Weighted Tiling Systems for Graphs

Evaluation Complexity

C. Aiswarya

Chennai Mathematical Institute
IRL ReLaX, CNRS

Paul Gastin

LSV, ENS Paris-Saclay, CNRS,
Université Paris-Saclay, France

WATA 2021

Outline

- Formalism - WTS
- WTS to model algorithmic problems on graphs
- Evaluation problem and complexity
- The case of Bounded tree-width graphs

Finite representations

Words

Sets	$\Sigma^* \to \{0,1\}$	NFA, Logic, …
Functions	$\Sigma^* \to S$	Weighted Automata, Weighted Logics [1]

Graphs

Sets	Graphs $\to \{0,1\}$	Graph acceptors, logic [2]
Functions	Graphs $\to S$	Weighted Tiling Automata, Weighted Logics [3]

1. Droste and Gastin, Weighted automata and weighted logics. TCS. 2007
2. Thomas, On logics, tilings and automata. ICALP. 1991
3. Droste, Dück, Weighted automata and logics on graphs. MFCS. 2015

Weighted Tiling Systems

node, edge labelled

labels come from a finite set

$$\textbf{WTS} : \mathrm{Graphs} \to S$$

Examples:

binary trees - lchild, rchild
grids - down, right

$$\begin{tikzcd}
1 \arrow[r] \arrow[d] & 1 \arrow[r] \arrow[d] & 1 \arrow[r] \arrow[d] & 1 \arrow[r] \arrow[d] & 1 \arrow[d] \\
0 \arrow[r] \arrow[d] & 1 \arrow[r] \arrow[d] & 0 \arrow[r] \arrow[d] & 0 \arrow[r] \arrow[d] & 0 \arrow[d] \\
0 \arrow[r] \arrow[d] & 0 \arrow[r] \arrow[d] & 1 \arrow[r] \arrow[d] & 1 \arrow[r] \arrow[d] & 1 \arrow[d] \\
0 \arrow[r] \arrow[d] & 1 \arrow[r] \arrow[d] & 1 \arrow[r] \arrow[d] & 0 \arrow[r] \arrow[d] & 0 \arrow[d] \\
0 \arrow[r] & 0 \arrow[r] & 0 \arrow[r] & 1 \arrow[r] & 0
\end{tikzcd}$$

Weighted Tiling Systems

node, edge labelled

labels come from a finite set

$$\textbf{WTS} : \mathrm{Graphs} \to S$$

Examples:

binary trees - lchild, rchild
grids - down, right

Weight domain

$$(S, \oplus, \otimes, \hat{0}, \hat{1})$$

Boolean = $(\{0,1\}, \vee, \wedge, 0, 1)$
Natural = $(\mathbb{N}, +, \times, 0, 1)$
Integer = $(\mathbb{Z}, +, \times, 0, 1)$
Rational = $(\mathbb{Q}, +, \times, 0, 1)$
max-plus-$\mathbb{N}$ = $(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$
max-plus-$\mathbb{Z}$ = $(\mathbb{Z} \cup \{-\infty\}, \max, +, -\infty, 0)$
min-plus-$\mathbb{N}$ = $(\mathbb{N} \cup \{+\infty\}, \min, +, +\infty, 0)$
min-plus-$\mathbb{Z}$ = $(\mathbb{Z} \cup \{+\infty\}, \min, +, +\infty, 0)$

Weighted Tiling Systems

a finite description to represent weight-functions on graphs

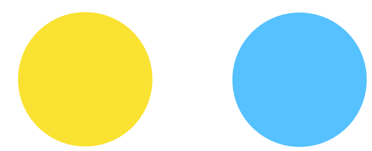
by means of tiling/coloring

Weighted Tiling Systems

a finite description to represent weight-functions on graphs

by means of tiling/coloring

finite set of colors/states

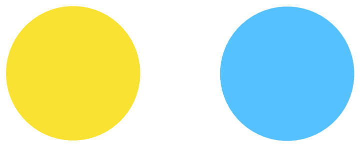


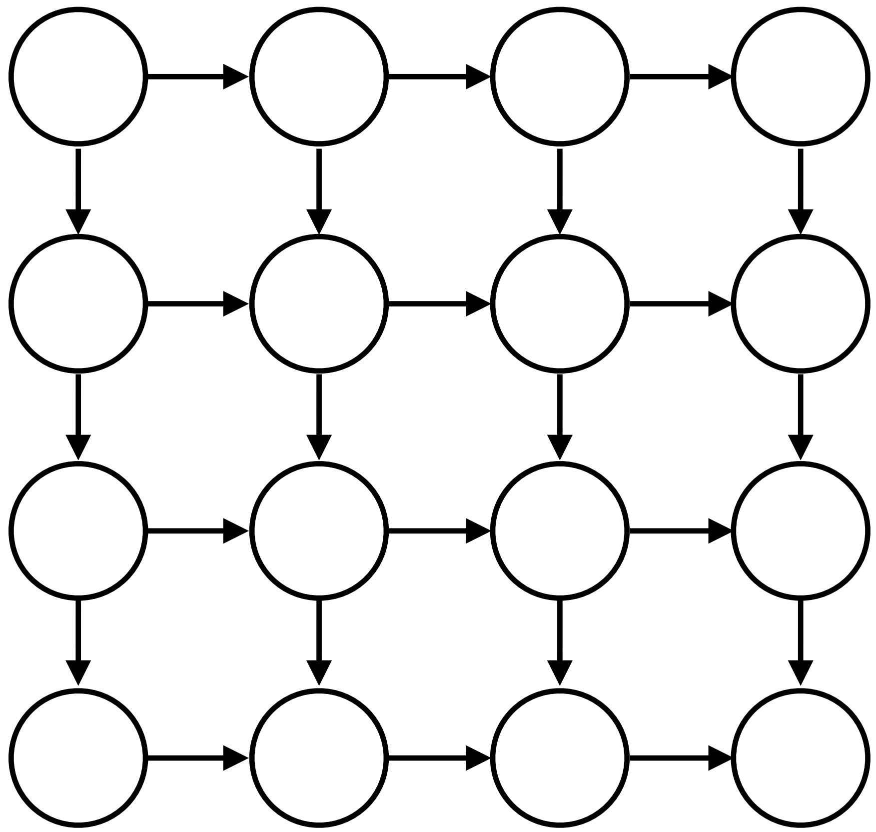
Weighted Tiling Systems

a finite description to represent weight-functions on graphs

by means of tiling/coloring

finite set of colors/states



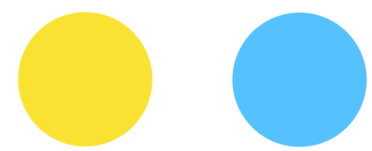


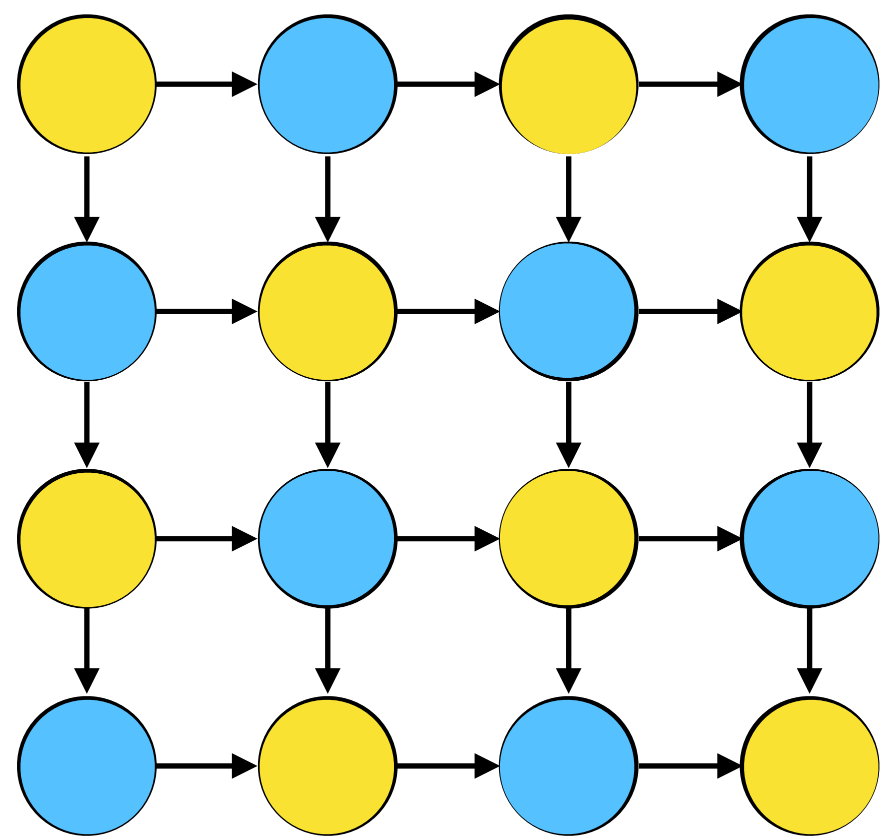
Weighted Tiling Systems

a finite description to represent weight-functions on graphs

by means of tiling/coloring

finite set of colors/states

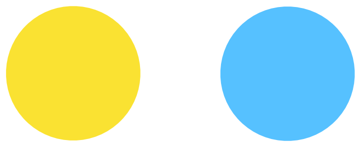


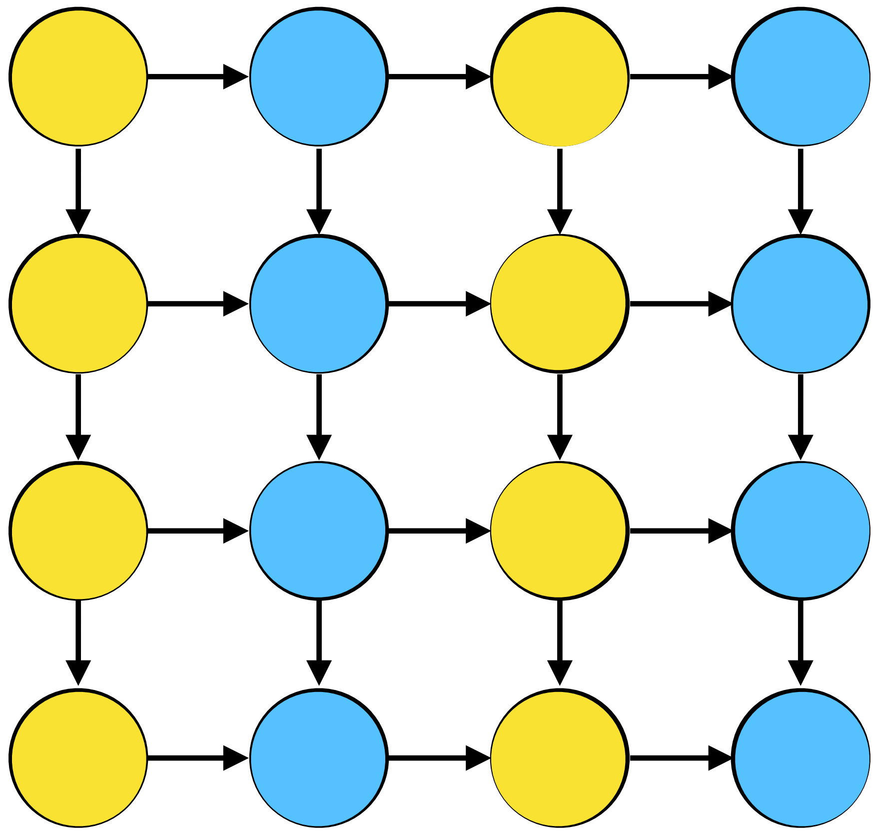


Weighted Tiling Systems

a finite description to represent weight-functions on graphs

by means of tiling/coloring

finite set of colors/states 

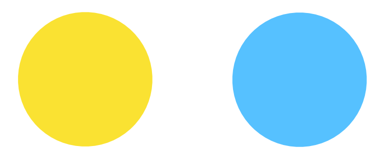


Weighted Tiling Systems

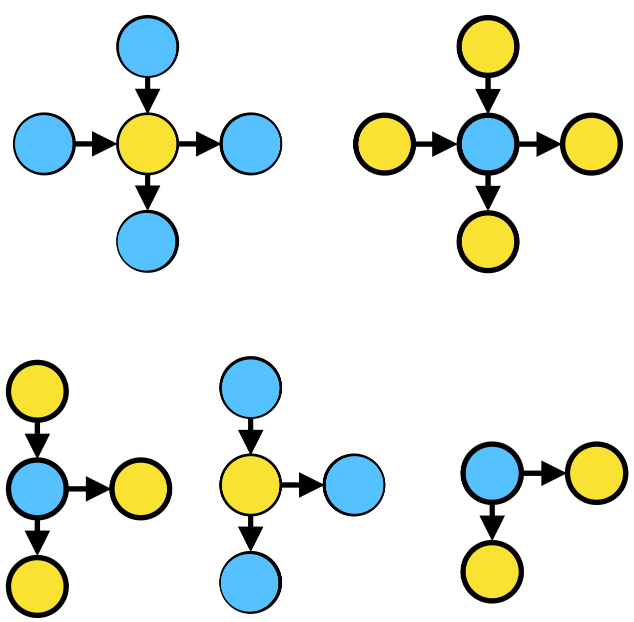
a finite description to represent weight-functions on graphs

by means of tiling/coloring

finite set of colors/states



permitted tiles



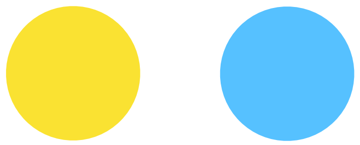
...

Weighted Tiling Systems

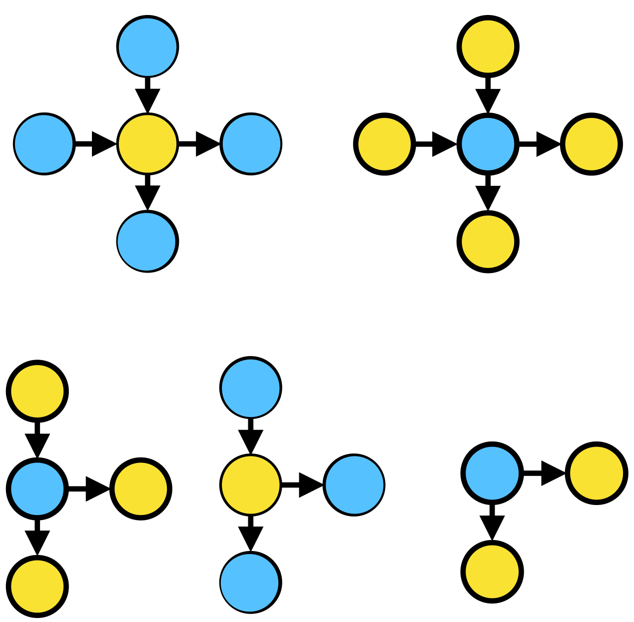
a finite description to represent weight-functions on graphs

by means of tiling/coloring

finite set of colors/states



permitted tiles + weights

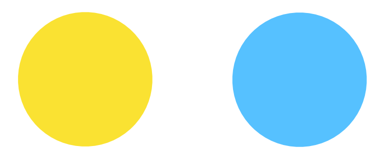


...

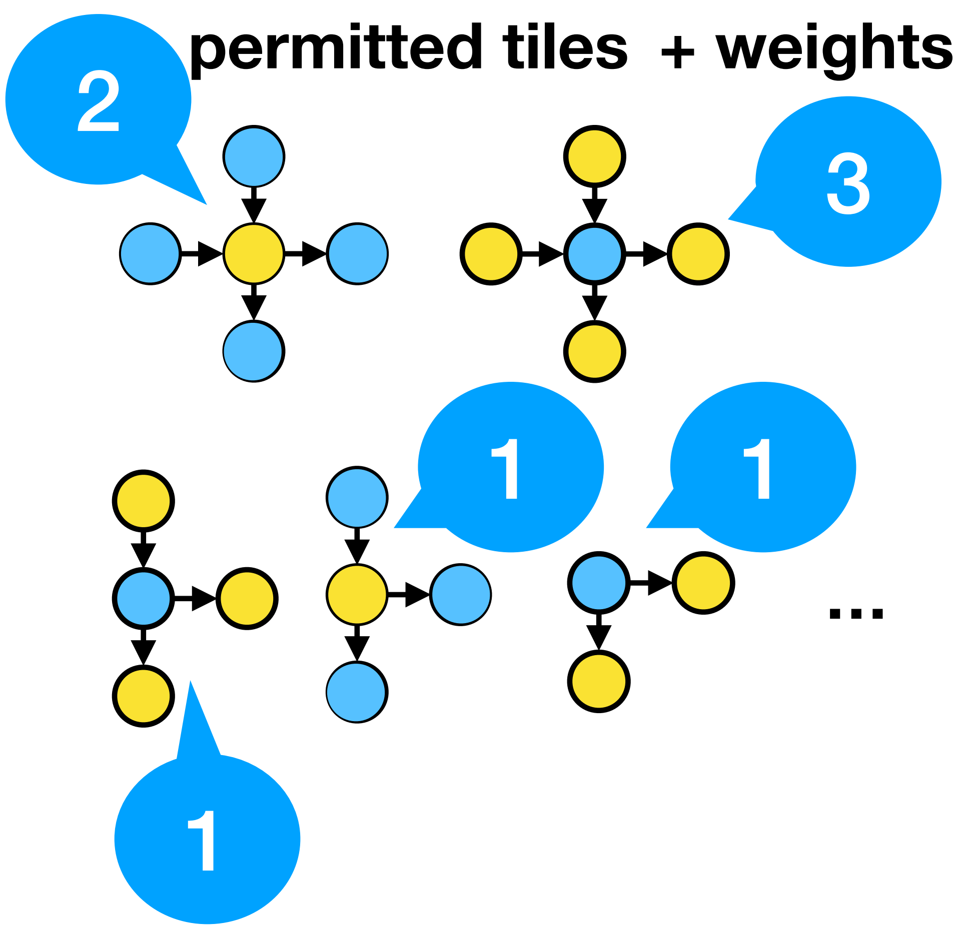
Weighted Tiling Systems

a finite description to represent weight-functions on graphs

by means of tiling/coloring

finite set of colors/states 

permitted tiles + weights



Weighted Tiling Systems

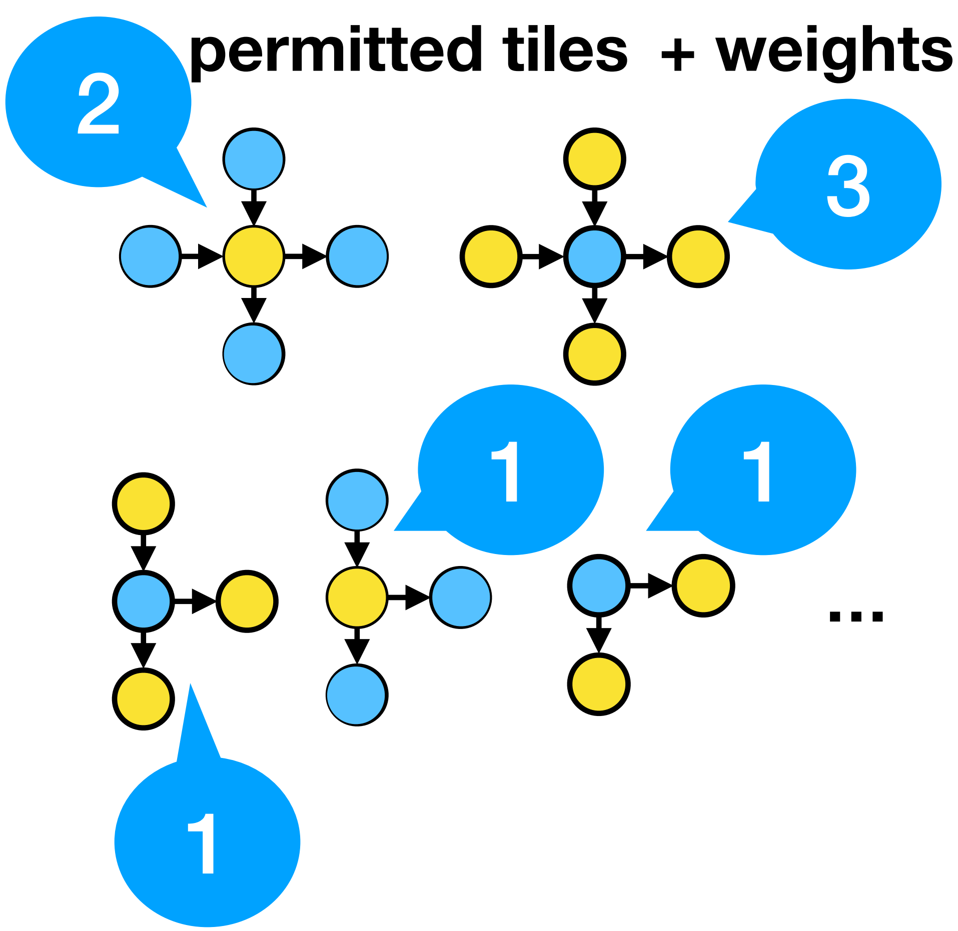
Weight of a coloring / run

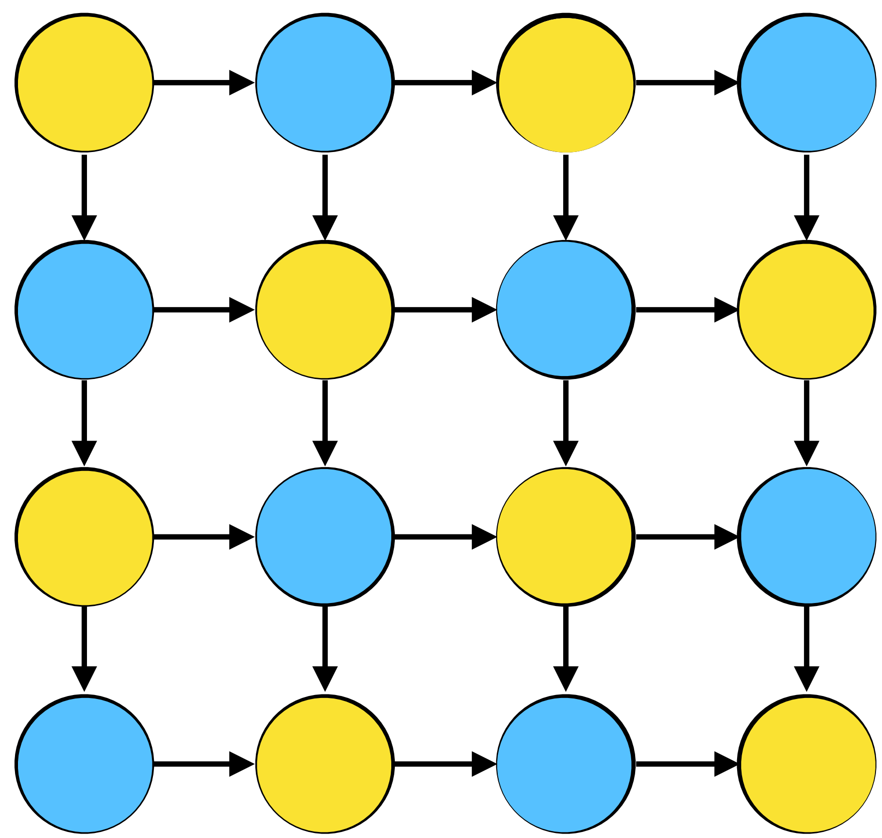
Product of weights of the tiles

Weighted Tiling Systems

Weight of a coloring / run

Product of weights of the tiles

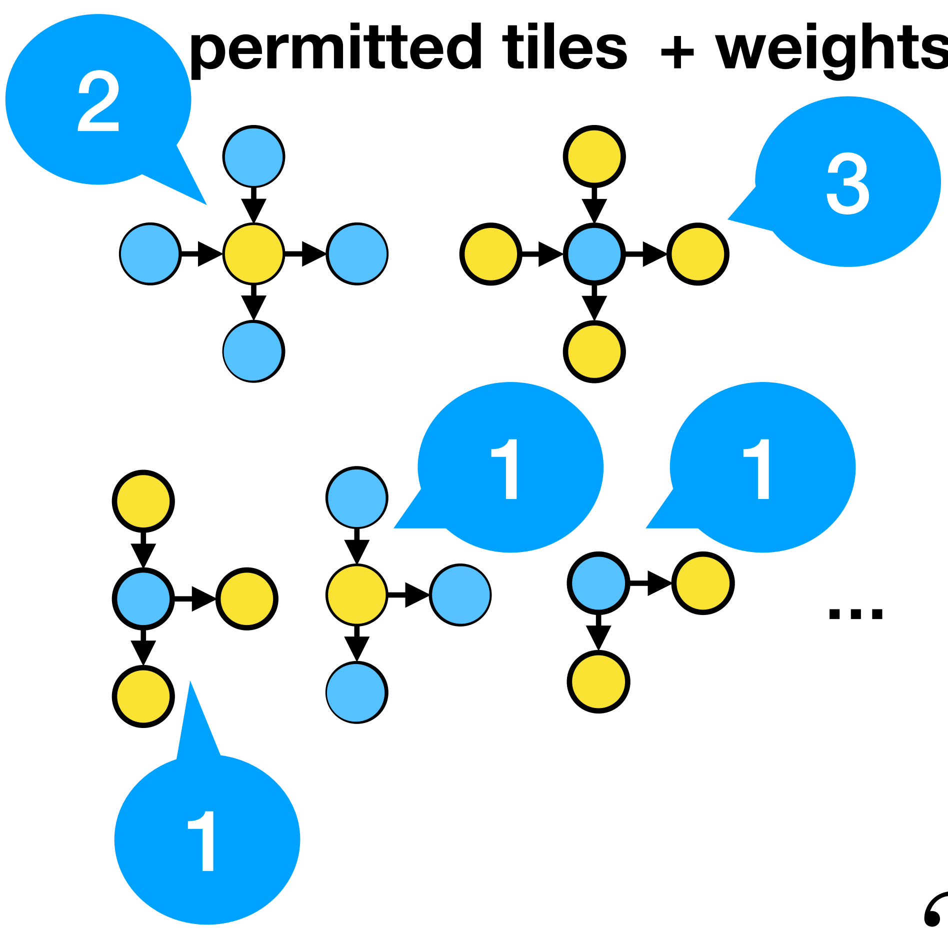


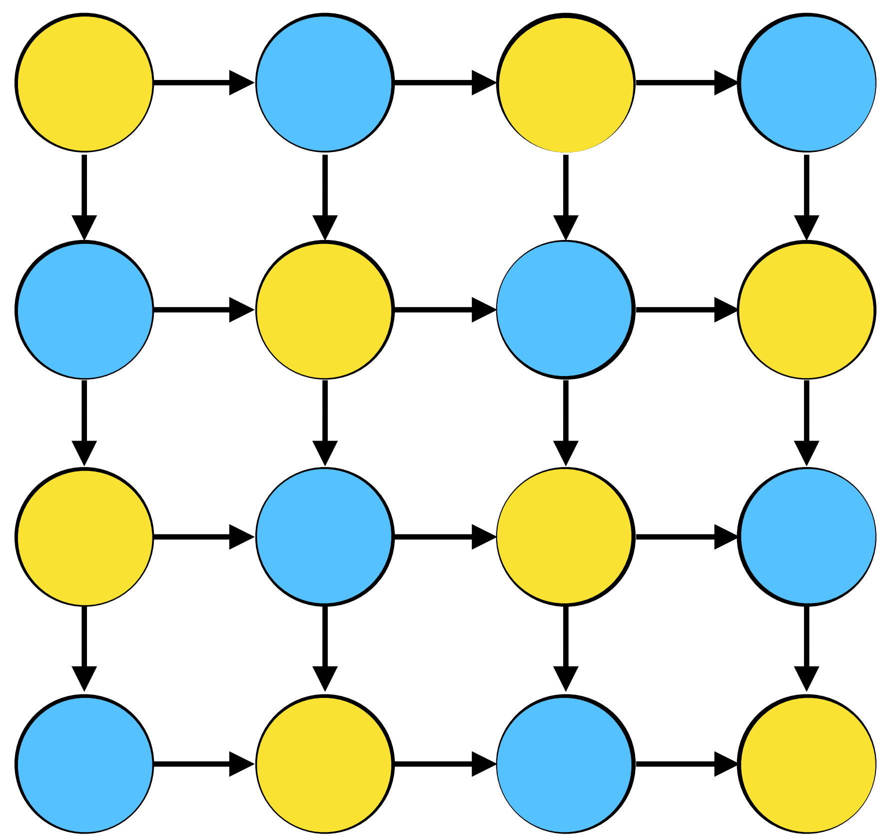


Weighted Tiling Systems

Weight of a coloring / run

Product of weights of the tiles





$$2^2 \times 3^2 \times 1^1 \times 1^1 \times \ldots$$

Weighted Tiling Systems

Weight of a coloring / run

Product of weights of the tiles

Weight assigned to a graph

Sum of the weights of all colourings

Outline

✓ Formalism - WTS

- WTS to model algorithmic problems on graphs
- Evaluation problem and complexity
- The case of Bounded tree-width graphs

Example: Computing the permanent

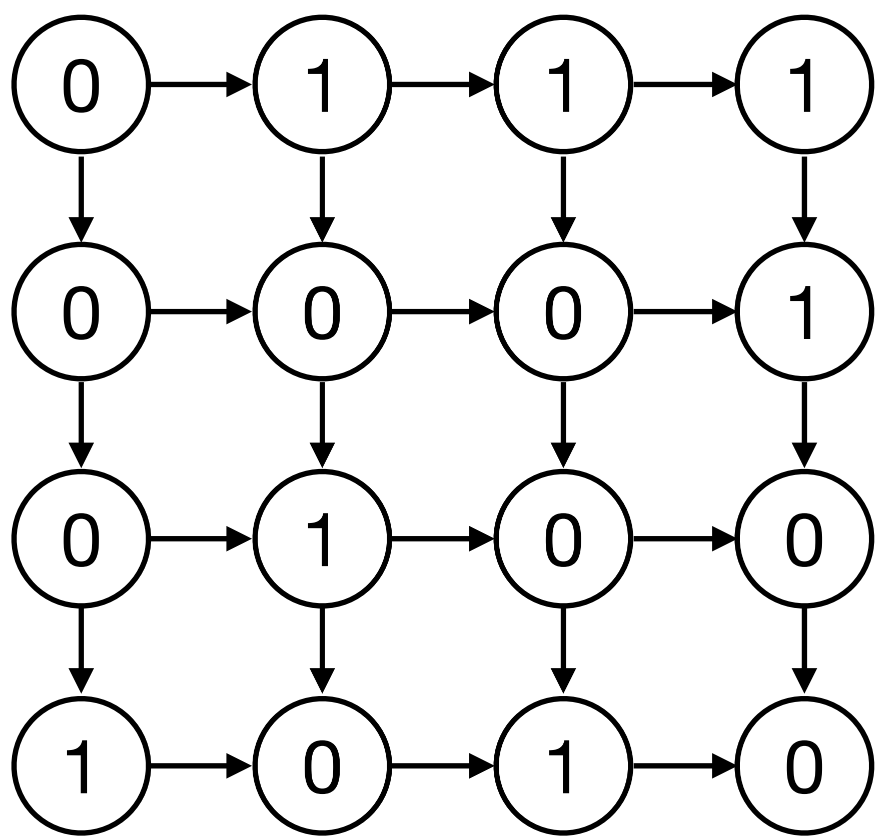
$n \times n$ matrix $A = (a_{i,j})$

$$\mathsf{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)}$$

Example: Computing the permanent

$n \times n$ matrix $A = (a_{i,j})$

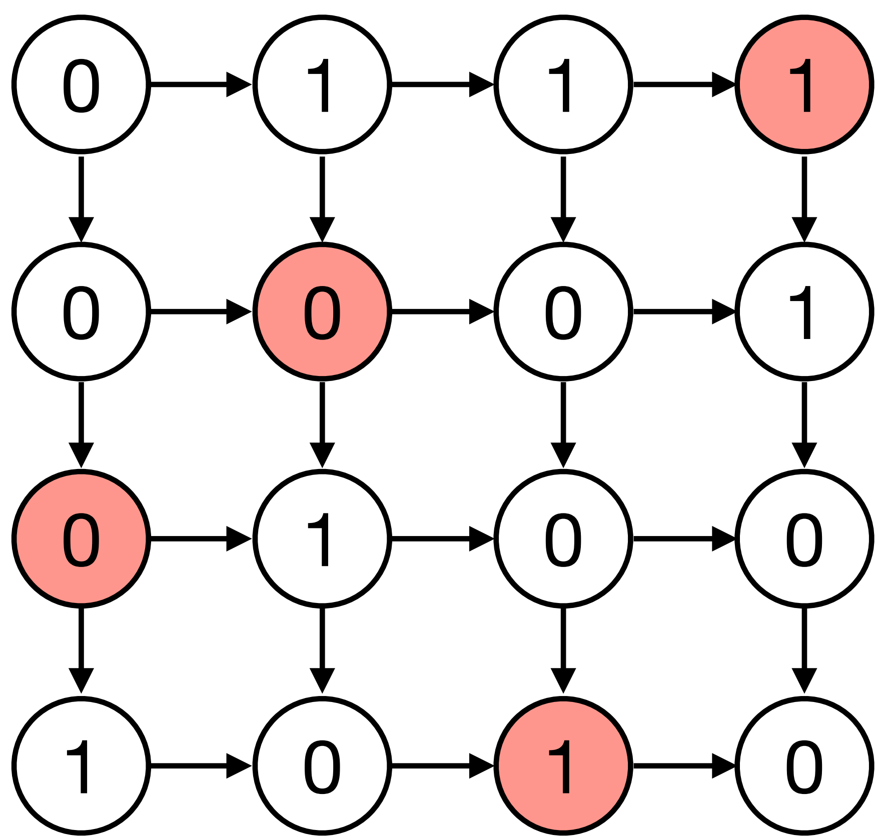
$$\mathsf{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)}$$



Example: Computing the permanent

$n \times n$ matrix $A = (a_{i,j})$

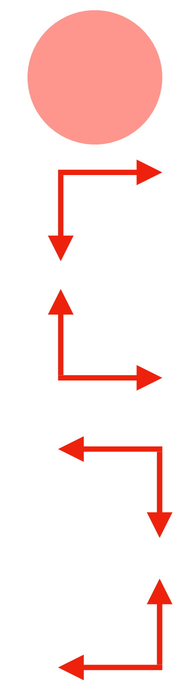
$$\mathsf{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)}$$

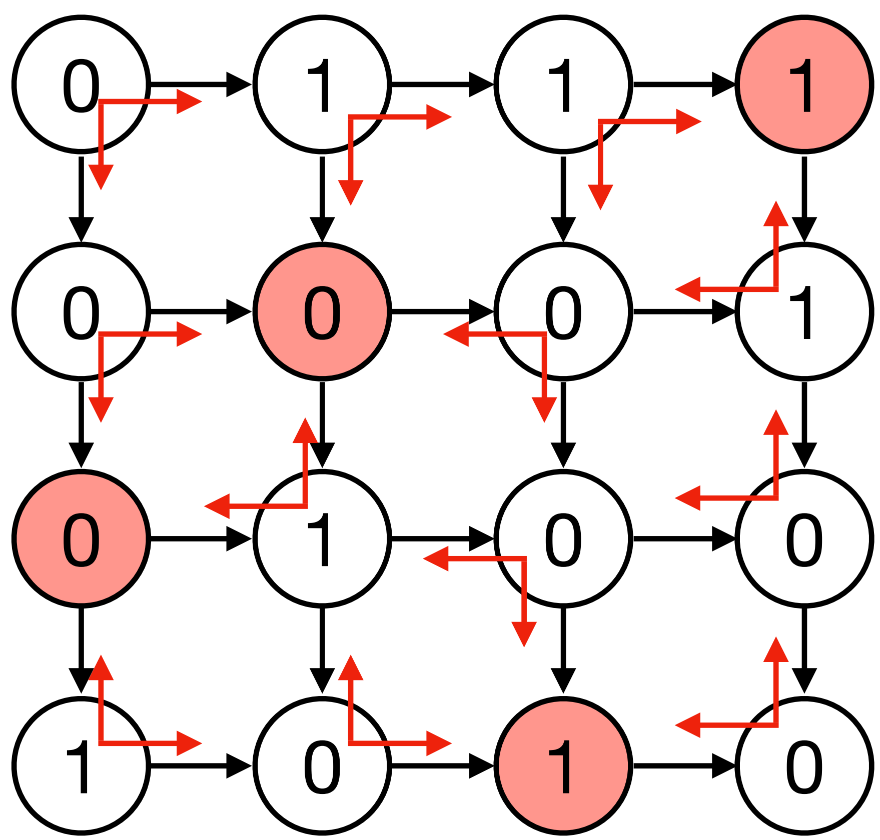


Example: Computing the permanent

$n \times n$ matrix $A = (a_{i,j})$

$$\mathsf{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)}$$

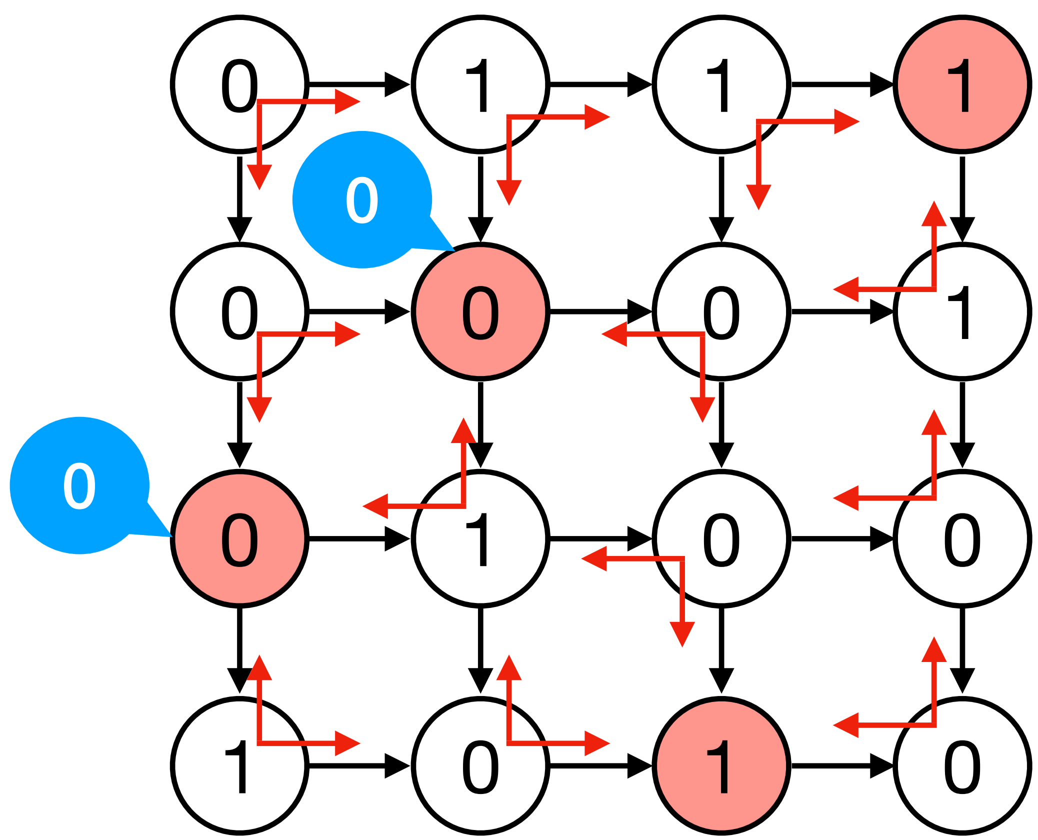


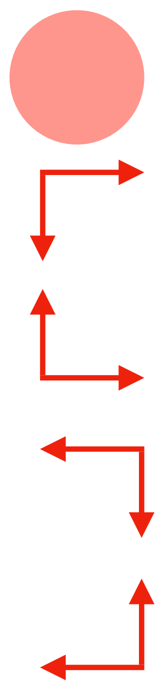


Example: Computing the permanent

$n \times n$ matrix $A = (a_{i,j})$

$$\mathsf{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)}$$

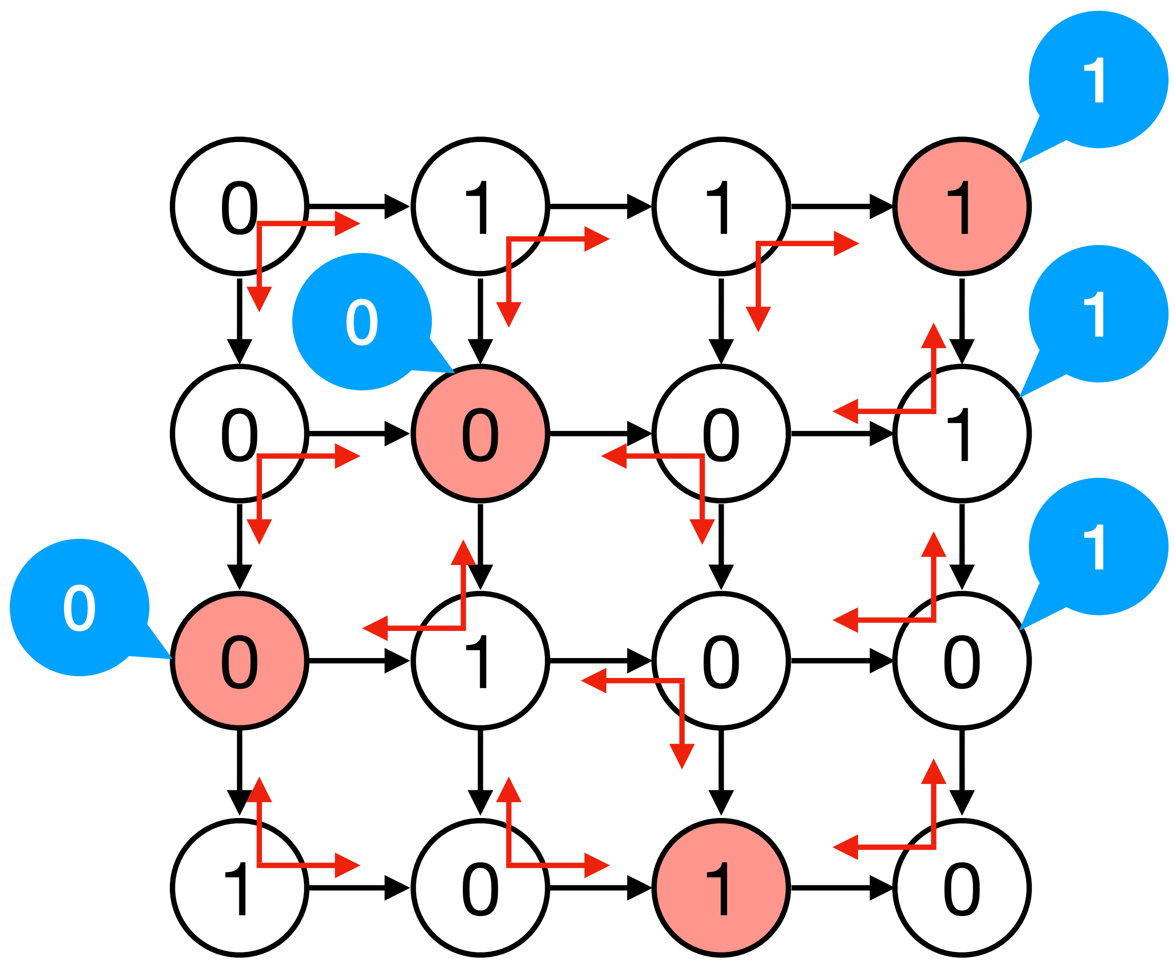


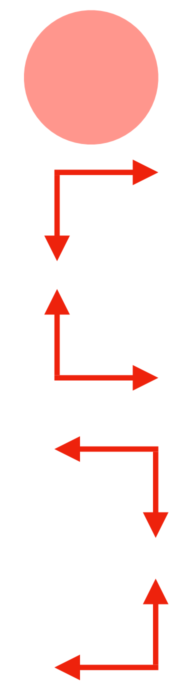


Example: Computing the permanent

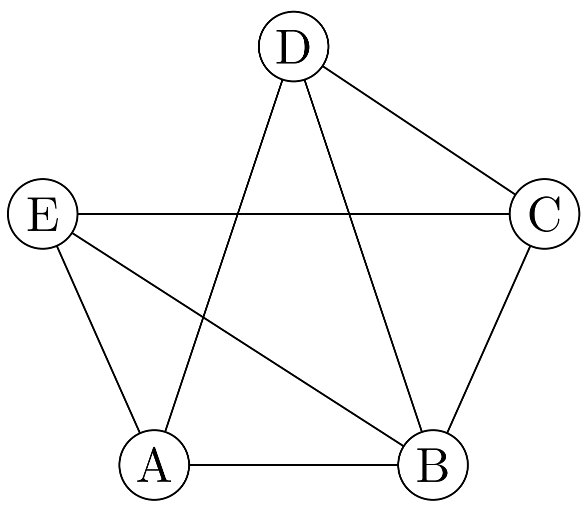
$n \times n$ matrix $A = (a_{i,j})$

$$\mathsf{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)}$$



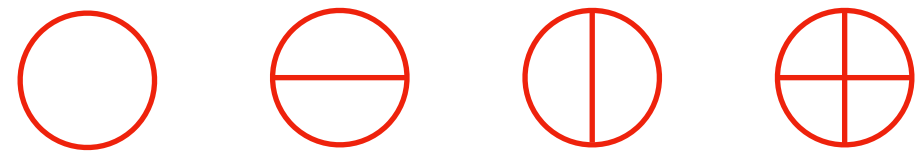


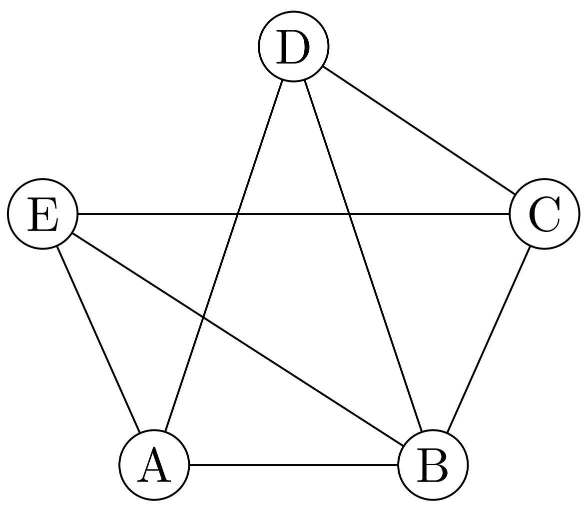
Example: Computing the clique number



$$
\begin{tikzcd}
 & & & & \mathrm{E} & \\
 & & & & & 1 \arrow[dd] \\
 & & & \mathrm{D} & & \\
 & & & & 1 \arrow[r] \arrow[dd] & 0 \arrow[dd] \\
 & & \mathrm{C} & & & \\
 & & & 1 \arrow[r] \arrow[dd] & 1 \arrow[r] \arrow[dd] & 1 \arrow[dd] \\
 & \mathrm{B} & & & & \\
 & & 1 \arrow[r] \arrow[dd] & 1 \arrow[r] \arrow[dd] & 1 \arrow[r] \arrow[dd] & 1 \arrow[dd] \\
\mathrm{A} & & & & & \\
 & 1 \arrow[r] & 1 \arrow[r] & 0 \arrow[r] & 1 \arrow[r] & 1
\end{tikzcd}
$$

Example: Computing the clique number

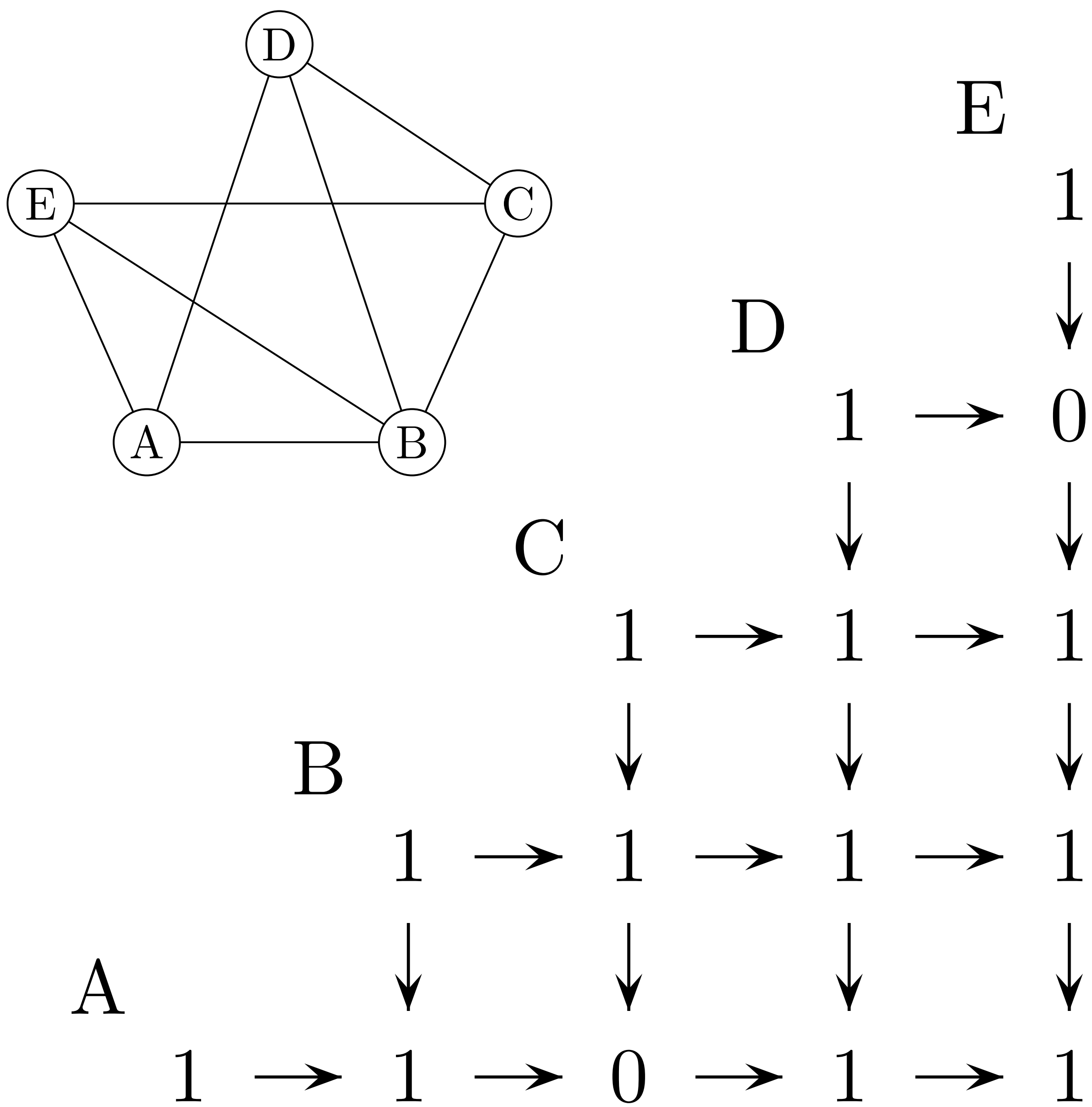


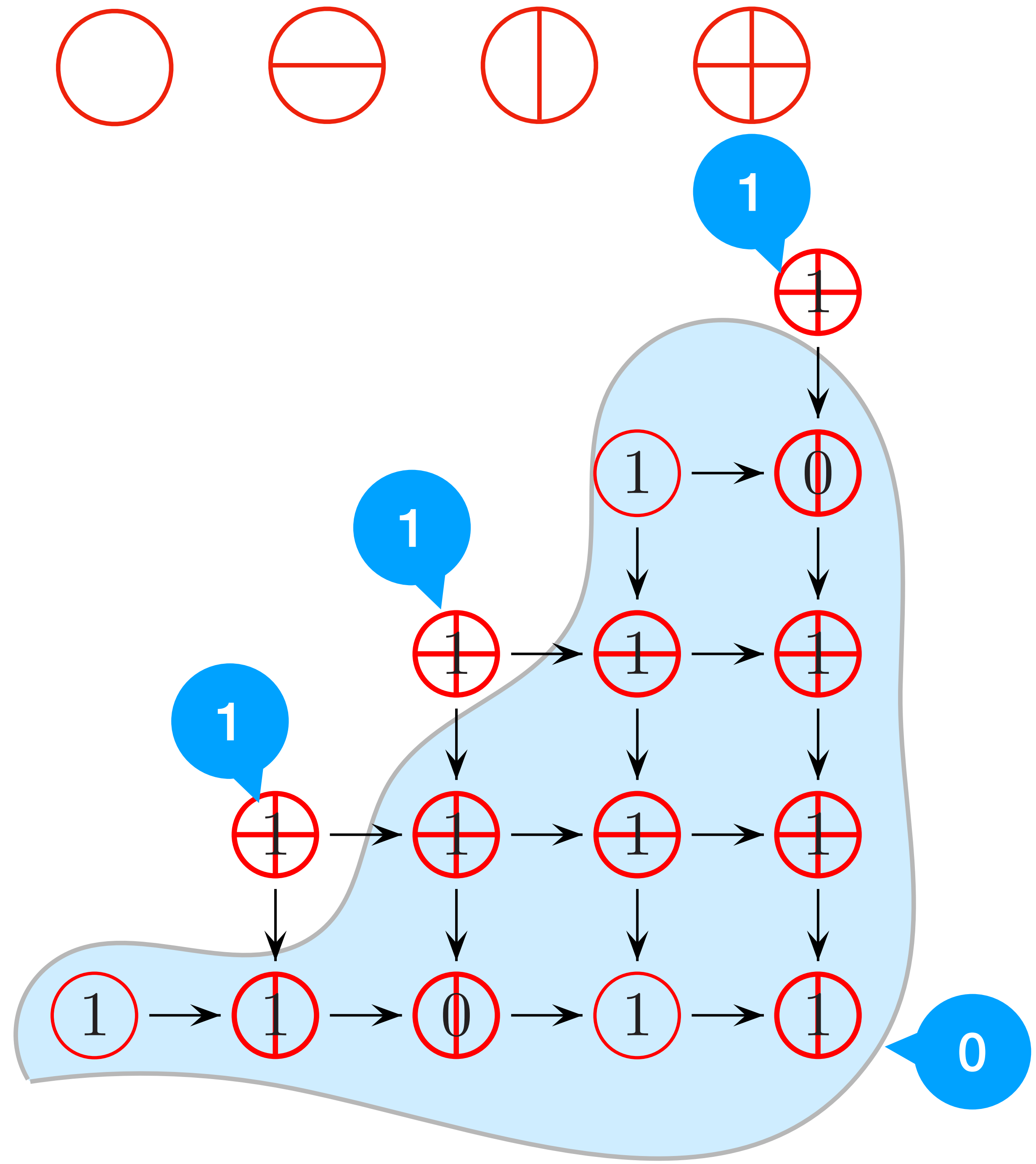


$$\begin{tikzcd}
 & & & & \mathrm{E} & 1 \arrow[d] \\
 & & & \mathrm{D} & 1 \arrow[r] \arrow[d] & 0 \arrow[d] \\
 & & \mathrm{C} & 1 \arrow[r] \arrow[d] & 1 \arrow[r] \arrow[d] & 1 \arrow[d] \\
 & \mathrm{B} & 1 \arrow[r] \arrow[d] & 1 \arrow[r] \arrow[d] & 1 \arrow[r] \arrow[d] & 1 \arrow[d] \\
 \mathrm{A} & 1 \arrow[r] & 1 \arrow[r] & 0 \arrow[r] & 1 \arrow[r] & 1
\end{tikzcd}$$

max-plus-$\mathbb{N}$ = $(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$

Example: Computing the clique number





max-plus-$\mathbb{N} = (\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$

Outline

- ✓ Formalism - WTS
- ✓ WTS to model algorithmic problems on graphs
- Evaluation problem and complexity
- The case of Bounded tree-width graphs

Evaluation Problem

Input : a WTS, a Graph

Output : Weight assigned to the Graph

Evaluation Problem

Input : a WTS, a Graph

Output : Weight assigned to the Graph

Theorem

Evaluation problem can be solved in Polynomial space.

Evaluation Problem : (+, x) -semiring

Input : a WTS, a Graph

Output : Weight assigned to the Graph

Theorem

Evaluation problem is #P-complete.

A function is #P, if there is a polynomial time TM such that the value on x equals the number of accepting paths of TM on input x.

Weights given in Binary

#P- upper bound:

- A poly time NTM will guess a colouring
- Compute and write in binary the weight of this tiling in a different tape
- Make as many accepting paths as the computed weight.

A function is #P, if there is a polynomial time TM such that the value on x equals the number of accepting paths of TM on input x.

#P- lower bound:

- Computing permanent is #P hard.

A function is #P, if there is a polynomial time TM such that the value on x equals the number of accepting paths of TM on input x.

Evaluation Problem : (max, +) -semiring

Input : a WTS, a Graph

Output : Weight assigned to the Graph

Theorem

Evaluation problem is $P^{NP[\log]}$-complete.

$P^{NP[\log]}$: makes logarithmically many oracle queries to an NP machine.

Weights given in Unary

$P^{NP[log]}$ - completeness:

- Computing clique number is $P^{NP[log]}$ hard.
- For the upper bound, do a binary search in the possible weights space, each time calling an NP machine to check if there is a colouring with at least that weight.

$P^{NP[log]}$: makes logarithmically many oracle queries to an NP machine.

Evaluation complexity: Bounded Tree-width

Input : a WTS, a Graph

Output : Weight assigned to the Graph

Theorem

Evaluation Problem can be solved in linear time wrt. Graph and polynomial time wrt. WTS.

- Get a tree decomposition using Bodlander’s algorithm
- Extract a tree term
- Construct a weighted tree automaton from the given weighted automata that runs on tree terms
    - Guess the colour for each node as it is added
    - Maintain the tile for every “active” node
    - Weight of a tile given when a node becomes “inactive”
    - Weight of all other transitions is 1.
- Evaluate the tree automaton on the tree term

Graphs with Bounded Tree-width

- Bounded tree-width captures many weighted systems studied in the literature.
    - Weighted Mazurkiewics traces [1]
    - Weighted pushdown systems, nested words [2]
    - Under-approximations of multi pushdown systems
    - Under-approximations of message passing systems
- Uniform evaluation procedure

1. Bollig and Meinecke, Weighted distributed systems and their logics. LFCS. 2007
2. Mathissen, Weighted logics for nested words and algebraic formal power series. LMCS. 2010

Conclusions

✓ Formalism - WTS

✓ WTS to model algorithmic problems on graphs

✓ Evaluation problem and complexity

✓ The case of Bounded tree-width graphs