SHONAN MEETING

Towards Derandomising Markov Chain Monte Carlo

Chunyang Wang (Nanjing University)

Joint work with

Weiming Feng (~~University of Edinburgh~~→UC Berkeley)

Heng Guo (University of Edinburgh)

Jiaheng Wang (University of Edinburgh)

Yitong Yin (Nanjing University)

Randomness in approximate counting

Estimating the volume of a convex body (with membership queries):

- No efficient deterministic polynomial-time algorithms exist! [Elekes '86, Bárány, Füredi '87]
- Efficient randomised algorithms do exist (Markov chain Monte Carlo)! [Dyer, Frieze, Kannan '91]

(Randomized) Counting to Sampling Reduction [Jerrum, Valiant, Varizani '86]

$$Z = \sum_{x \in \Omega} w(x) = \frac{Z(\sigma_{v_1} = 0)}{Z} \cdot \frac{Z(\sigma_{v_1} = 0, \sigma_{v_2} = 0)}{Z(\sigma_{v_1} = 0)} \cdot \dots \cdot \frac{Z(\bigwedge_{i=1}^{n} \sigma_{v_i} = 0)}{Z(\bigwedge_{i=1}^{n-1} \sigma_{v_i} = 0)}$$

Suffices to estimate marginal probabilities

Some Applications

Estimating partition function of ferromagnetic Ising models: [Jerrum, Sinclair'93]

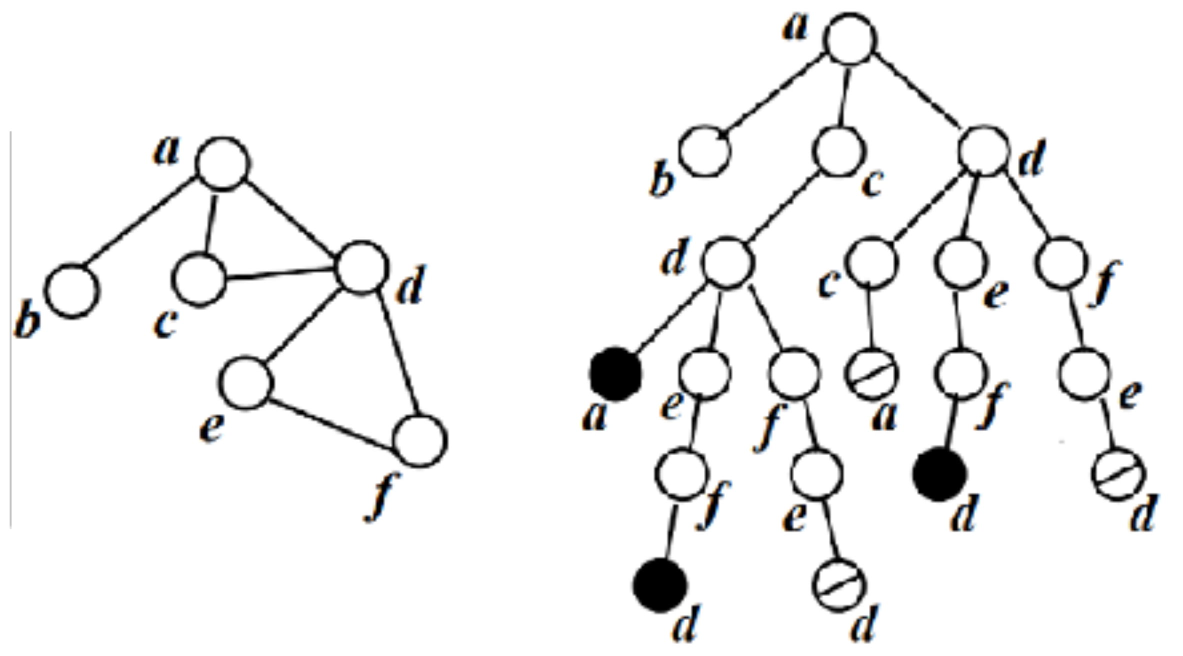
Estimating permanent of non-negative matrices: [Jerrum, Sinclair, Vigoda'04]

Estimating number of bases in matroids: [Anari, Liu, Oveis Gharan, Vinzant'19], [Cryan, Guo, Mousa'21]

Estimating partition functions of spin systems up to critical thresholds: [Anari, Liu, Oveis Gharan'20], [Chen, Liu, Vigoda'20, 21], [Chen, Feng,Yin, Zhang'21,22], [Anari, Jain, Koehler, Pham, Vuong'22], [Chen, Eldan'22]

Deterministic counting

Some approaches for efficient deterministic approximate counting:

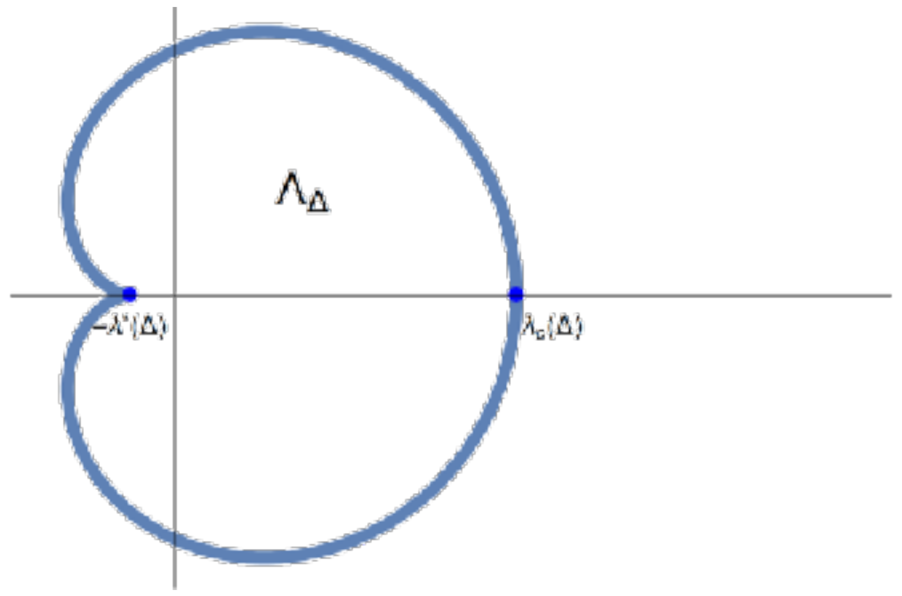


decay of correlation

[Weitz ’06]

[Bayati, Gamarnik, et.al. ’07]

[Gamarnik, Katz ’07]



zero-freeness

[Barvinok ’16]

[Patel, Regts ’17]

$$\text{for every } c \in [q],\ p^{x}_{x,y} = \sum_{c' \in [q]} p^{x^{u\leftarrow c}}_{x^{u\leftarrow c},\, y^{u\leftarrow c'}};$$

$$\text{for every } c \in [q],\ p^{y}_{x,y} = \sum_{c' \in [q]} p^{y^{u\leftarrow c}}_{x^{u\leftarrow c'},\, y^{u\leftarrow c}};$$

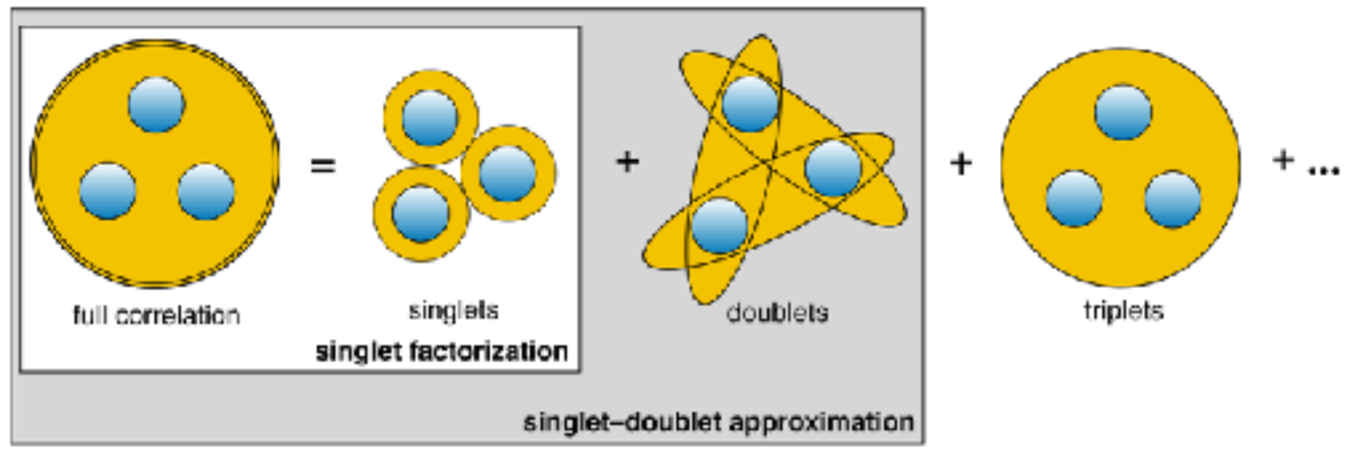
$$0 \le p^{x}_{x,y}, p^{y}_{x,y} \le 1.$$

linear programming for CSPs

[Moitra ’19]

[Guo, Liao, Lu, Zhang ’20]

[Jain, Pham, Vuong ’21]



cluster-expansion

[Helmuth, Perkins, Regts ’20]

[Jenssen, Keevash, Perkins ’20]

MCMC

Derandomisation?

Deterministic Approximate Counting

Markov chain Monte Carlo

(Single-Site) Glauber dynamics

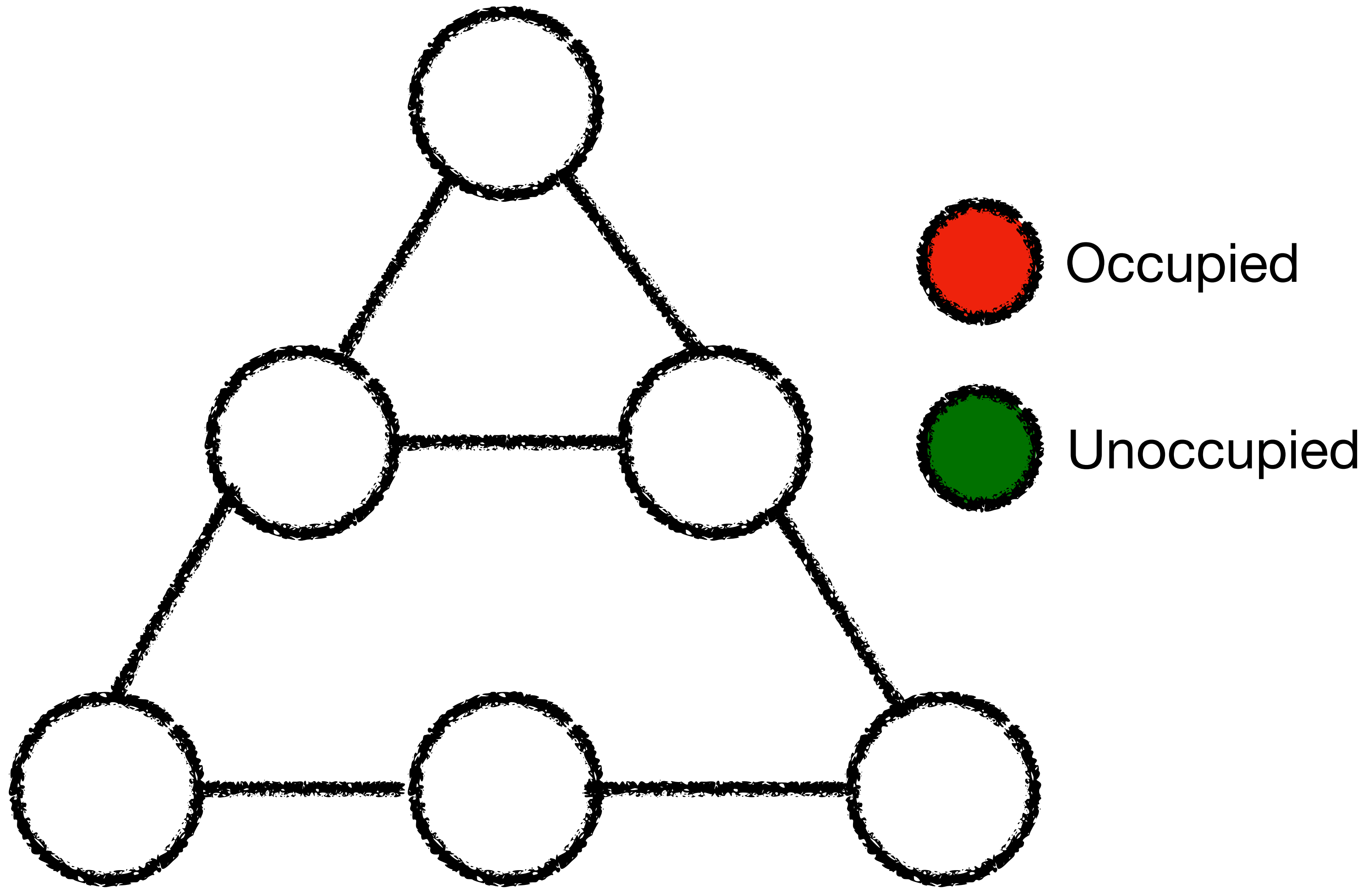
1. Start from any feasible configuration $\sigma \in \Omega$
2. For $t = 0,1\ldots,T-1$, update the configuration as follows:
   (1) Choose some $v \in V$ uniformly at random
   (2) Let $X_t \in [q]^V$ be constructed as that $X_t(u) = X_{t-1}(u)$ for all $u \neq v$, and $X_t(v)$ is drawn independently according to the marginal distribution $\mu_v^{X_{t-1}(V\setminus\{v\})}$.

Desired stationary distribution:
$\mu : [q]^V \to \mathbb{R}^{\geq 0}$

Markov chain Monte Carlo

(Single-Site) Glauber dynamics

1. Start from any feasible configuration $\sigma \in \Omega$
2. For $t = 0,1\ldots, T-1$, update the configuration as follows:
   (1) Choose some $v \in V$ uniformly at random
   (2) Draw a uniform random real $r_t \in [0,1]$
      a. If there exists any occupied neighbour of v: update v as unoccupied
      b. Otherwise, update v as unoccupied if $r_t < \dfrac{1}{1+\lambda}$; occupied otherwise.



Hardcore model

Input: a graph $G = (V, E)$, a fugacity parameter $\lambda > 0$

$\Omega =$ independent sets I of G; for any $\sigma \in I$, $w(\sigma) = \lambda^{|I|}$

Goal: sample from $\mu(\cdot)$, where $\mu(\sigma) = \dfrac{w(\sigma)}{\sum_{X\in\Omega} w(X)}$

Markov chain Monte Carlo

Systematic Scan Glauber dynamics

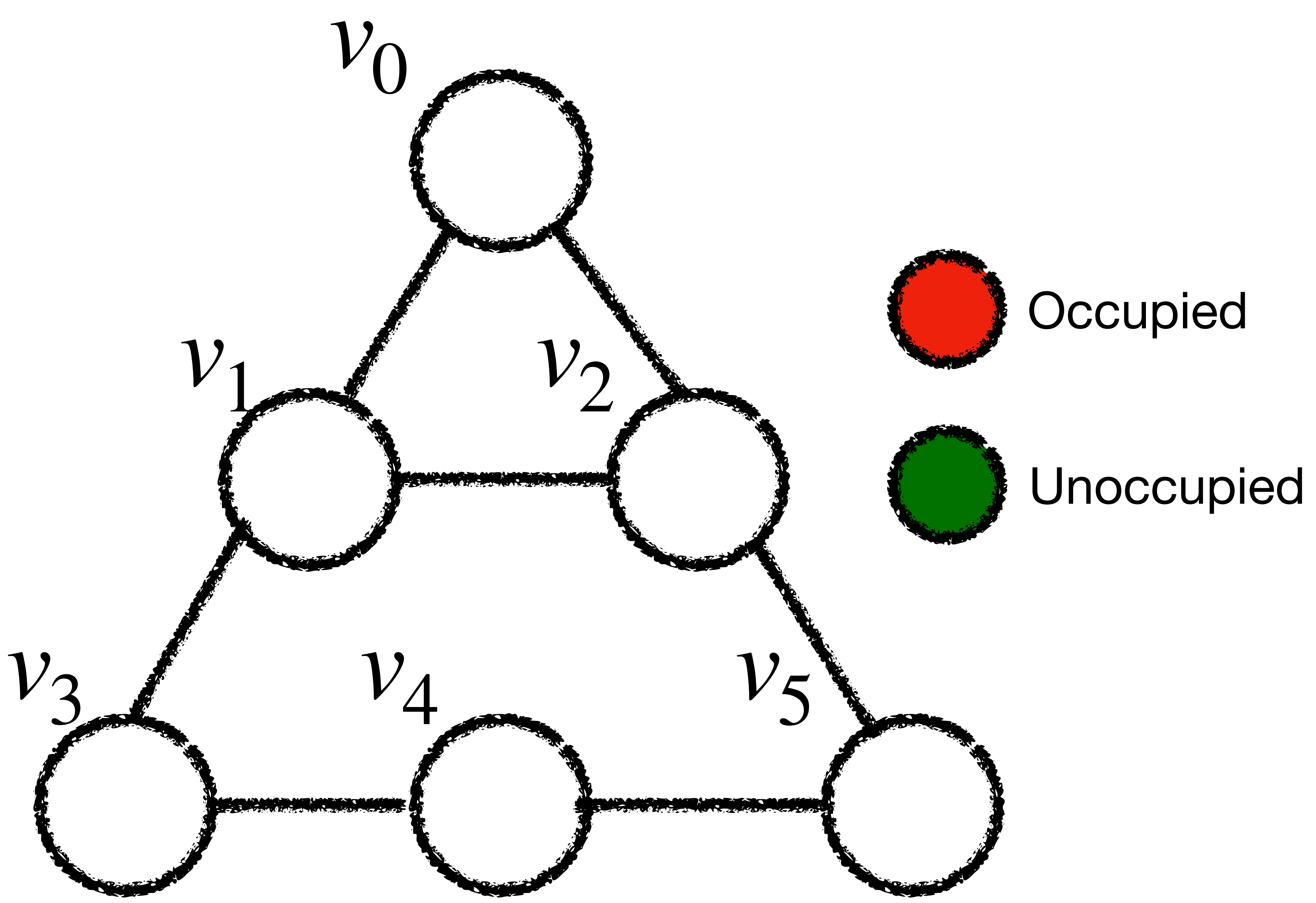
1. Start from any feasible configuration $\sigma \in \Omega$
2. For $t = 0,1\ldots,T-1$, update the configuration as follows:

   (1) Choose $v \in V$ **in the order of** $\mathbf{v_0}, \mathbf{v_1}, \ldots, \mathbf{v_{n-1}}, \mathbf{v_0}, \mathbf{v_1}, \ldots$

   (2) Draw a uniform random real $r_t \in [0,1]$

   a. If there exists any occupied neighbour of v: update v as unoccupied

   b. Otherwise, update v as unoccupied if $r_t < \dfrac{1}{1+\lambda}$; occupied otherwise.



Hardcore model

Input: a graph $G = (V, E)$, a fugacity parameter $\lambda > 0$

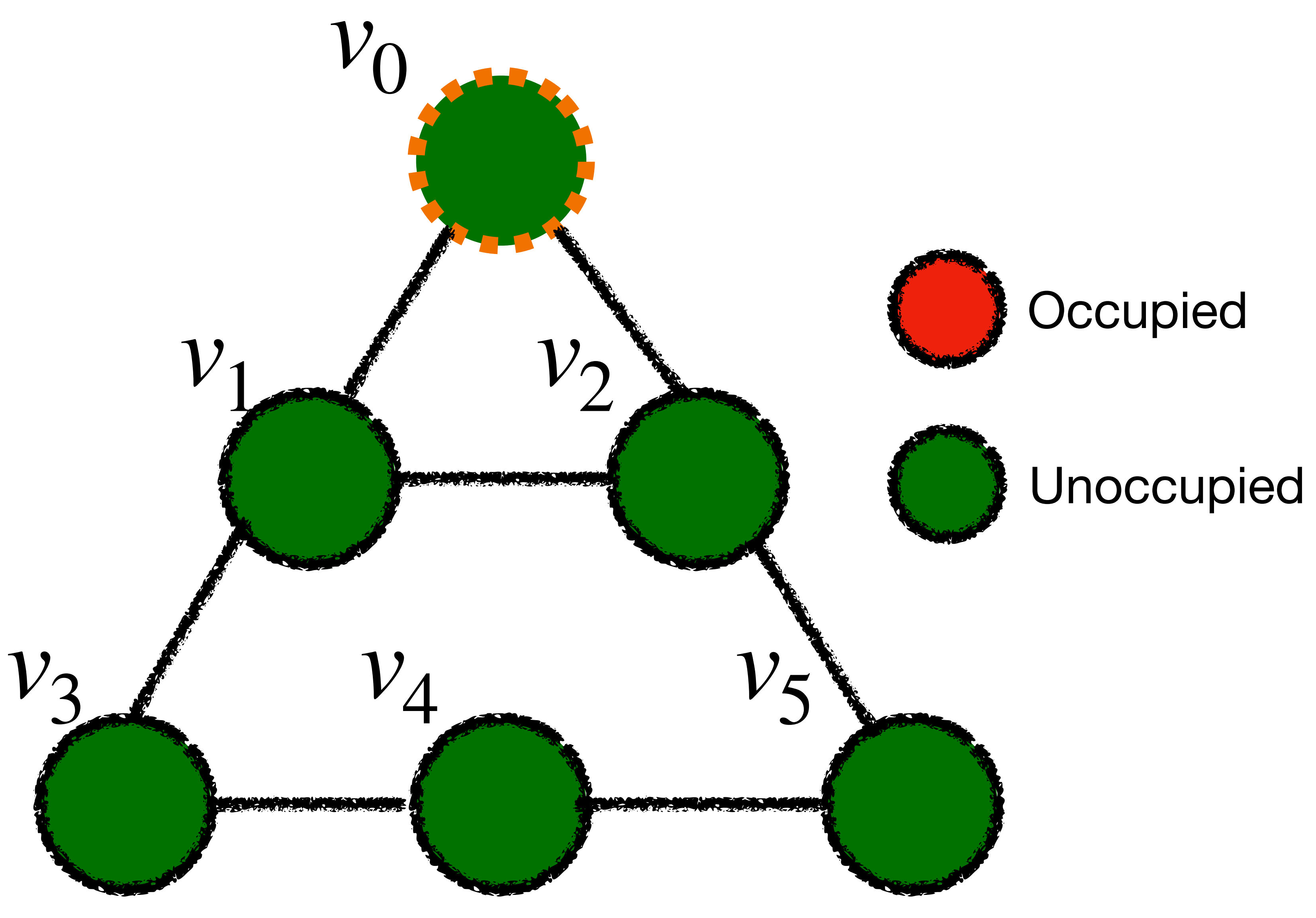
$\Omega =$ independent sets I of G; for any $\sigma \in I$, $w(\sigma) = \lambda^{|I|}$

Goal: sample from $\mu(\cdot)$, where $\mu(\sigma) = \dfrac{w(\sigma)}{\sum_{X\in\Omega} w(X)}$

Markov chain Monte Carlo

Systematic Scan Glauber dynamics

1. Start from any feasible configuration $\sigma \in \Omega$
2. For $t = 0,1\ldots,T-1$, update the configuration as follows:
   (1) Choose $v \in V$ in the order of $v_0, v_1, \ldots, v_{n-1}, v_0, v_1, \ldots$
   (2) Draw a uniform random real $r_t \in [0,1]$
      a. If there exists any occupied neighbour of v: update v as unoccupied
      b. Otherwise, update v as unoccupied if $r_t < \dfrac{1}{1+\lambda}$; occupied otherwise.



$t = 0$

Hardcore model

Input: a graph $G = (V, E)$, a fugacity parameter $\lambda > 0$

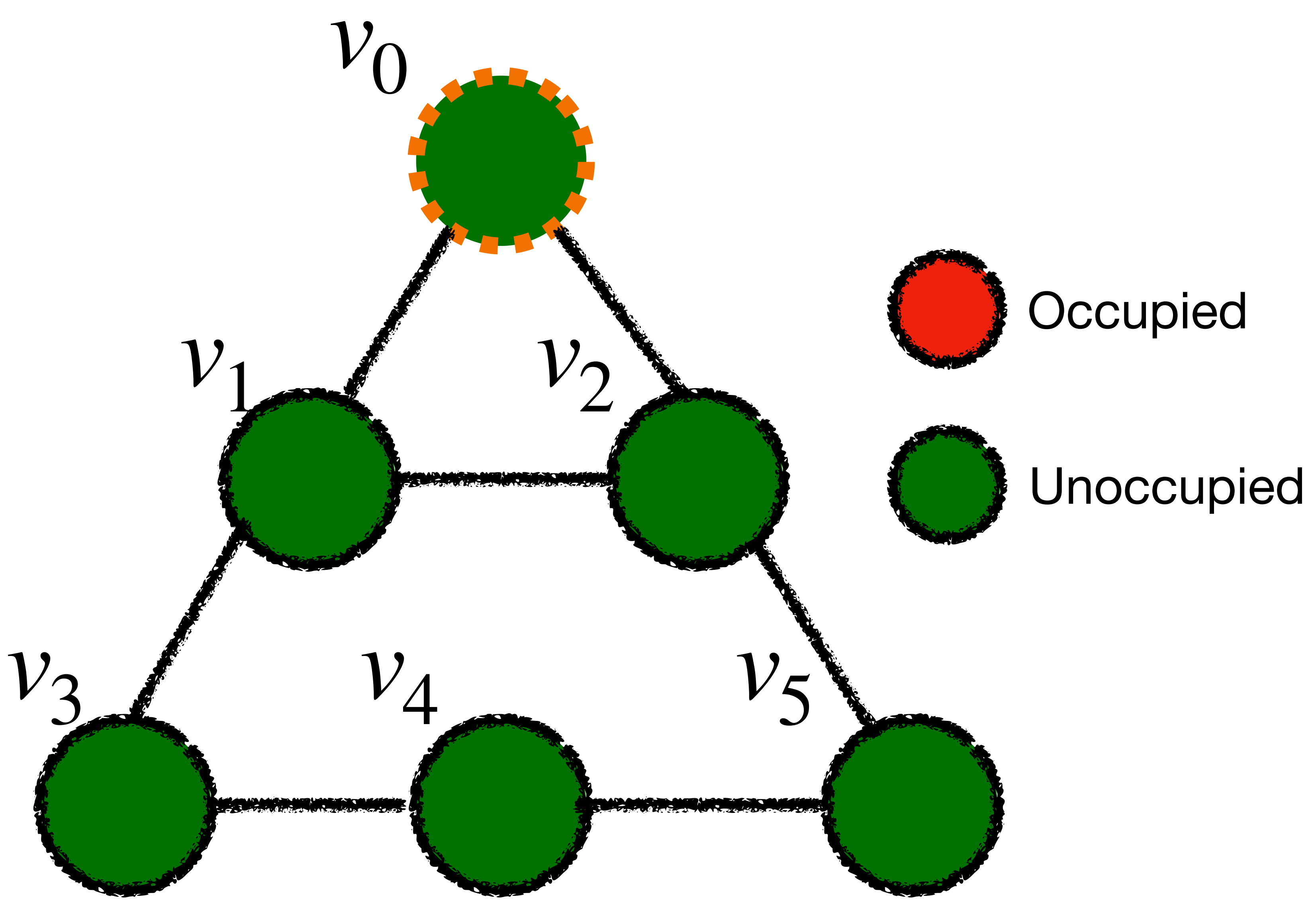
$\Omega =$ independent sets I of G; for any $\sigma \in I$, $w(\sigma) = \lambda^{|I|}$

Goal: sample from $\mu(\cdot)$, where $\mu(\sigma) = \dfrac{w(\sigma)}{\sum_{X \in \Omega} w(X)}$

Markov chain Monte Carlo

Systematic Scan Glauber dynamics

1. Start from any feasible configuration $\sigma \in \Omega$
2. For $t = 0,1\ldots,T-1$, update the configuration as follows:
   (1) Choose $v \in V$ in the order of $v_0, v_1, \ldots, v_{n-1}, v_0, v_1, \ldots$
   (2) Draw a uniform random real $r_t \in [0,1]$
      a. If there exists any occupied neighbour of v: update v as unoccupied
      b. Otherwise, update v as unoccupied if $r_t < \dfrac{1}{1+\lambda}$; occupied otherwise.



$t = 0$ $\quad r_0 \geq \dfrac{1}{1+\lambda}$

Hardcore model

Input: a graph $G = (V, E)$, a fugacity parameter $\lambda > 0$

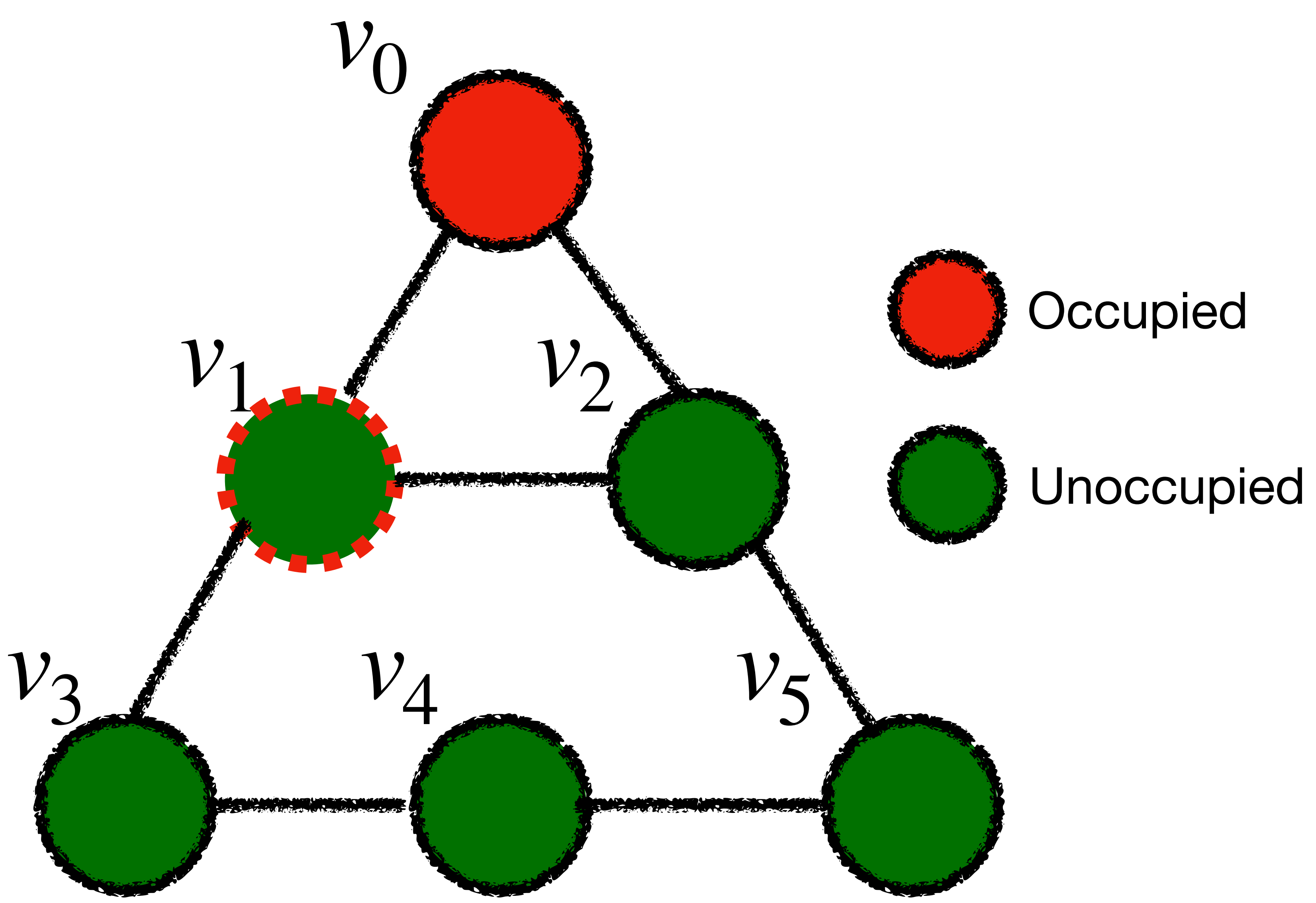
$\Omega =$ independent sets I of G; for any $\sigma \in I$, $w(\sigma) = \lambda^{|I|}$

Goal: sample from $\mu(\cdot)$, where $\mu(\sigma) = \dfrac{w(\sigma)}{\sum_{X \in \Omega} w(X)}$

Markov chain Monte Carlo

Systematic Scan Glauber dynamics

1. Start from any feasible configuration $\sigma \in \Omega$
2. For $t = 0,1\ldots, T-1$, update the configuration as follows:
   (1) Choose $v \in V$ in the order of $v_0, v_1, \ldots, v_{n-1}, v_0, v_1, \ldots$
   (2) Draw a uniform random real $r_t \in [0,1]$
      a. If there exists any occupied neighbour of v: update v as unoccupied
      b. Otherwise, update v as unoccupied if $r_t < \dfrac{1}{1+\lambda}$; occupied otherwise.



$t = 1 \qquad r_1 \geq \dfrac{1}{1+\lambda}$

Hardcore model

Input: a graph $G = (V, E)$, a fugacity parameter $\lambda > 0$

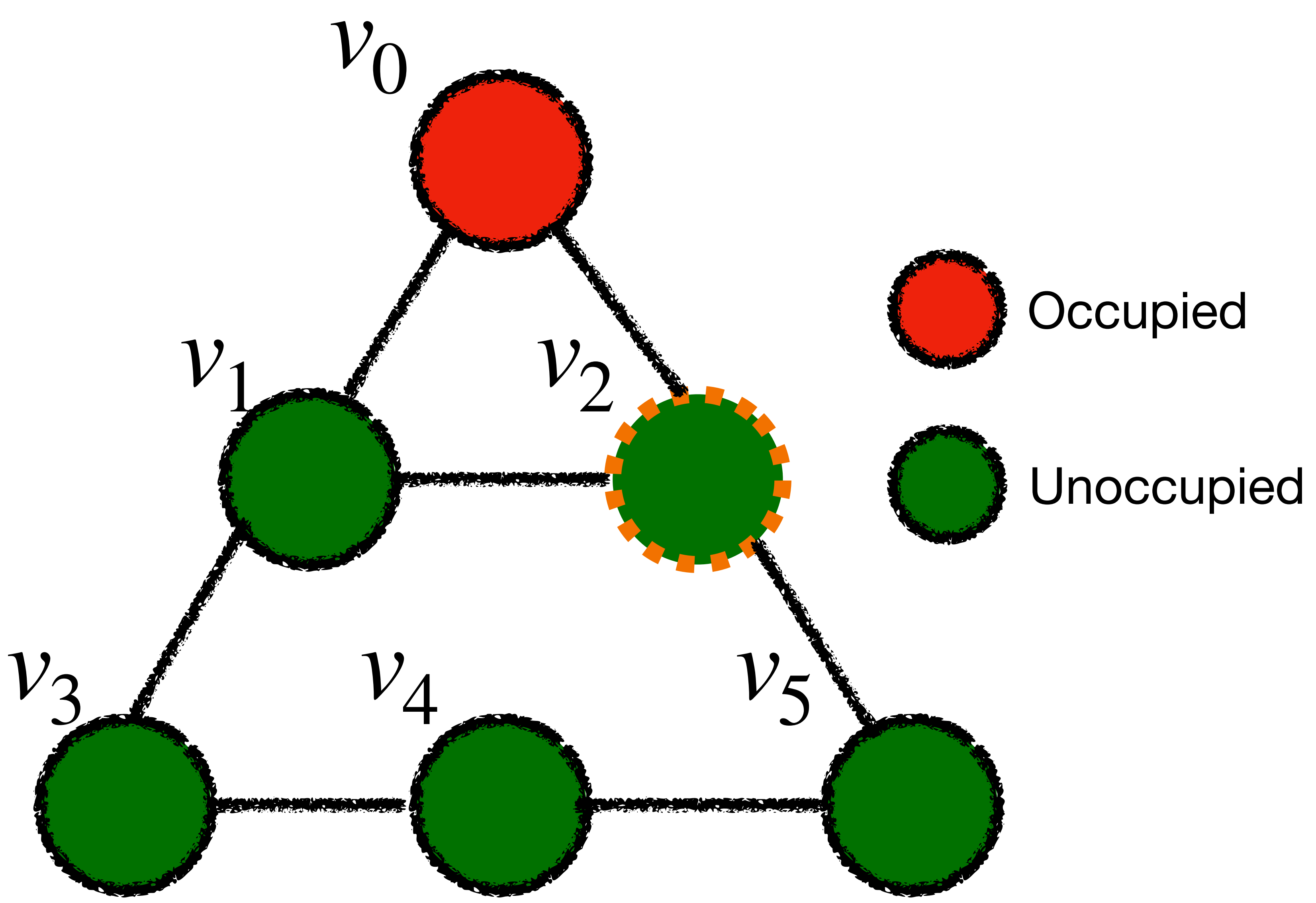
$\Omega =$ independent sets I of G; for any $\sigma \in I$, $w(\sigma) = \lambda^{|I|}$

Goal: sample from $\mu(\cdot)$, where $\mu(\sigma) = \dfrac{w(\sigma)}{\sum_{X \in \Omega} w(X)}$

Markov chain Monte Carlo

Systematic Scan Glauber dynamics

1. Start from any feasible configuration $\sigma \in \Omega$
2. For $t = 0,1\ldots,T-1$, update the configuration as follows:
   (1) Choose $v \in V$ in the order of $v_0, v_1, \ldots, v_{n-1}, v_0, v_1, \ldots$
   (2) Draw a uniform random real $r_t \in [0,1]$
      a. If there exists any occupied neighbour of v: update v as unoccupied
      b. Otherwise, update v as unoccupied if $r_t < \dfrac{1}{1+\lambda}$; occupied otherwise.



$t = 2$ $\quad$ $r_2 < \dfrac{1}{1+\lambda}$

Hardcore model

Input: a graph $G = (V, E)$, a fugacity parameter $\lambda > 0$

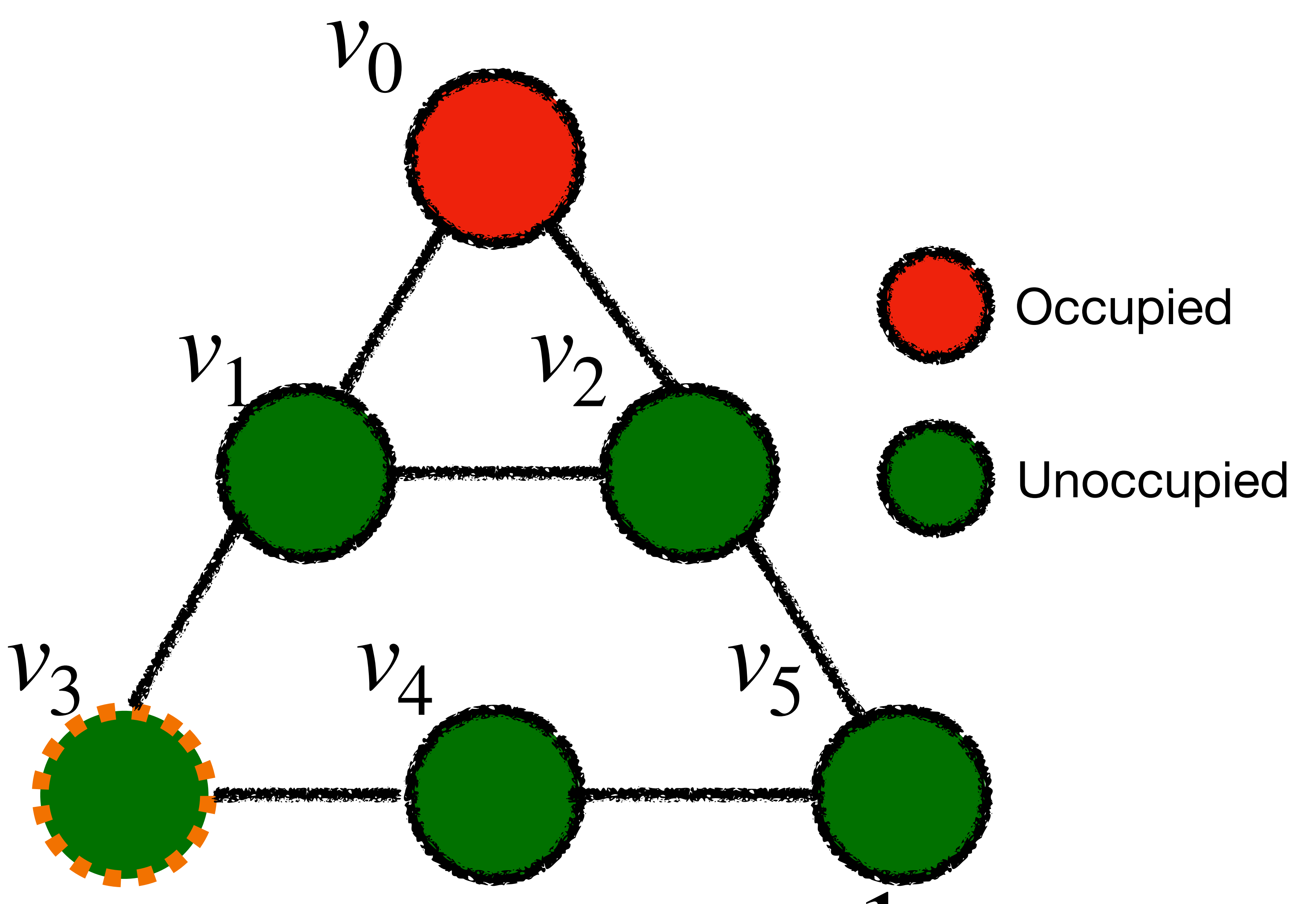
$\Omega =$ independent sets I of G; for any $\sigma \in I$, $w(\sigma) = \lambda^{|I|}$

Goal: sample from $\mu(\cdot)$, where $\mu(\sigma) = \dfrac{w(\sigma)}{\sum_{X \in \Omega} w(X)}$

Markov chain Monte Carlo

Systematic Scan Glauber dynamics

1. Start from any feasible configuration $\sigma \in \Omega$
2. For $t = 0,1\ldots, T-1$, update the configuration as follows:
   (1) Choose $v \in V$ in the order of $v_0, v_1, \ldots, v_{n-1}, v_0, v_1, \ldots$
   (2) Draw a uniform random real $r_t \in [0,1]$
      a. If there exists any occupied neighbour of v: update v as unoccupied
      b. Otherwise, update v as unoccupied if $r_t < \dfrac{1}{1+\lambda}$; occupied otherwise.



$t = 3$ $\quad r_3 \geq \dfrac{1}{1+\lambda}$

Hardcore model

Input: a graph $G = (V, E)$, a fugacity parameter $\lambda > 0$

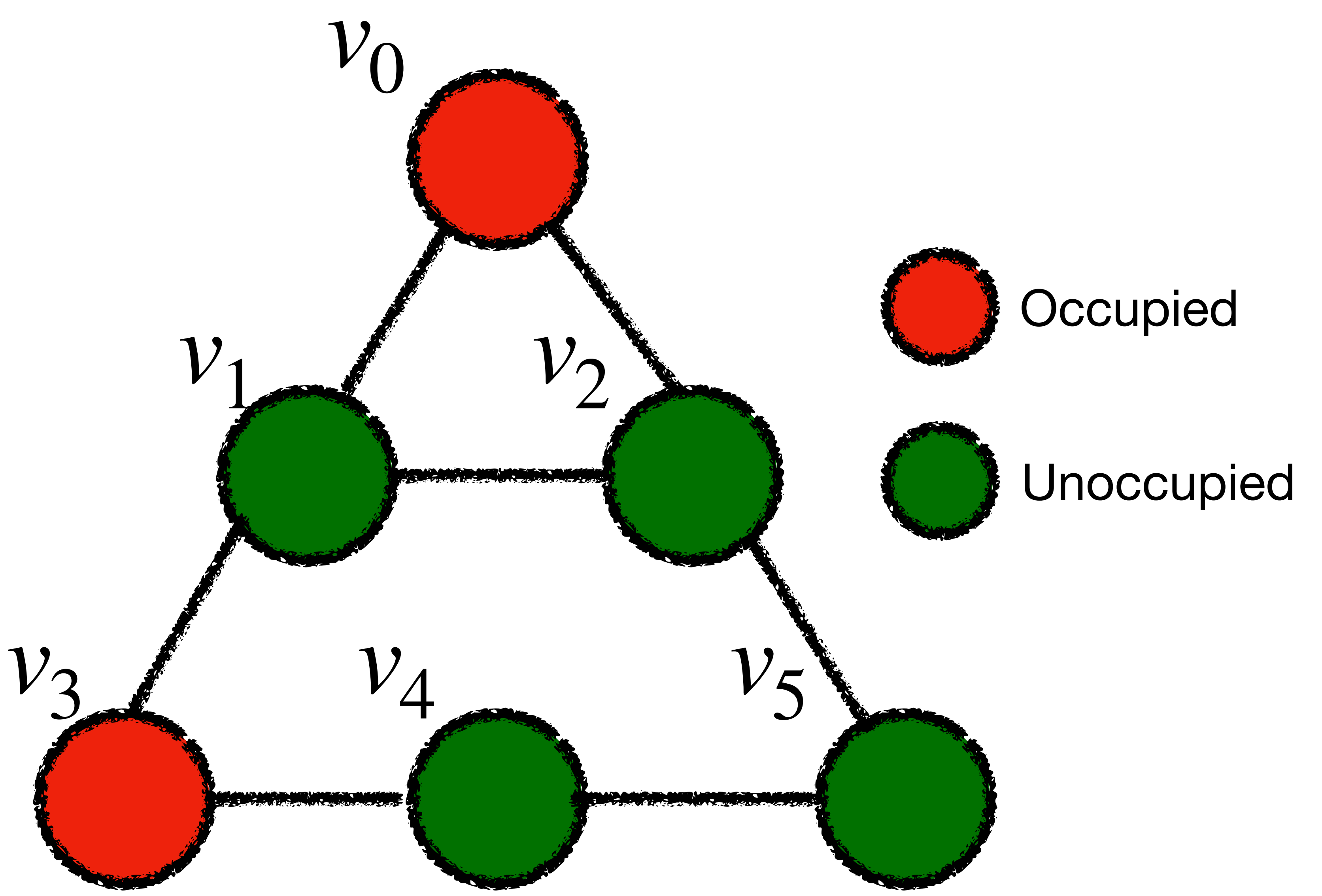
$\Omega =$ independent sets I of G; for any $\sigma \in I$, $w(\sigma) = \lambda^{|I|}$

Goal: sample from $\mu(\cdot)$, where $\mu(\sigma) = \dfrac{w(\sigma)}{\sum_{X \in \Omega} w(X)}$

Markov chain Monte Carlo

Systematic Scan Glauber dynamics

1. Start from any feasible configuration $\sigma \in \Omega$
2. For $t = 0,1\ldots, T-1$, update the configuration as follows:
   (1) Choose $v \in V$ in the order of $v_0, v_1, \ldots, v_{n-1}, v_0, v_1, \ldots$
   (2) Draw a uniform random real $r_t \in [0,1]$
      a. If there exists any occupied neighbour of v: update v as unoccupied
      b. Otherwise, update v as unoccupied if $r_t < \dfrac{1}{1+\lambda}$; occupied otherwise.



converges to $\mu(\cdot)$ as $T \to \infty$
when irreducible

run to $t_{\mathrm{mix}}(\varepsilon) \to$ sample with bias $\leq \varepsilon$

$$t_{\mathrm{mix}}(1/4) = \Omega(n \log n)$$

[Hayes, Sinclair'07]

Hardcore model

Input: a graph $G = (V, E)$, a fugacity parameter $\lambda > 0$

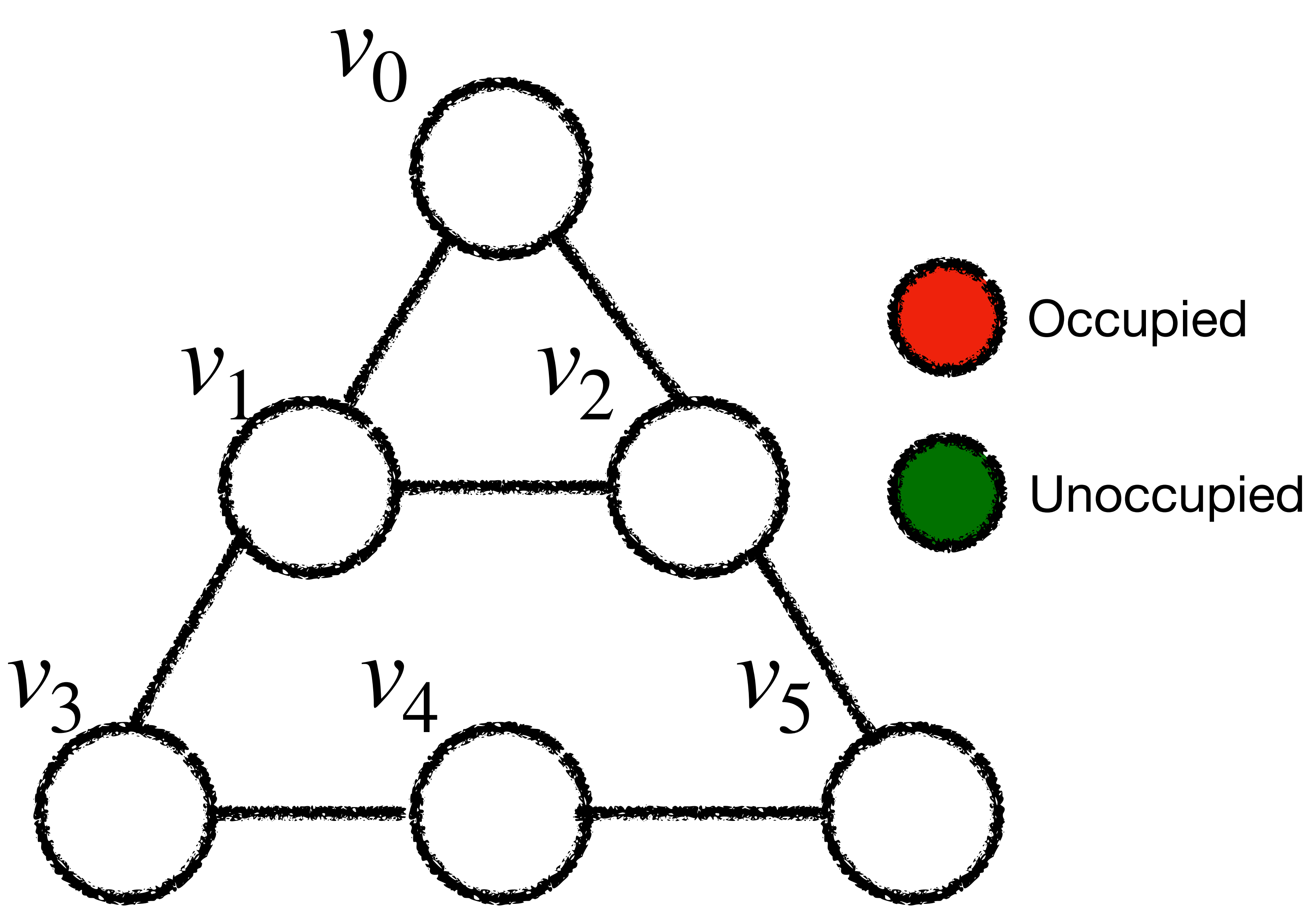
$\Omega =$ independent sets I of G; for any $\sigma \in I$, $w(\sigma) = \lambda^{|I|}$

Goal: sample from $\mu(\cdot)$, where $\mu(\sigma) = \dfrac{w(\sigma)}{\sum_{X \in \Omega} w(X)}$

A Marginal Sampler from MCMC

Systematic Scan Glauber dynamics

1. Start from any feasible configuration $\sigma \in \Omega$
2. For $t = 0,1\ldots,T-1$, update the configuration as follows:
   (1) Choose $v \in V$ in the order of $v_0, v_1, \ldots, v_{n-1}, v_0, v_1, \ldots$
   (2) Draw a uniform random real $r_t \in [0,1]$
      a. If there exists any occupied neighbour of v: update v as unoccupied
      b. Otherwise, update v as unoccupied if $r_t < \dfrac{1}{1+\lambda}$; occupied otherwise.



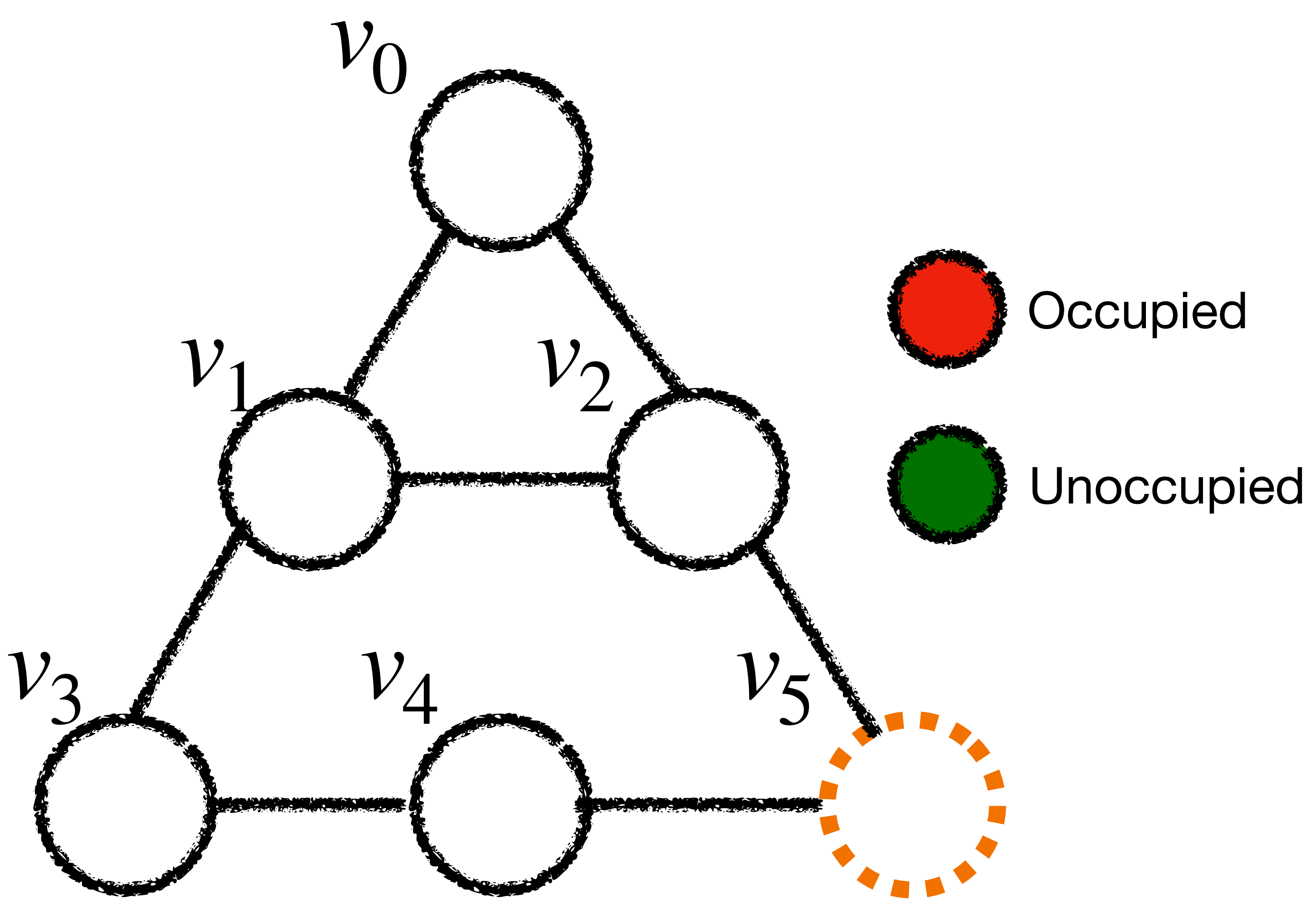
Monte Carlo step of MCMC does not require fully simulating the Markov chains!

- If $r_t < \dfrac{1}{1+\lambda}$, we know v must be updated to unoccupied;
- Otherwise, we need its neighbor's state to determine

A Marginal Sampler from MCMC

Systematic Scan Glauber dynamics

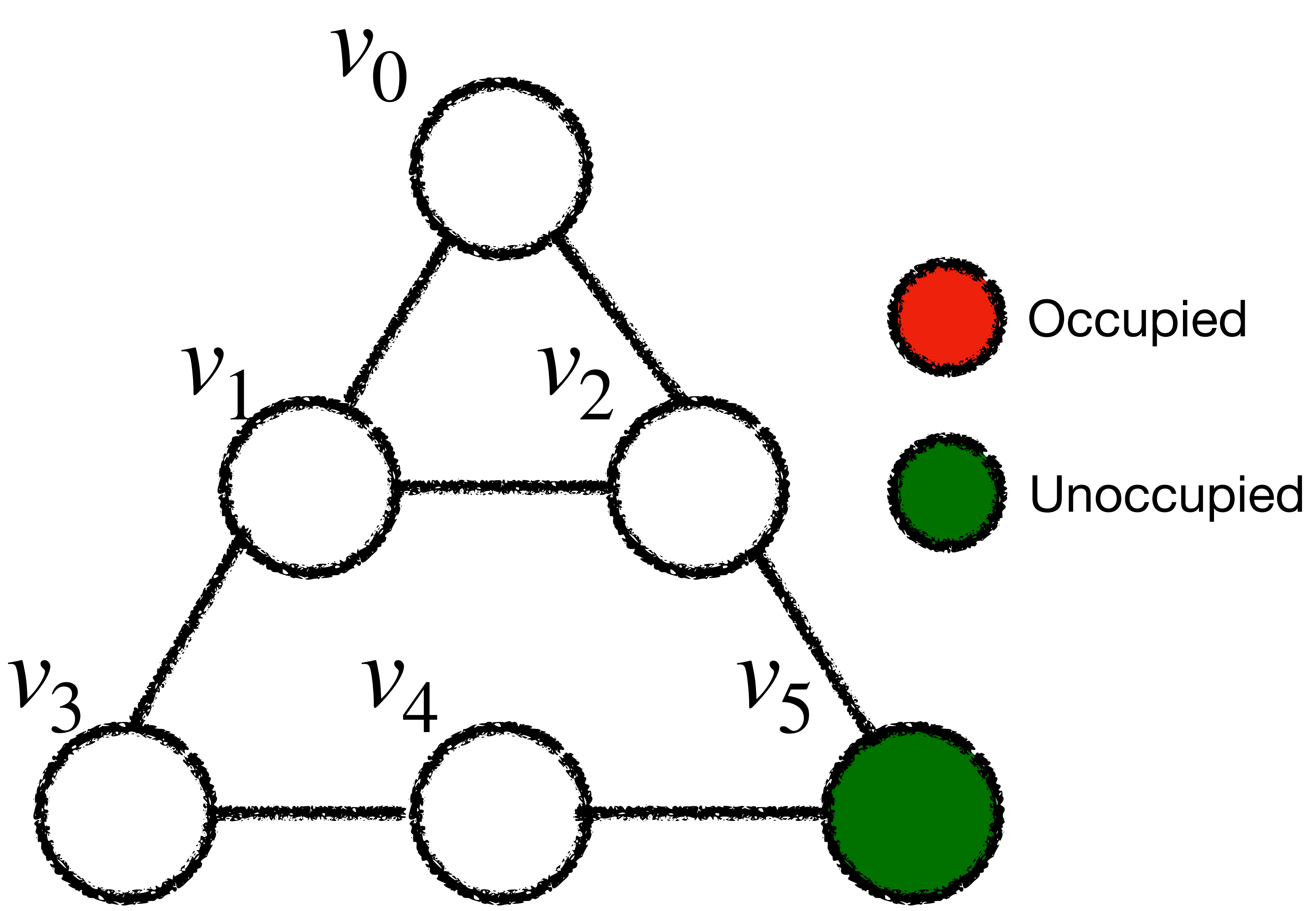
1. Start from any feasible configuration $\sigma \in \Omega$
2. For $t = -(T-1), \ldots, -1, 0$, update the configuration as follows:
   (1) Choose $v \in V$ in the order of $v_0, v_1, \ldots, v_{n-1}, v_0, v_1, \ldots$
   (2) Draw a uniform random real $r_t \in [0,1]$
      a. If there exists any occupied neighbour of v: update v as unoccupied
      b. Otherwise, update v as unoccupied if $r_t < \dfrac{1}{1+\lambda}$; occupied otherwise.



t	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0
v	v_0	v_1	v_2	v_3	v_4	v_5	v_0	v_1	v_2	v_3	v_4	v_5	v_0	v_1	v_2	v_3	v_4	v_5
r_t	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
result	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

A Marginal Sampler from MCMC

Systematic Scan Glauber dynamics



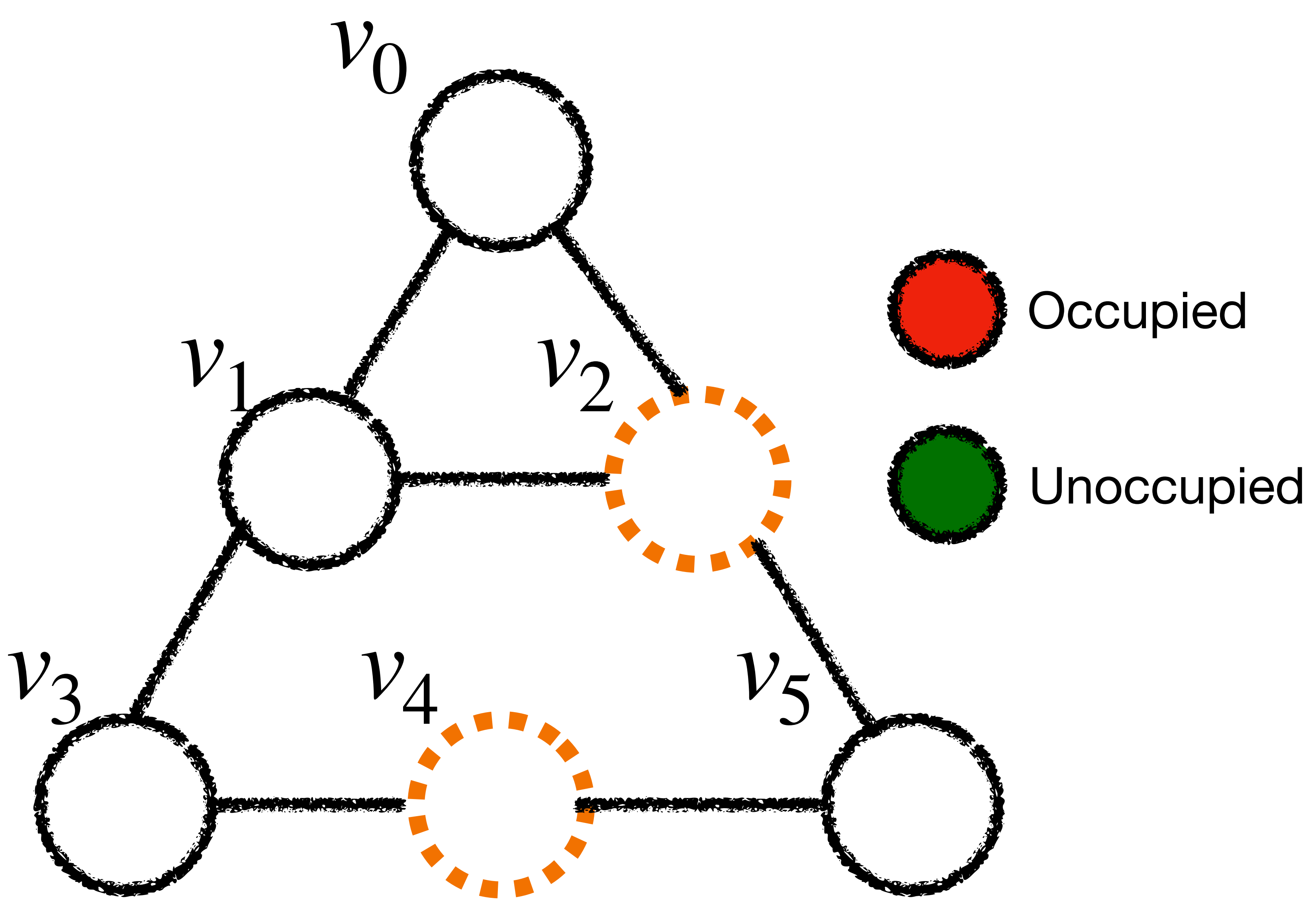
1. Start from any feasible configuration $\sigma \in \Omega$
2. For $t = -(T-1), \ldots, -1, 0$, update the configuration as follows:
   (1) Choose $v \in V$ in the order of $v_0, v_1, \ldots, v_{n-1}, v_0, v_1, \ldots$
   (2) Draw a uniform random real $r_t \in [0,1]$
      a. If there exists any occupied neighbour of v: update v as unoccupied
      b. Otherwise, update v as unoccupied if $r_t < \dfrac{1}{1+\lambda}$; occupied otherwise.

t	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0
v	v_0	v_1	v_2	v_3	v_4	v_5	v_0	v_1	v_2	v_3	v_4	v_5	v_0	v_1	v_2	v_3	v_4	v_5
r_t	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	$<$
result	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	

A Marginal Sampler from MCMC

Systematic Scan Glauber dynamics

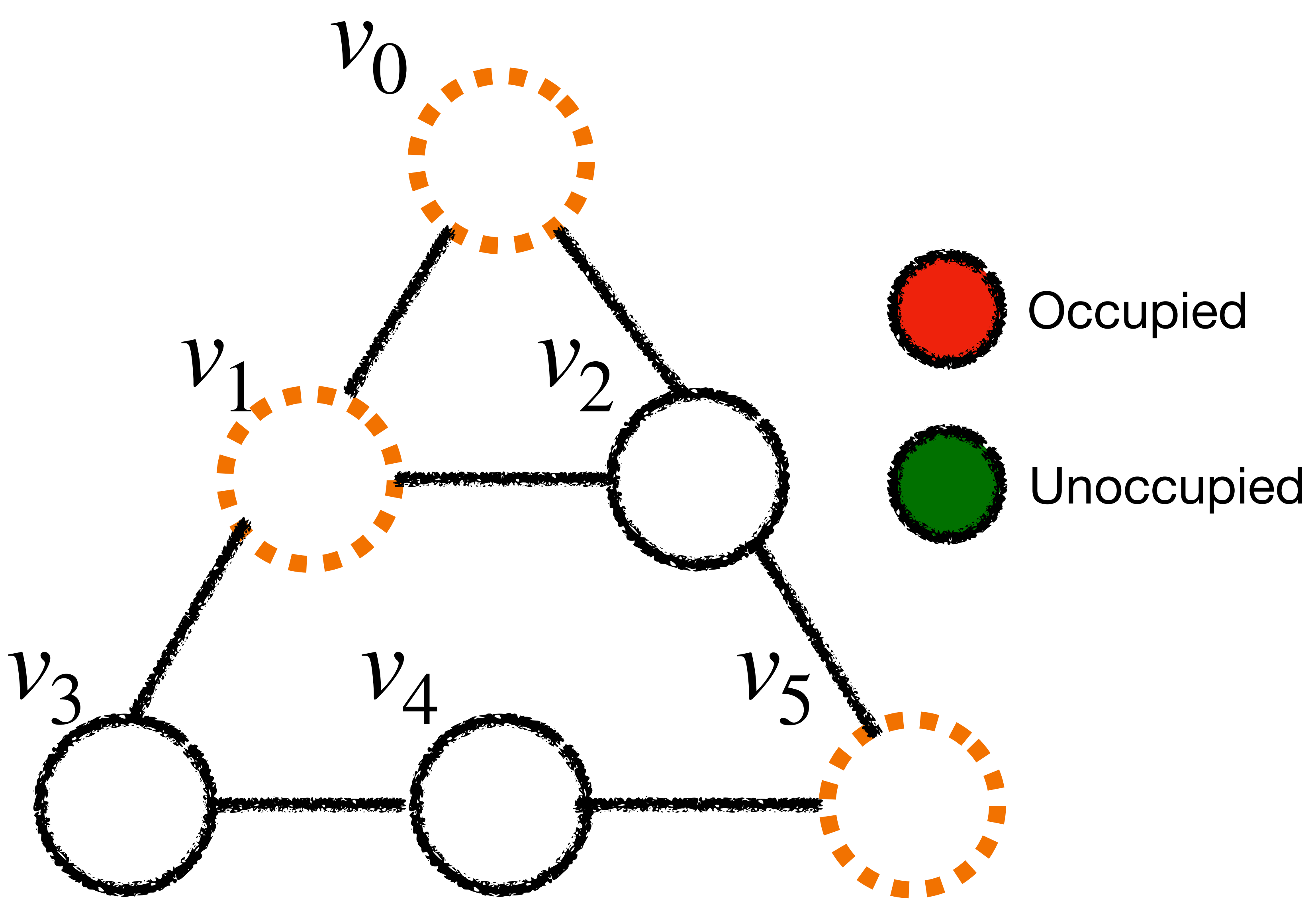
1. Start from any feasible configuration $\sigma \in \Omega$
2. For $t = -(T-1), \dots, -1, 0$, update the configuration as follows:
   (1) Choose $v \in V$ in the order of $v_0, v_1, \dots, v_{n-1}, v_0, v_1, \dots$
   (2) Draw a uniform random real $r_t \in [0,1]$
      a. If there exists any occupied neighbour of v: update v as unoccupied
      b. Otherwise, update v as unoccupied if $r_t < \dfrac{1}{1+\lambda}$; occupied otherwise.



t	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0
v	v_0	v_1	v_2	v_3	v_4	v_5	v_0	v_1	v_2	v_3	v_4	v_5	v_0	v_1	v_2	v_3	v_4	v_5
r_t	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	$\geq$
result	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

A Marginal Sampler from MCMC

Systematic Scan Glauber dynamics



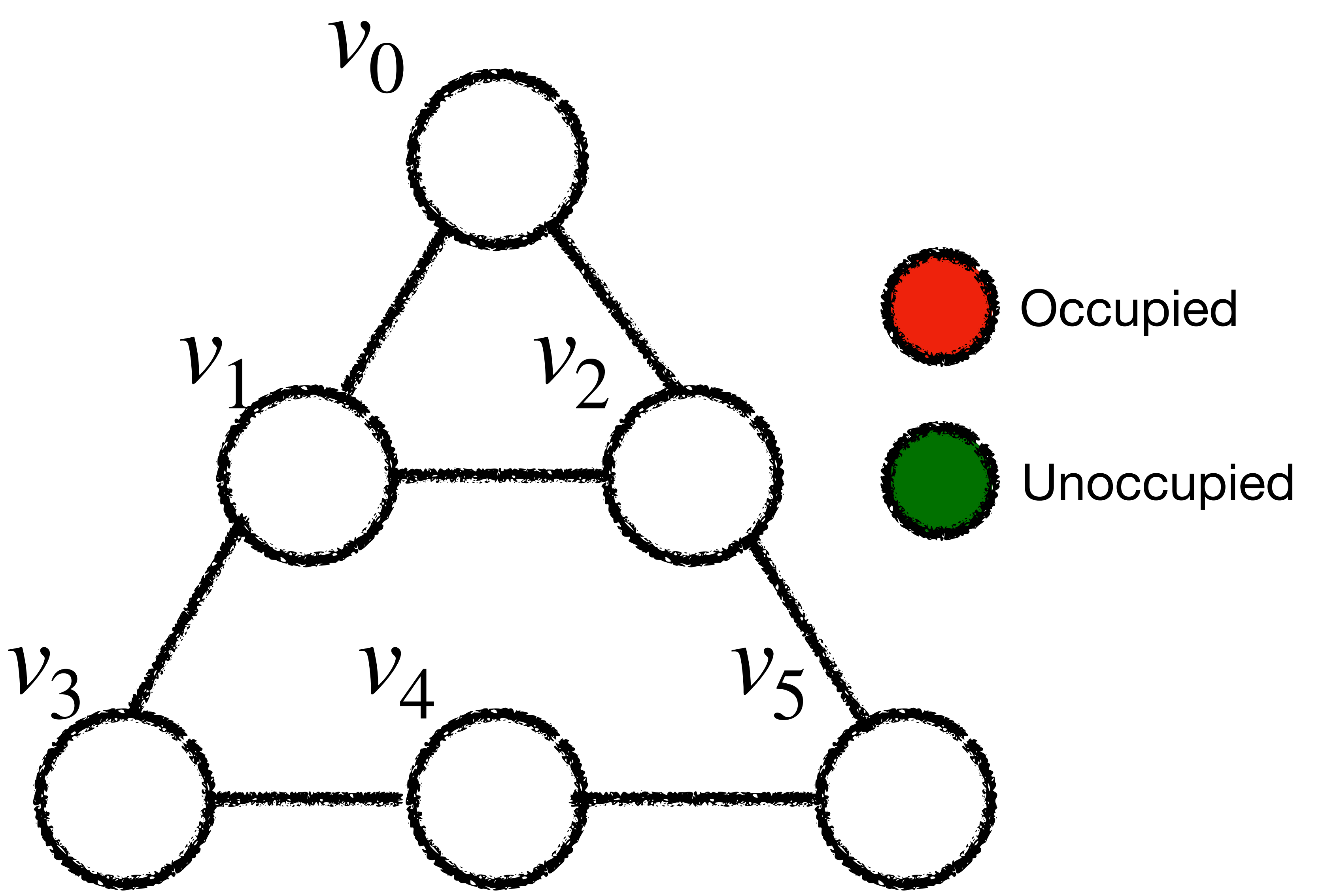
1. Start from any feasible configuration $\sigma \in \Omega$
2. For $t = -(T-1), \ldots, -1, 0$, update the configuration as follows:
   (1) Choose $v \in V$ in the order of $v_0, v_1, \ldots, v_{n-1}, v_0, v_1, \ldots$
   (2) Draw a uniform random real $r_t \in [0,1]$
      a. If there exists any occupied neighbour of v: update v as unoccupied
      b. Otherwise, update v as unoccupied if $r_t < \dfrac{1}{1+\lambda}$; occupied otherwise.

t	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0
v	v_0	v_1	v_2	v_3	v_4	v_5	v_0	v_1	v_2	v_3	v_4	v_5	v_0	v_1	v_2	v_3	v_4	v_5
r_t	?	?	?	?	?	?	?	?	?	?	?	?	?	?	$\geq$	?	$<$	$\geq$
result	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		?

A Marginal Sampler from MCMC

Systematic Scan Glauber dynamics

1. Start from any feasible configuration $\sigma \in \Omega$
2. For $t = -(T-1), \ldots, -1, 0$, update the configuration as follows:
   (1) Choose $v \in V$ in the order of $v_0, v_1, \ldots, v_{n-1}, v_0, v_1, \ldots$
   (2) Draw a uniform random real $r_t \in [0,1]$
      a. If there exists any occupied neighbour of v: update v as unoccupied
      b. Otherwise, update v as unoccupied if $r_t < \dfrac{1}{1+\lambda}$; occupied otherwise.

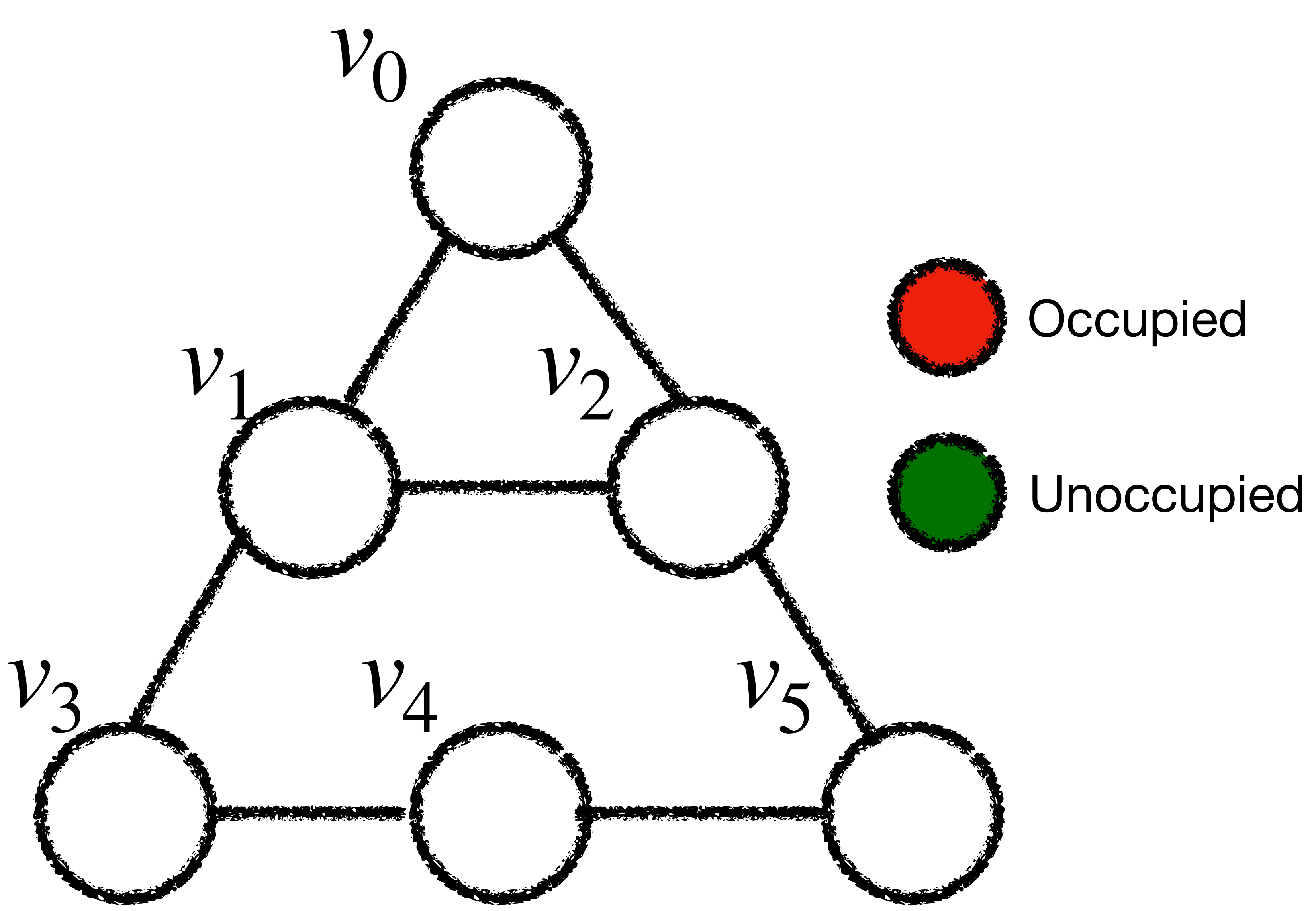


t	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0
v	v_0	v_1	v_2	v_3	v_4	v_5	v_0	v_1	v_2	v_3	v_4	v_5	v_0	v_1	v_2	v_3	v_4	v_5
r_t	?	?	?	?	?	?	?	?	?	?	?	$<$	$<$	$<$	$\geq$	?	$<$	$\geq$
result	?	?	?	?	?	?	?	?	?	?	?				?	?		?

A Marginal Sampler from MCMC

Systematic Scan Glauber dynamics

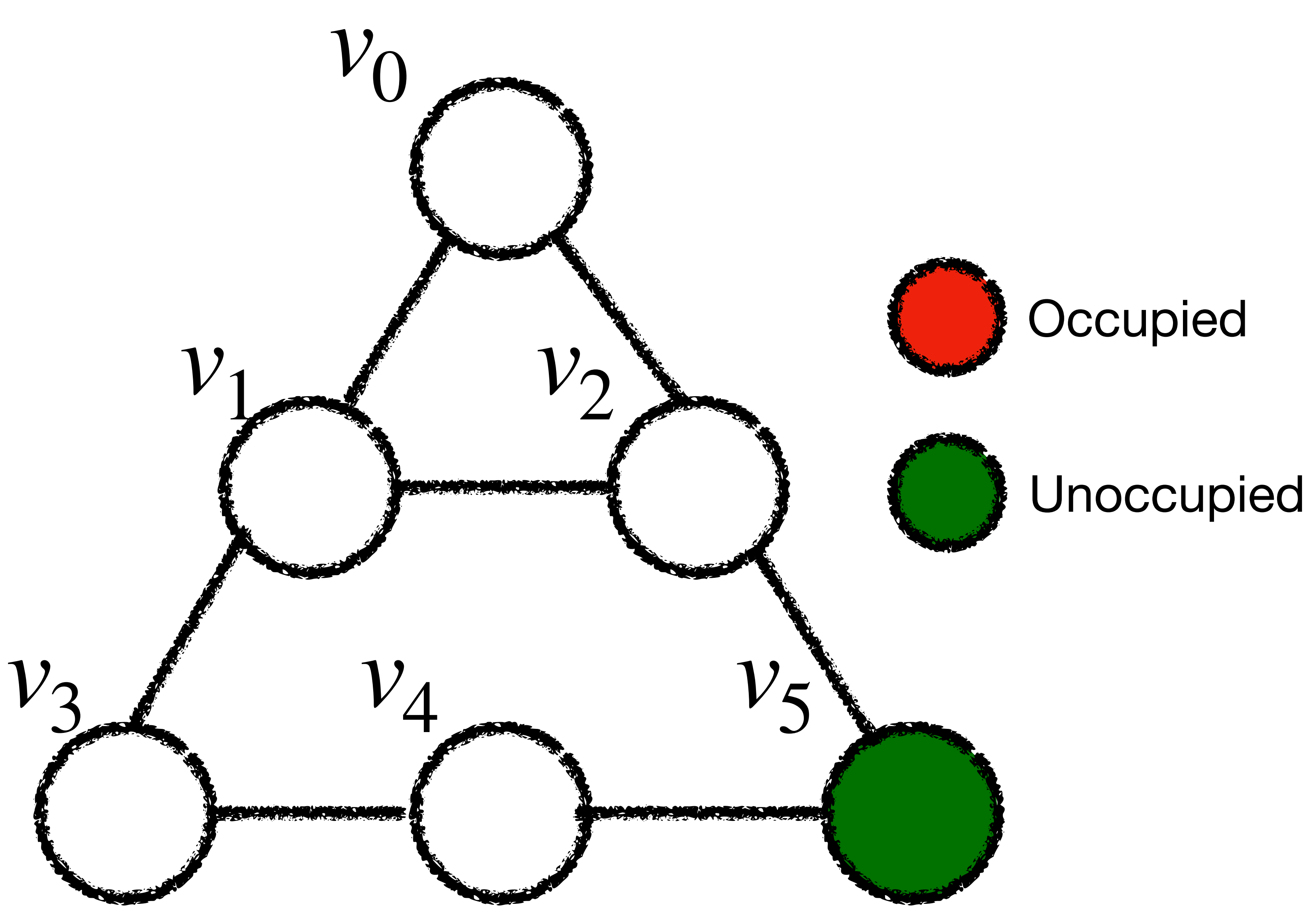
1. Start from any feasible configuration $\sigma \in \Omega$
2. For $t = -(T-1), \ldots, -1, 0$, update the configuration as follows:
   (1) Choose $v \in V$ in the order of $v_0, v_1, \ldots, v_{n-1}, v_0, v_1, \ldots$
   (2) Draw a uniform random real $r_t \in [0,1]$
      a. If there exists any occupied neighbour of v: update v as unoccupied
      b. Otherwise, update v as unoccupied if $r_t < \dfrac{1}{1+\lambda}$; occupied otherwise.



t	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0
v	v_0	v_1	v_2	v_3	v_4	v_5	v_0	v_1	v_2	v_3	v_4	v_5	v_0	v_1	v_2	v_3	v_4	v_5
r_t	?	?	?	?	?	?	?	?	?	?	?	$<$	$<$	$<$	$\geq$	?	$<$	$\geq$
result	?	?	?	?	?	?	?	?	?	?	?					?		?

A Marginal Sampler from MCMC

Systematic Scan Glauber dynamics



1. Start from any feasible configuration $\sigma \in \Omega$
2. For $t = -(T-1), \ldots, -1, 0$, update the configuration as follows:

   (1) Choose $v \in V$ in the order of $v_0, v_1, \ldots, v_{n-1}, v_0, v_1, \ldots$

   (2) Draw a uniform random real $r_t \in [0,1]$

   a. If there exists any occupied neighbour of v: update v as unoccupied

   b. Otherwise, update v as unoccupied if $r_t < \dfrac{1}{1+\lambda}$; occupied otherwise.

t	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0
v	v_0	v_1	v_2	v_3	v_4	v_5	v_0	v_1	v_2	v_3	v_4	v_5	v_0	v_1	v_2	v_3	v_4	v_5
r_t	?	?	?	?	?	?	?	?	?	?	?	$<$	$<$	$<$	$\geq$	?	$<$	$\geq$
result	?	?	?	?	?	?	?	?	?	?	?					?		

Coupling Towards The Past

(outcome of $v = v_{t \bmod n}$ at time t)

$\mathrm{Resolve}_T(t)$

$t \leq -T$

return according to Initial state

$t > -T$

(Stored in a map, drawn uniformly from $[0,1]$ at first visit)

visit r_t

$r_t < \frac{1}{1+\lambda}$

return unoccupied

$r_t \geq \frac{1}{1+\lambda}$

$\mathrm{pred}_u(t)$: last visit time of u before time t

recursively call $\mathrm{Resolve}_T(\mathrm{pred}_u(t))$ for each neighbour u of v;

any return occupied

return unoccupied

none return occupied

return occupied

An approximate marginal sampler!

Beyond hardcore model: marginal lower bounds

Hidden grand coupling:

the same set of $\{r_t\}$ is used.

Coupling Towards The Past

(outcome of $v = v_{t \bmod n}$ at time t)

$\mathrm{Resolve}_\infty(t)$

terminates if $\lambda < \dfrac{1}{\Delta - 1}$

$r_t < \dfrac{1}{1+\lambda}$

return unoccupied

$r_t \geq \dfrac{1}{1+\lambda}$

recursively call $\mathrm{Resolve}_\infty(\mathrm{pred}_u(t))$
for each neighbour u of v;

$\mathrm{pred}_u(t)$: last visit time of u before time t

any return occupied

return unoccupied

none return occupied

return occupied

A perfect marginal sampler!

When $\Pr[t_{\text{run}} \geq T] \leq \exp(-O_\Delta(T))$,

truncate up to $K = O_\Delta\left(\log \dfrac{n}{\varepsilon}\right)$ bits gives up to $\dfrac{\varepsilon}{n}$ bias.

Trivial derandomisation by enumeration!

Simulating a Markov chain: $O(n \log n)$ random bits

Simulating a single marginal: $O(\log n)$ random bits

Direct-sum style decomposition!

Related concepts

Lazy Depth-First Search Sampler [Anand, Jerrum '22]

A main source of inspiration for our work

Similarities:

- both give perfect marginal samplers with possibly logarithmic number of random bits
- both utilize the "marginal lower bounds" for early termination of the sampler

Distinctions:

- Our CTTP result comes from MCMC, while the AJ algorithm relies on spatial mixing properties.
- AJ algorithm encounters some difficulty in matching the state-of-art bounds for some randomised algorithms.

Coupling From The Past [Propp, Wilson '96]

Similarities:

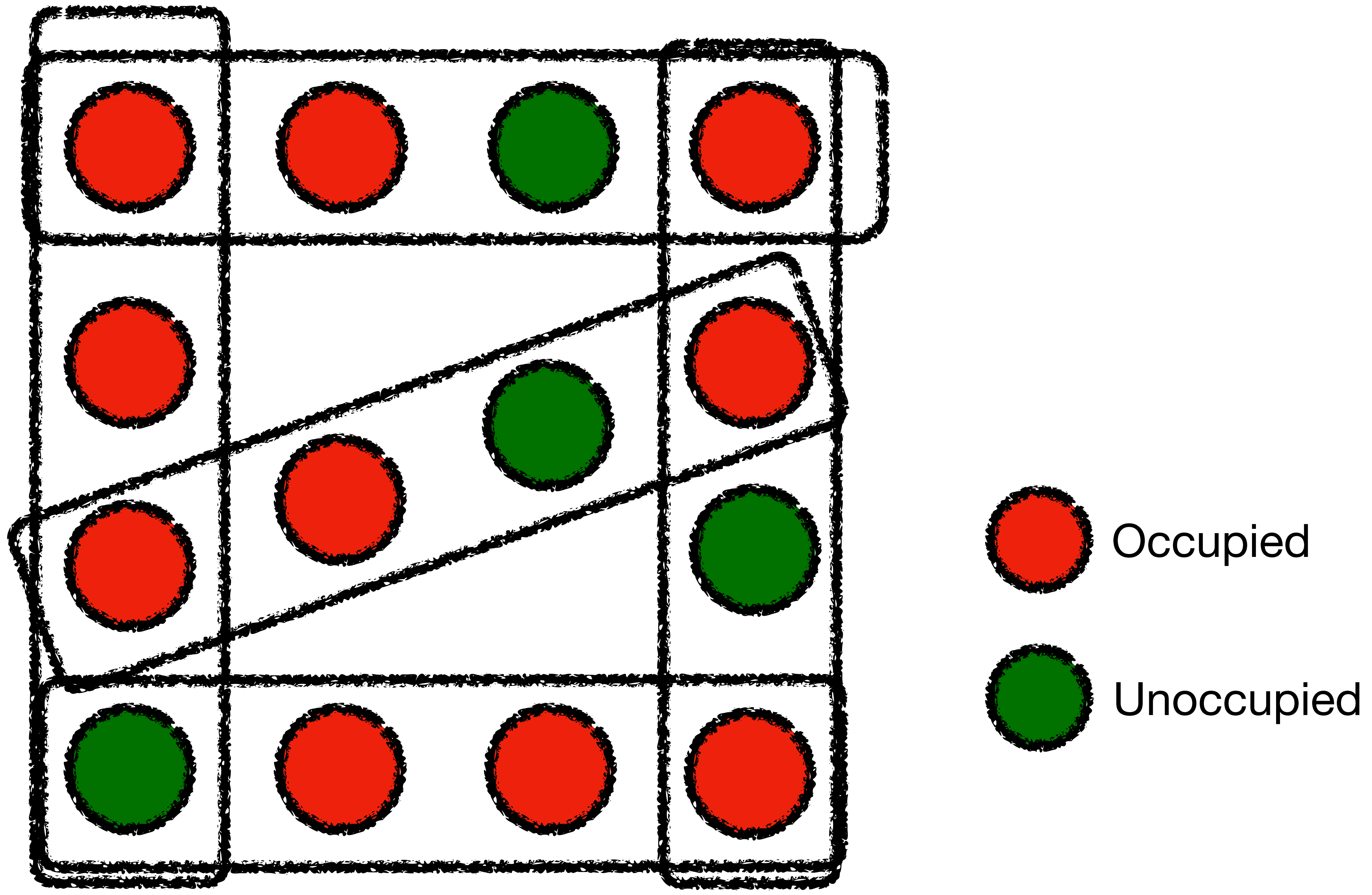
- both give perfect samplers from MCMC
- both run backwards in time and have underlying grand couplings

Distinction:

- CFTP needs to sequentially simulate the evolution of the whole state, which requires at least a linear number of random bits, while CTTP only needs logarithmic number of random bits under suitable conditions.

Hypergraph independent sets

Let $H = (V, \mathscr{E})$ be a k-uniform hypergraph with max. deg. Δ
A set $S \subseteq V$ is independent if $S \cap e \neq e$ for all $e \in \mathscr{E}$



[Hermon, Sly, Zhang '19]: $\Delta \leq c2^{k/2}$, Glauber dynamics

[Qiu, Wang, Zhang '22]: $\Delta \leq \frac{c}{k}2^{k/2}$, perfect sampler

[He, W., Yin '23]: $\Delta \lesssim 2^{k/5}$, FPTAS

[Bezáková, Galanis, Goldberg, Guo, Štefankovič '23]: $\Delta \geq 5 \cdot 2^{k/2}$, **NP-hard**

Our result for HIS

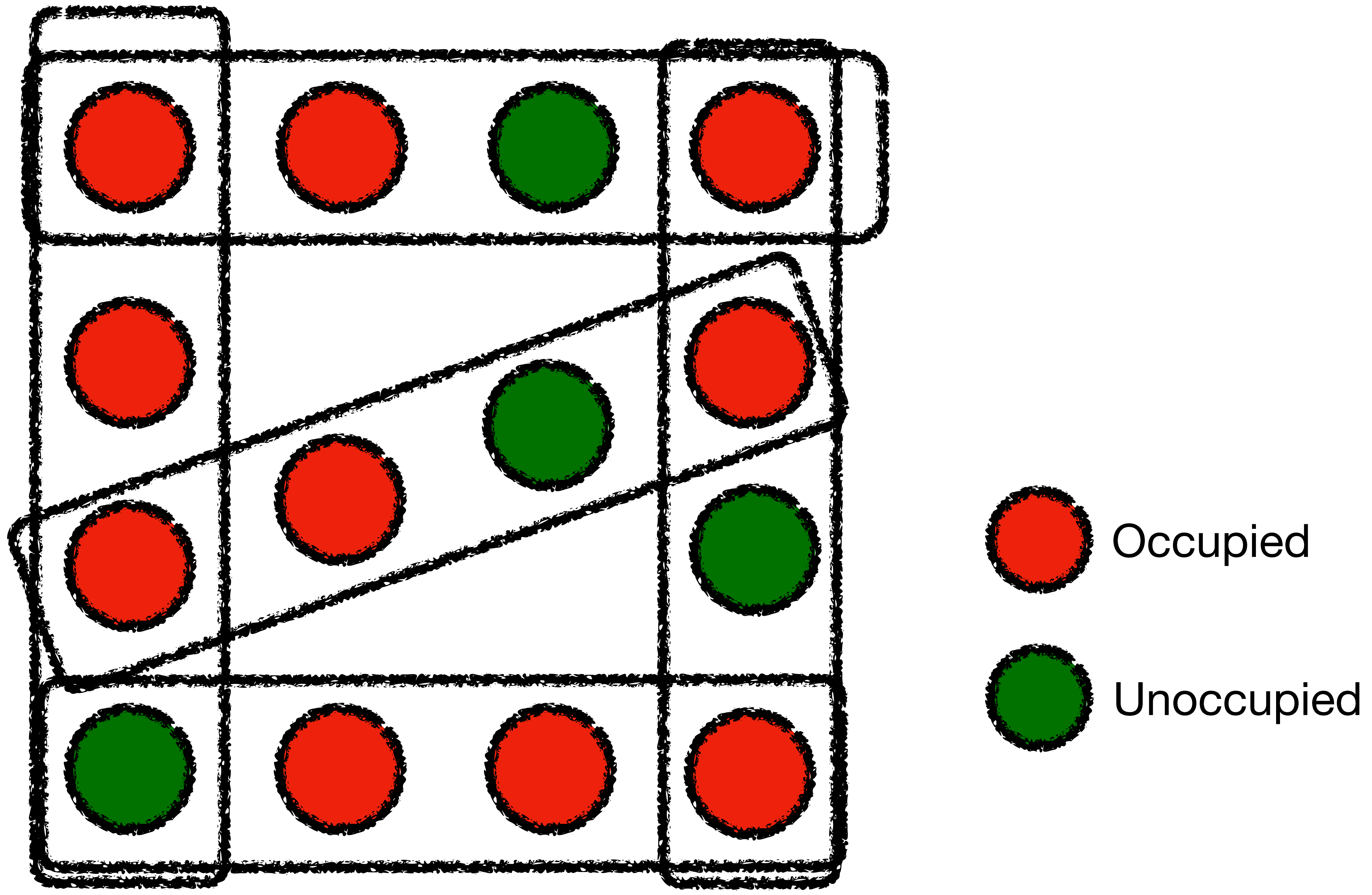
Let $k \geq 2$ and $\Delta \geq 2$ be two integers such that $\Delta \leq \frac{1}{\sqrt{8ek^2}} \cdot 2^{k/2}$.

There is an FPTAS for the number of independent sets in k-uniform hypergraphs with maximum degree Δ.

CTTP for HIS

We apply CTTP on the systematic scan GD for HIS.

Each vertex has a $\frac{1}{2}$ lower bound for "unoccupied".

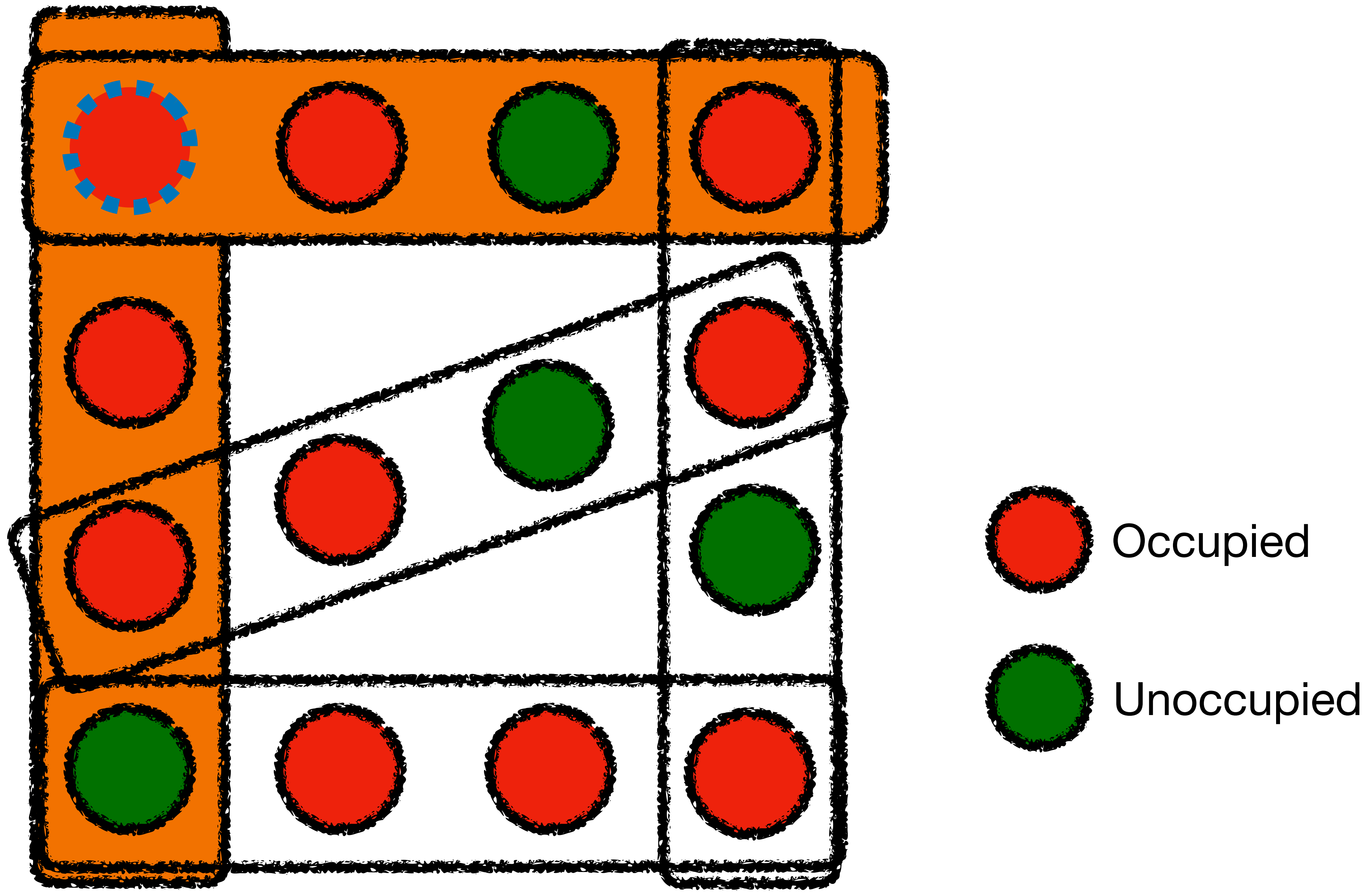


CTTP for HIS

We apply CTTP on the systematic scan GD for HIS.

Each vertex has a $\frac{1}{2}$ lower bound for "unoccupied".

When failing to determine using the lower bound,
we need to resolve the states of its neighborhood.

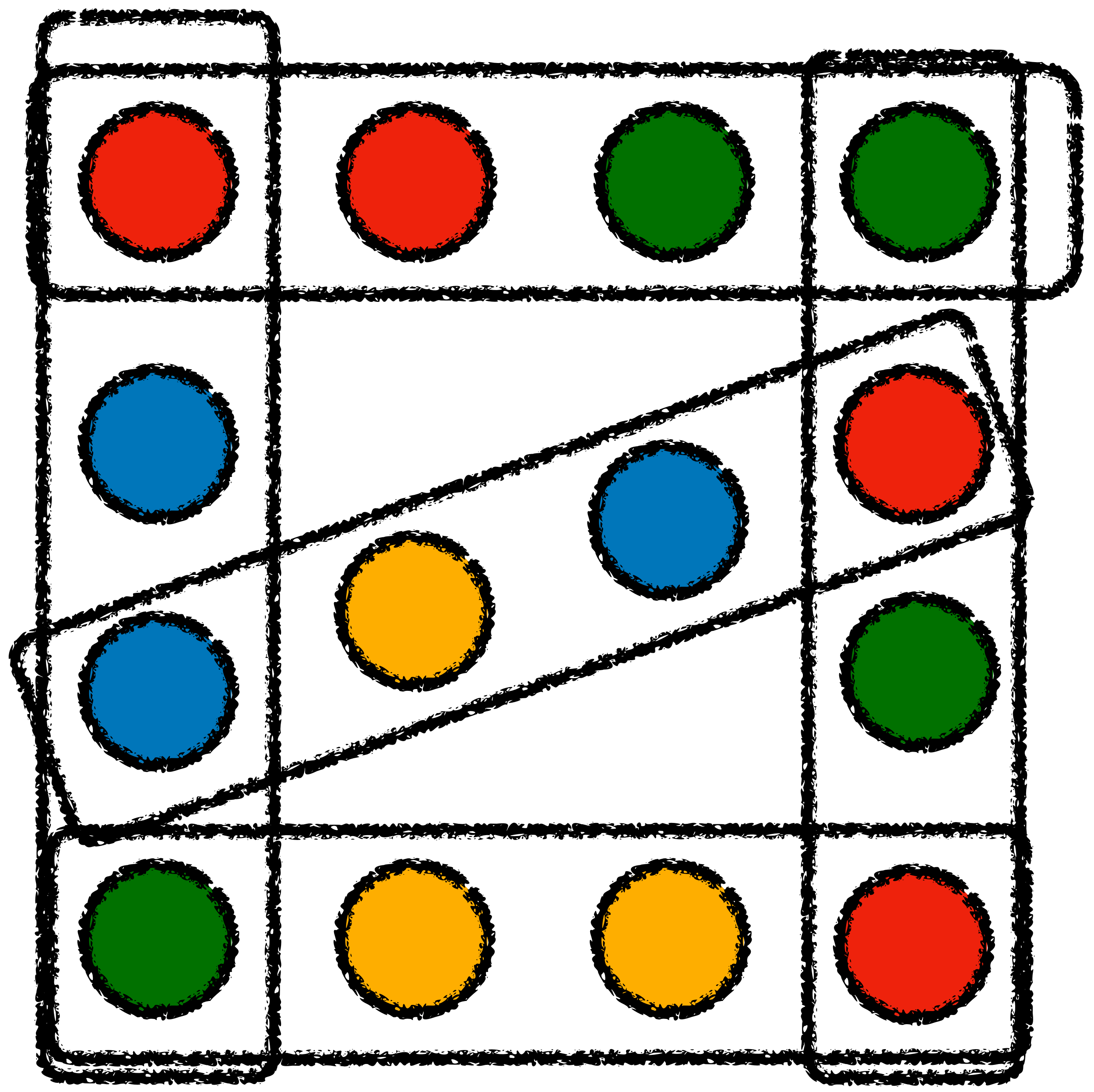


Direct recursion: Need $k\Delta < 2$ even for CTTP to terminate!

A more clever strategy: Recurse for a neighbouring hyperedge $e \in \mathscr{E}$ only if $r_{\mathrm{pred}_t(u)} \geq \frac{1}{2}$ for all $u \in e$

Hypergraph colourings

Let $H = (V, \mathscr{E})$ be a k-uniform hypergraph with max. deg. Δ
A q-colouring $\sigma \subseteq [q]^V$ is proper if no hyperedge is monochromatic



[Jain, Pham, and Vuong '21]: $\Delta \lesssim q^{k/3}$, compression+MCMC

[He, Sun, Wu '22]: $\Delta \lesssim q^{k/3}$, perfect sampler

[He, W., Yin '23]: $\Delta \lesssim q^{k/5}$, FPTAS

[Galanis, Guo, Wang '22]: $\Delta \geq 5 \cdot q^{k/2}$, even q, **NP-hard**

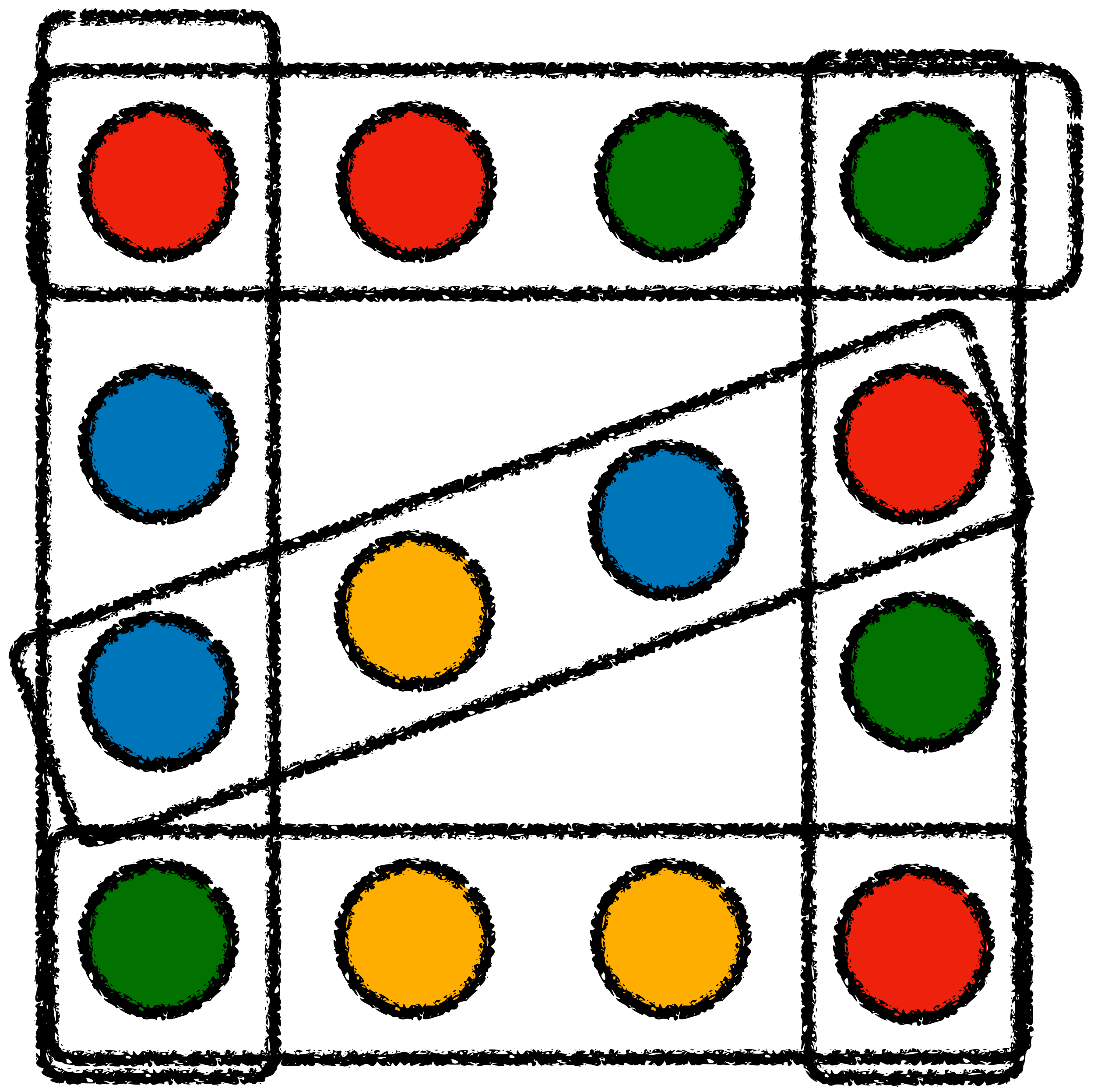
Our result for HC

Let $k \geq 20$ and $\Delta \geq 2$ be two integers such that $\Delta \leq \left(\frac{q}{64}\right)^{(k-5)/3}$.

There is an FPTAS for the number of proper q-colourings in k-uniform hypergraphs with maximum degree Δ.

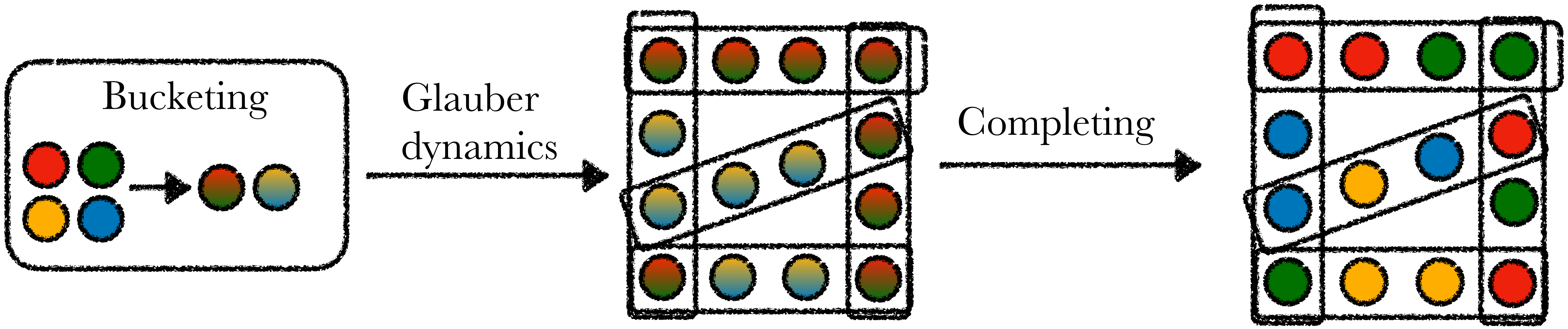
Compression + MCMC for HC

The natural Glauber dynamics for HC is not irreducible.



We need the idea of compression + MCMC
[Feng, Guo, Yin, Zhang ’21], [Feng, He, Yin ’21]

Compression: Divide the q colors into buckets of sizes s.
The sampling algorithm
1. Decides the bucket of each vertex (using Glauber dynamics)
2. Decides the final color, conditioning on the bucketing



CTTP for HC

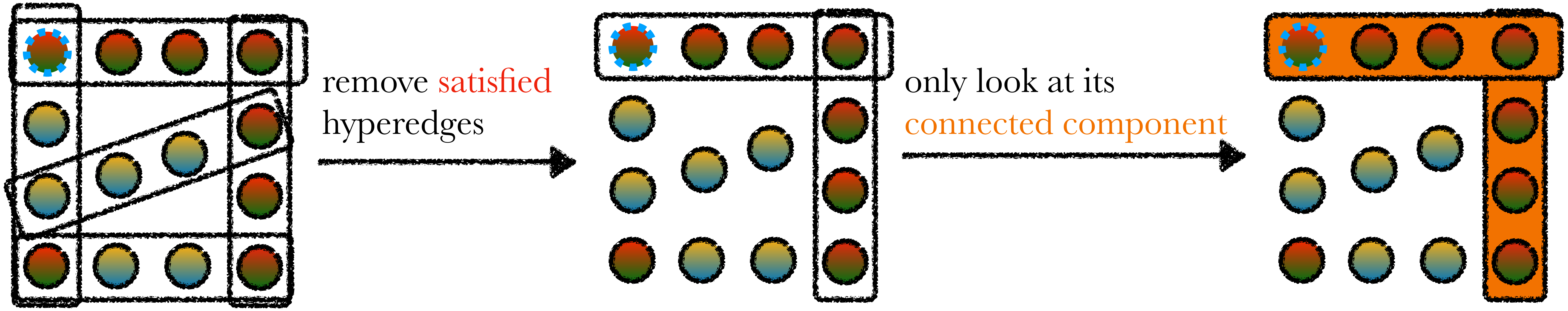
Local Uniformity [Erdos, Lovász '75], [Haeupler, Saha, Srinivasan '11]

If $\lfloor q/s \rfloor^k \geq 4eqsk\Delta$, then

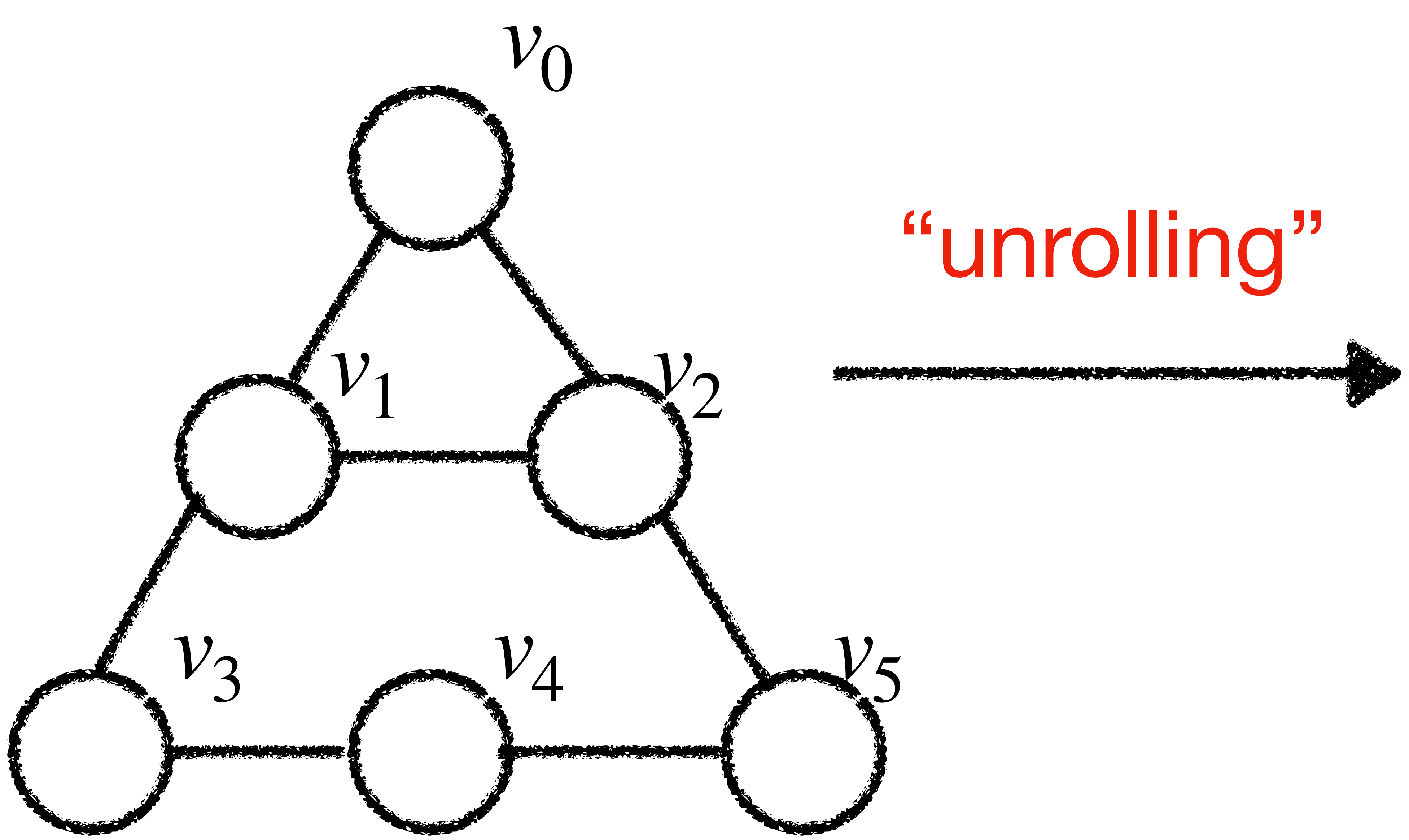
$$\frac{1}{s}\left(1-\frac{1}{4s}\right) \leq \text{marginal of any bucket under arbitrary pinning (of buckets)} \leq \frac{1}{s}\left(1+\frac{1}{4s}\right)$$

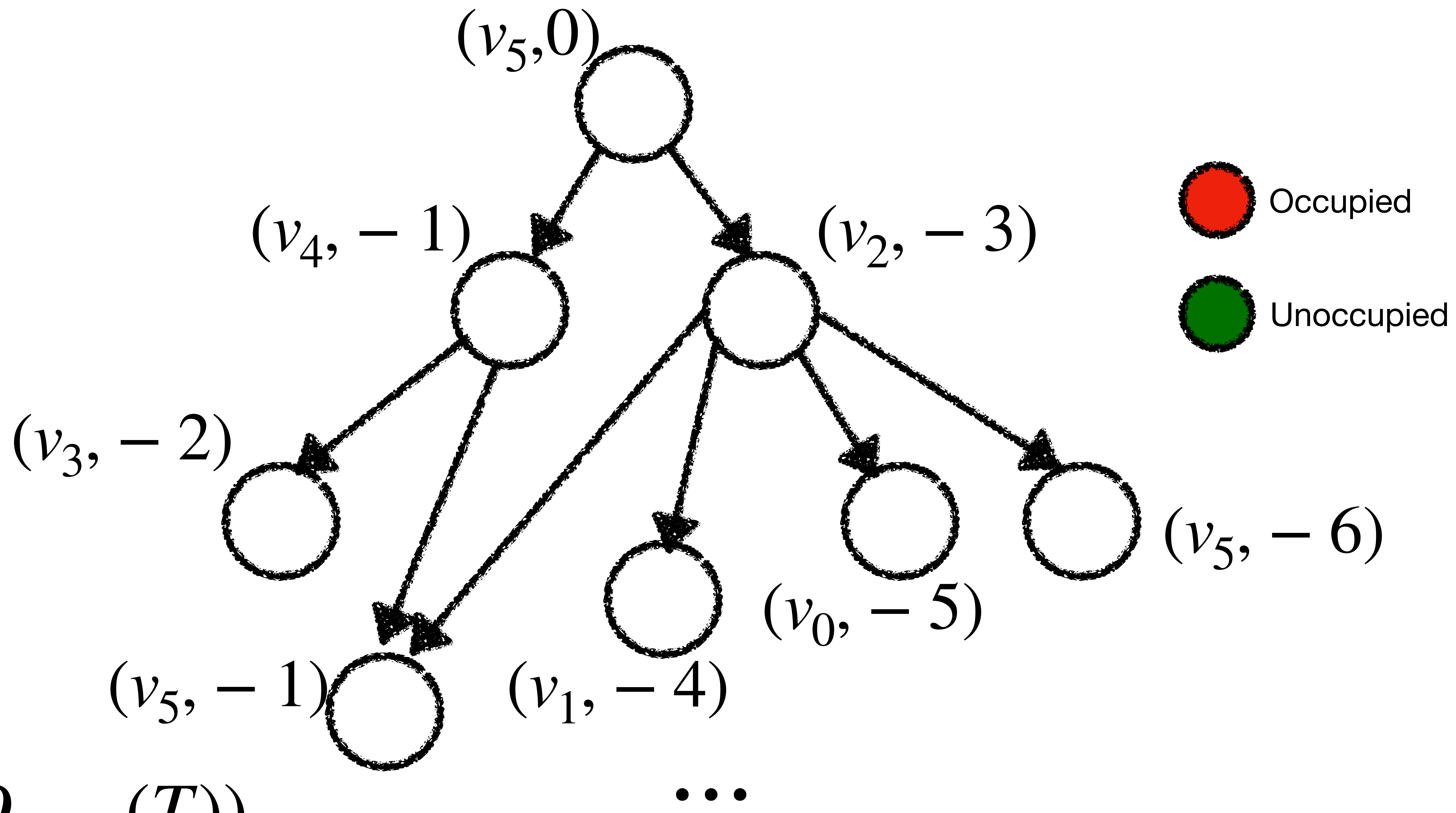
Complication for HC: no longer a Gibbs distribution after bucketing

When failing to determine using the lower bound,
we need to resolve the states of its connected component.



Analysis of the truncation error

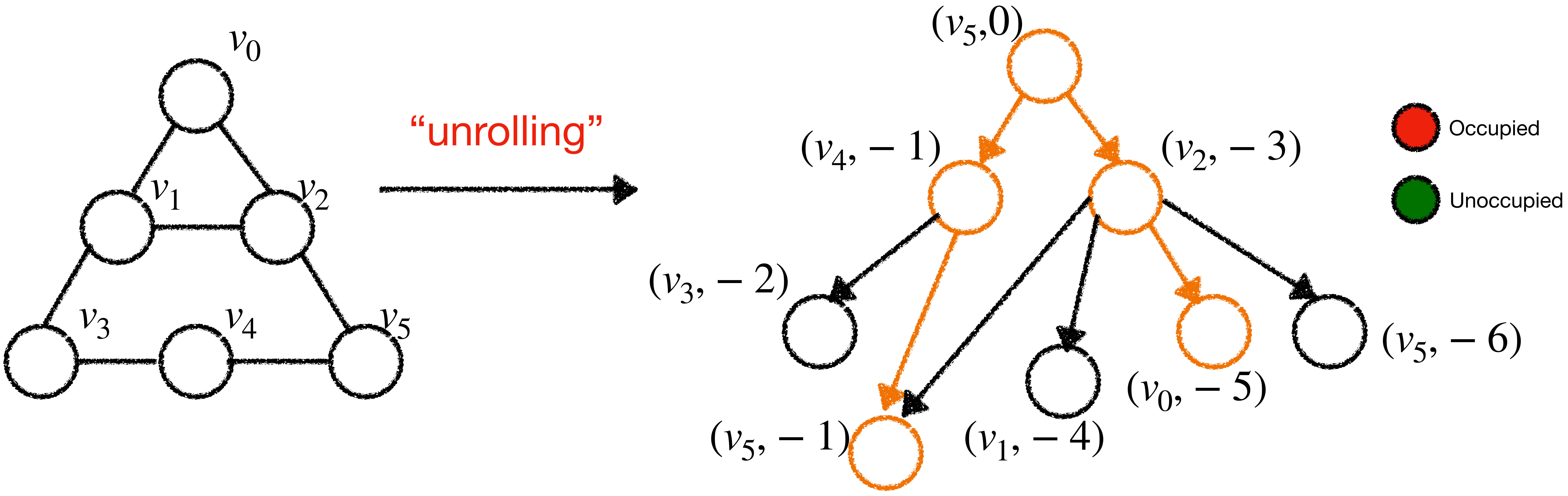




Goal: Show $\Pr[t_{\mathrm{run}} \geq T] \leq \exp(-O_{k,q,\Delta}(T))$

Time-space (hyper)graph
[Hermon, Sly, Zhang' 19]
[Jain, Pham, Vuong' 21]
[He, Sun, Wu '21]

Analysis of the truncation error



Goal: Show $\Pr[t_{\text{run}} \geq T] \leq \exp(-O_{k,q,\Delta}(T))$

…

Time-space (hyper)graph

[Hermon, Sly, Zhang’ 19]

[Jain, Pham, Vuong’ 21]

[He, Sun, Wu ’21]

witness argument + union bound

Derandomising random scan

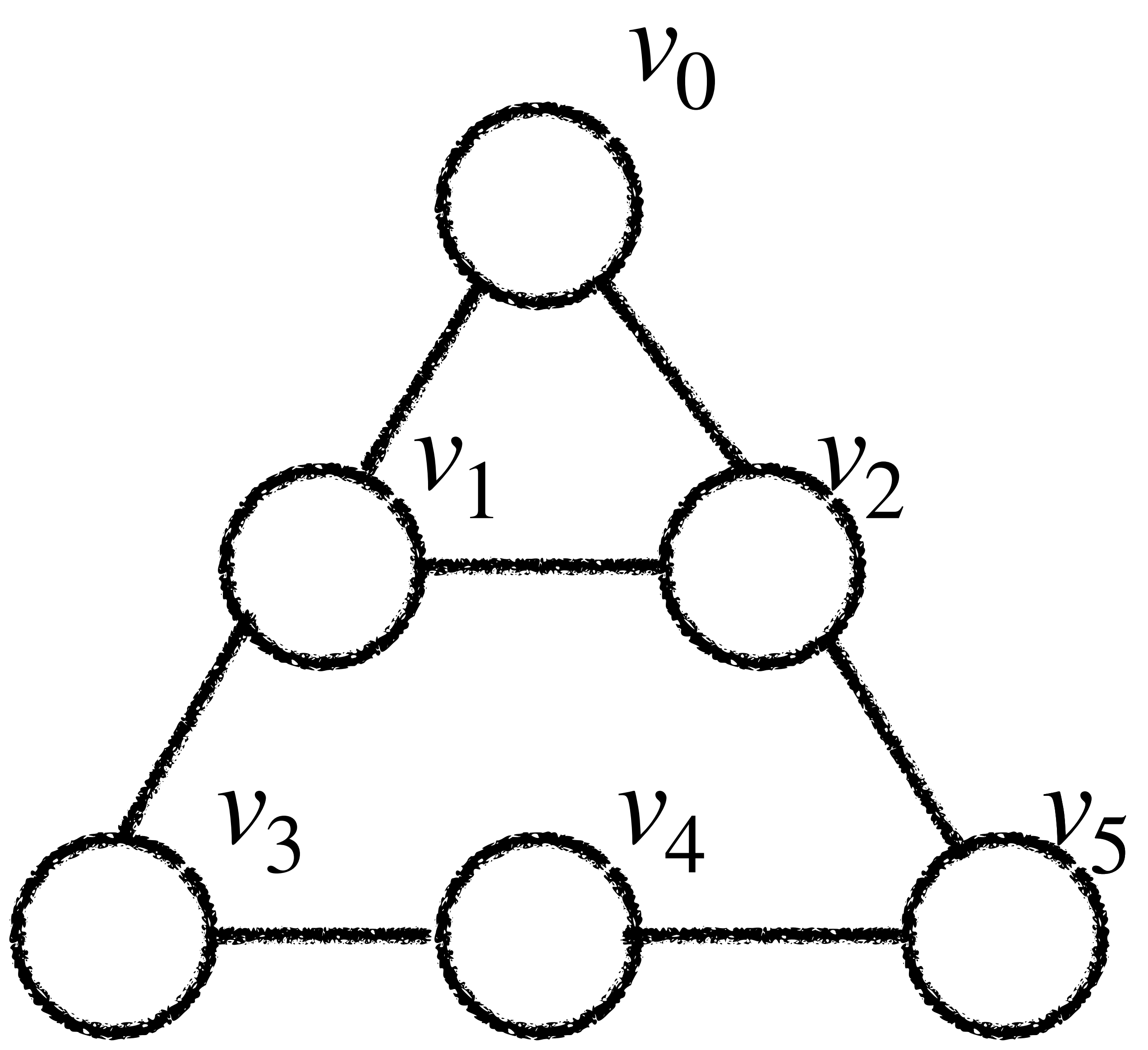
Random scan: each entry of the scan sequence is chosen for V u.a.r.

Enumerate all possible visited scan sequences within $K = O(\log n)$ random bits?

Construct witness tree

$v_2, v_5, v_1, v_0, v_2, v_3, v_4, v_0, v_5$ $\longrightarrow$

Go backwards in time, each time append to neighbour with largest depth



Derandomising random scan

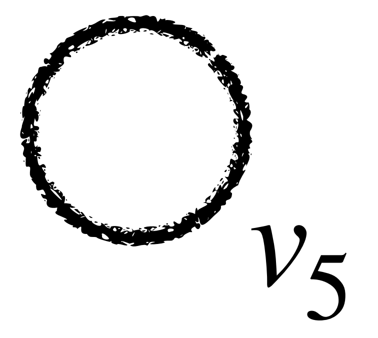
Random scan: each entry of the scan sequence is chosen for V u.a.r.

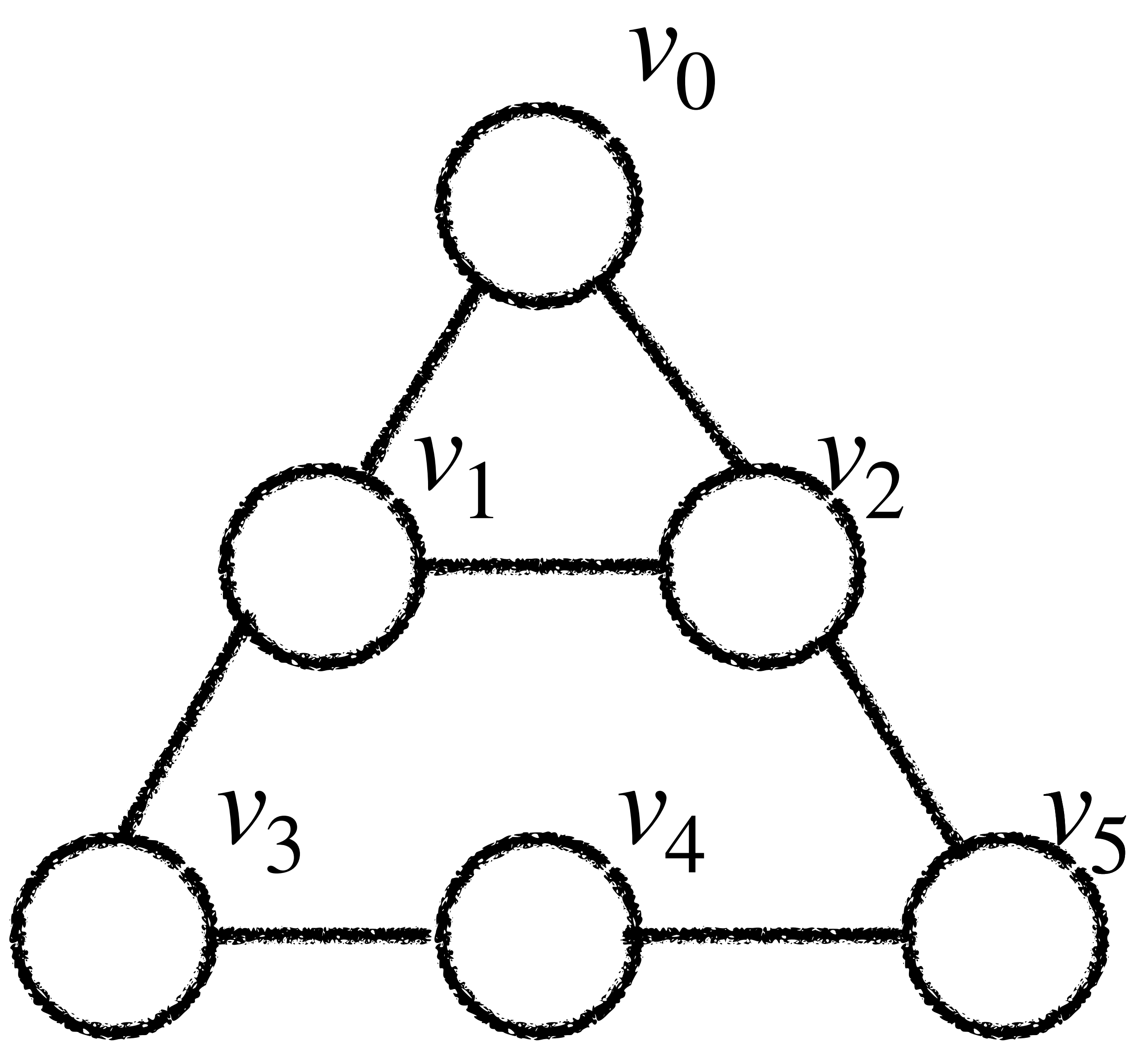
Enumerate all possible visited scan sequences within $K = O(\log n)$ random bits?

Construct witness tree

$v_2, v_5, v_1, v_0, v_2, v_3, v_4, v_0, v_5$

Go backwards in time, each time append to neighbour with largest depth





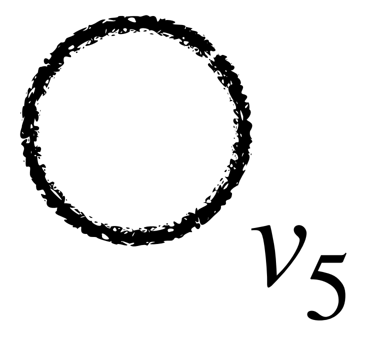
Derandomising random scan

Random scan: each entry of the scan sequence is chosen for V u.a.r.

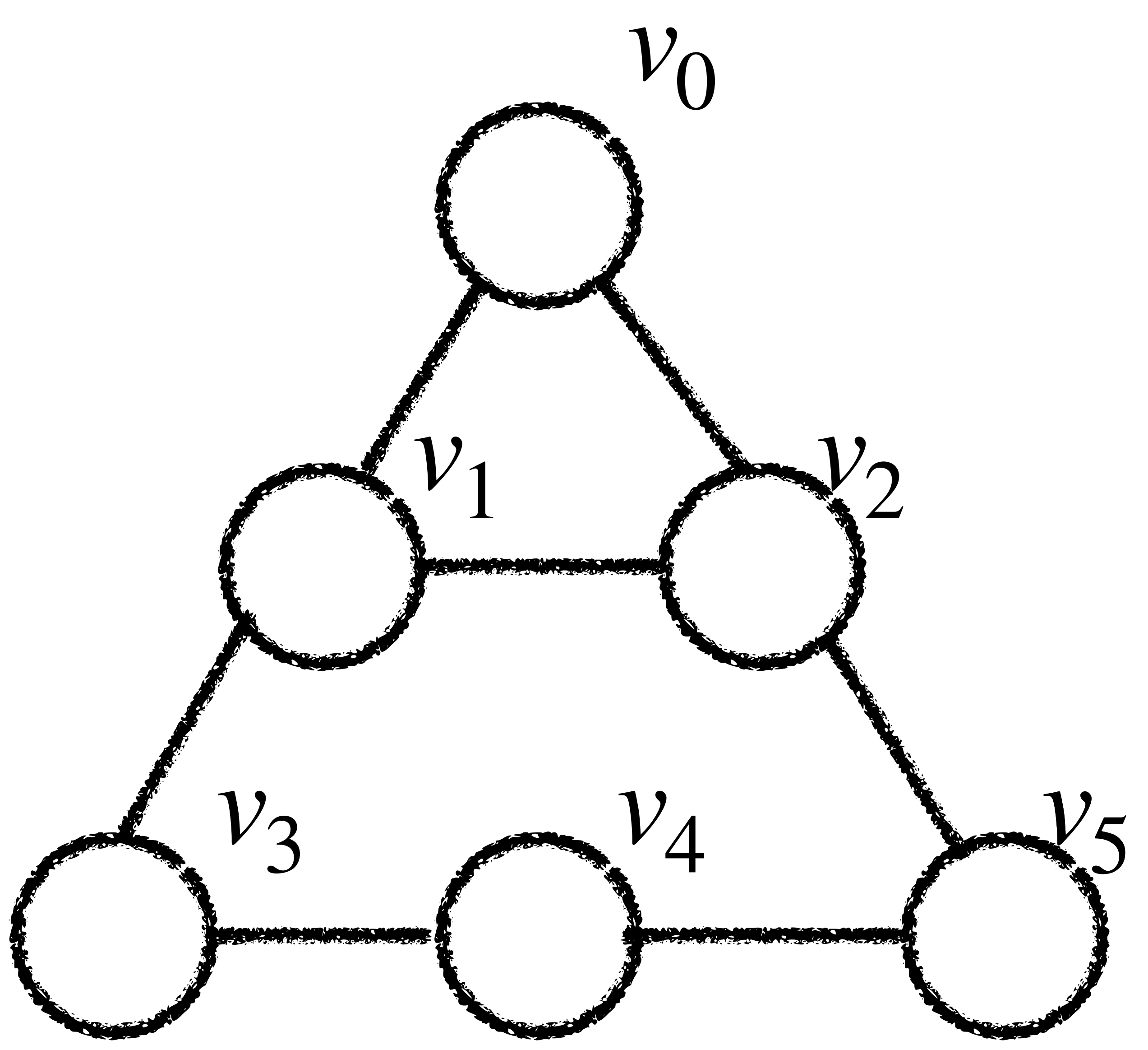
Enumerate all possible visited scan sequences within $K = O(\log n)$ random bits?

$v_2, v_5, v_1, v_0, v_2, v_3, v_4, v_0, v_5$

Construct witness tree



Go backwards in time, each time append to neighbour with largest depth



Derandomising random scan

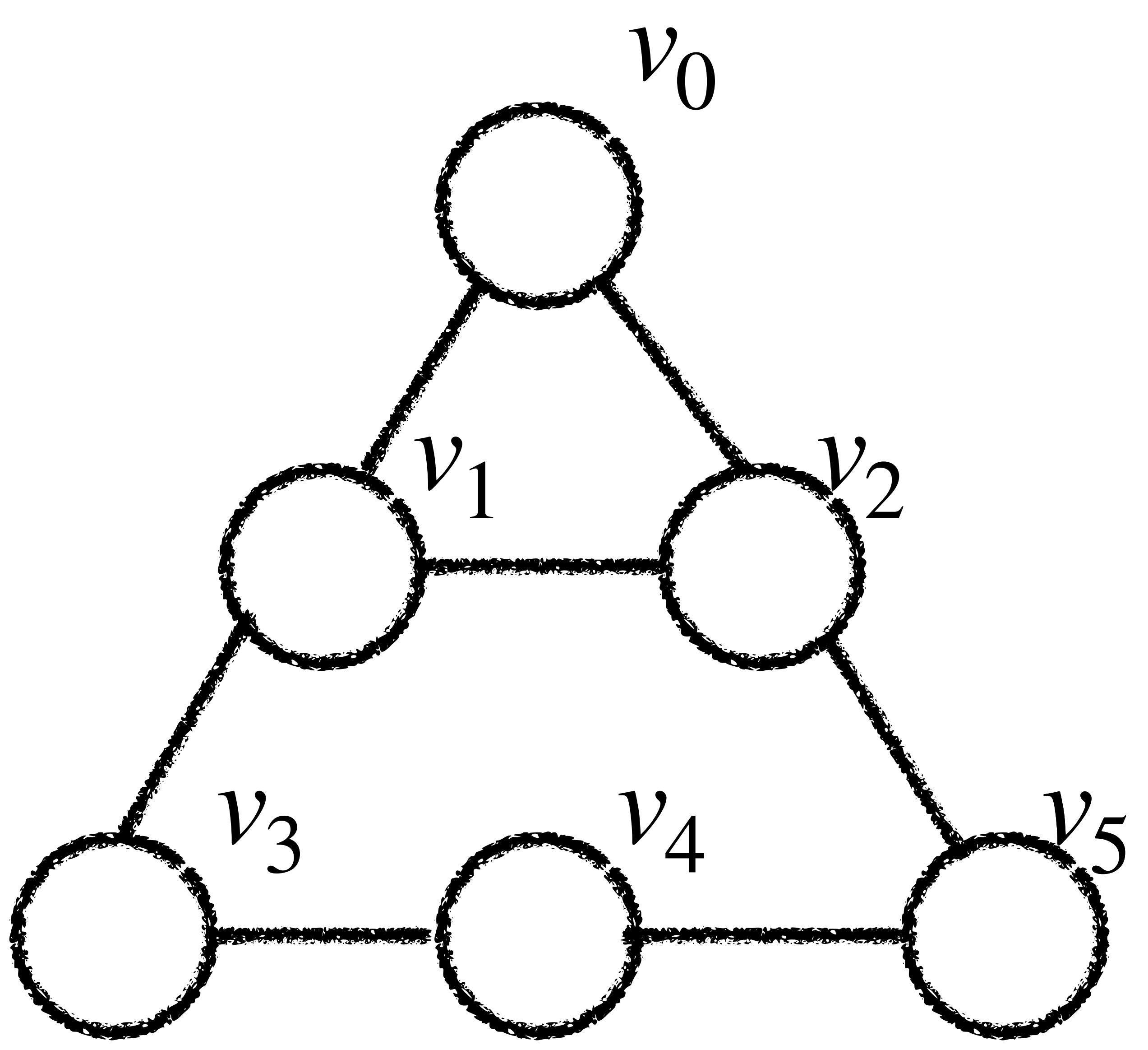
Random scan: each entry of the scan sequence is chosen for V u.a.r.

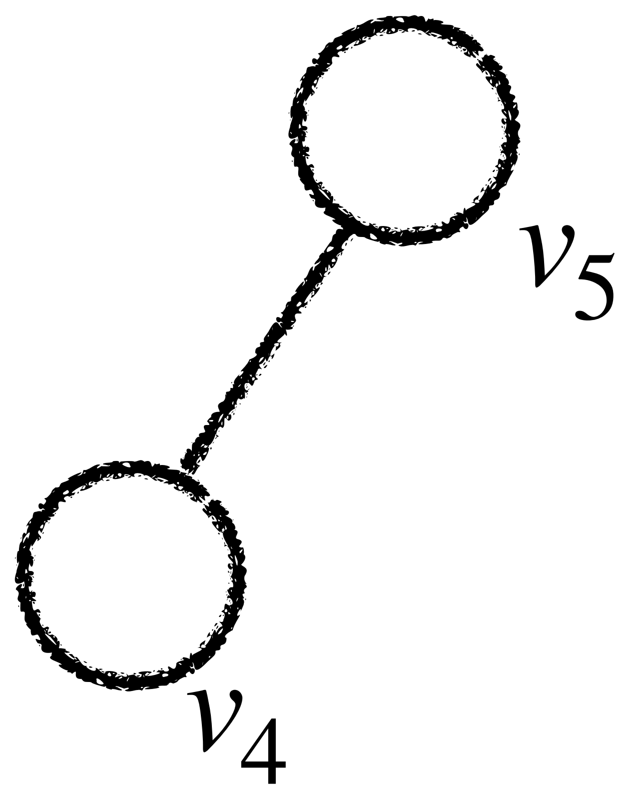
Enumerate all possible visited scan sequences within $K = O(\log n)$ random bits?

$v_2, v_5, v_1, v_0, v_2, v_3, v_4, v_0, v_5$

Construct witness tree

Go backwards in time, each time append to neighbour with largest depth





Derandomising random scan

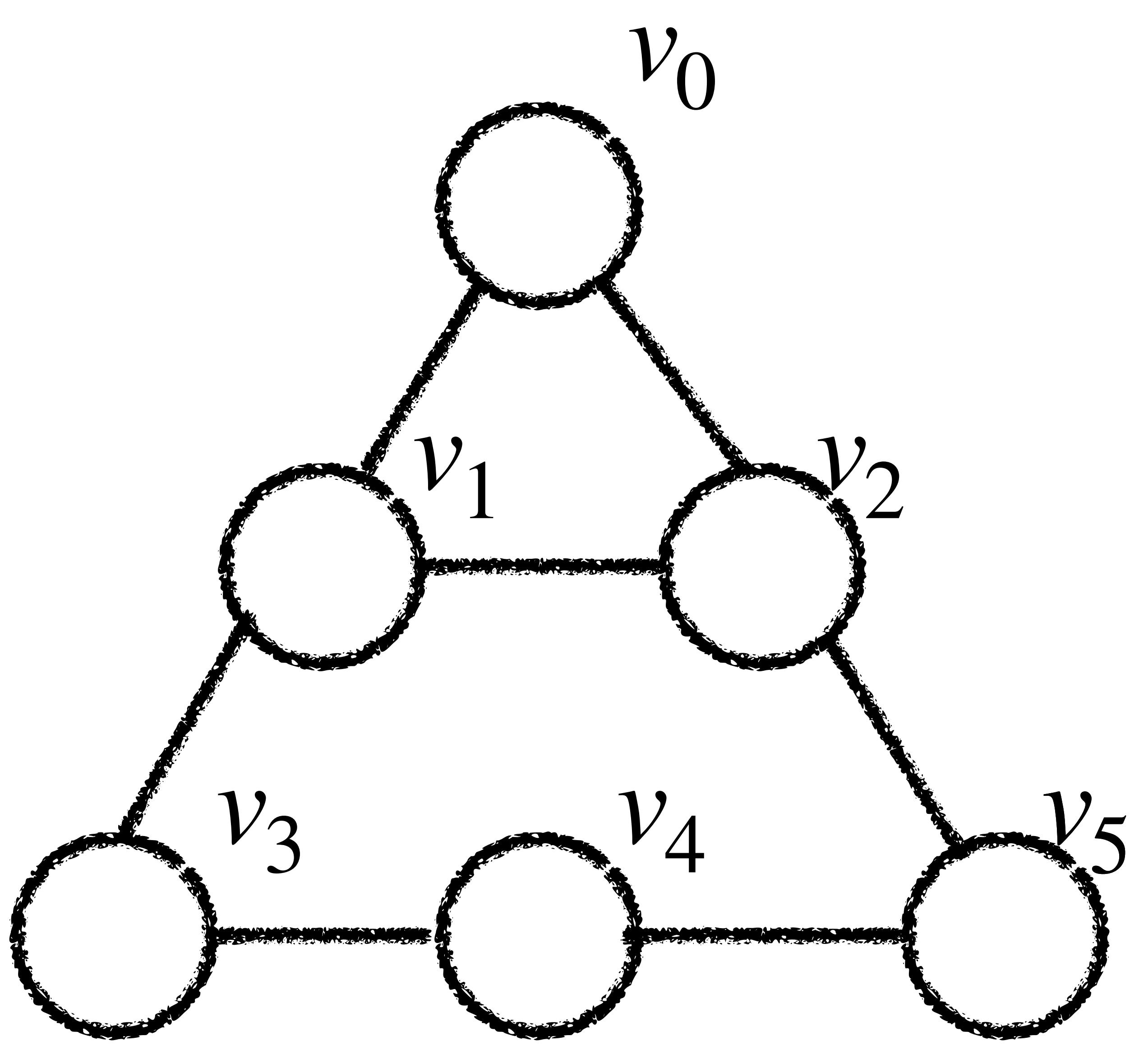
Random scan: each entry of the scan sequence is chosen for V u.a.r.

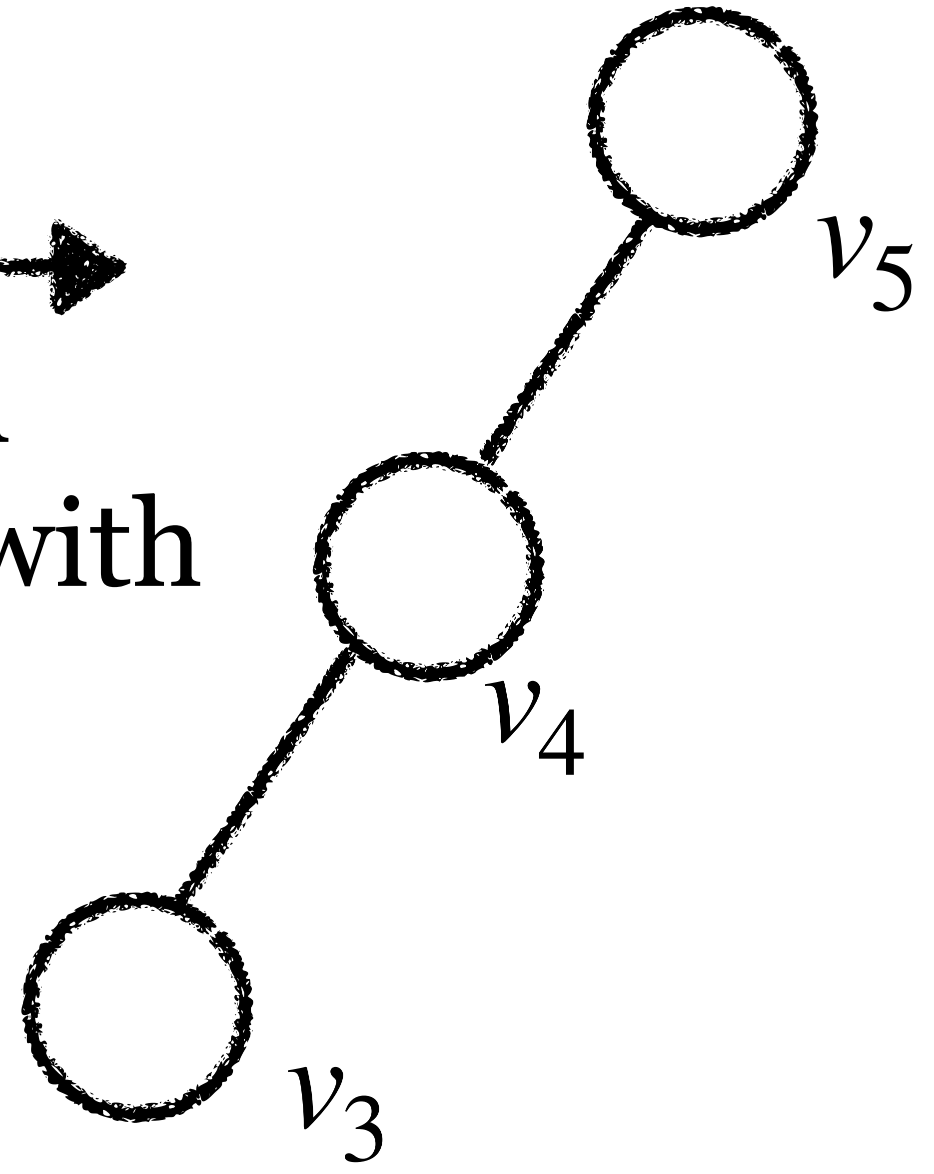
Enumerate all possible visited scan sequences within $K = O(\log n)$ random bits?

$v_2, v_5, v_1, v_0, v_2, v_3, v_4, v_0, v_5$

Construct witness tree

Go backwards in time, each time append to neighbour with largest depth



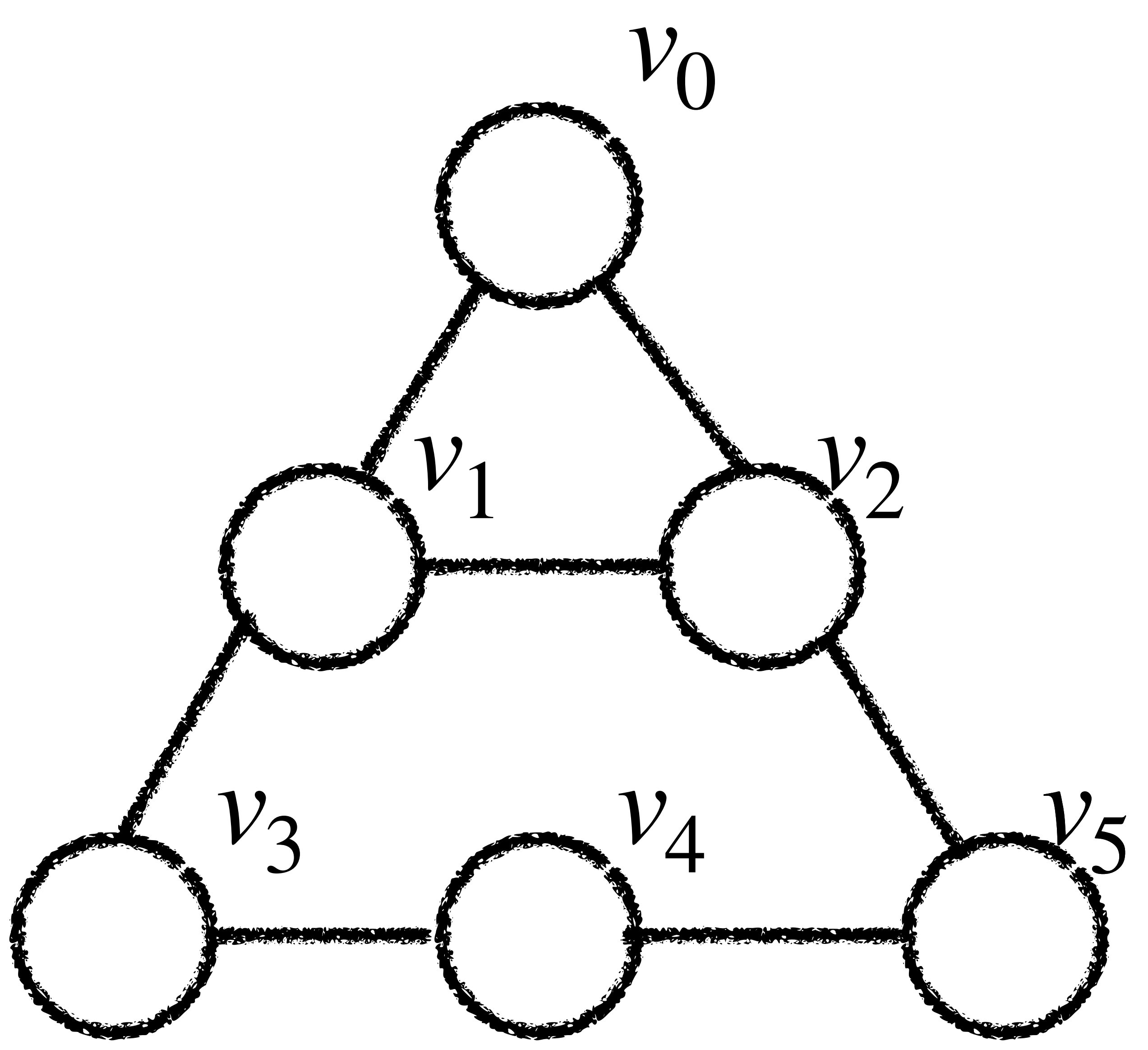


Derandomising random scan

Random scan: each entry of the scan sequence is chosen for V u.a.r.

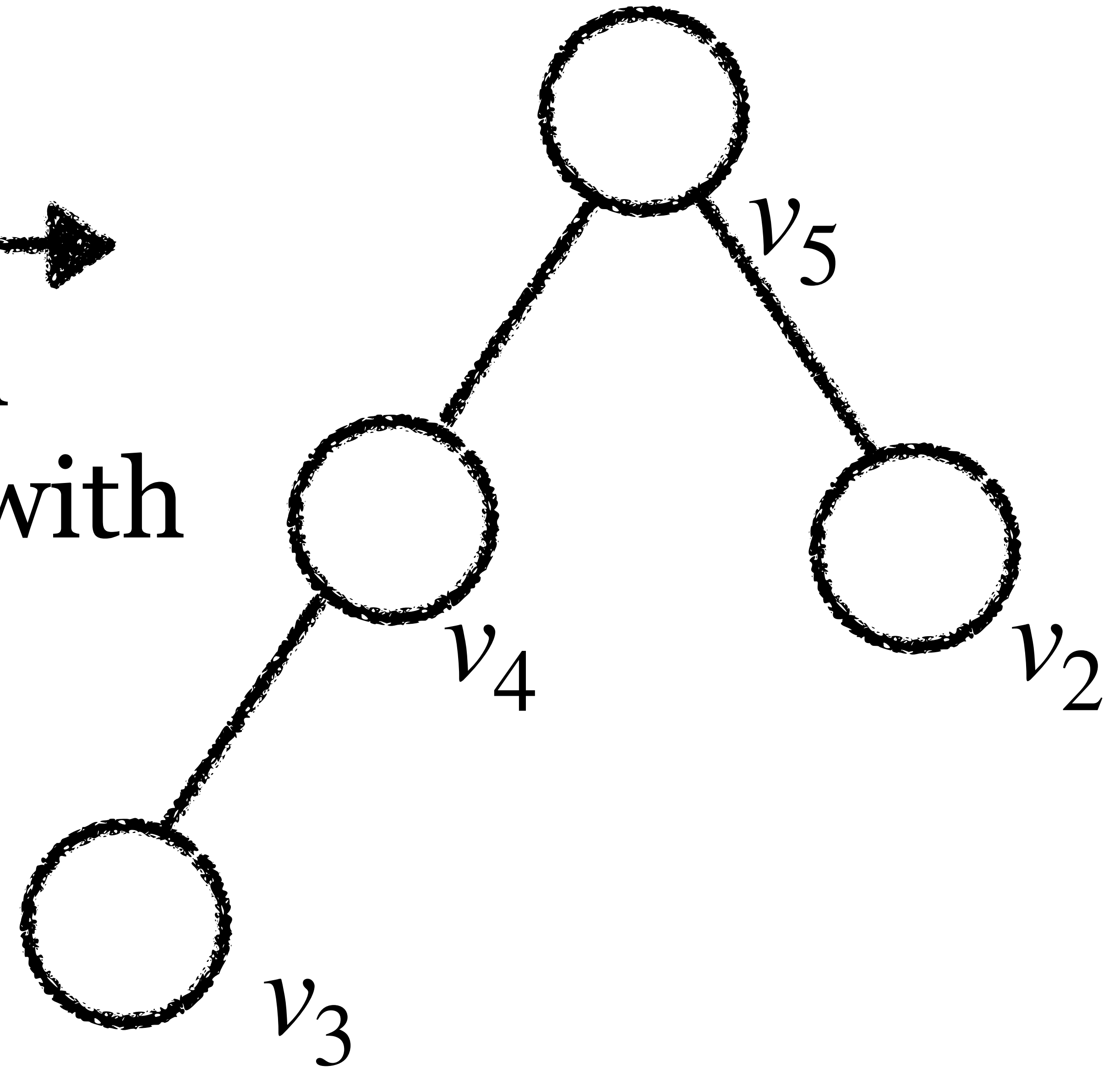
Enumerate all possible visited scan sequences within $K = O(\log n)$ random bits?

$v_2, v_5, v_1, v_0, v_2, v_3, v_4, v_0, v_5$



Construct witness tree

Go backwards in time, each time append to neighbour with largest depth



Derandomising random scan

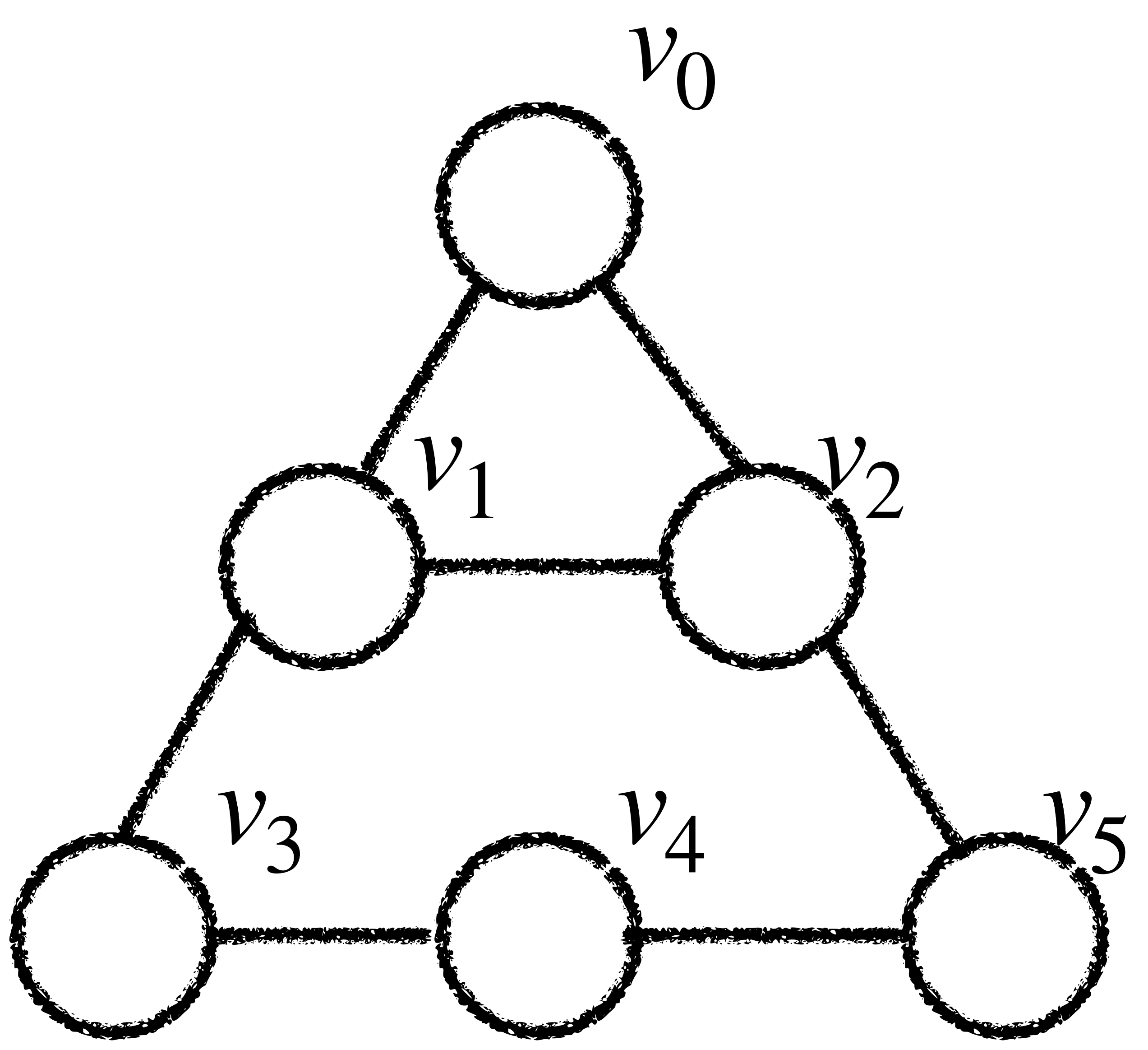
Random scan: each entry of the scan sequence is chosen for V u.a.r.

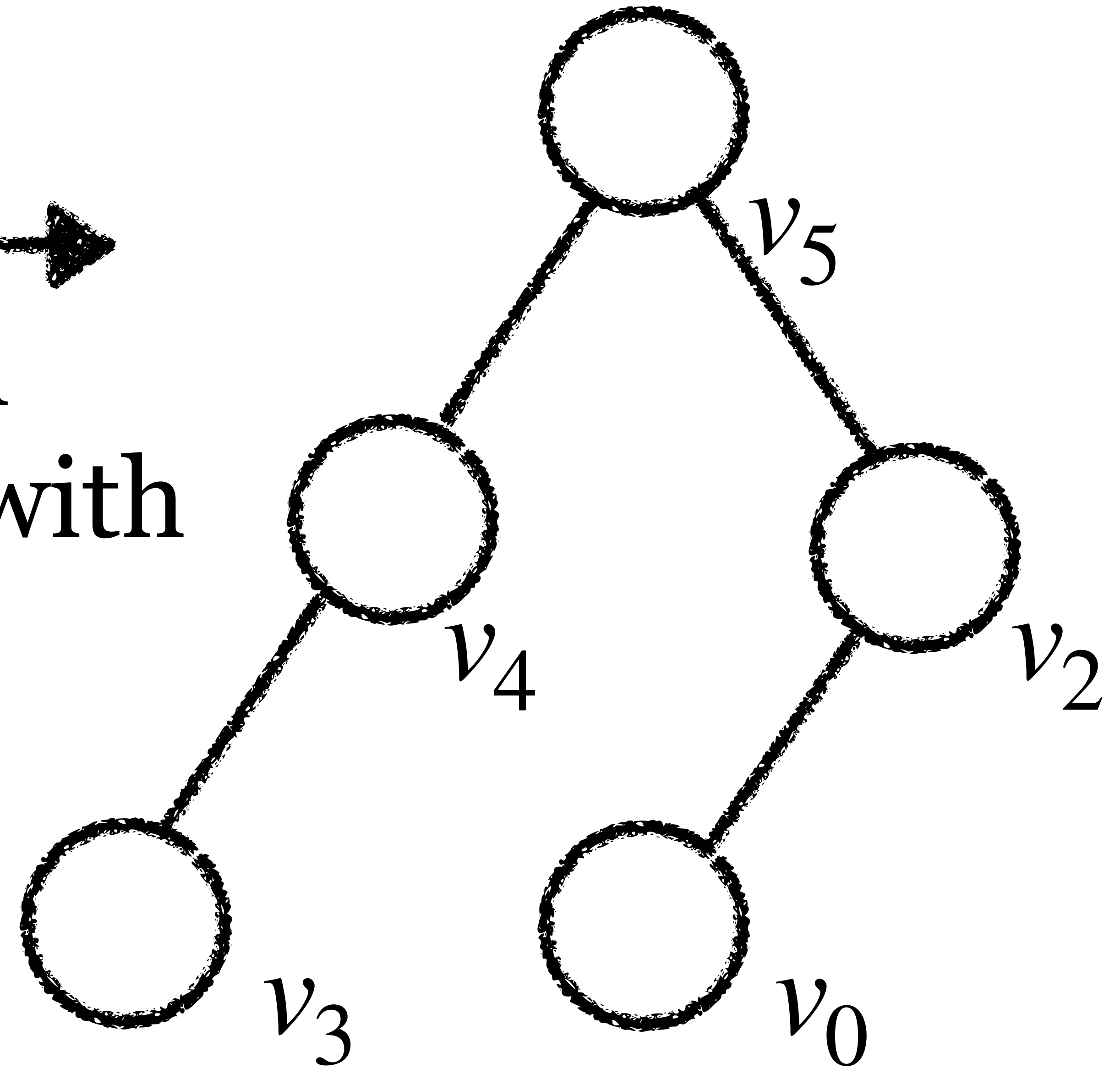
Enumerate all possible visited scan sequences within $K = O(\log n)$ random bits?

Construct witness tree

$v_2, v_5, v_1, v_0, v_2, v_3, v_4, v_0, v_5$

Go backwards in time, each time append to neighbour with largest depth





Derandomising random scan

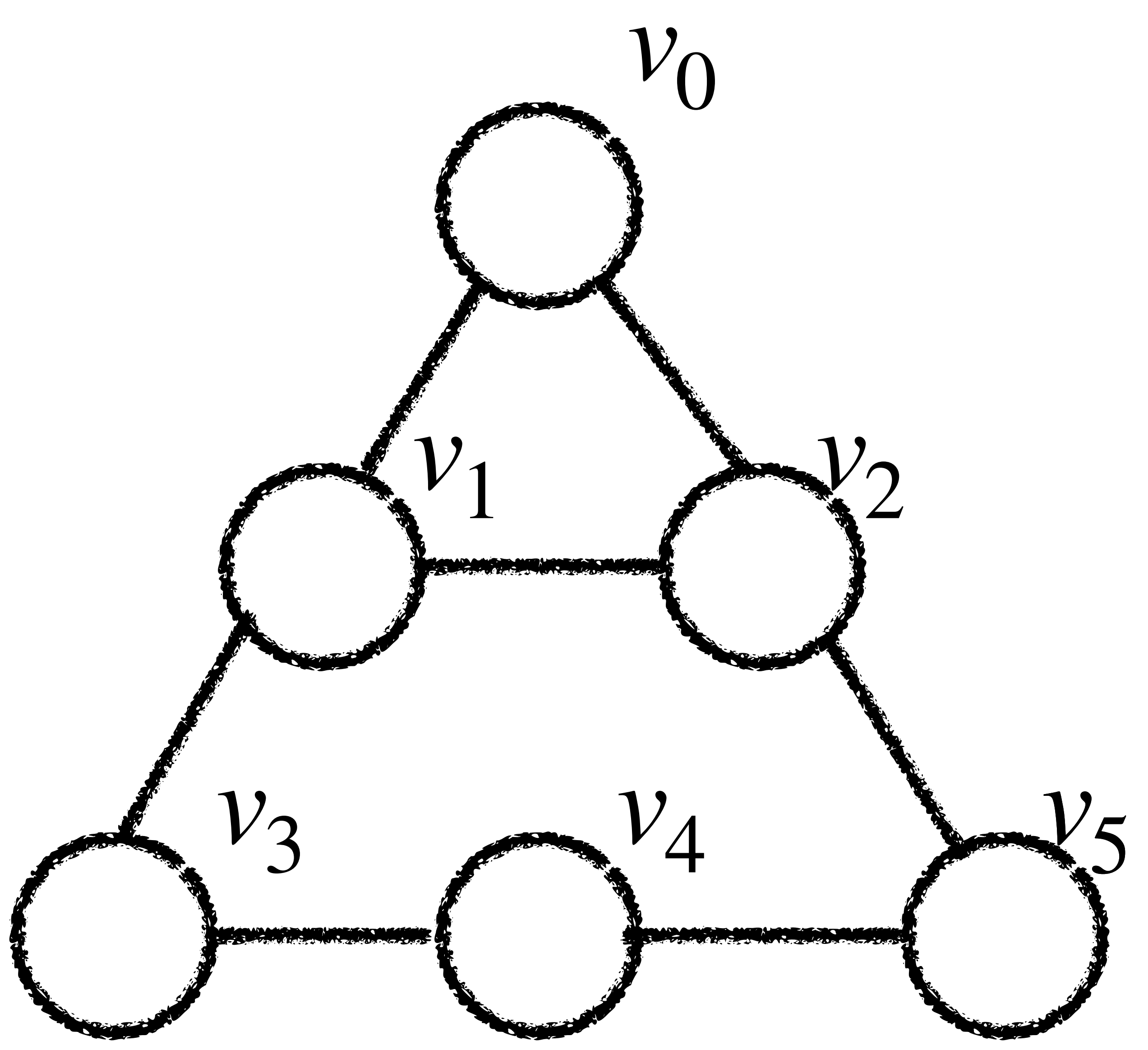
Random scan: each entry of the scan sequence is chosen for V u.a.r.

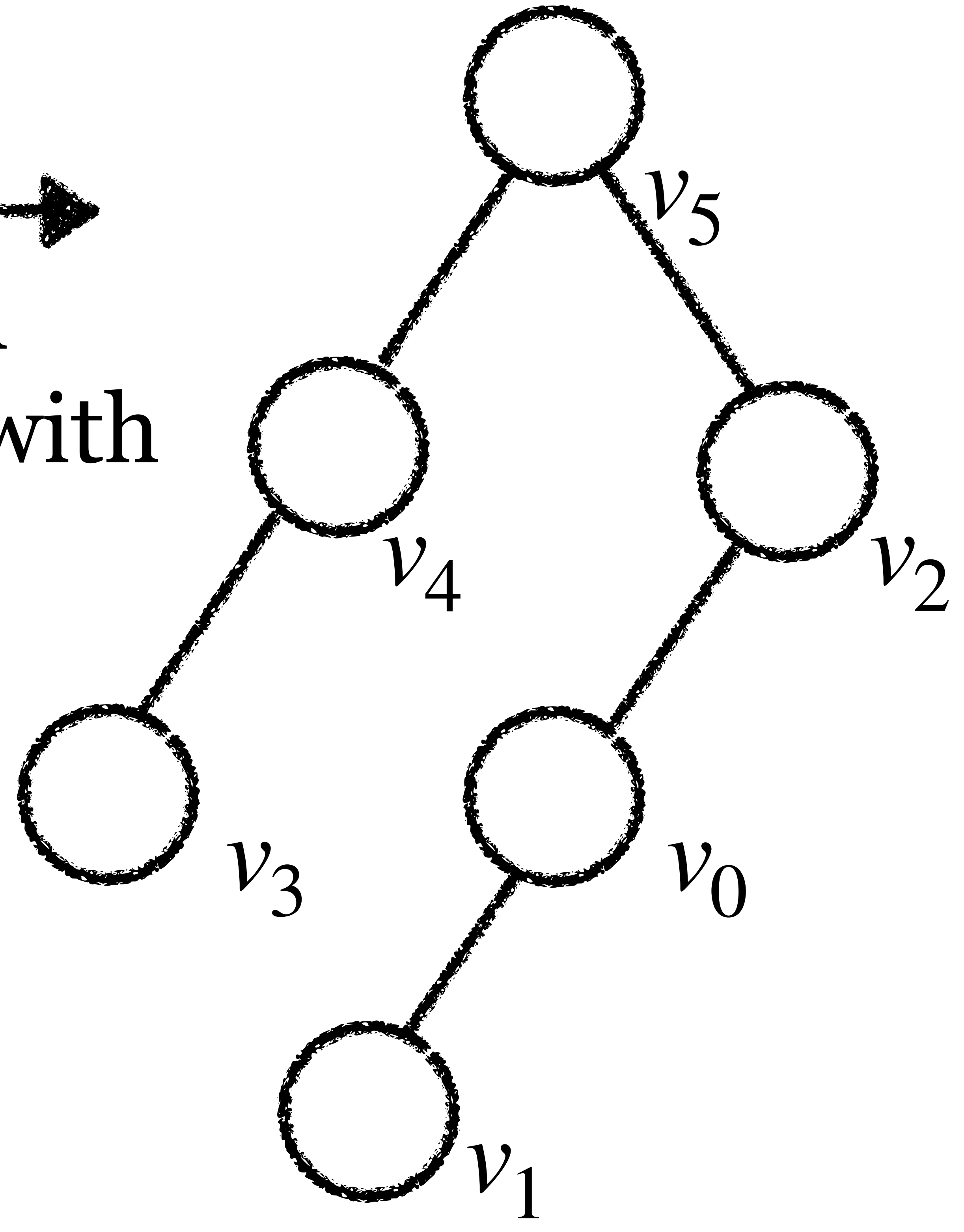
Enumerate all possible visited scan sequences within $K = O(\log n)$ random bits?

$v_2, v_5, v_1, v_0, v_2, v_3, v_4, v_0, v_5$

Construct witness tree

Go backwards in time, each time append to neighbour with largest depth



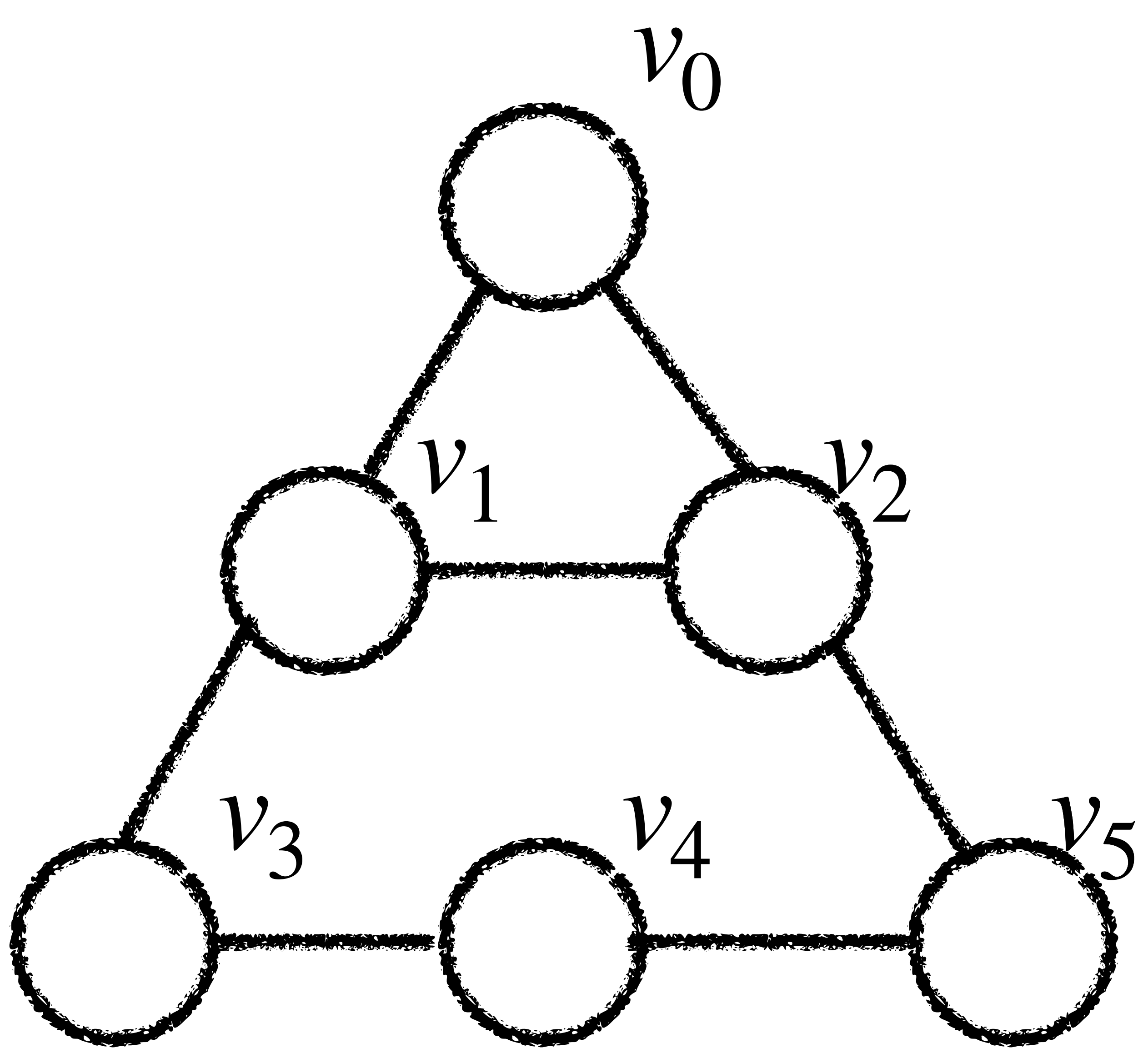


Derandomising random scan

Random scan: each entry of the scan sequence is chosen for V u.a.r.

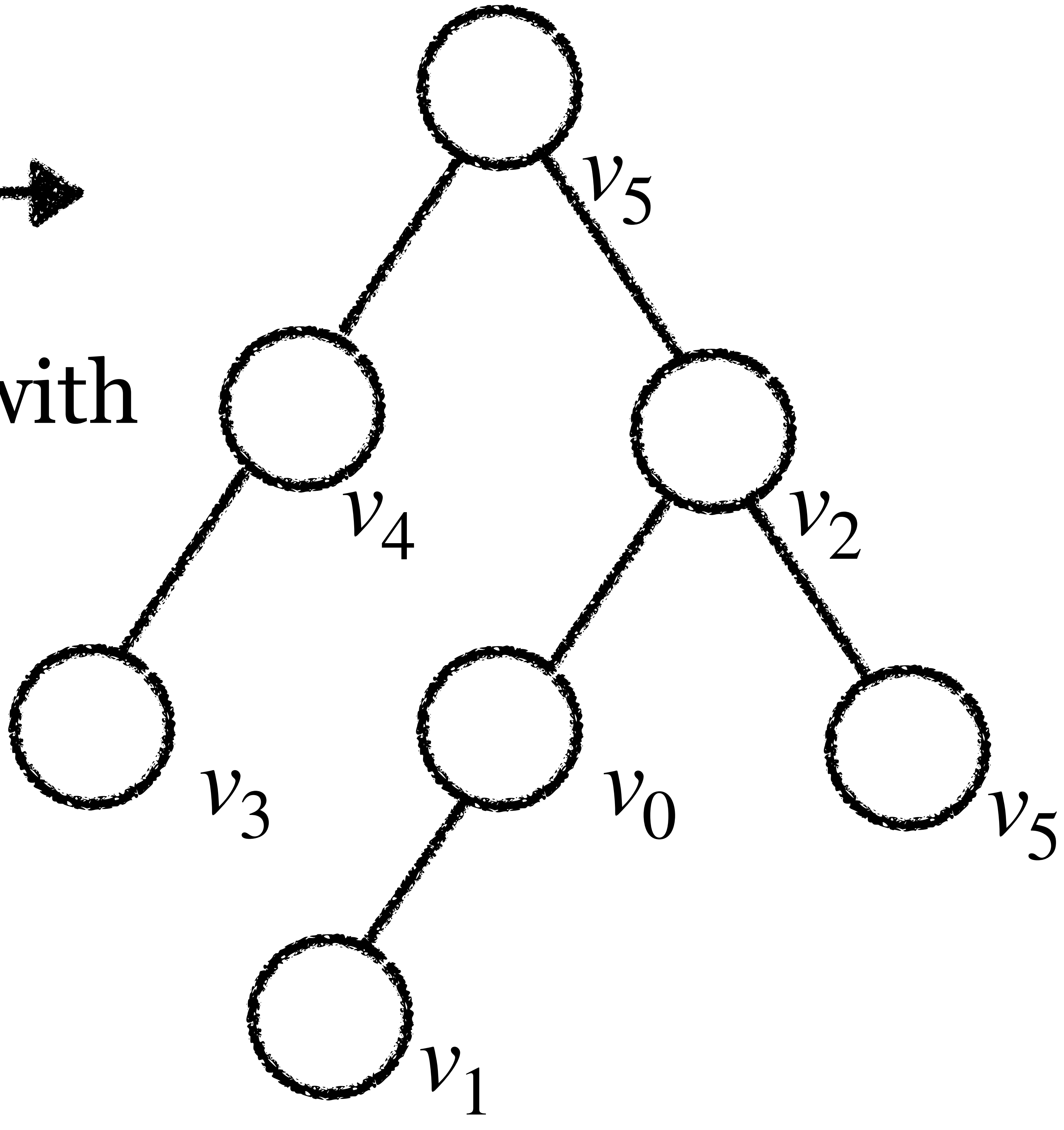
Enumerate all possible visited scan sequences within $K = O(\log n)$ random bits?

$v_2, v_5, v_1, v_0, v_2, v_3, v_4, v_0, v_5$



Construct witness tree

Go backwards in time, each time append to neighbour with largest depth

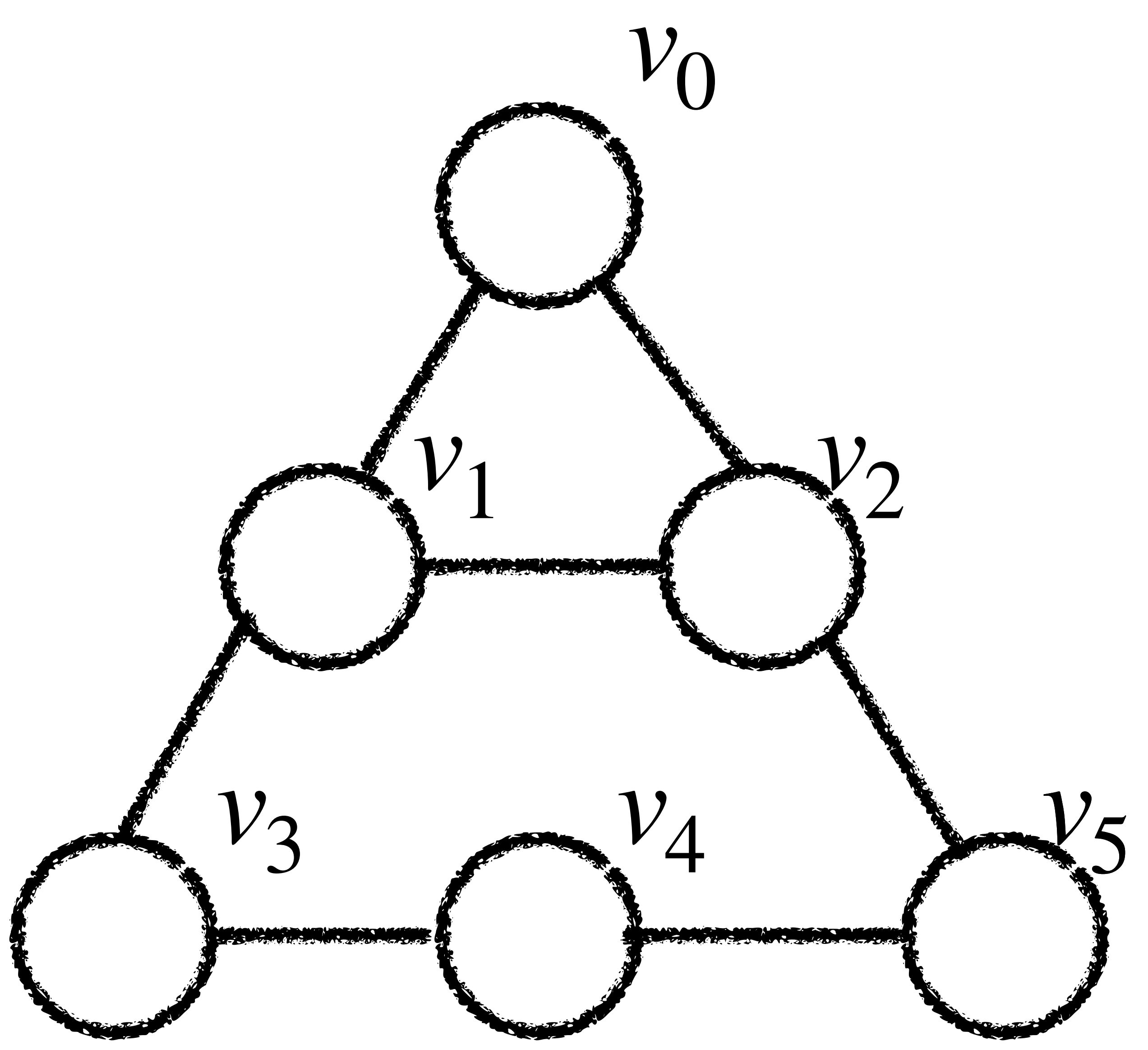


Derandomising random scan

Random scan: each entry of the scan sequence is chosen for V u.a.r.

Enumerate all possible visited scan sequences within $K = O(\log n)$ random bits?

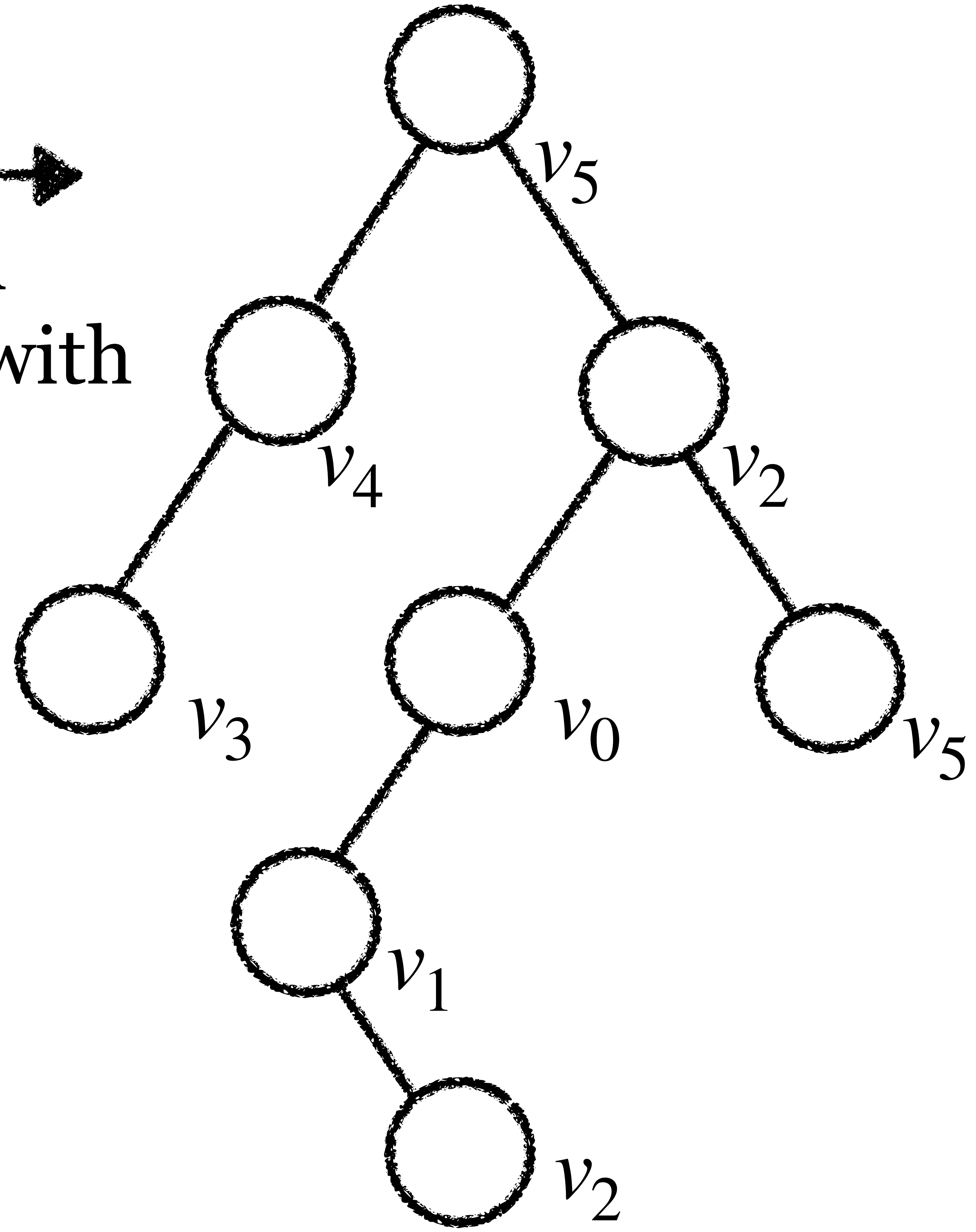
$v_2, v_5, v_1, v_0, v_2, v_3, v_4, v_0, v_5$



Construct witness tree

Go backwards in time, each time append to neighbour with largest depth

A "local total ordering"!

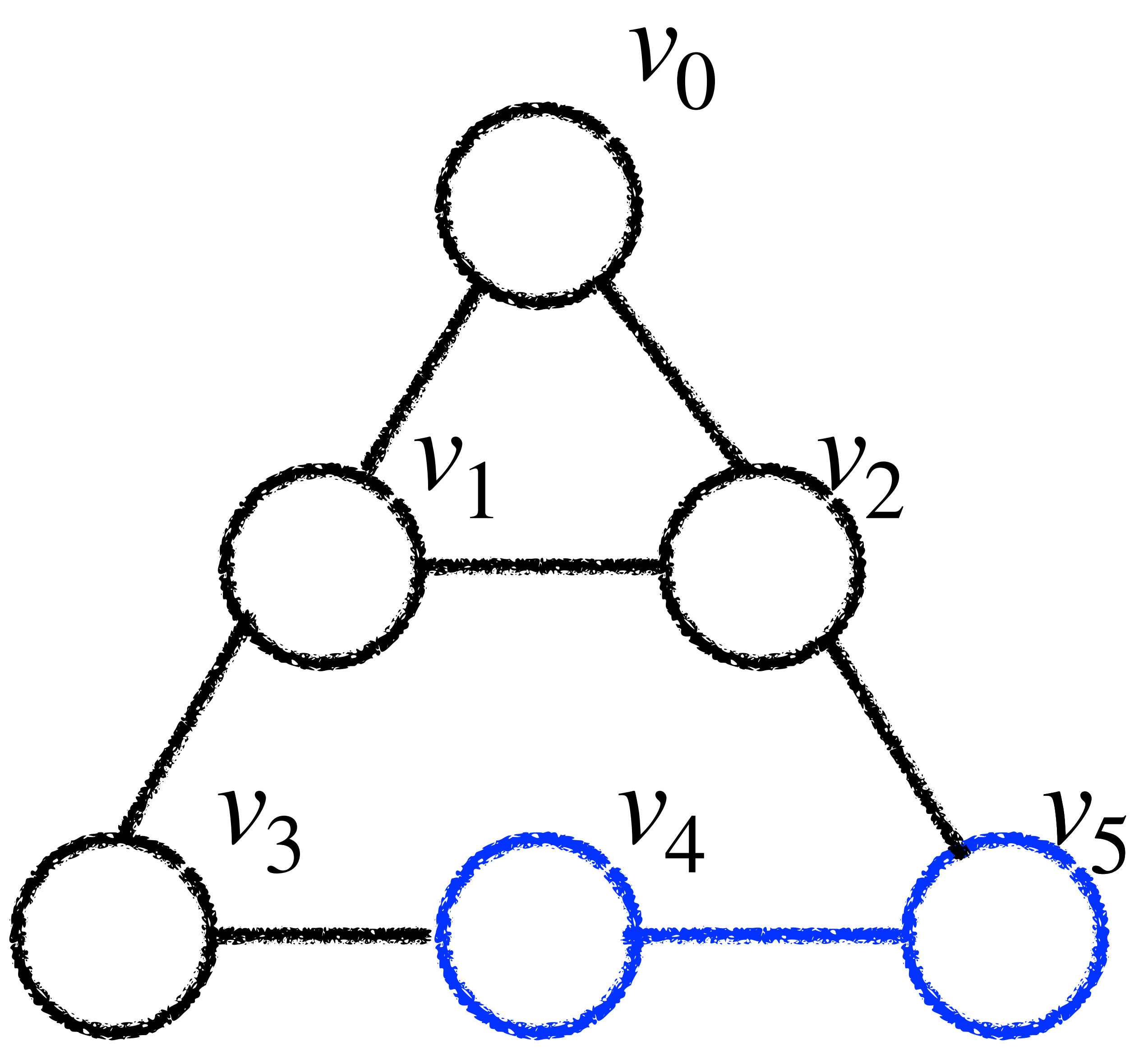


Derandomising random scan

Random scan: each entry of the scan sequence is chosen for V u.a.r.

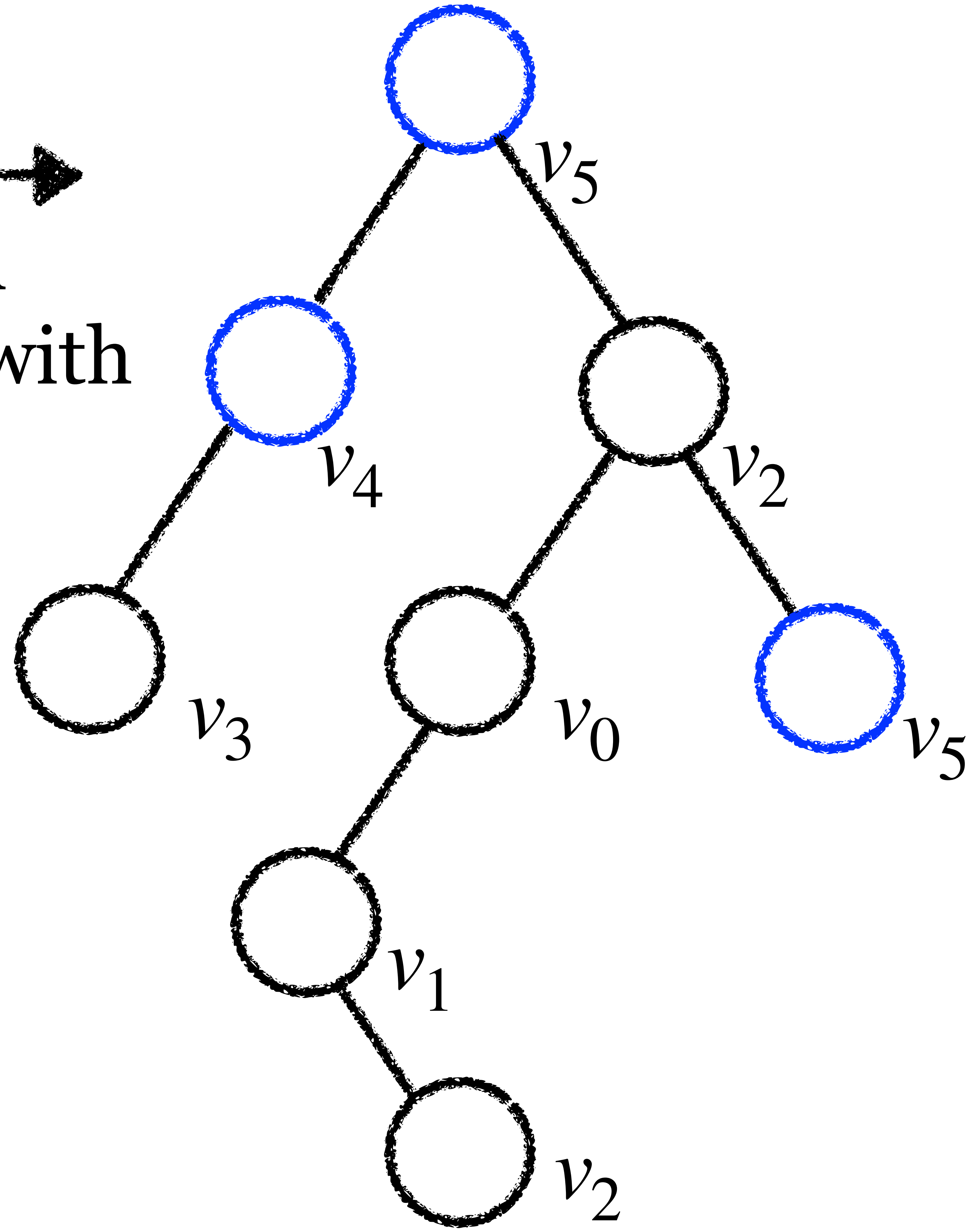
Enumerate all possible visited scan sequences within $K = O(\log n)$ random bits?

$v_2, v_5, v_1, v_0, v_2, v_3, v_4, v_0, v_5$



Construct witness tree

Go backwards in time, each time append to neighbour with largest depth



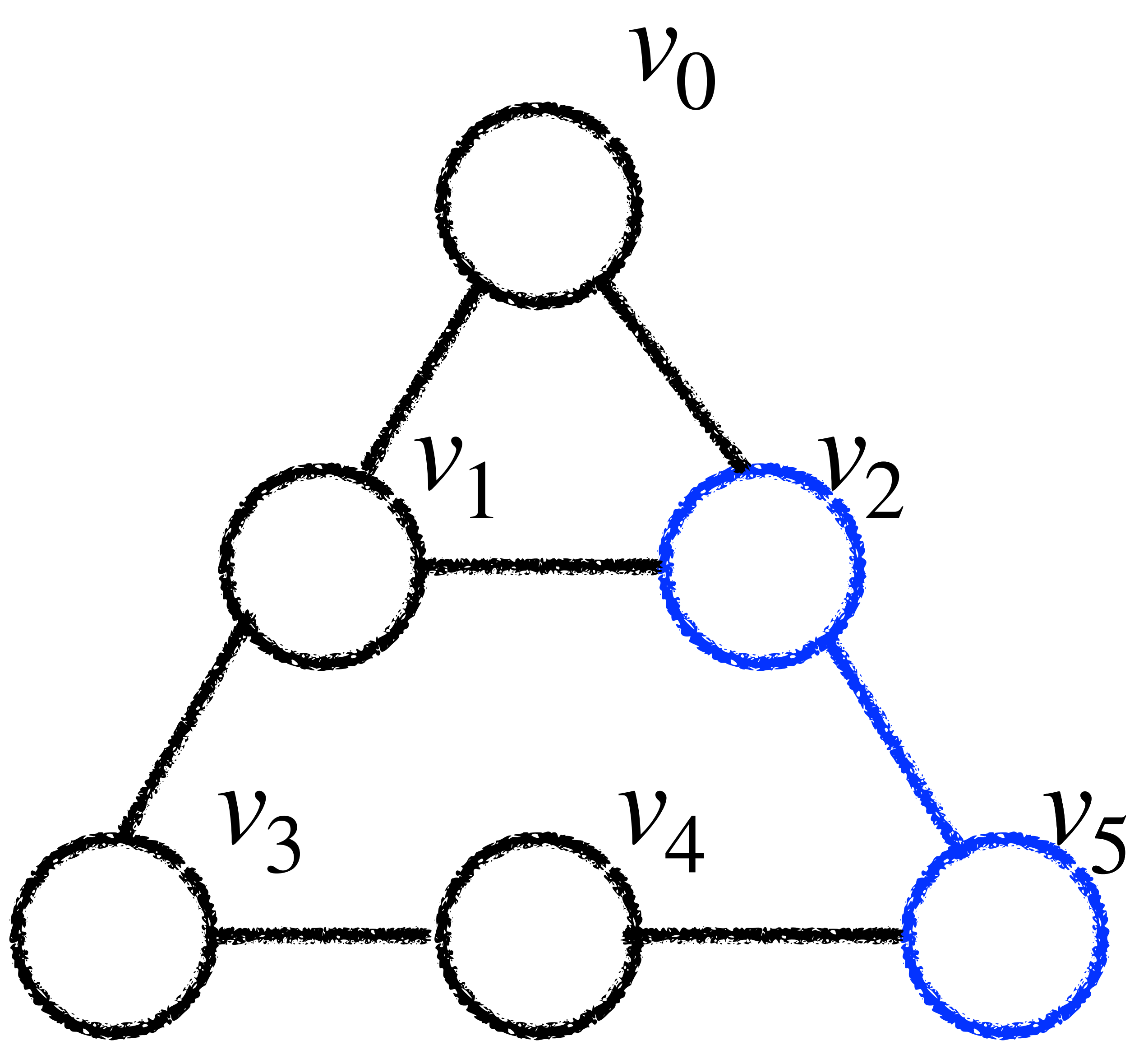
A “local total ordering”!

Derandomising random scan

Random scan: each entry of the scan sequence is chosen for V u.a.r.

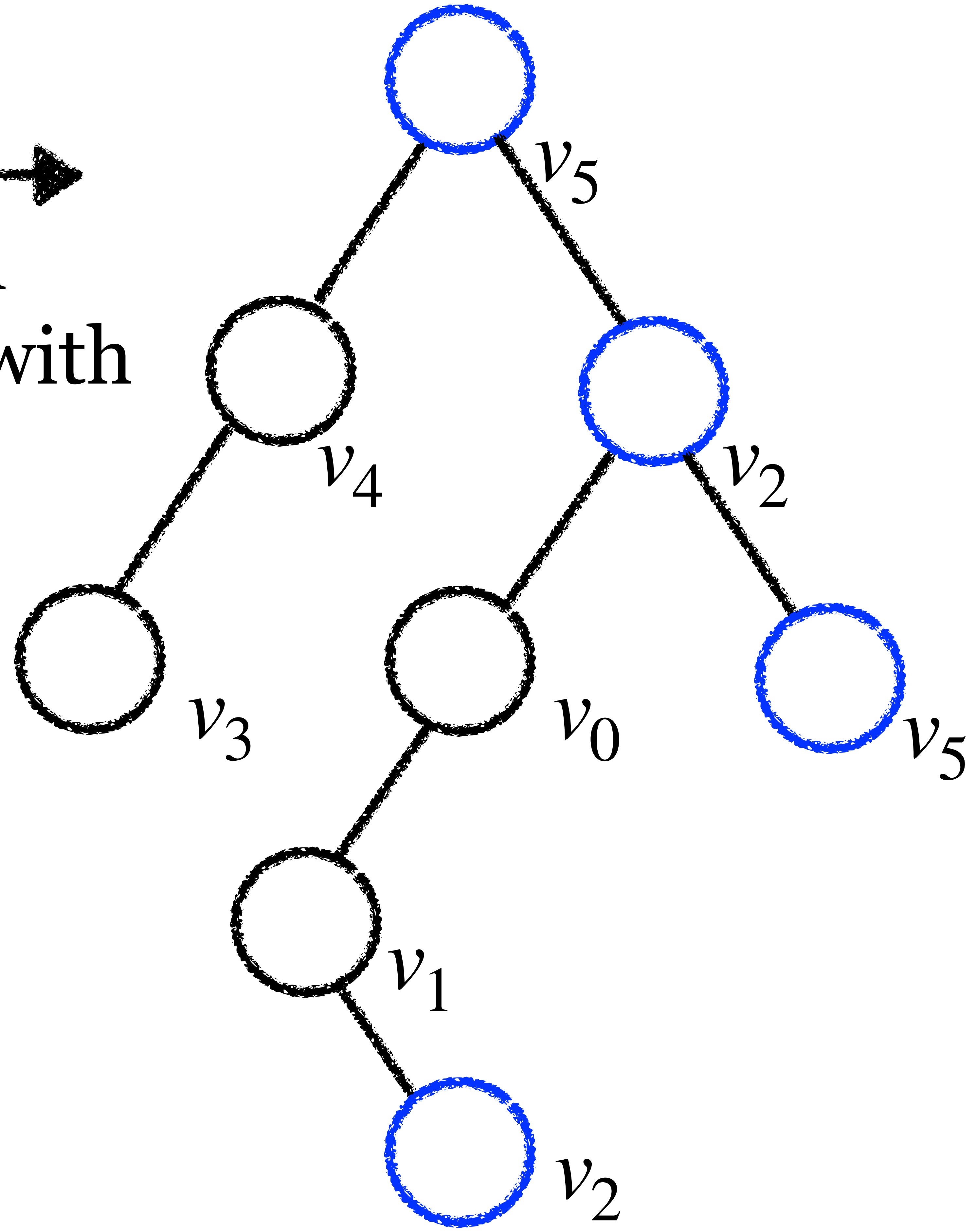
Enumerate all possible visited scan sequences within $K = O(\log n)$ random bits?

$v_2, v_5, v_1, v_0, v_2, v_3, v_4, v_0, v_5$



Construct witness tree

Go backwards in time, each time append to neighbour with largest depth



A “local total ordering”!

Derandomising random scan

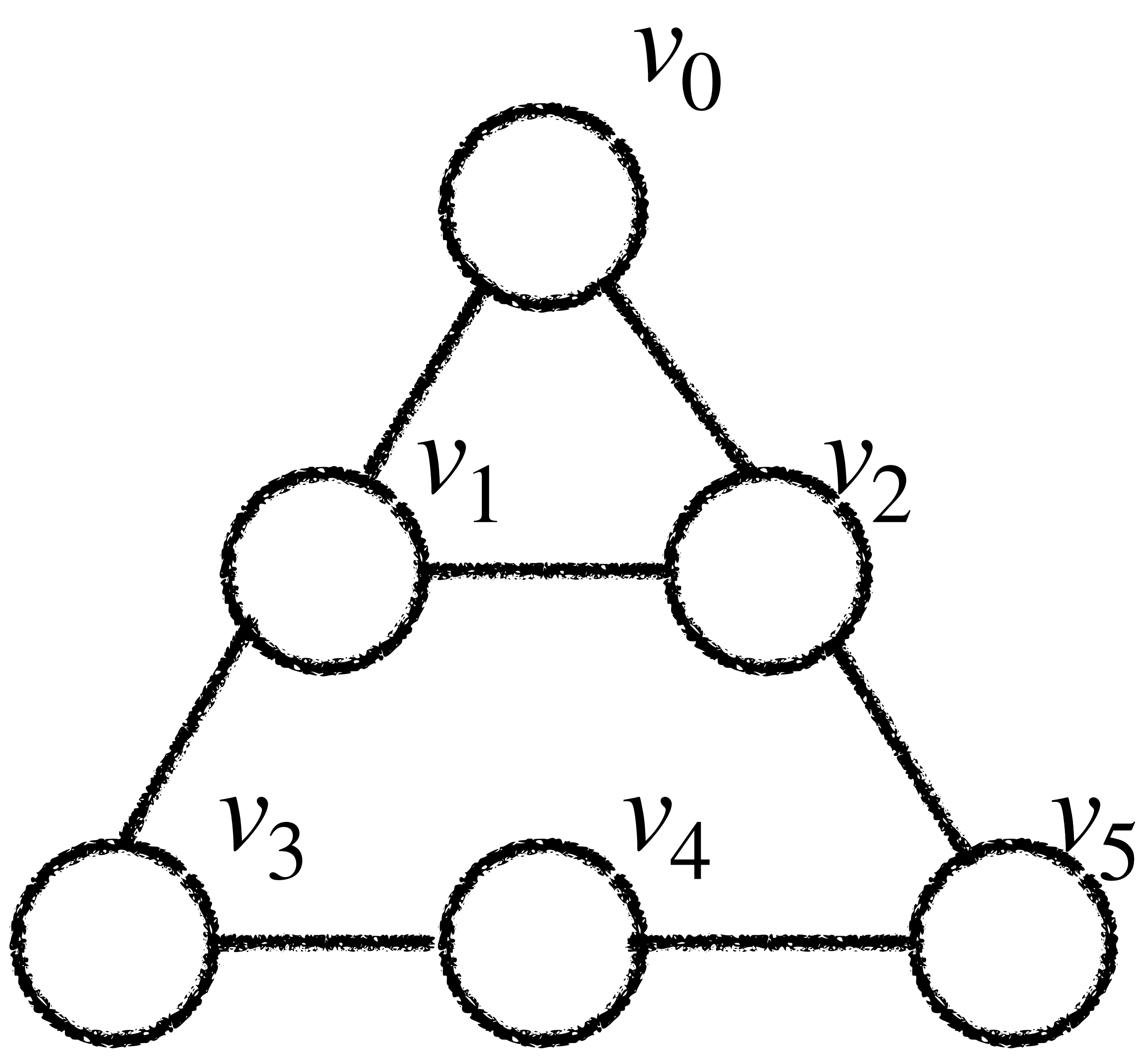
Random scan: each entry of the scan sequence is chosen for V u.a.r.

Enumerate all possible visited scan sequences within $K = O(\log n)$ random bits?

$v_2, v_5, v_1, v_0, v_2, v_3, v_4, v_0, v_5$

Construct witness tree

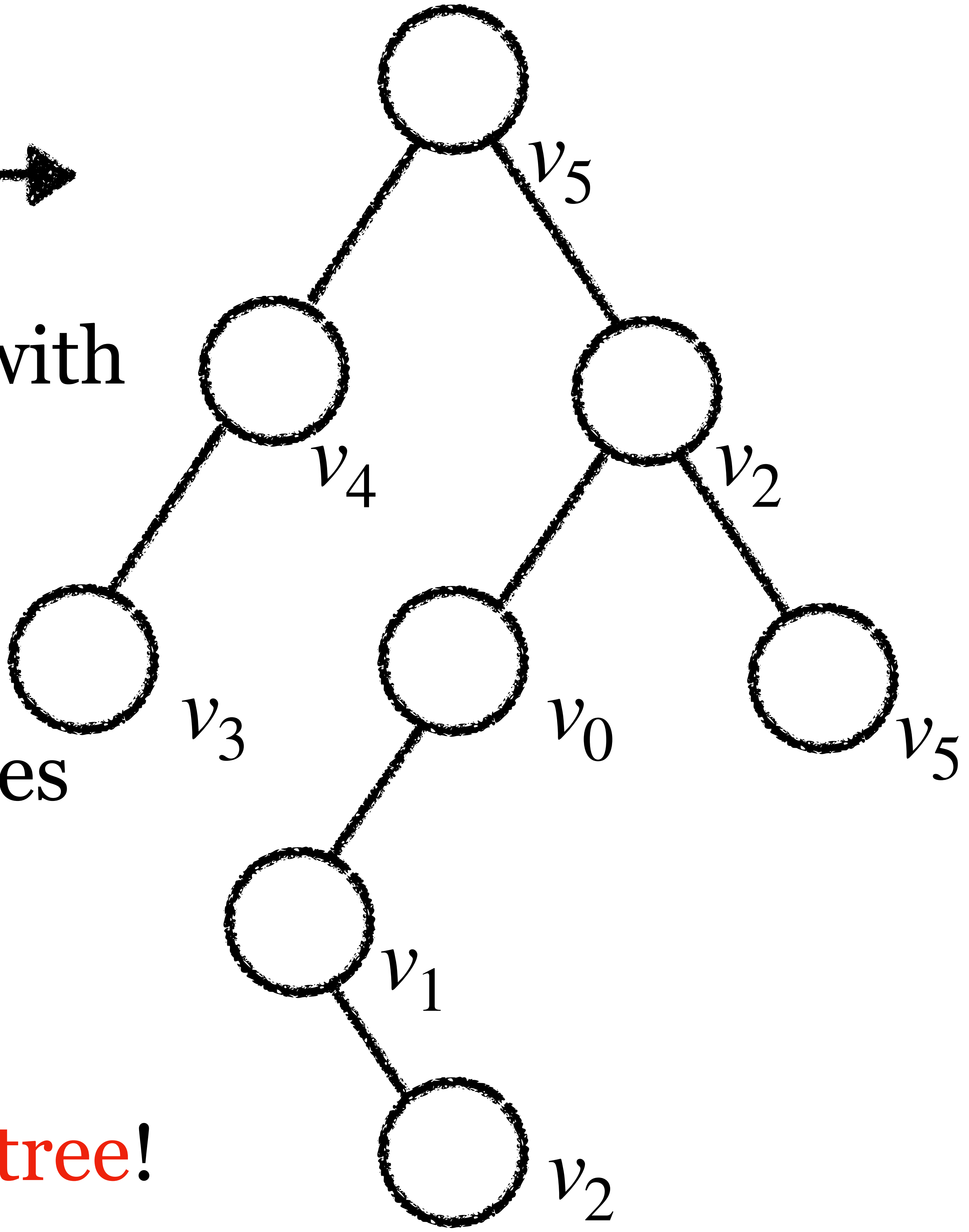
Go backwards in time, each time append to neighbour with largest depth



Only poly(Δ^K, q^K) possible witness trees

Use dynamic programming to solve. occurring prob. of each witness tree

Coincides with Moser-Tardos witness tree!



Summary

We propose a new framework (CTTP) which gives light-weight samplers that can draw from marginal distributions for derandomising MCMC algorithms.

As concrete applications, we obtain efficient deterministic approximate counting algorithms for hypergraph independent sets and hypergraph colourings, in regimes matching the state-of-the-art achieved by randomised counting/sampling algorithms.

Thanks! Any questions?

Future directions

- Beyond the marginal lower bound requirement/coupling technique?
- Beyond $O(n \log n)$ mixing time?
- Achieve truly polynomial ($f(k, \Delta, q)\left(\frac{n}{\varepsilon}\right)^c$ for some constant c) running time ?