CS 263: Counting and Sampling

Nima Anari

slides for

Bipartite Perfect Matchings
P, P’ reversible with same stationary distribution
Review

- $P, P'$ reversible with same stationary distribution
- Comparison: route $Q'$ through $Q$ with low congestion and length.

$$\pi(\text{path} \mid X_0 = s, X_\ell = t)$$
Review

- P, P’ reversible with same stationary distribution
- Comparison: route Q’ through Q with low congestion and length.

\[ \pi(\text{path} \mid X_0 = s, X_\ell = t) \]

\[ \text{s} \quad \text{t} \]

Congestion

Suppose \( \pi \) is dist over paths and Q is ergodic flow. Congestion is

\[ \max \left\{ \frac{\mathbb{P}_{\text{path} \sim \pi}[ (x \rightarrow y) \in \text{path}]}{Q(x,y)} \mid x \neq y \right\} \]
P, P′ reversible with same stationary distribution

Comparison: route Q′ through Q with low congestion and length.

\[ \pi(\text{path} \mid X_0 = s, X_\ell = t) \]

Lemma: comparison
Suppose \( \rho, \rho' \) are \( \chi^2 \) contraction rates:

\[ \rho \geq \frac{\rho'}{(\text{congestion}) \cdot (\text{max length})} \]

Congestion
Suppose \( \pi \) is dist over paths and Q is ergodic flow. Congestion is

\[
\max \left\{ \frac{p_{\text{path} \sim \pi}(x \rightarrow y) \in \text{path}}{Q(x, y)} \mid x \neq y \right\}
\]
Review

- $P, P'$ reversible with same stationary distribution
- Comparison: route $Q'$ through $Q$ with low congestion and length.
  
  $\pi(\text{path} \mid X_0 = s, X_\ell = t)$

**Lemma: comparison**

Suppose $\rho, \rho'$ are $\chi^2$ contraction rates:

$$\rho \geq \frac{\rho'}{(\text{congestion}) \cdot (\text{max length})}$$

- If $\text{len} \leq 1$, can use any $D_\phi$.

**Congestion**

Suppose $\pi$ is dist over paths and $Q$ is ergodic flow. Congestion is

$$\max \left\{ \frac{P_{\text{path} \sim \pi}[\{(x \to y) \in \text{path}\}]}{Q(x, y)} \mid x \neq y \right\}$$
Review

- P, P' reversible with same stationary distribution
- Comparison: route Q' through Q with low congestion and length.

\[ \pi(\text{path} \mid X_0 = s, X_\ell = t) \]

\[ \begin{array}{c}
\text{s} \quad \text{t}
\end{array} \]

Lemma: comparison

Suppose \( \rho, \rho' \) are \( \chi^2 \) contraction rates:

\[ \rho \geq \frac{\rho'}{(\text{congestion}) \cdot (\text{max length})} \]

- If \( \text{len} \leq 1 \), can use any \( D_\phi \).
- Canonical paths: a few-to-one mapping \( \text{enc} \) from \( (s, t) \)-pairs whose path passes \( x \to y \) to \( \Omega \):

\[ \mu(s)\mu(t) \leq C \cdot \mu(\text{enc}(s, t))Q(x, y) \]

Congestion

Suppose \( \pi \) is dist over paths and Q is ergodic flow. Congestion is

\[ \max \left\{ \frac{P_{\text{path} \sim \pi}[(x \to y) \in \text{path}]}{Q(x, y)} \mid x \neq y \right\} \]
Review

- P, P′ reversible with same stationary distribution
- Comparison: route Q′ through Q with low congestion and length.

\[ \pi(\text{path} | X_0 = s, X_\ell = t) \]

\[
\begin{tikzpicture}
    \node (s) at (0,0) {$s$};
    \node (t) at (2,0) {$t$};
    \node (1) at (1,1) {}; \node (2) at (1,-1) {};
    \draw[->] (s) to (1);
    \draw[->] (1) to (2);
    \draw[->] (2) to (t);
    \draw[->] (s) to (t);
\end{tikzpicture}
\]

Lemma: comparison

Suppose \( \rho, \rho' \) are \( \chi^2 \) contraction rates:

\[ \rho \geq \frac{\rho'}{(\text{congestion}) \cdot (\text{max length})} \]

- If \( \text{len} \leq 1 \), can use any \( D_\phi \).
- Canonical paths: a few-to-one mapping \( \text{enc} \) from \((s, t)\)-pairs whose path passes \( x \rightarrow y \) to \( \Omega \):

\[ \mu(s) \mu(t) \leq C \cdot \mu(\text{enc}(s, t)) Q(x, y) \]

- If M-to-one, then \( \text{cong} \leq CM \).
Review

- $P, P'$ reversible with same stationary distribution
- Comparison: route $Q'$ through $Q$ with low congestion and length.

\[ \pi(\text{path} | X_0 = s, X_\ell = t) \]

\[ s \longrightarrow \Omega \longrightarrow t \]

**Concentration**

Suppose $\pi$ is dist over paths and $Q$ is ergodic flow. Concentration is

\[ \max \left\{ \frac{P_{\text{path} \sim \pi}[(x \rightarrow y) \in \text{path}]}{Q(x, y)} \mid x \neq y \right\} \]

**Lemma: comparison**

Suppose $\rho, \rho'$ are $\chi^2$ contraction rates:

\[ \rho \geq \frac{\rho'}{(\text{congestion}) \cdot (\text{max length})} \]

- If $\text{len} \leq 1$, can use any $\mathcal{D}_\phi$.
- **Canonical paths:** a few-to-one mapping $\text{enc}$ from $(s, t)$-pairs whose path passes $x \rightarrow y$ to $\Omega$:

\[ \mu(s)\mu(t) \leq C \cdot \mu(\text{enc}(s, t))Q(x, y) \]

- If $M$-to-one, then $\text{cong} \leq CM$.
- Matching walks mix in $\text{poly}(n)$.
Perfect Matchings

- Monomer-dimer systems
- Log-concave sequences
- Bipartite graphs
Perfect Matchings

- Monomer-dimer systems
- Log-concave sequences
- Bipartite graphs
Monomer-dimer systems

Markov chain on matchings mixes in $\text{poly}(n)$ time [Jerrum-Sinclair’89].

What about perfect matchings? This is open. No strong indication/evidence either way!

However, for bipartite graphs, [Jerrum-Sinclair-Vigoda’04] showed we can approx sample/count in $\text{poly}(n)$ time.

Monomer-dimer system

Prob of matching $\propto \prod_{e \in M} \lambda_e \cdot \prod_{v \not\sim M} z_v$

Monomer weights $z_v$ can be absorbed into $\lambda_e$. So assume wlog that $z_v = 1$.

Mixing time is $\text{poly}(n, \lambda_{\text{max}})$ [Jerrum-Sinclair].

Sampling/counting possible in $\text{poly}(n, \log \lambda_{\text{max}})$ time on bipartite graphs [Jerrum-Sinclair-Vigoda].
Monomer-dimer systems

- Markov chain on matchings mixes in $\text{poly}(n)$ time [Jerrum-Sinclair’89].
- What about perfect matchings?
Monomer-dimer systems

- Markov chain on matchings mixes in $\text{poly}(n)$ time [Jerrum-Sinclair’89].

- What about perfect matchings?

- This is open. No strong indication/evidence either way! 😞
Monomer-dimer systems

- Markov chain on matchings mixes in $\text{poly}(n)$ time [Jerrum-Sinclair’89].

- What about perfect matchings?

- This is open. No strong indication/evidence either way! 😞

- However, for bipartite graphs, [Jerrum-Sinclair-Vigoda’04] showed we can approx sample/count in $\text{poly}(n)$ time. 😊

$\text{Prob of matching } \propto \prod_{e \in M} \lambda_e \cdot \prod_{v \not\sim M} z_v$

Monomer weights $z_v$ can be absorbed into $\lambda_e$. So assume wlog that $z_v = 1$.

Mixing time is $\text{poly}(n, \lambda_{\text{max}})$ [Jerrum-Sinclair].

Sampling/counting possible in $\text{poly}(n, \log \lambda_{\text{max}})$ time on bipartite graphs [Jerrum-Sinclair-Vigoda].
Monomer-dimer systems

- Markov chain on matchings mixes in $\text{poly}(n)$ time [Jerrum-Sinclair’89].
- What about perfect matchings?
- This is open. No strong indication/evidence either way! 😞
- However, for bipartite graphs, [Jerrum-Sinclair-Vigoda’04] showed we can approx sample/count in $\text{poly}(n)$ time. 😊

Prob of matching $\propto \prod_{e \in M} \lambda_e \cdot \prod_{v \not\in M} z_v$

- Monomer weights $z_v$ can be absorbed into $\lambda_e$. So assume wlog that $z_v = 1$.
- Mixing time is $\text{poly}(n, \lambda_{\text{max}})$ [Jerrum-Sinclair].
- Sampling/counting possible in $\text{poly}(n, \log \lambda_{\text{max}})$ time on bipartite graphs [Jerrum-Sinclair-Vigoda].
Monomer-dimer systems

- Markov chain on matchings mixes in \(\text{poly}(n)\) time [Jerrum-Sinclair’89].
- What about perfect matchings?

This is open. No strong indication/evidence either way! 😞

- However, for bipartite graphs, [Jerrum-Sinclair-Vigoda’04] showed we can approx sample/count in \(\text{poly}(n)\) time. 😊

\[
\text{Prob of matching } \propto \prod_{e \in M} \lambda_e \cdot \prod_{v \not\sim M} z_v
\]

- Monomer weights \(z_v\) can be absorbed into \(\lambda_e\). So assume wlog that \(z_v = 1\).
Monomer-dimer systems

- Markov chain on matchings mixes in $\text{poly}(n)$ time [Jerrum-Sinclair’89].
- What about perfect matchings?

![Perfect Matching Diagram]

- This is open. No strong indication/evidence either way! 😞
- However, for bipartite graphs, [Jerrum-Sinclair-Vigoda’04] showed we can approx sample/count in $\text{poly}(n)$ time. 😊

Monomer-dimer system

Prob of matching $\propto \prod_{e \in M} \lambda_e \cdot \prod_{v \not\sim M} z_v$

- Monomer weights $z_v$ can be absorbed into $\lambda_e$. So assume wlog that $z_v = 1$.
- Mixing time is $\text{poly}(n, \lambda_{\max})$ [Jerrum-Sinclair] 😊
Monomer-dimer systems

- Markov chain on matchings mixes in $\text{poly}(n)$ time [Jerrum-Sinclair’89].
- What about perfect matchings?

  - This is open. No strong indication/evidence either way! 😞
  - However, for bipartite graphs, [Jerrum-Sinclair-Vigoda’04] showed we can approx sample/count in $\text{poly}(n)$ time. 😊

Monomer-dimer system

$$\text{Prob of matching } \propto \prod_{e \in M} \lambda_e \cdot \prod_{v \not\sim M} z_v$$

- Monomer weights $z_v$ can be absorbed into $\lambda_e$. So assume wlog that $z_v = 1$.
- Mixing time is $\text{poly}(n, \lambda_{\text{max}})$ [Jerrum-Sinclair]. 😊
- Sampling/counting possible in $\text{poly}(n, \log \lambda_{\text{max}})$ time on bipartite graphs [Jerrum-Sinclair-Vigoda]. 😊
Theorem [Jerrum-Sinclair]
Mixing time is $\text{poly}(n, \lambda_{\text{max}})$. 

Proof: for the $x \rightarrow y$ transition:

Same encoding as before: $\text{enc}(s, t) = s \oplus t \oplus x$.

Using notation $\lambda_S = \prod_{e \in S} \lambda_e$:

$\lambda_s \lambda_t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda_{\text{enc}}(s, t)$.

Similarly:

$\lambda_s \lambda_t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda_{\text{enc}}(s, t)$.

Using Metropolis filter we get $Q(x, y) \geq \min\{\mu(x), \mu(y)\}$.

So we have $\mu(s) \mu(t) \leq \text{poly}(n, \lambda_{\text{max}}) \cdot \mu(\text{enc}(s, t)) Q(x, y)$.

What if we want perfect matchings?

Idea 1: restrict chain to perfect and near-perfect one fewer edge matchings.

Idea 2: set $\lambda_e = \lambda$ very large.

Dist of matching size: $0 \cdot 1 \cdot 2 \cdot \ldots \cdot n^2$.

If $\lambda_k \cdot \#(k\text{-matchings})$ maximized for $k = n^2$, use rejection sampling.
Theorem [Jerrum-Sinclair]
Mixing time is $\text{poly}(n, \lambda_{\text{max}})$.

Proof: for the $x \rightarrow y$ transition:
Theorem [Jerrum-Sinclair]
Mixing time is \( \text{poly}(n, \lambda_{\text{max}}) \).

Proof: for the \( x \to y \) transition:

- Same encoding as before:
  \[
  \text{enc}(s, t) = s \oplus t \oplus x - \text{couple edges}
  \]
Theorem [Jerrum-Sinclair]

Mixing time is $\text{poly}(n, \lambda_{\text{max}})$.

Proof: for the $x \to y$ transition:

- Same encoding as before:
  \[ \text{enc}(s, t) = s \oplus t \oplus x - \text{couple edges} \]

- Using notation $\lambda^S = \prod_{e \in S} \lambda_e$:
  \[ \lambda^s \lambda^t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda^{\text{enc}(s, t)} \cdot \lambda^x \]
  
  couple edges

Idea 1: restrict chain to perfect and near-perfect matchings.

Idea 2: set $\lambda_e = \lambda$ very large.

Dist of matching size:

\[ 0 \quad 1 \quad 2 \quad \cdots \quad n \]

If $\lambda_k \cdot \#(k\text{-matchings})$ maximized for $k = n/2$, use rejection sampling.
Theorem [Jerrum-Sinclair]
Mixing time is $\text{poly}(n, \lambda_{\text{max}})$.

Proof: for the $x \to y$ transition:

- Same encoding as before:
  
  $\text{enc}(s, t) = s \oplus t \oplus x$ -- couple edges

- Using notation $\lambda^S = \prod_{e \in S} \lambda_e$:
  
  $\lambda^s \lambda^t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda^{\text{enc}(s, t)} \lambda^x$

- Similarly: couple edges
  
  $\lambda^s \lambda^t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda^{\text{enc}(s, t)} \lambda^y$
Theorem [Jerrum-Sinclair]
Mixing time is $\text{poly}(n, \lambda_{\text{max}})$.

Proof: for the $x \rightarrow y$ transition:

- Same encoding as before:
  $$\text{enc}(s, t) = s \oplus t \oplus x$$

- Using notation $\lambda^S = \prod_{e \in S} \lambda_e$:
  $$\lambda^S \lambda^t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda^{\text{enc}(s, t)} \lambda^x$$

- Similarly:
  $$\lambda^S \lambda^t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda^{\text{enc}(s, t)} \lambda^y$$

- Using Metropolis filter we get
  $$Q(x, y) \geq \frac{\min\{\mu(x), \mu(y)\}}{\text{poly}(n)}$$
**Theorem** [Jerrum-Sinclair]

Mixing time is $\text{poly}(n, \lambda_{\text{max}})$. 

Proof: for the $x \rightarrow y$ transition:

- Same encoding as before:
  \[ \text{enc}(s, t) = s \oplus t \oplus x - \text{couple edges} \]

- Using notation $\lambda^S = \prod_{e \in S} \lambda_e$:
  \[ \lambda^S \lambda^t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda^{\text{enc}(s, t)} \lambda^x \]

- Similarly: couple edges
  \[ \lambda^S \lambda^t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda^{\text{enc}(s, t)} \lambda^y \]

- Using Metropolis filter we get
  \[ Q(x, y) \geq \frac{\min\{\mu(x), \mu(y)\}}{\text{poly}(n)} \]

- So we have
  \[ \mu(s) \mu(t) \leq \text{poly}(n, \lambda_{\text{max}}) \cdot \mu(\text{enc}(s, t)) Q(x, y) \]
Theorem [Jerrum-Sinclair]
Mixing time is \(\text{poly}(n, \lambda_{\text{max}})\).

Proof: for the \(x \rightarrow y\) transition:

- Same encoding as before:
  \[
  \text{enc}(s, t) = s \oplus t \oplus x - \text{couple edges}
  \]

- Using notation \(\lambda^S = \prod_{e \in S} \lambda_e\):
  \[
  \lambda^s \lambda^t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda^{\text{enc}(s, t)} \lambda^x
  \]

- Similarly: couple edges
  \[
  \lambda^s \lambda^t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda^{\text{enc}(s, t)} \lambda^y
  \]

- Using Metropolis filter we get
  \[
  Q(x, y) \geq \frac{\min\{\mu(x), \mu(y)\}}{\text{poly}(n)}
  \]

- So we have \(\mu(s)\mu(t) \leq \text{poly}(n, \lambda_{\text{max}}) \cdot \mu(\text{enc}(s, t)) Q(x, y)\)
Theorem [Jerrum-Sinclair]
Mixing time is \( \text{poly}(n, \lambda_{\text{max}}) \).

Proof: for the \( x \rightarrow y \) transition:

- Same encoding as before:
  \[
  \text{enc}(s, t) = s \oplus t \oplus x - \text{couple edges}
  \]
- Using notation \( \lambda^S = \prod_{e \in S} \lambda_e \):
  \[
  \lambda^s \lambda^t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda^{\text{enc}(s, t)} \lambda^x
  \]
- Similarly: couple edges
  \[
  \lambda^s \lambda^t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda^{\text{enc}(s, t)} \lambda^y
  \]
- Using Metropolis filter we get
  \[
  Q(x, y) \geq \frac{\min\{\mu(x), \mu(y)\}}{\text{poly}(n)}
  \]
- So we have \( \mu(s)\mu(t) \leq \text{poly}(n, \lambda_{\text{max}}) \cdot \mu(\text{enc}(s, t))Q(x, y) \)

What if we want perfect matchings?
- Idea 1: restrict chain to perfect and near-perfect matchings.
- Idea 2: set \( \lambda_e = \lambda \) very large.

Dist of matching size:

\[
\begin{array}{cccccc}
0 & 1 & 2 & \cdots & n/2 & n/2+1 & \cdots & n
\end{array}
\]

If \( \lambda_k \cdot \#(k\text{-matchings}) \) maximized for \( k = n/2 \), use rejection sampling.
Theorem [Jerrum-Sinclair]

Mixing time is $\text{poly}(n, \lambda_{\text{max}})$.

Proof: for the $x \rightarrow y$ transition:
- Same encoding as before:
  $$\text{enc}(s, t) = s \oplus t \oplus x - \text{couple edges}$$
- Using notation $\lambda^S = \prod_{e \in S} \lambda_e$:
  $$\lambda^s \lambda^t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda^{\text{enc}(s, t)} \lambda^x$$
- Similarly: couple edges
  $$\lambda^s \lambda^t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda^{\text{enc}(s, t)} \lambda^y$$
- Using Metropolis filter we get
  $$Q(x, y) \geq \frac{\min\{\mu(x), \mu(y)\}}{\text{poly}(n)}$$
- So we have $\mu(s)\mu(t) \leq$
  $$\text{poly}(n, \lambda_{\text{max}}) \cdot \mu(\text{enc}(s, t)) Q(x, y)$$

- What if we want perfect matchings?
- Idea 1: restrict chain to perfect and near-perfect matchings.
  one fewer edge
- Idea 2: set $\lambda_e = \lambda$ very large.
Theorem [Jerrum-Sinclair]

Mixing time is $\text{poly}(n, \lambda_{\text{max}})$.

Proof: for the $x \rightarrow y$ transition:

- Same encoding as before:
  \[ \text{enc}(s, t) = s \oplus t \oplus x - \text{couple edges} \]

- Using notation $\lambda^S = \prod_{e \in S} \lambda_e$:
  \[ \lambda^s \lambda^t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda^{\text{enc}(s, t)} \lambda^x \]

- Similarly: couple edges
  \[ \lambda^s \lambda^t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda^{\text{enc}(s, t)} \lambda^y \]

- Using Metropolis filter we get
  \[ Q(x, y) \geq \frac{\min\{\mu(x), \mu(y)\}}{\text{poly}(n)} \]

- So we have $\mu(s)\mu(t) \leq \text{poly}(n, \lambda_{\text{max}}) \cdot \mu(\text{enc}(s, t))Q(x, y)$

- What if we want perfect matchings?
  - Idea 1: restrict chain to perfect and near-perfect matchings.
    - one fewer edge
  - Idea 2: set $\lambda_e = \lambda$ very large.

- Dist of matching size:
  - If $\lambda_k \cdot \#(k\text{-matchings})$ maximized for $k = n/2$, use rejection sampling.
**Theorem [Jerrum-Sinclair]**

Mixing time is $\text{poly}(n, \lambda_{\text{max}})$.

Proof: for the $x \rightarrow y$ transition:

- Same encoding as before: $\text{enc}(s, t) = s \oplus t \oplus x$ — couple edges
- Using notation $\lambda^S = \prod_{e \in S} \lambda_e$:
  \[
  \lambda^s \lambda^t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda^\text{enc}(s, t) \lambda^x
  \]
- Similarly: couple edges
  \[
  \lambda^s \lambda^t \leq \text{poly}(\lambda_{\text{max}}) \cdot \lambda^\text{enc}(s, t) \lambda^y
  \]
- Using Metropolis filter we get
  \[
  Q(x, y) \geq \frac{\min\{\mu(x), \mu(y)\}}{\text{poly}(n)}
  \]
- So we have $\mu(s)\mu(t) \leq \text{poly}(n, \lambda_{\text{max}}) \cdot \mu(\text{enc}(s, t))Q(x, y)$

- What if we want perfect matchings?
- **Idea 1**: restrict chain to perfect and near-perfect matchings.
  - one fewer edge
- **Idea 2**: set $\lambda_e = \lambda$ very large.
- Dist of matching size:

![Matching Size Distribution](image)

- If $\lambda^k \cdot \#(k\text{-matchings})$ maximized for $k = \frac{n}{2}$, use rejection sampling.
Fact: log-concavity of matchings

If $m_k$ is $\#(k\text{-matchings})$, then

$$\frac{m_1}{m_0} \leq \frac{m_2}{m_1} \leq \ldots \leq \frac{m_{n/2-1}}{m_{n/2}}$$

So just need to set $\lambda \geq \frac{m_{n/2} - 1}{m_{n/2}}$

Corollary: if $m_{n/2} - 1 \leq \text{poly}(n) \cdot m_{n/2}$ can sample perfect matchings.

Note: same cond for idea 1.

Bad example: chain of boxes

There are bad examples.

In chain of boxes, we have 1 perfect and $2^{\Omega(n)}$ near-perfect matchings.

Exercise: modify chain of boxes to get slow mixing for idea 1.

Idea: since there can be many more near-perfect matchings, why not reweigh matchings based on size?

[Jerrum-Sinclair-Vigoda'04] showed this works on bipartite graphs.
Fact: log-concavity of matchings

If \( m_k \) is \( \#(k\text{-matchings}) \), then

\[
\frac{m_1}{m_0} \leq \frac{m_2}{m_1} \leq \ldots \leq \frac{m_{n/2-1}}{m_{n/2}}
\]

So just need to set

\[
\lambda \geq \frac{m_{n/2-1}}{m_{n/2}}
\]
Fact: log-concavity of matchings

If $m_k$ is $(k\text{-matchings})$, then

$$\frac{m_1}{m_0} \leq \frac{m_2}{m_1} \leq \ldots \leq \frac{m_{n/2-1}}{m_{n/2}}$$

- So just need to set
  $$\lambda \geq \frac{m_{n/2-1}}{m_{n/2}}$$

- Corollary: if
  $$m_{n/2-1} \leq \text{poly}(n) \cdot m_{n/2}$$
  can sample perfect matchings.
**Fact: log-concavity of matchings**

If $m_k$ is $\#(k\text{-matchings})$, then

$$\frac{m_1}{m_0} \leq \frac{m_2}{m_1} \leq \ldots \leq \frac{m_{n/2-1}}{m_{n/2}}$$

- So just need to set
  $$\lambda \geq \frac{m_{n/2-1}}{m_{n/2}}$$

- Corollary: if
  $$m_{n/2-1} \leq \text{poly}(n) \cdot m_{n/2}$$
  can sample perfect matchings.

- Note: same cond for idea 1.
Fact: log-concavity of matchings

If \( m_k \) is \( \#(k\text{-matchings}) \), then

\[
\frac{m_1}{m_0} \leq \frac{m_2}{m_1} \leq \ldots \leq \frac{m_{n/2-1}}{m_{n/2}}
\]

- So just need to set
  \[
  \lambda \geq \frac{m_{n/2-1}}{m_{n/2}}
  \]
- Corollary: if
  \[
  m_{n/2-1} \leq \text{poly}(n) \cdot m_{n/2}
  \]
  can sample perfect matchings.
- Note: same cond for idea 1.

Bad example: chain of boxes

There are bad examples. In chain of boxes, we have 1 perfect and \( \Omega(n) \) near-perfect matchings.

Exercise: modify chain of boxes to get slow mixing for idea 1.

Idea: since there can be many more near-perfect matchings, why not reweigh matchings based on size?

[Jerrum-Sinclair-Vigoda'04] showed this works on bipartite graphs.
Fact: log-concavity of matchings

If $m_k$ is $\#(k\text{-matchings})$, then

$$\frac{m_1}{m_0} \leq \frac{m_2}{m_1} \leq \ldots \leq \frac{m_{n/2-1}}{m_{n/2}}$$

- So just need to set
  $$\lambda \geq \frac{m_{n/2-1}}{m_{n/2}}$$

- Corollary: if
  $$m_{n/2-1} \leq \text{poly}(n) \cdot m_{n/2}$$
  can sample perfect matchings.

- Note: same cond for idea 1.

Bad example: chain of boxes

There are bad examples. 😞

Idea: since there can be many more near-perfect matchings, why not reweigh matchings based on size?

[Jerrum-Sinclair-Vigoda’04] showed this works on bipartite graphs.
Fact: log-concavity of matchings

If \( m_k \) is \( \#(k\text{-matchings}) \), then

\[
\frac{m_1}{m_0} \leq \frac{m_2}{m_1} \leq \cdots \leq \frac{m_{n/2-1}}{m_{n/2}}
\]

- So just need to set

\[
\lambda \geq \frac{m_{n/2-1}}{m_{n/2}}
\]

- Corollary: if

\[
m_{n/2-1} \leq \text{poly}(n) \cdot m_{n/2}
\]

  can sample perfect matchings.

- Note: same cond for idea 1.

There are bad examples. 😞

In chain of boxes, we have 1 perfect and \( 2^{\Omega(n)} \) near-perfect matchings.

Bad example: chain of boxes
Fact: log-concavity of matchings

If $m_k$ is $\#(k\text{-matchings})$, then

$$\frac{m_1}{m_0} \leq \frac{m_2}{m_1} \leq \cdots \leq \frac{m_{n/2-1}}{m_{n/2}}$$

- So just need to set
  $$\lambda \geq \frac{m_{n/2-1}}{m_{n/2}}$$

- Corollary: if
  $$m_{n/2-1} \leq \text{poly}(n) \cdot m_{n/2}$$
  can sample perfect matchings.

- Note: same cond for idea 1.

There are bad examples. 😞

In chain of boxes, we have 1 perfect and $2^{\Omega(n)}$ near-perfect matchings.

Exercise: modify chain of boxes to get slow mixing for idea 1.

Bad example: chain of boxes

![Diagram of a chain of boxes with some perfect and near-perfect matchings indicated.]
Fact: log-concavity of matchings

If $m_k$ is $\#(k$-matchings), then

$$\frac{m_1}{m_0} \leq \frac{m_2}{m_1} \leq \ldots \leq \frac{m_{n/2-1}}{m_{n/2}}$$

- So just need to set

  $$\lambda \geq \frac{m_{n/2-1}}{m_{n/2}}$$

- Corollary: if

  $$m_{n/2-1} \leq \text{poly}(n) \cdot m_{n/2}$$

  can sample perfect matchings.

- Note: same cond for idea 1.

There are bad examples. 😞

In chain of boxes, we have 1 perfect and $2^{\Omega(n)}$ near-perfect matchings.

Exercise: modify chain of boxes to get slow mixing for idea 1.

Idea: since there can be many more near-perfect matchings, why not reweigh matchings based on size?

Bad example: chain of boxes

![Chain of boxes diagram](image)
Fact: log-concavity of matchings

If \( m_k \) is \(#(k\text{-matchings})\), then

\[
\frac{m_1}{m_0} \leq \frac{m_2}{m_1} \leq \cdots \leq \frac{m_{n/2-1}}{m_{n/2}}
\]

- So just need to set
  \[ \lambda \geq \frac{m_{n/2-1}}{m_{n/2}} \]

- Corollary: if
  \[ m_{n/2-1} \leq \text{poly}(n) \cdot m_{n/2} \]
  can sample perfect matchings.
- Note: same cond for idea 1.

There are bad examples. 😞

In chain of boxes, we have 1 perfect and \( 2^{\Omega(n)} \) near-perfect matchings.

Exercise: modify chain of boxes to get slow mixing for idea 1.

Idea: since there can be many more near-perfect matchings, why not reweigh matchings based on size?

Bad example: chain of boxes

[Jerrum-Sinclair-Vigoda’04] showed this works on bipartite graphs.
Let $\Omega_S$ be the class of matchings whose monomers are $S$. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss $u, v$. 

Theorem [Jerrum-Sinclair-Vigoda'04] If $P$ is Metropolis walk restricted to perfect and near-perfect matchings weighted $\propto \mu$, and graph is bipartite, $t$ mixes $(P, 1)$-max-weight $M = \text{poly}(n)$. We need max-weight $M$ as start to ensure $\log \chi_2(\nu_0 \parallel \mu) = \text{poly}(n)$. 

The Chicken-and-Egg Problem How can we compute $\lambda(\Omega_S)$? By sampling. How to sample? Use counting.
Let $\Omega_S$ be the class of matchings whose monomers are $S$. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss $u,v$.

Let $\lambda^M$ denote monomer-dimer weight of $M$:

$$\lambda^M = \prod_{e \in M} \lambda_e$$
Let $\Omega_S$ be the class of matchings whose monomers are $S$. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss $u$, $v$.

Let $\lambda^M$ denote monomer-dimer weight of $M$:

$$\lambda^M = \prod_{e \in M} \lambda_e$$

We get weights for each class:

$$\lambda(\Omega_S) = \sum_{M \in \Omega_S} \lambda^M$$
Let $\Omega_S$ be the class of matchings whose monomers are $S$. Example: $\Omega_{\emptyset}$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss $u, v$.

Let $\lambda^M$ denote monomer-dimer weight of $M$:

$$\lambda^M = \prod_{e \in M} \lambda_e$$

We get weights for each class:

$$\lambda(\Omega_S) = \sum_{M \in \Omega_S} \lambda^M$$

Define modified distribution on matchings:

$$\mu(M) \propto \frac{\lambda^M}{\lambda(\Omega_{\text{monomers}(M)})}$$
Let $\Omega_S$ be the class of matchings whose monomers are $S$. Example: $\Omega_{\emptyset}$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss $u, v$.

Let $\lambda^M$ denote monomer-dimer weight of $M$:

$$\lambda^M = \prod_{e \in M} \lambda_e$$

We get weights for each class:

$$\lambda(\Omega_S) = \sum_{M \in \Omega_S} \lambda^M$$

Define modified distribution on matchings:

$$\mu(M) \propto \frac{\lambda^M}{\lambda(\Omega_{\text{monomers}(M)})}$$

Note: $\mu(\Omega_S)$ is the same for all $S$. 

Theorem [Jerrum-Sinclair-Vigoda’04]

If $P$ is Metropolis walk restricted to perfect and near-perfect matchings weighted $\propto \mu$, and graph is bipartite $t \text{mix}(P,1) \text{max-weight } \mu = \text{poly}(n)$.

We need max-weight $M$ as start to ensure $\log \chi^2(\nu_0, \|\mu\|) = \text{poly}(n)$. 

The Chicken-and-Egg Problem

How can we compute $\lambda(\Omega_S)$? By sampling. How to sample? Use counting.
Let $\Omega_S$ be the class of matchings whose monomers are $S$. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss $u, v$.

Let $\lambda^M$ denote monomer-dimer weight of $M$:

$$\lambda^M = \prod_{e \in M} \lambda_e$$

We get weights for each class:

$$\lambda(\Omega_S) = \sum_{M \in \Omega_S} \lambda^M$$

Define modified distribution on matchings:

$$\mu(M) \propto \frac{\lambda^M}{\lambda(\Omega_{\text{monomers}(M)})}$$

Note: $\mu(\Omega_S)$ is the same for all $S$. 

**Theorem [Jerrum-Sinclair-Vigoda’04]**

If $P$ is Metropolis walk restricted to perfect and near-perfect matchings weighted $\propto \mu$, and graph is bipartite

$$t_{\text{mix}}(P, 1_{\text{max-weight } M}) = \text{poly}(n)$$
Let $\Omega_S$ be the class of matchings whose monomers are $S$. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss $u, v$.

Let $\lambda^M$ denote monomer-dimer weight of $M$:

$$\lambda^M = \prod_{e \in M} \lambda_e$$

We get weights for each class:

$$\lambda(\Omega_S) = \sum_{M \in \Omega_S} \lambda^M$$

Define modified distribution on matchings:

$$\mu(M) \propto \frac{\lambda^M}{\lambda(\Omega_{\text{monomers}(M)})}$$

Note: $\mu(\Omega_S)$ is the same for all $S$.

---

**Theorem [Jerrum-Sinclair-Vigoda’04]**

If $P$ is Metropolis walk restricted to perfect and near-perfect matchings weighted $\propto \mu$, and graph is bipartite

$$t_{\text{mix}}(P, 1_{\text{max-weight } M}) = \text{poly}(n)$$

We need max-weight $M$ as start to ensure $\log \chi^2(\nu_0 \parallel \mu) = \text{poly}(n)$. 

---

The Chicken-and-Egg Problem

How can we compute $\lambda(\Omega_S)$?

By sampling.

How to sample?

Use counting.
Let $\Omega_S$ be the class of matchings whose monomers are $S$. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss $u,v$.

Let $\lambda^M$ denote monomer-dimer weight of $M$:

$$\lambda^M = \prod_{e \in M} \lambda_e$$

We get weights for each class:

$$\lambda(\Omega_S) = \sum_{M \in \Omega_S} \lambda^M$$

Define modified distribution on matchings:

$$\mu(M) \propto \frac{\lambda^M}{\lambda(\Omega_{\text{monomers}(M)})}$$

Note: $\mu(\Omega_S)$ is the same for all $S$.

Theorem [Jerrum-Sinclair-Vigoda’04]

If $P$ is Metropolis walk restricted to perfect and near-perfect matchings weighted $\propto \mu$, and graph is bipartite

$$t_{\text{mix}}(P, 1_{\text{max-weight } M}) = \text{poly}(n)$$

We need max-weight $M$ as start to ensure $\log \chi^2(\nu_0 \parallel \mu) = \text{poly}(n)$. 

The Chicken-and-Egg Problem
Let $\Omega_S$ be the class of matchings whose monomers are $S$. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss $u,v$.

Let $\lambda^M$ denote monomer-dimer weight of $M$:

$$\lambda^M = \prod_{e \in M} \lambda_e$$

We get weights for each class:

$$\lambda(\Omega_S) = \sum_{M \in \Omega_S} \lambda^M$$

Define modified distribution on matchings:

$$\mu(M) \propto \frac{\lambda^M}{\lambda(\Omega_{\text{monomers}(M)})}$$

Note: $\mu(\Omega_S)$ is the same for all $S$.

---

**Theorem [Jerrum-Sinclair-Vigoda’04]**

If $P$ is Metropolis walk restricted to perfect and near-perfect matchings weighted $\propto \mu$, and graph is bipartite

$$t_{\text{mix}}(P, 1_{\text{max-weight } M}) = \text{poly}(n)$$

We need max-weight $M$ as start to ensure $\log \chi^2(\nu_0 \parallel \mu) = \text{poly}(n)$.

The Chicken-and-Egg Problem

How can we compute $\lambda(\Omega_S)$?
Let $\Omega_S$ be the class of matchings whose monomers are $S$. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss $u, v$.

Let $\lambda^M$ denote monomer-dimer weight of $M$:

\[ \lambda^M = \prod_{e \in M} \lambda_e \]

We get weights for each class:

\[ \lambda(\Omega_S) = \sum_{M \in \Omega_S} \lambda^M \]

Define modified distribution on matchings:

\[ \mu(M) \propto \frac{\lambda^M}{\lambda(\Omega_{\text{monomers}(M)})} \]

Note: $\mu(\Omega_S)$ is the same for all $S$.

**Theorem [Jerrum-Sinclair-Vigoda'04]**

If $P$ is Metropolis walk restricted to perfect and near-perfect matchings weighted $\propto \mu$, and graph is bipartite

\[ t_{\text{mix}}(P, 1_{\text{max-weight } M}) = \text{poly}(n) \]

We need max-weight $M$ as start to ensure $\log \chi^2(\nu_0 \parallel \mu) = \text{poly}(n)$.

The Chicken-and-Egg Problem

How can we compute $\lambda(\Omega_S)$?

By sampling.
Let $\Omega_S$ be the class of matchings whose monomers are $S$. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega\{u,v\}$ matchings that only miss $u,v$.

Let $\lambda^M$ denote monomer-dimer weight of $M$:

$$\lambda^M = \prod_{e \in M} \lambda_e$$

We get weights for each class:

$$\lambda(\Omega_S) = \sum_{M \in \Omega_S} \lambda^M$$

Define modified distribution on matchings:

$$\mu(M) \propto \frac{\lambda^M}{\lambda(\Omega_{\text{monomers}(M)})}$$

Note: $\mu(\Omega_S)$ is the same for all $S$.

**Theorem [Jerrum-Sinclair-Vigoda’04]**

If $P$ is Metropolis walk restricted to perfect and near-perfect matchings weighted $\propto \mu$, and graph is bipartite

$$t_{\text{mix}}(P, I_{\text{max-weight}} M) = \text{poly}(n)$$

We need max-weight $M$ as start to ensure $\log \chi^2(\nu_0 \parallel \mu) = \text{poly}(n)$.

The Chicken-and-Egg Problem

How can we compute $\lambda(\Omega_S)$?

By sampling.

How to sample?
Let $\Omega_S$ be the class of matchings whose monomers are $S$. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss $u, v$.

Let $\lambda^M$ denote monomer-dimer weight of $M$:

$$\lambda^M = \prod_{e \in M} \lambda_e$$

We get weights for each class:

$$\lambda(\Omega_S) = \sum_{M \in \Omega_S} \lambda^M$$

Define modified distribution on matchings:

$$\mu(M) \propto \frac{\lambda^M}{\lambda(\Omega_{\text{monomers}(M)})}$$

Note: $\mu(\Omega_S)$ is the same for all $S$.

Theorem [Jerrum-Sinclair-Vigoda'04]

If $P$ is Metropolis walk restricted to perfect and near-perfect matchings weighted $\propto \mu$, and graph is bipartite

$$t_{\text{mix}}(P, 1_{\text{max-weight } M}) = \text{poly}(n)$$

We need max-weight $M$ as start to ensure $\log \chi^2(\nu_0 \parallel \mu) = \text{poly}(n)$.

The Chicken-and-Egg Problem

How can we compute $\lambda(\Omega_S)$?

By sampling.

How to sample?

Use counting. 😞
Resolving the chicken-and-egg problem: gradual change.
Resolving the chicken-and-egg problem: gradual change.

**Theorem**

Chain mixes fast even if $\lambda(\Omega_S)$ are replaced by approximations in $\mu$.

Say factor $10$ approx
Resolving the chicken-and-egg problem: gradual change.

**Theorem**

Chain mixes fast even if $\lambda(\Omega_S)$ are replaced by approximations in $\mu$.

say factor 10 approx

Start with easy case. Take $G = K_{n/2,n/2}$, and $\lambda_e = 1$. 
Resolving the chicken-and-egg problem: gradual change.

**Theorem**
Chain mixes fast even if $\lambda(\Omega_S)$ are replaced by approximations in $\mu$.

say factor 10 approx

Start with easy case. Take $G = K_{n/2, n/2}$, and $\lambda_e = 1$.

Slowly change $\lambda_e$s:

$\lambda^{(0)} \rightarrow \lambda^{(1)} \rightarrow \cdots \rightarrow \lambda^{(t)}$

by $1 \pm 1/n$ each time
Resolving the chicken-and-egg problem: gradual change.

**Theorem**

Chain mixes fast even if $\lambda(\Omega_S)$ are replaced by approximations in $\mu$.

- Start with easy case. Take $G = K_{n/2, n/2}$, and $\lambda_e = 1$.
- Slowly change $\lambda_e$s:
  
  \[
  \lambda^{(0)} \rightarrow \lambda^{(1)} \rightarrow \ldots \rightarrow \lambda^{(t)}
  \]

  by $1 \pm 1/n$ each time

- Use Markov chain for each $\lambda^{(i)}$ to estimate $\lambda^{(i)}(\Omega_S)$ for $|S| \leq 2$. 
Resolving the chicken-and-egg problem: gradual change.

**Theorem**
Chain mixes fast even if $\lambda(\Omega_S)$ are replaced by approximations in $\mu$.

say factor 10 approx

Start with easy case. Take $G = K_{n/2,n/2}$, and $\lambda_e = 1$.

Slowly change $\lambda_e$s:

$$\lambda^{(0)} \rightarrow \lambda^{(1)} \rightarrow \ldots \rightarrow \lambda^{(t)}$$

by $1 \pm 1/n$ each time

Use Markov chain for each $\lambda^{(i)}$ to estimate $\lambda^{(i)}(\Omega_S)$ for $|S| \leq 2$.

Use estimates to define next $\mu$. 

Note: $\lambda^0_e \approx \exp(-n^2)$
Resolving the chicken-and-egg problem: gradual change.

**Theorem**
Chain mixes fast even if \( \lambda(\Omega_S) \) are replaced by approximations in \( \mu \).

say factor 10 approx

Start with easy case. Take \( G = K_{n/2,n/2} \), and \( \lambda_e = 1 \).

Slowly change \( \lambda_e \)s:

\[
\lambda^{(0)} \rightarrow \lambda^{(1)} \rightarrow \cdots \rightarrow \lambda^{(t)}
\]

by \( 1 \pm 1/n \) each time

Use Markov chain for each \( \lambda^{(i)} \) to estimate \( \lambda^{(i)}(\Omega_S) \) for \( |S| \leq 2 \).

Use estimates to define next \( \mu \).

Note: \( (\lambda_e = 0) \approx (\lambda_e = \exp(-n^2)) \)
Resolving the chicken-and-egg problem: gradual change.

**Theorem**
Chain mixes fast even if $\lambda(\Omega_S)$ are replaced by approximations in $\mu$.

- Start with easy case. Take $G = K_{n/2,n/2}$, and $\lambda_e = 1$.
- Slowly change $\lambda_e$s:
  $\lambda^{(0)} \rightarrow \lambda^{(1)} \rightarrow \cdots \rightarrow \lambda^{(t)}$
by $1 \pm 1/n$ each time
- Use Markov chain for each $\lambda^{(i)}$ to estimate $\lambda^{(i)}(\Omega_S)$ for $|S| \leq 2$.
- Use estimates to define next $\mu$.

Note: $(\lambda_e = 0) \approx (\lambda_e = \exp(-n^2))$
It just remains to prove fast mixing.
Resolving the chicken-and-egg problem: gradual change.

**Theorem**
Chain mixes fast even if \( \lambda(\Omega_S) \) are replaced by approximations in \( \mu \).

Say factor 10 approx

Start with easy case. Take \( G = K_{n/2,n/2} \), and \( \lambda_e = 1 \).

Slowly change \( \lambda_e \) s:

\[
\lambda^{(0)} \rightarrow \lambda^{(1)} \rightarrow \cdots \rightarrow \lambda^{(t)}
\]

by \( 1 \pm 1/n \) each time

Use Markov chain for each \( \lambda^{(i)} \) to estimate \( \lambda^{(i)}(\Omega_S) \) for \( |S| \leq 2 \).

Use estimates to define next \( \mu \).

Note: \( (\lambda_e = 0) \approx (\lambda_e = \exp(-n^2)) \)

It just remains to prove fast mixing.

We use canonical paths. Enough to consider \( s \in \Omega_{\{u,v\}} \) and \( t \in \Omega_{\emptyset} \).
Resolving the chicken-and-egg problem: gradual change.

**Theorem**

Chain mixes fast even if $\lambda(\Omega_S)$ are replaced by approximations in $\mu$.

Say factor 10 approx

Start with easy case. Take $G = K_{n/2,n/2}$, and $\lambda_e = 1$.

Slowly change $\lambda_e$s:

$\lambda^{(0)} \rightarrow \lambda^{(1)} \rightarrow \cdots \rightarrow \lambda^{(t)}$

by $1 \pm 1/n$ each time

Use Markov chain for each $\lambda^{(i)}$ to estimate $\lambda^{(i)}(\Omega_S)$ for $|S| \leq 2$.

Use estimates to define next $\mu$.

Note: $(\lambda_e = 0) \approx (\lambda_e = \exp(-n^2))$

It just remains to prove fast mixing.

We use canonical paths. Enough to consider $s \in \Omega_{\{u,v\}}$ and $t \in \Omega_{\emptyset}$.

Assume $\lambda(\Omega_S)$ is accurate, because the inequality

$\mu(s)\mu(t) \leq C\mu(\text{enc}(s, t))Q(x, y)$

is robust to approximation.
Resolving the chicken-and-egg problem: gradual change.

**Theorem**
Chain mixes fast even if $\lambda(\Omega_S)$ are replaced by approximations in $\mu$.

say factor 10 approx

Start with easy case. Take $G = K_{n/2,n/2}$, and $\lambda_e = 1$.

Slowly change $\lambda_e$s:
$\lambda^0 \to \lambda^1 \to \cdots \to \lambda^t$
by $1 \pm 1/n$ each time

Use Markov chain for each $\lambda^{(i)}$ to estimate $\lambda^{(i)}(\Omega_S)$ for $|S| \leq 2$.

Use estimates to define next $\mu$.

Note: $(\lambda_e = 0) \approx (\lambda_e = \exp(-n^2))$

It just remains to prove fast mixing.

We use canonical paths. Enough to consider $s \in \Omega_{\{u,v\}}$ and $t \in \Omega_\emptyset$.

Assume $\lambda(\Omega_S)$ is accurate, because the inequality
$\mu(s)\mu(t) \leq C\mu(\text{enc}(s, t))Q(x, y)$
is robust to approximation.

Use the same encoding as before:
$\text{enc}(s, t) = s \oplus t \oplus x$ — couple edges
Traverse alternating path first. Ensures all $x$ on the $st$-path are perfect/near-perfect.
Traverse alternating path first. Ensures all $x$ on the $st$-path are perfect/near-perfect.

Issue: encoding might not be perfect/near-perfect:

![Diagram]

This is fine! We still get $\sum |S| \leq 4 \mu(\Omega_S) \leq \text{poly}(n) \cdot \sum |S| \leq 2 \mu(\Omega_S)$.
 Traverse alternating path first. Ensures all $x$ on the $st$-path are perfect/near-perfect.

Issue: encoding might not be perfect/near-perfect:

![Diagram showing the traversal process](image)

This is fine! We still get $\text{cong} \leq \text{poly}(n)$ because

$$\sum_{|S| \leq 4} \mu(\Omega_S) \leq \text{poly}(n) \cdot \sum_{|S| \leq 2} \mu(\Omega_S)$$
Traverse alternating path first. Ensures all $x$ on the $st$-path are perfect/near-perfect.

Issue: encoding might not be perfect/near-perfect:

This is fine! We still get $\text{cong} \leq \text{poly}(n)$ because

$$\sum_{|S| \leq 4} \mu(\Omega S) \leq \text{poly}(n) \cdot \sum_{|S| \leq 2} \mu(\Omega S)$$

We just need to show $\mu(s)\mu(t) \leq \text{poly}(n) \cdot \min\{\mu(x), \mu(y)\} \cdot \mu(\text{enc})$
Traverse alternating path first. Ensures all $x$ on the $st$-path are perfect/near-perfect.

Issue: encoding might not be perfect/near-perfect:

This is fine! We still get $\text{cong} \leq \text{poly}(n)$ because

\[
\sum_{|S| \leq 4} \mu(\Omega_S) \leq \text{poly}(n) \cdot \sum_{|S| \leq 2} \mu(\Omega_S)
\]

We just need to show $\mu(s)\mu(t) \leq \text{poly}(n) \cdot \min\{\mu(x), \mu(y)\} \cdot \mu(\text{enc})$

It’s a bit of case analysis, but hardest case is in the middle of unraveling a cycle:
Note that $\lambda_s \lambda_t = \lambda_e \lambda_f \lambda_x \lambda_{enc}$. Let $e$’s endpoints be $a, a'$ and $f$’s endpoints be $b, b'$. Prove:

via injective map

$\lambda(\Omega_{\emptyset}) \lambda(\Omega_{\{u,v\}}) \geq 1 \text{poly}(n) \cdot \lambda(\Omega_{\{a,b\}}) \lambda(\Omega_{\{u,v,a',b'\}} \text{enc})$

Thus $\mu(s) \mu(t) \leq \text{poly}(n) \cdot \mu(x) \mu(\text{enc})$. Similar ineqs yield $\mu(s) \mu(t) \leq \text{poly}(n) \cdot \mu(y) \mu(\text{enc})$. So cong $\leq \text{poly}(n)$. 
enc = s ⊕ t ⊕ x − e − f

Note that $\lambda_s \lambda_t = \lambda_e \lambda_f \lambda_x$. Let $e$’s endpoints be $a, a'$ and $f$’s endpoints be $b, b'$. Prove:

via injective map $\lambda(\Omega \emptyset t) \geq \lambda(\Omega \{u,v\} s) \lambda(\Omega \emptyset x) \lambda(\Omega \{a,b\} x) \lambda(\Omega \{a',b',u,v\} enc)$.

Thus $\mu(s) \mu(t) \leq \lambda(\Omega \emptyset y) \mu(x) \mu(enc)$. Similar ineqs yield $\mu(s) \mu(t) \leq \lambda(\Omega \emptyset y) \mu(y) \mu(enc)$. So $\cong \leq \lambda(\Omega \emptyset n)$. 
\[
\text{Note that } \lambda^s \lambda^t = \lambda_e \lambda_f \lambda^x \lambda^{\text{enc}}. \text{ Let } e\text{'s endpoints be } a, a' \text{ and } f\text{'s endpoints be } b, b'. \text{ Prove: via injective map}
\]

\[
\lambda(\Omega_\emptyset) \lambda(\Omega_{\{u,v\}}) \geq \frac{1}{\text{poly}(n)} \cdot \lambda_e \lambda_f \lambda(\Omega_{\{a,b\}}) \lambda(\Omega_{\{u,v,a',b'\}})
\]

\[
\text{enc} = s \oplus t \oplus x - e - f
\]
Note that $\lambda^s \lambda^t = \lambda_e \lambda_f \lambda^x \lambda^{\text{enc}}$. Let $e$’s endpoints be $a, a'$ and $f$’s endpoints be $b, b'$. Prove: via injective map

$$\lambda(\Omega_\emptyset) \lambda(\Omega_{\{u,v\}}) \geq \frac{1}{\text{poly}(n)} \cdot \lambda_e \lambda_f \lambda(\Omega_{\{a,b\}}) \lambda(\Omega_{\{u,v,a',b'\}}).$$

Thus $\mu(s) \mu(t) \leq \text{poly}(n) \cdot \mu(x) \mu(\text{enc})$. Similar inequations yield $\mu(s) \mu(t) \leq \text{poly}(n) \cdot \mu(y) \mu(\text{enc})$. So $\text{cong} \leq \text{poly}(n)$. 