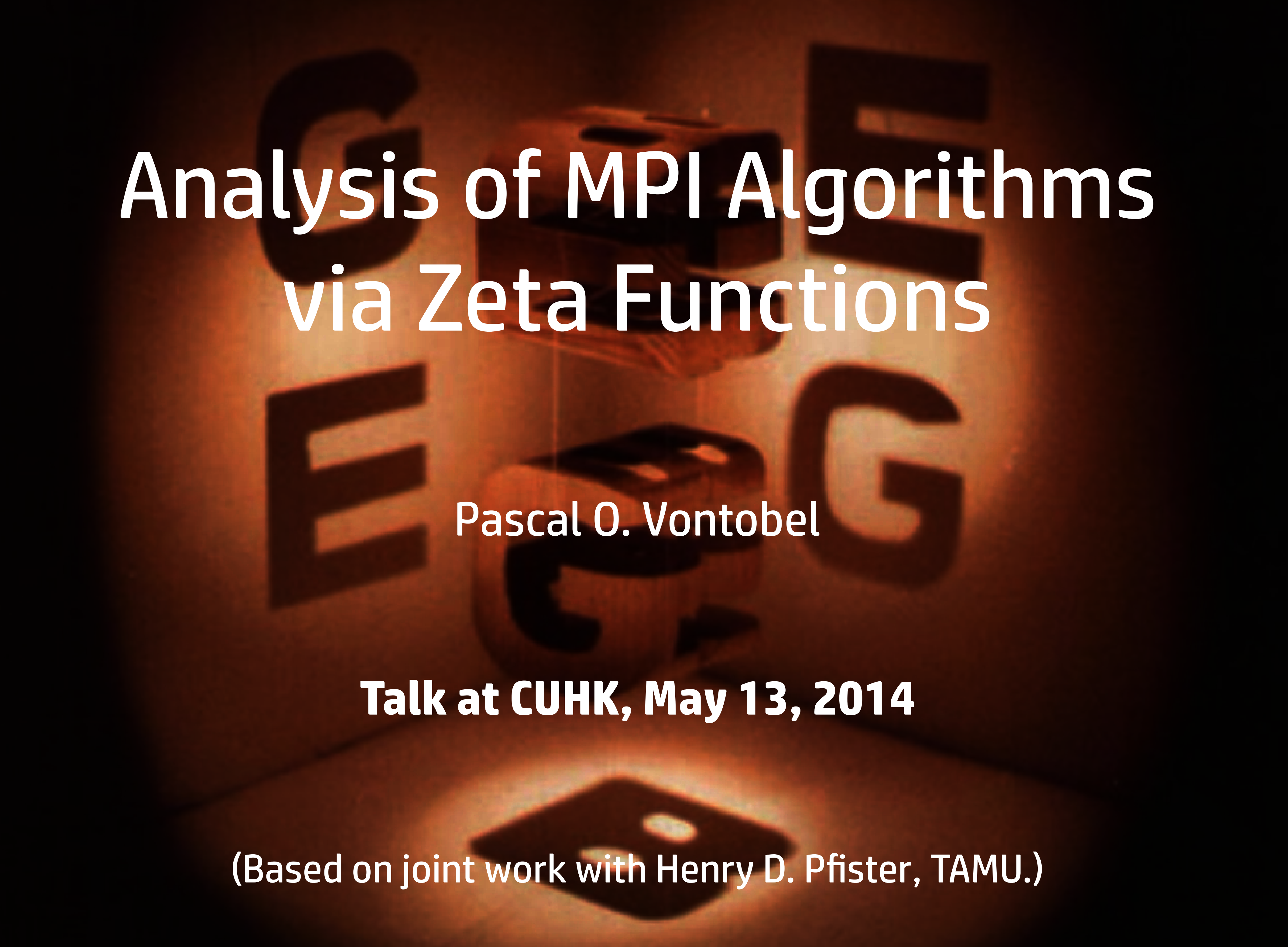


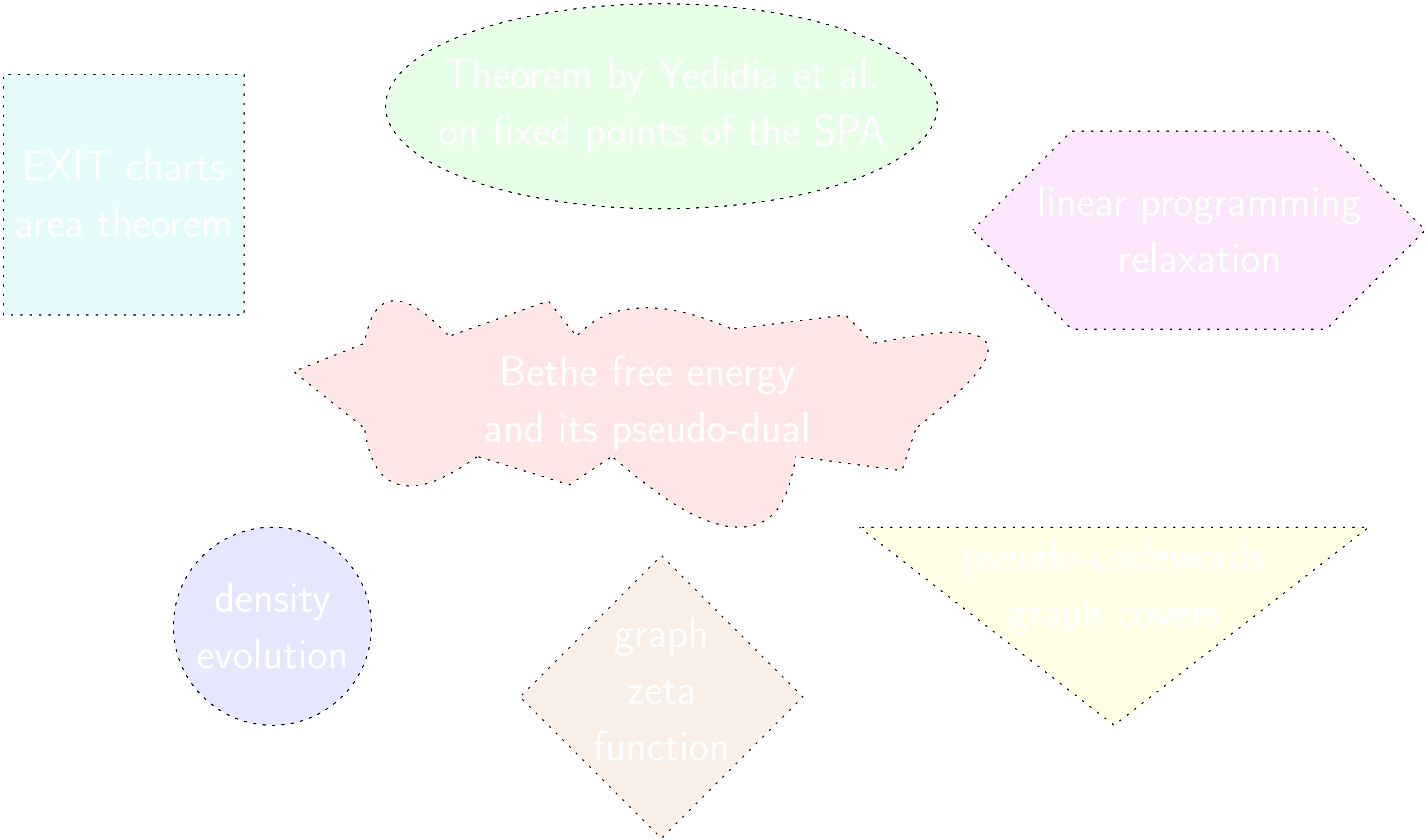
Analysis of MPI Algorithms via Zeta Functions

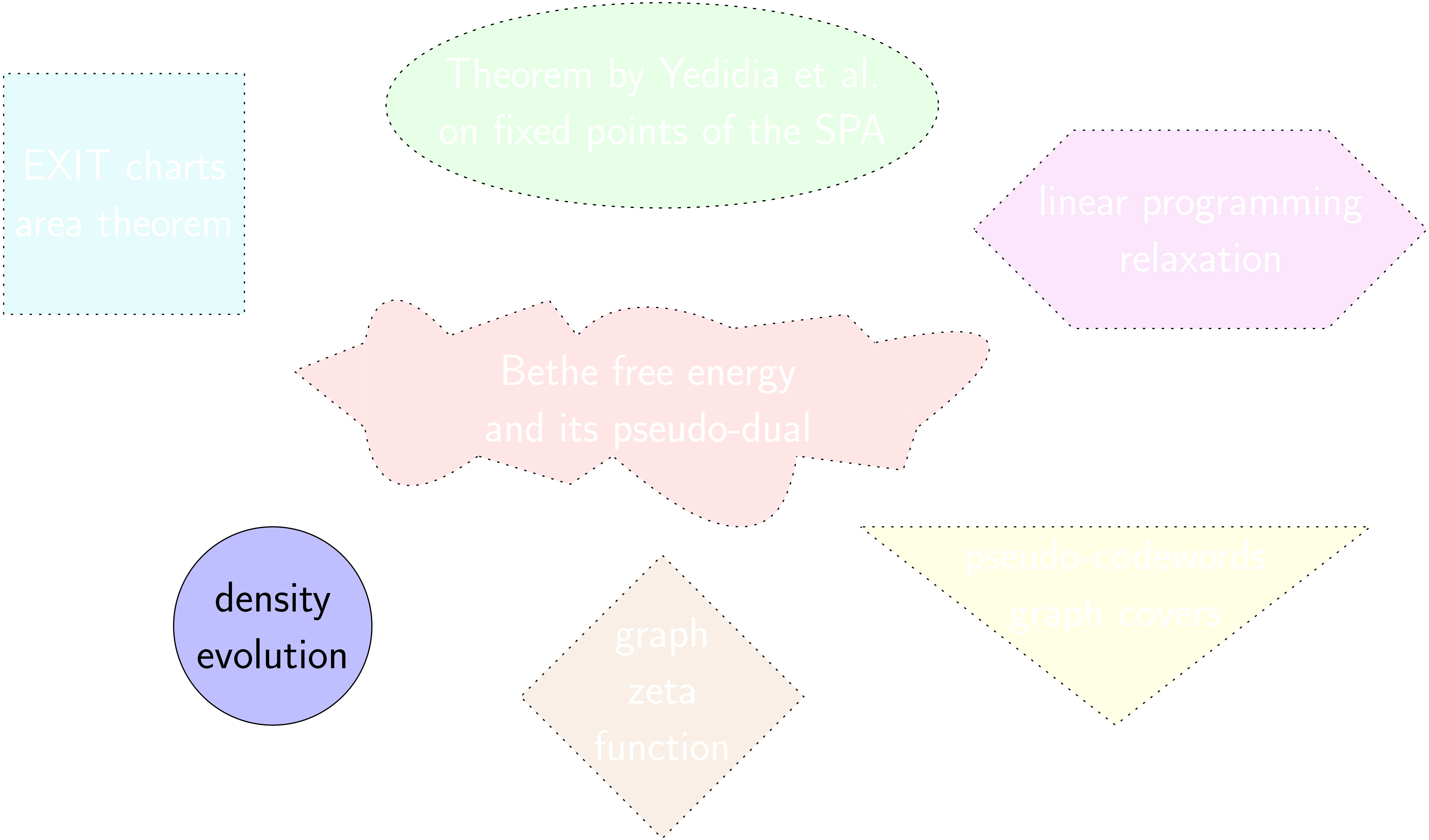
Pascal O. Vontobel

Talk at CUHK, May 13, 2014

(Based on joint work with Henry D. Pfister, TAMU.)

G
E
E
G
B


EXIT charts
area theorem
Theorem by Yedidia et al.
on fixed points of the SPA
linear programming
relaxation
Bethe free energy
and its pseudo-dual
density
evolution
graph
zeta
function
pseudo-codewords
graph covers



EXIT charts
area theorem

Theorem by Yedidia et al.
on fixed points of the SPA

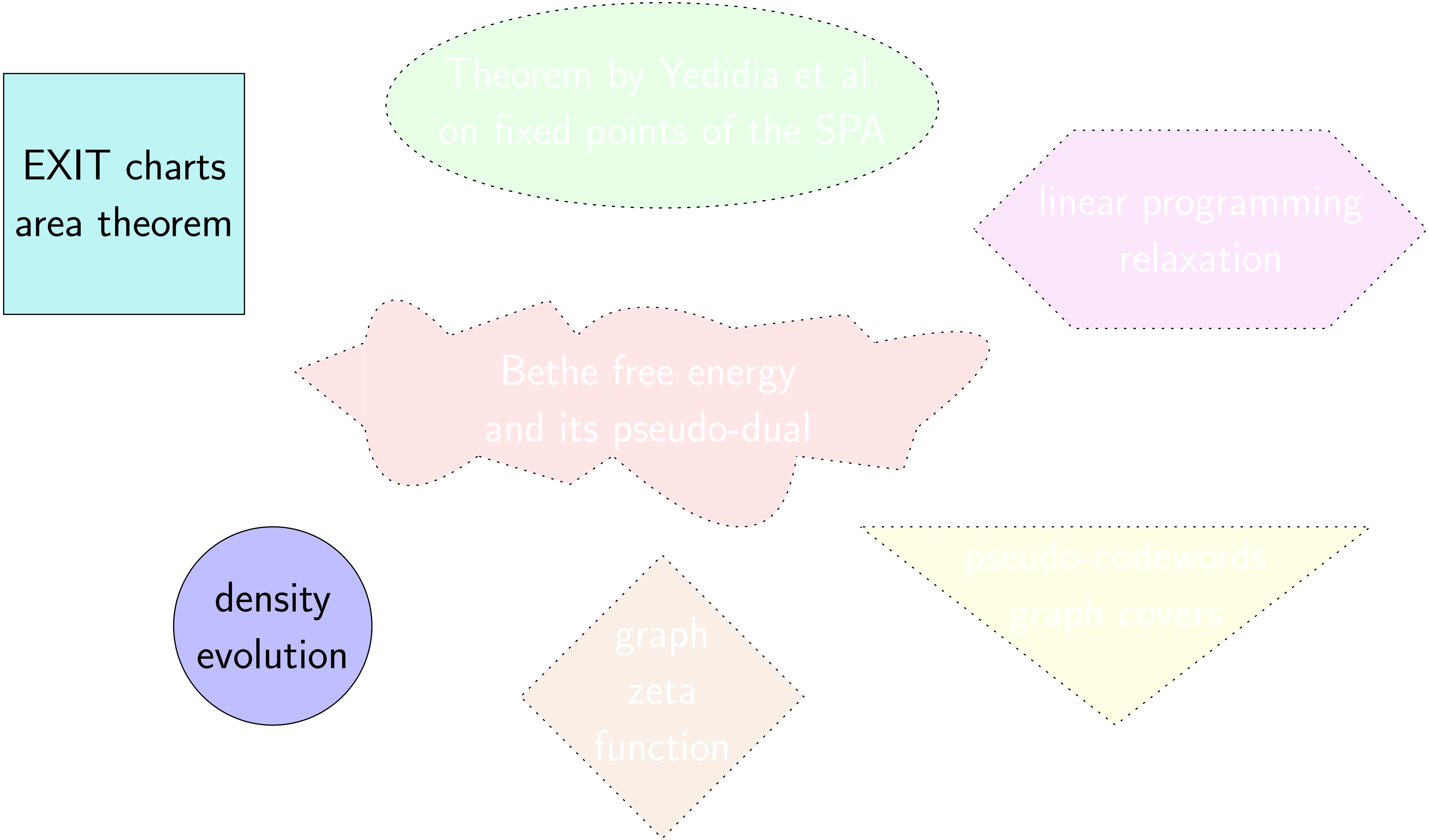
linear programming
relaxation

Bethe free energy
and its pseudo-dual

density
evolution

graph
zeta
function

pseudo-codewords
graph covers



EXIT charts
area theorem

Theorem by Yedidia et al.
on fixed points of the SPA

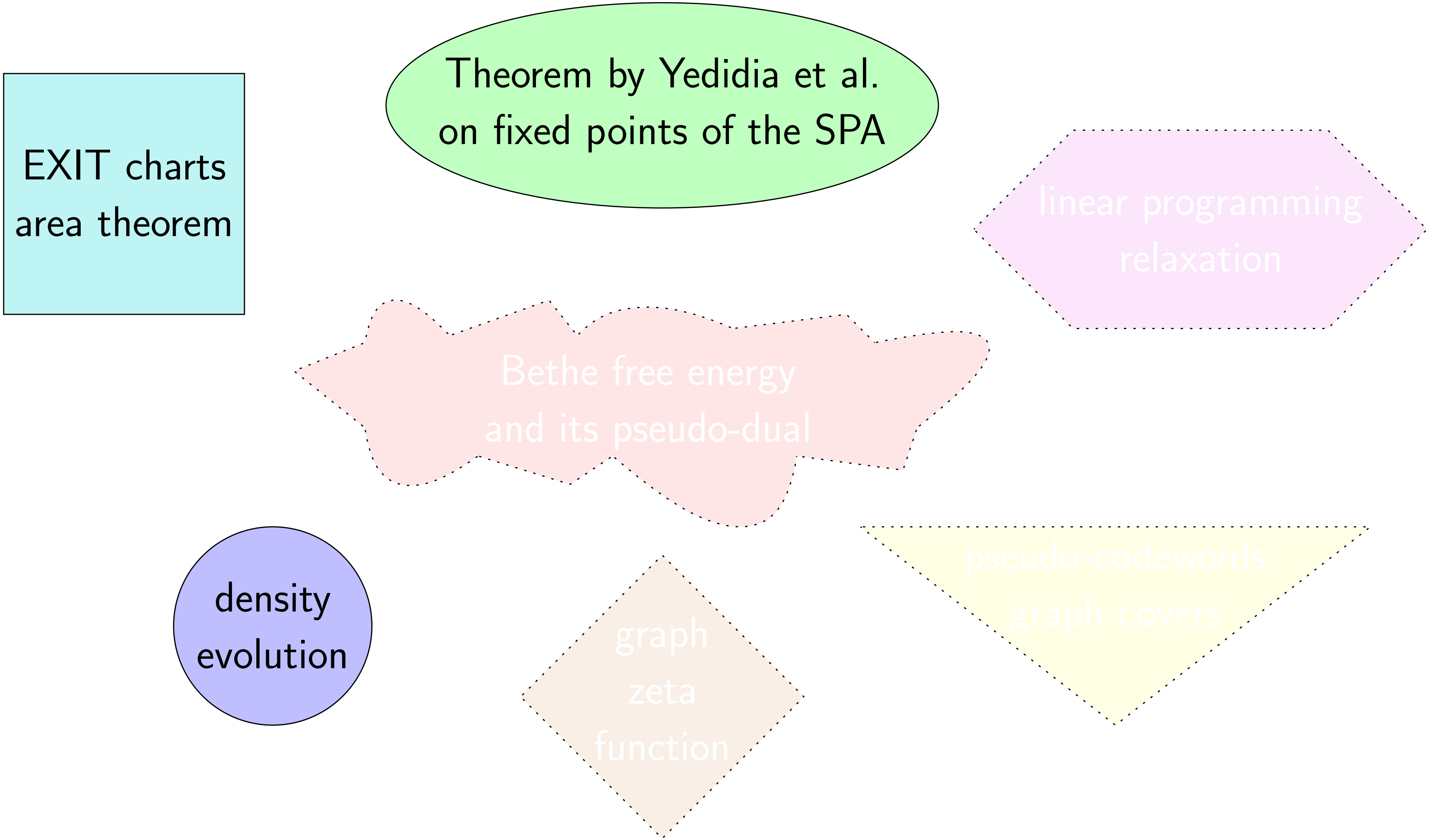
linear programming
relaxation

Bethe free energy
and its pseudo-dual

density
evolution

pseudo-codewords
graph covers

graph
zeta
function



EXIT charts
area theorem

Theorem by Yedidia et al.
on fixed points of the SPA

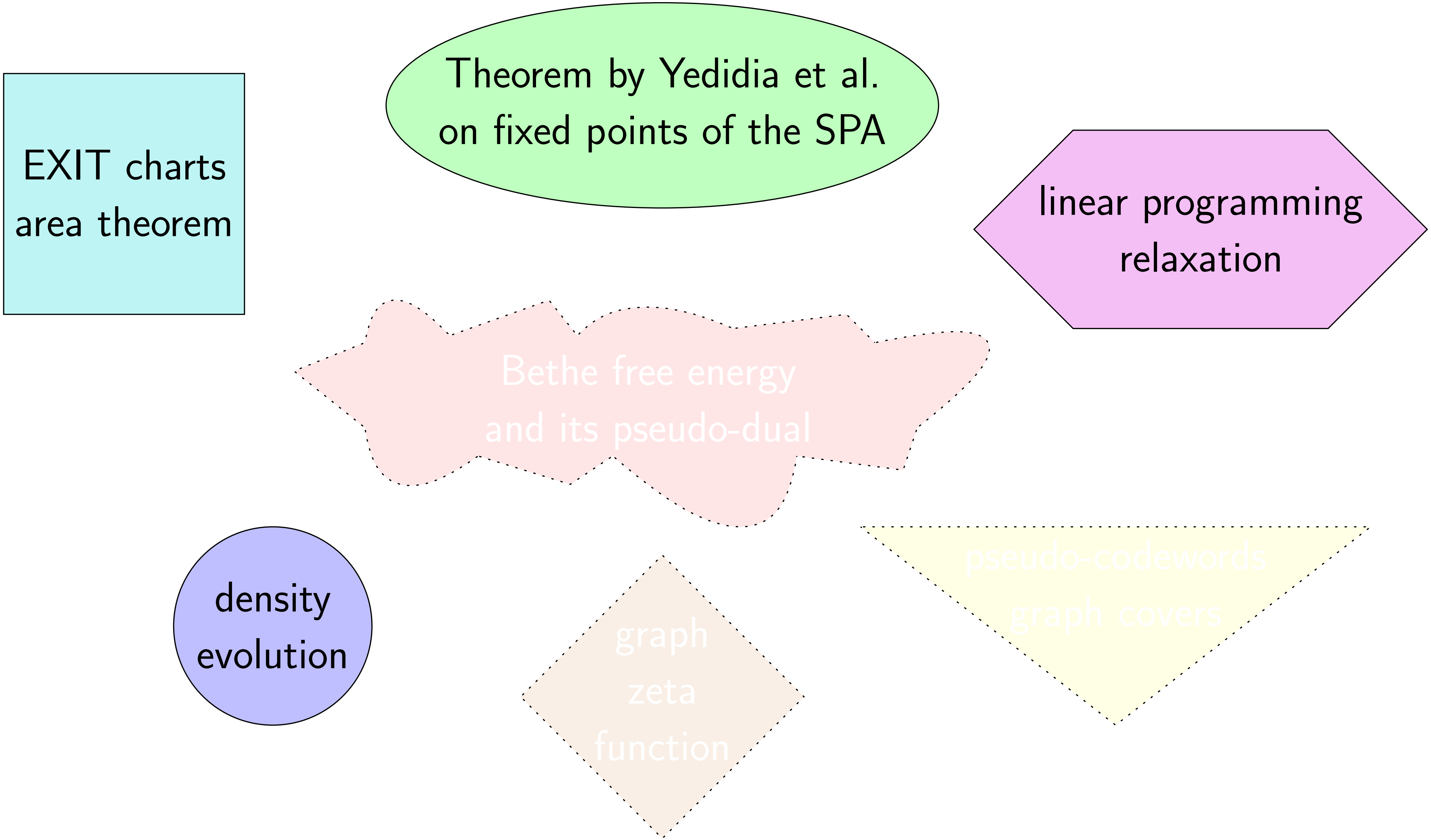
linear programming
relaxation

Bethe free energy
and its pseudo-dual

density
evolution

pseudo-codewords
graph covers

graph
zeta
function



EXIT charts
area theorem

Theorem by Yedidia et al.
on fixed points of the SPA

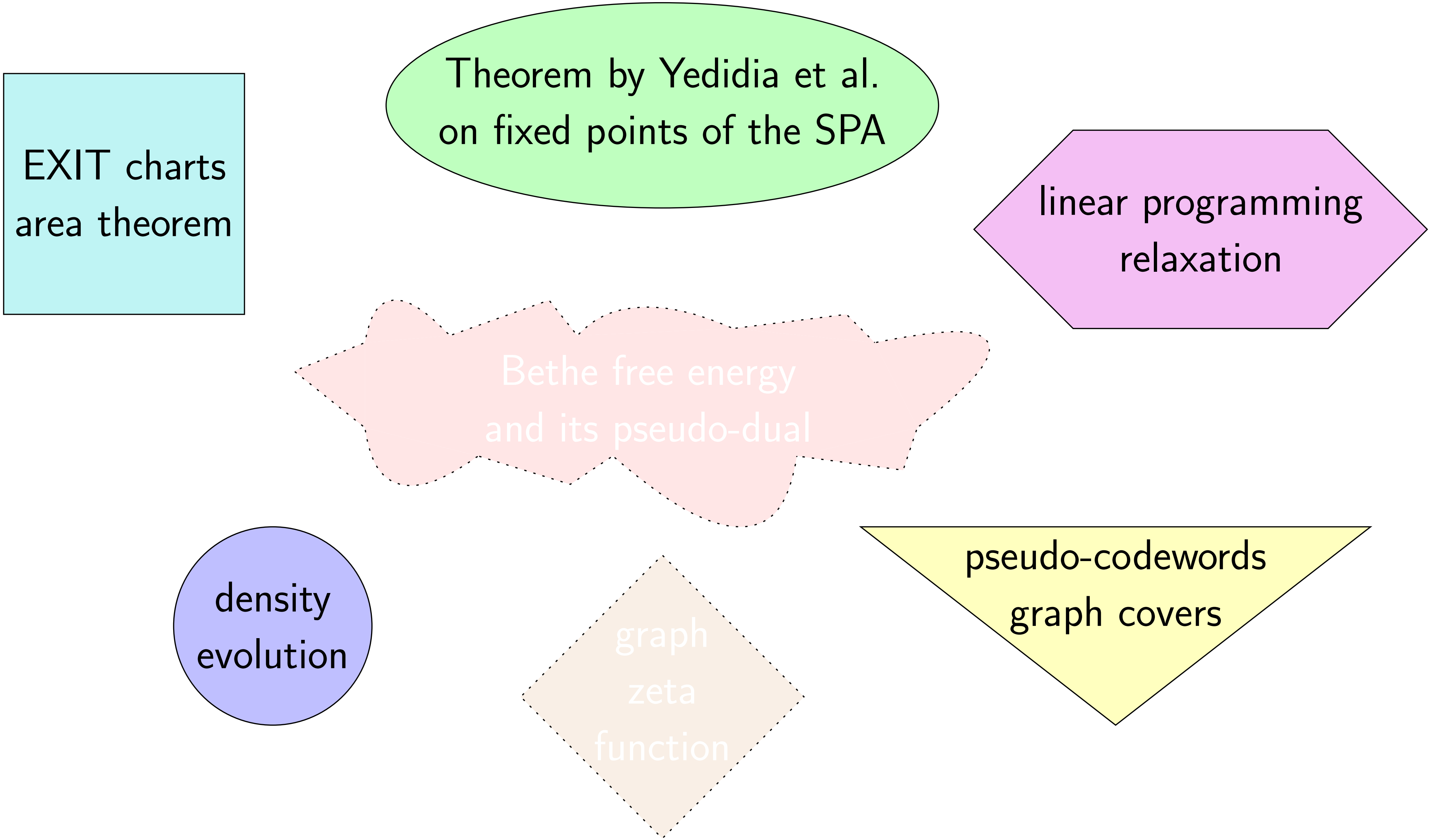
linear programming
relaxation

Bethe free energy
and its pseudo-dual

density
evolution

pseudo-codewords
graph covers

graph
zeta
function



EXIT charts
area theorem

Theorem by Yedidia et al.
on fixed points of the SPA

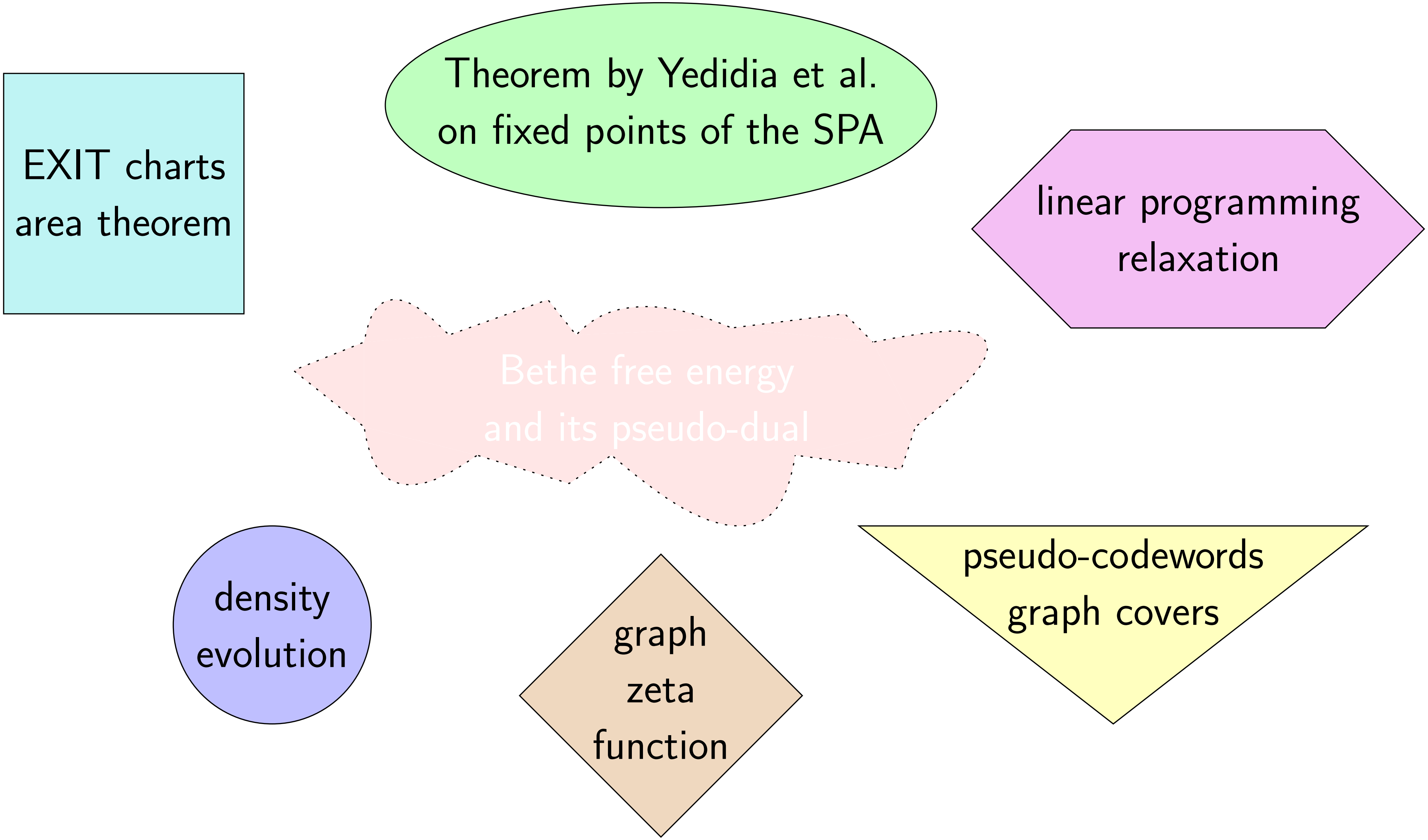
linear programming
relaxation

Bethe free energy
and its pseudo-dual

density
evolution

graph
zeta
function

pseudo-codewords
graph covers



EXIT charts
area theorem

Theorem by Yedidia et al.
on fixed points of the SPA

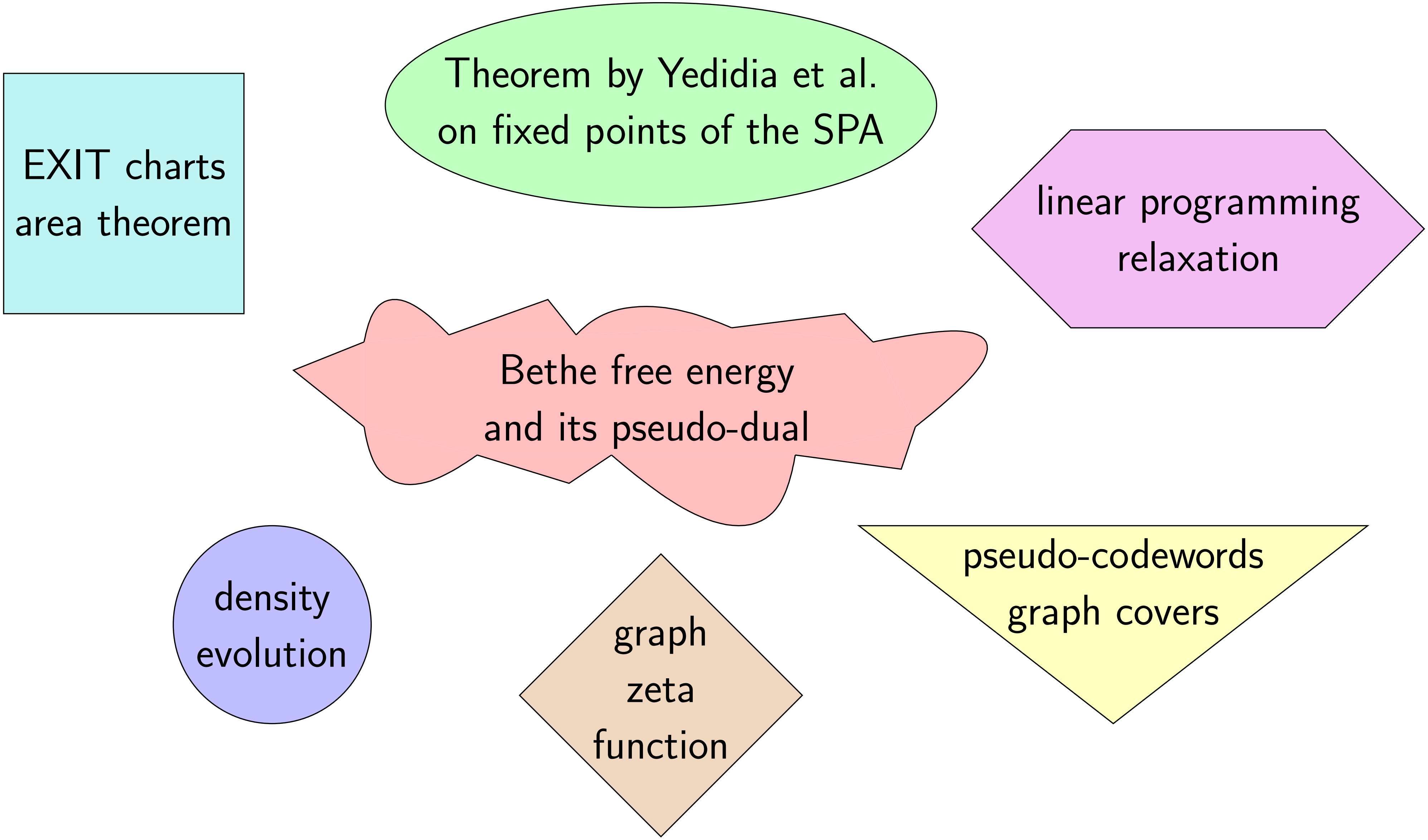
linear programming
relaxation

Bethe free energy
and its pseudo-dual

density
evolution

graph
zeta
function

pseudo-codewords
graph covers



Theorem by Yedidia et al.
on fixed points of the SPA

EXIT charts
area theorem

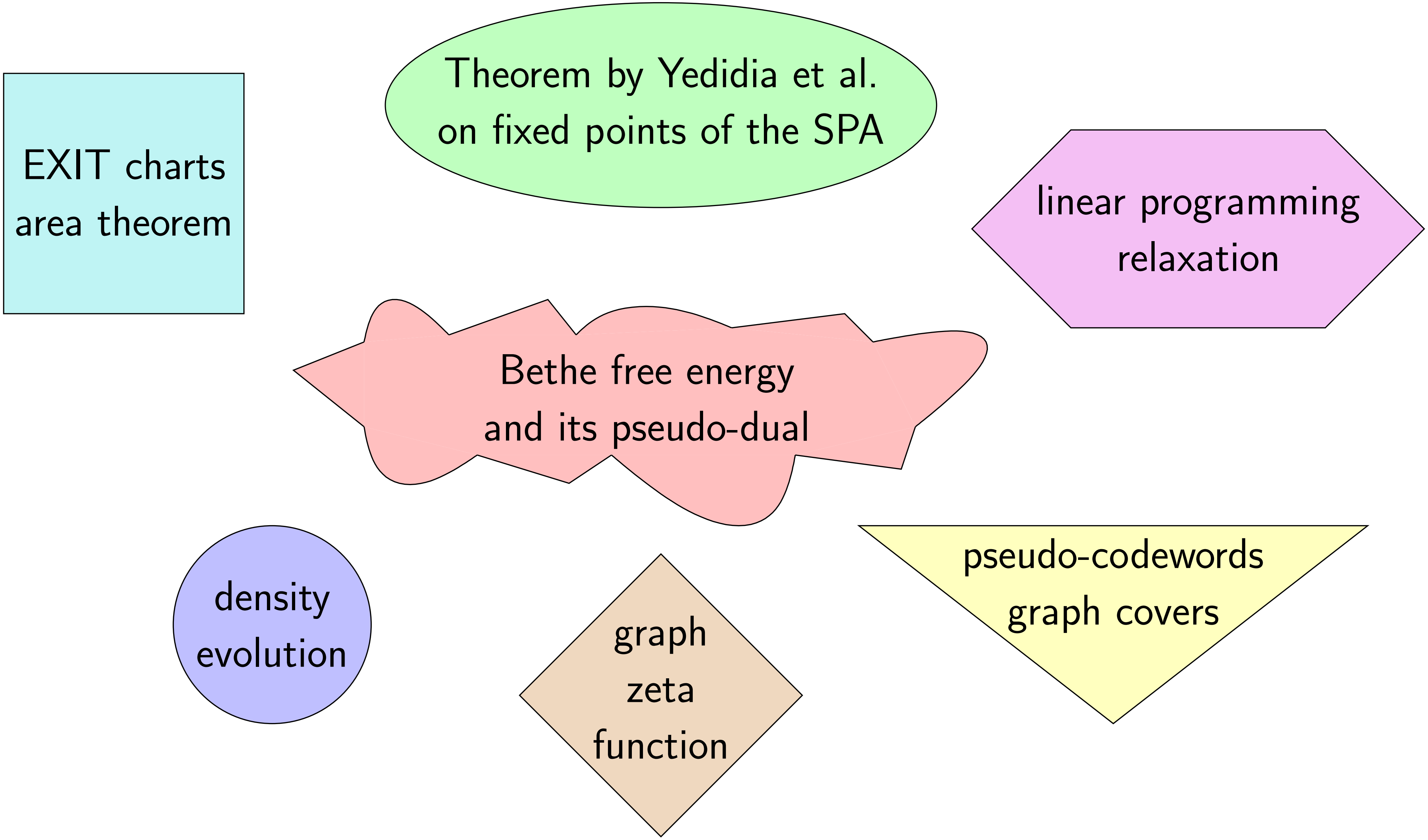
linear programming
relaxation

Bethe free energy
and its pseudo-dual

density
evolution

graph
zeta
function

pseudo-codewords
graph covers



EXIT charts area theorem

Theorem by Yedidia et al. on fixed points of the SPA

linear programming relaxation

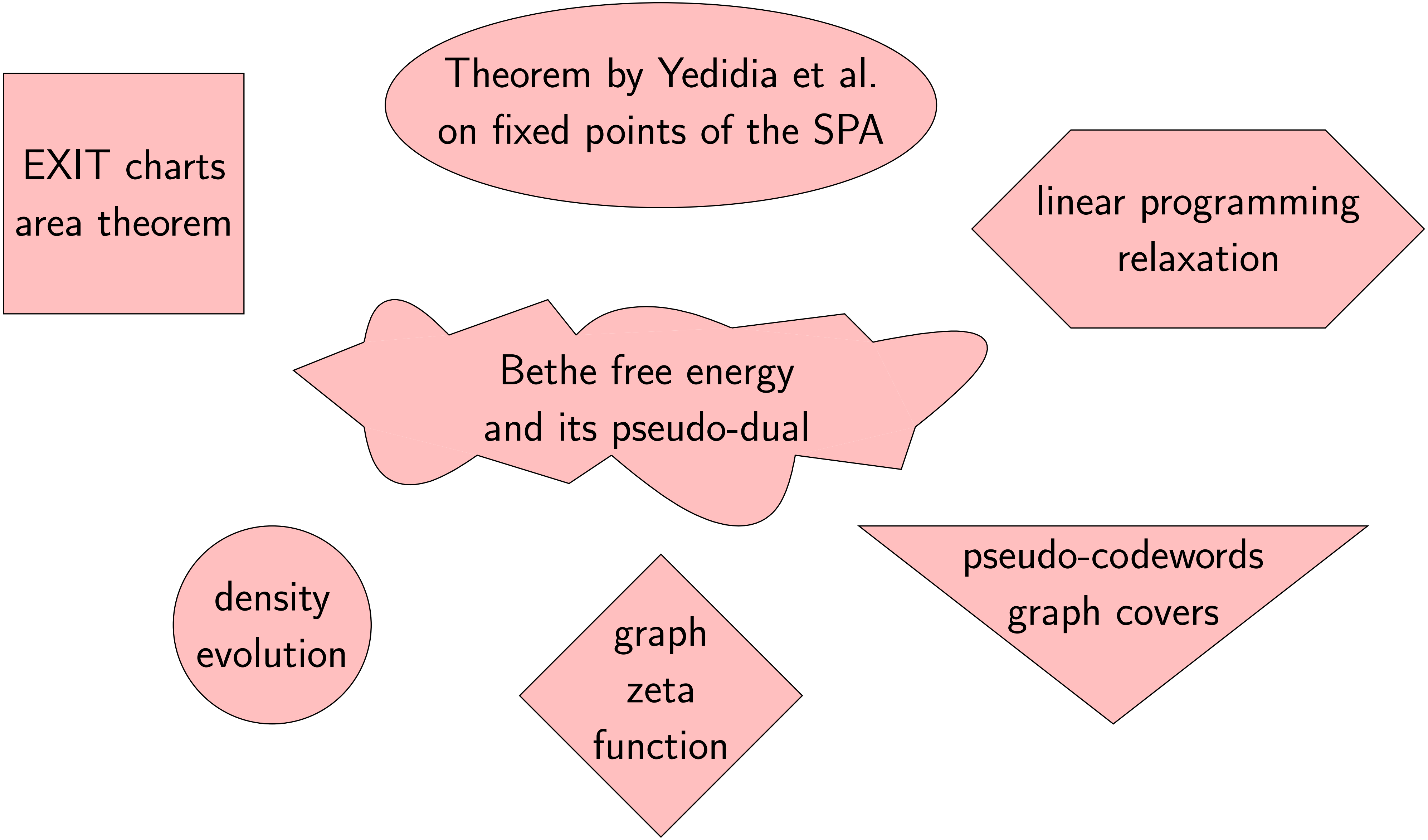
Bethe free energy and its pseudo-dual

density evolution

graph zeta function

pseudo-codewords graph covers

We are looking for a unifying perspective to these topics.



Theorem by Yedidia et al. on fixed points of the SPA

EXIT charts area theorem

linear programming relaxation

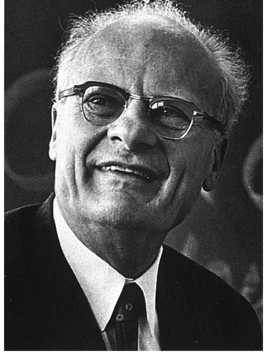
Bethe free energy and its pseudo-dual

density evolution

graph zeta function

pseudo-codewords graph covers

We are looking for a unifying perspective to these topics.



Hans Bethe

EXIT charts
area theorem

Theorem by Yedidia et al.
on fixed points of the SPA

linear programming
relaxation

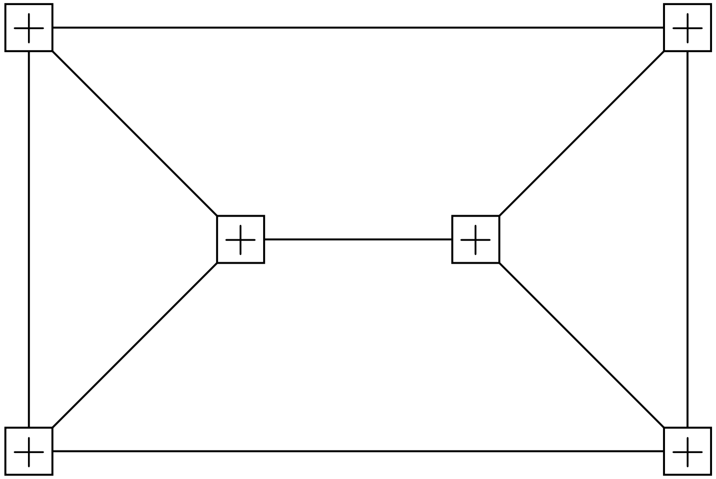
Bethe free energy
and its pseudo-dual

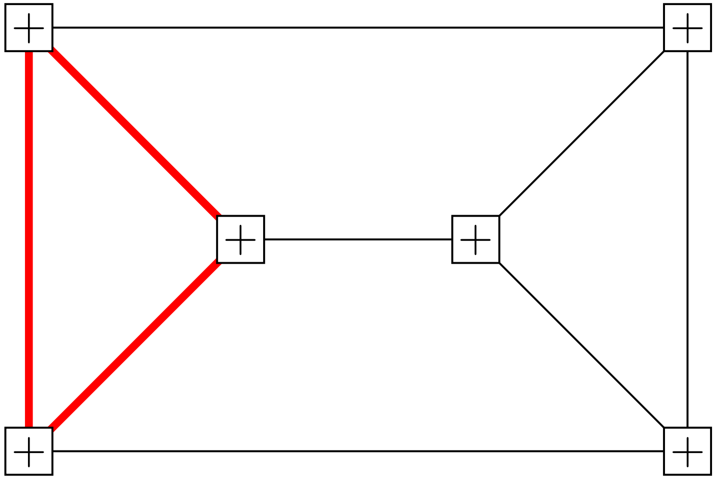
density
evolution

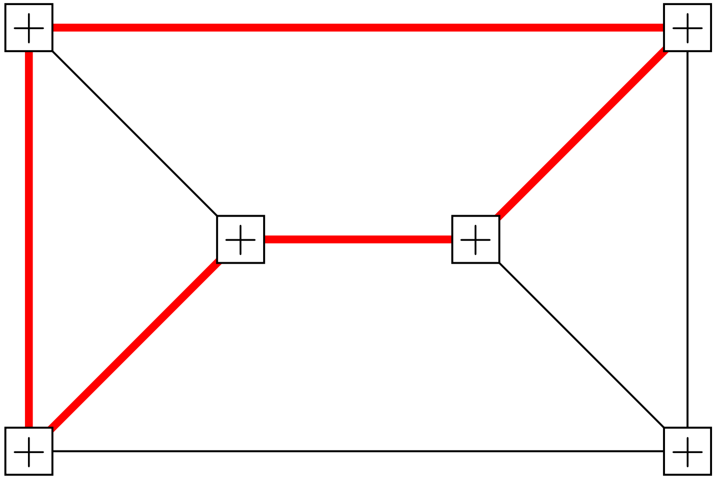
graph
zeta
function

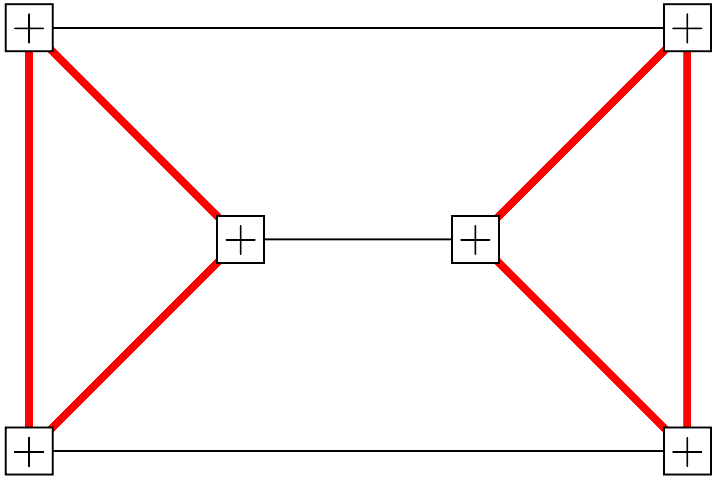
pseudo-codewords
graph covers

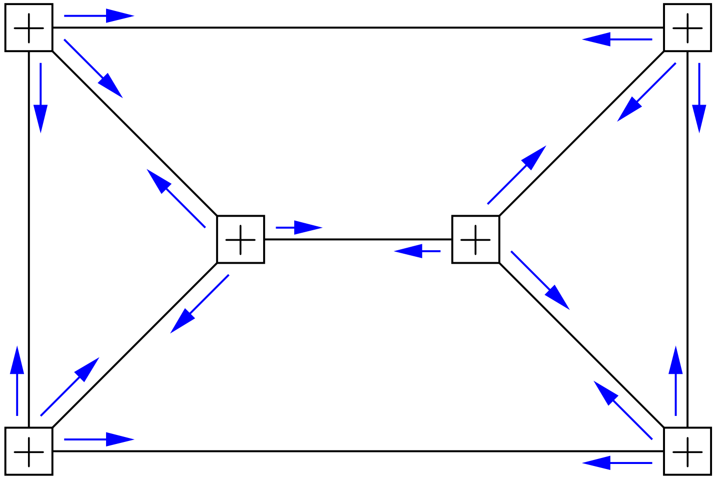
We are looking for a unifying perspective to these topics.

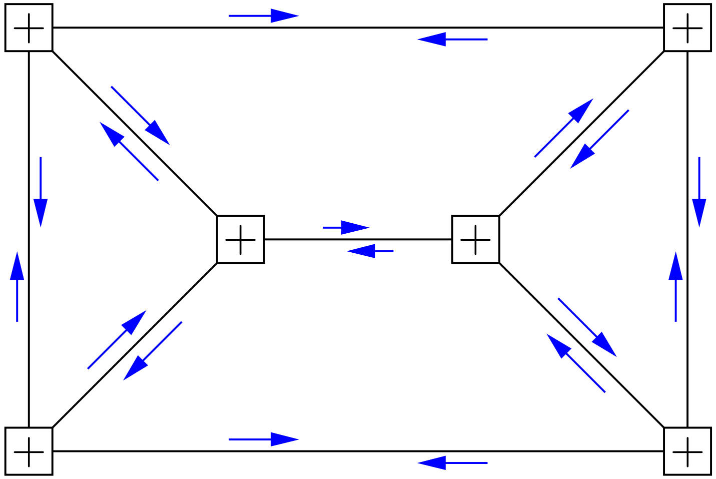


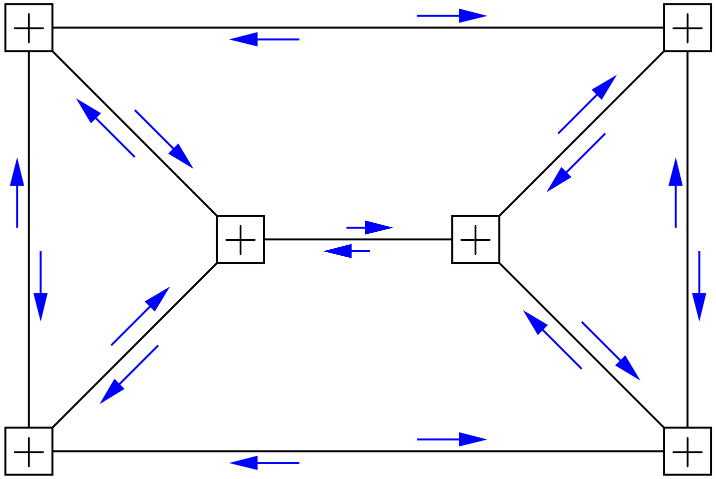


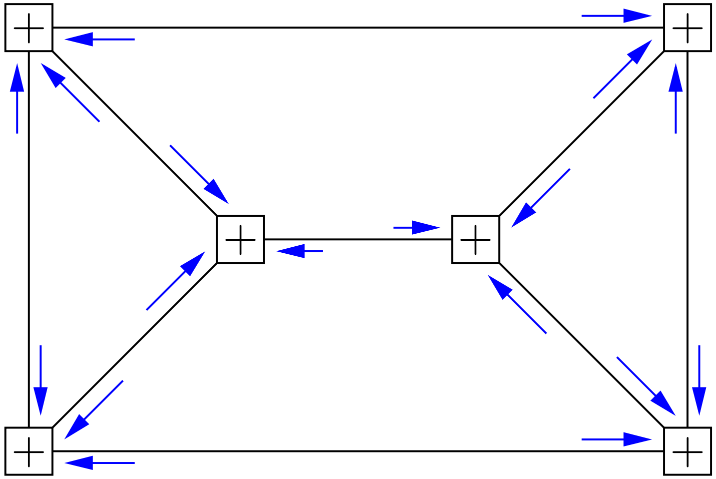


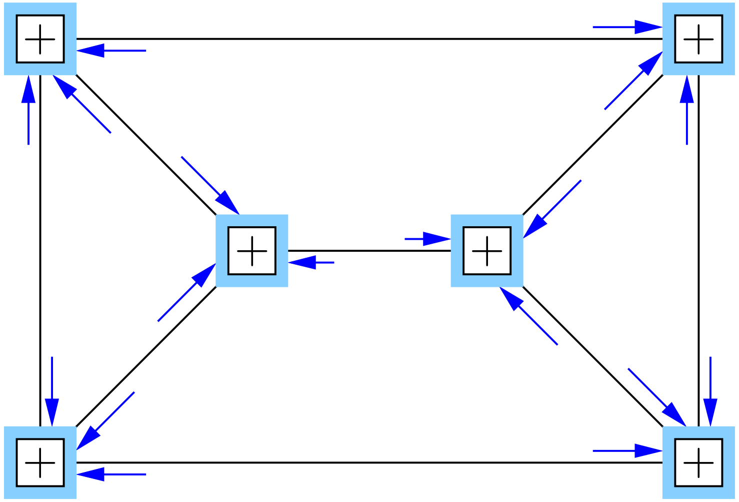


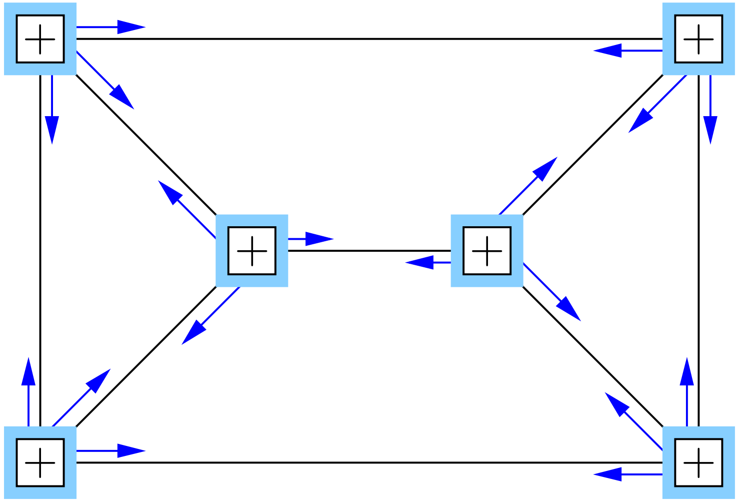


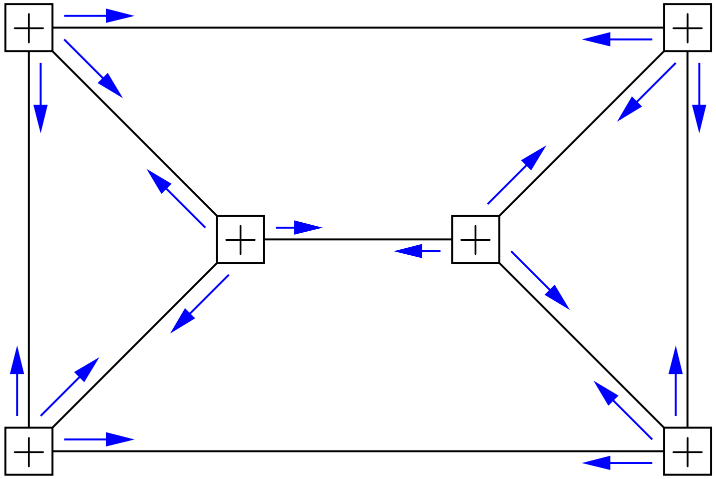


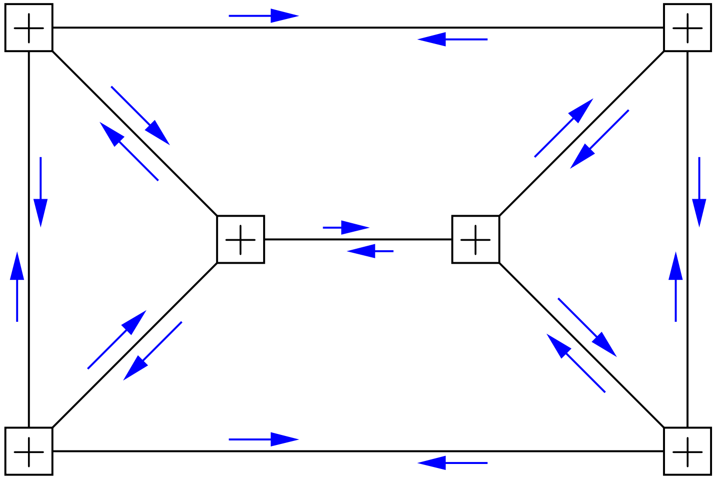


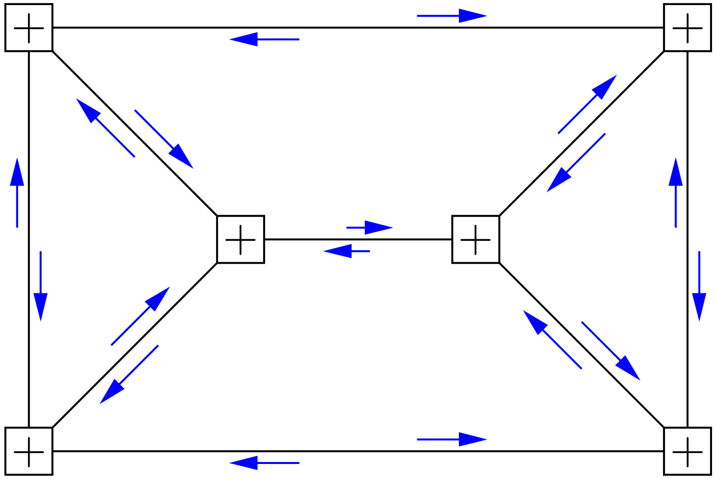


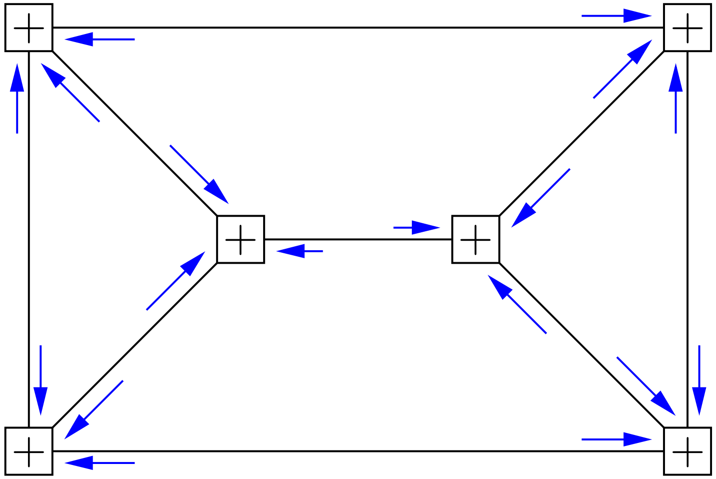


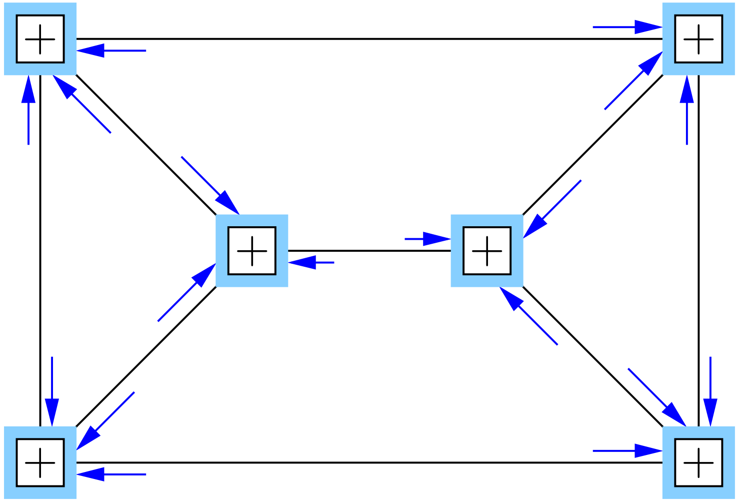


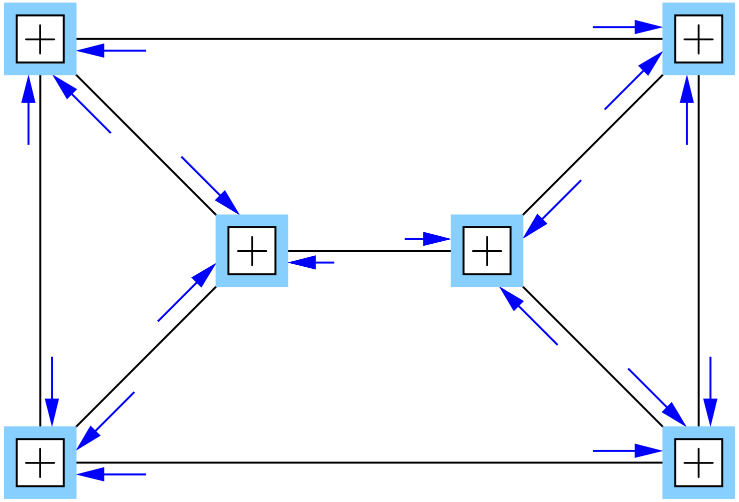


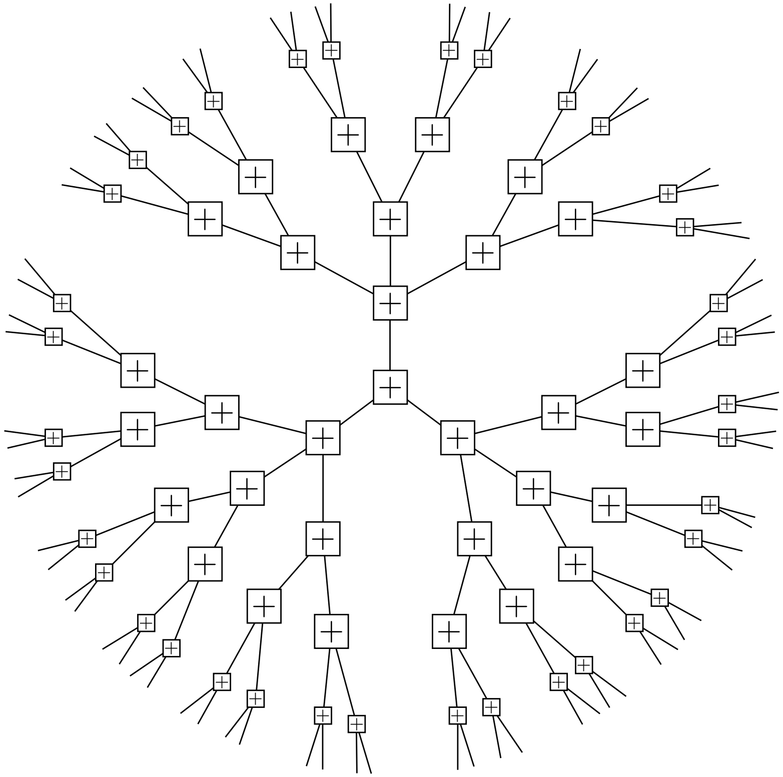


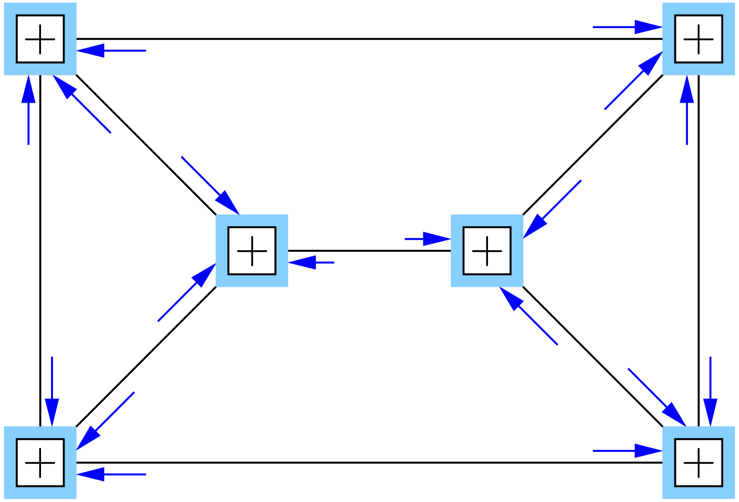


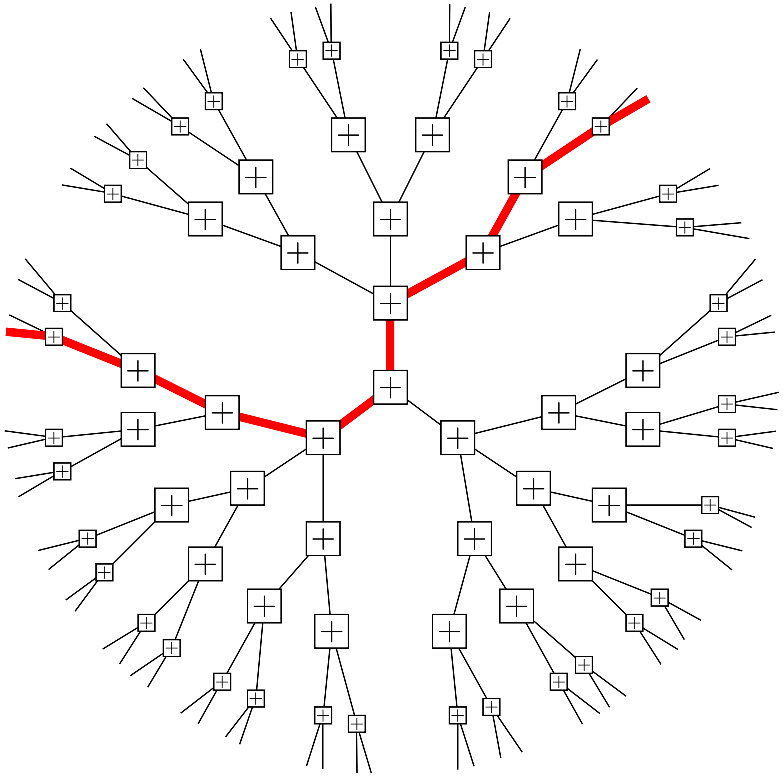


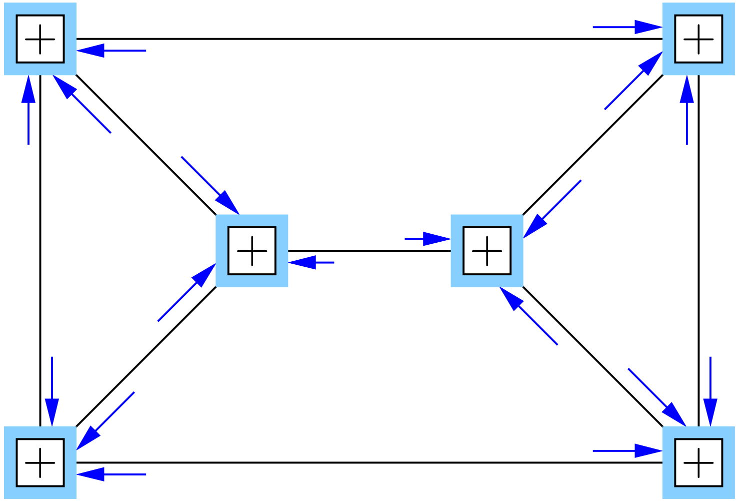


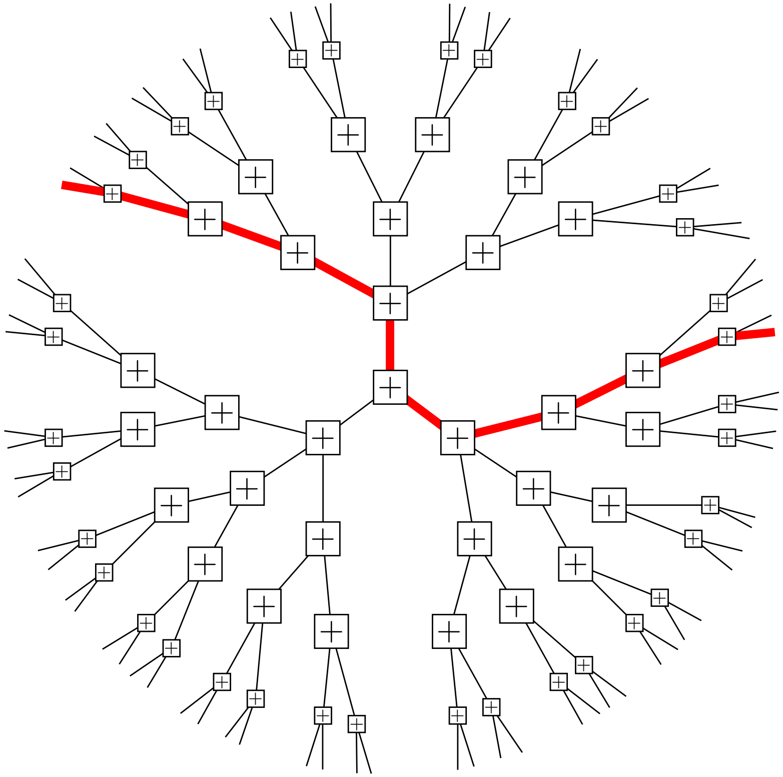


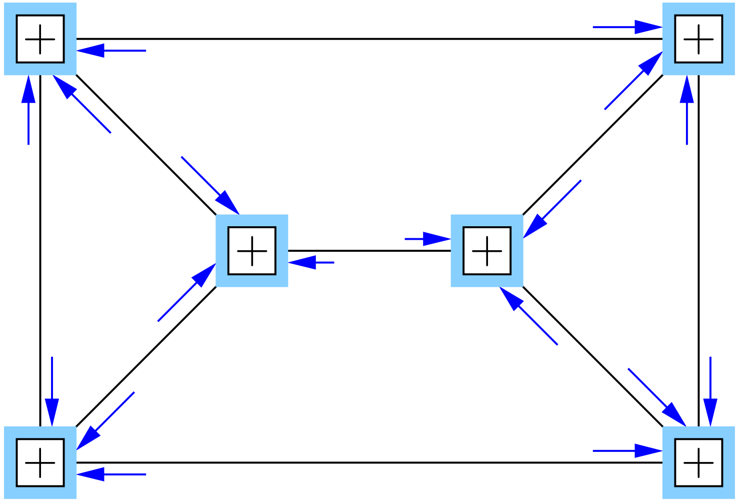


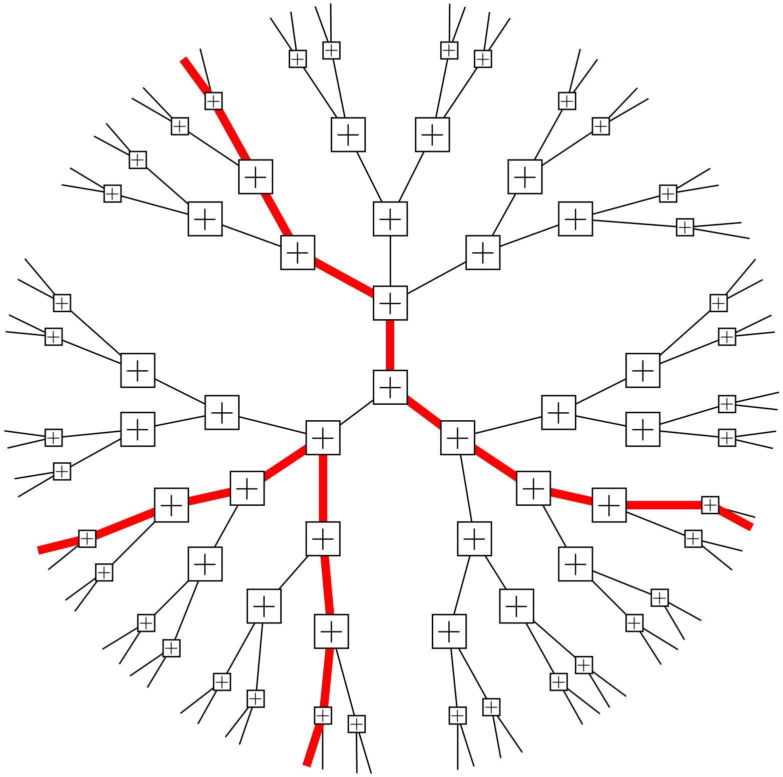


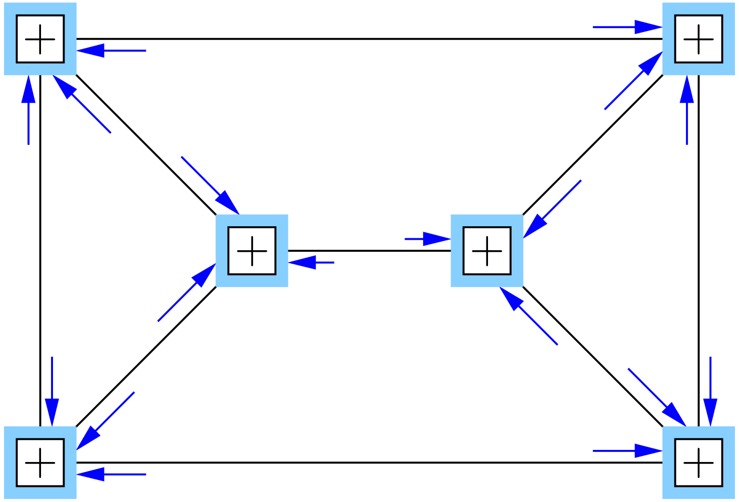


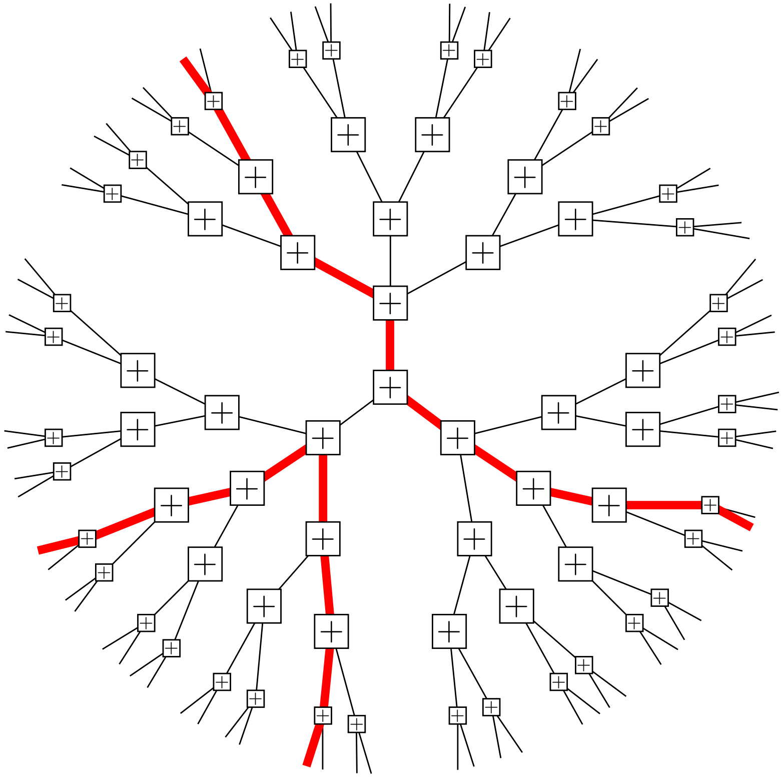


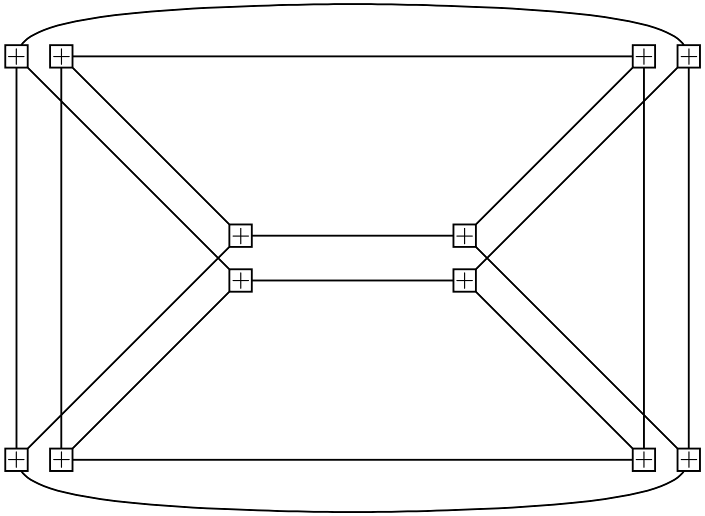


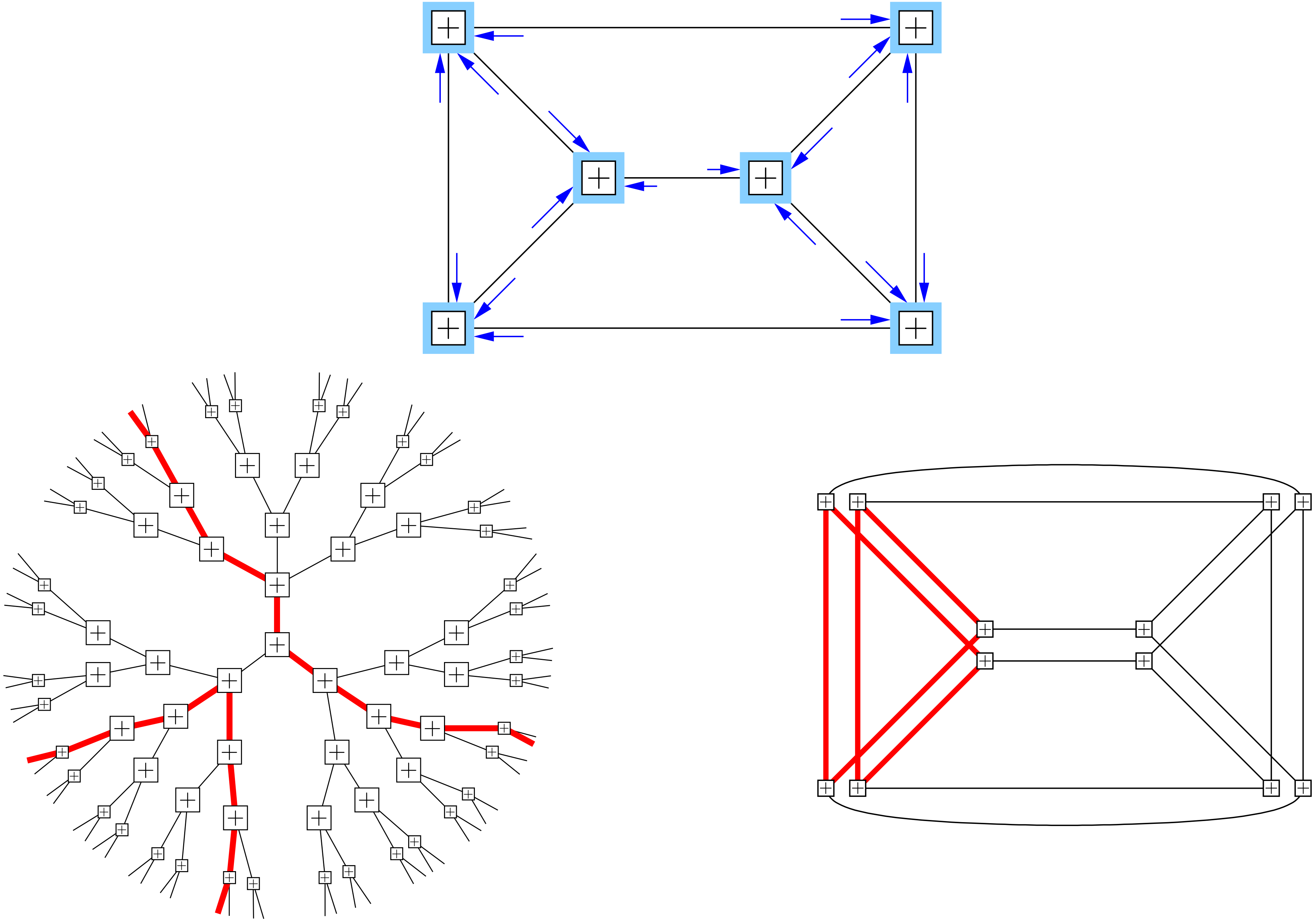


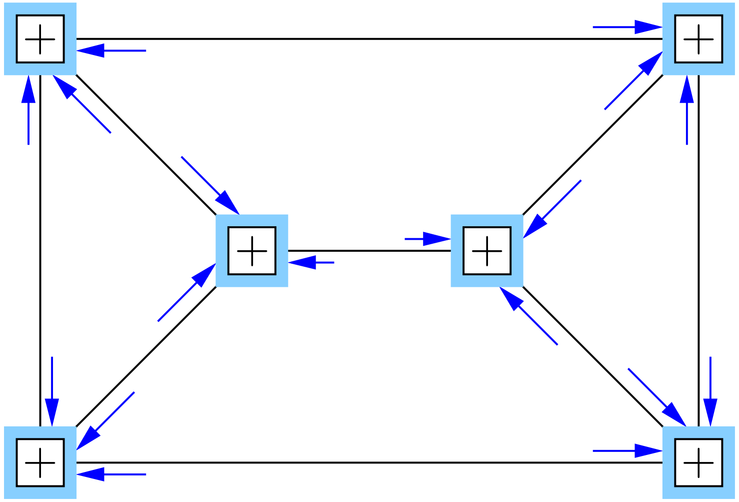


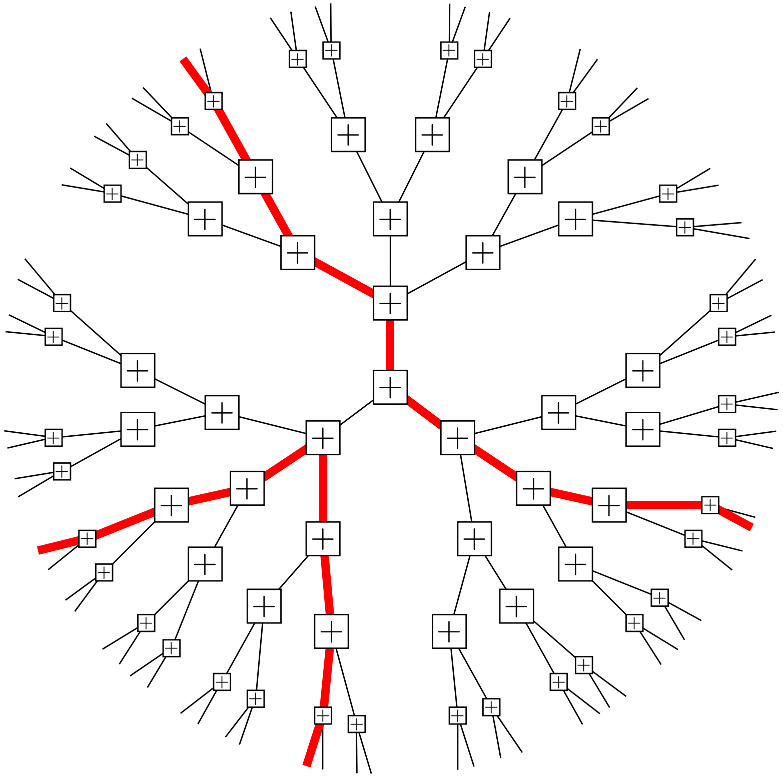


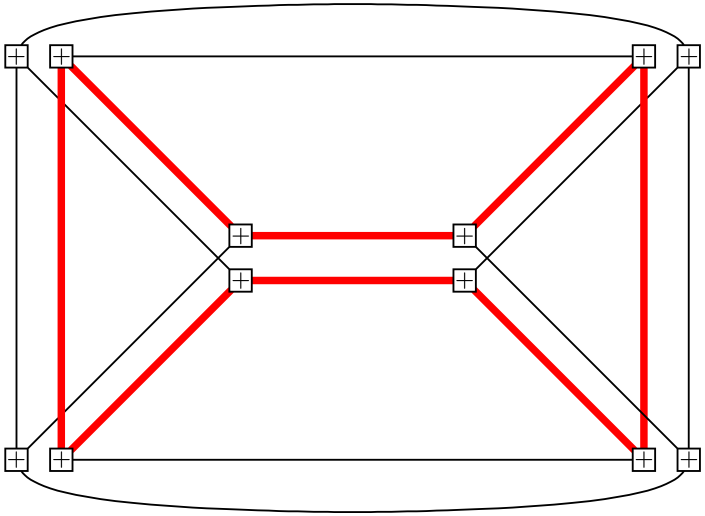


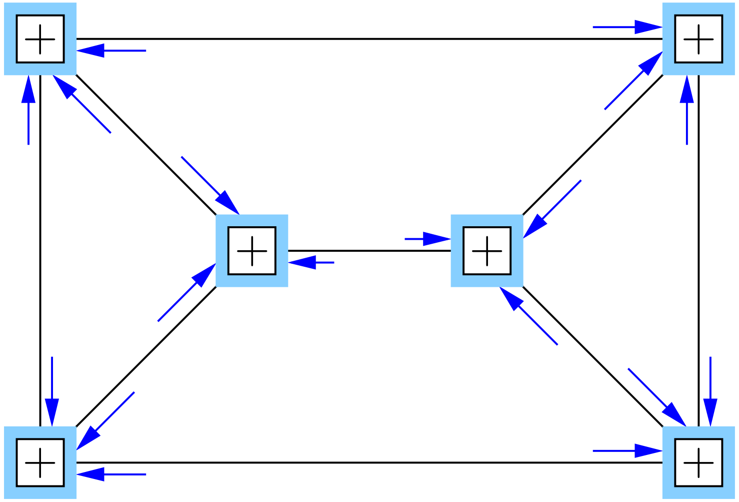


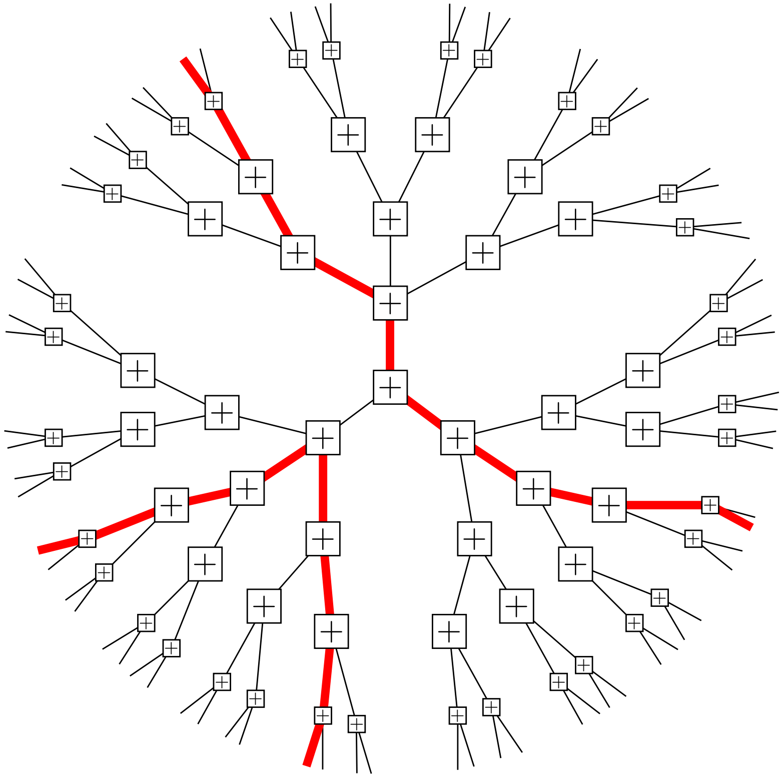


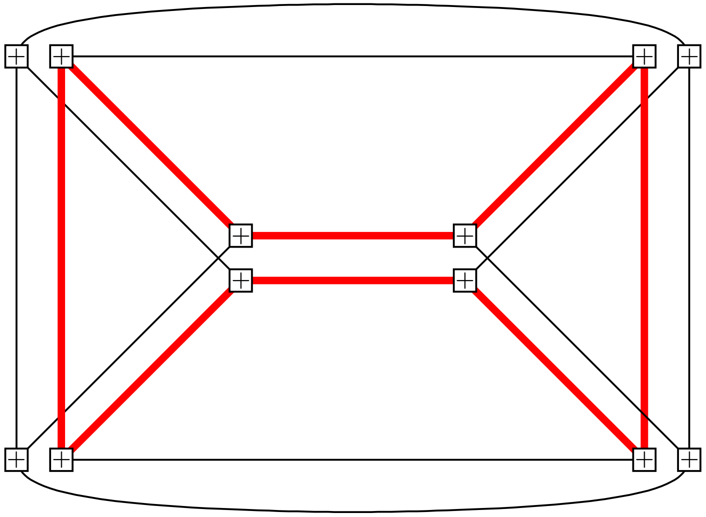




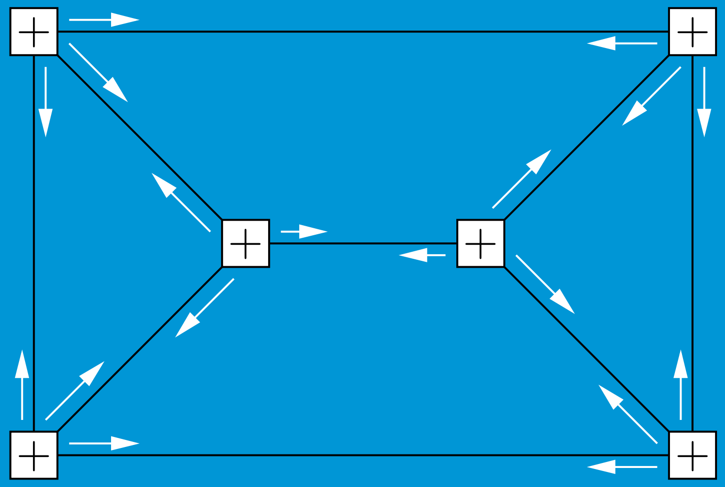


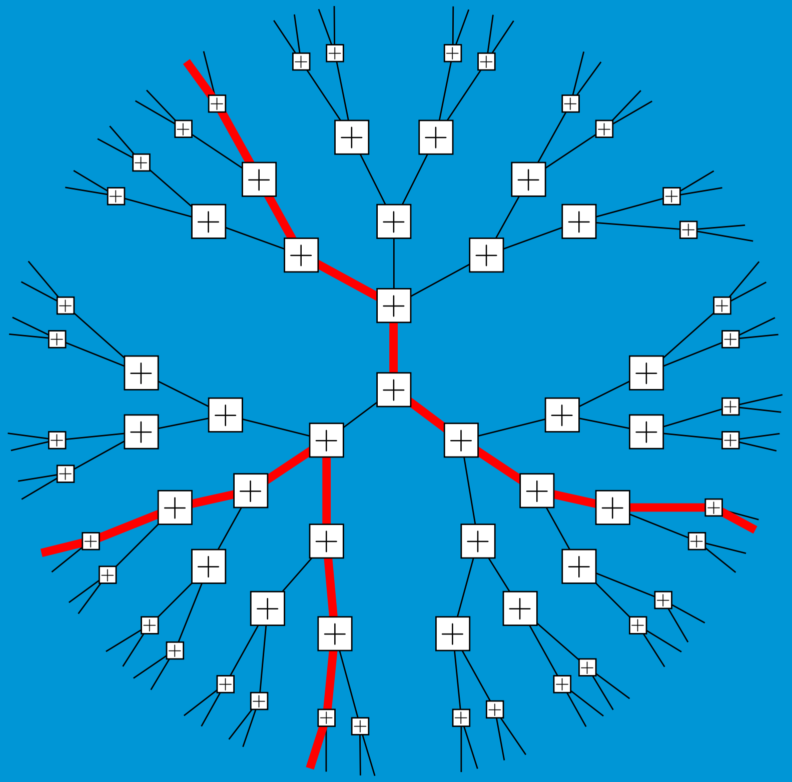


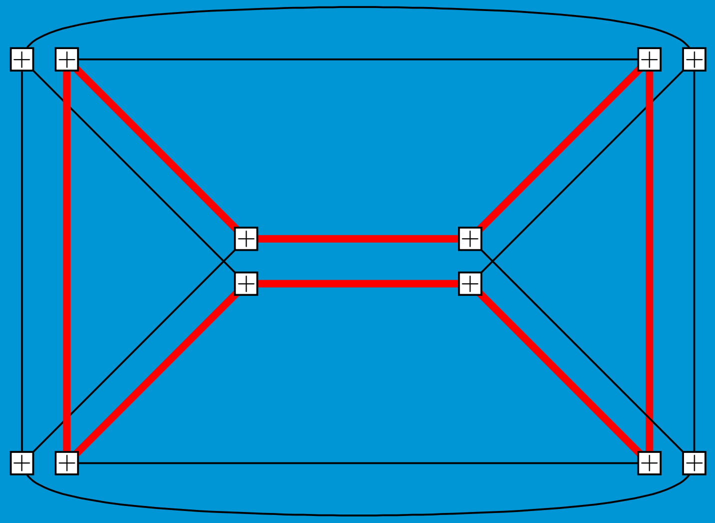




$$\zeta(\mathsf{V}_1,\ldots,\mathsf{V}_n) = \sum_{\mathbf{k}} \zeta_{\mathbf{k}} \mathbf{V}^{\mathbf{k}} = \prod_{[\Gamma]} \frac{1}{1-g(\Gamma,\mathbf{V})}$$

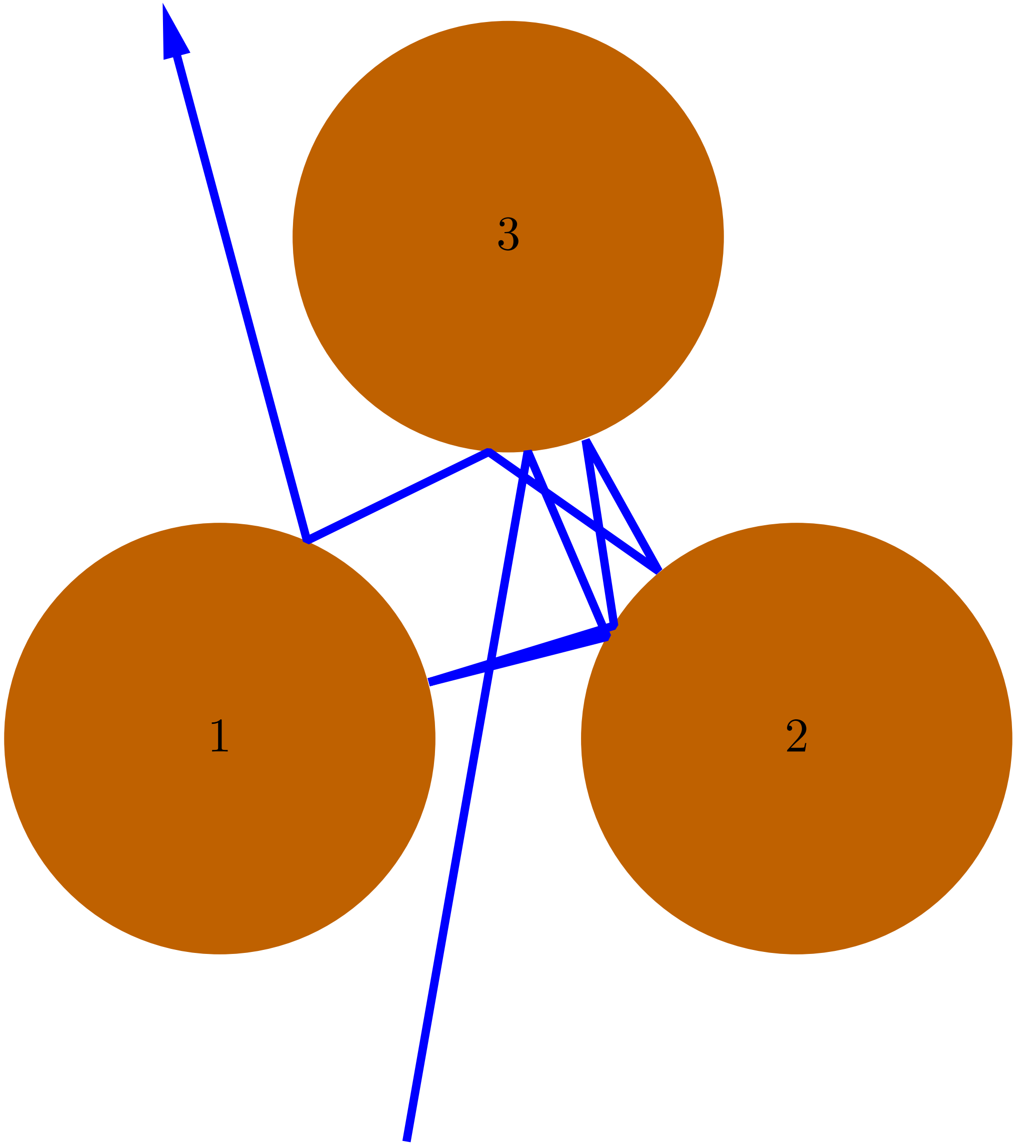




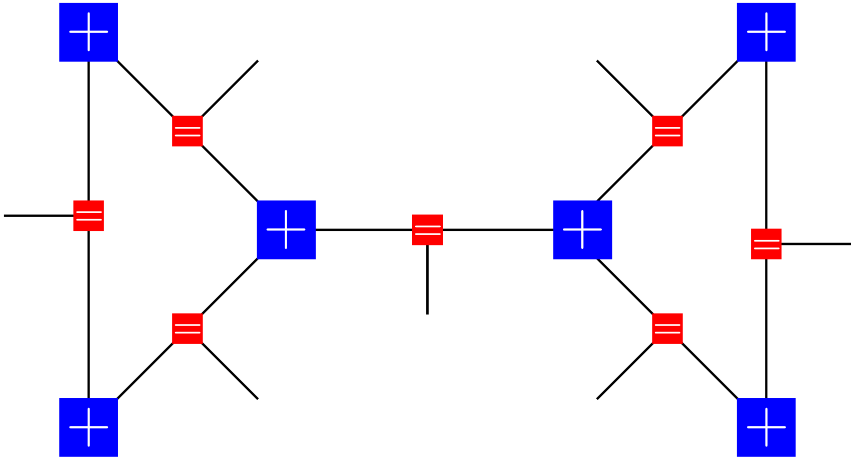


$$\zeta(\mathsf{V}_1,\ldots,\mathsf{V}_n) \;=\; \sum_{\mathbf{k}} \zeta_{\mathbf{k}} \mathbf{V}^{\mathbf{k}} \;=\; \prod_{[\Gamma]} \frac{1}{1-g(\Gamma,\mathbf{V})}$$

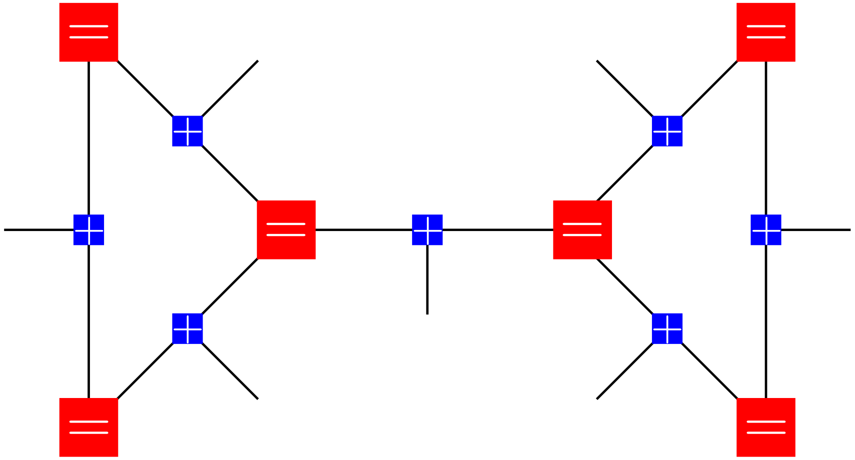
Pinball



Cycle Code NFG –vs– Community Det. NFG

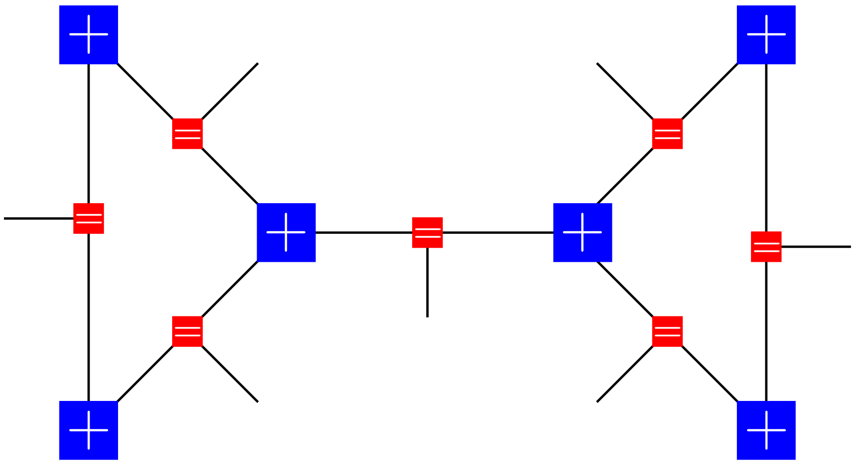


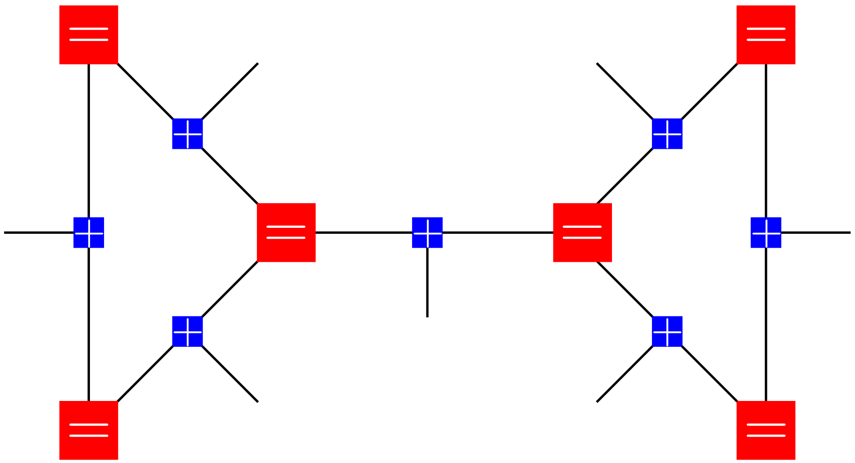
cycle code
normal factor graph



community detection
normal factor graph

Cycle Code NFG –vs– Community Det. NFG



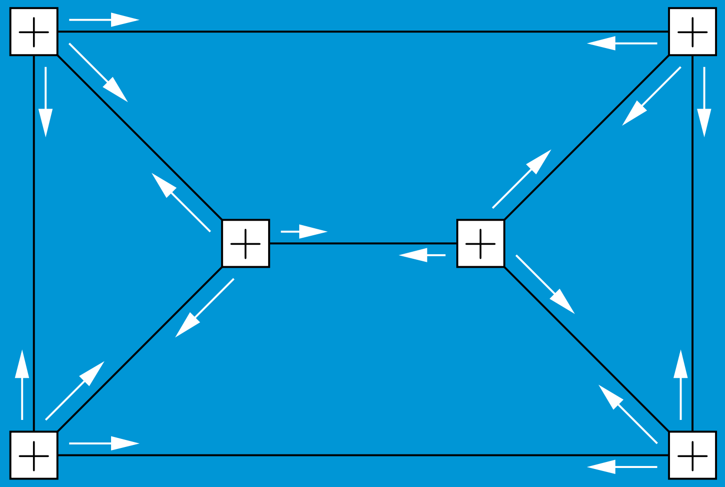


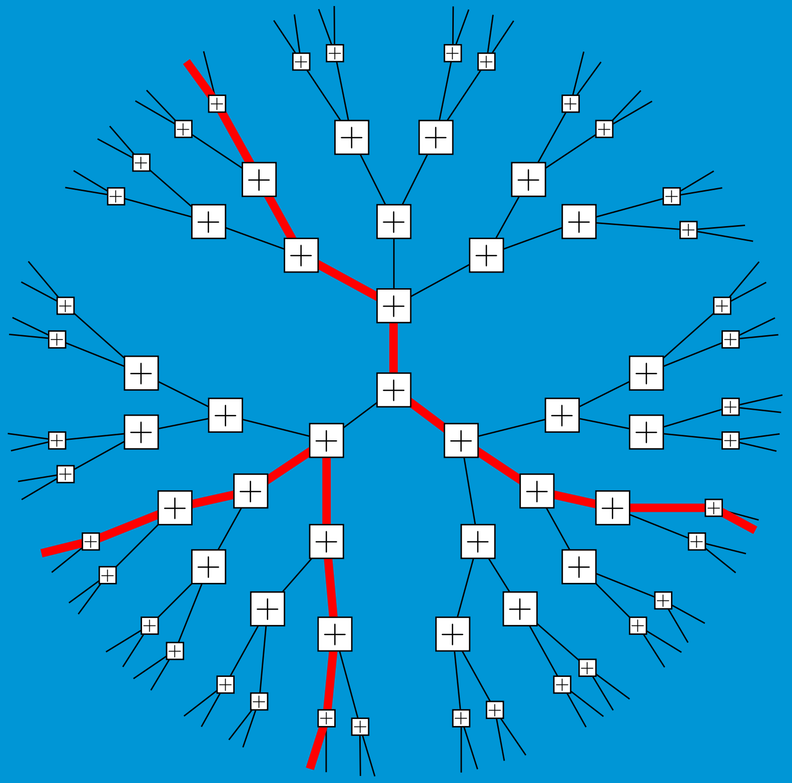
cycle code
normal factor graph

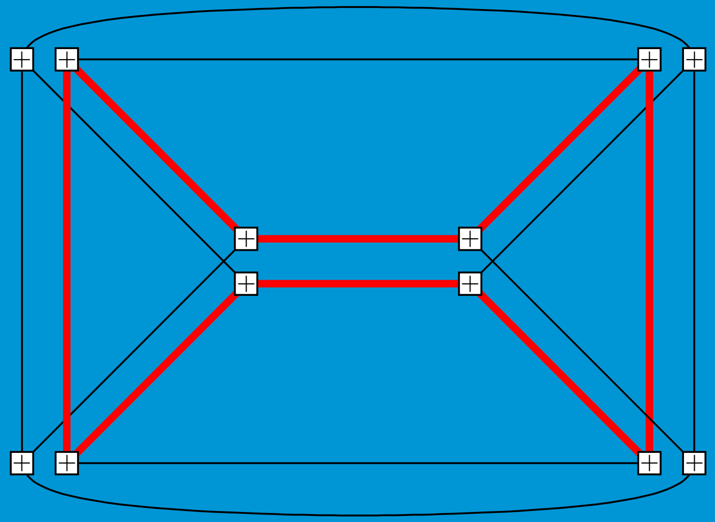
○—●

community detection
normal factor graph

Connection given by normal factor graph duality, cf. [Forney, 2001].







$$\zeta(\mathsf{V}_1,\ldots,\mathsf{V}_n) \;=\; \sum_{\mathbf{k}} \zeta_{\mathbf{k}} \mathbf{V}^{\mathbf{k}} \;=\; \prod_{[\Gamma]} \frac{1}{1-g(\Gamma,\mathbf{V})}$$

What Can a Power Series Do For You?

Consider the power series $\theta(\mathsf{V})$:

$$\theta(\mathsf{V}) \triangleq \sum_{k} \theta_k \mathsf{V}^k = 1\mathsf{V}^0 + 2\mathsf{V}^1 + 8\mathsf{V}^3 + 16\mathsf{V}^4 + 64\mathsf{V}^6 + 128\mathsf{V}^7 + \cdots$$

What Can a Power Series Do For You?

Consider the power series $\theta(\mathrm{V})$:

$$\theta(\mathrm{V}) \triangleq \sum_k \theta_k \mathrm{V}^k = 1\mathrm{V}^0 + 2\mathrm{V}^1 + 8\mathrm{V}^3 + 16\mathrm{V}^4 + 64\mathrm{V}^6 + 128\mathrm{V}^7 + \cdots$$

We can obtain useful information from

What Can a Power Series Do For You?

Consider the power series $\theta(\mathsf{V})$:

$$\theta(\mathsf{V}) \triangleq \sum_k \theta_k \mathsf{V}^k = 1\mathsf{V}^0 + 2\mathsf{V}^1 + 8\mathsf{V}^3 + 16\mathsf{V}^4 + 64\mathsf{V}^6 + 128\mathsf{V}^7 + \cdots$$

We can obtain useful information from

- . . . the exponents of $\theta(\mathsf{V})$

What Can a Power Series Do For You?

Consider the power series $\theta(\mathrm{V})$:

$$\theta(\mathrm{V}) \triangleq \sum_{k} \theta_k \mathrm{V}^k = 1\mathrm{V}^0 + 2\mathrm{V}^1 + 8\mathrm{V}^3 + 16\mathrm{V}^4 + 64\mathrm{V}^6 + 128\mathrm{V}^7 + \cdots$$

We can obtain useful information from

- ...the exponents of $\theta(\mathrm{V})$
- ...the coefficients of $\theta(\mathrm{V})$

What Can a Power Series Do For You?

Consider the power series $\theta(\mathrm{V})$:

$$\theta(\mathrm{V}) \triangleq \sum_k \theta_k \mathrm{V}^k = 1\mathrm{V}^0 + 2\mathrm{V}^1 + 8\mathrm{V}^3 + 16\mathrm{V}^4 + 64\mathrm{V}^6 + 128\mathrm{V}^7 + \cdots$$

We can obtain useful information from

- . . . the exponents of $\theta(\mathrm{V})$
- . . . the coefficients of $\theta(\mathrm{V})$
- . . . the evaluation of $\theta(\mathrm{V})$ for some V

What Can a Power Series Do For You?

Consider the power series $\theta(\mathrm{V})$:

$$\theta(\mathrm{V}) \triangleq \sum_k \theta_k \mathrm{V}^k = 1\mathrm{V}^0 + 2\mathrm{V}^1 + 8\mathrm{V}^3 + 16\mathrm{V}^4 + 64\mathrm{V}^6 + 128\mathrm{V}^7 + \cdots$$

We can obtain useful information from

- ...the exponents of $\theta(\mathrm{V})$
- ...the coefficients of $\theta(\mathrm{V})$
- ...the evaluation of $\theta(\mathrm{V})$ for some V
- ...the convergence radius of $\theta(\mathrm{V})$

What Can a Power Series Do For You?

Consider the power series $\theta(\mathrm{V})$:

$$\theta(\mathrm{V}) \triangleq \sum_k \theta_k \mathrm{V}^k = 1\mathrm{V}^0 + 2\mathrm{V}^1 + 8\mathrm{V}^3 + 16\mathrm{V}^4 + 64\mathrm{V}^6 + 128\mathrm{V}^7 + \cdots$$

We can obtain useful information from

- . . . the exponents of $\theta(\mathrm{V})$
- . . . the coefficients of $\theta(\mathrm{V})$
- . . . the evaluation of $\theta(\mathrm{V})$ for some V
- . . . the convergence radius of $\theta(\mathrm{V})$
- . . .

What Can a Power Series Do For You?

Consider the power series $\theta(\mathsf{V}_1, \ldots, \mathsf{V}_n)$:

$$\theta(\mathsf{V}_1, \ldots, \mathsf{V}_n) \triangleq \sum_{k_1,\ldots,k_n} \theta_{k_1,\ldots,k_n} \mathsf{V}_1^{k_1} \cdots \mathsf{V}_n^{k_n}$$

We can obtain useful information from

- ...the exponent vectors of $\theta(\mathsf{V}_1, \ldots, \mathsf{V}_n)$
- ...the coefficients of $\theta(\mathsf{V}_1, \ldots, \mathsf{V}_n)$
- ...the evaluation of $\theta(\mathsf{V}_1, \ldots, \mathsf{V}_n)$ for some $(\mathsf{V}_1, \ldots, \mathsf{V}_n)$
- ...the convergence region of $\theta(\mathsf{V}_1, \ldots, \mathsf{V}_n)$
- ...

What Can a Power Series Do For You?

Consider the power series (zeta function) $\zeta(\mathbf{V})$:

$$\zeta(\mathbf{V}) \triangleq \sum_{\mathbf{k}} \zeta_{\mathbf{k}} \mathbf{V}^{\mathbf{k}}$$

We can obtain useful information from

- . . . the expon. vecs. of $\zeta(\mathbf{V})$
- . . . the coefficients of $\zeta(\mathbf{V})$
- . . . the evaluation of $\zeta(\mathbf{V})$ for some $\mathbf{V}$
- . . . the convergence region of $\zeta(\mathbf{V})$
- . . .

Use of zeta functions for analyzing graphical models.

What Can a Power Series Do For You?

Consider the power series (zeta function) $\zeta(\mathbf{V})$:

$$\zeta(\mathbf{V}) \triangleq \sum_{\mathbf{k}} \zeta_{\mathbf{k}} \mathbf{V}^{\mathbf{k}}$$

We can obtain useful information from

- …the expon. vecs. of $\zeta(\mathbf{V})$ [Koetter, Li, V., Walker, 2004/2007]
- …the coefficients of $\zeta(\mathbf{V})$
- … the evaluation of $\zeta(\mathbf{V})$ for some $\mathbf{V}$
- …the convergence region of $\zeta(\mathbf{V})$
- …

Use of zeta functions for analyzing graphical models.

What Can a Power Series Do For You?

Consider the power series (zeta function) $\zeta(\mathbf{V})$:

$$\zeta(\mathbf{V}) \triangleq \sum_{\mathbf{k}} \zeta_{\mathbf{k}} \mathbf{V}^{\mathbf{k}}$$

We can obtain useful information from

- ...the expon. vecs. of $\zeta(\mathbf{V})$ [Koetter, Li, V., Walker, 2004/2007]
- ...the coefficients of $\zeta(\mathbf{V})$ [V., 2009/2010]
- ... the evaluation of $\zeta(\mathbf{V})$ for some $\mathbf{V}$
- ...the convergence region of $\zeta(\mathbf{V})$
- ...

Use of zeta functions for analyzing graphical models.

What Can a Power Series Do For You?

Consider the power series (zeta function) $\zeta(\mathbf{V})$:

$$\zeta(\mathbf{V}) \triangleq \sum_{\mathbf{k}} \zeta_{\mathbf{k}} \mathbf{V}^{\mathbf{k}}$$

We can obtain useful information from

- ...the expon. vecs. of $\zeta(\mathbf{V})$ [Koetter, Li, V., Walker, 2004/2007]
- ...the coefficients of $\zeta(\mathbf{V})$ [V., 2009/2010]
- ... the evaluation of $\zeta(\mathbf{V})$ for some $\mathbf{V}$ [Watanabe, 2009/2010]
- ...the convergence region of $\zeta(\mathbf{V})$
- ...

Use of zeta functions for analyzing graphical models.

What Can a Power Series Do For You?

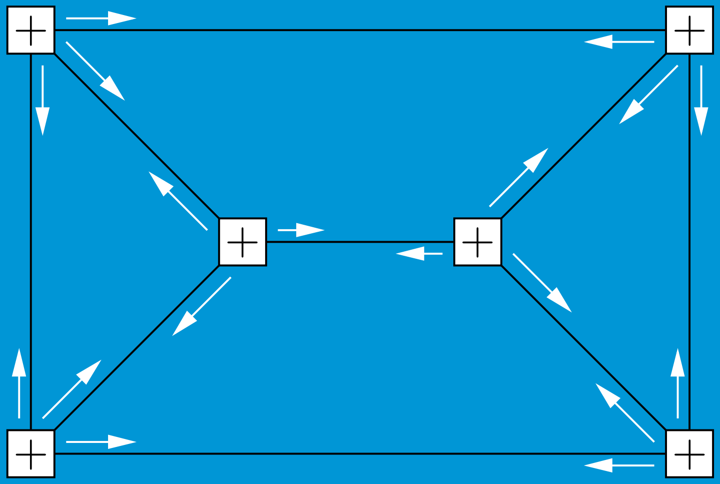
Consider the power series (zeta function) $\zeta(\mathbf{V})$:

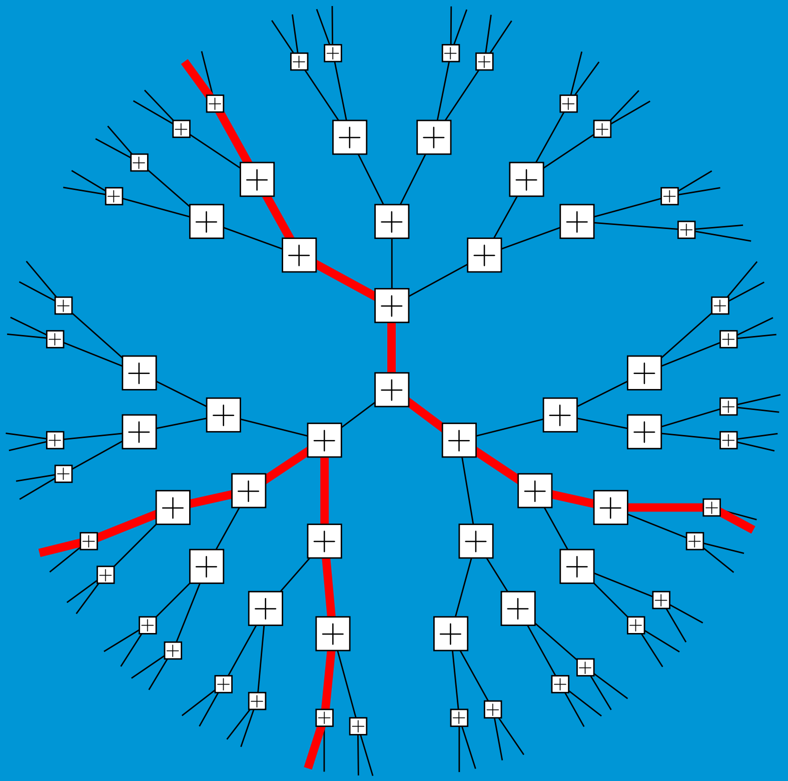
$$\zeta(\mathbf{V}) \triangleq \sum_{\mathbf{k}} \zeta_{\mathbf{k}} \mathbf{V}^{\mathbf{k}}$$

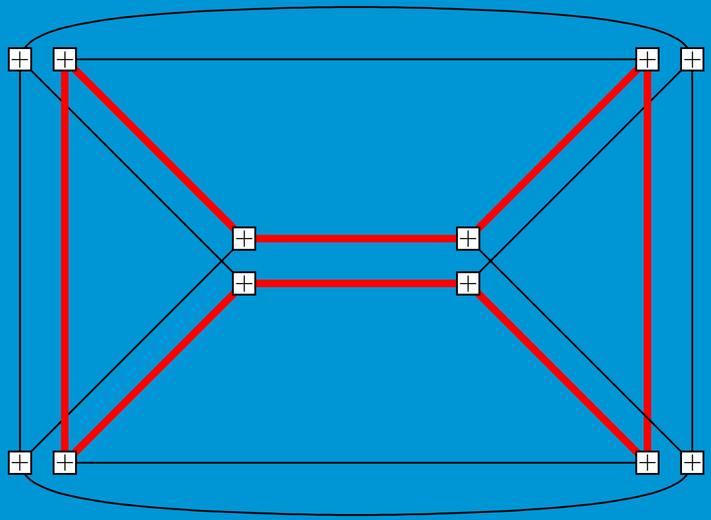
We can obtain useful information from

- ...the expon. vecs. of $\zeta(\mathbf{V})$ [Koetter, Li, V., Walker, 2004/2007]
- ...the coefficients of $\zeta(\mathbf{V})$ [V., 2009/2010] [today]
- ... the evaluation of $\zeta(\mathbf{V})$ for some $\mathbf{V}$ [Watanabe, 2009/2010]
- ...the convergence region of $\zeta(\mathbf{V})$ [today]
- ...

Use of zeta functions for analyzing graphical models.

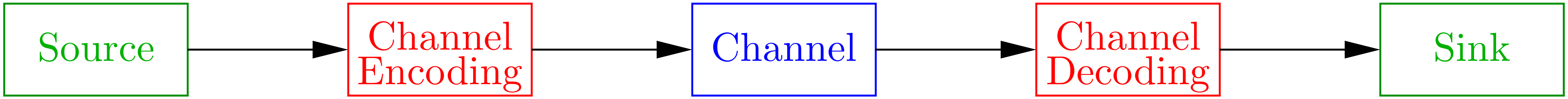




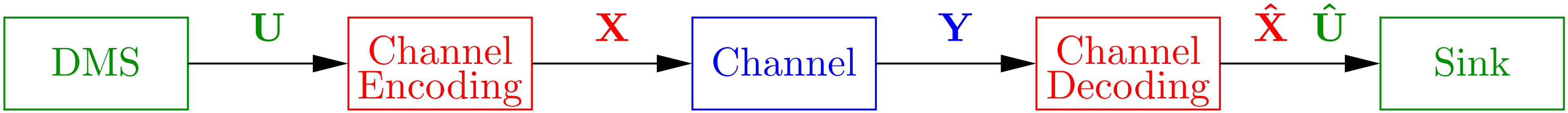


$$\zeta(\mathsf{V}_1,\ldots,\mathsf{V}_n) = \sum_{\mathbf{k}} \zeta_{\mathbf{k}} \mathsf{\mathbf{V}}^{\mathbf{k}} = \prod_{[\Gamma]} \frac{1}{1-g(\Gamma,\mathsf{\mathbf{V}})}$$

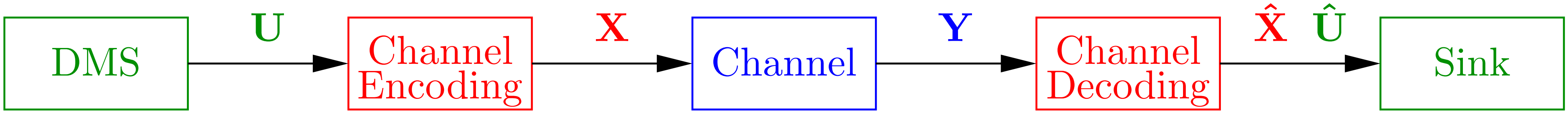
Communication Model



Communication Model



Communication Model

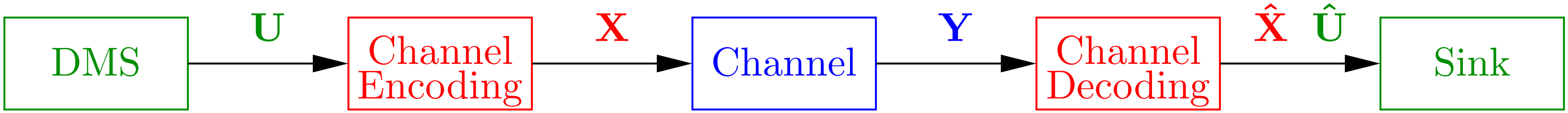


Information word: $\mathbf{u} = (u_1, \ldots, u_k) \in \mathcal{U}^k$

Sent codeword: $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{C} \subseteq \mathcal{X}^n$

Received word: $\mathbf{y} = (y_1, \ldots, y_n) \in \mathcal{Y}^n$

Communication Model



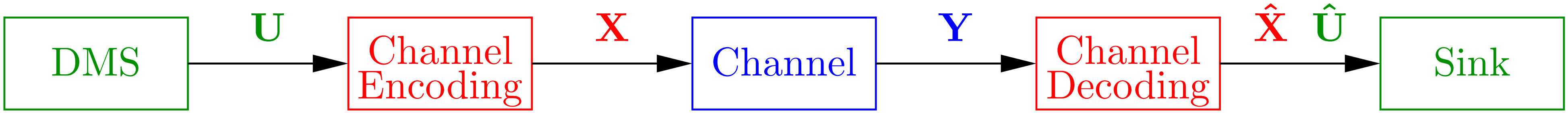
Information word: $\mathbf{u} = (u_1, \ldots, u_k) \in \mathcal{U}^k$

Sent codeword: $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{C} \subseteq \mathcal{X}^n$

Received word: $\mathbf{y} = (y_1, \ldots, y_n) \in \mathcal{Y}^n$

Decoding: Based on $\mathbf{y}$ we would like to estimate the transmitted codeword $\hat{\mathbf{x}}$ or the information word $\hat{\mathbf{u}}$.

Communication Model



Information word: $\mathbf{u} = (u_1, \ldots, u_k) \in \mathcal{U}^k$

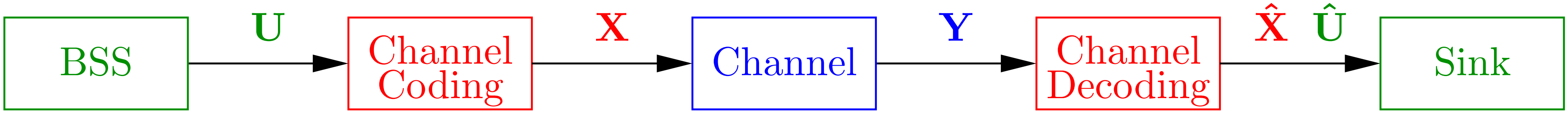
Sent codeword: $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{C} \subseteq \mathcal{X}^n$

Received word: $\mathbf{y} = (y_1, \ldots, y_n) \in \mathcal{Y}^n$

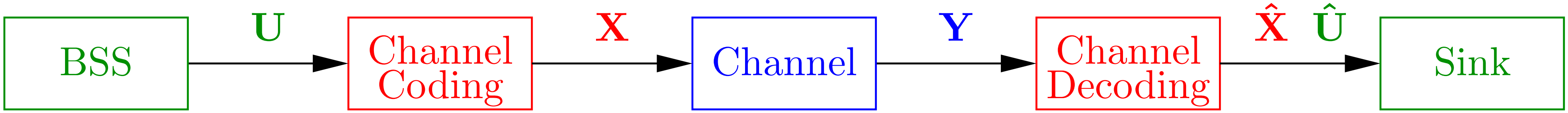
Decoding: Based on $\mathbf{y}$ we would like to estimate the transmitted codeword $\hat{\mathbf{x}}$ or the information word $\hat{\mathbf{u}}$.

Depending on what criterion we optimize, we obtain different decoding algorithms.

Symbol-Wise MAP Decoding (Part 1)



Symbol-Wise MAP Decoding (Part 1)



Minimizing the symbol error probability (for each $i = 1, \ldots, k$) results in symbol-wise MAP decoding.

For each $i = 1, \ldots, k$:

$$\hat{u}_i^{\text{symbol}}(\mathbf{y}) = \underset{u_i \in \mathcal{U}}{\operatorname{argmax}} \, P_{U_i|\mathbf{Y}}(u_i|\mathbf{y}) = \underset{u_i \in \mathcal{U}}{\operatorname{argmax}} \, P_{U_i,\mathbf{Y}}(u_i, \mathbf{y}).$$

Symbol-Wise MAP Decoding (Part 2)

Rewriting symbol-wise MAP decoding for symbol i we obtain

$$\hat{u}_i^{\text{symbol}}(\mathbf{y}) = \underset{u_i \in \mathcal{U}}{\operatorname{argmax}} \, P_{U_i,\mathbf{Y}}(u_i, \mathbf{y})$$

Symbol-Wise MAP Decoding (Part 2)

Rewriting symbol-wise MAP decoding for symbol i we obtain

$$\begin{aligned}\hat{u}_i^{\text{symbol}}(\mathbf{y}) &= \operatorname*{argmax}_{u_i \in \mathcal{U}} P_{U_i,\mathbf{Y}}(u_i, \mathbf{y}) \\ &= \underbrace{\underbrace{\operatorname*{argmax}_{u_i \in \mathcal{U}}}_{} \underbrace{\sum_{\substack{\mathbf{u} \in \mathcal{U}^k,\ \mathbf{x} \in \mathcal{X}^n \\ u_i \ \text{fixed}}} \underbrace{P_{\mathbf{UXY}}(\mathbf{u}, \mathbf{x}, \mathbf{y}),}_{}}_{}}_{}\end{aligned}$$

Symbol-Wise MAP Decoding (Part 2)

Rewriting symbol-wise MAP decoding for symbol i we obtain

$$\begin{aligned}\hat{u}_i^{\text{symbol}}(\mathbf{y}) &= \operatorname*{argmax}_{u_i \in \mathcal{U}} P_{U_i,\mathbf{Y}}(u_i, \mathbf{y}) \\ &= \underbrace{\underbrace{\operatorname*{argmax}_{u_i \in \mathcal{U}}}_{} \underbrace{\sum_{\substack{\mathbf{u}\in\mathcal{U}^k,\ \mathbf{x}\in\mathcal{X}^n \\ u_i \text{ fixed}}} \underbrace{P_{\mathbf{UXY}}(\mathbf{u}, \mathbf{x}, \mathbf{y}),}_{\text{Joint pmf/pdf}}}_{}}_{}\end{aligned}$$

Symbol-Wise MAP Decoding (Part 2)

Rewriting symbol-wise MAP decoding for symbol i we obtain

$$\hat{u}_i^{\text{symbol}}(\mathbf{y}) = \operatorname*{argmax}_{u_i\in\mathcal{U}} P_{U_i,\mathbf{Y}}(u_i,\mathbf{y})$$

$$= \underbrace{\underbrace{\operatorname*{argmax}_{u_i\in\mathcal{U}}}_{} \underbrace{\sum_{\substack{\mathbf{u}\in\mathcal{U}^k,\ \mathbf{x}\in\mathcal{X}^n\\ u_i\ \text{fixed}}} \underbrace{P_{\mathbf{UXY}}(\mathbf{u},\mathbf{x},\mathbf{y}),}_{\text{Joint pmf/pdf}}}_{\text{Marginal function}}}_{}$$

Symbol-Wise MAP Decoding (Part 2)

Rewriting symbol-wise MAP decoding for symbol i we obtain

$$\begin{aligned}
\hat{u}_i^{\text{symbol}}(\mathbf{y}) &= \operatorname*{argmax}_{u_i\in\mathcal{U}} P_{U_i,\mathbf{Y}}(u_i,\mathbf{y}) \\
&= \underbrace{\underbrace{\operatorname*{argmax}_{u_i\in\mathcal{U}}}_{\text{Decision taking}} \underbrace{\sum_{\substack{\mathbf{u}\in\mathcal{U}^k,\ \mathbf{x}\in\mathcal{X}^n \\ u_i \text{ fixed}}} \underbrace{P_{\mathbf{UXY}}(\mathbf{u},\mathbf{x},\mathbf{y}),}_{\text{Joint pmf/pdf}}}_{\text{Marginal function}}}_{}
\end{aligned}$$

Symbol-Wise MAP Decoding (Part 2)

Rewriting symbol-wise MAP decoding for symbol i we obtain

$$
\begin{aligned}
\hat{u}_i^{\text{symbol}}(\mathbf{y}) &= \operatorname*{argmax}_{u_i \in \mathcal{U}} P_{U_i,\mathbf{Y}}(u_i, \mathbf{y}) \\
&= \underbrace{\underbrace{\operatorname*{argmax}_{u_i \in \mathcal{U}}}_{\text{Decision taking}} \underbrace{\sum_{\substack{\mathbf{u}\in\mathcal{U}^k,\ \mathbf{x}\in\mathcal{X}^n \\ u_i \ \text{fixed}}} \underbrace{P_{\mathbf{UXY}}(\mathbf{u}, \mathbf{x}, \mathbf{y}),}_{\text{Joint pmf/pdf}}}_{\text{Marginal function}}}_{\text{Decision about symbol } u_i \text{ based on symbol-wise decoding}}
\end{aligned}
$$

Binary Linear Codes

Let $\mathbf{H}$ be a parity-check matrix, e.g.

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Binary Linear Codes

Let $\mathbf{H}$ be a parity-check matrix, e.g.

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

The code C described by $\mathbf{H}$ is then

$$C = \left\{ (x_1, x_2, x_3, x_4, x_5) \in \mathbb{F}_2^5 \;\middle|\; \mathbf{H} \cdot \mathbf{x}^\top = \mathbf{0}^\top \pmod 2 \right\}.$$

Binary Linear Codes

Let $\mathbf{H}$ be a parity-check matrix, e.g.

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

The code C described by $\mathbf{H}$ is then

$$C = \left\{ (x_1, x_2, x_3, x_4, x_5) \in \mathbb{F}_2^5 \;\middle|\; \mathbf{H} \cdot \mathbf{x}^\mathsf{T} = \mathbf{0}^\mathsf{T} \pmod 2 \right\}.$$

A vector $\mathbf{x} \in \mathbb{F}_2^5$ is a codeword if and only if

$$\mathbf{H} \cdot \mathbf{x}^\mathsf{T} = \mathbf{0}^\mathsf{T} \pmod 2.$$

Binary Linear Codes

This means that $\mathbf{x}$ is a codeword if and only if $\mathbf{x}$ fulfills the following two equations:

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Binary Linear Codes

This means that $\mathbf{x}$ is a codeword if and only if $\mathbf{x}$ fulfills the following two equations:

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix} \quad \Rightarrow \quad x_1 + x_2 + x_3 = 0 \pmod 2$$

Binary Linear Codes

This means that $\mathbf{x}$ is a codeword if and only if $\mathbf{x}$ fulfills the following two equations:

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix} \quad \Rightarrow \quad \begin{aligned} x_1 + x_2 + x_3 &= 0 \pmod 2 \\ x_2 + x_4 + x_5 &= 0 \pmod 2 \end{aligned}$$

Binary Linear Codes

This means that $\mathbf{x}$ is a codeword if and only if $\mathbf{x}$ fulfills the following two equations:

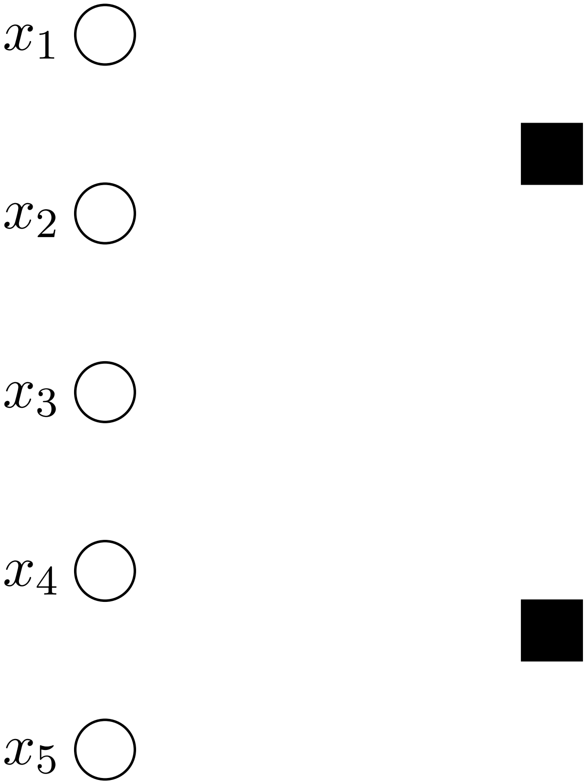
$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix} \quad \Rightarrow \quad \begin{aligned} x_1 + x_2 + x_3 &= 0 \pmod 2 \\ x_2 + x_4 + x_5 &= 0 \pmod 2 \end{aligned}$$

In summary,

$$\begin{aligned} C &= \left\{ (x_1, x_2, x_3, x_4, x_5) \in \mathbb{F}_2^5 \,\middle|\, \mathbf{H} \cdot \mathbf{x}^\top = \mathbf{0}^\top \pmod 2 \right\} \\ &= \left\{ (x_1, x_2, x_3, x_4, x_5) \in \mathbb{F}_2^5 \,\middle|\, \begin{aligned} x_1 + x_2 + x_3 &= 0 \pmod 2 \\ x_2 + x_4 + x_5 &= 0 \pmod 2 \end{aligned} \right\}. \end{aligned}$$

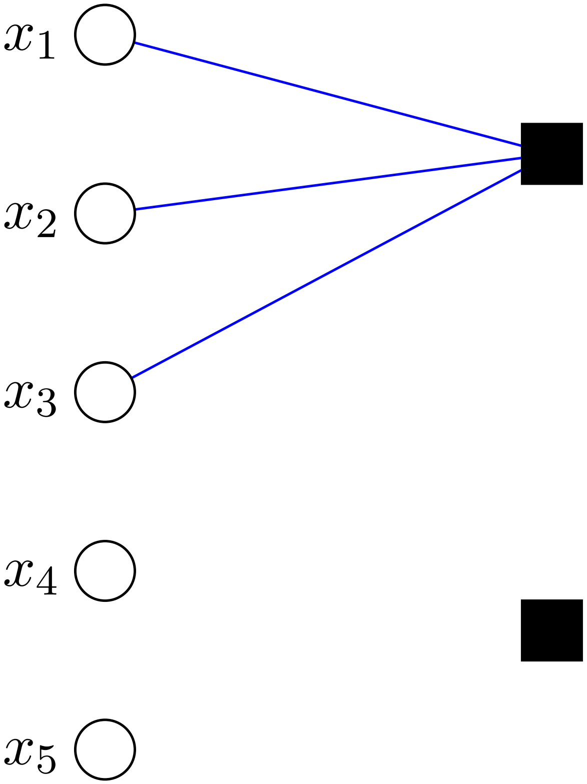
Graphical Representation of a Code

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$



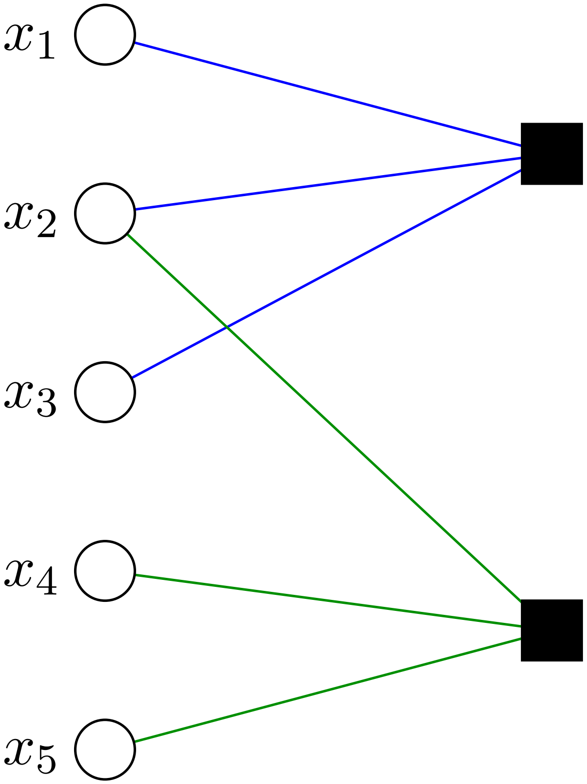
Graphical Representation of a Code

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$



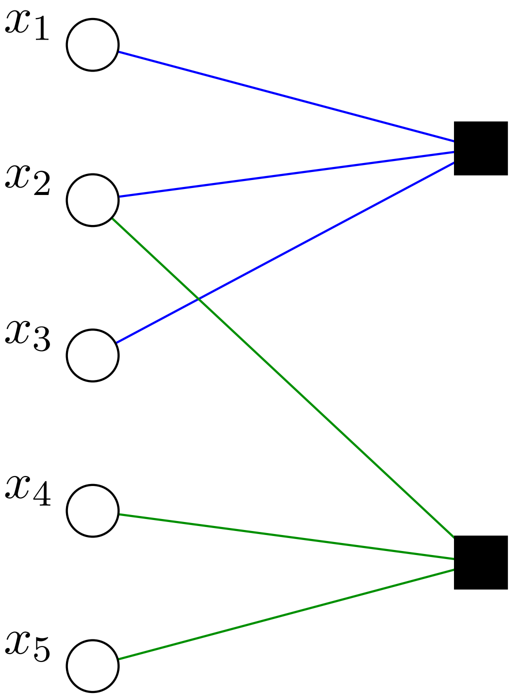
Graphical Representation of a Code

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$



FG of a Data Communication System based on a parity-check code

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

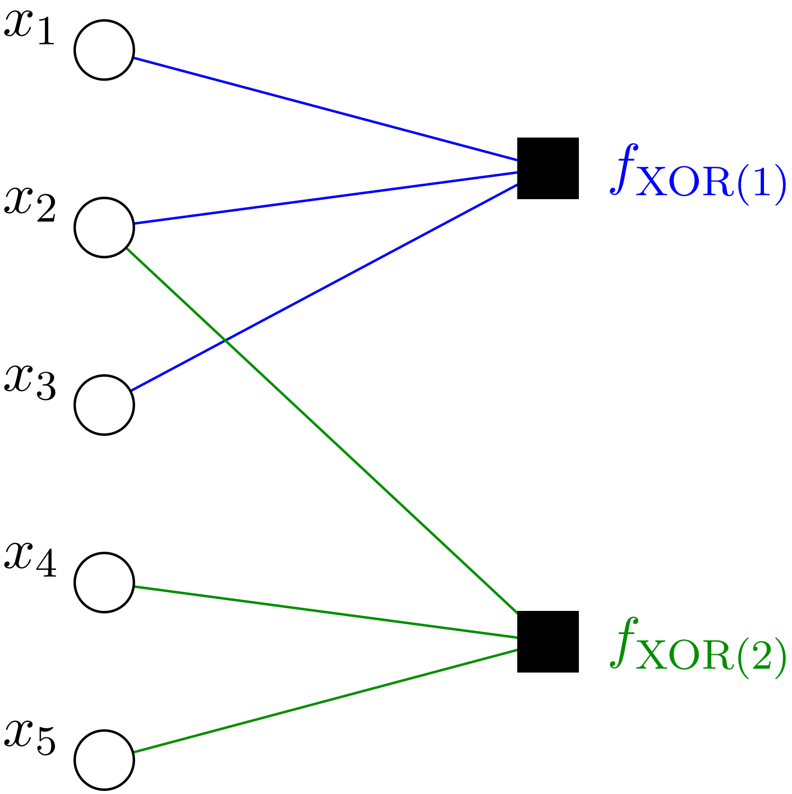


$$f_{\text{XOR}}(x_1, x_2, x_3) \;=\; [x_1 + x_2 + x_3 = 0 \pmod 2]$$

$$f_{\text{XOR}}(x_2, x_4, x_5) \;=\; [x_2 + x_4 + x_5 = 0 \pmod 2]$$

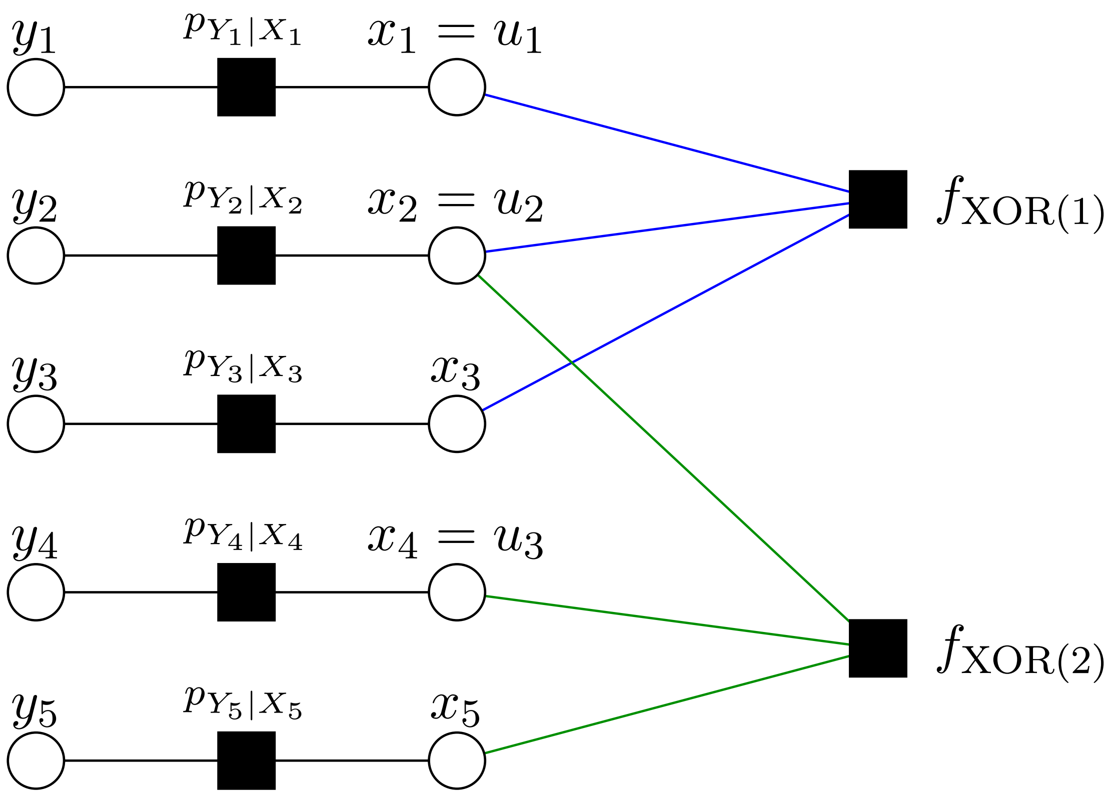
FG of a Data Communication System based on a parity-check code

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$



FG of a Data Communication System based on a parity-check code

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

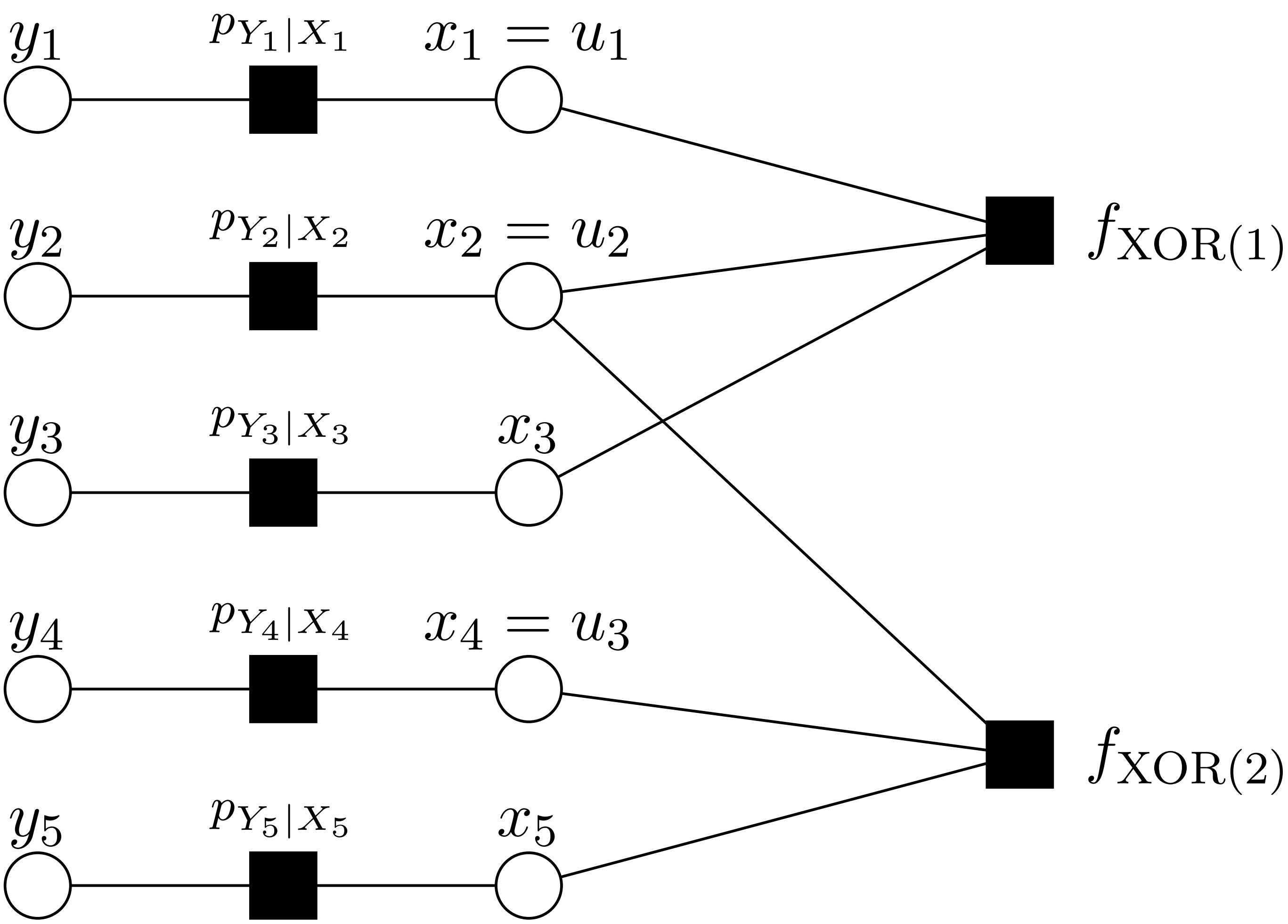


Symbol-Wise MAP Decoding

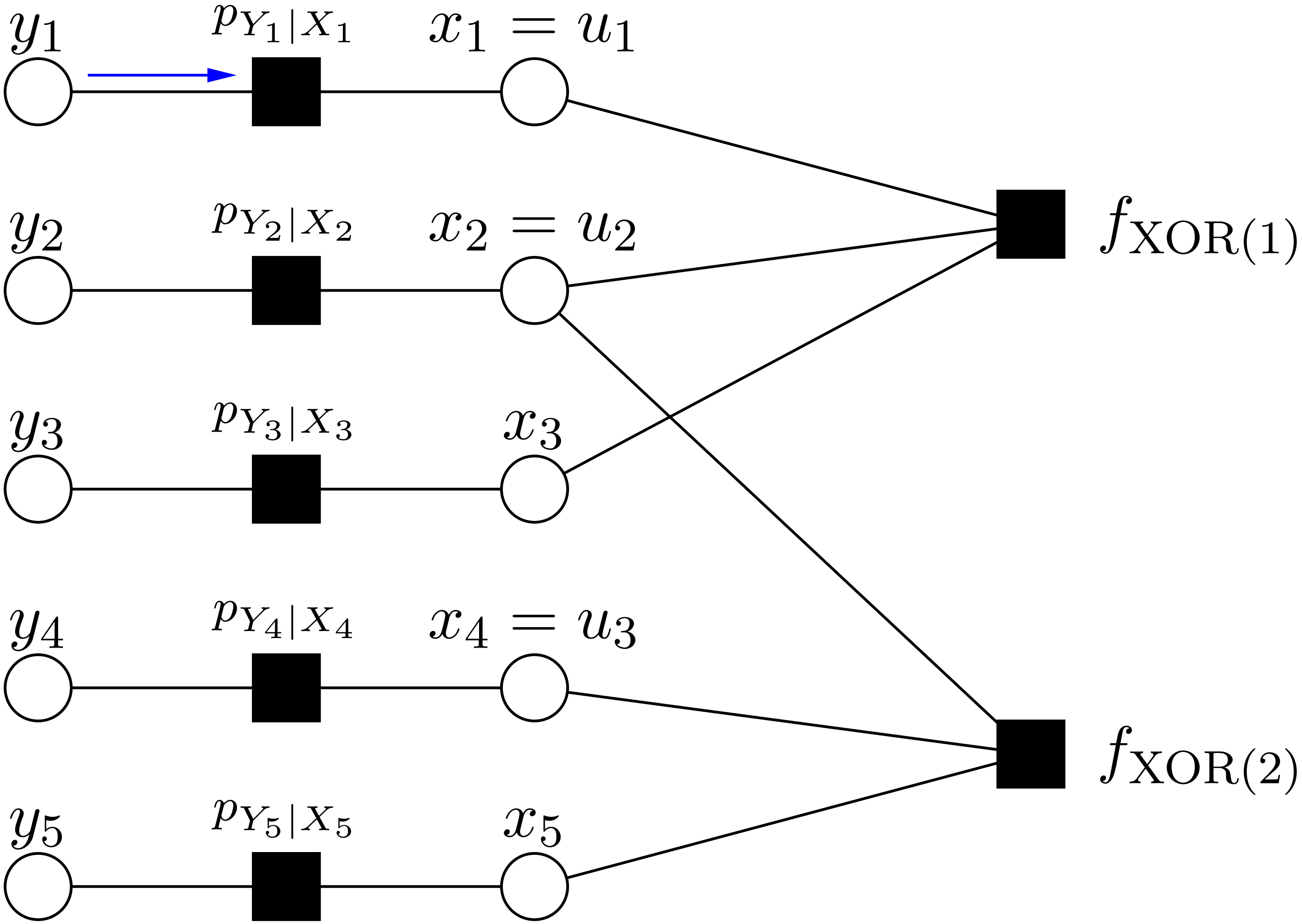
Remember, symbol-wise MAP decoding for symbol i can be written as

$$\hat{u}_i^{\text{symbol}}(\mathbf{y}) = \underbrace{\underbrace{\operatorname*{argmax}_{u_i\in\mathcal{U}}}_{\text{Decision taking}} \underbrace{\sum_{\substack{\mathbf{u}\in\mathcal{U}^k,\ \mathbf{x}\in\mathcal{X}^n\\ u_i\ \text{fixed}}} \underbrace{P_{\mathbf{UXY}}(\mathbf{u},\mathbf{x},\mathbf{y}),}_{\text{Joint pmf/pdf}}}_{\text{Marginal function}}}_{\text{Decision about symbol } u_i \text{ based on symbol-wise decoding}}$$

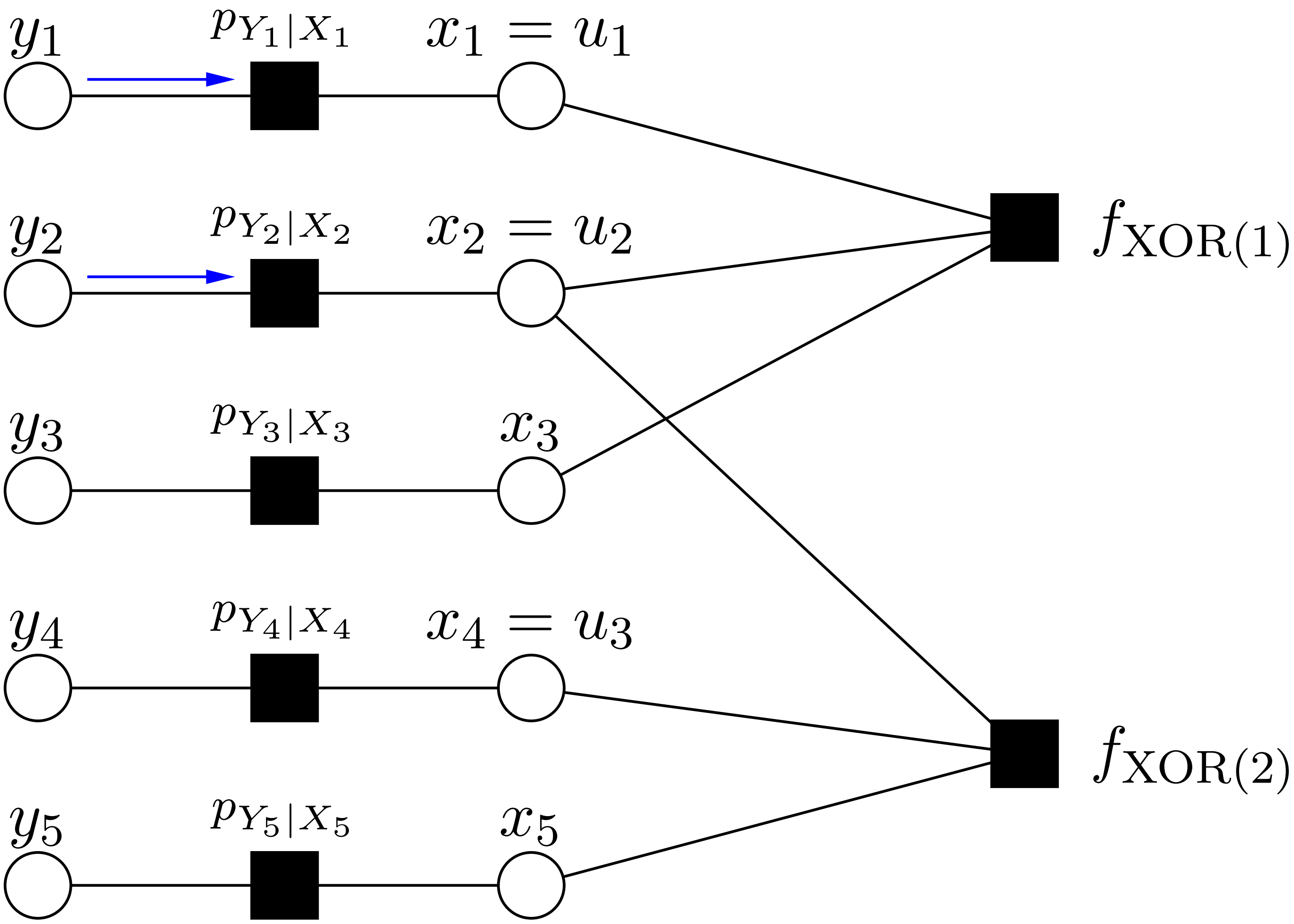
SPA Decoding (Factor graph without cycles)



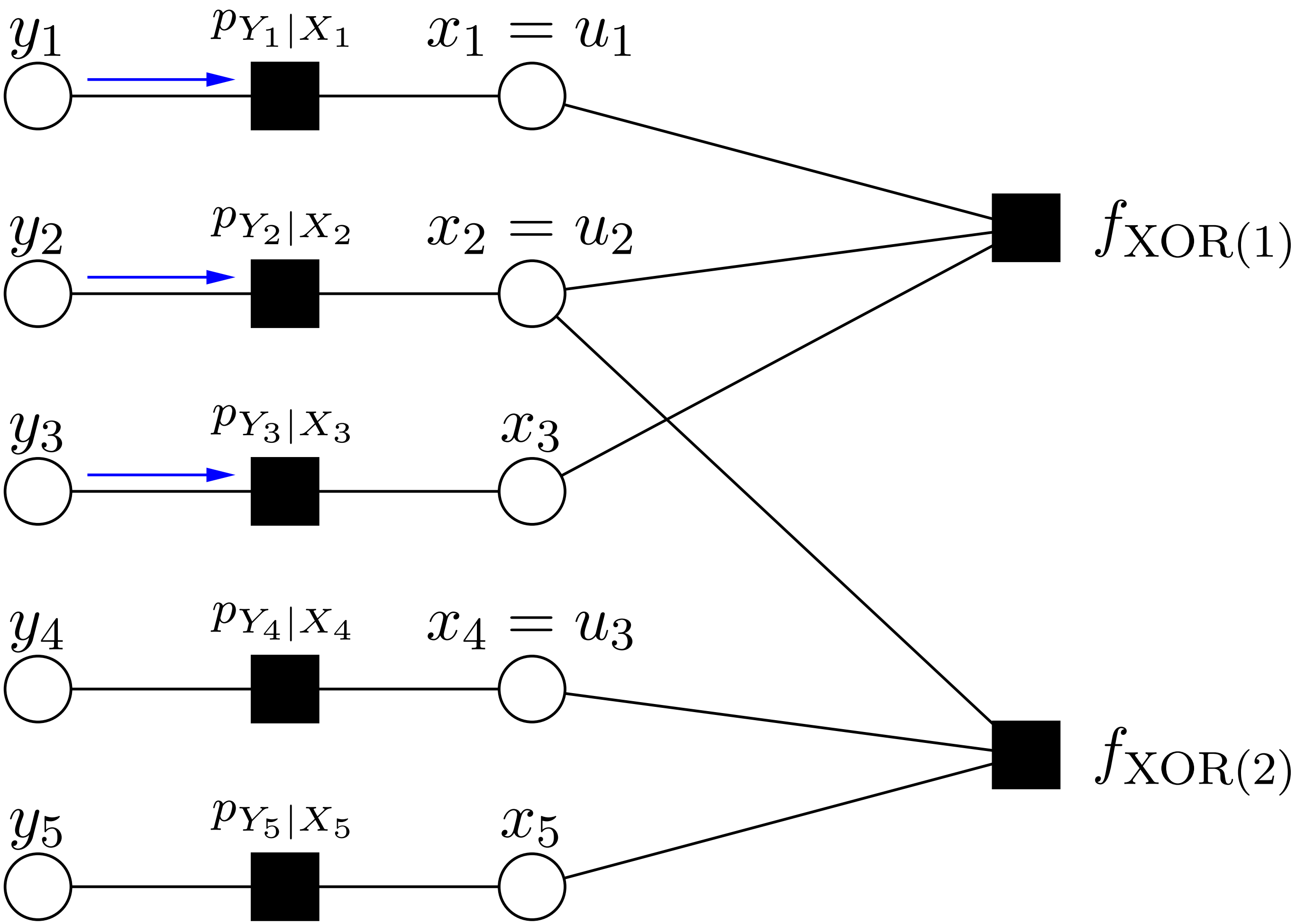
SPA Decoding (Factor graph without cycles)



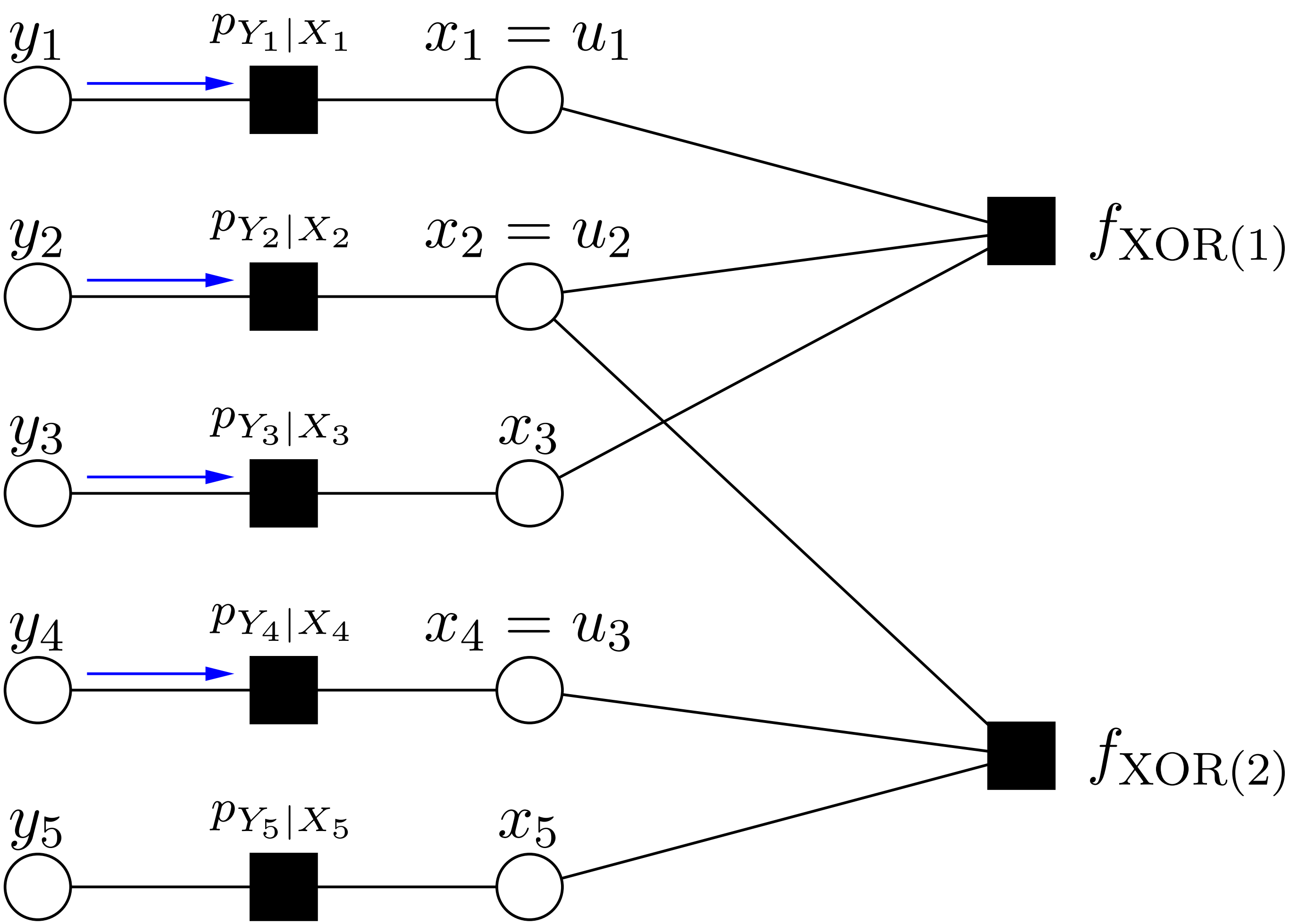
SPA Decoding (Factor graph without cycles)



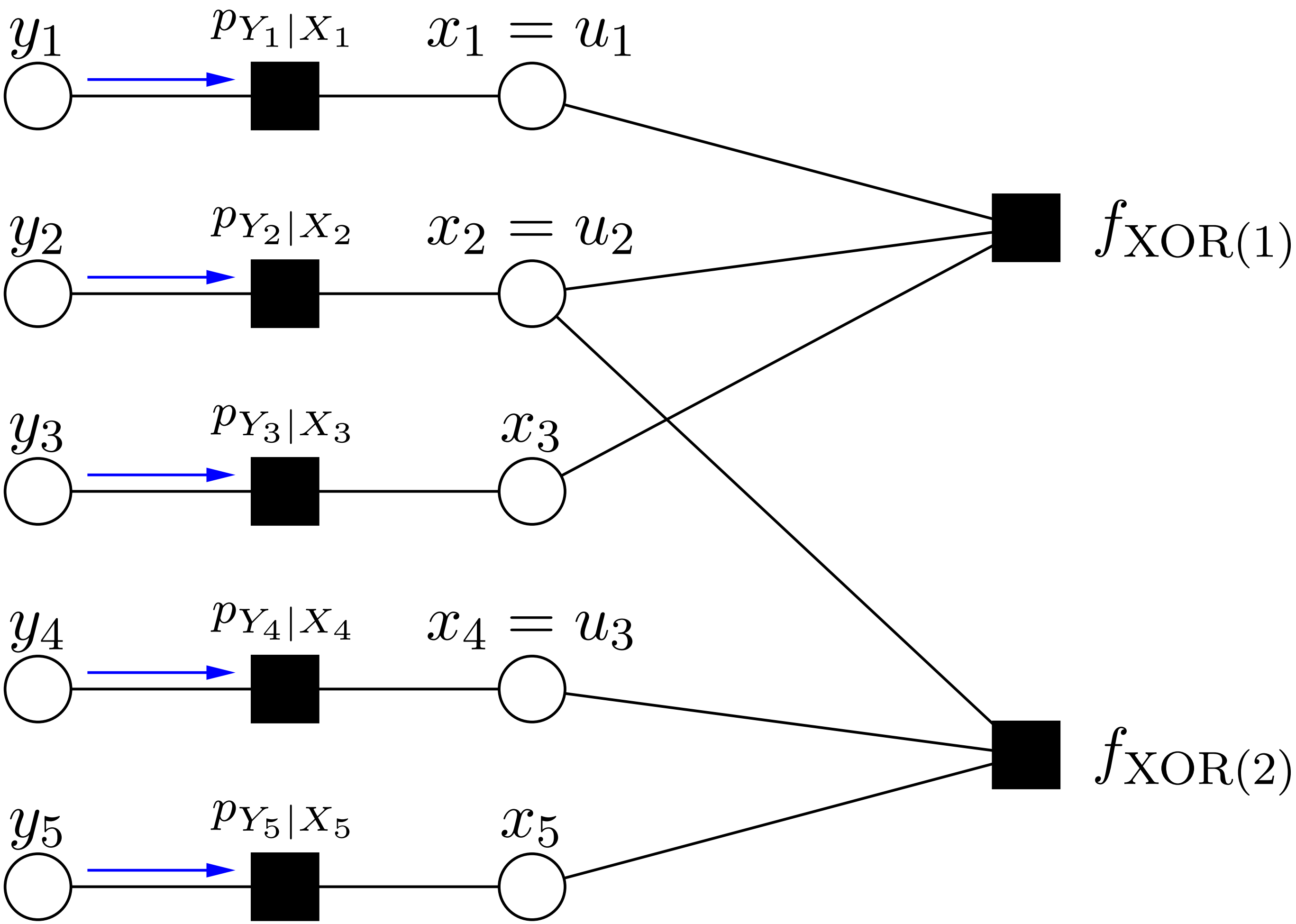
SPA Decoding (Factor graph without cycles)



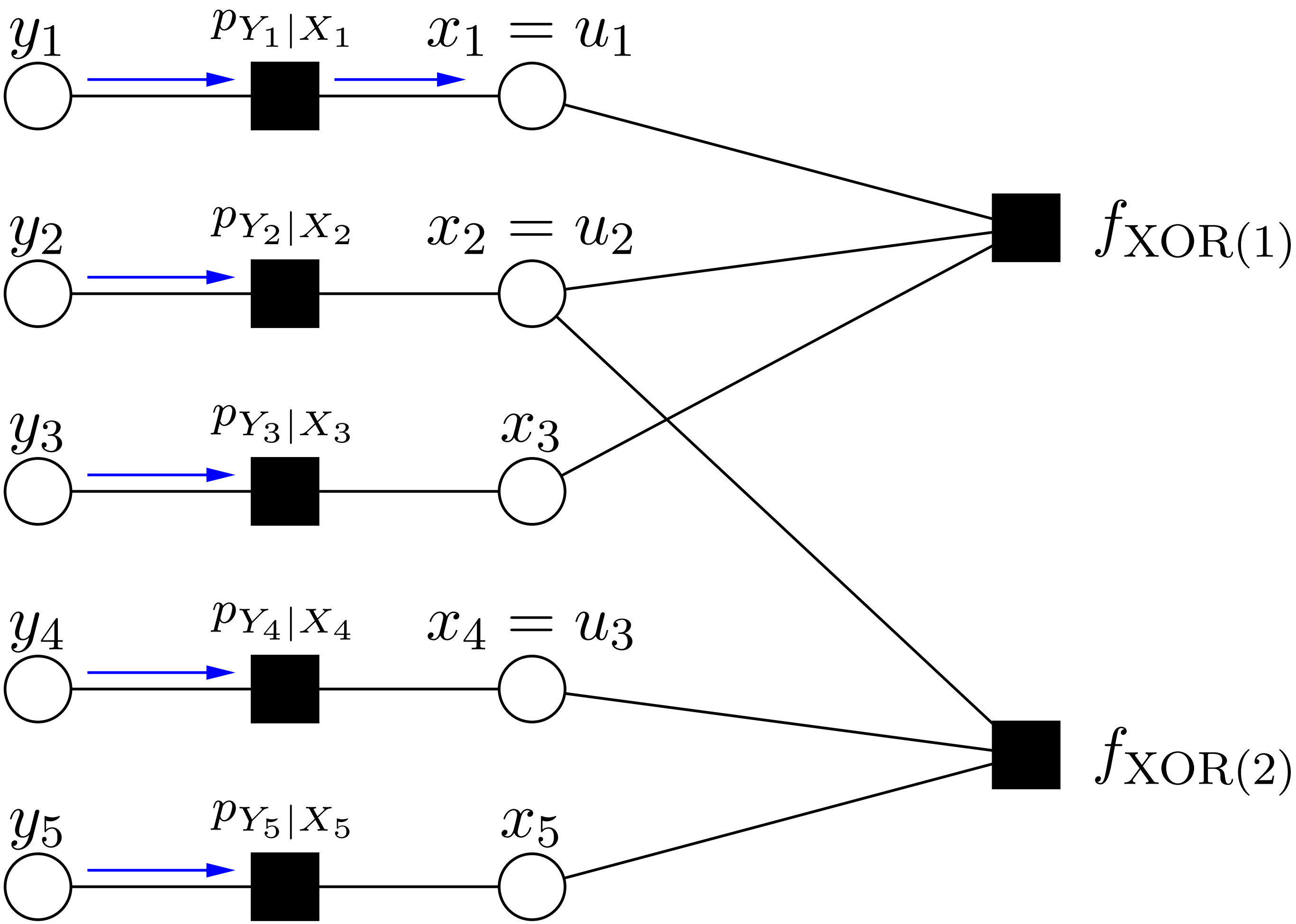
SPA Decoding (Factor graph without cycles)



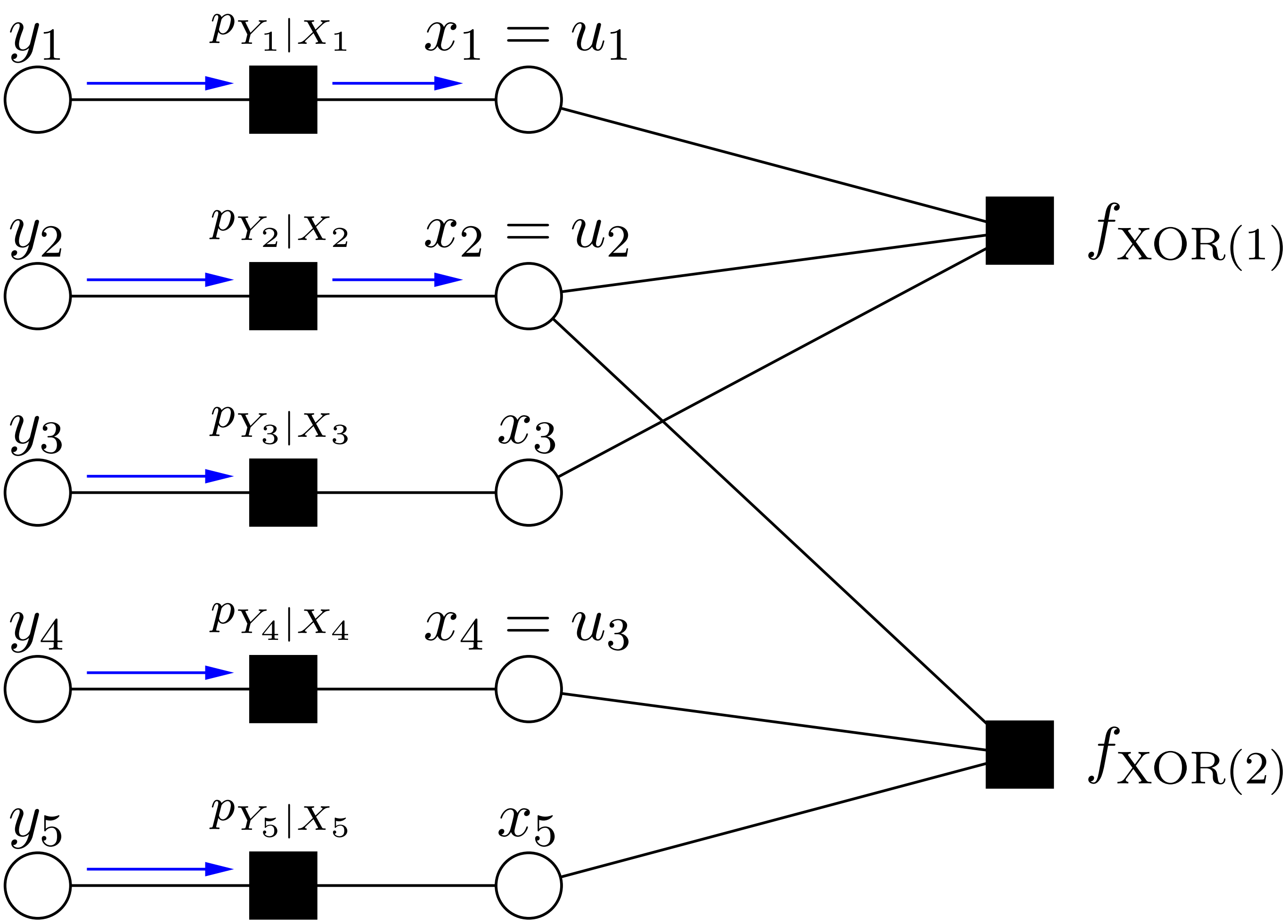
SPA Decoding (Factor graph without cycles)



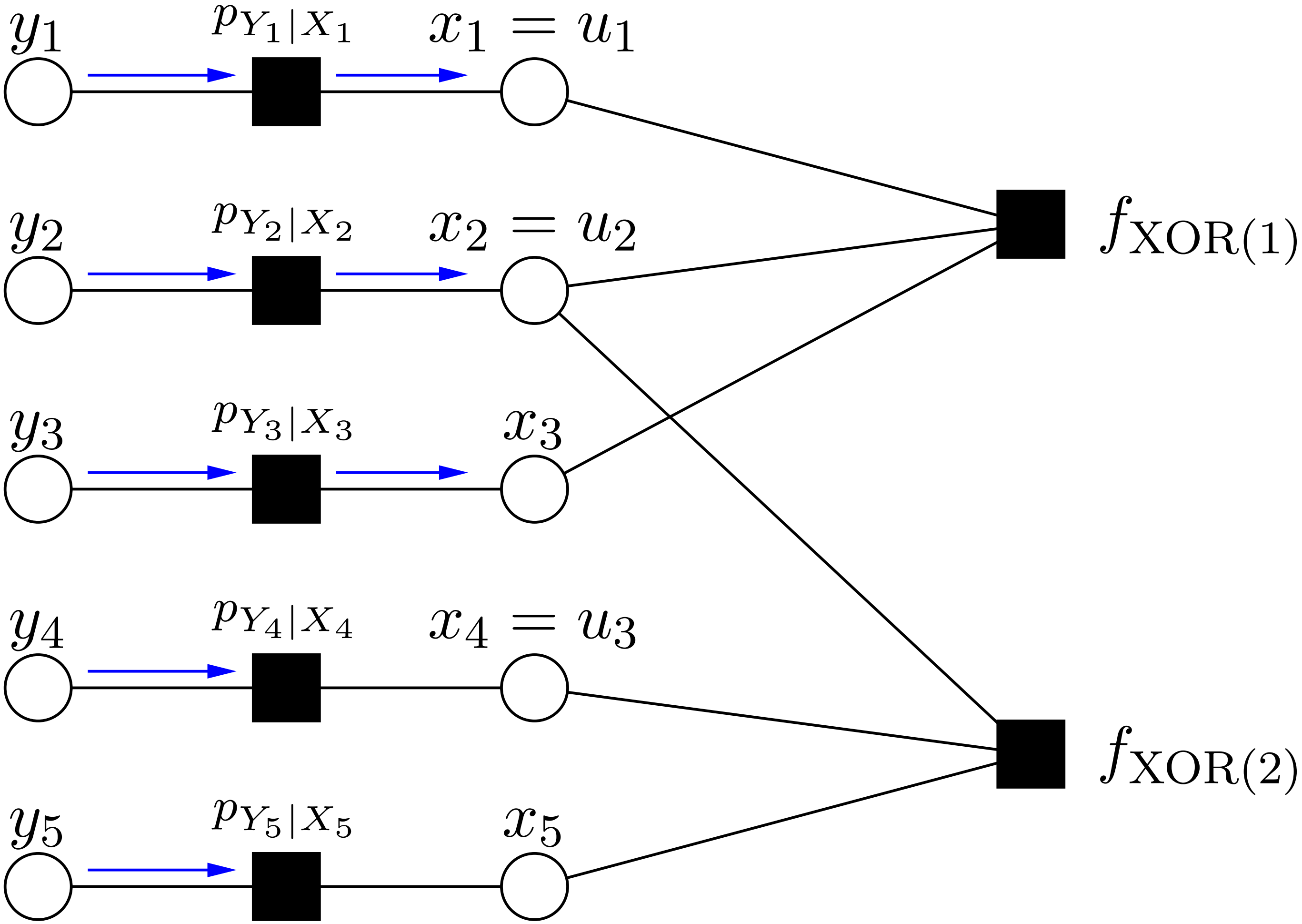
SPA Decoding (Factor graph without cycles)



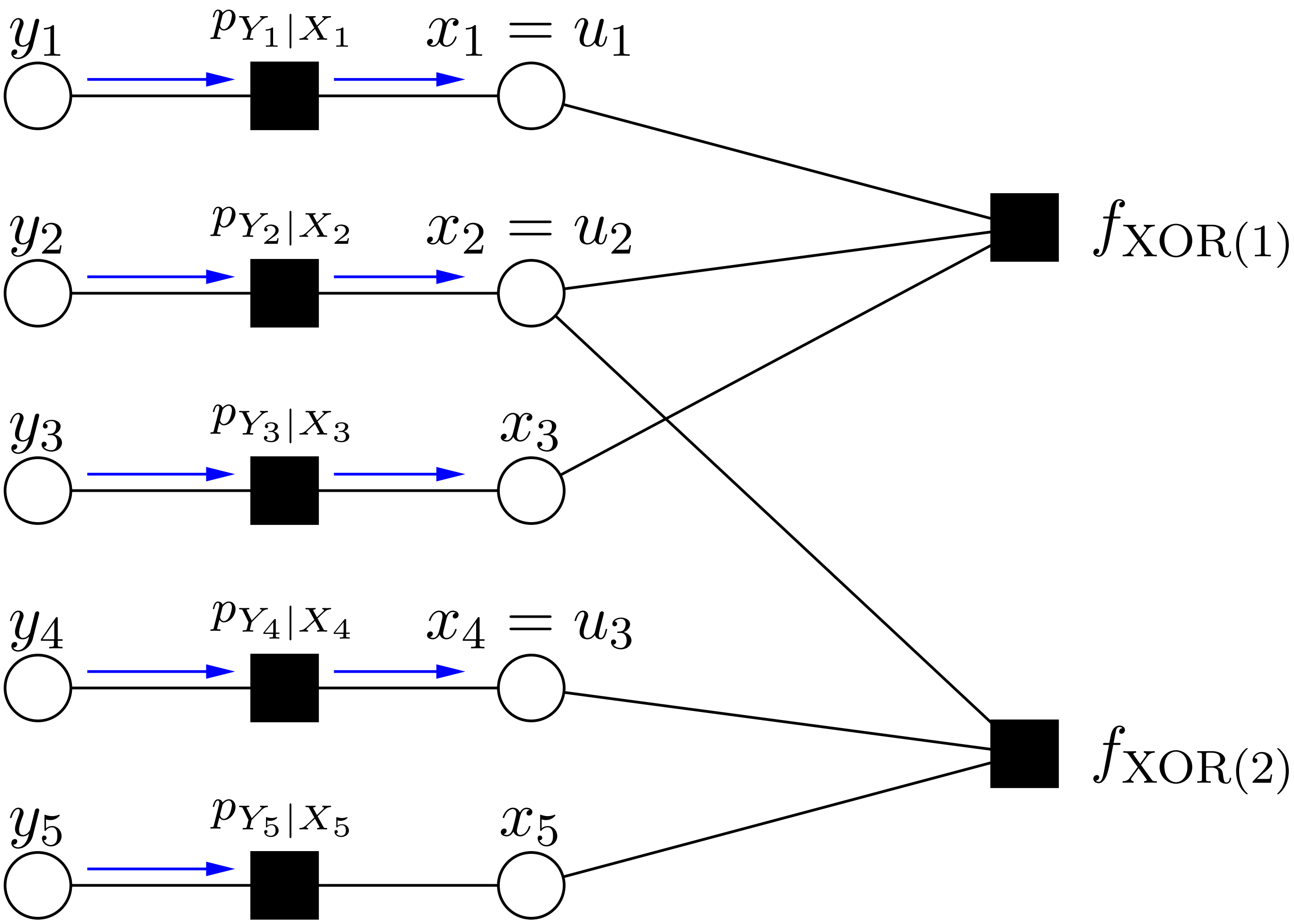
SPA Decoding (Factor graph without cycles)



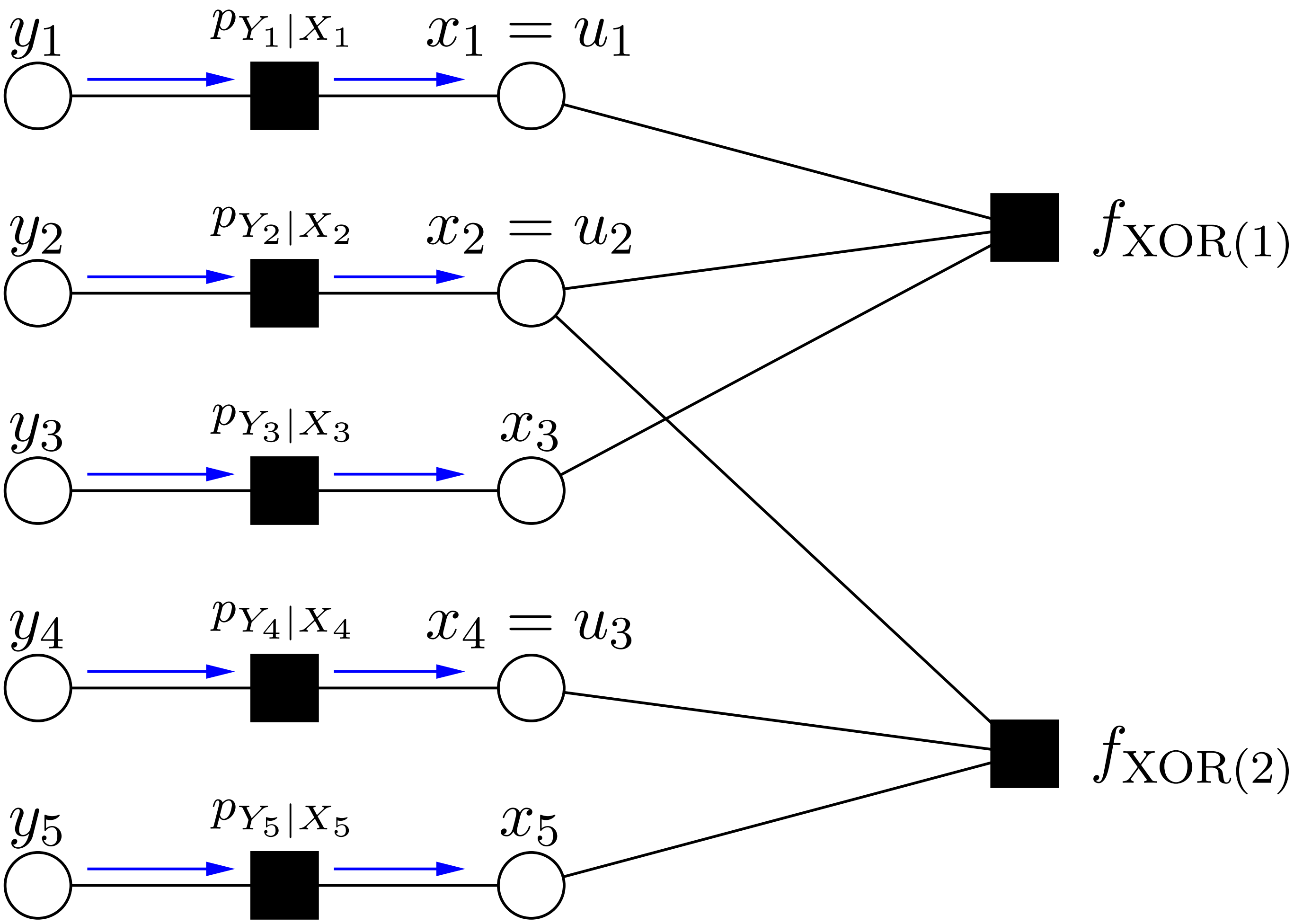
SPA Decoding (Factor graph without cycles)



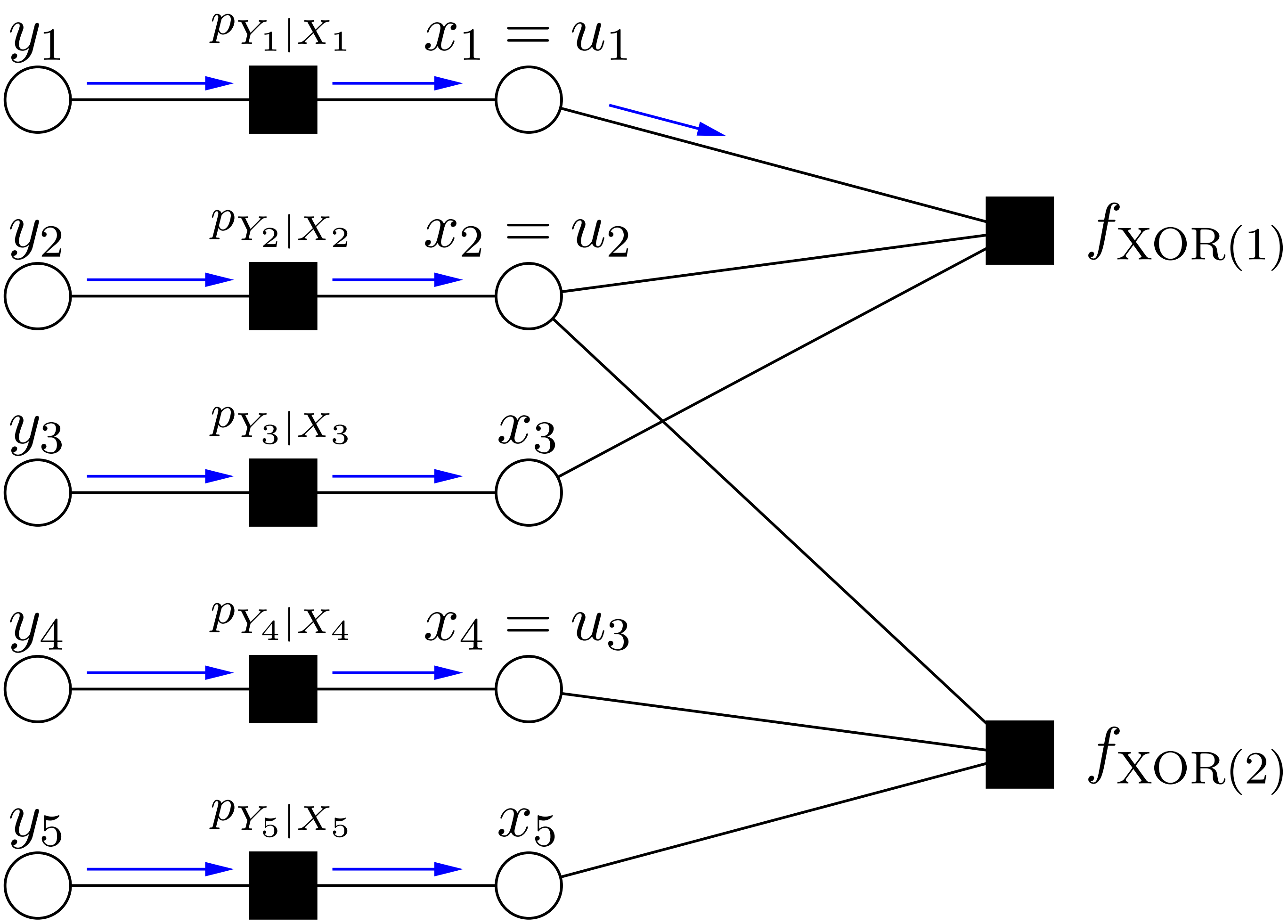
SPA Decoding (Factor graph without cycles)



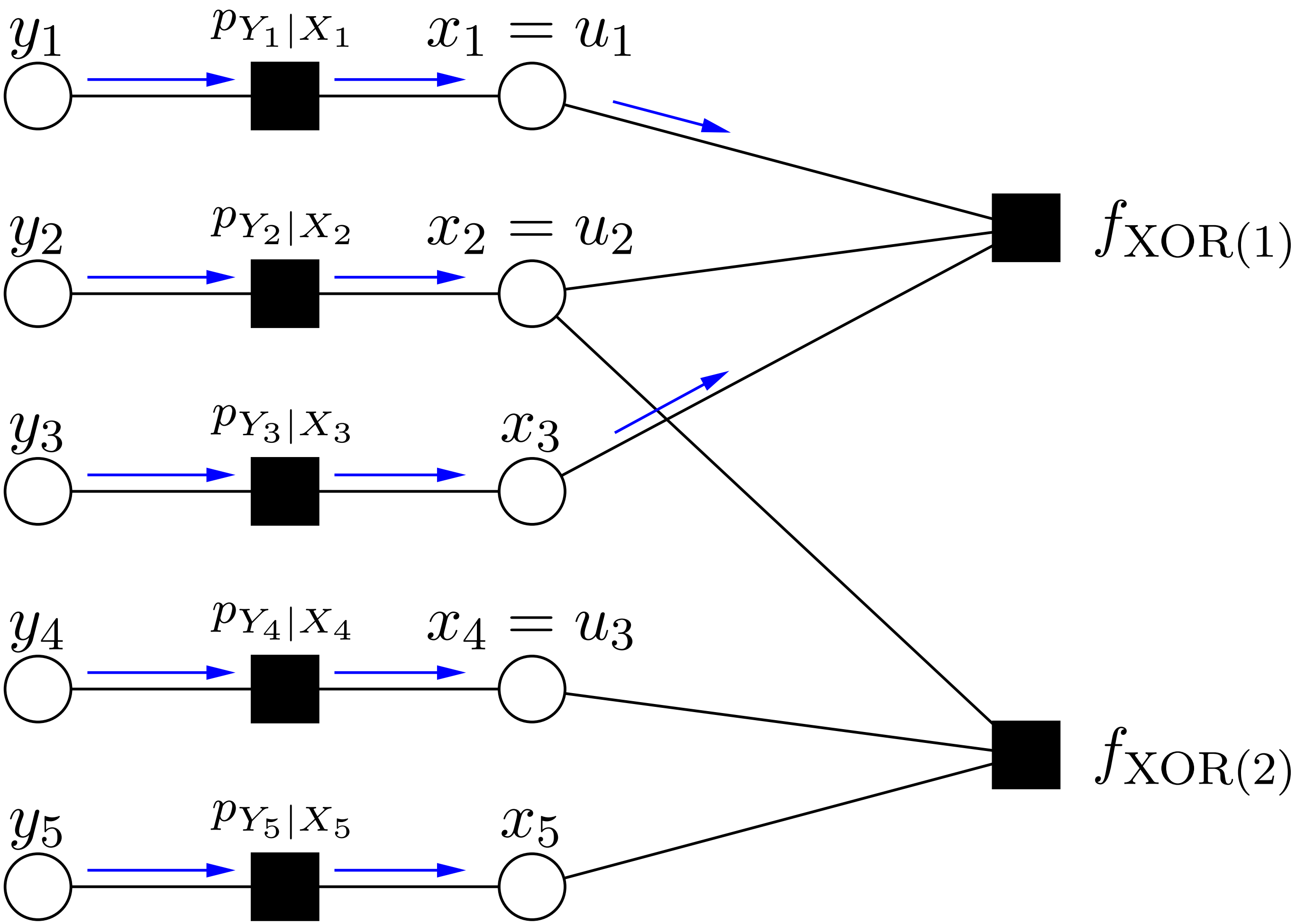
SPA Decoding (Factor graph without cycles)



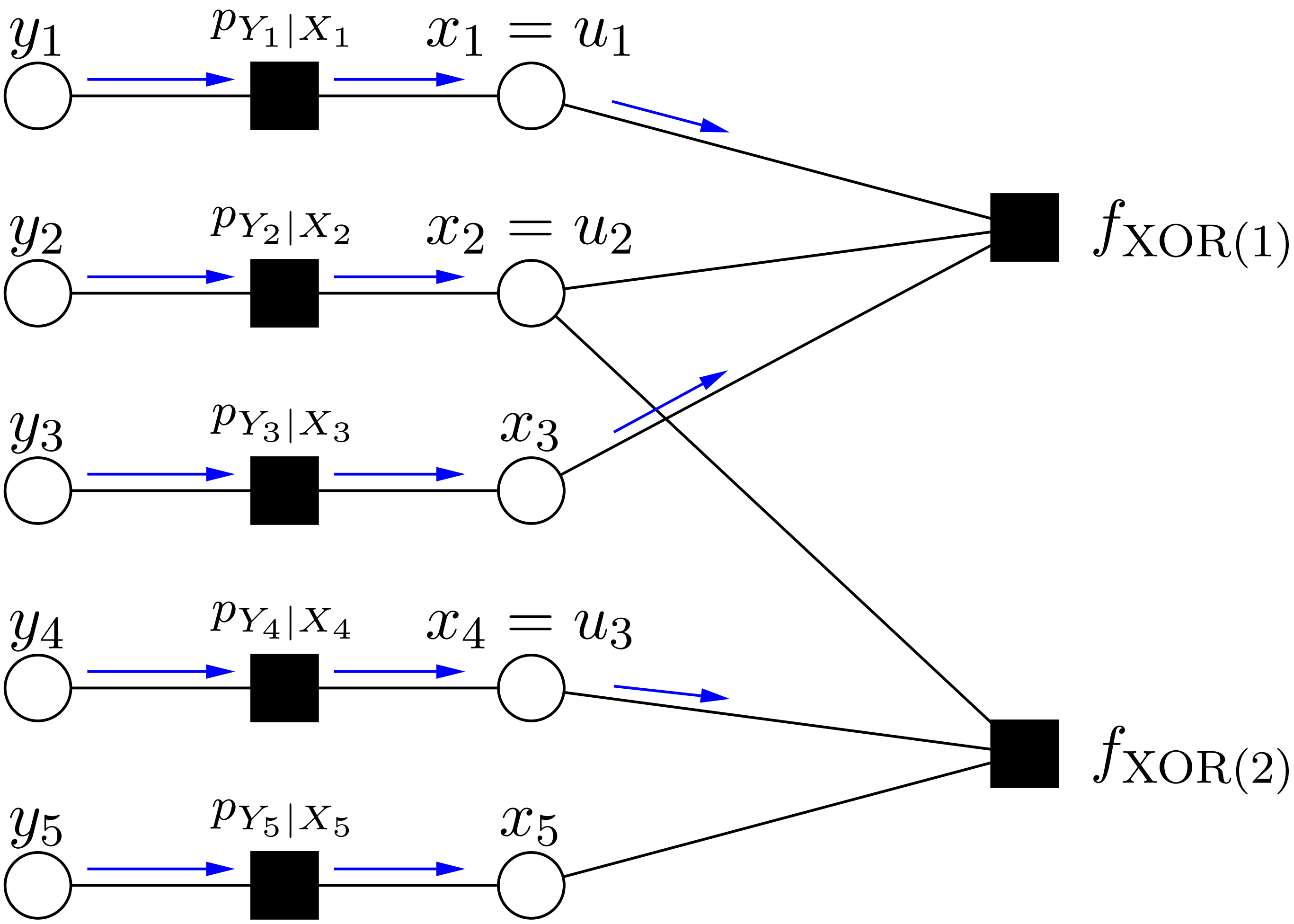
SPA Decoding (Factor graph without cycles)



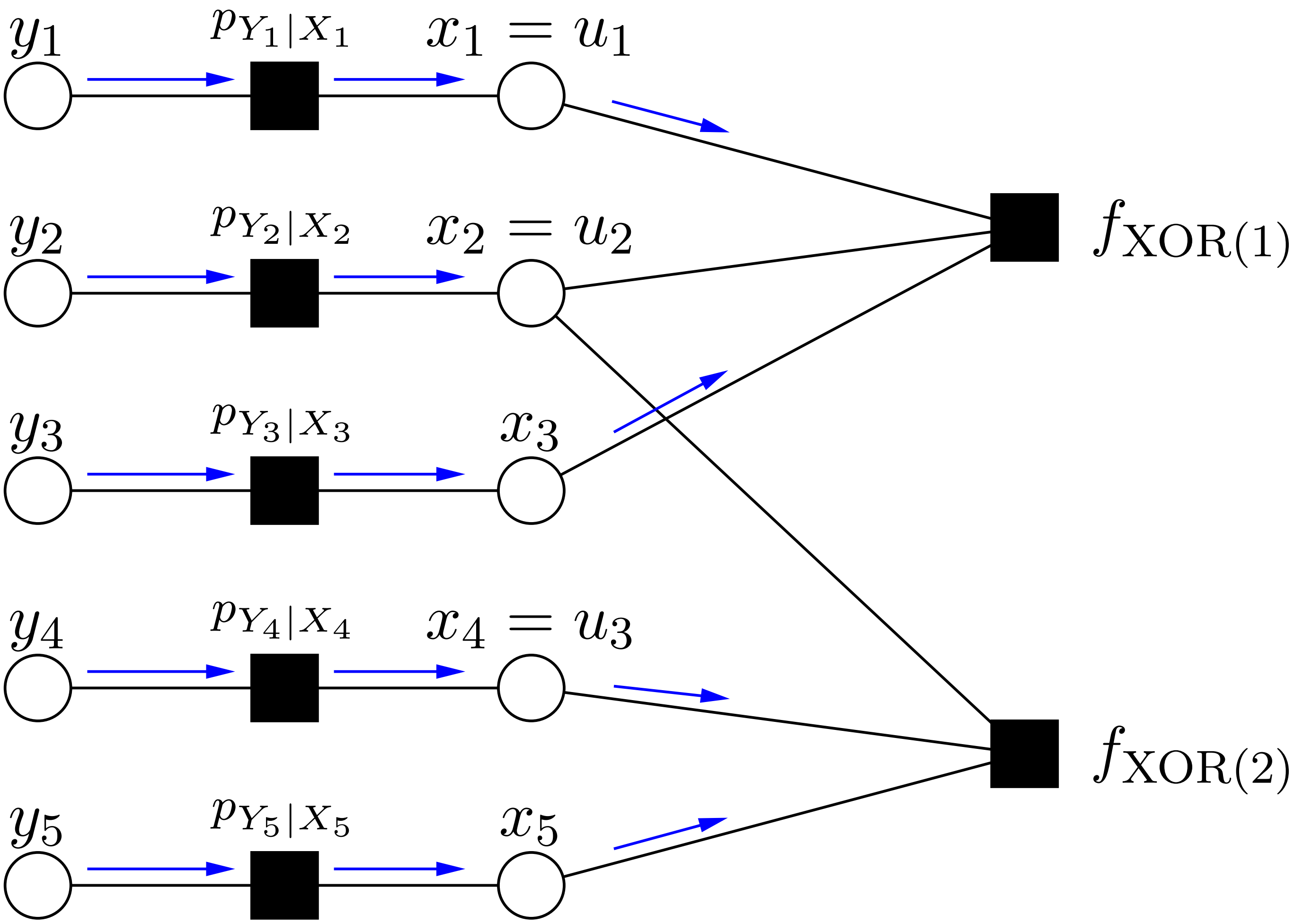
SPA Decoding (Factor graph without cycles)



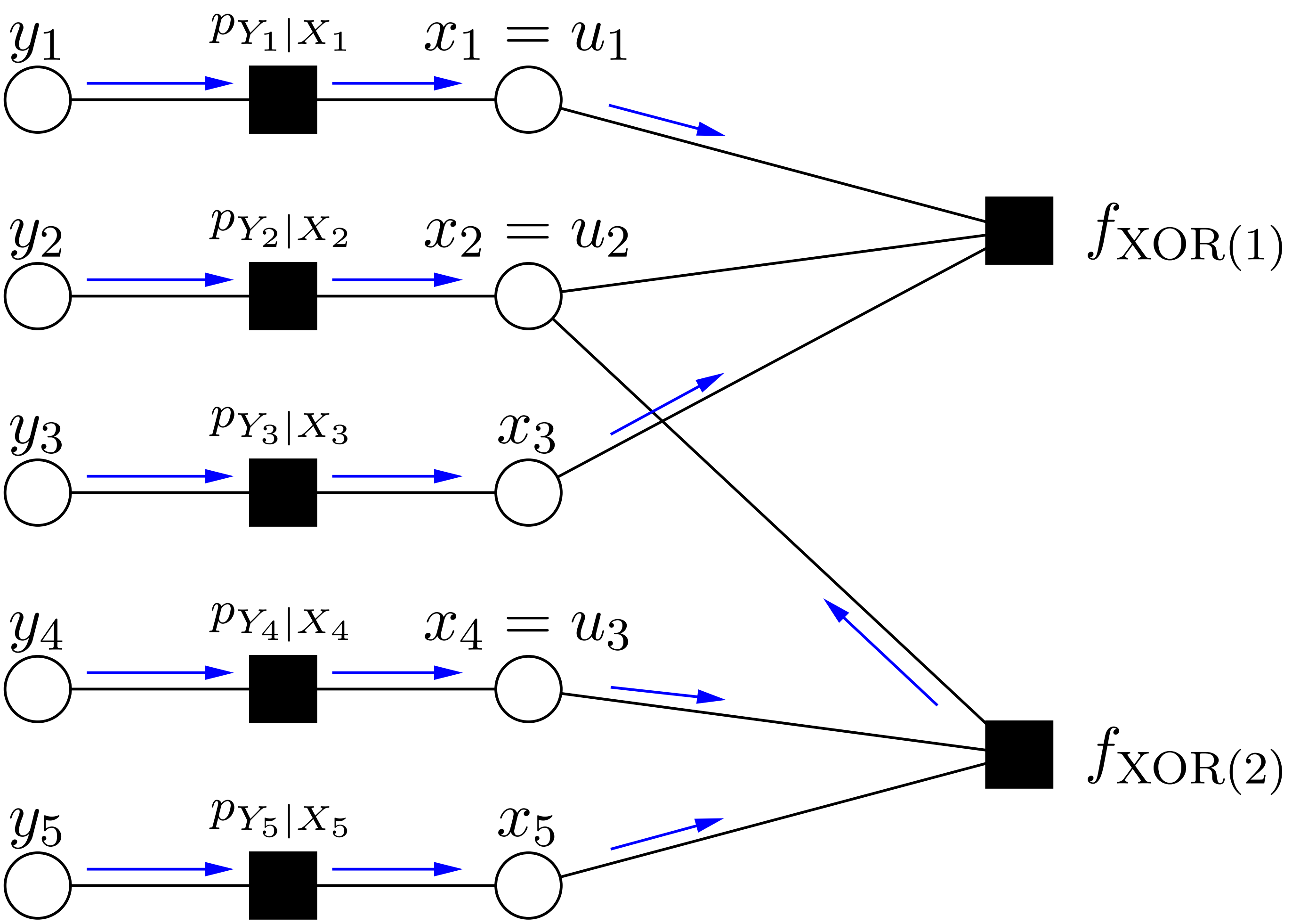
SPA Decoding (Factor graph without cycles)



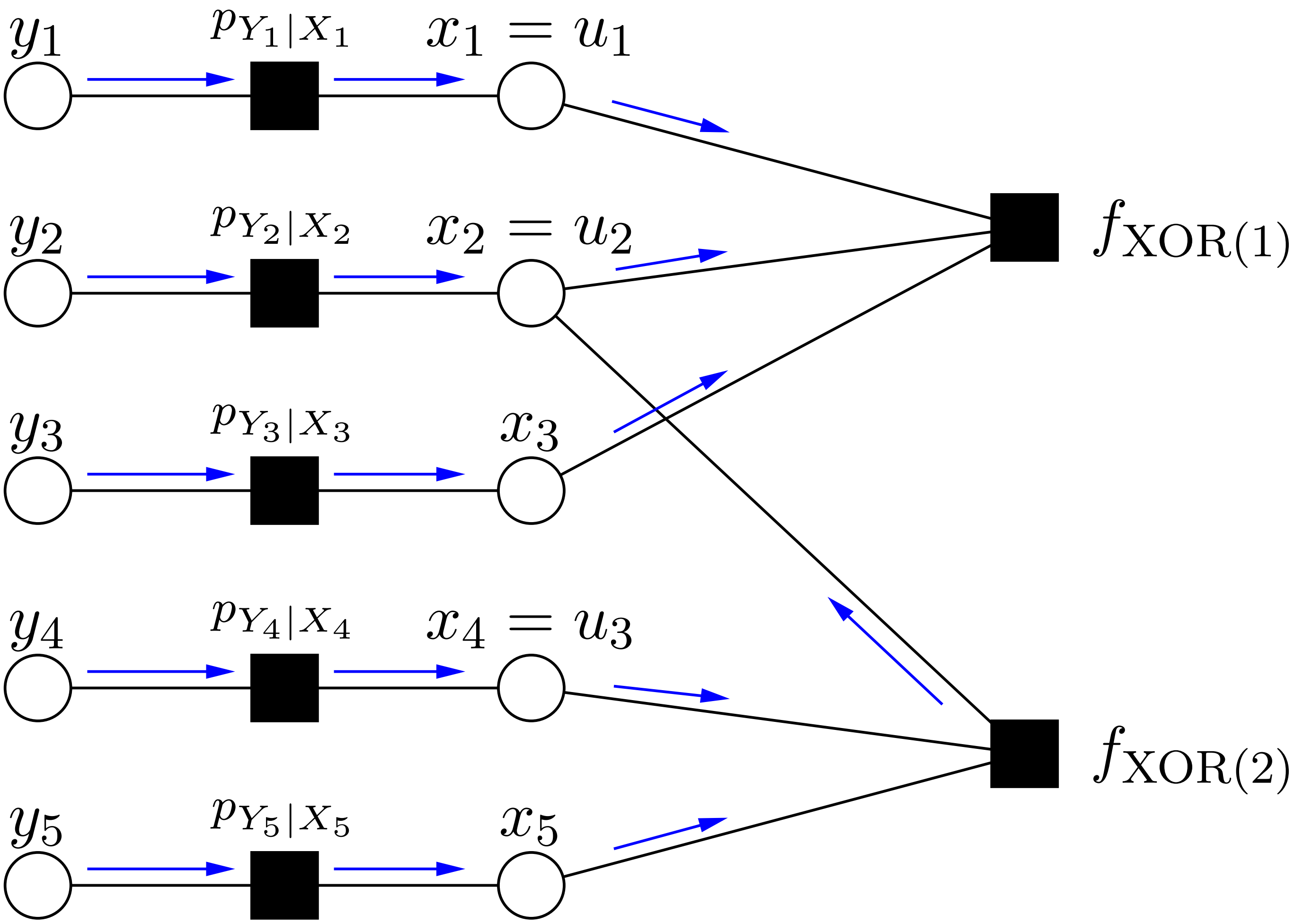
SPA Decoding (Factor graph without cycles)



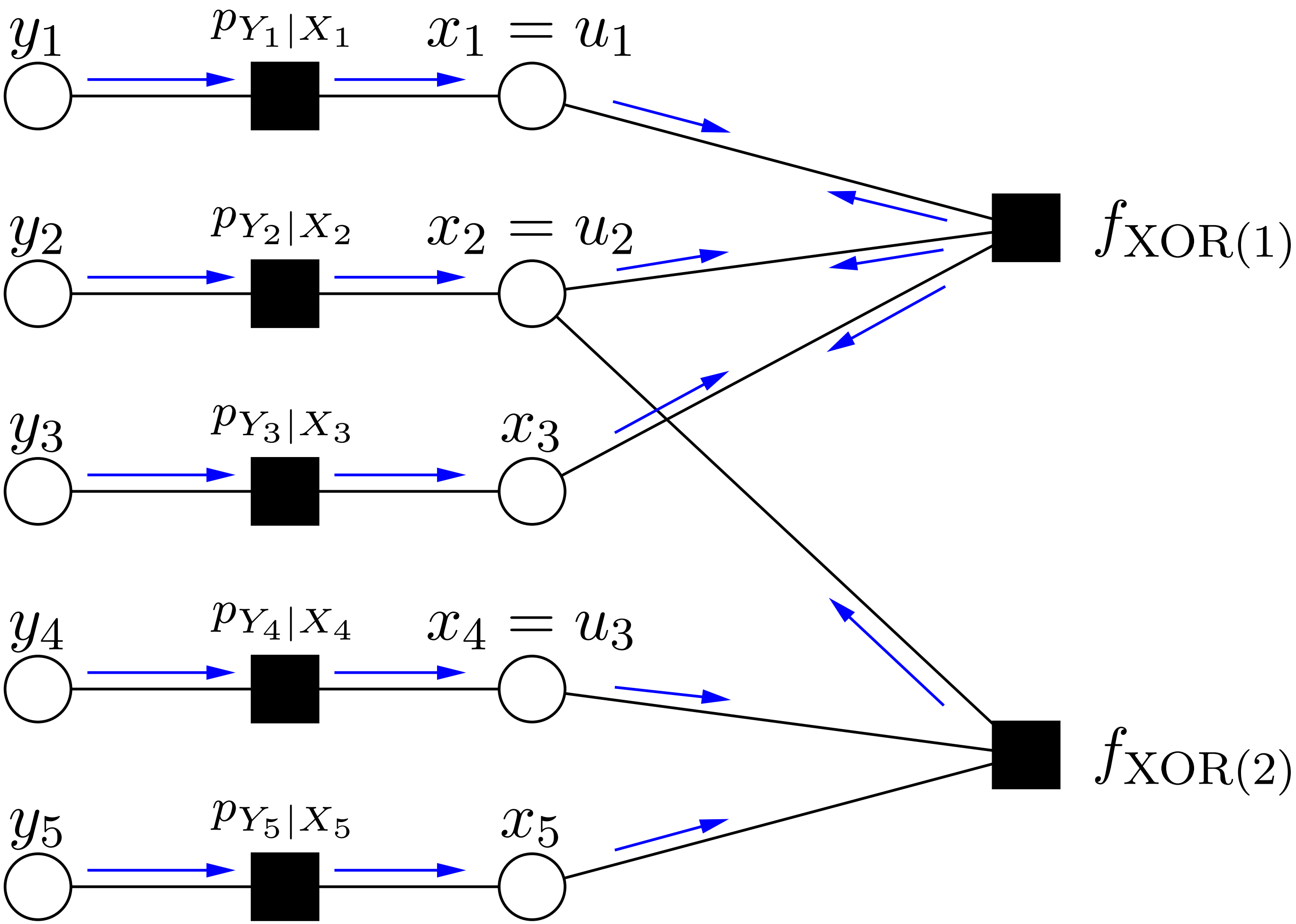
SPA Decoding (Factor graph without cycles)



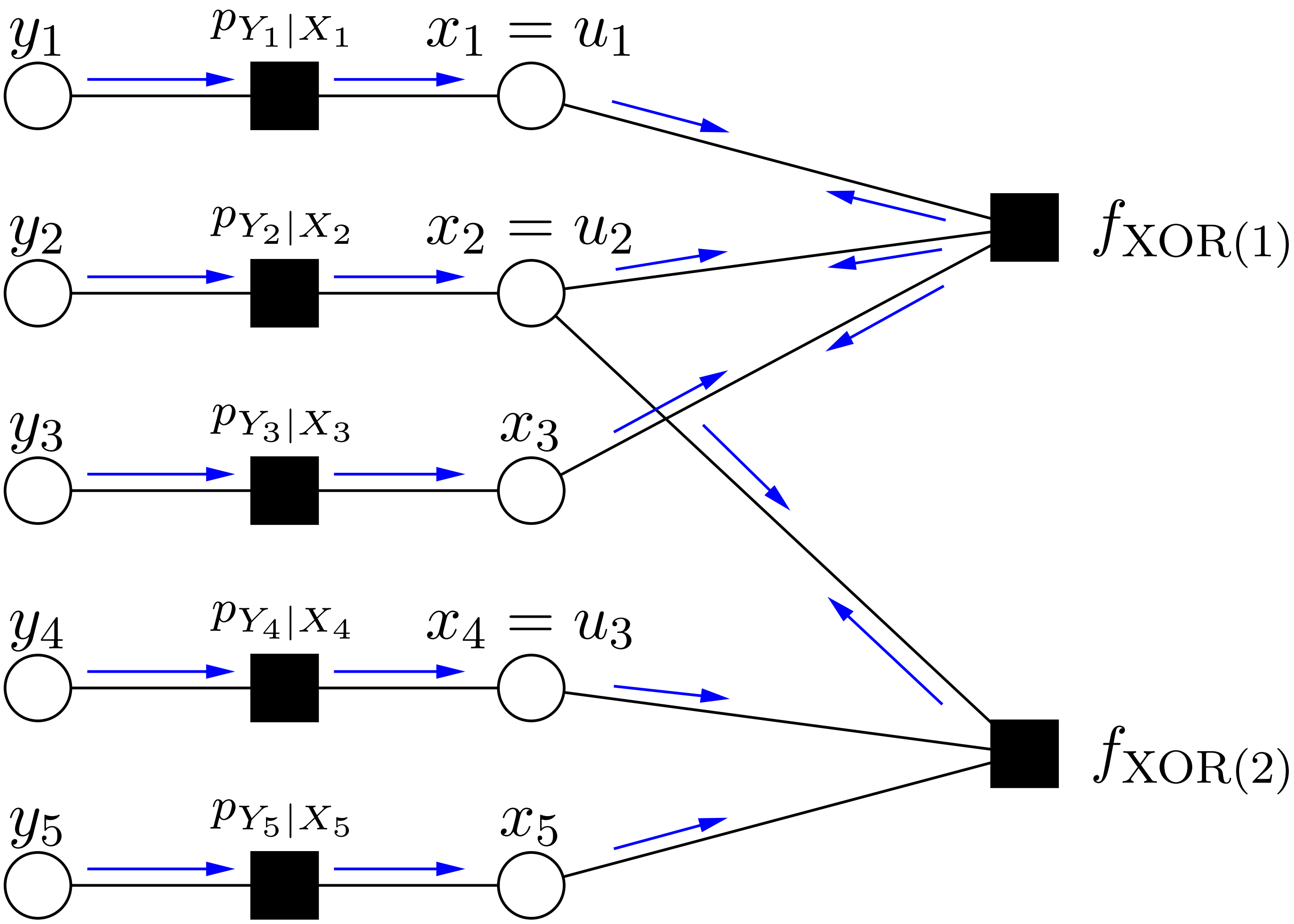
SPA Decoding (Factor graph without cycles)



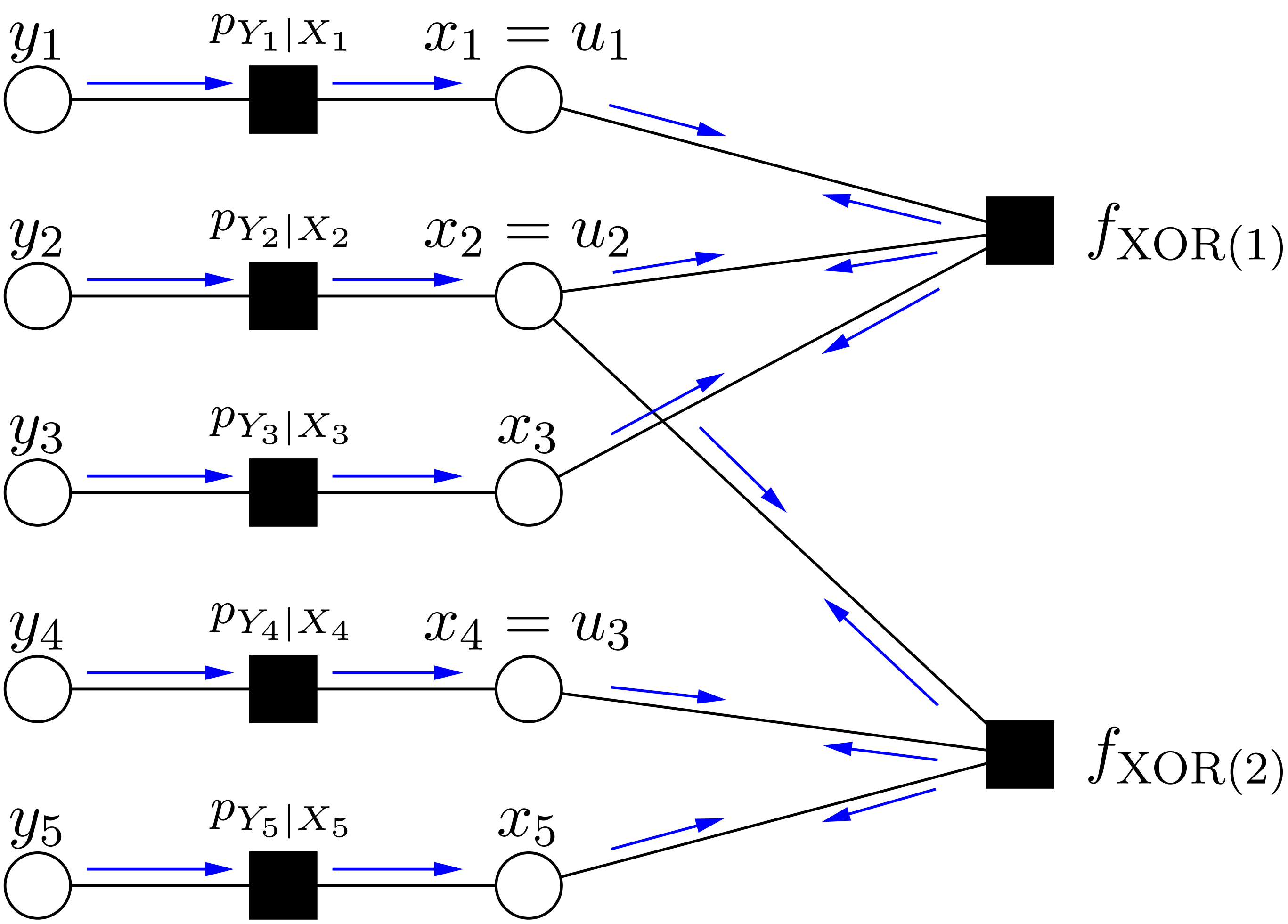
SPA Decoding (Factor graph without cycles)



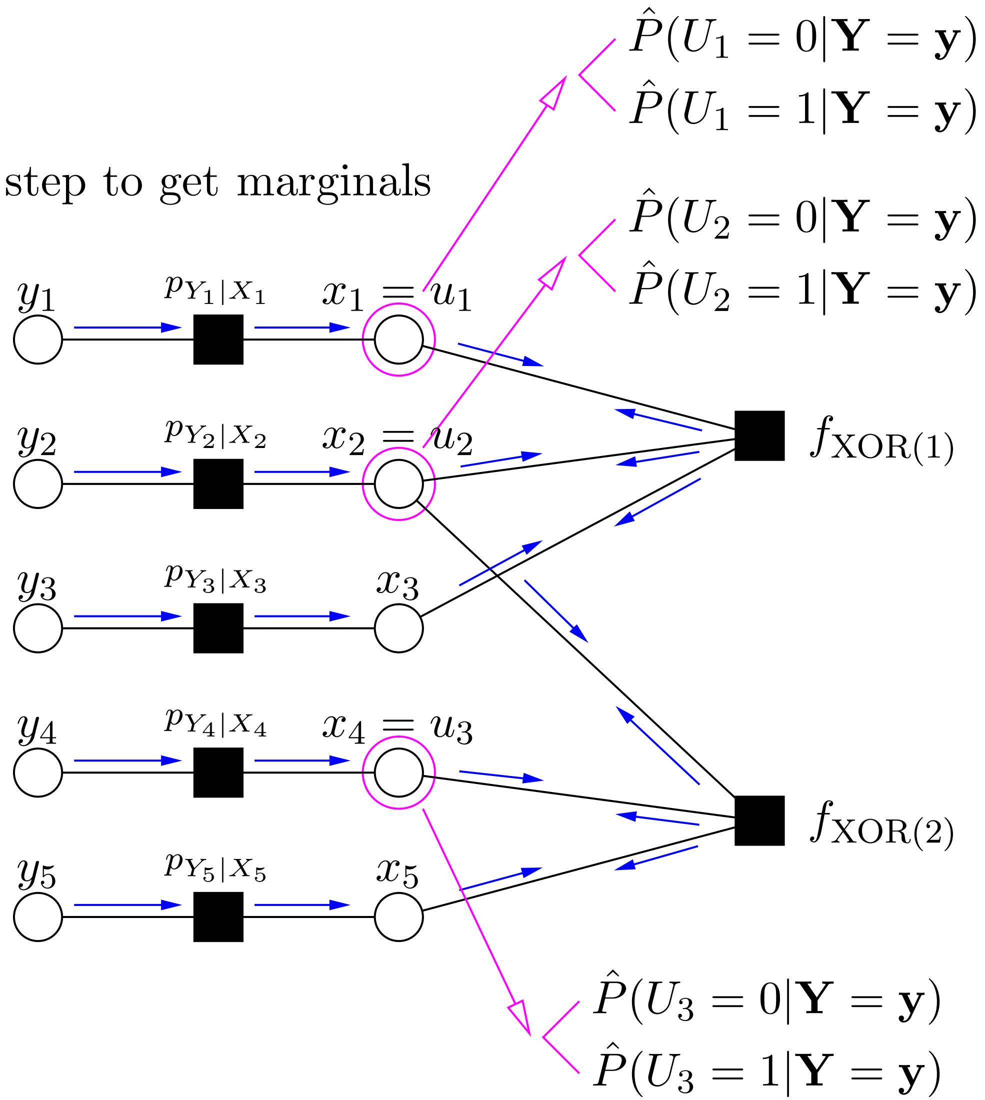
SPA Decoding (Factor graph without cycles)



SPA Decoding (Factor graph without cycles)



SPA Decoding (Factor graph without cycles)


$\hat{P}(U_1 = 0|\mathbf{Y} = \mathbf{y})$
$\hat{P}(U_1 = 1|\mathbf{Y} = \mathbf{y})$
step to get marginals
$\hat{P}(U_2 = 0|\mathbf{Y} = \mathbf{y})$
$\hat{P}(U_2 = 1|\mathbf{Y} = \mathbf{y})$
y_1
$p_{Y_1|X_1}$
$x_1 = u_1$
y_2
$p_{Y_2|X_2}$
$x_2 = u_2$
$f_{\text{XOR}(1)}$
y_3
$p_{Y_3|X_3}$
x_3
y_4
$p_{Y_4|X_4}$
$x_4 = u_3$
$f_{\text{XOR}(2)}$
y_5
$p_{Y_5|X_5}$
x_5
$\hat{P}(U_3 = 0|\mathbf{Y} = \mathbf{y})$
$\hat{P}(U_3 = 1|\mathbf{Y} = \mathbf{y})$

Symbol-Wise MAP Decoding

Remember, symbol-wise MAP decoding for symbol i can be written as

$$\hat{u}_i^{\text{symbol}}(\mathbf{y}) = \underbrace{\underbrace{\operatorname*{argmax}_{u_i \in \mathcal{U}}}_{\text{Decision taking}} \underbrace{\sum_{\substack{\mathbf{u}\in\mathcal{U}^k,\ \mathbf{x}\in\mathcal{X}^n \\ u_i \text{ fixed}}} \underbrace{P_{\mathbf{UXY}}(\mathbf{u},\mathbf{x},\mathbf{y})}_{\text{Joint pmf/pdf}}}_{\text{Marginal function}},}_{\text{Decision about symbol } u_i \text{ based on symbol-wise decoding}}$$

Sum-Product Algorithm Decoding

Sum-product algorithm (SPA) decoding:

$$\hat{u}_i^{\text{symbol}}(\mathbf{y}) \approx \underbrace{\underbrace{\operatorname*{argmax}_{u_i\in\mathcal{U}}}_{\text{Decision taking}} \underbrace{\sum_{\substack{\mathbf{u}\in\mathcal{U}^k,\ \mathbf{x}\in\mathcal{X}^n \\ u_i\ \text{fixed}}} \underbrace{P_{\mathbf{UXY}}(\mathbf{u},\mathbf{x},\mathbf{y}),}_{\text{Joint pmf/pdf}}}_{\text{Marginal function is approximated by Sum-Product Algorithm}}}_{\text{Decision about symbol } u_i \text{ based on symbol-wise decoding}}$$

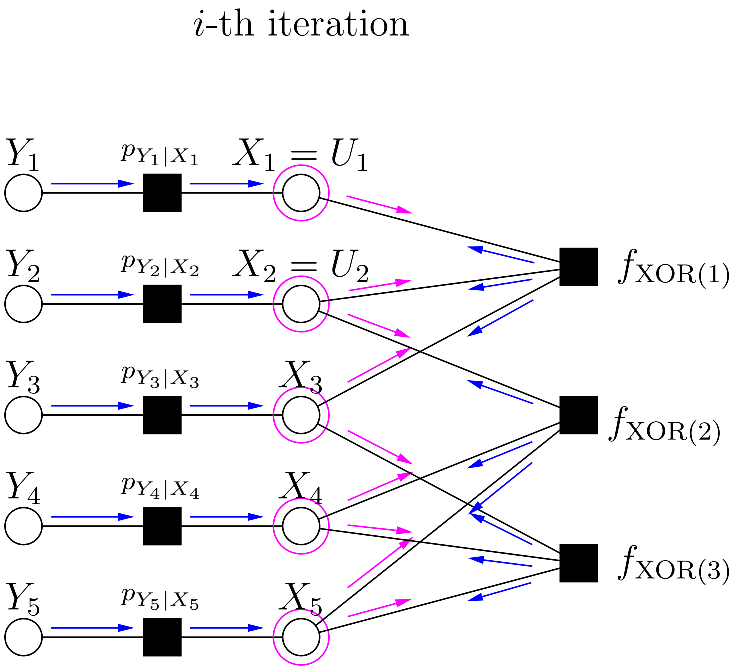
Sum-Product Algorithm Decoding

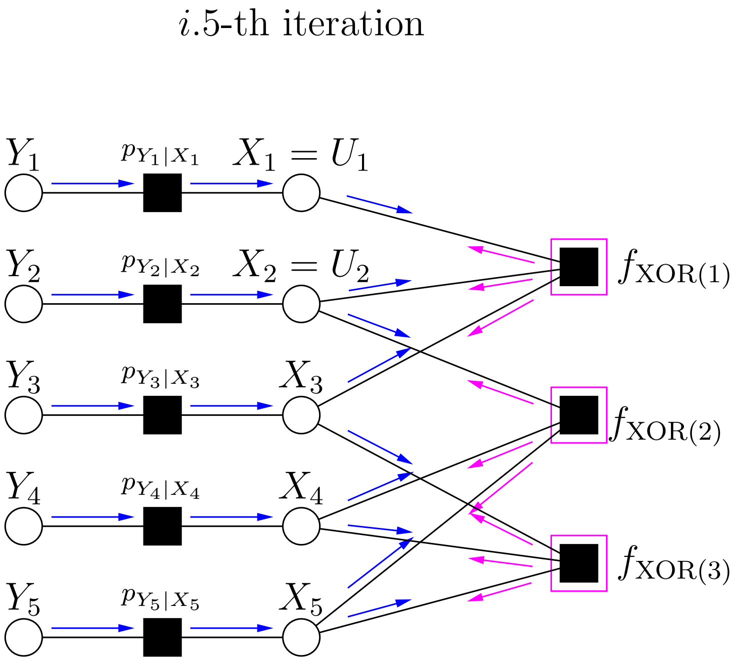
Sum-product algorithm (SPA) decoding:

$$\hat{u}_i^{\text{symbol}}(\mathbf{y}) \approx \underbrace{\underbrace{\operatorname*{argmax}_{u_i \in \mathcal{U}}}_{\text{Decision taking}} \quad \underbrace{\sum_{\substack{\mathbf{u} \in \mathcal{U}^k,\ \mathbf{x} \in \mathcal{X}^n \\ u_i \text{ fixed}}} \underbrace{P_{\mathbf{UXY}}(\mathbf{u}, \mathbf{x}, \mathbf{y}),}_{\text{Joint pmf/pdf}}}_{\substack{\text{Marginal function is approximated} \\ \text{by Sum-Product Algorithm}}}}_{\text{Decision about symbol } u_i \text{ based on symbol-wise decoding}}$$

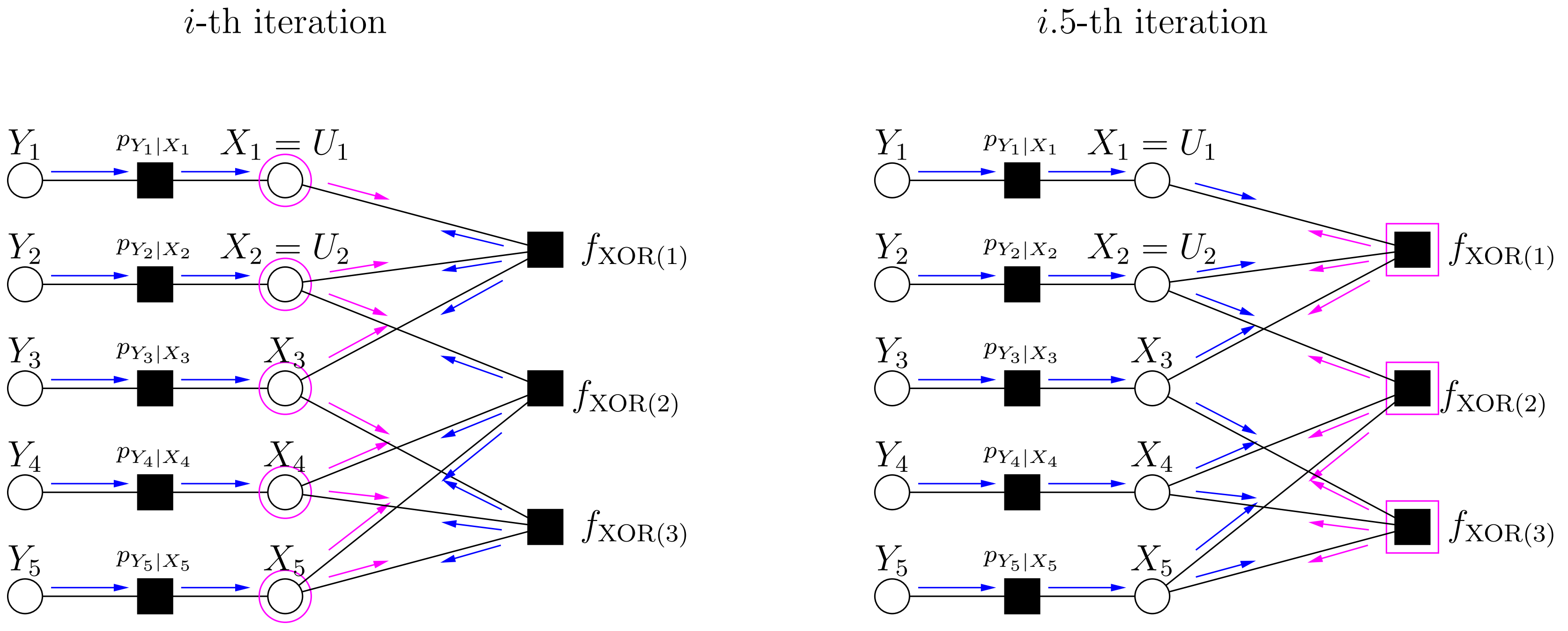
On factor graphs without cycles, the approximation is exact.

SPA Decoding (Factor graph with cycles)





SPA Decoding (Factor graph with cycles)

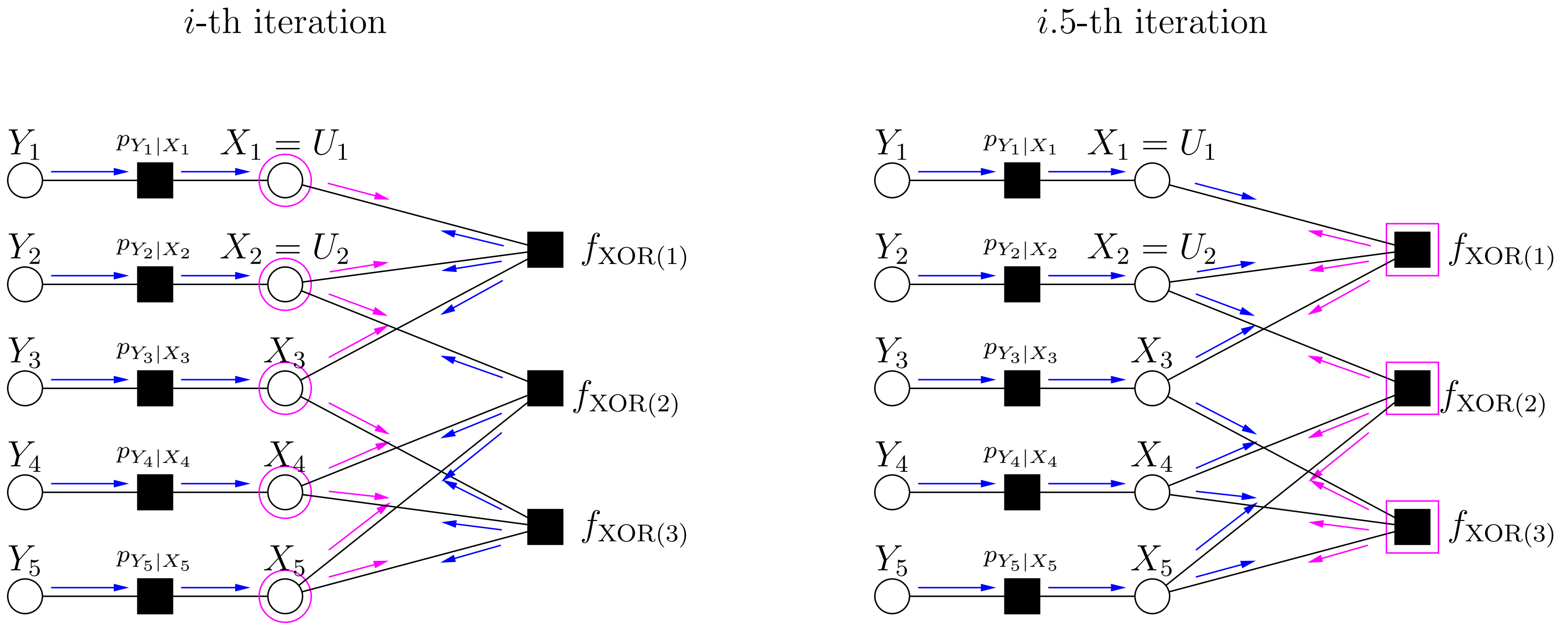

i-th iteration

i.5-th iteration

A message-passing algorithm

- sends messages along the edges,
- does processing of the messages at the vertices.

SPA Decoding (Factor graph with cycles)

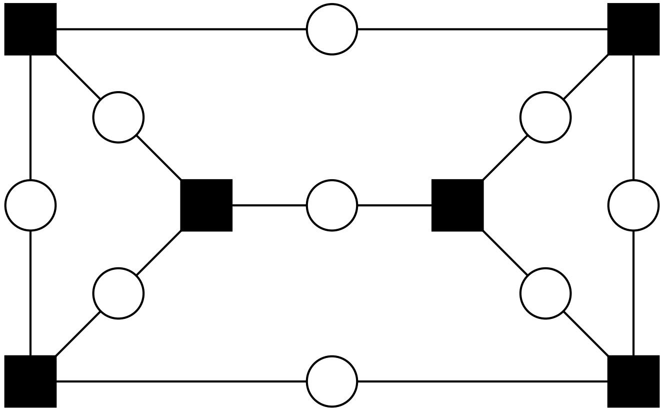


A message-passing algorithm

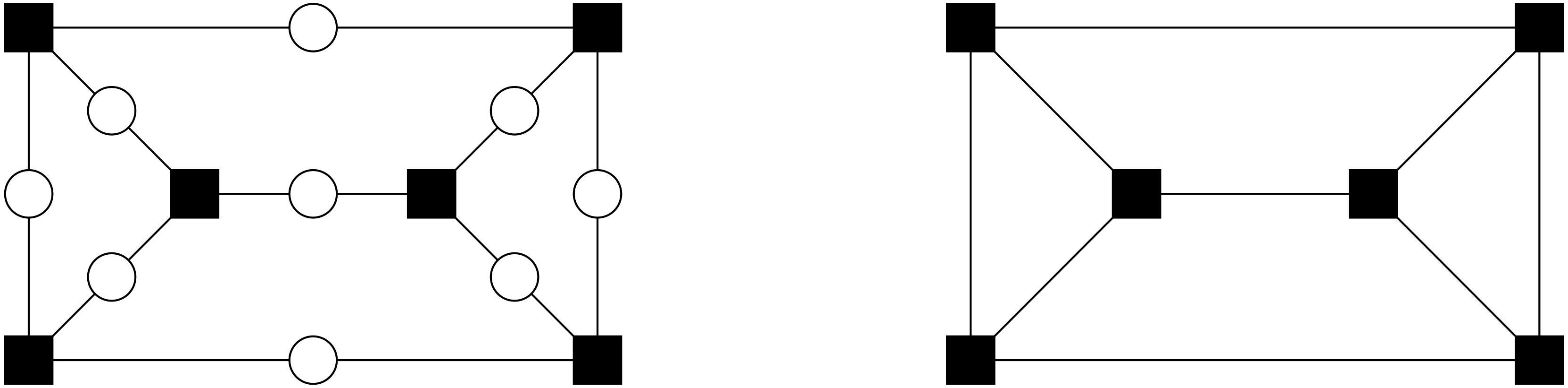
- sends messages along the edges,
- does processing of the messages at the vertices.

Note: all operations are performed locally!

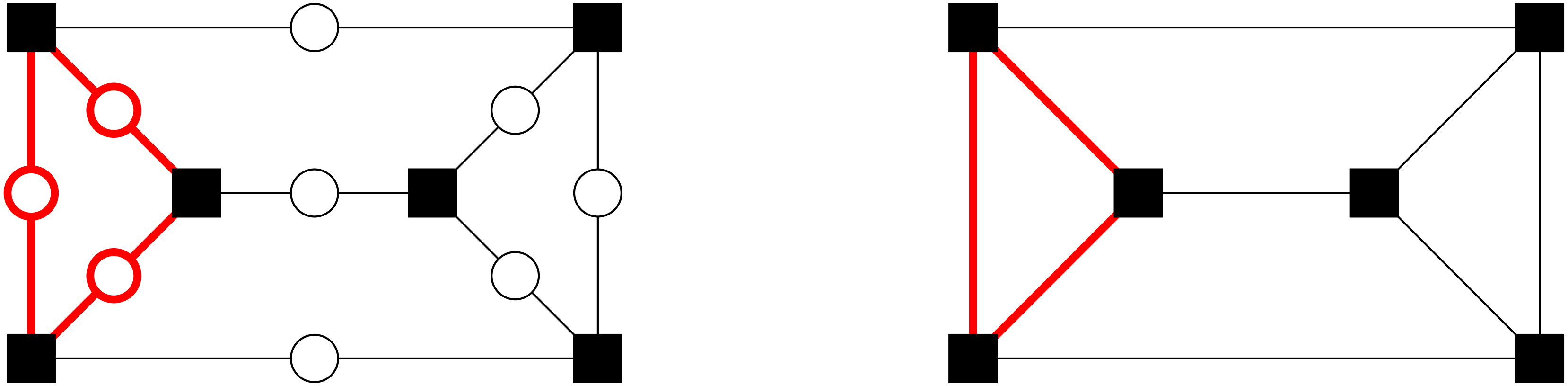
Factor Graph of a Cycle Code



Factor Graph of a Cycle Code

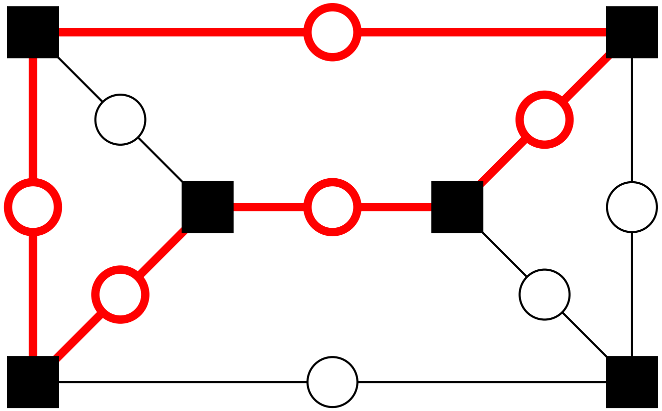


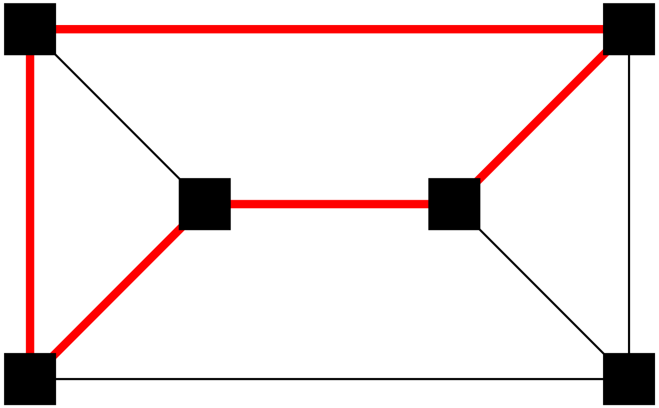
Factor Graph of a Cycle Code



Cycle codes are called cycle codes because codewords correspond to simple cycles (or to the symmetric difference set of simple cycles) in the Tanner/factor graph.

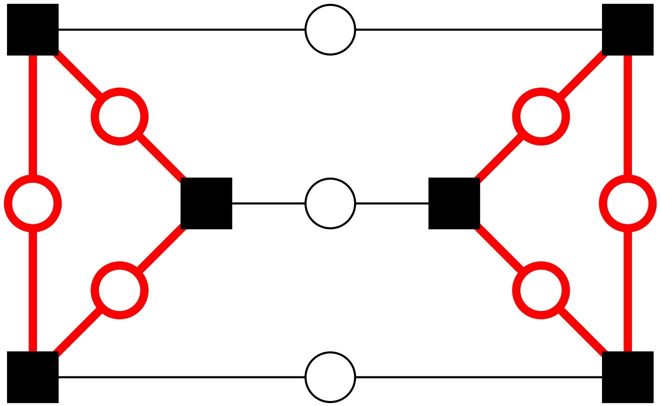
Factor Graph of a Cycle Code

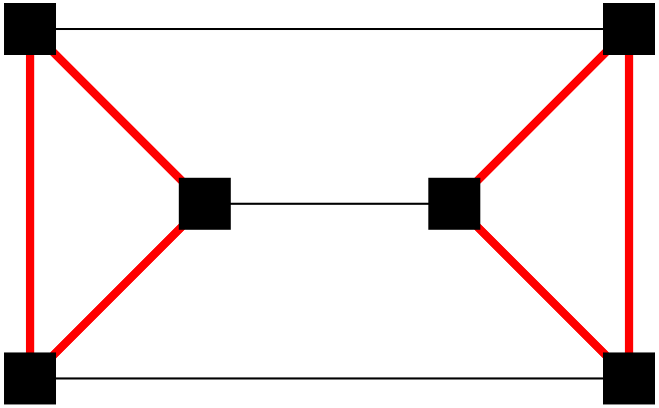




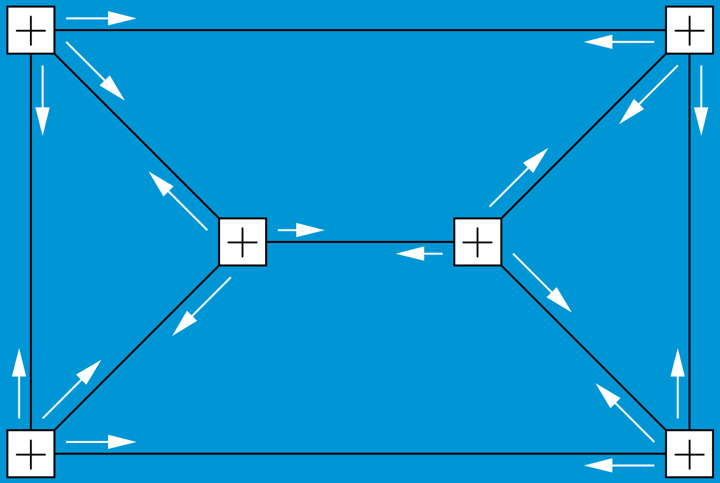
Cycle codes are called cycle codes because codewords correspond to simple cycles (or to the symmetric difference set of simple cycles) in the Tanner/factor graph.

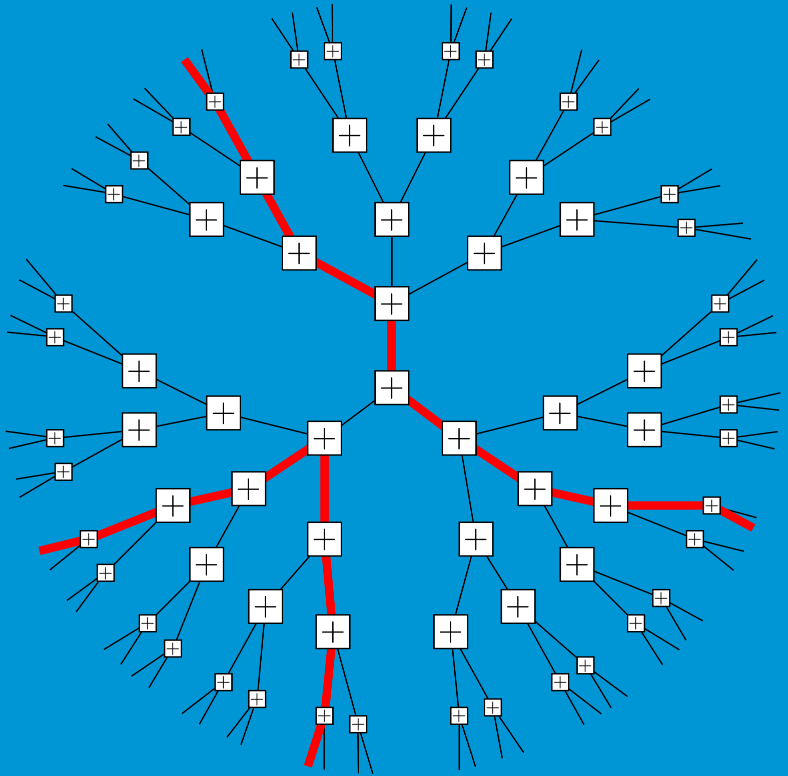
Factor Graph of a Cycle Code

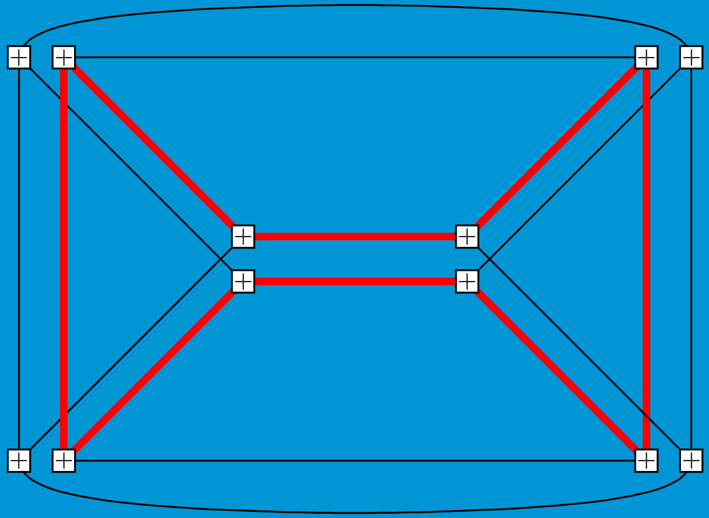




Cycle codes are called cycle codes because codewords correspond to simple cycles (or to the symmetric difference set of simple cycles) in the Tanner/factor graph.

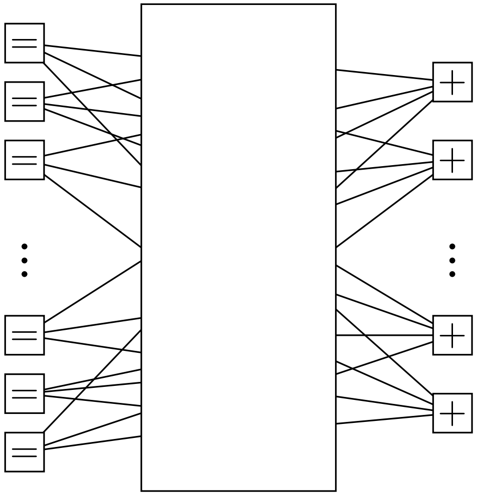




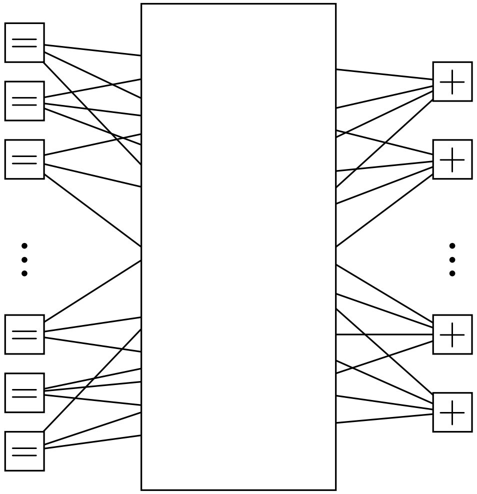


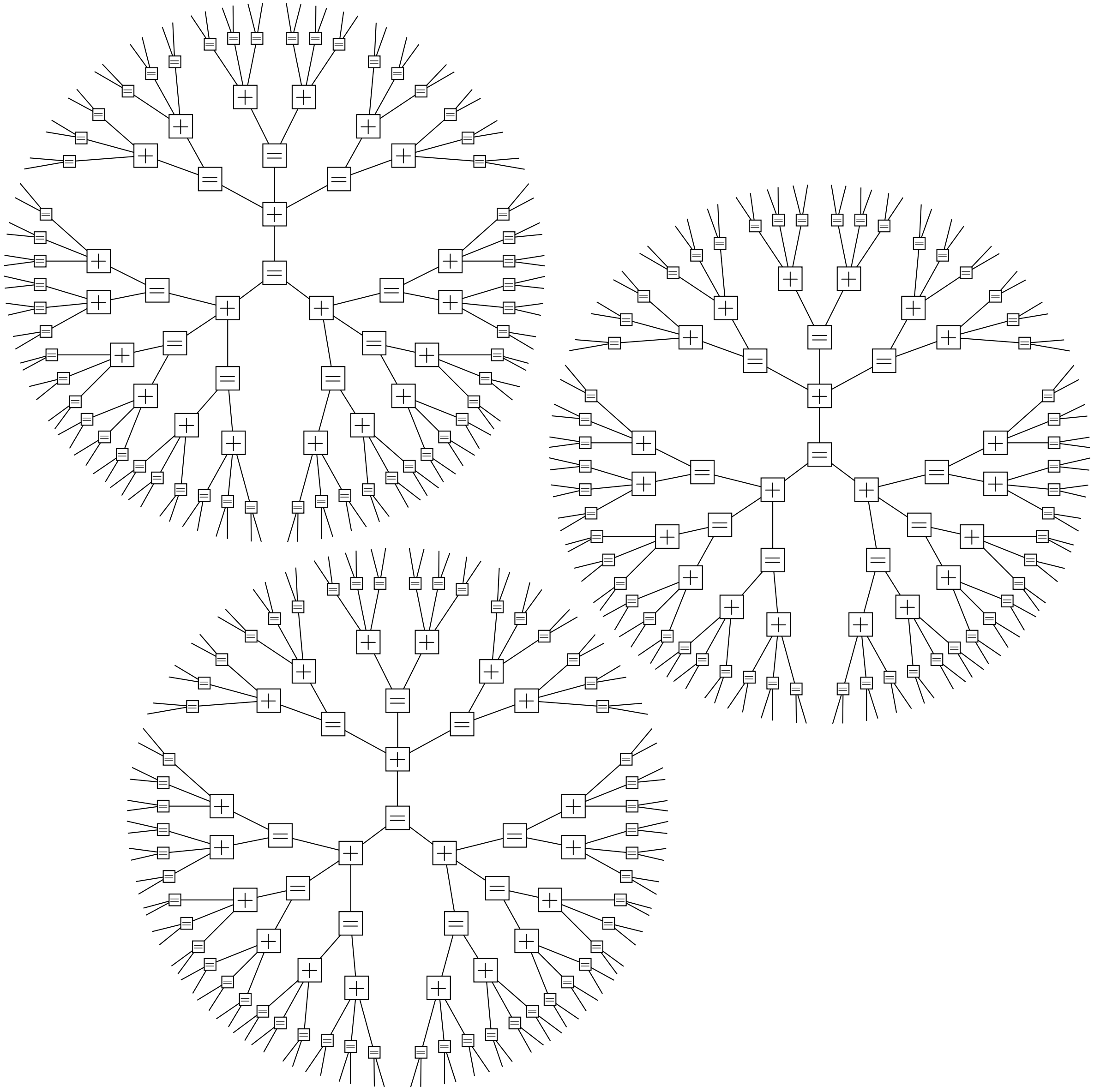
$$\zeta(\mathsf{V}_1,\ldots,\mathsf{V}_n) \;=\; \sum_{\mathbf{k}} \zeta_{\mathbf{k}} \mathbf{V}^{\mathbf{k}} \;=\; \prod_{[\Gamma]} \frac{1}{1-g(\Gamma,\mathbf{V})}$$

Computation trees

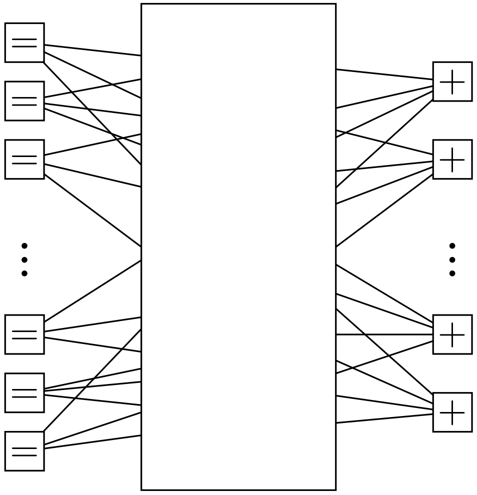


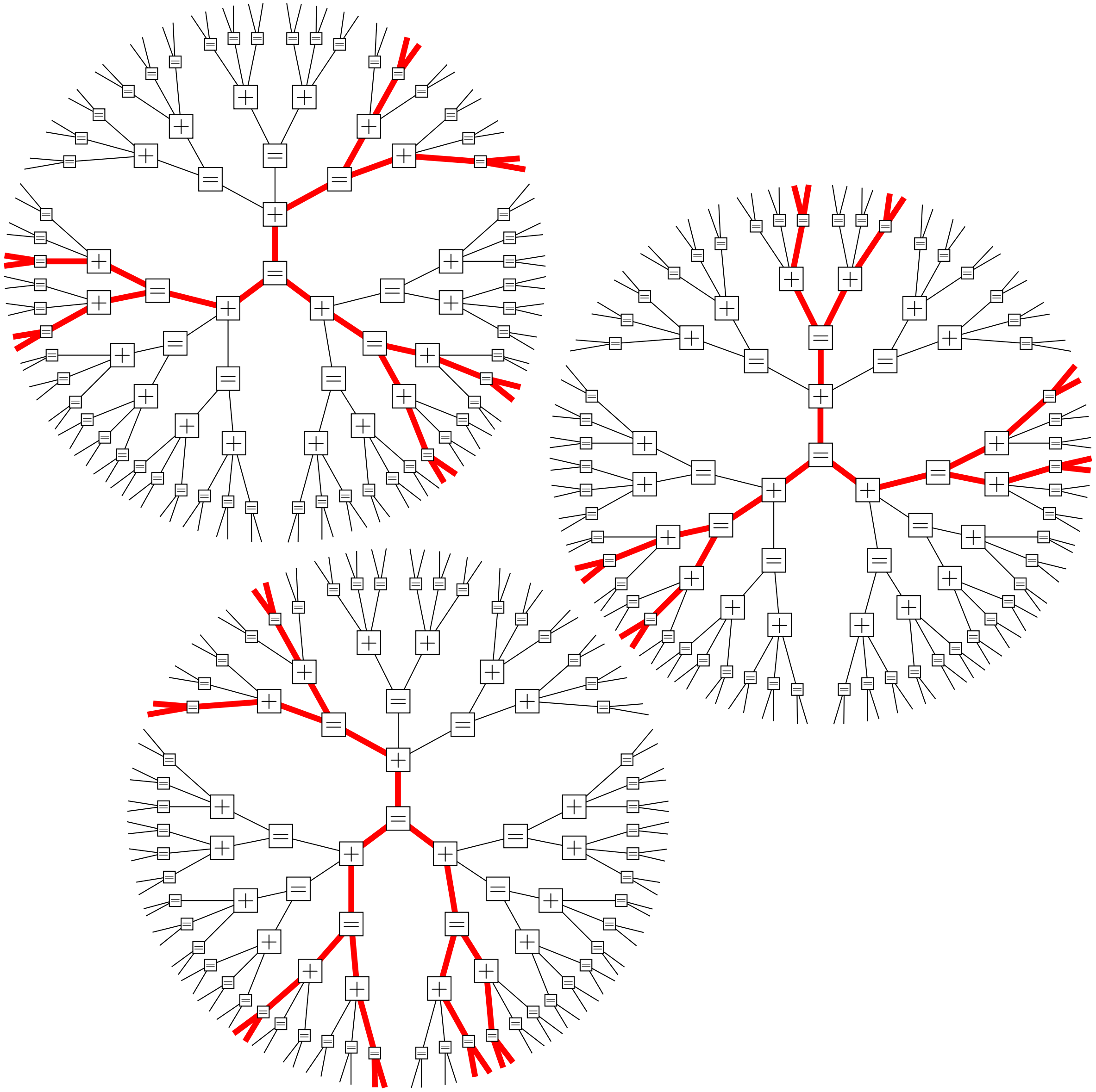
Computation trees

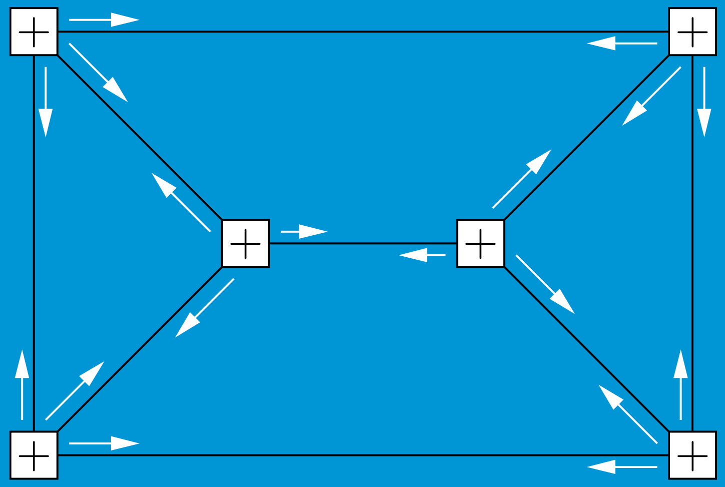


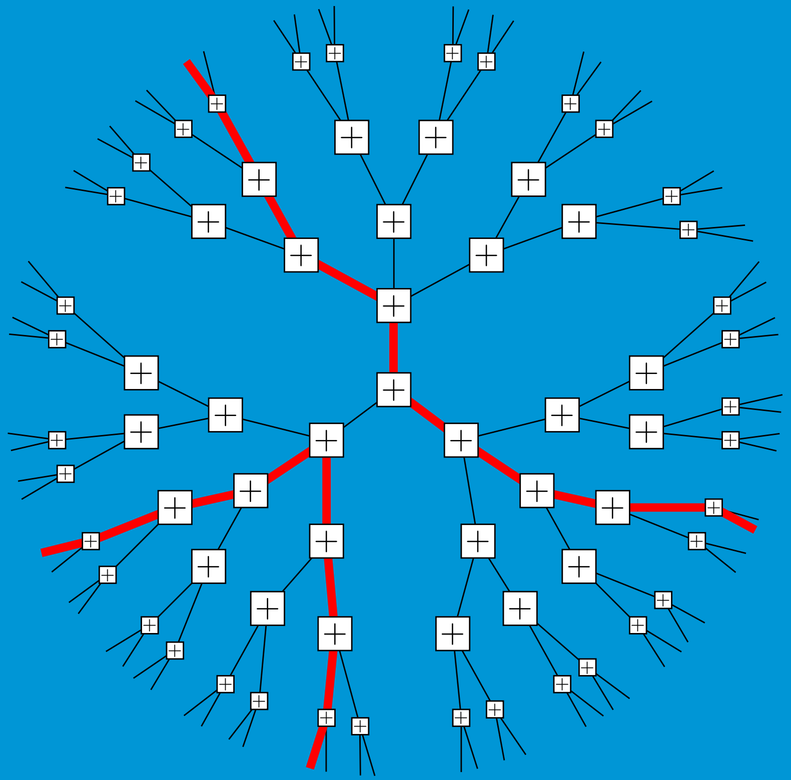


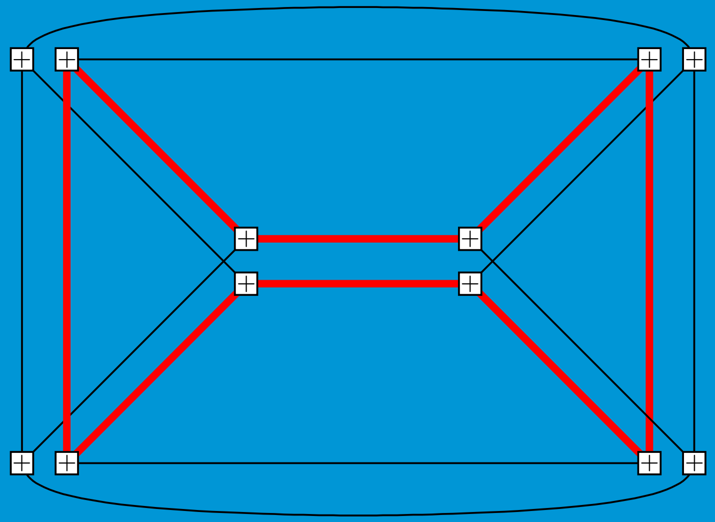
Computation trees





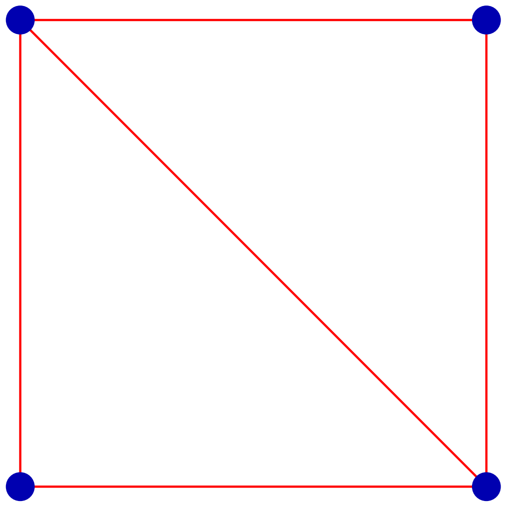






$$\zeta(\mathsf{V}_1,\ldots,\mathsf{V}_n) \;=\; \sum_{\mathbf{k}} \zeta_{\mathbf{k}}\mathbf{V}^{\mathbf{k}} \;=\; \prod_{[\Gamma]} \frac{1}{1-g(\Gamma,\mathbf{V})}$$

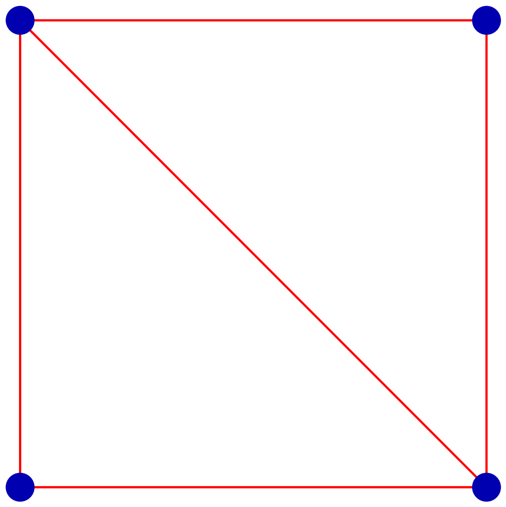
Finite Graph Covers



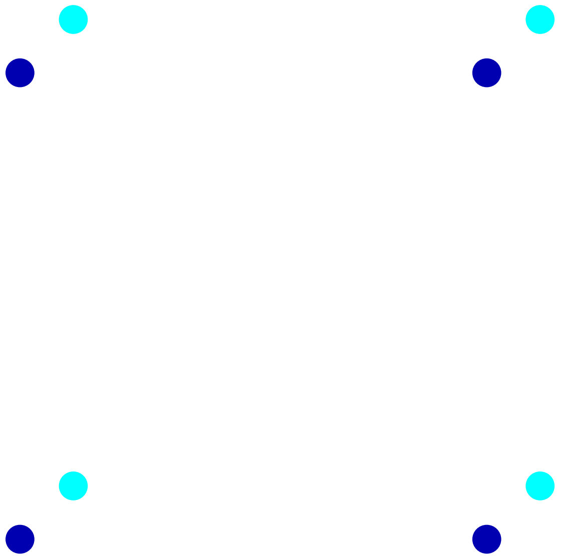
original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



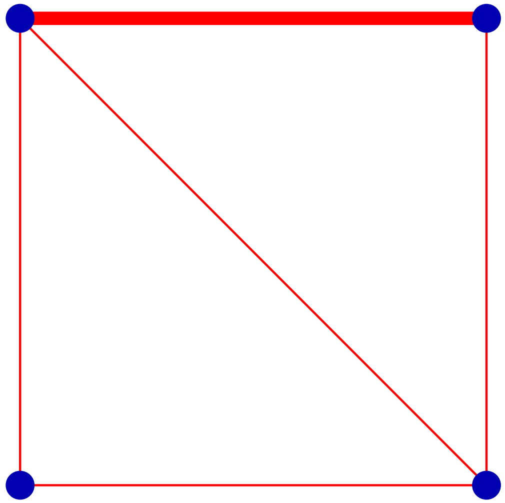
original graph



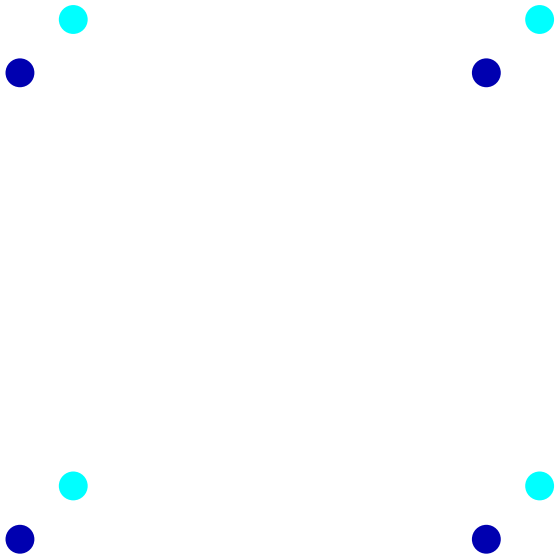
2-fold cover of original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



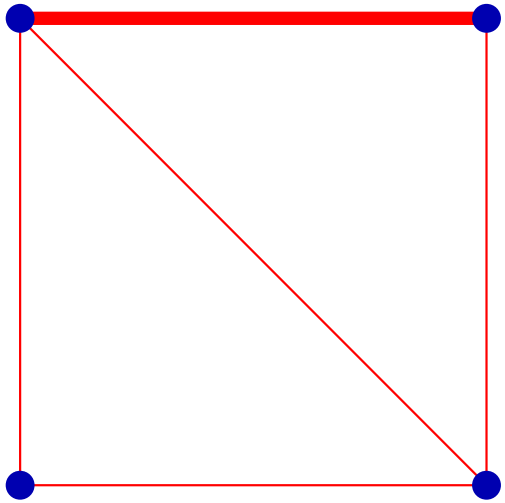
original graph



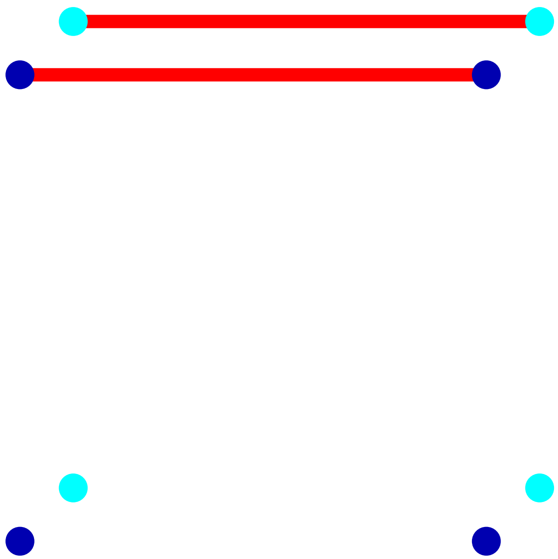
2-fold cover of original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



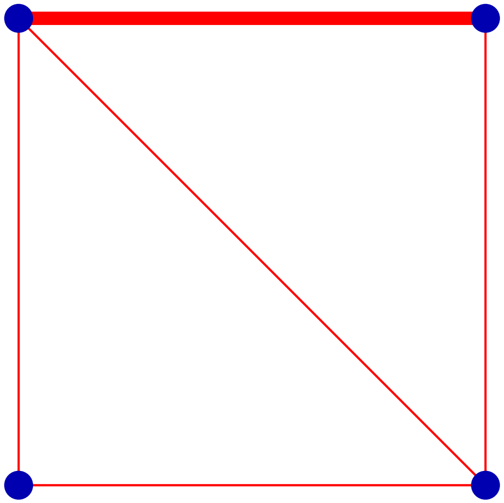
original graph



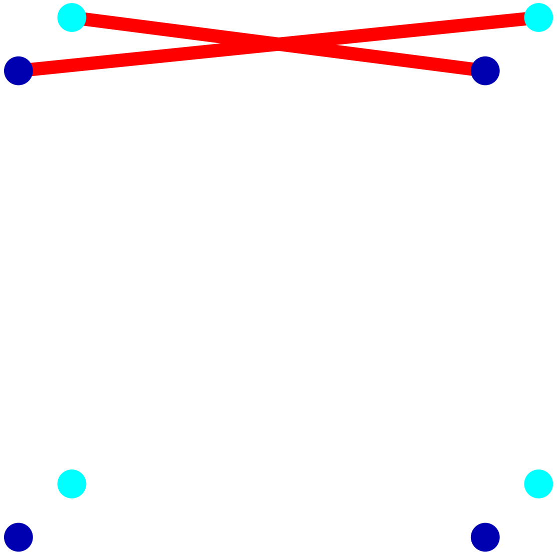
2-fold cover of original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



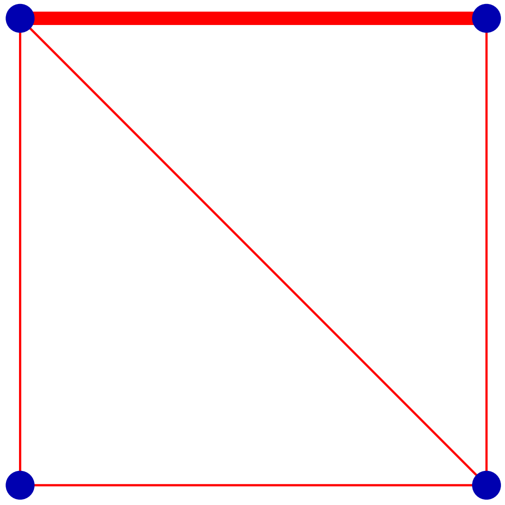
original graph



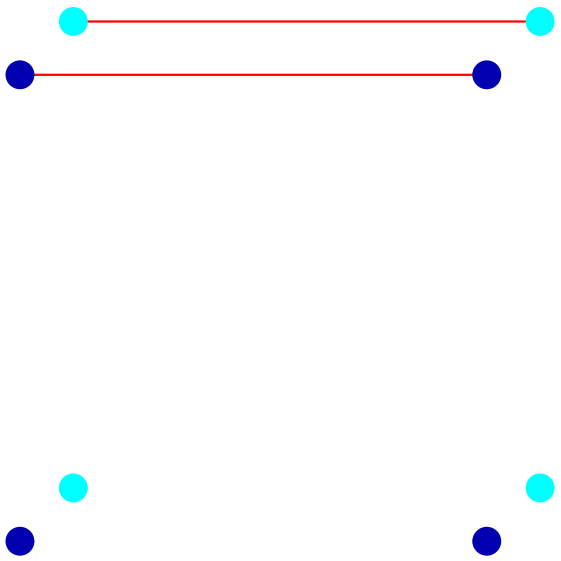
2-fold cover of
original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



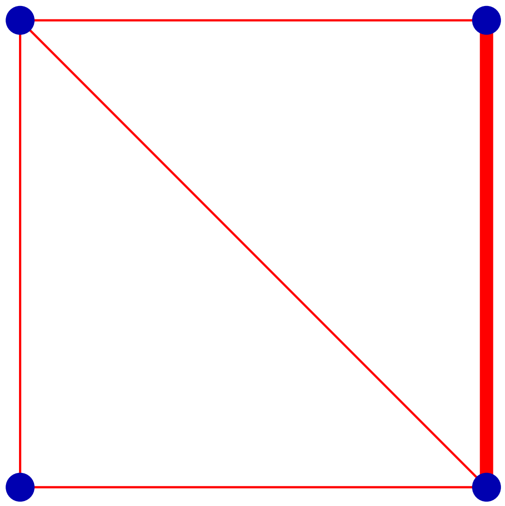
original graph



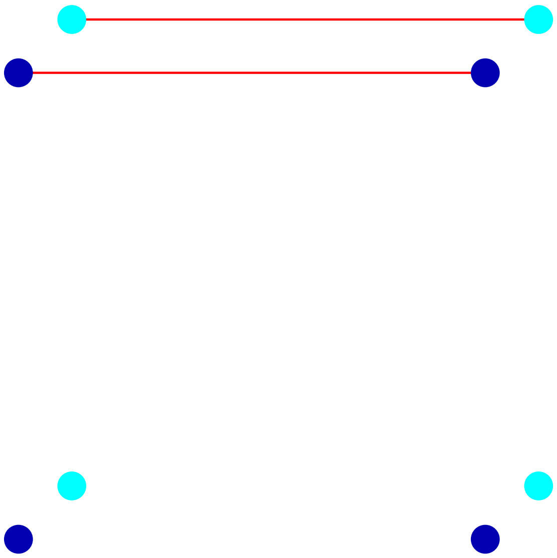
2-fold cover of original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



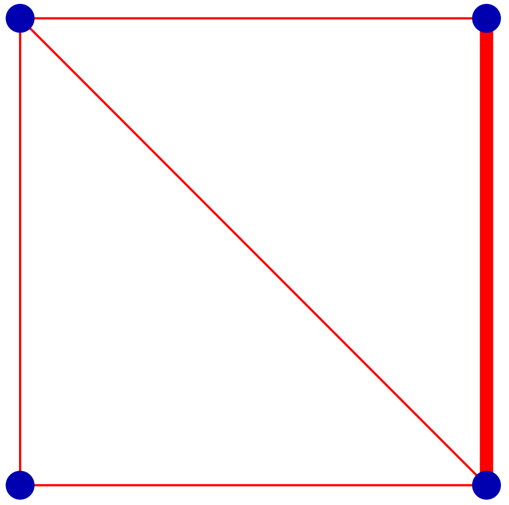
original graph



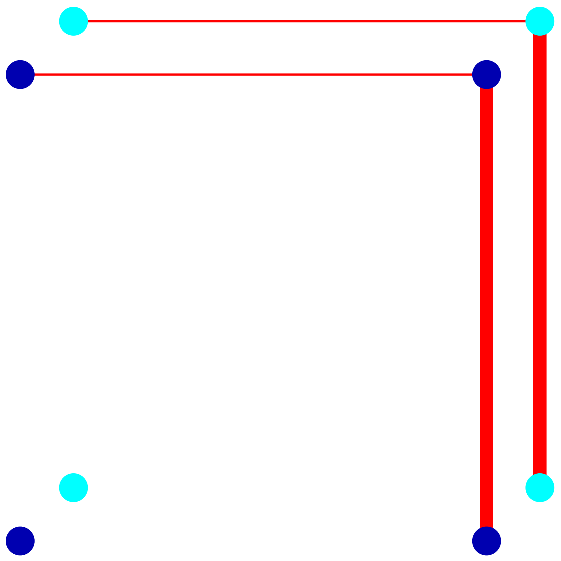
2-fold cover of original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



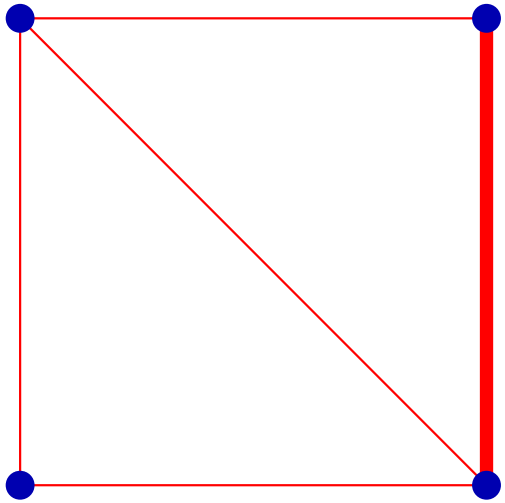
original graph



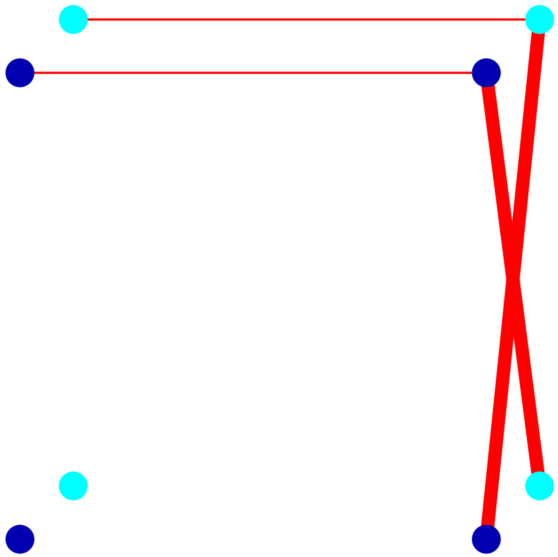
2-fold cover of original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



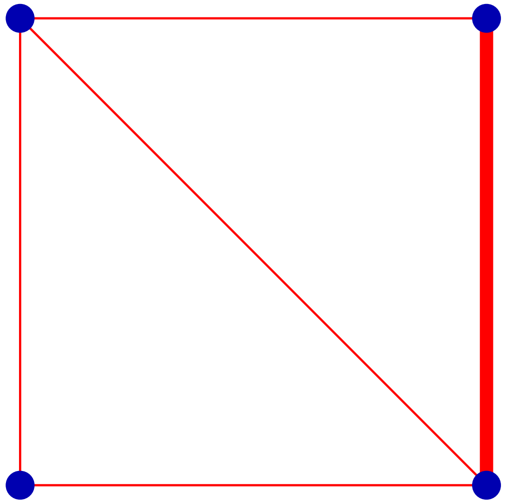
original graph



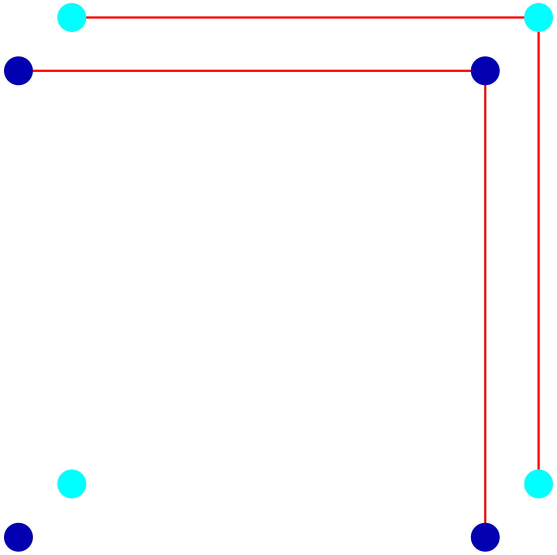
2-fold cover of
original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



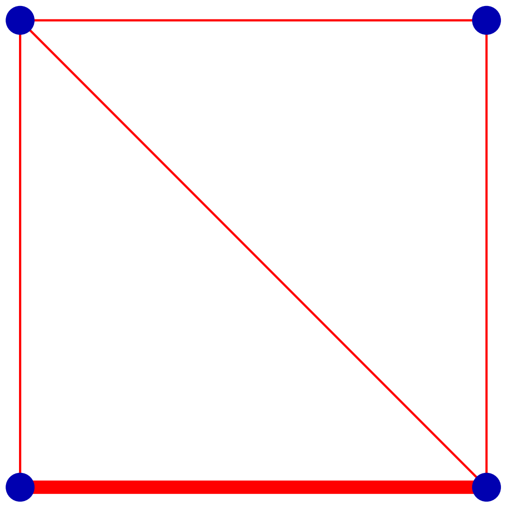
original graph



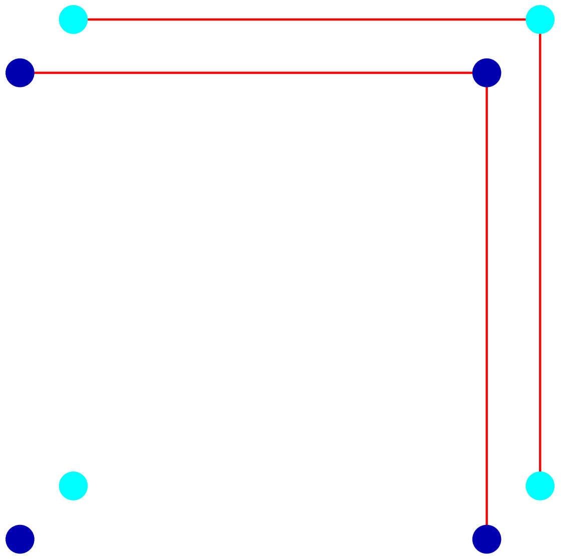
2-fold cover of
original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



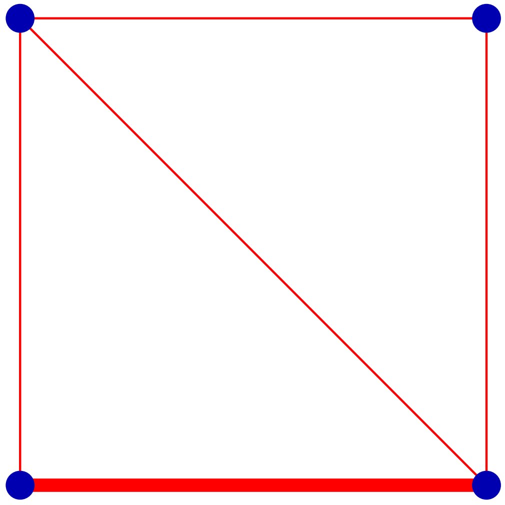
original graph



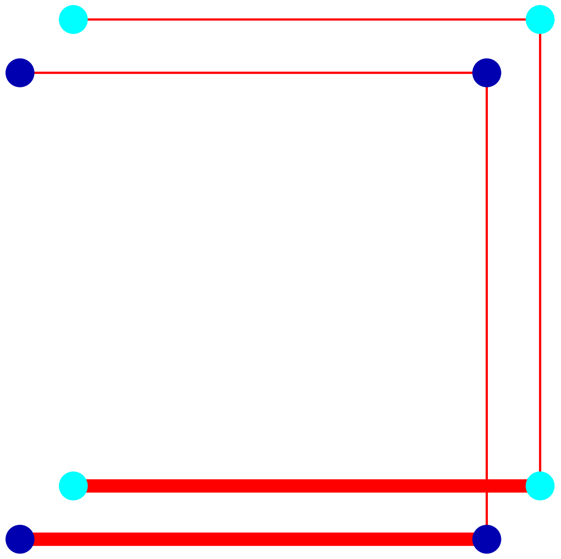
2-fold cover of original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



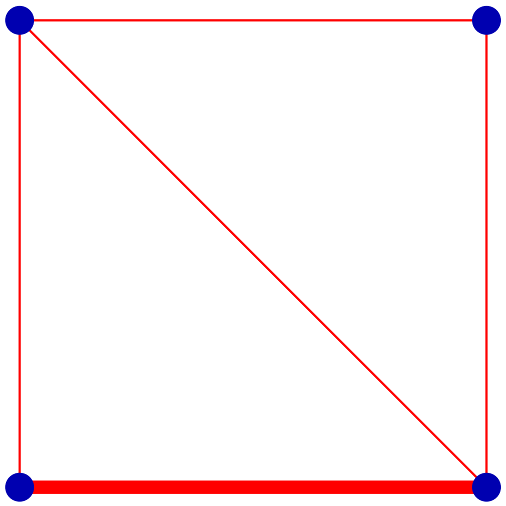
original graph



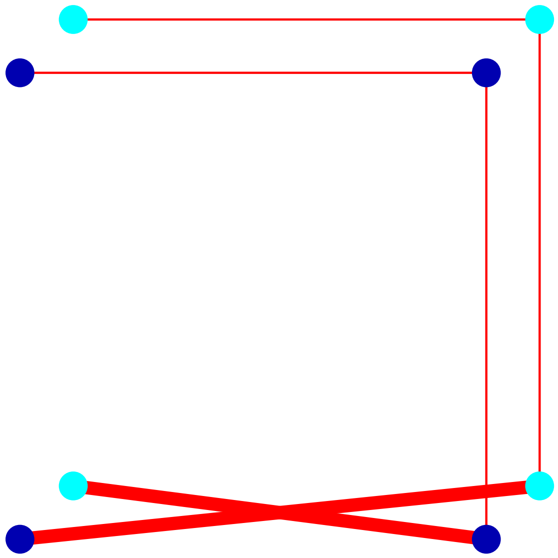
2-fold cover of
original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



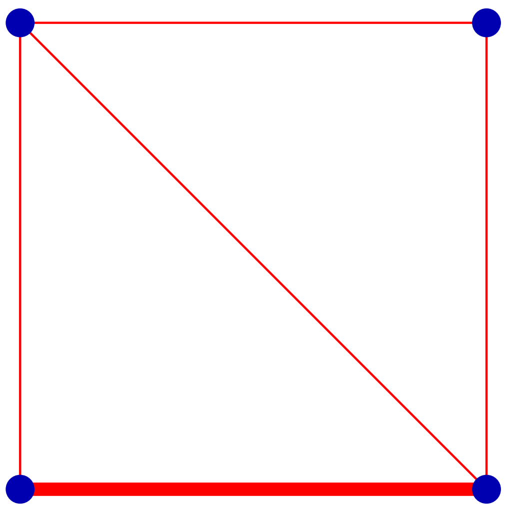
original graph



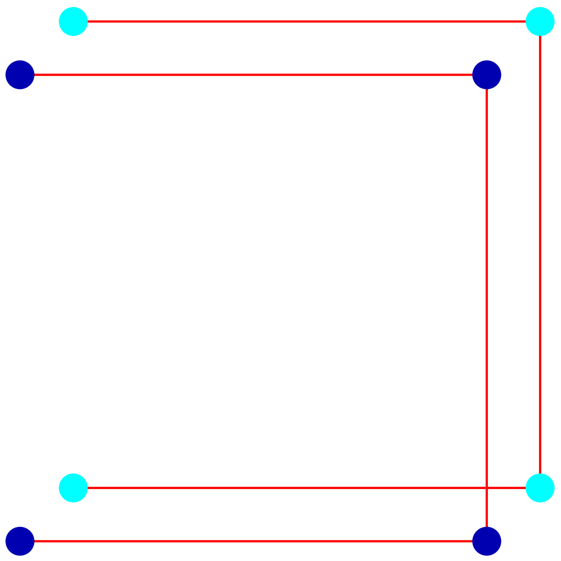
2-fold cover of original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



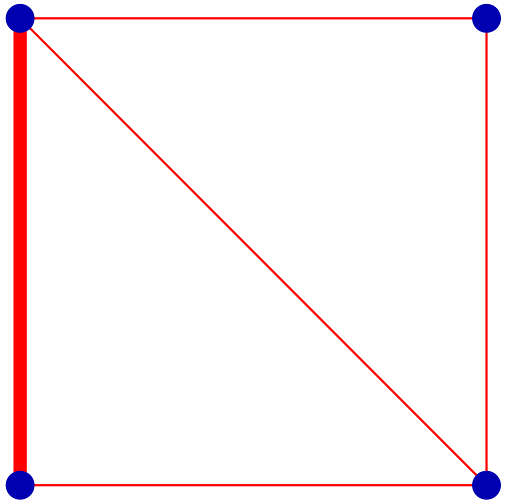
original graph



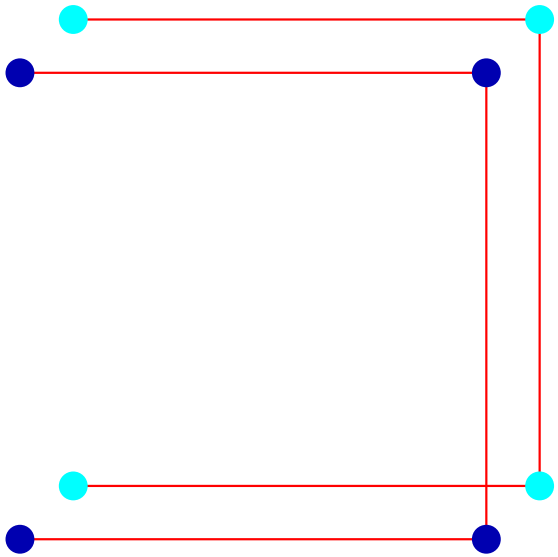
2-fold cover of original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



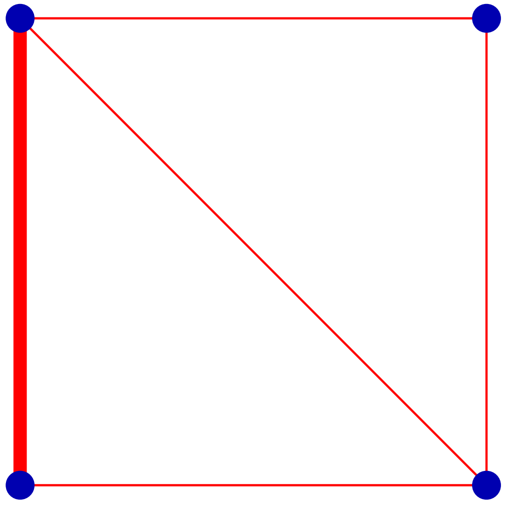
original graph



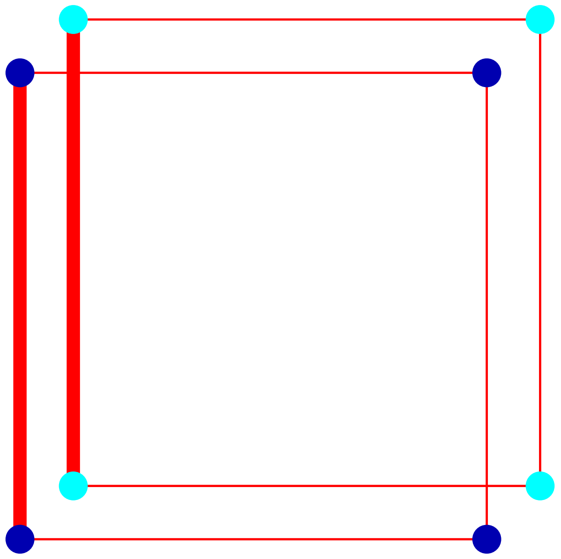
2-fold cover of original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



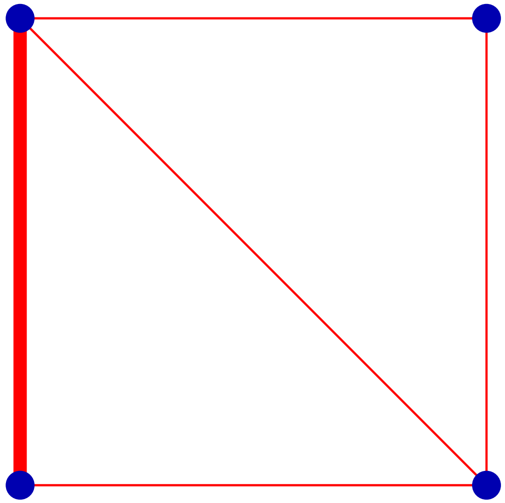
original graph



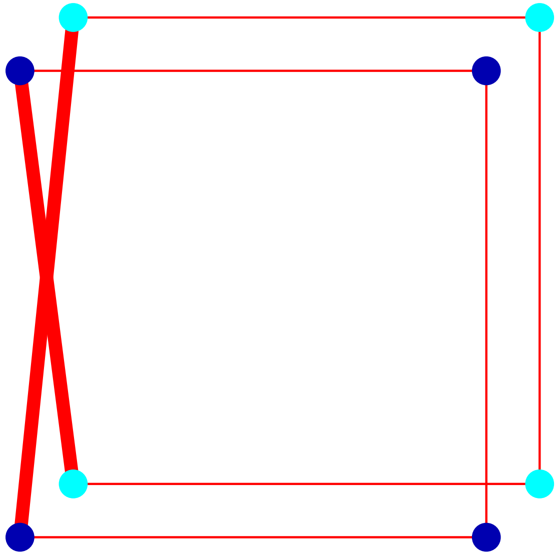
2-fold cover of
original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



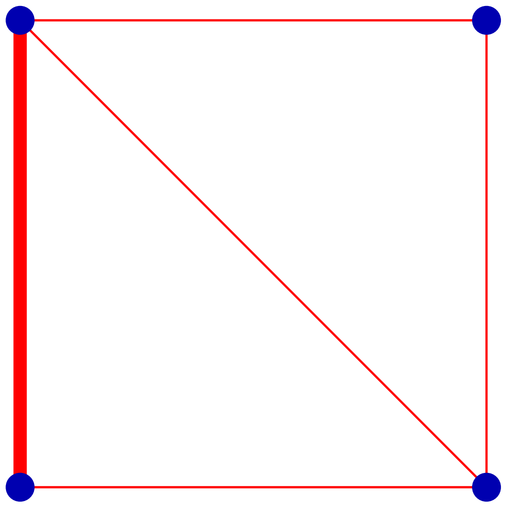
original graph



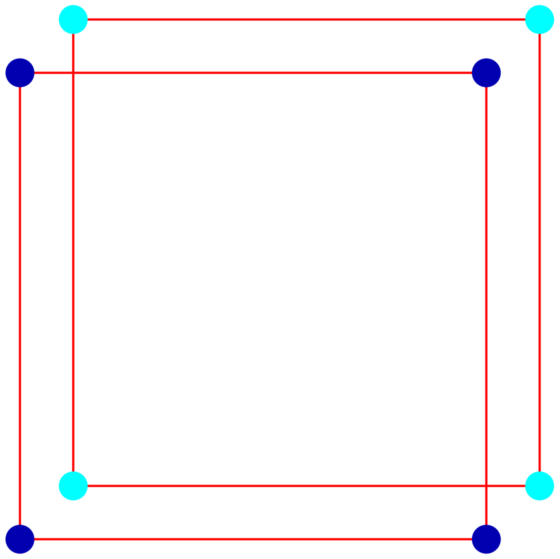
2-fold cover of
original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



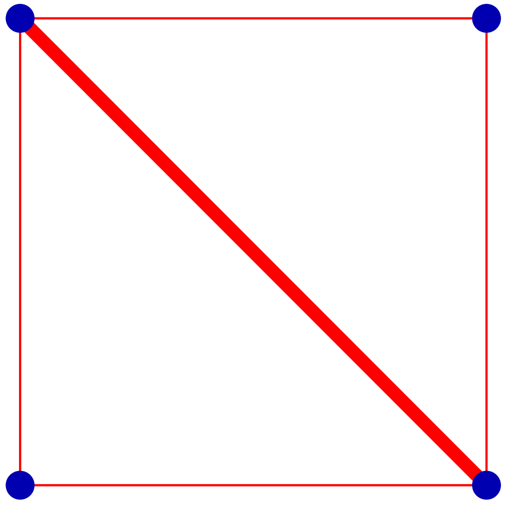
original graph



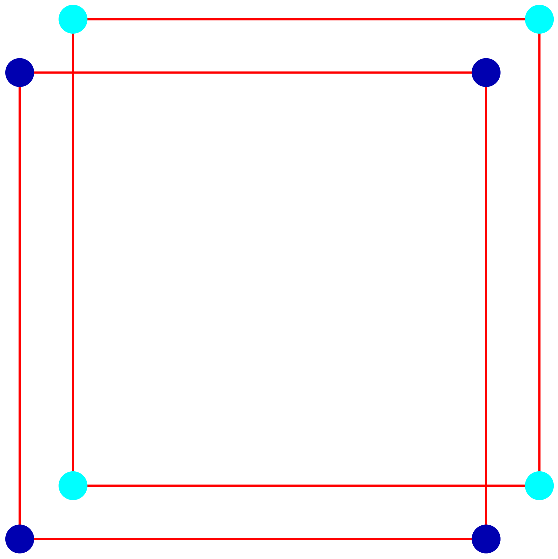
2-fold cover of original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



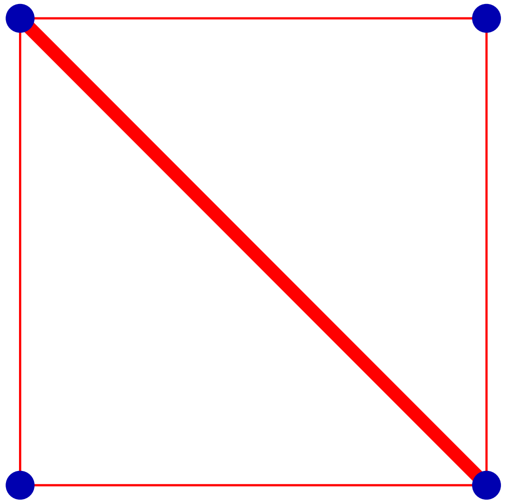
original graph



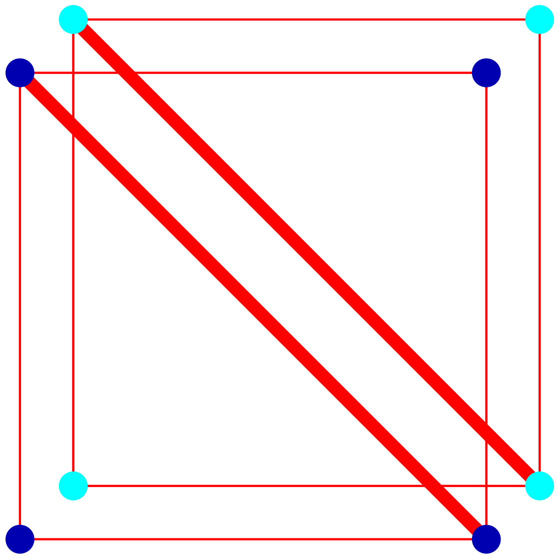
2-fold cover of original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



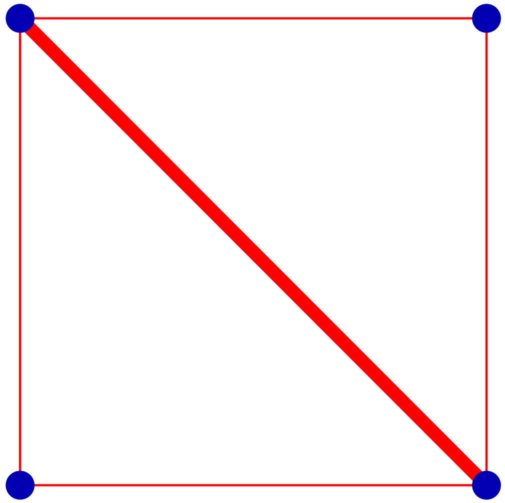
original graph



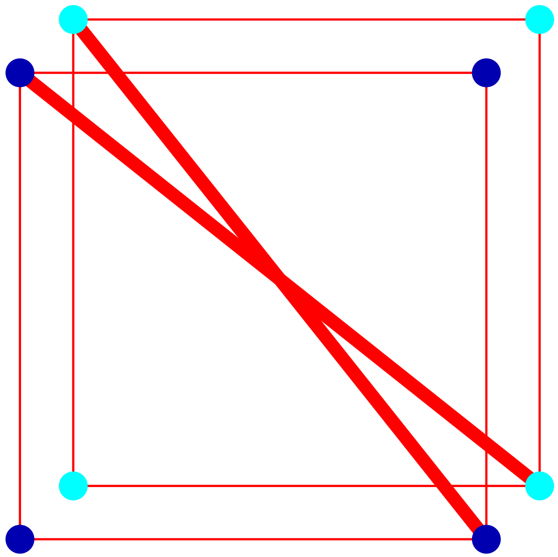
2-fold cover of original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



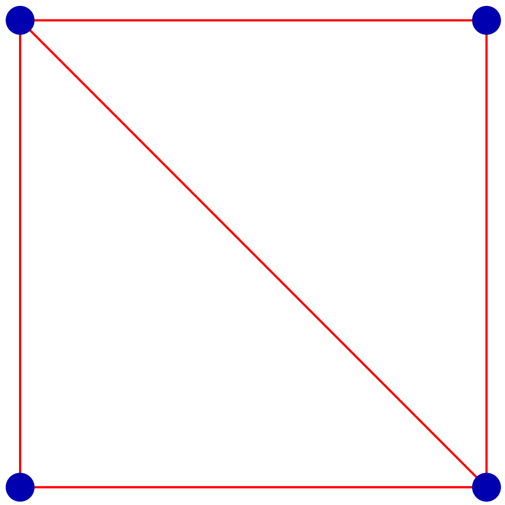
original graph



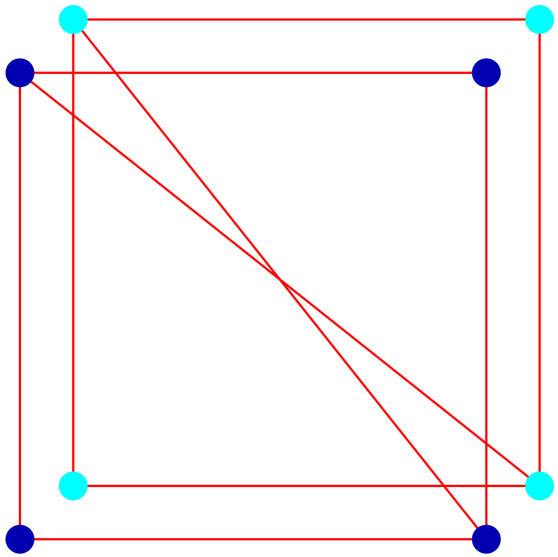
2-fold cover of original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



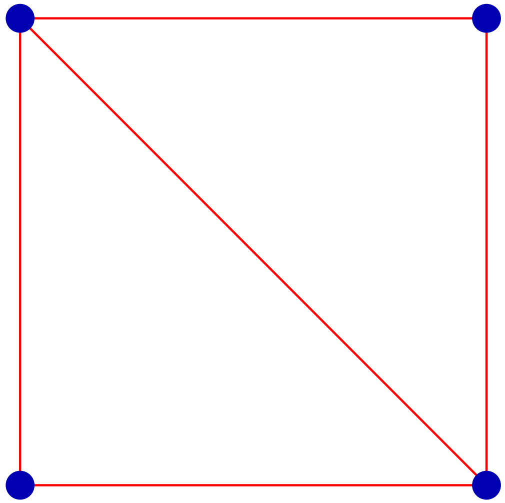
original graph



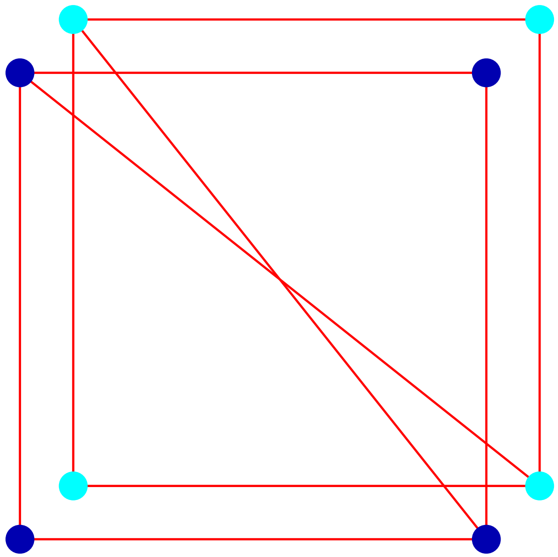
2-fold cover of original graph

Definition: A double cover of a graph is . . .

Finite Graph Covers



original graph

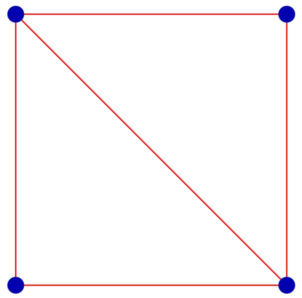


2-fold cover of original graph

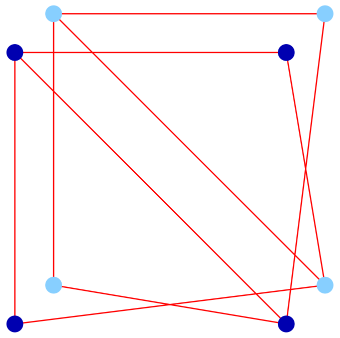
Definition: A double cover of a graph is . . .

Note: the above graph has $2! \cdot 2! \cdot 2! \cdot 2! \cdot 2! = (2!)^5$ double covers.

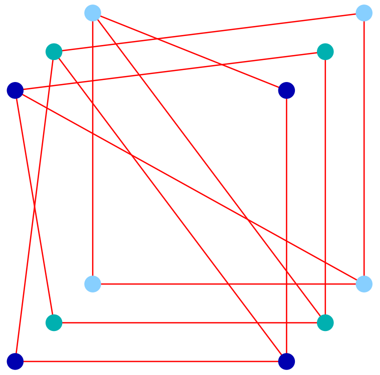
Graph Covers



original graph



(a possible)
double cover of
the original graph

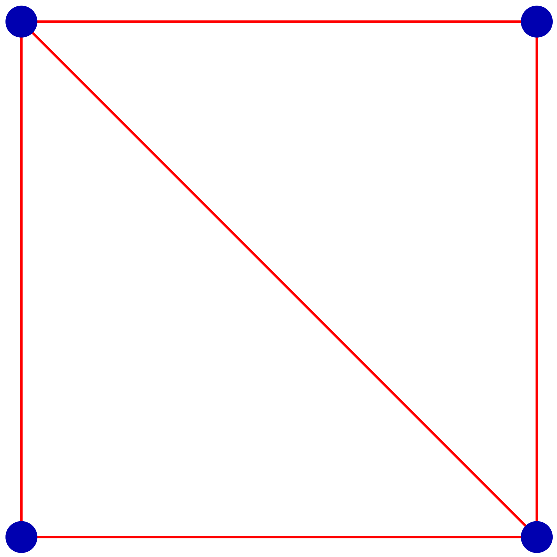


(a possible)
triple cover of
the original graph

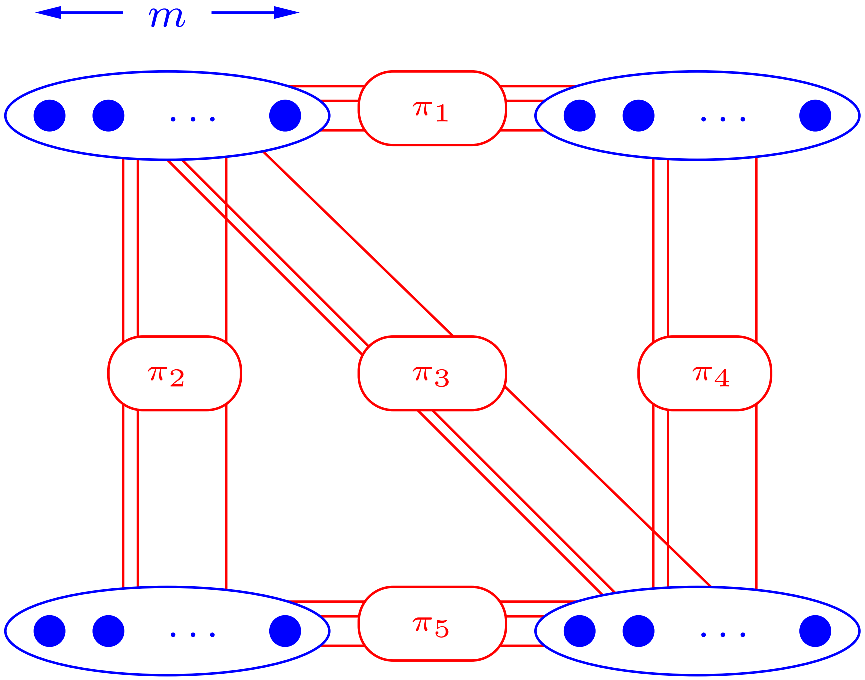
. . .

Besides double covers, a graph also has many triple covers, quadruple covers, quintuple covers, etc.

Graph Covers



original graph



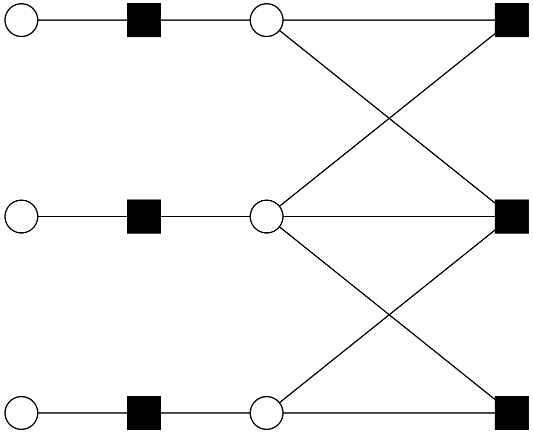
(possible)
m-fold cover of
original graph

An m-fold cover is also called a cover of degree m.

Do not confuse this degree with the degree of a vertex!

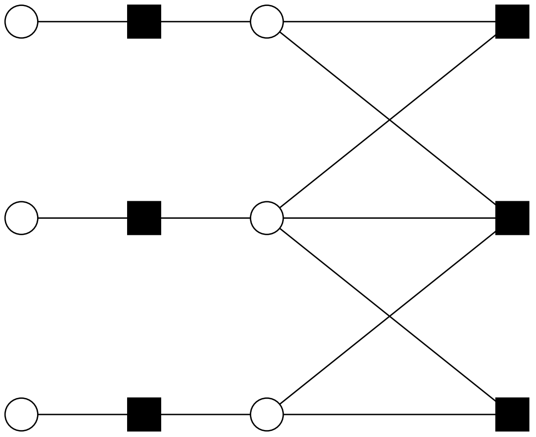
Graph Covers and the Sum-Product Algorithm

Consider this factor graph:

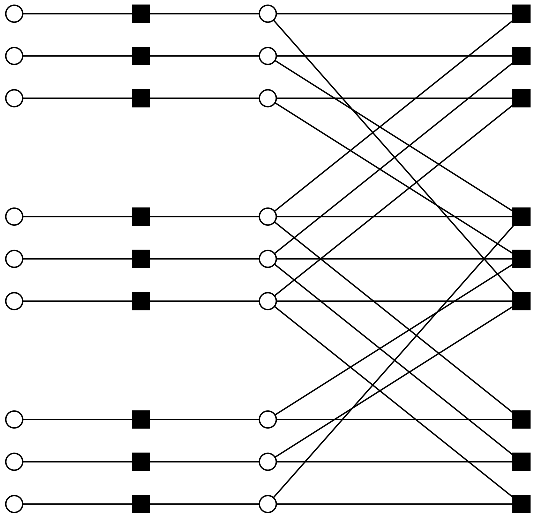


Graph Covers and the Sum-Product Algorithm

Consider this factor graph:

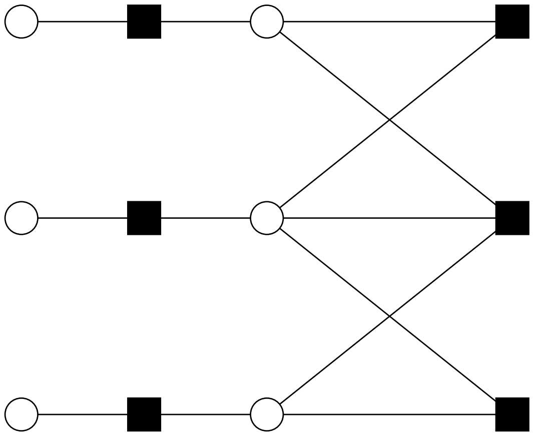


Here is a so-called triple cover of the above factor graph:

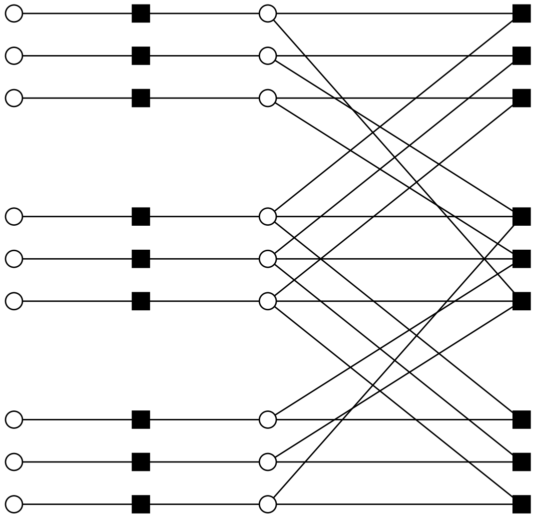


Graph Covers and the Sum-Product Algorithm

Consider this factor graph:



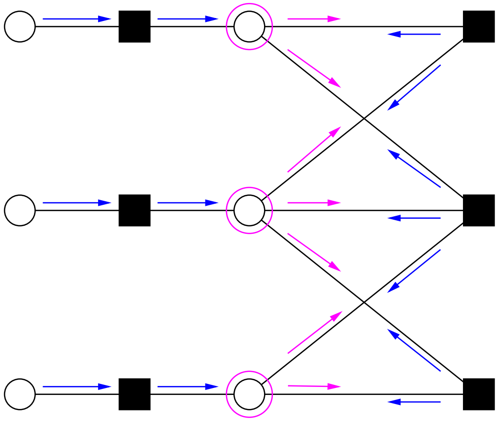
Here is a so-called triple cover of the above factor graph:



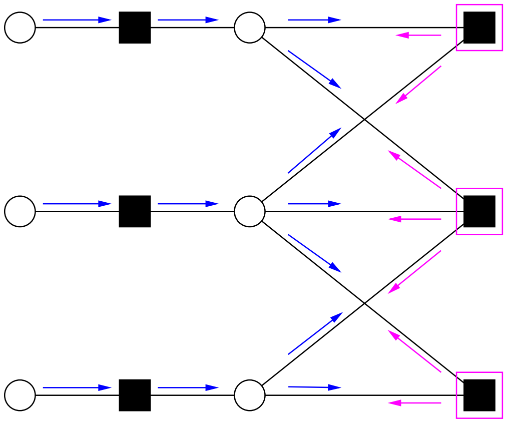
Why do factor graph covers matter?

Graph Covers and the Sum-Product Algorithm

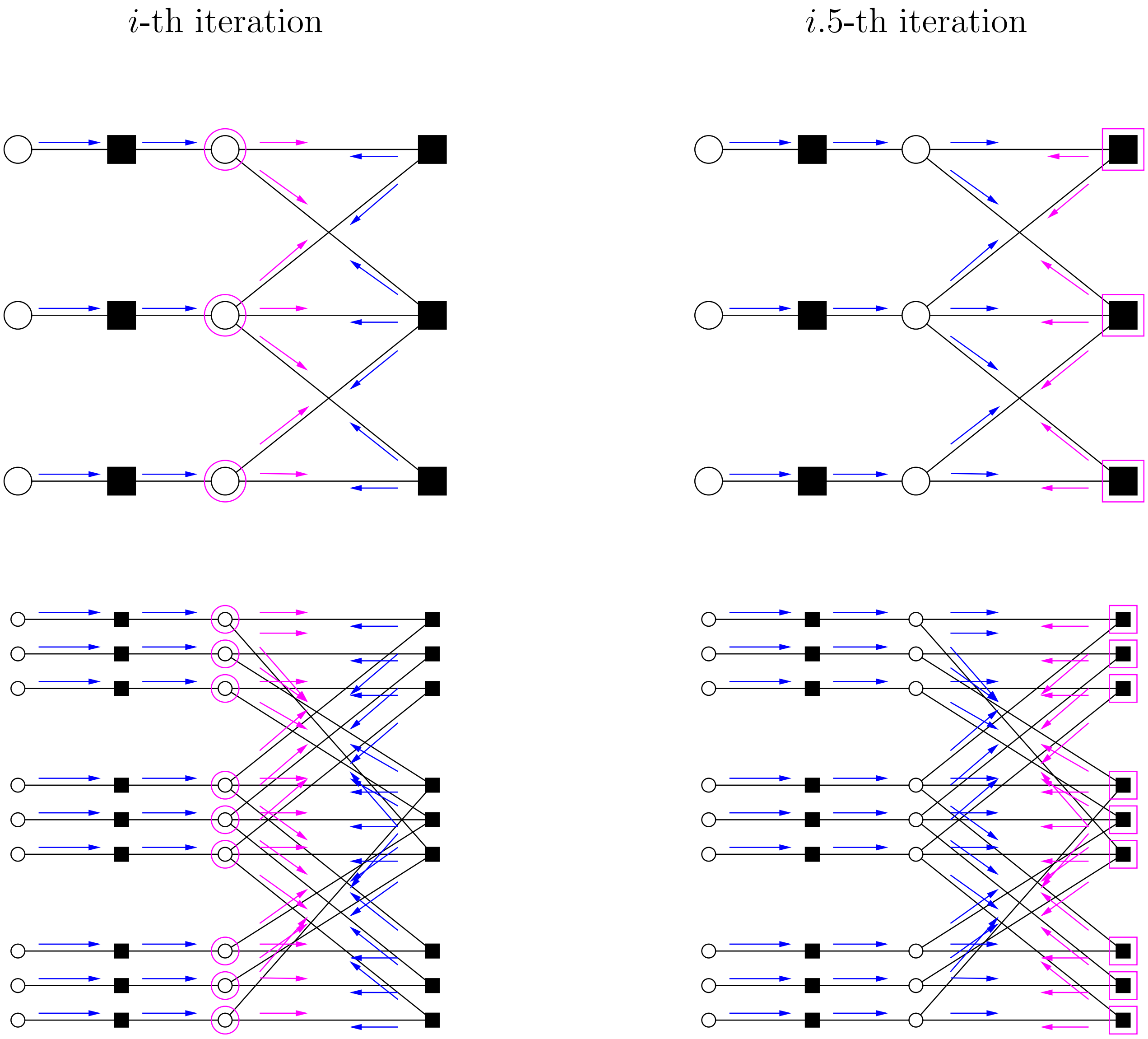
i-th iteration



$i.5$-th iteration



Graph Covers and the Sum-Product Algorithm

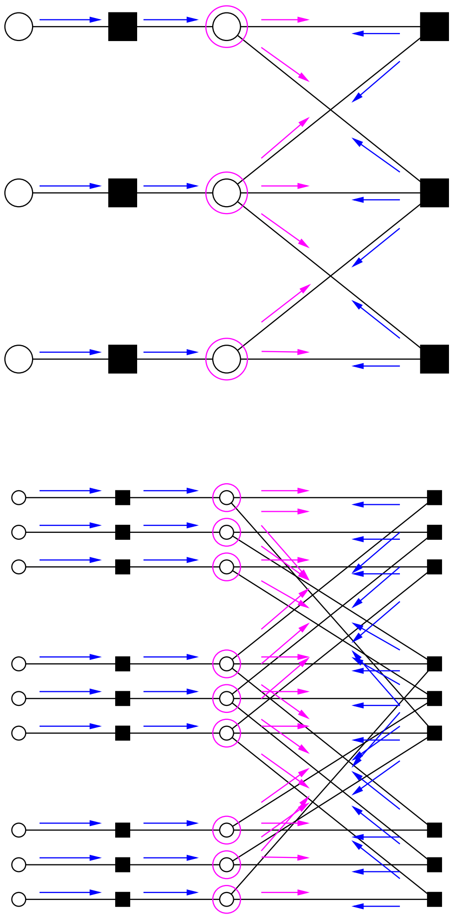


i-th iteration

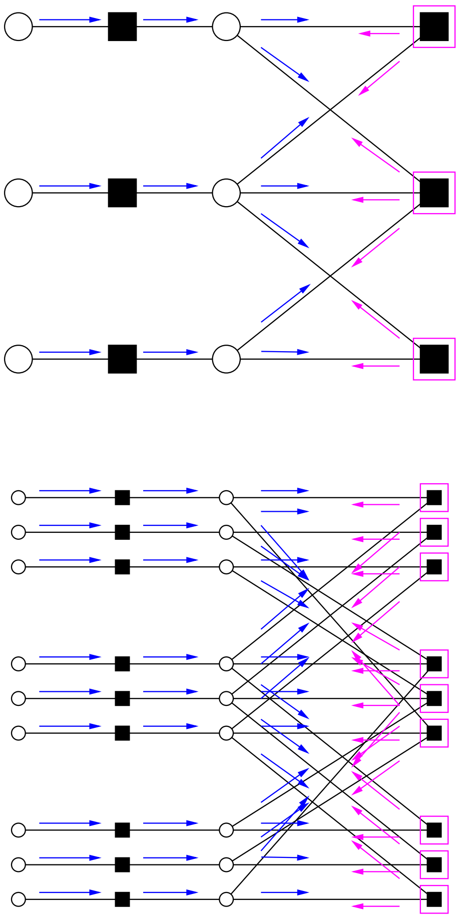
$i.5$-th iteration

Graph Covers and the Sum-Product Algorithm

i-th iteration

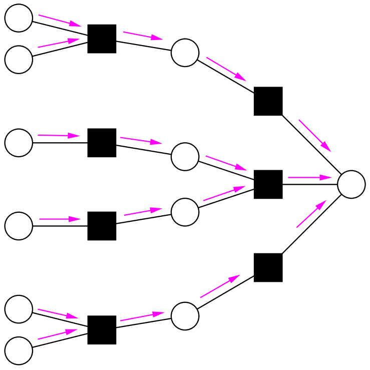


$i.5$-th iteration



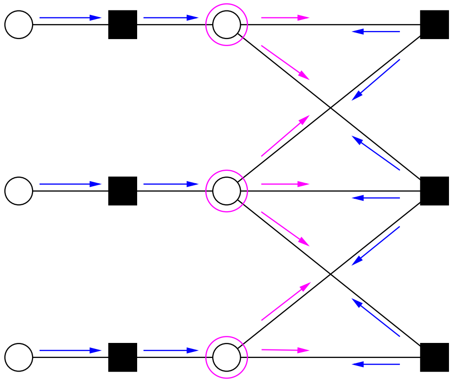
computation tree (without channel function nodes) . . .

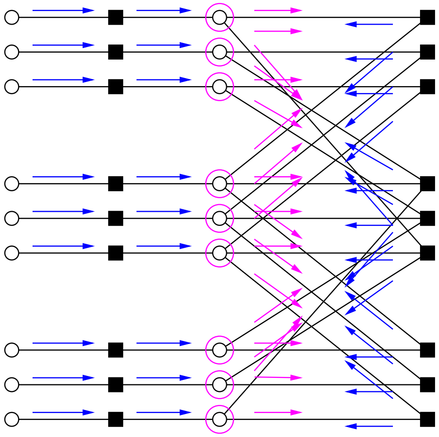
. . . where root is bit node 2



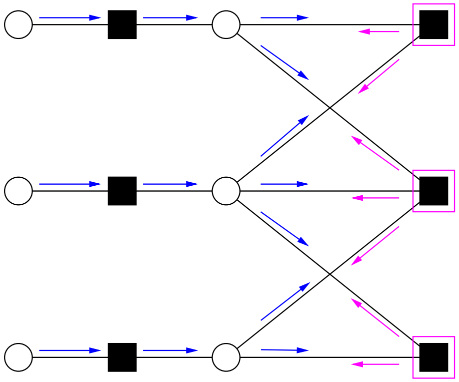
Graph Covers and the Sum-Product Algorithm

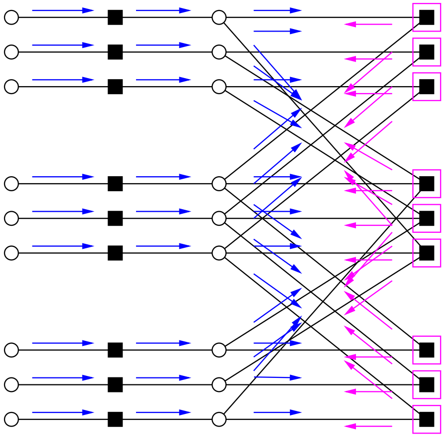
i-th iteration





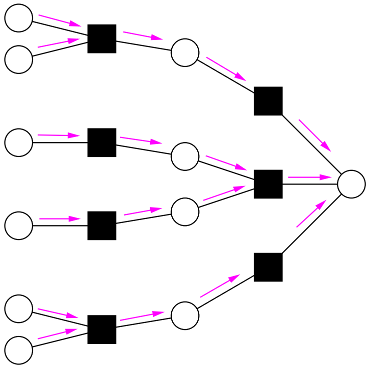
$i.5$-th iteration



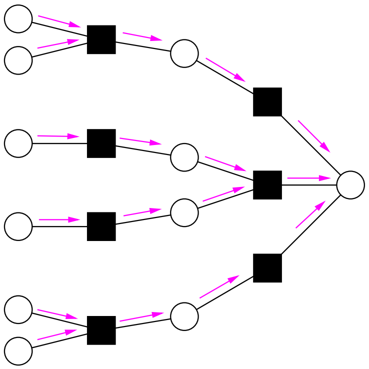


computation tree (without channel function nodes) . . .

. . . where root is bit node 2

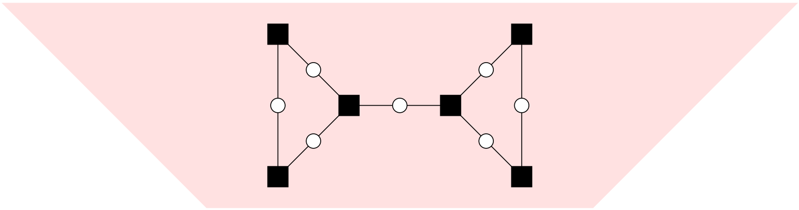


. . . where root is a copy of bit node 2

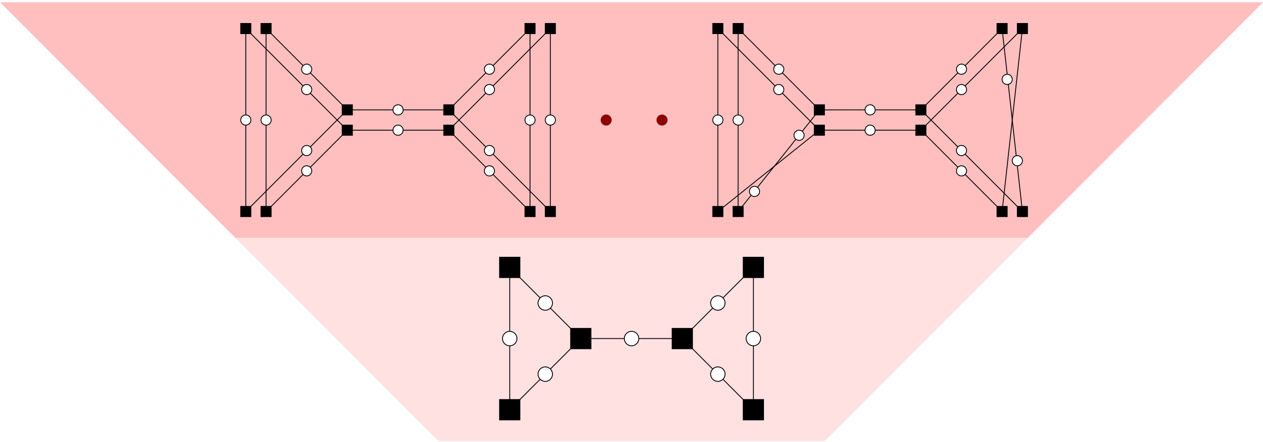


Graph Cover Hierarchy

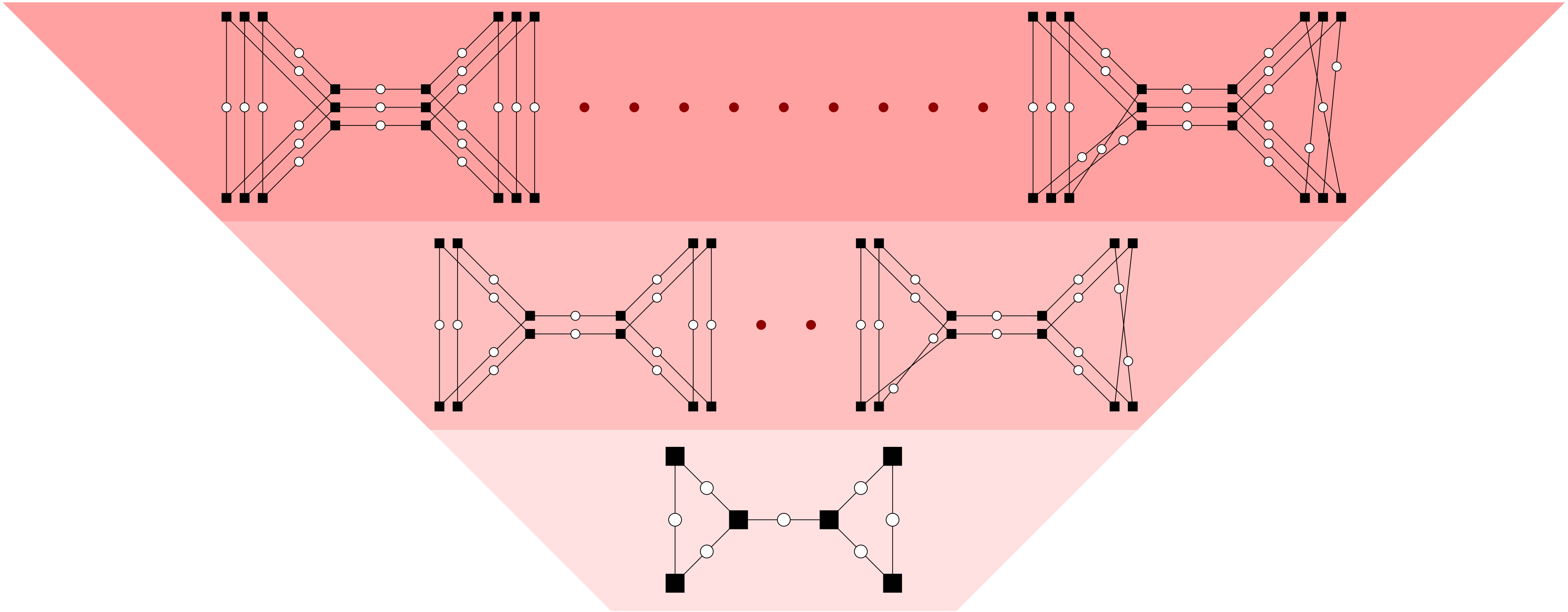
Graph Cover Hierarchy



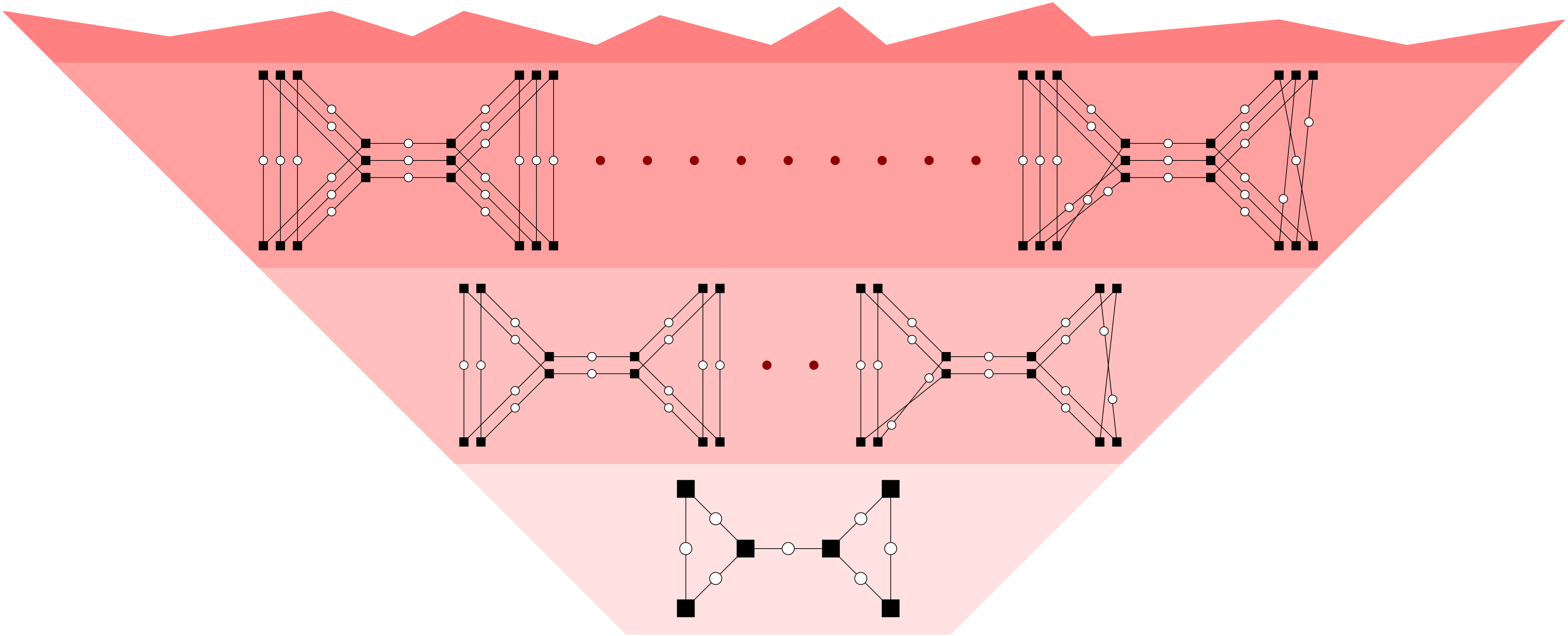
Graph Cover Hierarchy

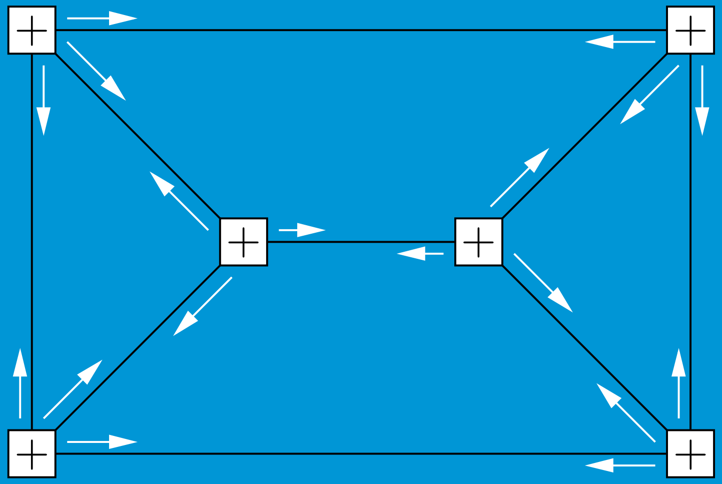


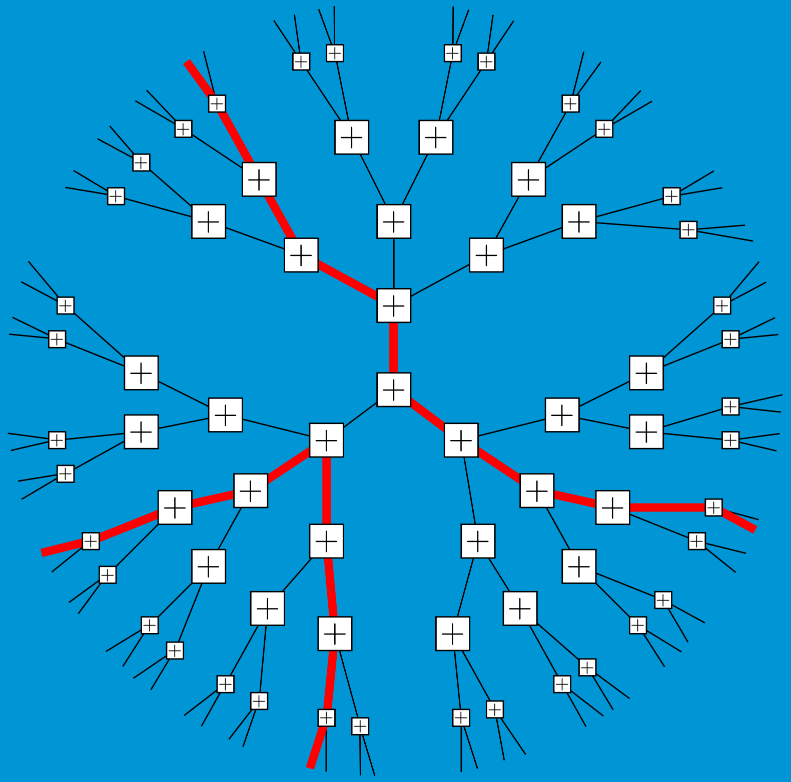
Graph Cover Hierarchy

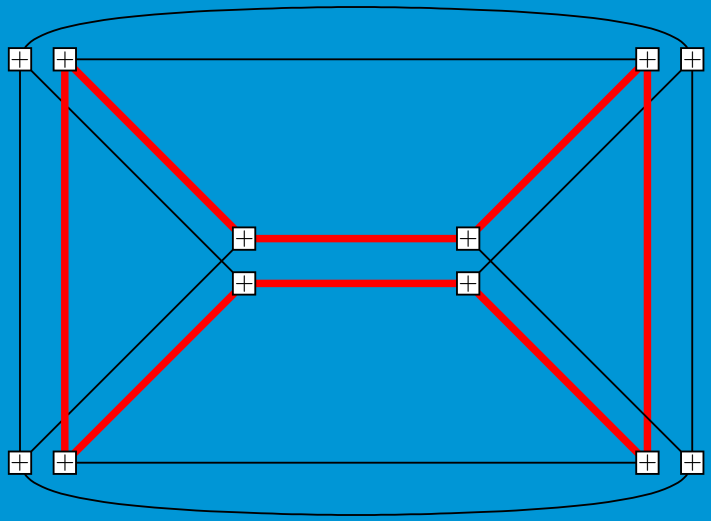


Graph Cover Hierarchy





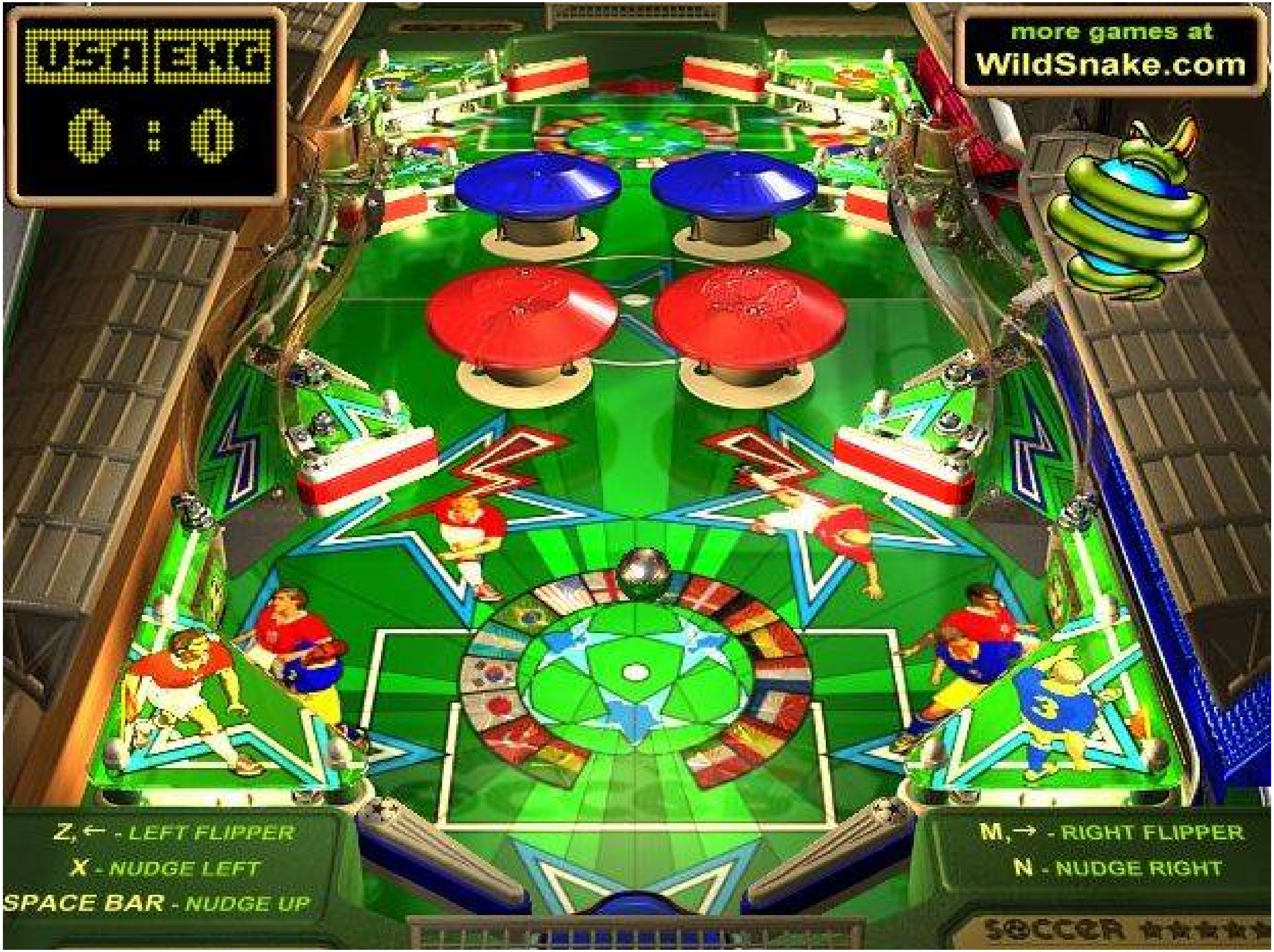




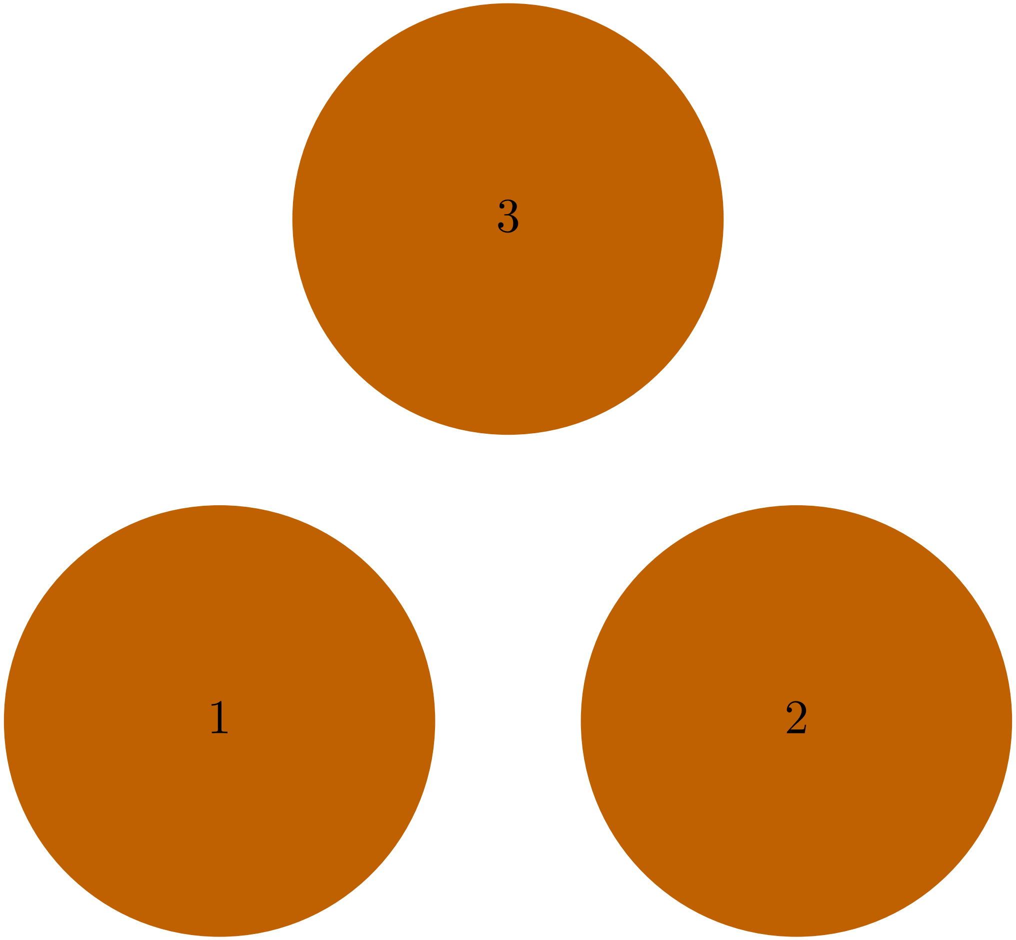
$$\zeta(\mathsf{V}_1,\ldots,\mathsf{V}_n) \;=\; \sum_{\mathbf{k}} \zeta_{\mathbf{k}} \mathbf{V}^{\mathbf{k}} \;=\; \prod_{[\Gamma]} \frac{1}{1-g(\Gamma,\mathbf{V})}$$

Pinball

Pinball

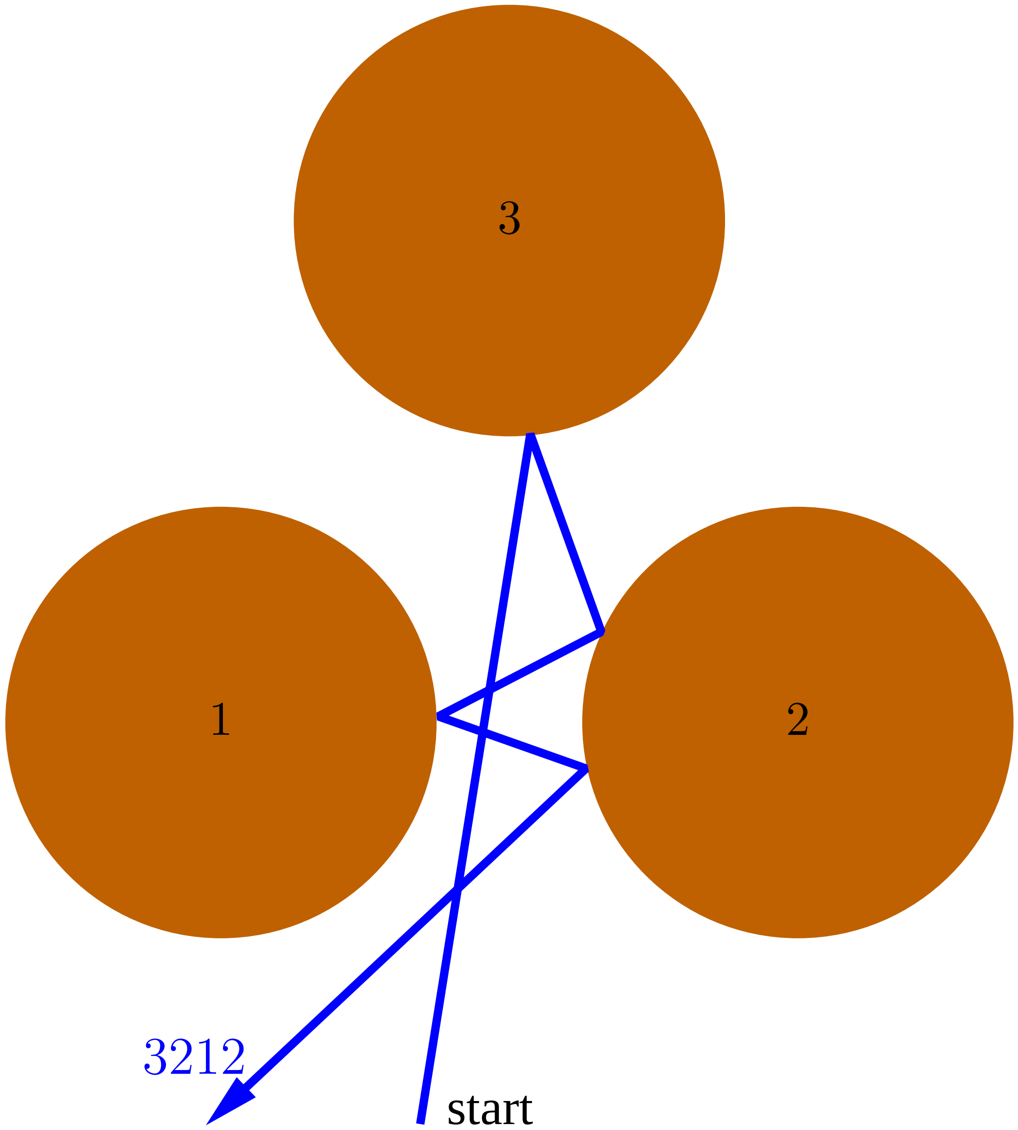

USA ENG
0 : 0
more games at
WildSnake.com
Z,← - LEFT FLIPPER
X - NUDGE LEFT
SPACE BAR - NUDGE UP
M,→ - RIGHT FLIPPER
N - NUDGE RIGHT
SOCCER

Pinball



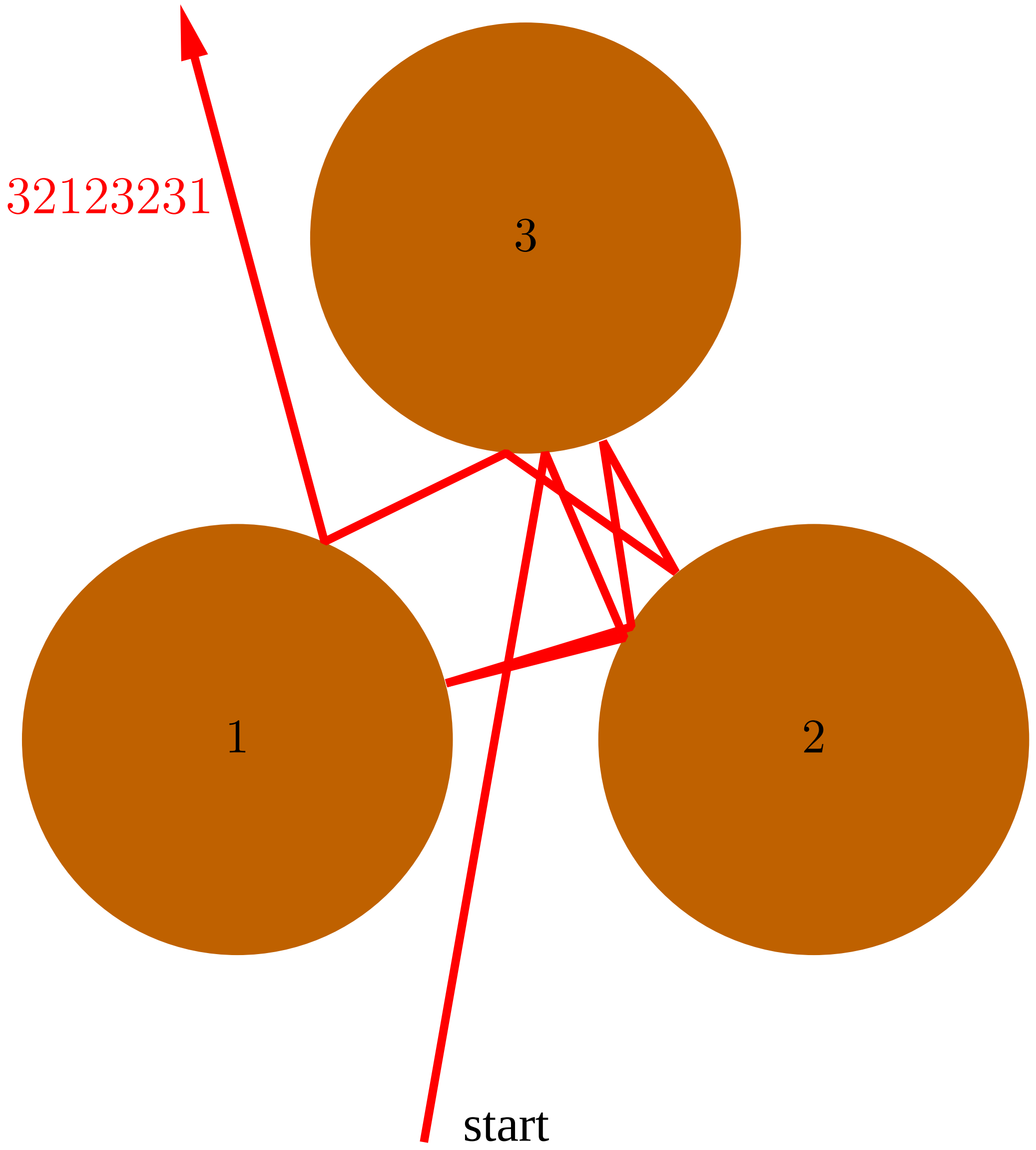
[picture adapted from chaosbook.org]

Pinball



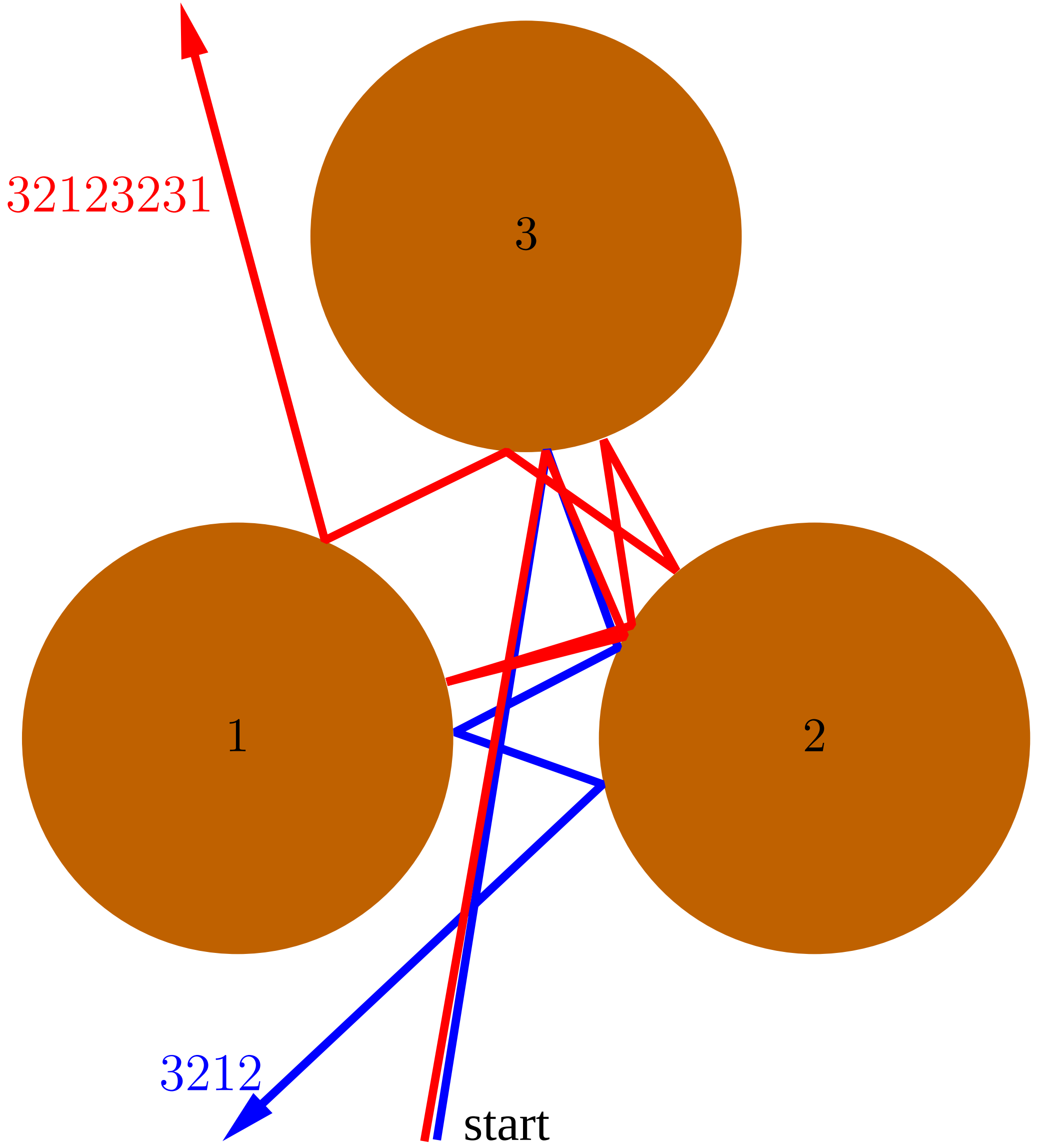
[picture adapted from chaosbook.org]

Pinball



[picture adapted from chaosbook.org]

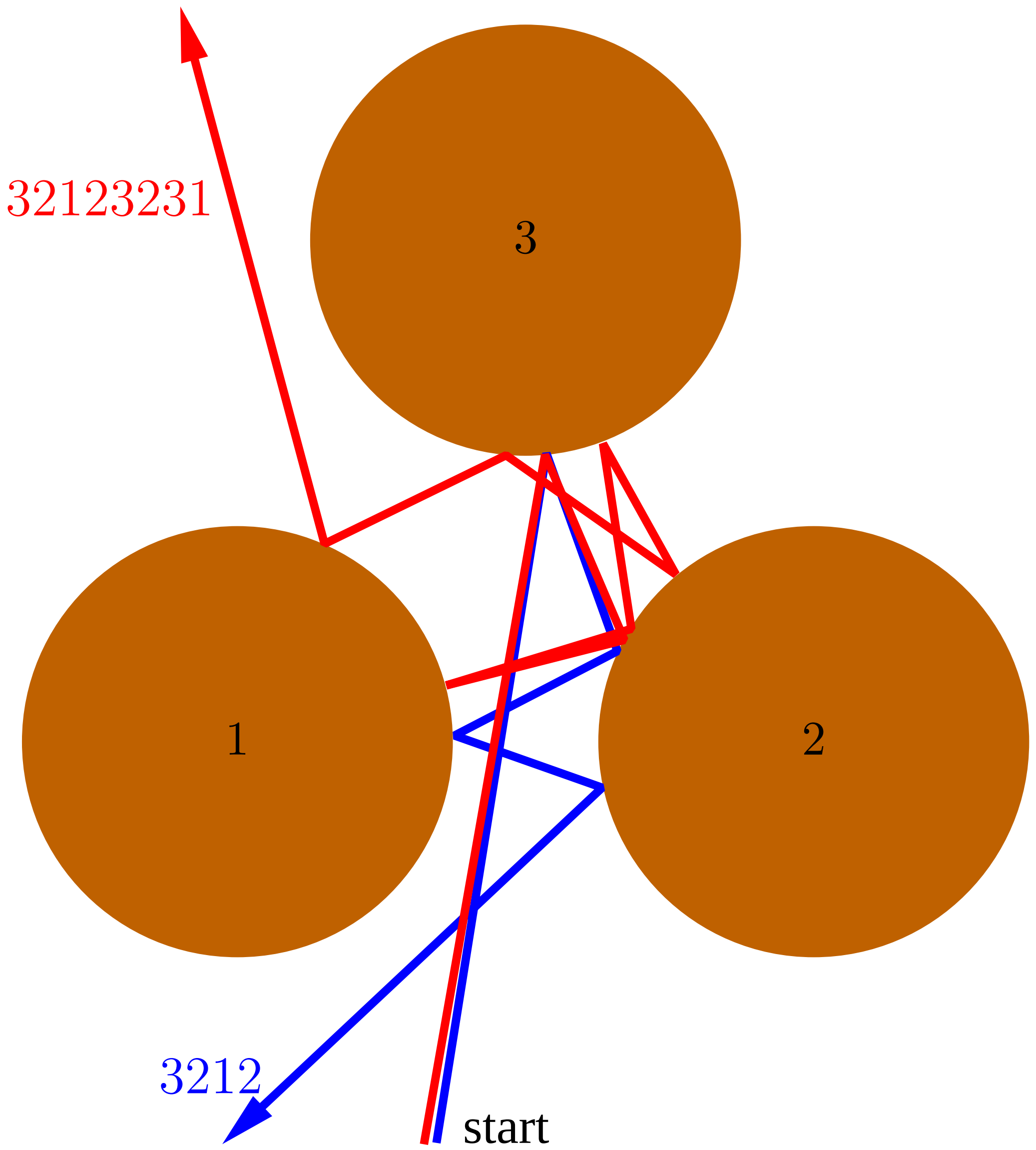
Pinball



[picture adapted from chaosbook.org]

Pinball

The trajectories are difficult to predict.

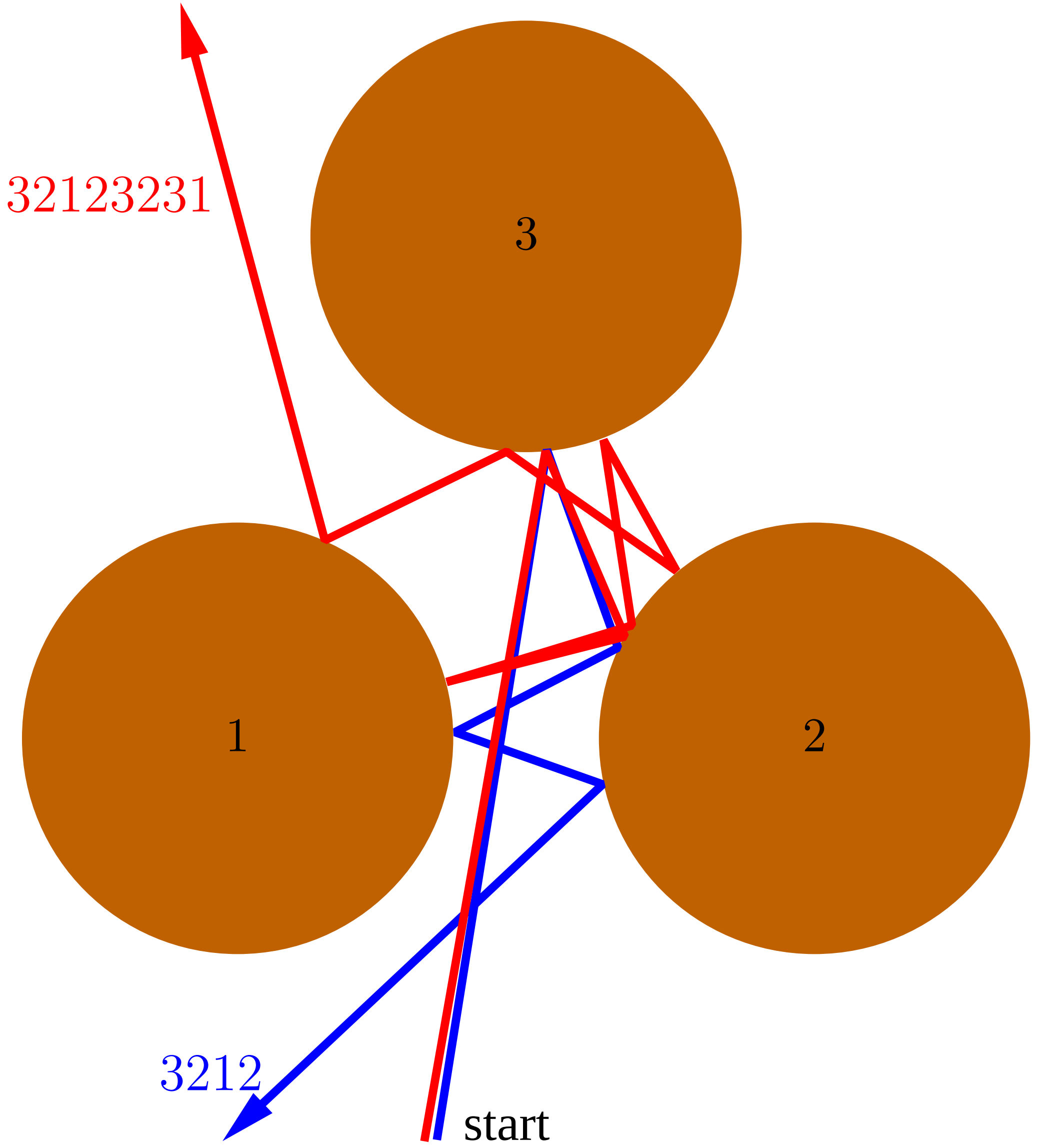


[picture adapted from chaosbook.org]

Pinball

The trajectories are difficult to predict.

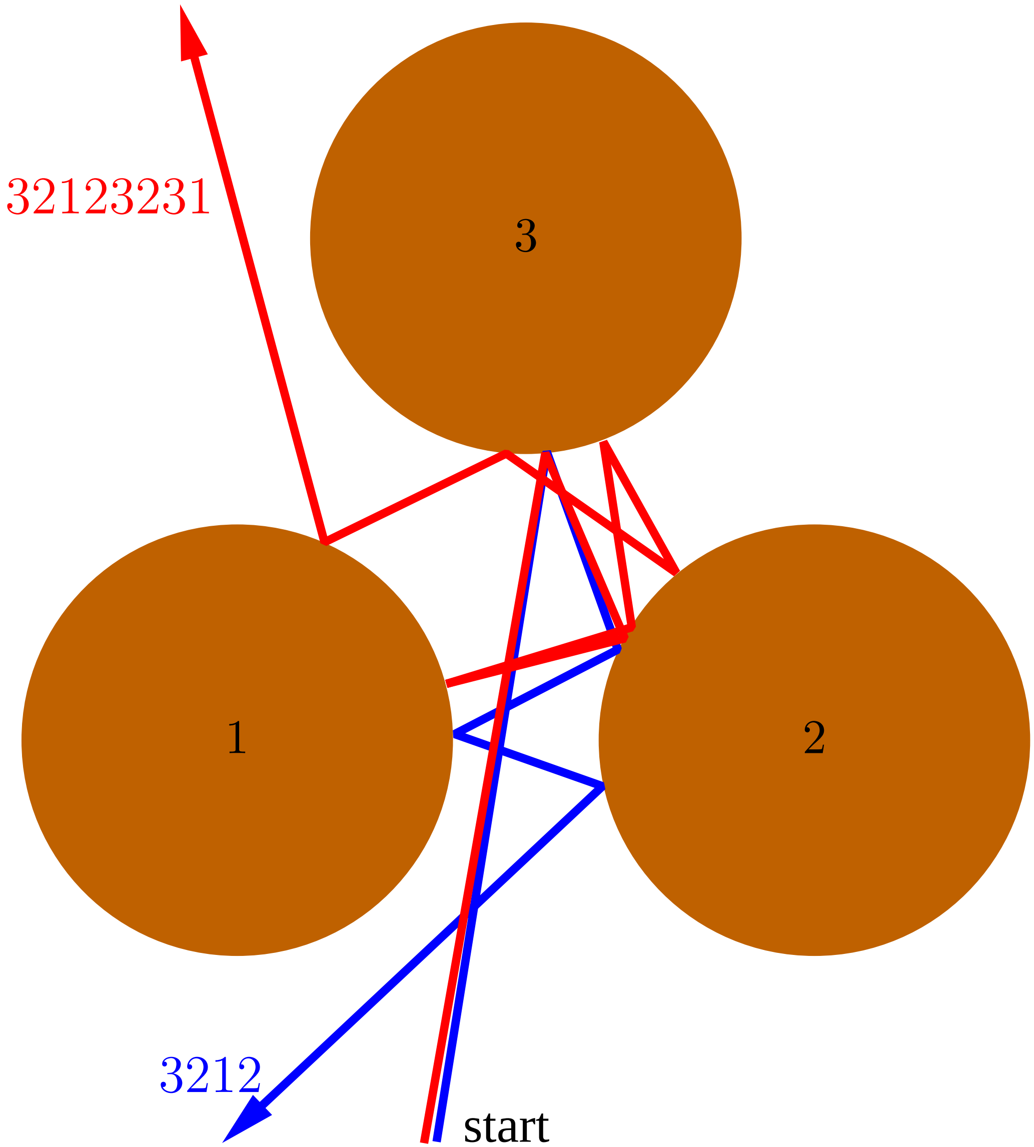
“chaotic system”



[picture adapted from chaosbook.org]

Pinball

The trajectory is difficult to predict.

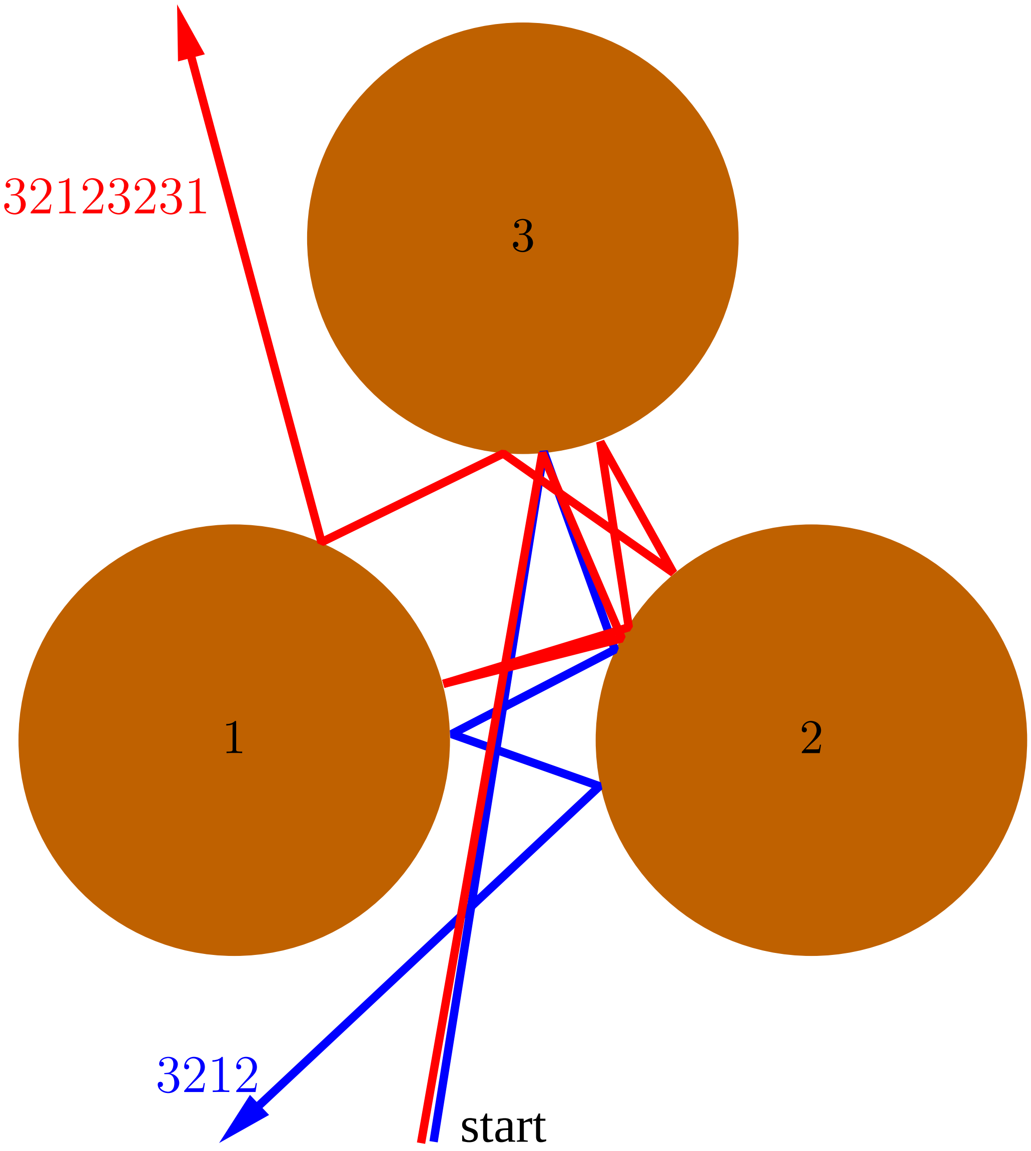


[picture adapted from chaosbook.org]

Pinball

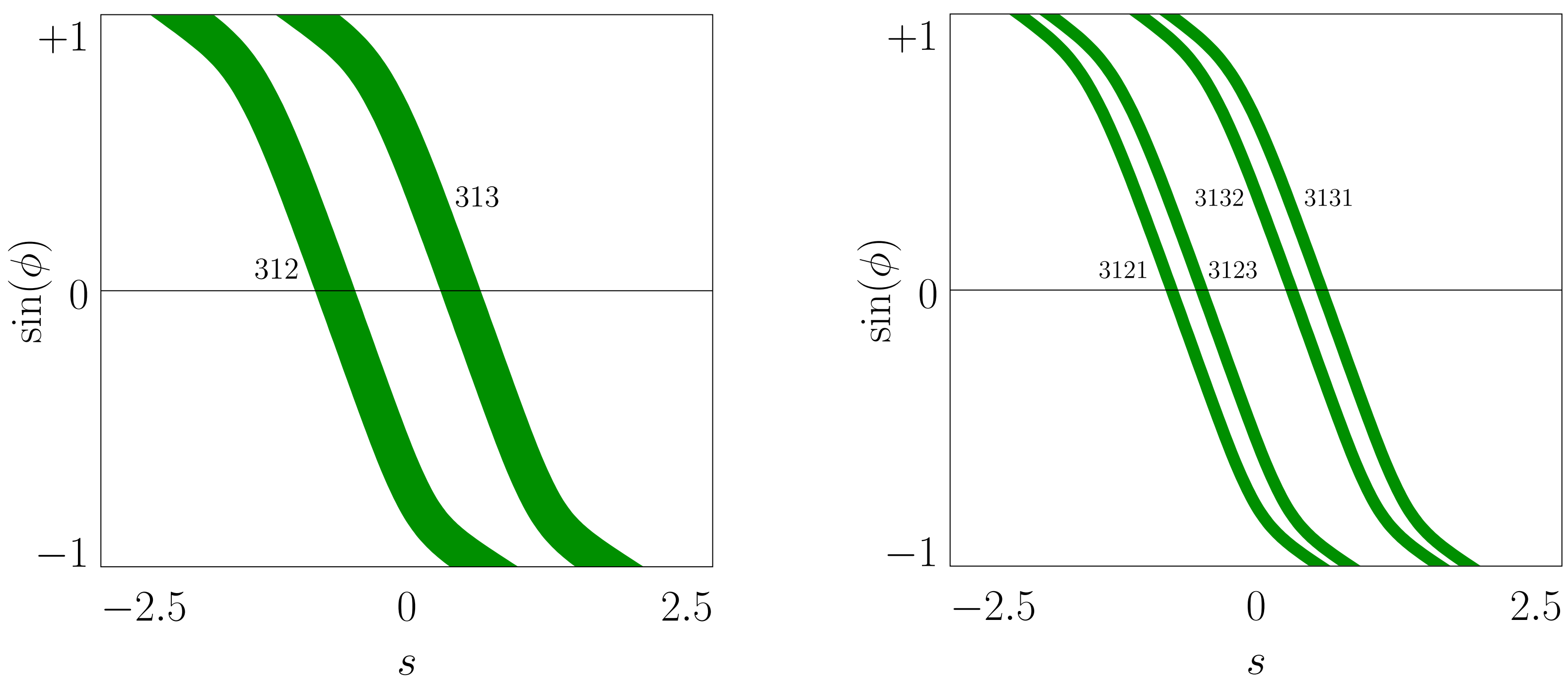
The trajectory is difficult to predict.

However, quantities like the escape rate can be quantified.



[picture adapted from chaosbook.org]

Pinball



$\mathcal{M}_3$: initial conditions for which the ball bounces at 3.

$\mathcal{M}_{31}$: initial conditions for which the ball bounces at 3, 1.

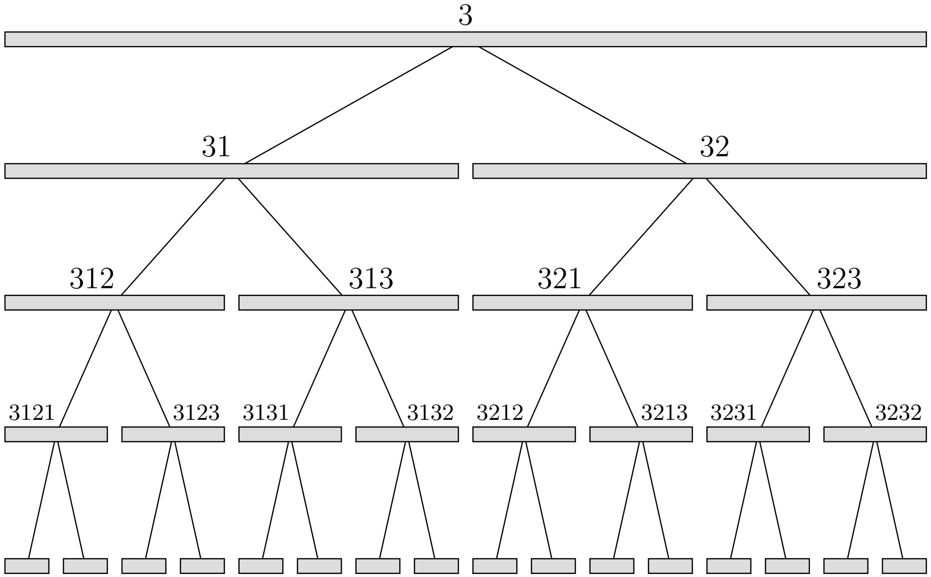
$\mathcal{M}_{312}$: initial conditions for which the ball bounces at 3, 1, 2.

$\mathcal{M}_{3121}$: initial conditions for which the ball bounces at 3, 1, 2, 1.

$\dots$: $\dots$

[picture adapted from chaosbook.org]

Pinball



[picture adapted from chaosbook.org]

Pinball

$$\hat{\theta}_1 = \frac{|\mathcal{M}_1|}{|\mathcal{M}|} + \frac{|\mathcal{M}_2|}{|\mathcal{M}|} + \frac{|\mathcal{M}_3|}{|\mathcal{M}|}$$

Pinball

$$\hat{\theta}_1 = \frac{|\mathcal{M}_1|}{|\mathcal{M}|} + \frac{|\mathcal{M}_2|}{|\mathcal{M}|} + \frac{|\mathcal{M}_3|}{|\mathcal{M}|}$$

$$\hat{\theta}_2 = \frac{|\mathcal{M}_{12}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{13}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{21}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{22}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{31}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{32}|}{|\mathcal{M}|}$$

Pinball

$$
\begin{aligned}
\hat{\theta}_1 &= \frac{|\mathcal{M}_1|}{|\mathcal{M}|} + \frac{|\mathcal{M}_2|}{|\mathcal{M}|} + \frac{|\mathcal{M}_3|}{|\mathcal{M}|} \\
\hat{\theta}_2 &= \frac{|\mathcal{M}_{12}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{13}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{21}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{22}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{31}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{32}|}{|\mathcal{M}|} \\
\vdots & \qquad \vdots
\end{aligned}
$$

Pinball

$$\hat{\theta}_1 = \frac{|\mathcal{M}_1|}{|\mathcal{M}|} + \frac{|\mathcal{M}_2|}{|\mathcal{M}|} + \frac{|\mathcal{M}_3|}{|\mathcal{M}|}$$

$$\hat{\theta}_2 = \frac{|\mathcal{M}_{12}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{13}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{21}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{22}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{31}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{32}|}{|\mathcal{M}|}$$

$$\vdots \qquad \vdots$$

$$\hat{\theta}_n = \sum_{\text{sequence } \mathbf{s} \text{ of length } n} \frac{|\mathcal{M}_{\mathbf{s}}|}{|\mathcal{M}|}$$

Pinball

$$
\begin{aligned}
\hat{\theta}_1 &= \frac{|\mathcal{M}_1|}{|\mathcal{M}|} + \frac{|\mathcal{M}_2|}{|\mathcal{M}|} + \frac{|\mathcal{M}_3|}{|\mathcal{M}|} \\
\hat{\theta}_2 &= \frac{|\mathcal{M}_{12}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{13}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{21}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{22}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{31}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{32}|}{|\mathcal{M}|} \\
\vdots & \qquad \vdots \\
\hat{\theta}_n &= \sum_{\text{sequence } \mathbf{s} \text{ of length } n} \frac{|\mathcal{M}_{\mathbf{s}}|}{|\mathcal{M}|} \\
\vdots & \qquad \vdots
\end{aligned}
$$

Pinball

$$\hat{\theta}_1 = \frac{|\mathcal{M}_1|}{|\mathcal{M}|} + \frac{|\mathcal{M}_2|}{|\mathcal{M}|} + \frac{|\mathcal{M}_3|}{|\mathcal{M}|}$$

$$\hat{\theta}_2 = \frac{|\mathcal{M}_{12}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{13}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{21}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{22}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{31}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{32}|}{|\mathcal{M}|}$$

$$\vdots \qquad \vdots$$

$$\hat{\theta}_n = \sum_{\text{sequence } \mathbf{s} \text{ of length } n} \frac{|\mathcal{M}_{\mathbf{s}}|}{|\mathcal{M}|}$$

$$\vdots \qquad \vdots$$

$$\frac{\hat{\theta}_{n+1}}{\hat{\theta}_n} = \exp(-\gamma_n)$$

Pinball

$$\hat{\theta}_1 = \frac{|\mathcal{M}_1|}{|\mathcal{M}|} + \frac{|\mathcal{M}_2|}{|\mathcal{M}|} + \frac{|\mathcal{M}_3|}{|\mathcal{M}|}$$

$$\hat{\theta}_2 = \frac{|\mathcal{M}_{12}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{13}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{21}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{22}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{31}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{32}|}{|\mathcal{M}|}$$

$$\vdots \qquad \vdots$$

$$\hat{\theta}_n = \sum_{\text{sequence } \mathbf{s} \text{ of length } n} \frac{|\mathcal{M}_{\mathbf{s}}|}{|\mathcal{M}|}$$

$$\vdots \qquad \vdots$$

$$\frac{\hat{\theta}_{n+1}}{\hat{\theta}_n} = \exp(-\gamma_n) \quad \rightarrow \quad \exp(-\gamma)$$

Pinball

$$\hat{\theta}_1 = \frac{|\mathcal{M}_1|}{|\mathcal{M}|} + \frac{|\mathcal{M}_2|}{|\mathcal{M}|} + \frac{|\mathcal{M}_3|}{|\mathcal{M}|}$$

$$\hat{\theta}_2 = \frac{|\mathcal{M}_{12}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{13}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{21}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{22}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{31}|}{|\mathcal{M}|} + \frac{|\mathcal{M}_{32}|}{|\mathcal{M}|}$$

$$\vdots \qquad \vdots$$

$$\hat{\theta}_n = \sum_{\text{sequence } \mathrm{s} \text{ of length } n} \frac{|\mathcal{M}_{\mathbf{s}}|}{|\mathcal{M}|}$$

$$\vdots \qquad \vdots$$

$$\frac{\hat{\theta}_{n+1}}{\hat{\theta}_n} = \exp(-\gamma_n) \quad \to \quad \exp(-\gamma)$$

γ : escape rate

Pinball

- Ideally, we compute $\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \ldots$, and determine from this γ.

Pinball

- Ideally, we compute $\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \ldots$, and determine from this γ. However, usually this is too complicated.

Pinball

- Ideally, we compute $\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \ldots$, and determine from this γ. However, usually this is too complicated.

- Note that the **power series**

$$\hat{\theta}(z) = \sum_{n=1}^{\infty} \hat{\theta}_n z^n$$

has **convergence radius** $\exp(\gamma)$.

Pinball

- Ideally, we compute $\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \ldots$, and determine from this γ. However, usually this is too complicated.

- Note that the **power series**

$$\hat{\theta}(z) = \sum_{n=1}^{\infty} \hat{\theta}_n z^n$$

has **convergence radius** $\exp(\gamma)$.

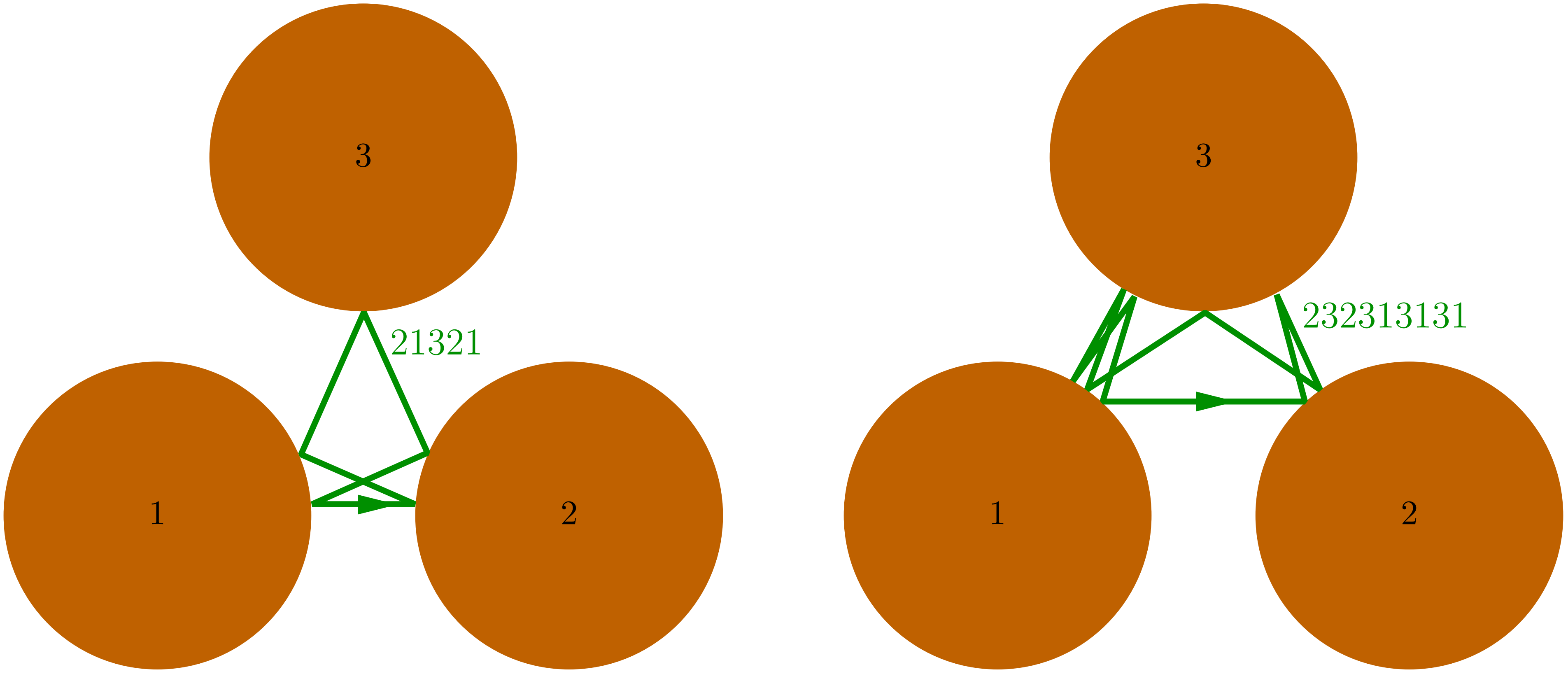
- Alternative approach: set up a **new power series**

$$\theta(z) = \sum_{n=1}^{\infty} \theta_n z^n$$

so that its **convergence radius** equals $\exp(\gamma)$.

Pinball

Main idea: look at periodic trajectories.



[picture adapted from chaosbook.org]

Pinball

Pinball

- Let the string $\mathbf{s}$ label the periodic trajectories.

Pinball

- Let the string $\mathbf{s}$ label the periodic trajectories.
- Let the periodic trajectory $\mathbf{s}$ go through $x_{\mathbf{s}}$.

Pinball

- Let the string $\mathbf{s}$ label the periodic trajectories.
- Let the periodic trajectory $\mathbf{s}$ go through $x_{\mathbf{s}}$.
- Let the periodic trajectory $\mathbf{s}$ have period $T_{\mathrm{p},\mathbf{s}}$.

Pinball

- Let the string $\mathbf{s}$ label the periodic trajectories.
- Let the periodic trajectory $\mathbf{s}$ go through $x_{\mathbf{s}}$.
- Let the periodic trajectory $\mathbf{s}$ have period $T_{\mathrm{p},\mathbf{s}}$.
- Then define

$$
\begin{aligned}
\theta(z) &\triangleq \sum_{n=1}^{\infty} z^n \sum_{\text{sequence } \mathbf{s} \text{ of length } n} \frac{1}{|\Lambda_{\mathbf{s}}|} \\
&= \frac{z^1}{|\Lambda_1|} + \frac{z^1}{|\Lambda_2|} + \frac{z^1}{|\Lambda_3|} + \frac{z^2}{|\Lambda_{12}|} + \frac{z^2}{|\Lambda_{13}|} + \cdots
\end{aligned}
$$

where Λ_{s} is the unstable eigenvalue of the Jacobian matrix $J^t(x_i)$ evaluated for $t = T_{\mathrm{p},\mathbf{s}}$. (Due to the low dimensionality, the Jacobian can have at most one unstable eigenvalue for the present setup.)

Pinball

- $\theta(z)$ can be rewritten as follows:

$$\theta(z) = \sum_{\text{prime cycle } \mathbf{p}} n_{\mathbf{p}} \sum_{r=1}^{\infty} \left(\frac{z^{n_{\mathbf{p}}}}{|\Lambda_{\mathbf{p}}|} \right)^{r}$$

Pinball

- $\theta(z)$ can be rewritten as follows:

$$\theta(z) = \sum_{\text{prime cycle } \mathbf{p}} n_{\mathbf{p}} \sum_{r=1}^{\infty} \left(\frac{z^{n_{\mathbf{p}}}}{|\Lambda_{\mathbf{p}}|} \right)^{r} = \sum_{\substack{p \\ \text{prime cycle}}} \frac{n_{\mathbf{p}} t_{\mathbf{p}}}{1 - t_{\mathbf{p}}}, \qquad t_{\mathbf{p}} = \frac{z^{n_{\mathbf{p}}}}{|\Lambda_{\mathbf{p}}|}$$

Pinball

- $\theta(z)$ can be rewritten as follows:

$$\theta(z) = \sum_{\text{prime cycle } \mathbf{p}} n_{\mathbf{p}} \sum_{r=1}^{\infty} \left(\frac{z^{n_{\mathbf{p}}}}{|\Lambda_{\mathbf{p}}|} \right)^r = \sum_{\substack{p \\ \text{prime cycle}}} \frac{n_{\mathbf{p}} t_{\mathbf{p}}}{1 - t_{\mathbf{p}}}, \qquad t_{\mathbf{p}} = \frac{z^{n_{\mathbf{p}}}}{|\Lambda_{\mathbf{p}}|}$$

- Definition of dynamical zeta function:

$$\zeta(z) \triangleq \prod_{\text{prime cycle } \mathbf{p}} \frac{1}{1 - t_{\mathbf{p}}}, \qquad t_{\mathbf{p}} = \frac{z^{n_{\mathbf{p}}}}{|\Lambda_{\mathbf{p}}|}$$

Pinball

- $\theta(z)$ can be rewritten as follows:

$$\theta(z) = \sum_{\text{prime cycle } \mathbf{p}} n_{\mathbf{p}} \sum_{r=1}^{\infty} \left(\frac{z^{n_{\mathbf{p}}}}{|\Lambda_{\mathbf{p}}|} \right)^{r} = \sum_{\substack{p \\ \text{prime cycle}}} \frac{n_{\mathbf{p}} t_{\mathbf{p}}}{1 - t_{\mathbf{p}}}, \qquad t_{\mathbf{p}} = \frac{z^{n_{\mathbf{p}}}}{|\Lambda_{\mathbf{p}}|}$$

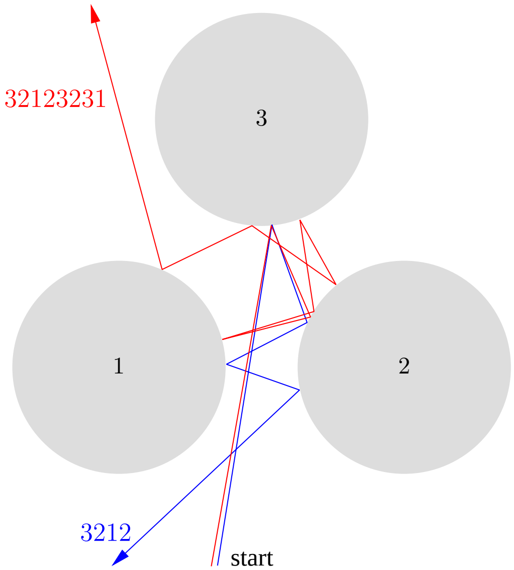
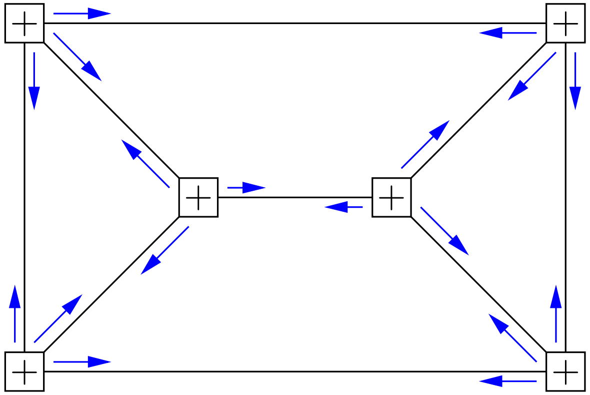
- Definition of dynamical zeta function:

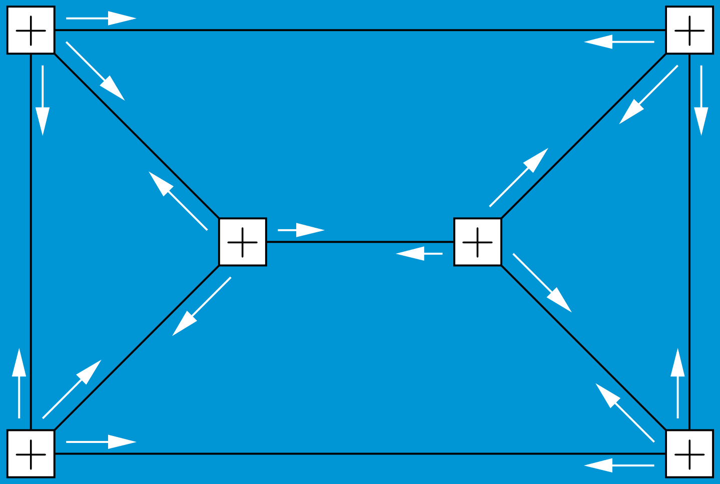
$$\zeta(z) \triangleq \prod_{\text{prime cycle } \mathbf{p}} \frac{1}{1 - t_{\mathbf{p}}}, \qquad t_{\mathbf{p}} = \frac{z^{n_{\mathbf{p}}}}{|\Lambda_{\mathbf{p}}|}$$

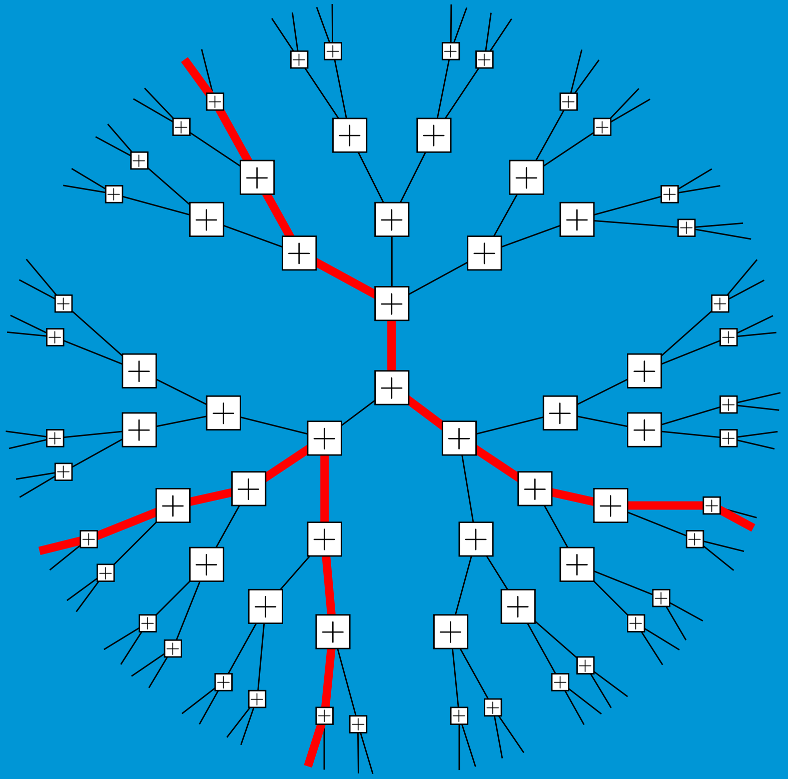
- Note:

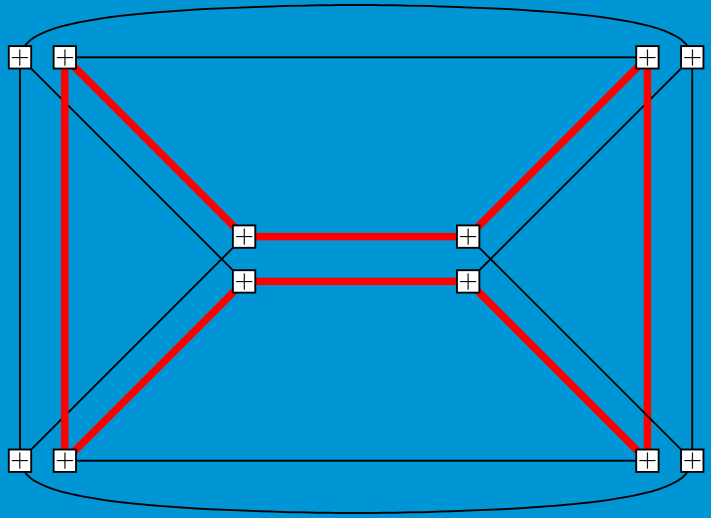
$$\theta(z) = z \frac{\mathrm{d}}{\mathrm{d}z} \log\left(\zeta(z)\right)$$

Analogy Pinball vs. MPI Decoding

 pinball	 message-passing iterative decoding of cycle codes
trajectory	minimial deviation in computation trees
periodic trajectory	codeword in finite graph cover / graph-cover pseudo-codeword
dynamical edge zeta function	graph zeta function



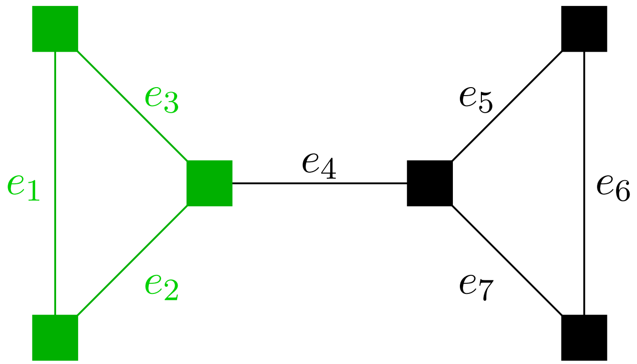




$$\zeta(\mathsf{V}_1,\ldots,\mathsf{V}_n) = \sum_{\mathbf{k}} \zeta_{\mathbf{k}} \mathbf{V}^{\mathbf{k}} = \prod_{[\Gamma]} \frac{1}{1-g(\Gamma,\mathbf{V})}$$

The Edge Zeta Function of a Graph

Definition (Hashimoto, see also Stark/Terras):



Here: $\Gamma = (e_1, e_2, e_3)$

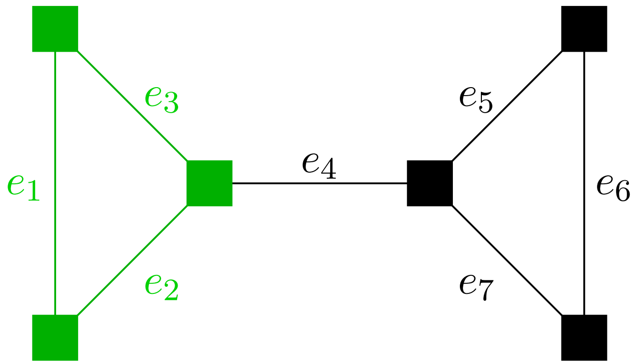
Let Γ be a path in a graph X with edge-set E; write

$$\Gamma = (e_{i_1}, \ldots, e_{i_k})$$

to indicate that Γ begins with the edge e_{i_1} and ends with the edge e_{i_k}.

The Edge Zeta Function of a Graph

Definition (Hashimoto, see also Stark/Terras):

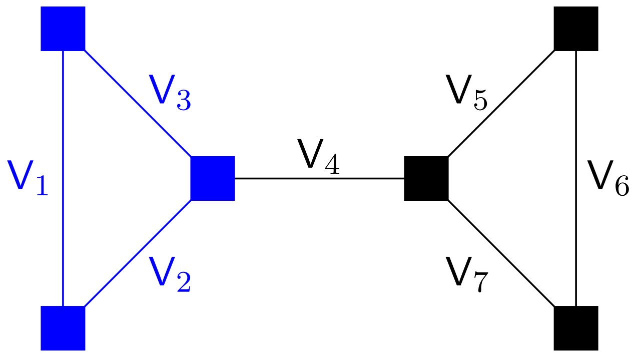


Here: $\Gamma = (e_1, e_2, e_3)$

Let Γ be a path in a graph X with edge-set E; write

$$\Gamma = (e_{i_1}, \ldots, e_{i_k})$$

to indicate that Γ begins with the edge e_{i_1} and ends with the edge e_{i_k}.



Here: $g(\Gamma, \mathbf{V}) = \mathsf{V}_1\mathsf{V}_2\mathsf{V}_3$

The monomial of Γ is given by

$$g(\Gamma, \mathbf{V}) \triangleq \mathsf{V}_{i_1} \cdots \mathsf{V}_{i_k},$$

where the V_i's are indeterminates.

The Edge Zeta Function of a Graph

Definition (Hashimoto, see also Stark/Terras):

The edge zeta function of X is defined to be the power series

$$\zeta_X(\mathbf{V}) = \zeta_X(\mathsf{V}_1, \ldots, \mathsf{V}_n) \in \mathbb{Z}[[\mathsf{V}_1, \ldots, \mathsf{V}_n]]$$

given by

$$\zeta_X(\mathbf{V}) = \zeta_X(\mathsf{V}_1, \ldots, \mathsf{V}_n) = \prod_{[\Gamma] \in A(X)} \frac{1}{1 - g(\Gamma, \mathbf{V})},$$

where $A(X)$ is the collection of equivalence classes of

backtrackless, tailless, primitive cycles in X.

The Edge Zeta Function of a Graph

Definition (Hashimoto, see also Stark/Terras):

The edge zeta function of X is defined to be the power series

$$\zeta_X(\mathbf{V}) = \zeta_X(\mathsf{V}_1, \ldots, \mathsf{V}_n) \in \mathbb{Z}[[\mathsf{V}_1, \ldots, \mathsf{V}_n]]$$

given by

$$\zeta_X(\mathbf{V}) = \zeta_X(\mathsf{V}_1, \ldots, \mathsf{V}_n) = \prod_{[\Gamma] \in A(X)} \frac{1}{1 - g(\Gamma, \mathbf{V})},$$

where $A(X)$ is the collection of equivalence classes of

backtrackless, tailless, primitive cycles in X.

Note: unless X contains only one cycle,

the set $A(X)$ will be countably infinite.

The Edge Zeta Function of a Graph

Theorem (Bass):

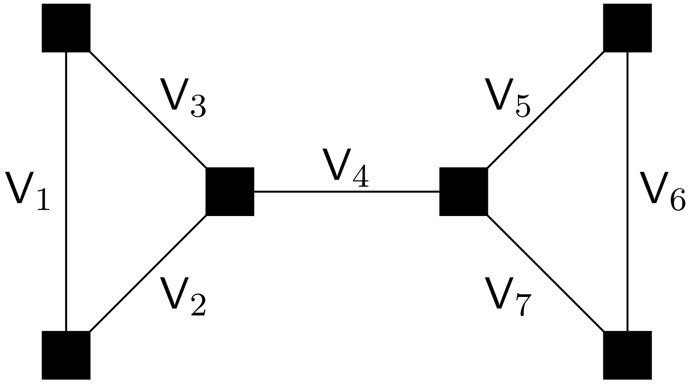
- The edge zeta function $\zeta_X(\mathsf{V}_1, \ldots, \mathsf{V}_n)$ is a rational function.
- More precisely, for any directed graph $\vec{X}$ of X, we have

$$\zeta_X(\mathsf{V}_1, \ldots, \mathsf{V}_n) = \frac{1}{\det\left(\mathbf{I} - \mathbf{V}\mathbf{M}(\vec{X})\right)} = \frac{1}{\det\left(\mathbf{I} - \mathbf{M}(\vec{X})\mathbf{V}\right)}$$

  where

  - $\mathbf{I}$ is the identity matrix of size $2n$,
  - $\mathbf{V} = \text{diag}(\mathsf{V}_1, \ldots, \mathsf{V}_n, \mathsf{V}_1, \ldots, \mathsf{V}_n)$ is a diagonal matrix of indeterminants.
  - $\mathbf{M}(\vec{X})$ is a $2n \times 2n$ matrix derived from some directed graph version $\vec{X}$ of X.

Example of Edge Zeta Function

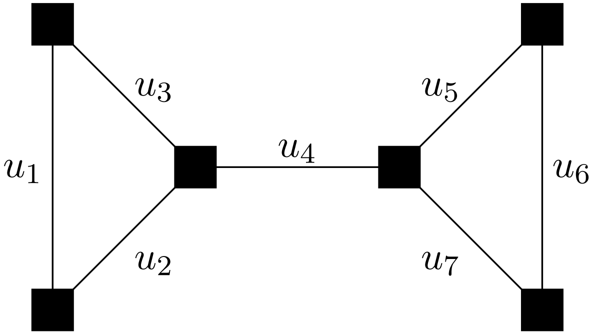


This normal graph N has the following edge zeta function:

$$\zeta_N(V_1, \ldots, V_7) = \frac{1}{\det(\mathbf{I}_{14} - \mathbf{VM})}$$

$$= \frac{1}{\begin{array}{l} 1 - 2V_1V_2V_3 + V_1^2V_2^2V_3^2 - 2V_5V_6V_7 + 4V_1V_2V_3V_5V_6V_7 - 2V_1^2V_2^2V_3^2V_5V_6V_7 \\ -4V_1V_2V_3V_4^2V_5V_6V_7 + 4V_1^2V_2^2V_3^2V_4^2V_5V_6V_7 + V_5^2V_6^2V_7^2 - 2V_1V_2V_3V_5^2V_6^2V_7^2 \\ +V_1^2V_2^2V_3^2V_5^2V_6^2V_7^2 + 4V_1V_2V_3V_4^2V_5^2V_6^2V_7^2 - 4V_1^2V_2^2V_3^2V_4^2V_5^2V_6^2V_7^2 \end{array}}$$

Example of Edge Zeta Function

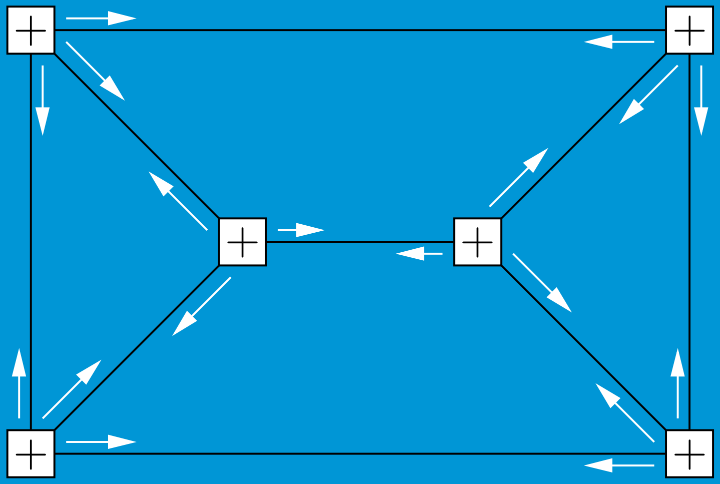


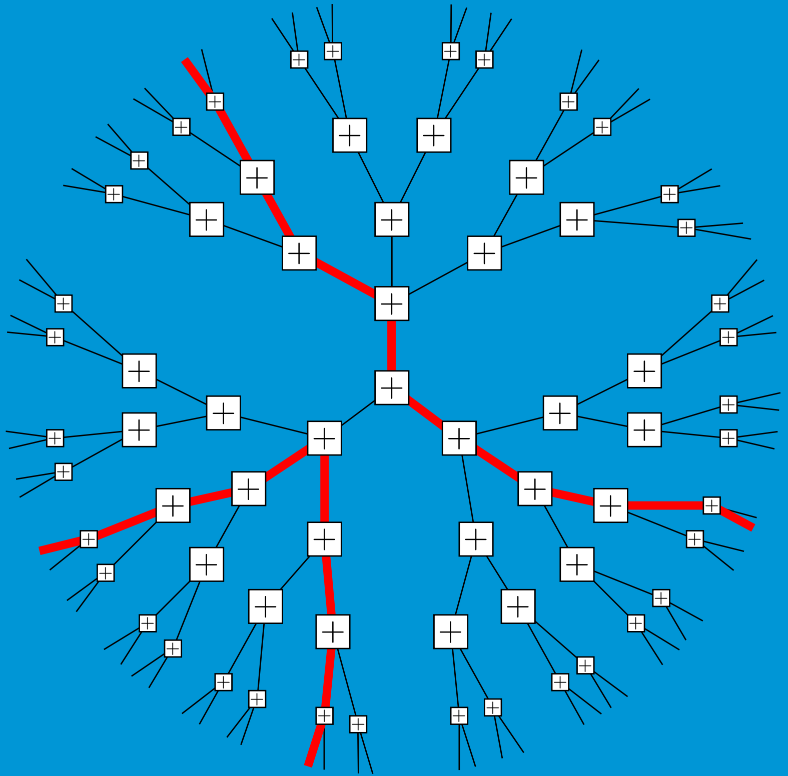
The Taylor series exansion is

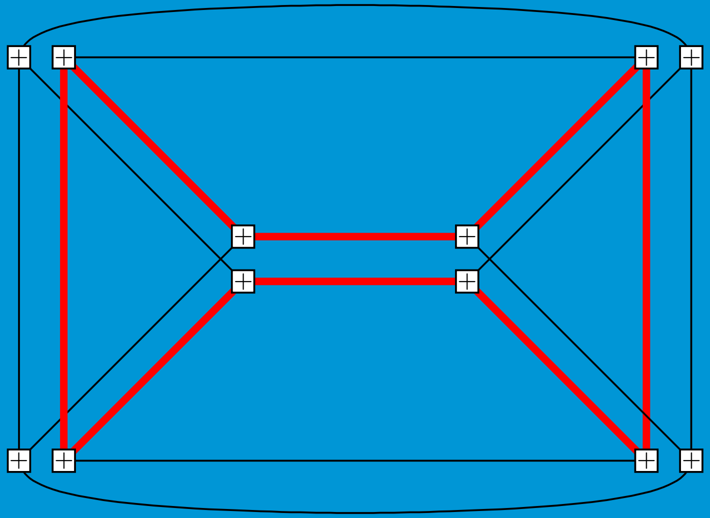
$$\begin{aligned}\zeta_N(V_1, \ldots, V_7) &= 1 + 2V_1V_2V_3 + 3V_1^2V_2^2V_3^2 + 2V_5V_6V_7 \\ &\quad + 4V_1V_2V_3V_5V_6V_7 + 6V_1^2V_2^2V_3^2V_5V_6V_7 \\ &\quad + 4V_1V_2V_3V_4^2V_5V_6V_7 + 12V_1^2V_2^2V_3^2V_4^2V_5V_6V_7 \\ &\quad + \cdots\end{aligned}$$

We get the following exponent vectors:

$(0, 0, 0, 0, 0, 0, 0)$	codeword
$(1, 1, 1, 0, 0, 0, 0)$	codeword
$(2, 2, 2, 0, 0, 0, 0)$	pseudo-codeword (in $\mathbb{Z}$-span)
$(0, 0, 0, 0, 1, 1, 1)$	codeword
$(1, 1, 1, 0, 1, 1, 1)$	codeword
$(2, 2, 2, 0, 1, 1, 1)$	pseudo-codeword (in $\mathbb{Z}$-span)
$(1, 1, 1, 2, 1, 1, 1)$	pseudo-codeword (not in $\mathbb{Z}$-span)
$(2, 2, 2, 2, 1, 1, 1)$	pseudo-codeword (in $\mathbb{Z}$-span)







$$\zeta(\mathsf{V}_1,\ldots,\mathsf{V}_n) = \sum_{\mathbf{k}} \zeta_{\mathbf{k}} \mathbf{V}^{\mathbf{k}} = \prod_{[\Gamma]} \frac{1}{1-g(\Gamma,\mathbf{V})}$$

New Theorem for Cycle Codes

Rough statement:

$$\begin{pmatrix} \text{region of convergence} \\ \text{of sum-product algorithm} \\ \text{to the all-zero codeword} \end{pmatrix} = \begin{pmatrix} \text{region of convergence} \\ \text{of the edge zeta function} \end{pmatrix}$$

New Theorem for Cycle Codes

Rough statement:

$$\begin{pmatrix}\text{region of convergence}\\ \text{of sum-product algorithm}\\ \text{to the all-zero codeword}\end{pmatrix} = \begin{pmatrix}\text{region of convergence}\\ \text{of the edge zeta function}\end{pmatrix}$$

More precisely:

$$\begin{pmatrix}\text{The sum-product algorithm}\\ \text{converges to the all-zero codeword}\\ \text{for the log-likelihood vector } \lambda\end{pmatrix} \Leftrightarrow \begin{pmatrix}\text{V is in the region of convergence}\\ \text{of the edge zeta function,}\\ \text{where } \mathrm{V}_e = \exp(-\lambda_e) \ \forall e\end{pmatrix}$$

New Theorem for Cycle Codes

Rough statement:

$$\begin{pmatrix}\text{region of convergence}\\ \text{of sum-product algorithm}\\ \text{to the all-zero codeword}\end{pmatrix} = \begin{pmatrix}\text{region of convergence}\\ \text{of the edge zeta function}\end{pmatrix}$$

More precisely:

$$\begin{pmatrix}\text{The sum-product algorithm}\\ \text{converges to the all-zero codeword}\\ \text{for the log-likelihood vector } \lambda\end{pmatrix} \Leftrightarrow \begin{pmatrix}\mathbf{V}\text{ is in the region of convergence}\\ \text{of the edge zeta function,}\\ \text{where } \mathrm{V}_e = \exp(-\lambda_e)\ \ \forall e\end{pmatrix}$$

Note: global convergence result!

New Theorem for Cycle Codes

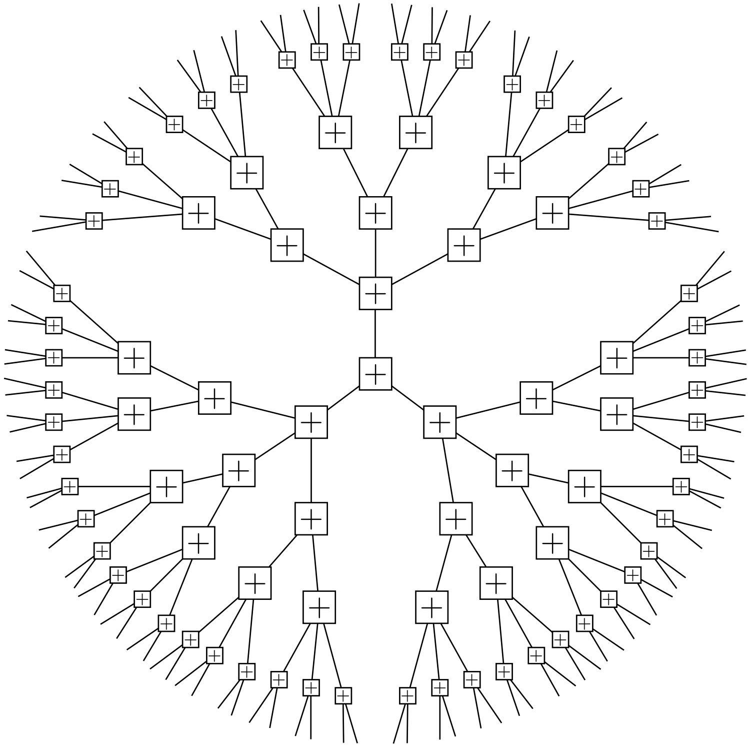
Rough statement:

$$\begin{pmatrix}\text{region of convergence}\\ \text{of sum-product algorithm}\\ \text{to the all-zero codeword}\end{pmatrix} = \begin{pmatrix}\text{region of convergence}\\ \text{of the edge zeta function}\end{pmatrix}$$

Corollary:

The region of all log-likelihood vectors λ
for which the sum-product algorithm converges
to the all-zero codeword
is given by a determinantal expression.

Some intuition behind this statement



$$\zeta_N(\mathsf{V}_1,\ldots,\mathsf{V}_n) \triangleq \prod_{[\Gamma]\in A(N)} \frac{1}{1-g(\Gamma,\mathbf{V})}$$

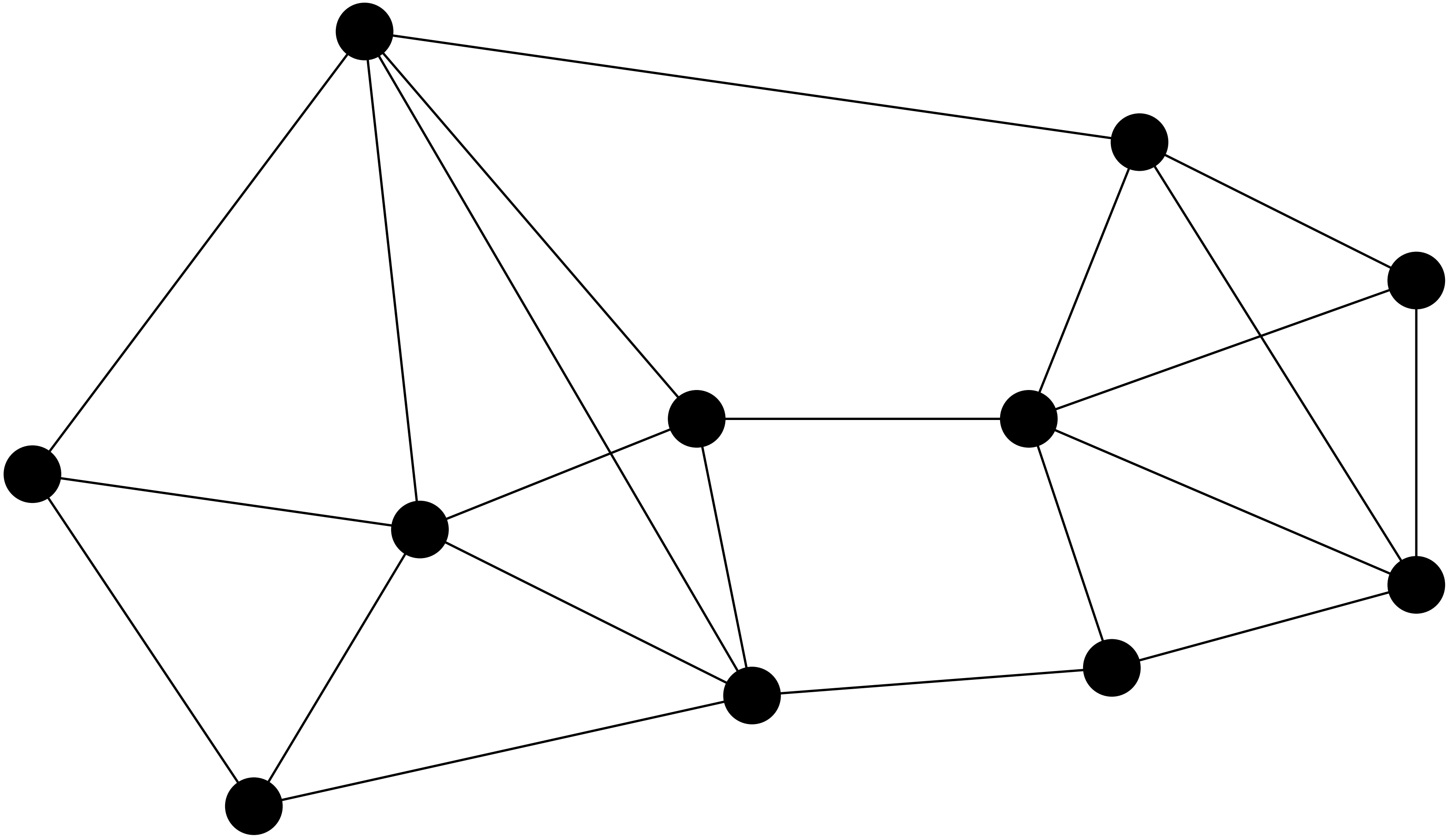
$$= \frac{1}{\det\left(\mathbf{I}-\mathbf{M}(\vec{N})\mathbf{V}\right)}$$

It turns out that key objects for analyzing computation trees of the normal factor graph N are

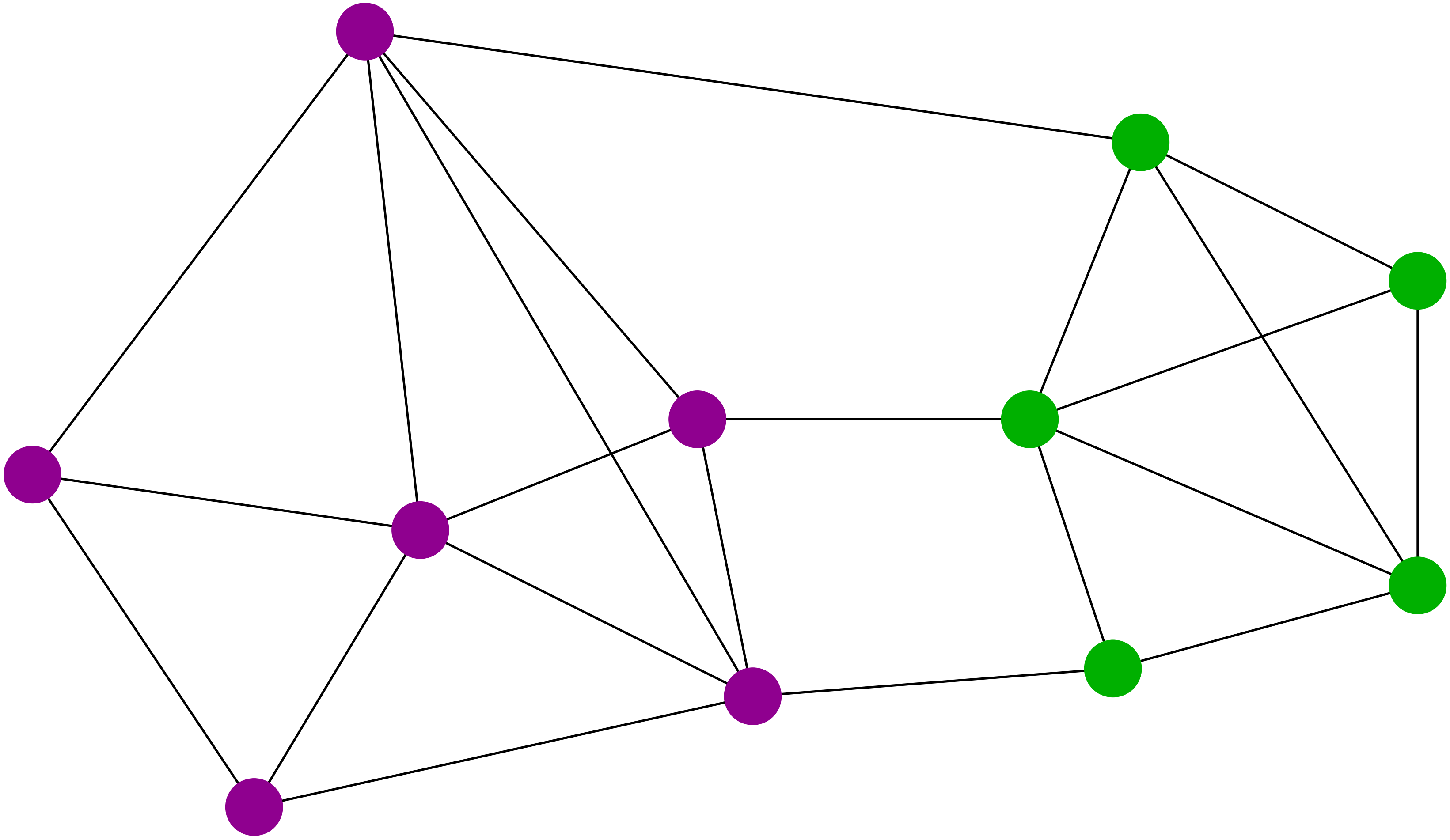
$$(\mathbf{M}(\vec{N})\mathbf{V})^k, \quad k \geq 0.$$

Community detection in networks

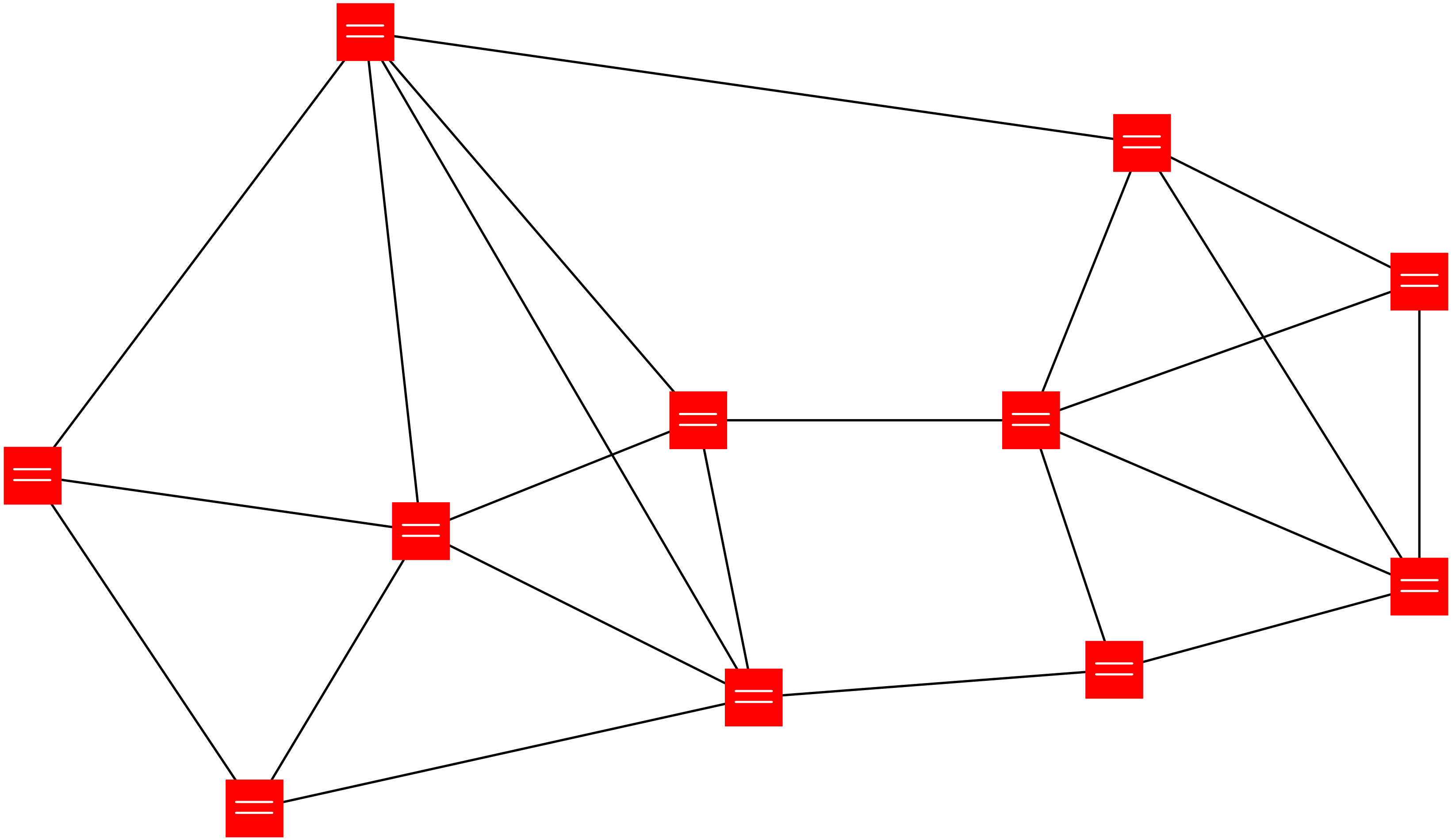
Community Detection in Networks



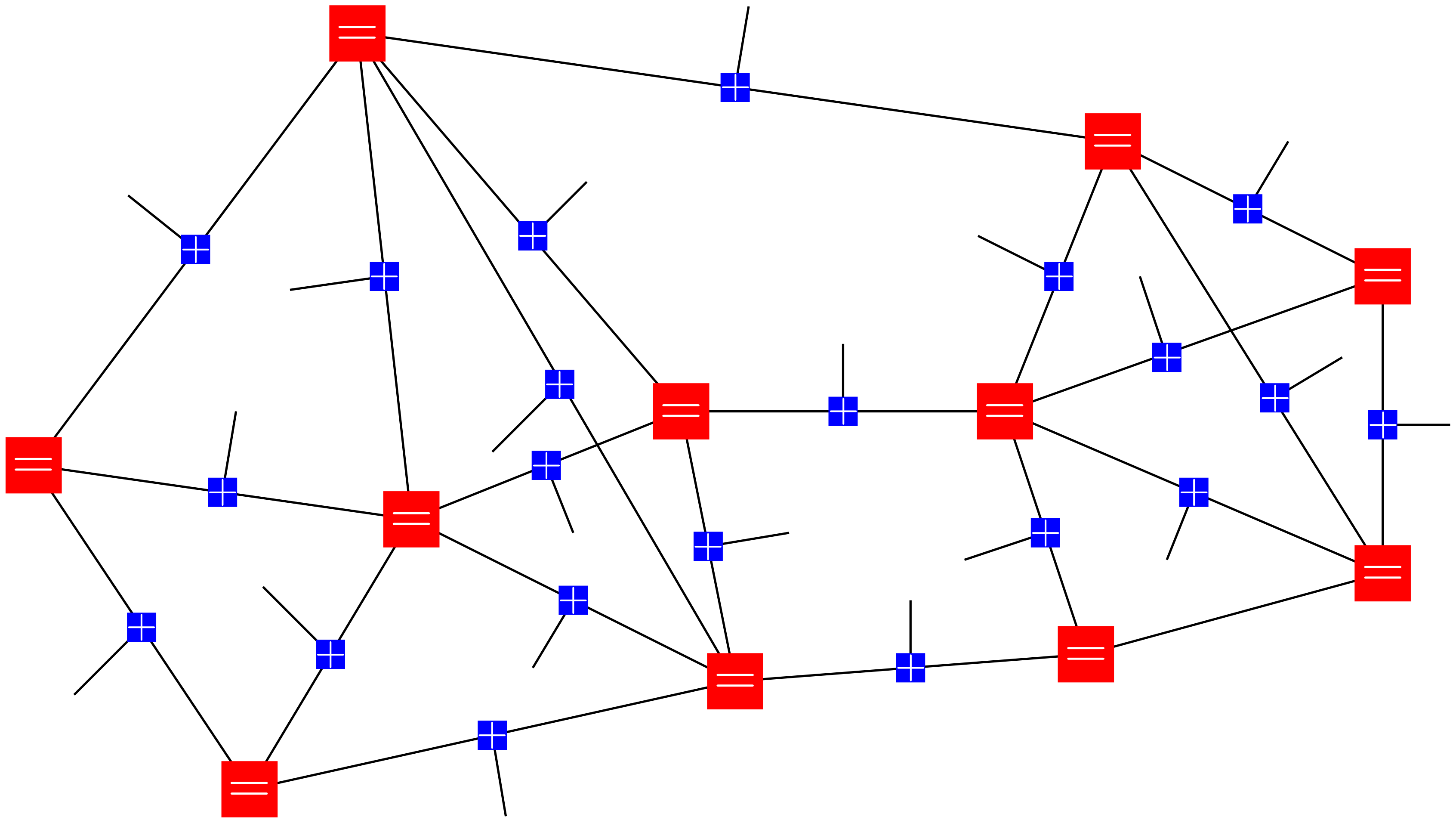
Community Detection in Networks



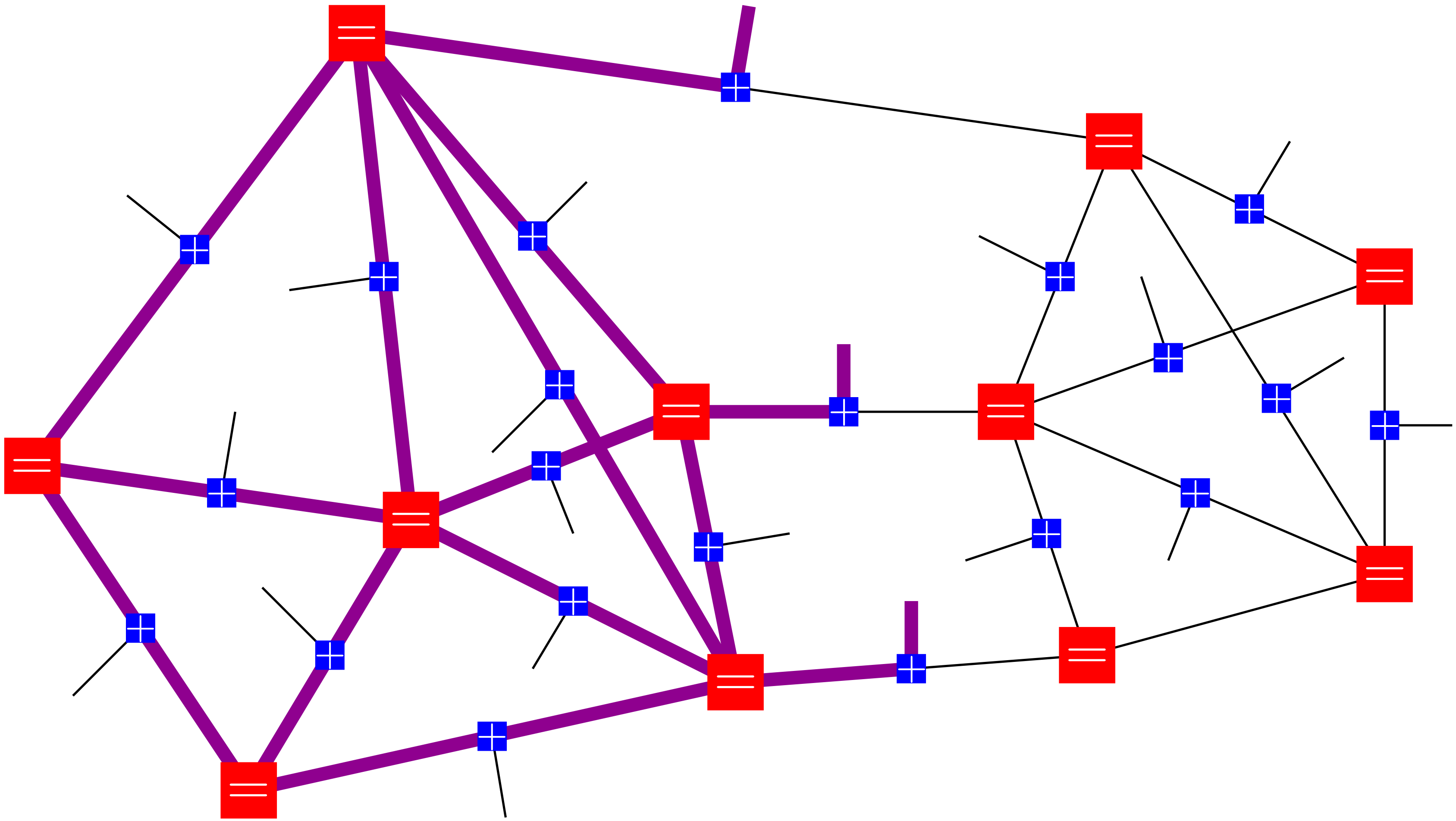
Community Detection in Networks



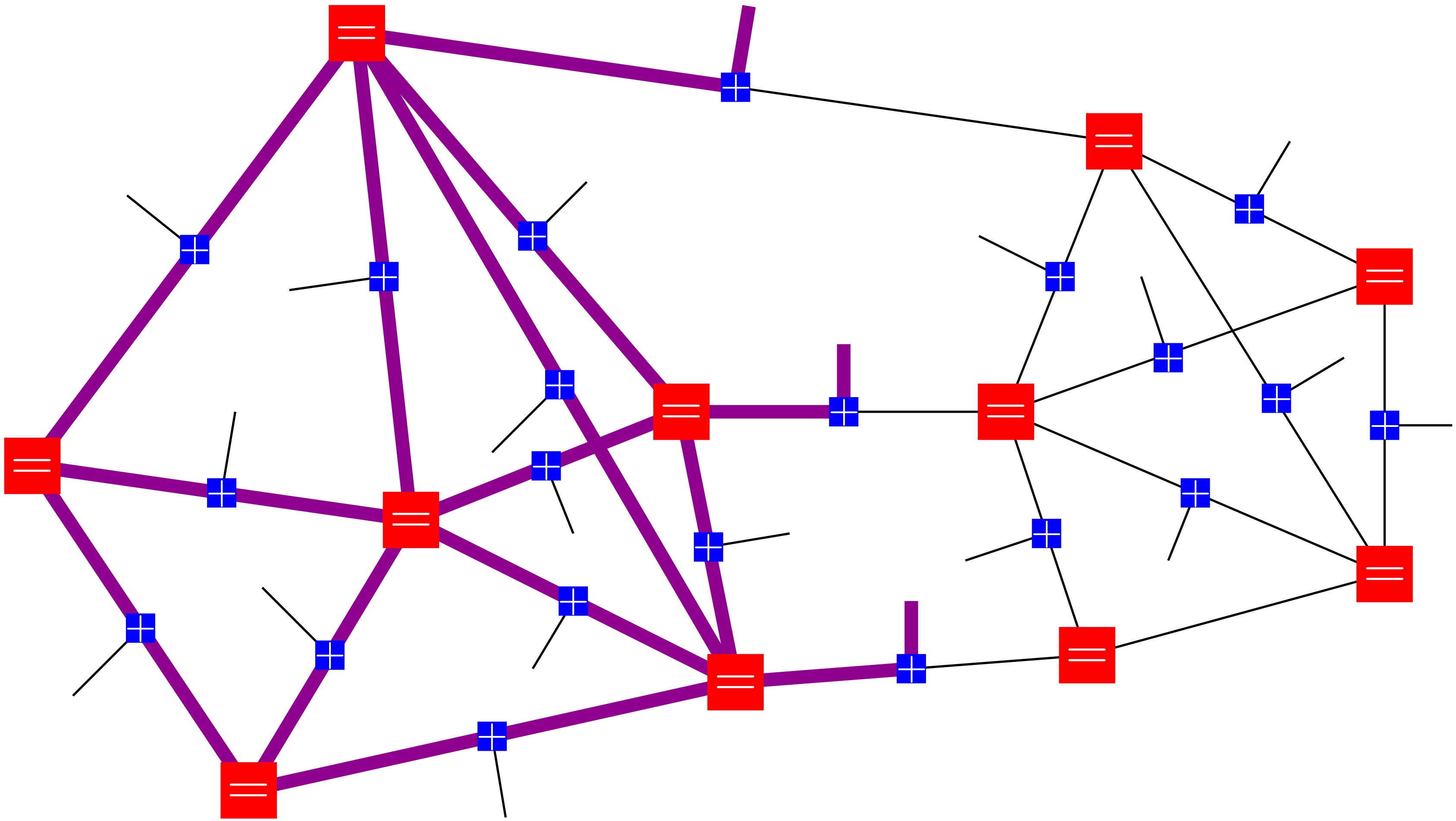
Community Detection in Networks



Community Detection in Networks



Community Detection in Networks

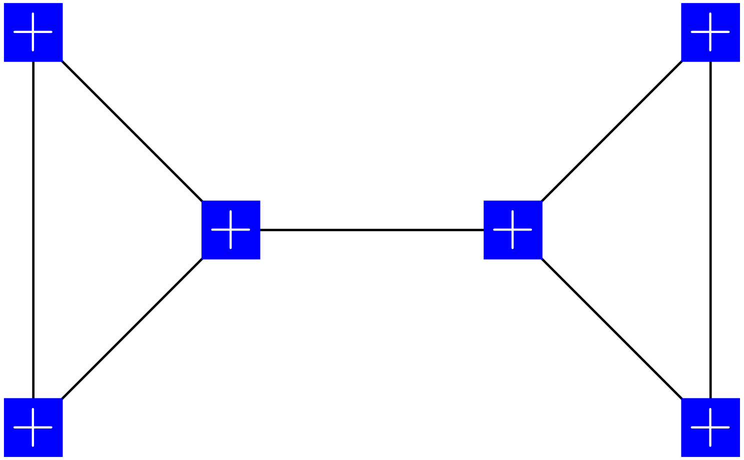


Community detection with the help of MPI algorithms:

[Krzakala, Moore, Mossel, Neeman, Sly, Zdeborová, Zhang, 2013]

Cycle Code NFG vs. Community Detection NFG

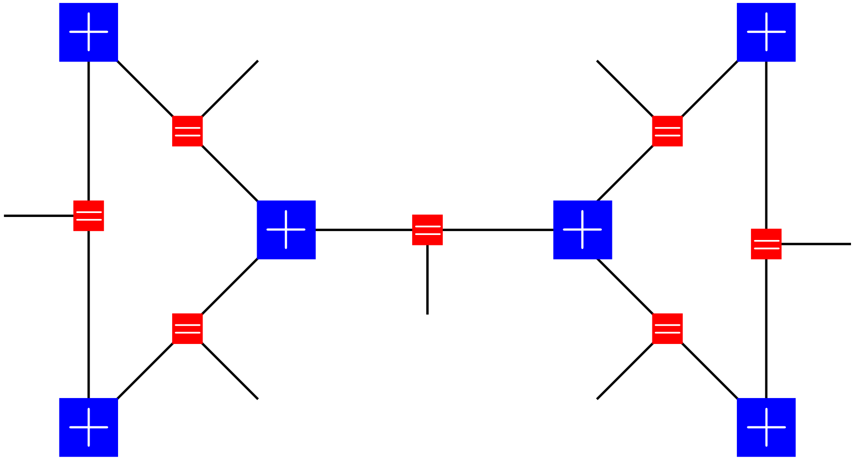
It turns out that the cycle code normal factor graph and the community detection normal factor graph are dual to each other in the sense that the sets of valid configurations are given by dual normal factor graphs, cf. NFG duality in [Forney, 2001].



cycle code
normal factor graph

Cycle Code NFG vs. Community Detection NFG

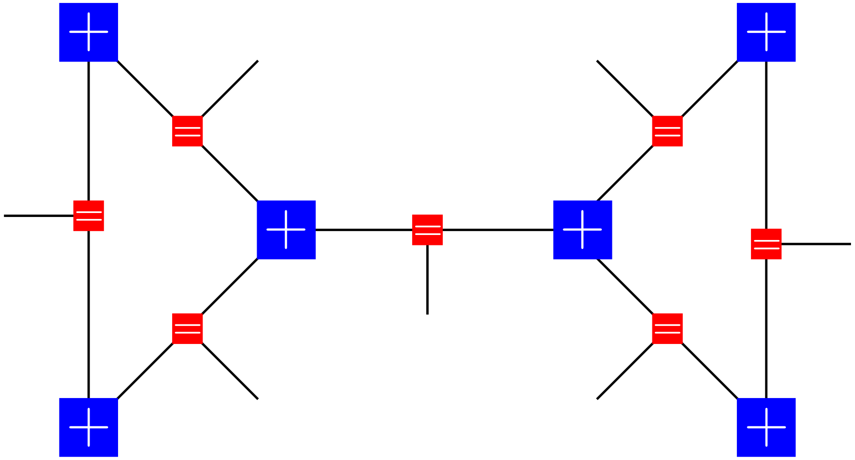
It turns out that the cycle code normal factor graph and the community detection normal factor graph are dual to each other in the sense that the sets of valid configurations are given by dual normal factor graphs, cf. NFG duality in [Forney, 2001].

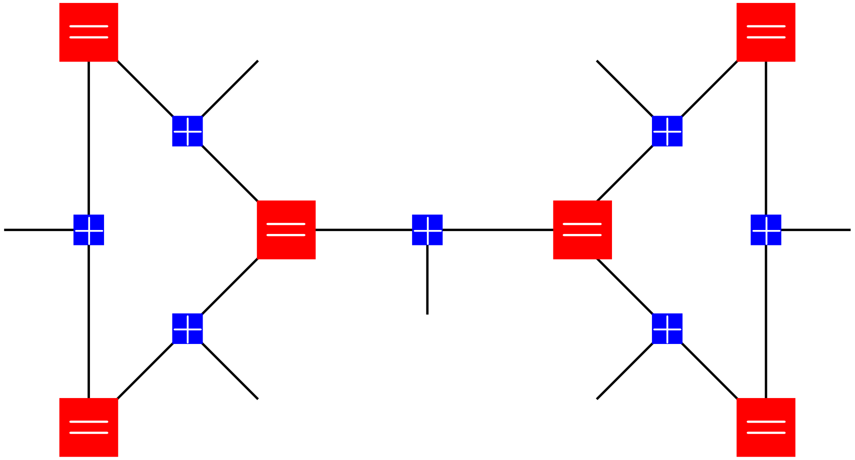


cycle code
normal factor graph

Cycle Code NFG
vs. Community Detection NFG

It turns out that the cycle code normal factor graph and the community detection normal factor graph are dual to each other in the sense that the sets of valid configurations are given by dual normal factor graphs, cf. NFG duality in [Forney, 2001].

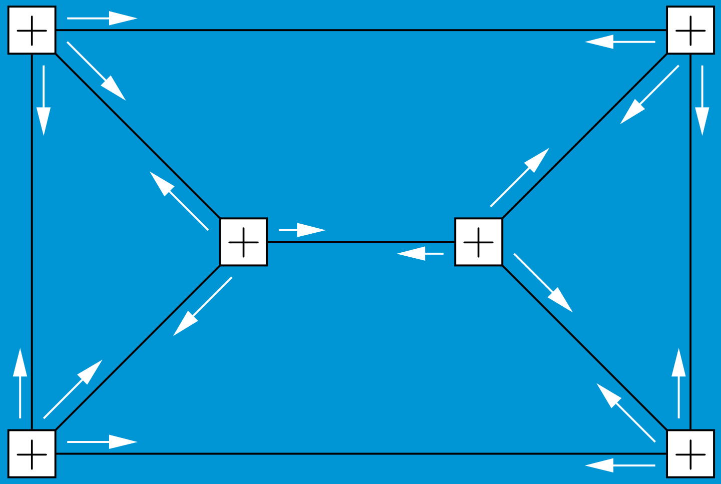


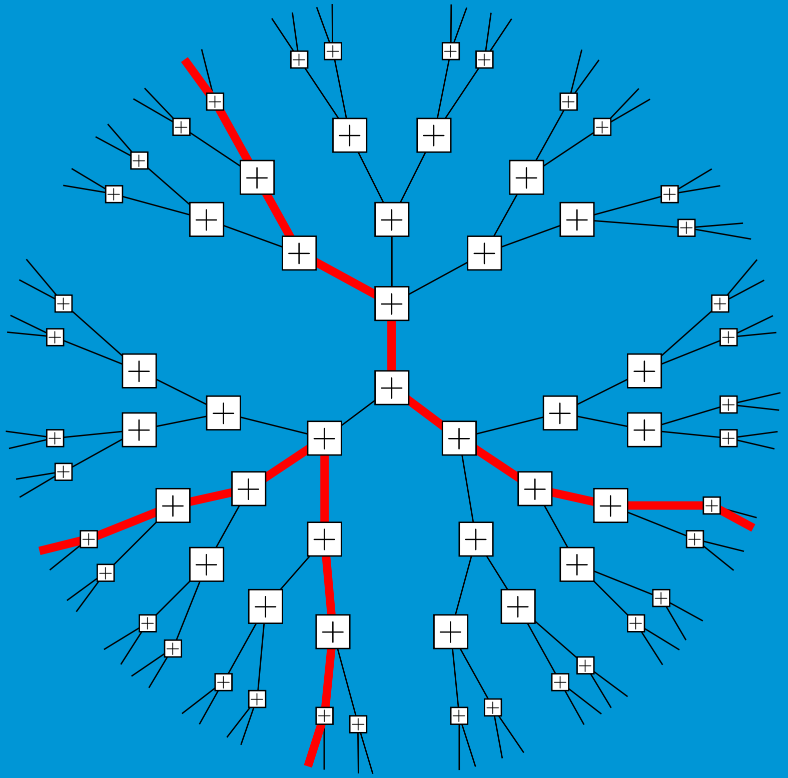


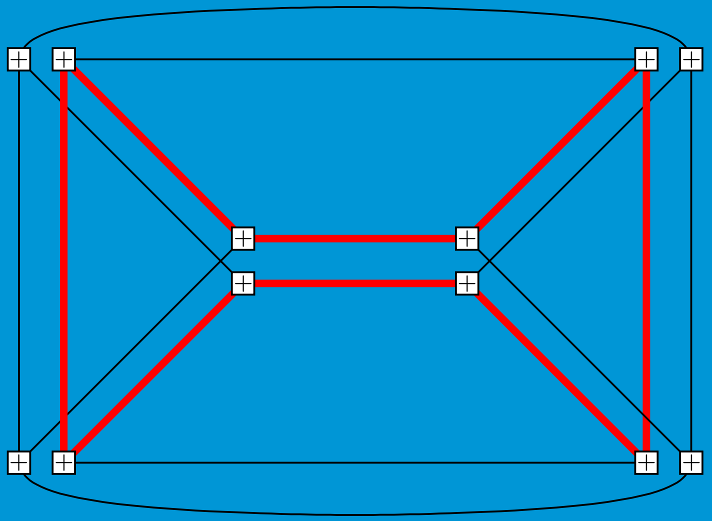
cycle code
normal factor graph

○—●

community detection
normal factor graph







$$\zeta(\mathsf{V}_1,\ldots,\mathsf{V}_n) \;=\; \sum_{\mathbf{k}} \zeta_{\mathbf{k}} \mathbf{V}^{\mathbf{k}} \;=\; \prod_{[\Gamma]} \frac{1}{1-g(\Gamma,\mathbf{V})}$$

What Can a Power Series Do For You?

Consider the power series $\zeta(\mathbf{V})$:

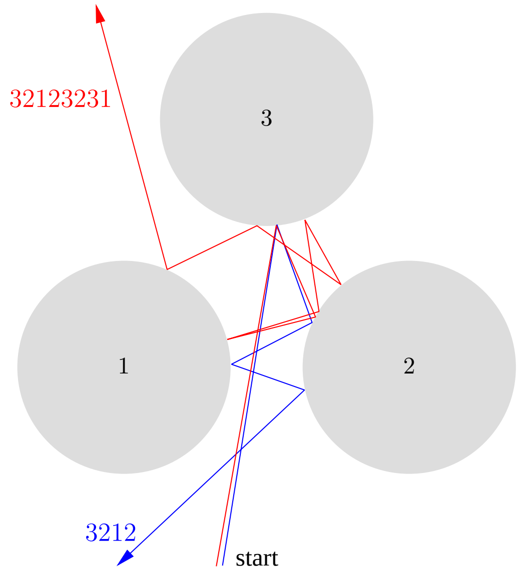
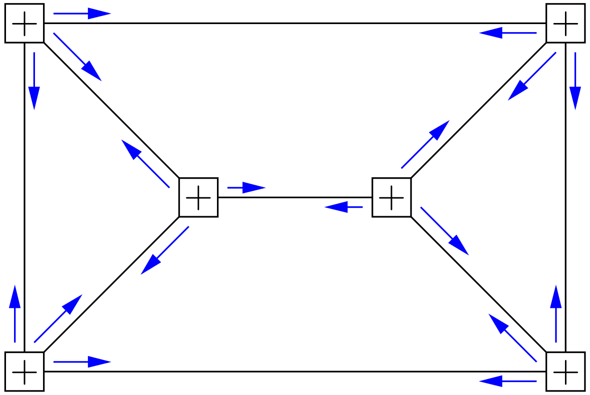
$$\zeta(\mathbf{V}) = \sum_{\mathbf{k}} \zeta_{\mathbf{k}} \mathbf{V}^{\mathbf{k}} = \prod_{[\Gamma]} \frac{1}{1 - g(\Gamma, \mathbf{V})}$$

We can obtain useful information from

- ...the expon. vecs. of $\zeta(\mathbf{V})$ [Koetter, Li, V., Walker, 2004/2007]
- ...the coefficients of $\zeta(\mathbf{V})$ [V., 2009/2010] [today]
- ... the evaluation of $\zeta(\mathbf{V})$ for some $\mathbf{V}$ [Watanabe, 2009/2010]
- ...the convergence region of $\zeta(\mathbf{V})$ [today]
- ...

Use of zeta functions for analyzing graphical models.

Analogy Pinball vs. MPI Decoding

 pinball	 message-passing iterative decoding of cycle codes
trajectory	minimial deviation in computation trees
periodic trajectory	codeword in finite graph cover / graph-cover pseudo-codeword
dynamical edge zeta function	graph zeta function

Pinball

- Ideally, we compute $\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \ldots$, and determine from this γ. However, usually this is too complicated.

- Note that the **power series**

$$\hat{\theta}(z) = \sum_{n=1}^{\infty} \hat{\theta}_n z^n$$

has **convergence radius** $\exp(\gamma)$.

- Alternative approach: set up a **new power series**

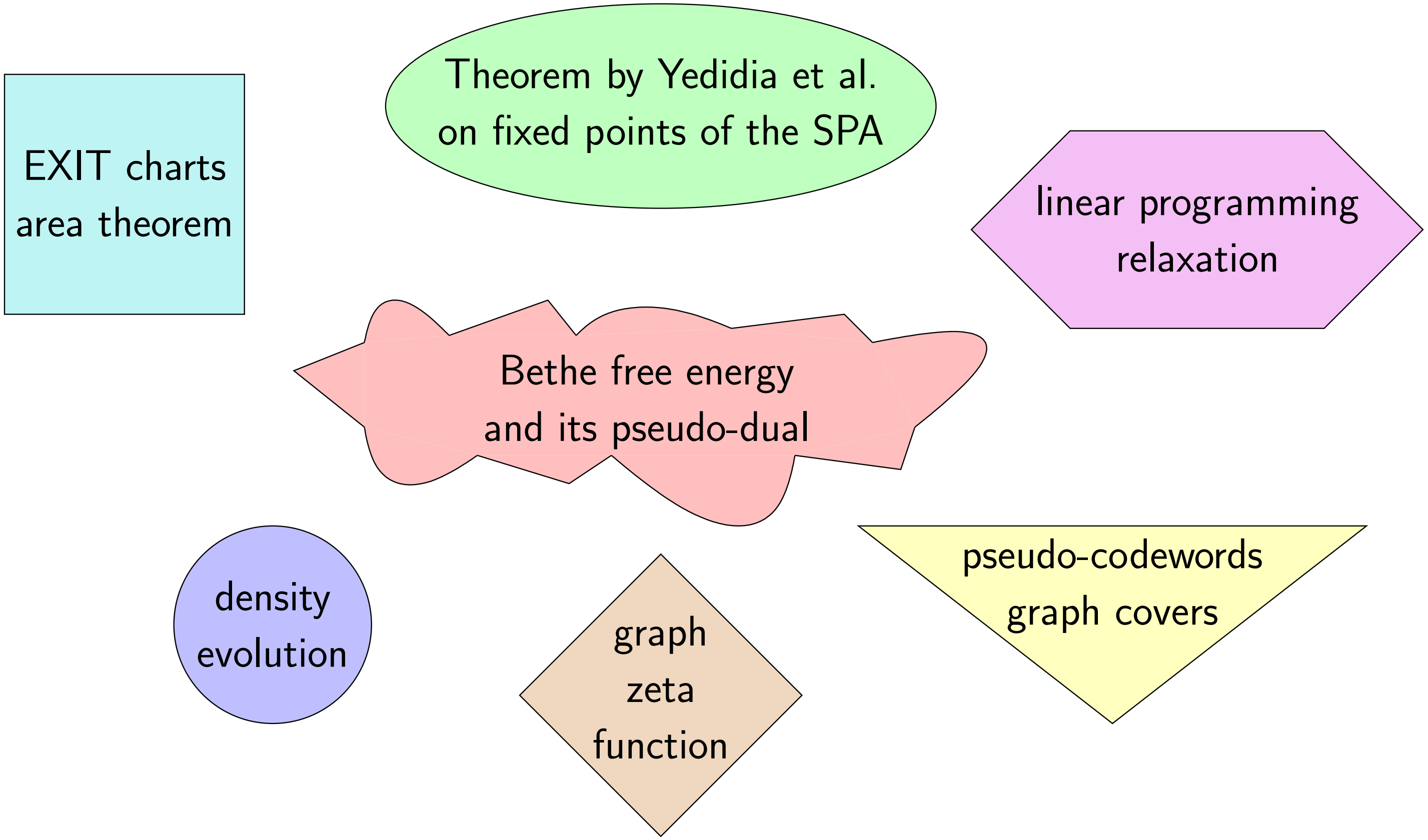
$$\theta(z) = \sum_{n=1}^{\infty} \theta_n z^n$$

so that its **convergence radius** equals $\exp(\gamma)$.

Bethe Free Energy Function

Some of the properties of the Bethe free energy function of the cycle code normal factor graph:

- The induced Bethe free energy function is a convex.
- The sum-product algorithm finds its minimum.



Comments

- We have some generalizations of the above results to general LDPC codes under attenuated SPA decoding.
- Note that [Watanabe, 2010] connects zeta function values to the Hessian of the Bethe free energy function for general factor graphs.
- Use of other concepts from chaos theory for understanding graphical models:
  - Agrawal and Vardy, "The turbo decoding algorithm and its phase trajectories," IEEE Trans. Inf. Theory, 2001.
  - Kocarev, Lehmann, Maggio, Scanavino, Tasev, and Vardy, "Nonlinear dynamics of iterative decoding systems: analysis and applications," IEEE Trans. Inf. Theory, 2006.
  - ...

References

H. D. Pfister and P. O. Vontobel, “On the relevance of graph covers and zeta functions for the analysis of SPA decoding of cycle codes,” Proc. ISIT 2013. (A longer version of this paper is in preparation.)

This work uses and extends results from:

- P. O. Vontobel, “Counting in graph covers: a combinatorial characterization of the Bethe entropy function,” IEEE Trans. Inf. Theory, vol. 59, no 9, pp. 6018–6048, Sep. 2013.
- P. O. Vontobel, “The Bethe permanent of a non-negative matrix,” IEEE Trans. Inf. Theory, vol. 59, no. 3, pp. 1866–1901, Mar. 2013.

Thank you!