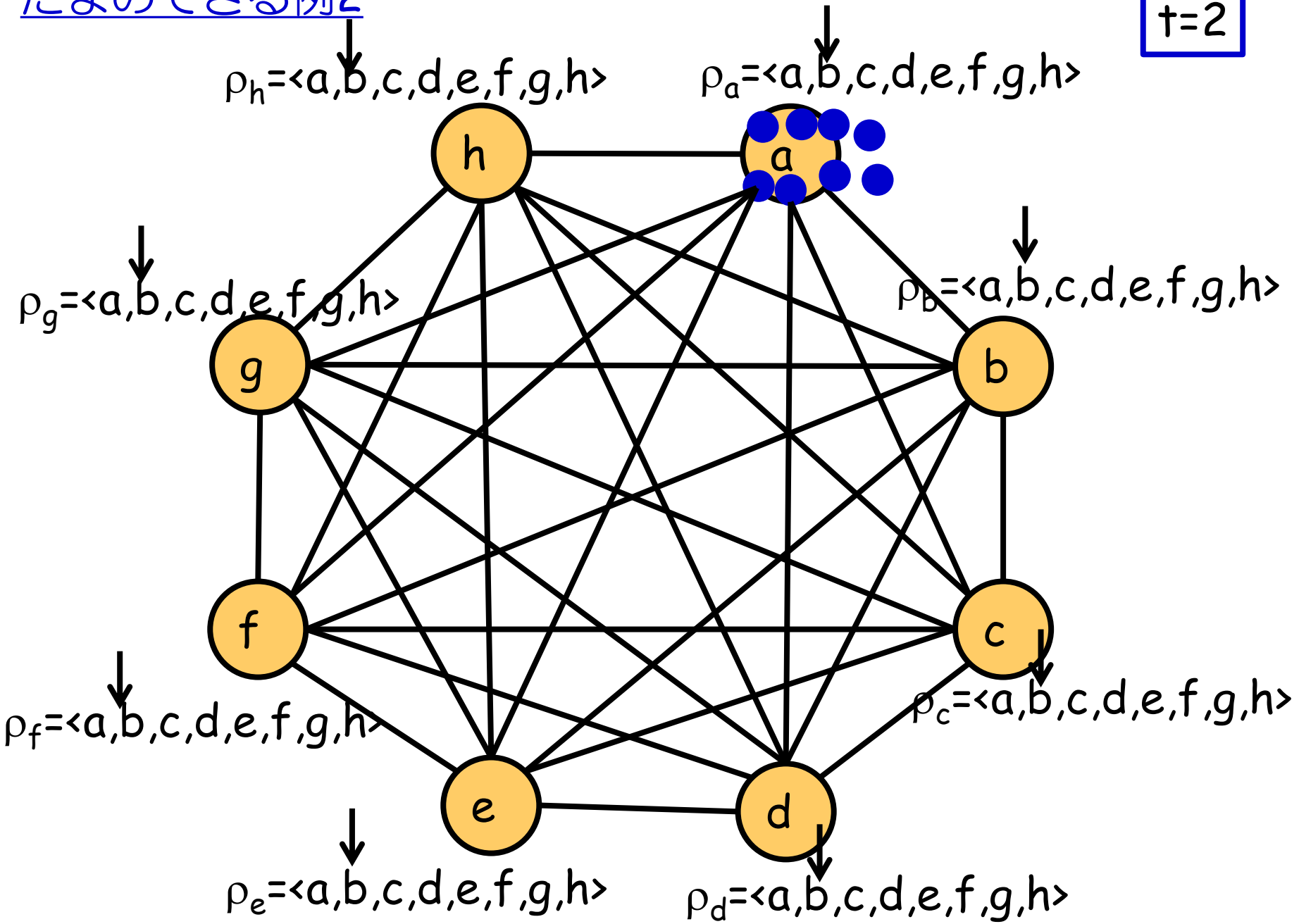
だまのできる例2

t=1


ρh=<a,b,c,d,e,f,g,h>
ρa=<a,b,c,d,e,f,g,h>
h
a
ρg=<a,b,c,d,e,f,g,h>
ρb=<a,b,c,d,e,f,g,h>
g
b
f
c
ρf=<a,b,c,d,e,f,g,h>
ρc=<a,b,c,d,e,f,g,h>
e
d
ρe=<a,b,c,d,e,f,g,h>
ρd=<a,b,c,d,e,f,g,h>

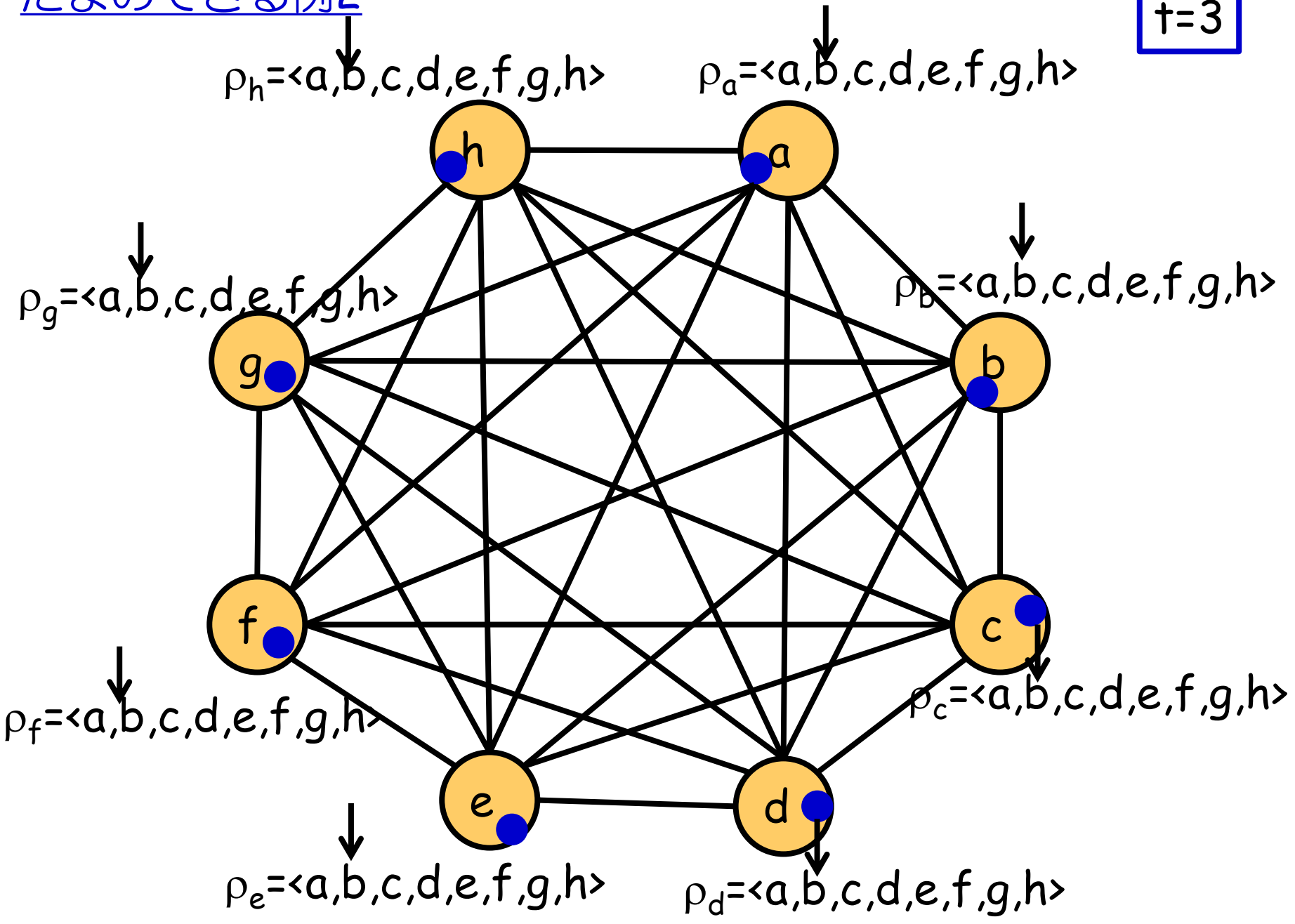
だまのできる例2

t=2


ρh=<a,b,c,d,e,f,g,h>
ρa=<a,b,c,d,e,f,g,h>
h
a
ρg=<a,b,c,d,e,f,g,h>
ρb=<a,b,c,d,e,f,g,h>
g
b
f
c
ρf=<a,b,c,d,e,f,g,h>
ρc=<a,b,c,d,e,f,g,h>
e
d
ρe=<a,b,c,d,e,f,g,h>
ρd=<a,b,c,d,e,f,g,h>

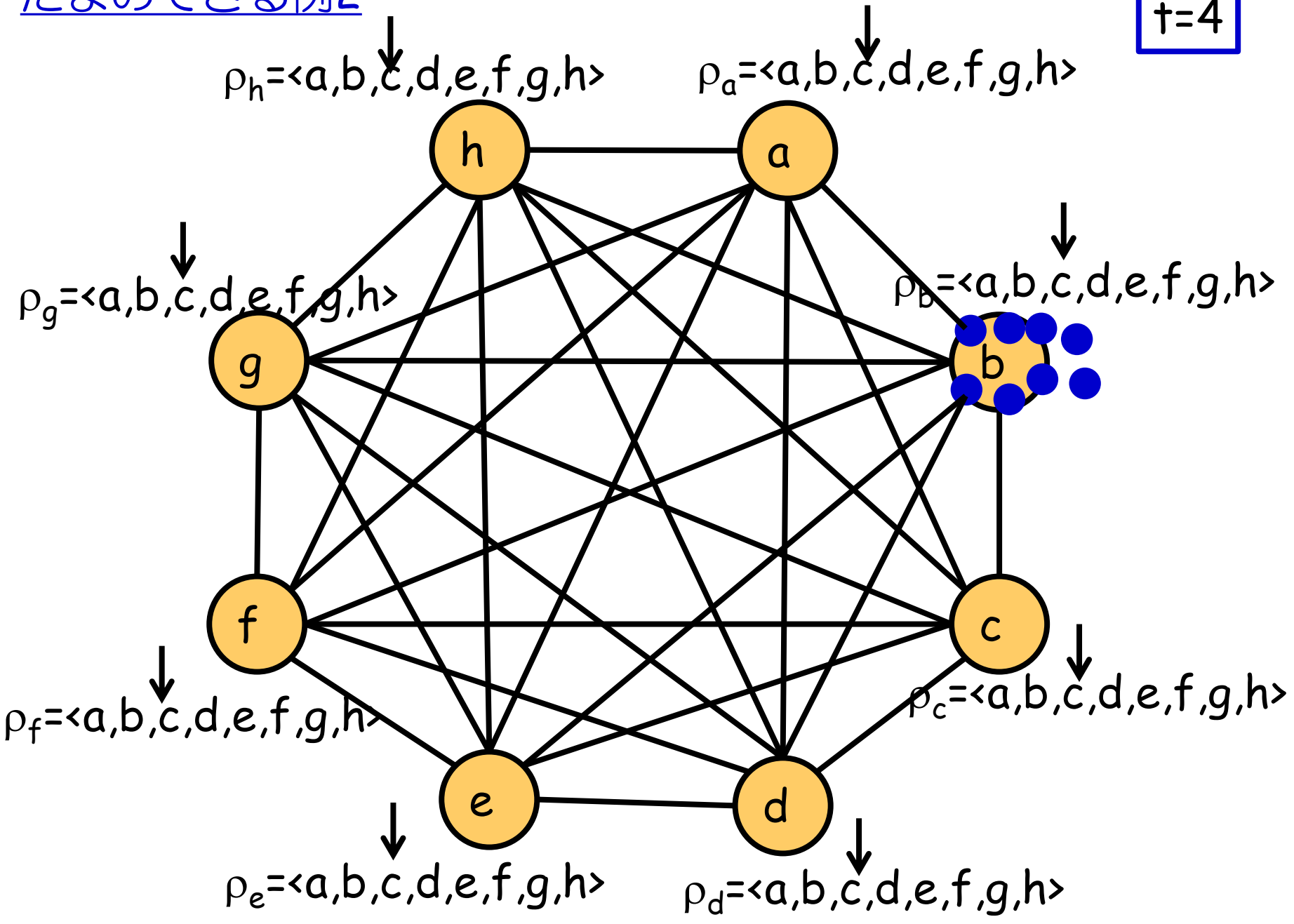
だまのできる例2

t=3


ρh=<a,b,c,d,e,f,g,h>
ρa=<a,b,c,d,e,f,g,h>
h
a
ρg=<a,b,c,d,e,f,g,h>
ρb=<a,b,c,d,e,f,g,h>
g
b
f
c
ρf=<a,b,c,d,e,f,g,h>
ρc=<a,b,c,d,e,f,g,h>
e
d
ρe=<a,b,c,d,e,f,g,h>
ρd=<a,b,c,d,e,f,g,h>

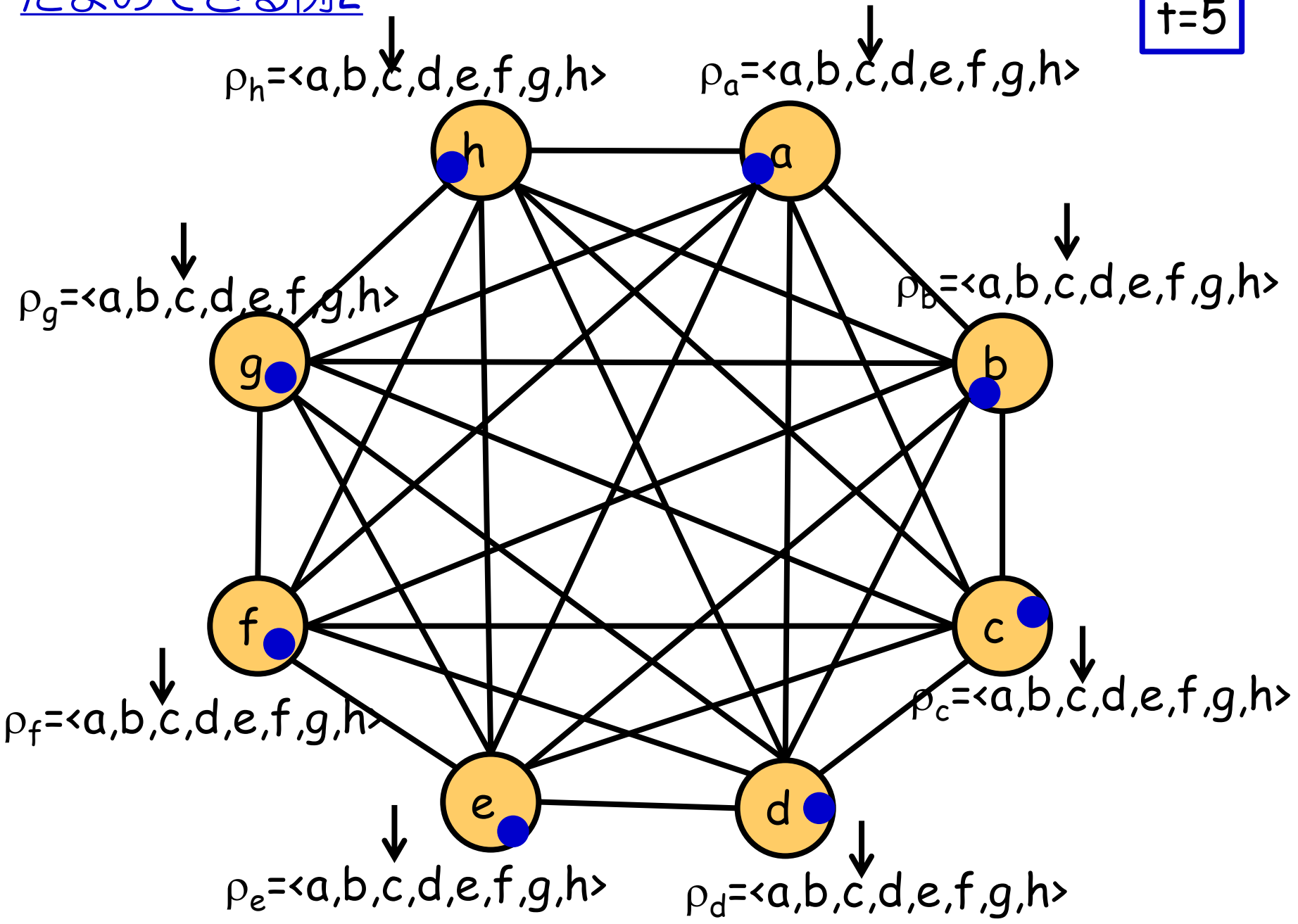
だまのできる例2

t=4


ρh=<a,b,c,d,e,f,g,h>
ρa=<a,b,c,d,e,f,g,h>
h
a
ρg=<a,b,c,d,e,f,g,h>
ρb=<a,b,c,d,e,f,g,h>
g
b
f
c
ρf=<a,b,c,d,e,f,g,h>
ρc=<a,b,c,d,e,f,g,h>
e
d
ρe=<a,b,c,d,e,f,g,h>
ρd=<a,b,c,d,e,f,g,h>

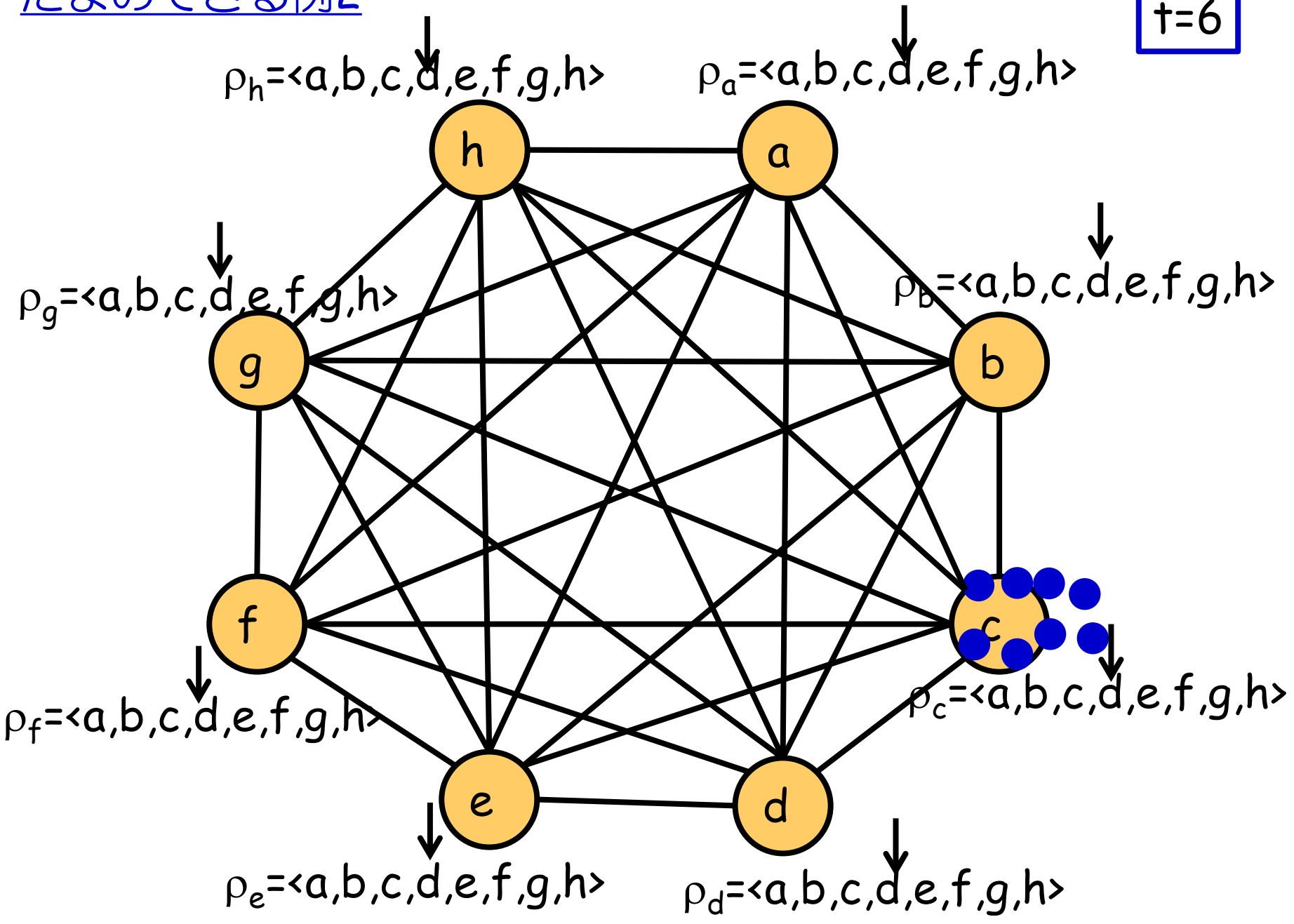
だまのできる例2

t=5


ρh=<a,b,c,d,e,f,g,h>
ρa=<a,b,c,d,e,f,g,h>
h
a
ρg=<a,b,c,d,e,f,g,h>
ρb=<a,b,c,d,e,f,g,h>
g
b
f
c
ρf=<a,b,c,d,e,f,g,h>
ρc=<a,b,c,d,e,f,g,h>
e
d
ρe=<a,b,c,d,e,f,g,h>
ρd=<a,b,c,d,e,f,g,h>

だまのできる例2

t=6


ρh=<a,b,c,d,e,f,g,h>
ρa=<a,b,c,d,e,f,g,h>
h
a
ρg=<a,b,c,d,e,f,g,h>
ρb=<a,b,c,d,e,f,g,h>
g
b
f
c
ρf=<a,b,c,d,e,f,g,h>
ρc=<a,b,c,d,e,f,g,h>
e
d
ρe=<a,b,c,d,e,f,g,h>
ρd=<a,b,c,d,e,f,g,h>

誤差の下界

定理 [K, Koga, Makino 10+]

ある多重有向グラフ$\mathcal{G}$=(V,$\mathcal{E}$),

ある初期状態, あるrotor-routerが存在して,

$$\left|\chi_w^{(T)} - \mu_w^{(T)}\right| \geq \Omega(m)$$

但し, mは多重グラフの頂点数,枝数,

$\chi_w^{(T)}$:　# トークン　@ w∈ V , @ 時刻 T > 0 in Propp 機械

$\mu_w^{(T)}$: E[# トークン] @ w∈ V , @ 時刻 T > 0 in RW

主結果 2

$\chi_w^{(T)}$: # トークン @ w∈ V , @ 時刻 T > 0 in Propp 機械

$\mu_w^{(T)}$: E[# トークン] @ w∈ V , @ 時刻 T > 0 in RW

定理 [K, Koga, Makino 10+]

対応する推移確率行列Pの固有値がすべて非負ならば,
任意の多重有向グラフ, 任意の初期状態, 任意のrotor-router,
任意の頂点w, 任意の時刻tについて,

$$\left|\chi_w^{(T)} - \mu_w^{(T)}\right| \leq (2m-n)\left(\max_{i\in\{1,\ldots,\kappa\}} n_i + n + 3\right) \leq 4mn + O(m)$$

但し, n,mは多重グラフの頂点数,枝数. n_iはJordan cellのサイズ.

Remark

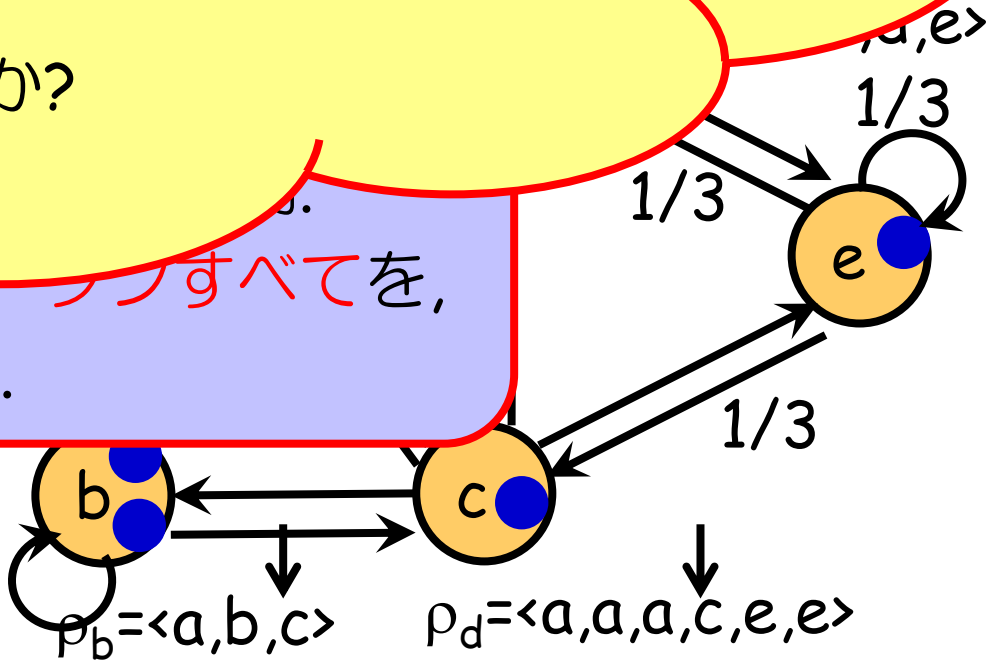
「推移確率行列Pの固有値がすべて非負」という条件.

- reversible lazy Markov chainはこの条件を満たす.
  - ✓ MCMC法で使われるマルコフ連鎖
- +Pが対称 ⇒ Pは半正定値行列

Propp機械はランダムウォークを模倣できるか?

N個のトークンがグラフ上を移動する.

- ✓ χ^0: 初期配置 ($\chi^0 = \mu^0$)
- ✓ (エルゴード)マルコフ連鎖は定常分布に収束する
  - ➢ "平滑化"される.
- ✓ Propp機械はランダムウォークを模倣できるか?
  - ➢ "平滑化"できるか?
  - ➢ "だま"はできないのか?
- ✓ 各
- ✓ 各ノードは時刻tに居るトークンすべてを,
  時刻t+1までに移動させる.



チップ数に依存しない

先行研究と本研究の成果

定理 [Cooper & Spencer 2006]

$\mathbb{Z}^d$上で, 任意の初期状態, 任意のrotor-router,
任意の頂点w, 任意の時刻tについて,
(dのみに依存する)定数c_dが存在して,

$$\left|\chi_w^{(T)} - \mu_w^{(T)}\right| \leq C_d$$

単一頂点誤差に関する先行研究

2006	Cooper, Spencer	Z^d上のPropp機械 ➢ 誤差 $\leq C_d$
2007	Cooper, Doerr, Spencer, Tardos	Z^1上のPropp機械 ➢ $C_1 \leq 2.29$
2008	Cooper, Doerr, Friedrich, Spencer	無限のk正則木上のPropp機械 ➢ 誤差 $> \Omega(\sqrt{kT})$ at time T
2009	Doerr, Friedrich	Z^2上のPropp機械 ➢ $C_2 \leq 7.83$ (上右下左) ➢ $C_2 \leq 7.29$ (上下左右)
2010+	K, Koga, Makino (this work)	有限多重有向グラフ G上のPropp機械 ➢ 誤差$\leq 4mn+O(m)$ $\{0,1\}^d$上のPropp機械 ➢ 誤差$\leq O(d3)$ (頂点数のpoly log)

主結果 2

定理 [K, Koga, Makino 10+]

対応する推移確率行列Pの固有値がすべて非負ならば,
任意の多重有向グラフ, 任意の初期状態, 任意のrotor-router,
任意の頂点w, 任意の時刻tについて,

$$\left|\chi_w^{(T)} - \mu_w^{(T)}\right| \leq (2m-n)\left(\max_{i\in\{1,\ldots,\kappa\}} n_i + n + 3\right) \leq 4mn + \mathrm{O}(m)$$

但し, n,mは多重グラフの頂点数,枝数. n_iはJordan cellのサイズ.

Remark

「推移確率行列Pの固有値がすべて非負」という条件.

- reversible lazy Markov chainはこの条件を満たす.
  - ✓ MCMC法で使われるマルコフ連鎖
- (+) Pが対称 ⇒ Pは半正定値行列

主結果 3

定理 [K, Koga, Makino 10+]

$\{0,1\}^d$超立方体の稜線グラフに対して,
任意の初期状態, 任意のrotor-router,
任意の頂点w, 任意の時刻tについて,

$$\left|\chi_w^{(T)} - \mu_w^{(T)}\right| \leq \frac{3}{2}d^3 + \mathrm{O}(d^2)$$

定理 [K, Koga, Makino 10+]

Johnsonグラフ$J(d,c)$に対して,
任意の初期状態, 任意のrotor-router,
任意の頂点w, 任意の時刻tについて,

$$\left|\chi_w^{(T)} - \mu_w^{(T)}\right| \leq 2c^3 \cdot (d-c)^2 + \mathrm{O}(c^3 \cdot (d-c))$$

Johnsonグラフ $J(d,c) = (V_J, E_J)$
$V_J = \{S \subset \{1,\dots,d\} \mid |S| = c\}$,
$E_J = \{\{S,T\} \in V_J^2 \mid |S \oplus T| = 2\}$

頂点数に対して
対数多項式上界

$\chi_w^{(T)}$: # トークン @ w∈ V , @ 時刻 T > 0 in Propp 機械

$\mu_w^{(T)}$: E[# トークン] @ w∈ V , @ 時刻 T > 0 in RW

証明概略

$$X_v^{(t)} := \sum_{s=0}^{t} \chi_v^{(s)}$$

$$s_v(i) := \min\left\{ t \geq 0 \mid i < \sum X_v^{(t)} \right\},$$

Cooper & Spencer 2006の手法を推移確率行列Pで理解

補題 1

$$\chi_w^{(T)} - \mu_w^{(T)} = \sum_{v \in V} \sum_{i=0}^{X_v^{(T-1)}-1} \left(P^{T-s_v(i)-1}(\rho_v(i), w) - P^{T-s_v(i)}(v, w) \right).$$

「時刻†以前Propp機械, 時刻†以後RW」過程. (初期配置はχ⁰(と同一)とする.)

ζ(w;t,T): 時刻Tの期待トークン配置.

Remark
- ✓ ζ(w;T,T) = χ^T
- ✓ ζ(w;0,T) = μ^T

$$\chi_w^{(T)} - \mu_w^{(T)} = \sum_{t=0}^{T-1} (\zeta(w; t+1, T) - \zeta(w; t, T)).$$

$$\zeta(w; t+1, T) - \zeta(w; t, T) = \sum_{v \in V} \sum_{i=X_v^{(t-1)}}^{X_v^{(t)}-1} \left(P^{T-t-1}(\rho_v(i), w) - P^{T-t}(v, w) \right)$$

関連研究

- ✓ IDLA (Internal Diffusion-Limited Aggregation)
  - Levine & Peres 2005
- ✓ Information Spreading
  - Doerr, Friedrich, & Sauerwald 2008
  - Doerr, Friedrich, Kunnemann, & Sauerwald 2009
- ✓ Hitting time, Cover time
  - Friedrich & Sauerwald 2010
  - Holroyd & Propp 2010+

cf. [K, Koga, Makino 10+]
$|\chi^t_v/N - \pi^t_v| \leq O(mn/N)$

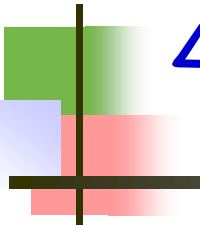
定理 [Holroyd & Propp 2010+]
単一トークンのPropp機械を考える.
F^t_v := 時刻0からtまでに頂点vを訪れた回数.
任意の有限グラフについて,
$|F^t_v/t - \pi^*_v| \leq O(mn/t)$
ただしπ^*は対応するマルコフ連鎖の定常分布.

今後の課題

- ✓ 上下界の一致. (O(mn), Ω(m))
- ✓ 組合せ構造に由来するグラフに対するpolylogの上界.
- ✓ Blanket time vs Mixing time.
- ✓ MCMC法の脱乱択化.
- ✓ 乱数とは？
  - ➢ 乱択アルゴリズムにおける「乱数」の持つべき性質は？
    - ◆ 準モンテカルロ (quasi Monte Carlo)
    - ◆ カオス系列 (Chaos time series)

P. Gopalan, A. Klivans. R. Meka, D. Stefankovic, S. Vempala, E. Vigoda

An FPTAS for #Knapsack and Related Counting Problems, (FOCS 2011)



4. まとめ

1. 乱択の威力: ストリーム中の頻出アイテム検知
   - ✓ $O(\log \log N)$領域計算法
2. 高度な乱択技法: ランダム生成 ≈ 数える
3. 脱乱択化: ランダムウォークの脱乱択化

今後の課題

「乱択アルゴリズムにおいて、乱数に真に求める性質は何か？」

今後の課題

「乱択アルゴリズムにおいて、乱数に真に求める性質は何か？」

講演者の現在

確率的アルゴリズムについて、いろいろ研究

- 最速RW (hitting time, cover time)

[with 野中良哲, 小野廣隆, 山下雅史]

- 順列集合上のオンライン線形最適化

[with 安武 翔太, 畑埜 晃平, 瀧本 英二, 竹田 正幸]

- 自動掃除ロボットは部屋の隅を上手に掃除できるか？

[with 平原昂樹, 山下雅史] etc.

The end

Thank you for the attention.