Julius-Maximilians-UNIVERSITÄT WÜRZBURG

Chair for INFORMATICS I
Efficient Algorithms and Knowledge-Based Systems

Institute for Informatics

The Partial Visibility Representation Extension Problem

Steven Chaplick*, Grzegorz Guśpiel†, Grzegorz Gutowski†
Tomasz Krawczyk†, Giuseppe Liotta+.

∗ Julius-Maximilians-Universität Würzburg, Germany

† Theoretical Computer Science Department, Jagiellonian University, Poland.

+ Dipartimento di Ingegneria, Università degli Studi di Perugia, Italy.

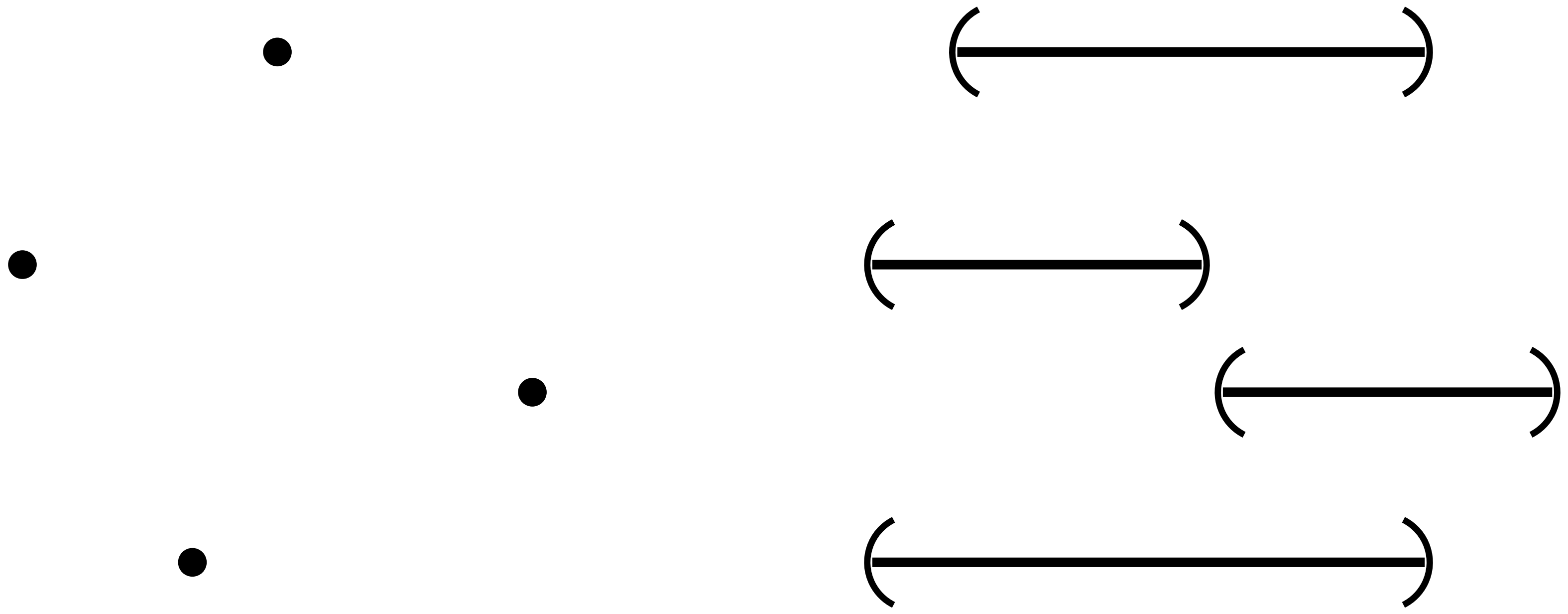
Symposium on Graph Drawing and Network Visualization (GD 2016)

Bar Visibility Representations

Vertices correspond to horizontal open line segments (bars)

Bar Visibility Representations

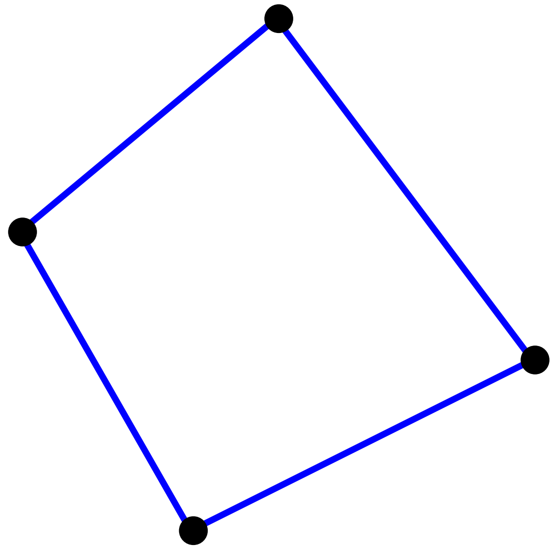
Vertices correspond to horizontal open line segments (bars)

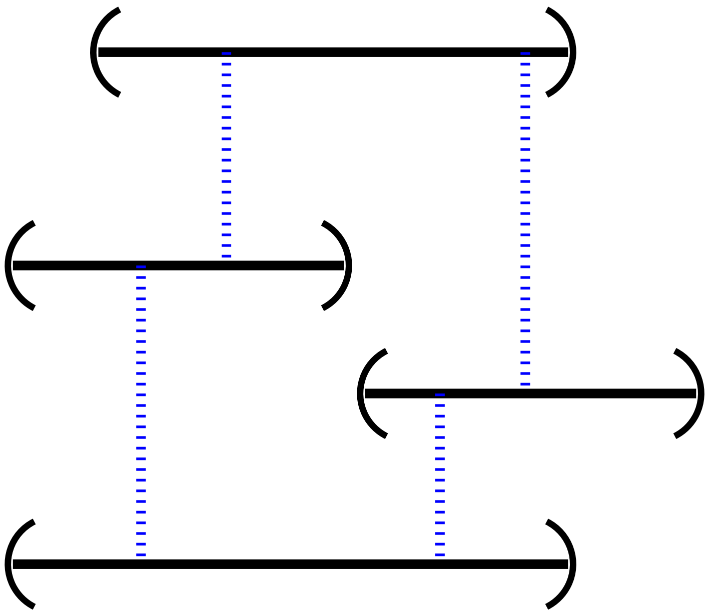


Bar Visibility Representations

Vertices correspond to horizontal open line segments (bars)

Edges correspond to vertical unobstructed vertical sightlines

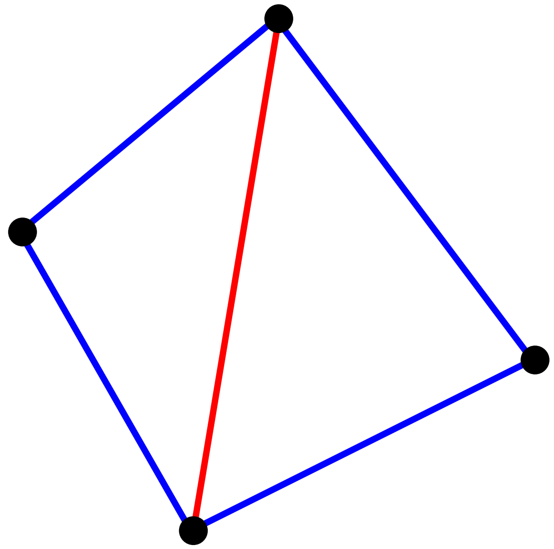


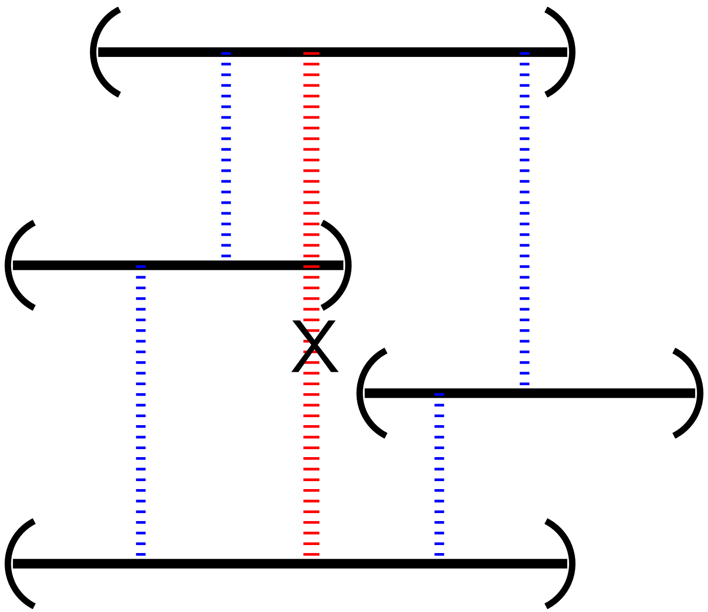


Bar Visibility Representations

Vertices correspond to horizontal open line segments (bars)

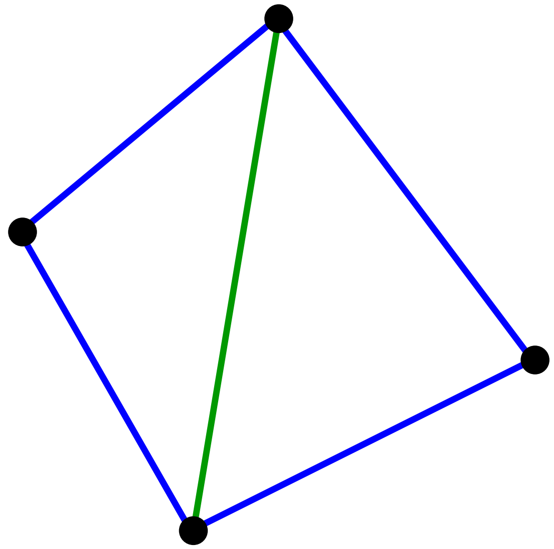
Edges correspond to vertical unobstructed vertical sightlines

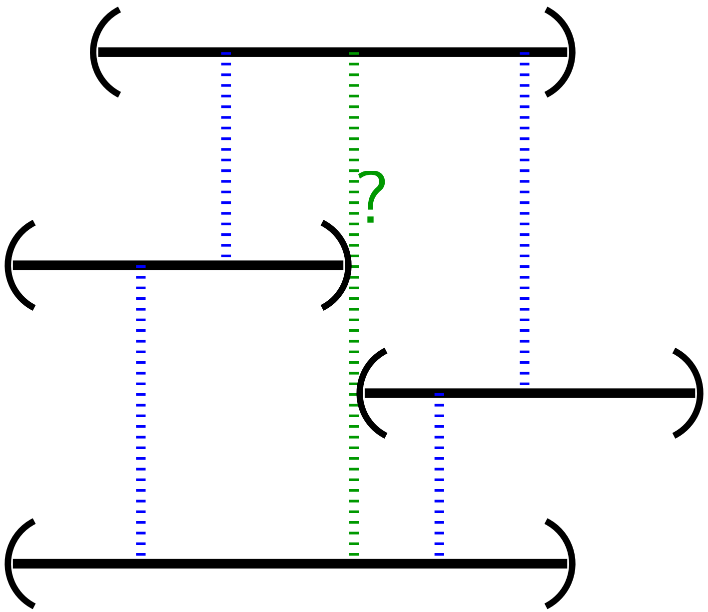




Bar Visibility Representations

Vertices correspond to horizontal open line segments (bars)
Edges correspond to vertical unobstructed vertical sightlines



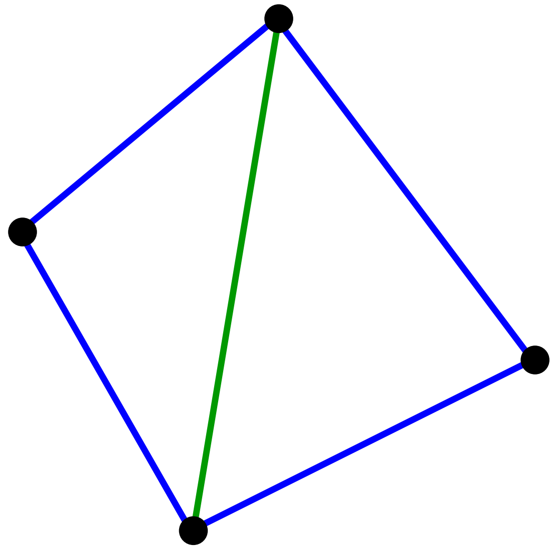


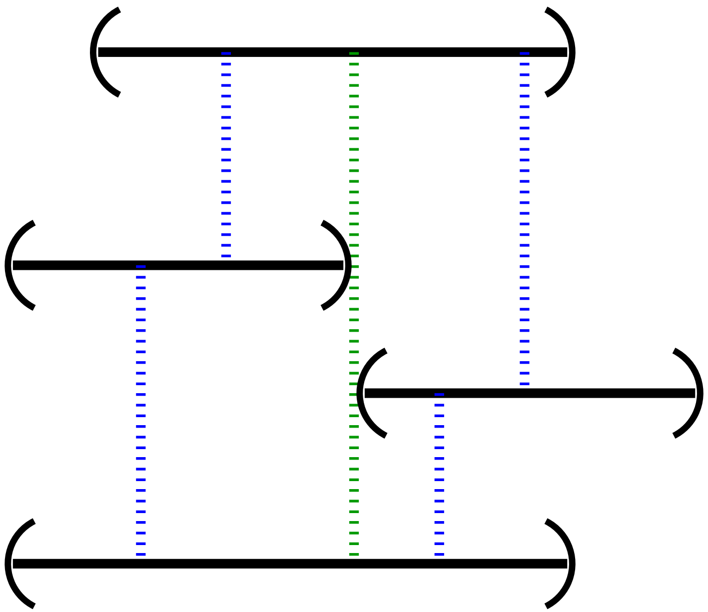
Models:

Bar Visibility Representations

Vertices correspond to horizontal open line segments (bars)

Edges correspond to vertical unobstructed vertical sightlines





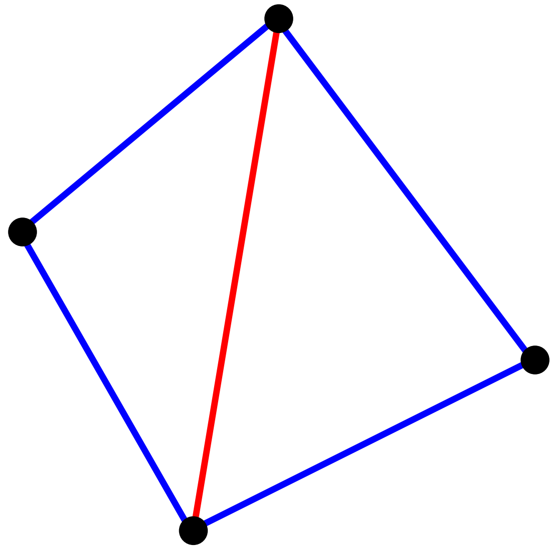
Models:

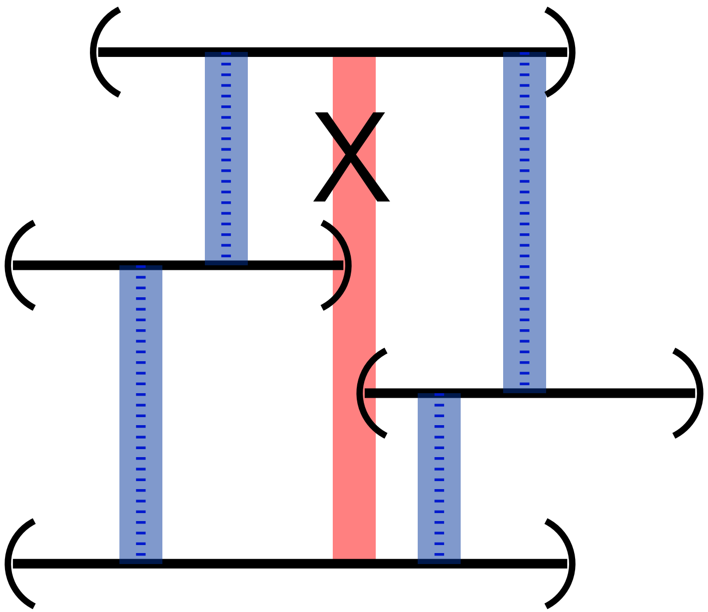
Strong: edge $uv \Leftrightarrow$ unobstructed (0-width) vertical sightline

Bar Visibility Representations

Vertices correspond to horizontal open line segments (bars)

Edges correspond to vertical unobstructed vertical sightlines





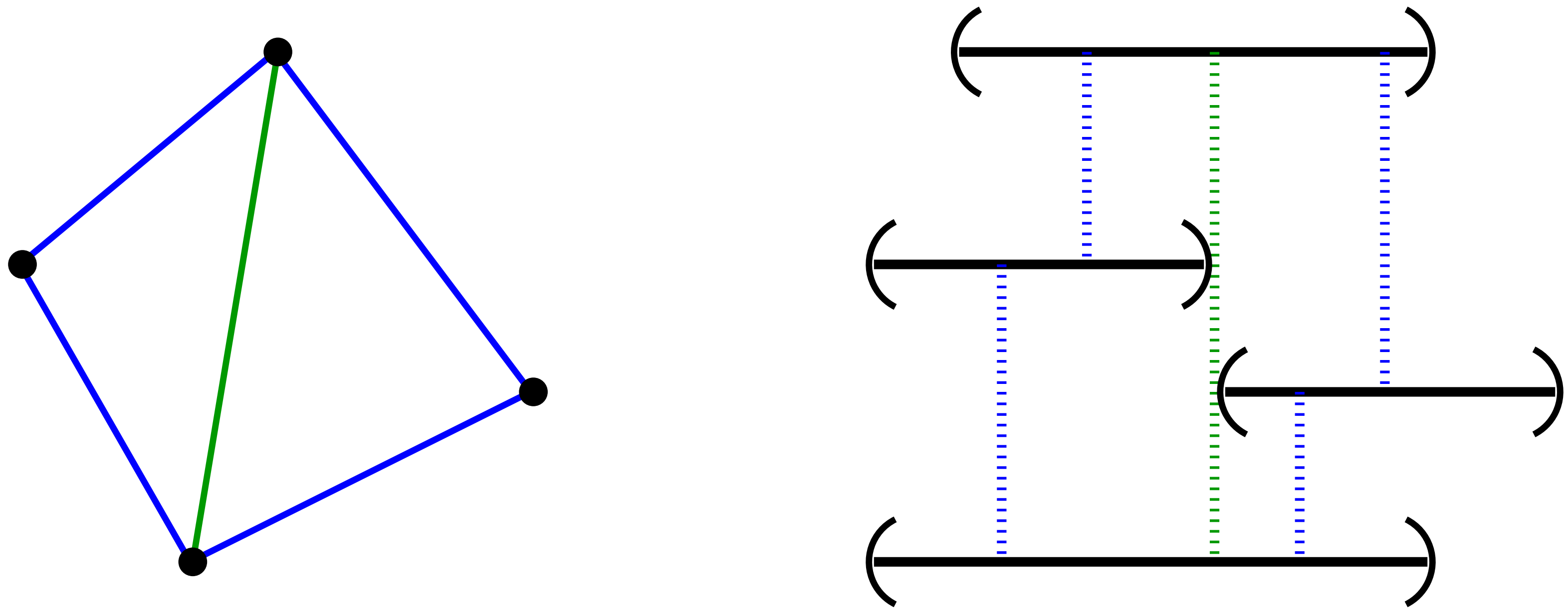
Models:

Strong: edge $uv \Leftrightarrow$ unobstructed (0-width) vertical sightline

ε : edge $uv \Leftrightarrow \varepsilon$ wide sight-line for $\varepsilon > 0$

Bar Visibility Representations

Vertices correspond to horizontal open line segments (bars)
Edges correspond to vertical unobstructed vertical sightlines



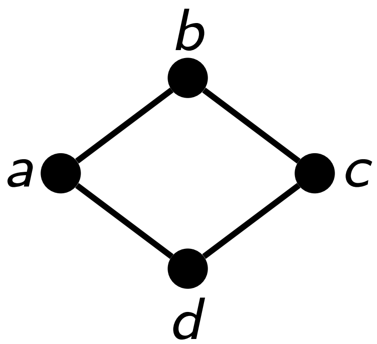
Models:

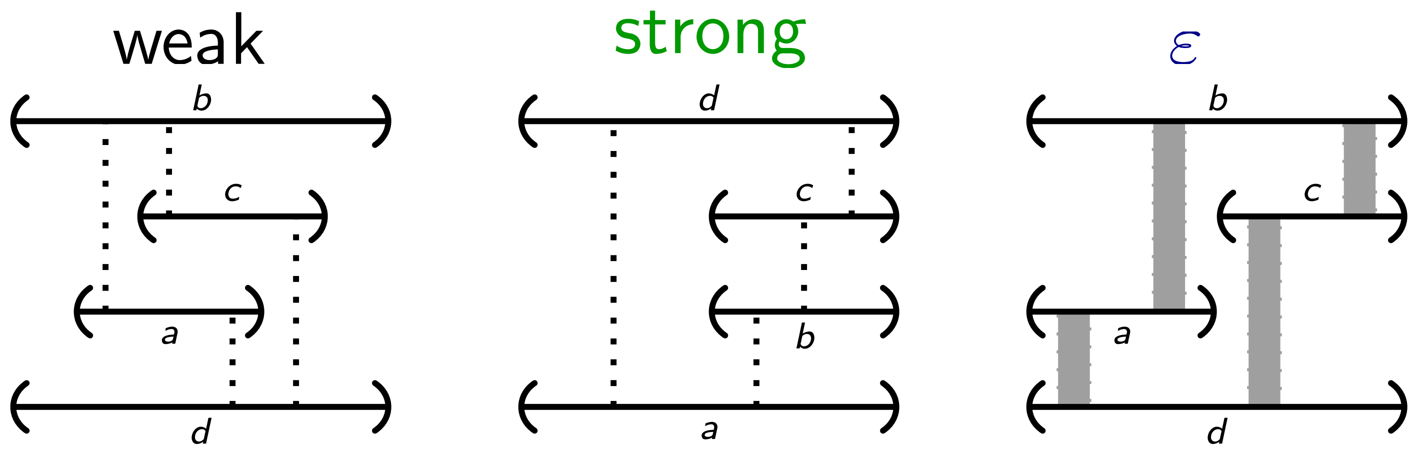
Strong: edge $uv \Leftrightarrow$ unobstructed (0-width) vertical sightline

ε : edge $uv \Leftrightarrow \varepsilon$ wide sight-line for $\varepsilon > 0$

Weak: edge $uv \Rightarrow$ unobstructed sightline
i.e., any subset of *visible* pairs

Problems

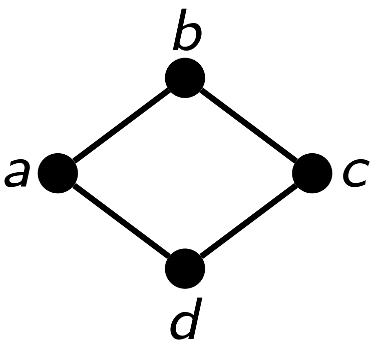


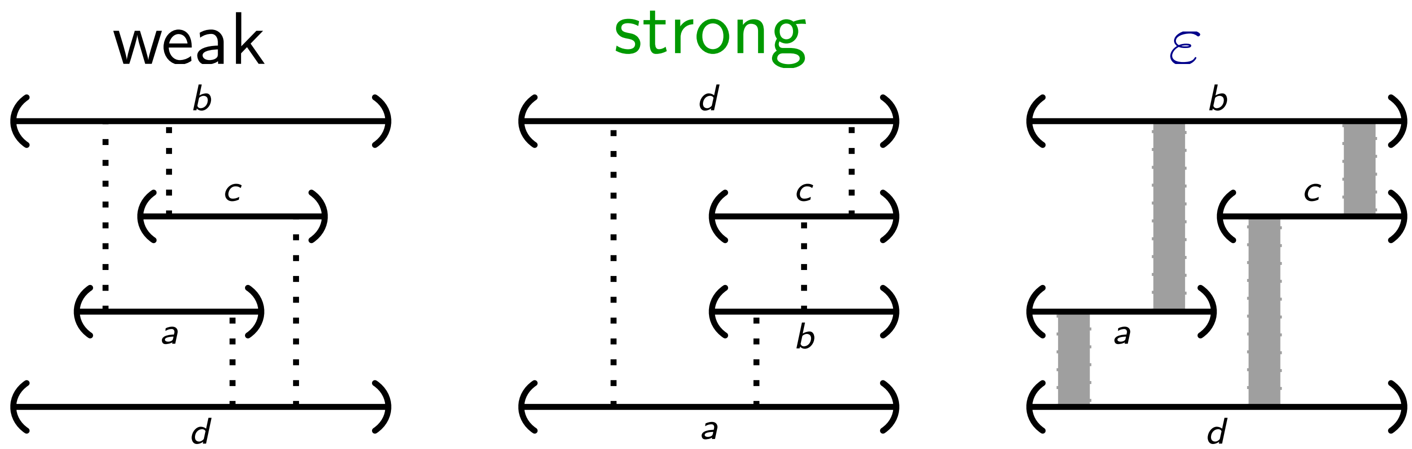


Recognition:

Given a graph G, **decide** if there exists a weak/strong/ε bar visibility representation ψ of G.

Problems

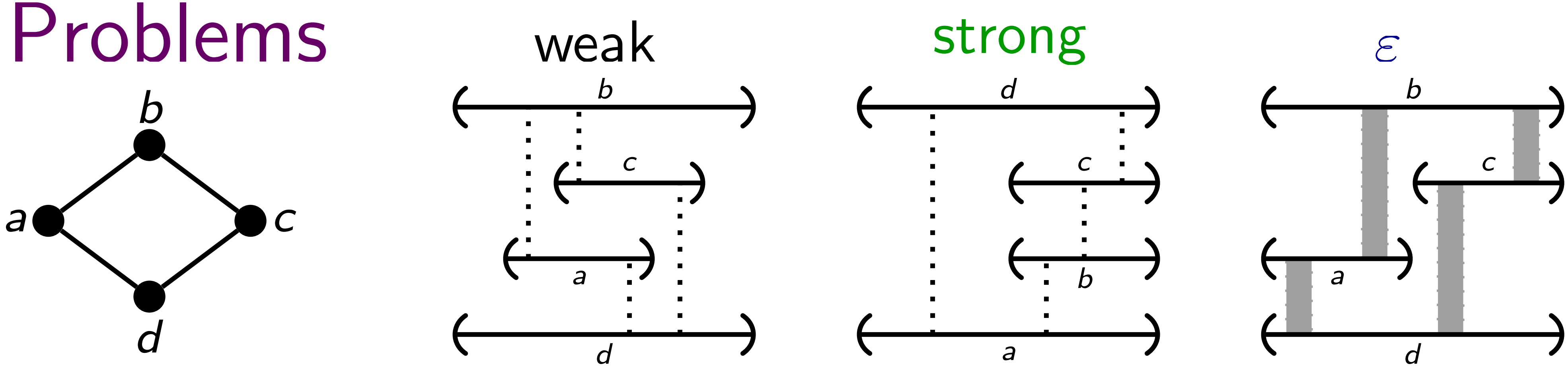




Recognition:
Given a graph G, **decide** if there exists a weak/strong/ε bar visibility representation ψ of G.

Construction:
Given a graph G, **construct** a weak/strong/ε bar visibility representation ψ of G when one exists.

Problems

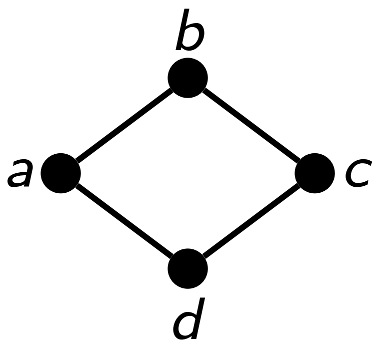


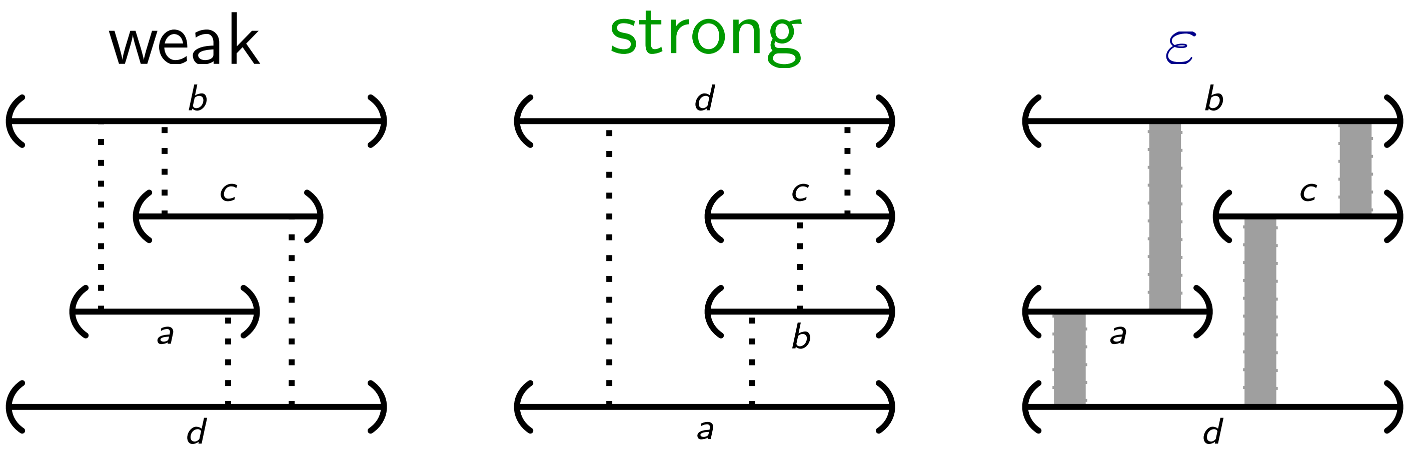
Recognition:
Given a graph G, **decide** if there exists a weak/strong/ε bar visibility representation ψ of G.

Construction:
Given a graph G, **construct** a weak/strong/ε bar visibility representation ψ of G when one exists.

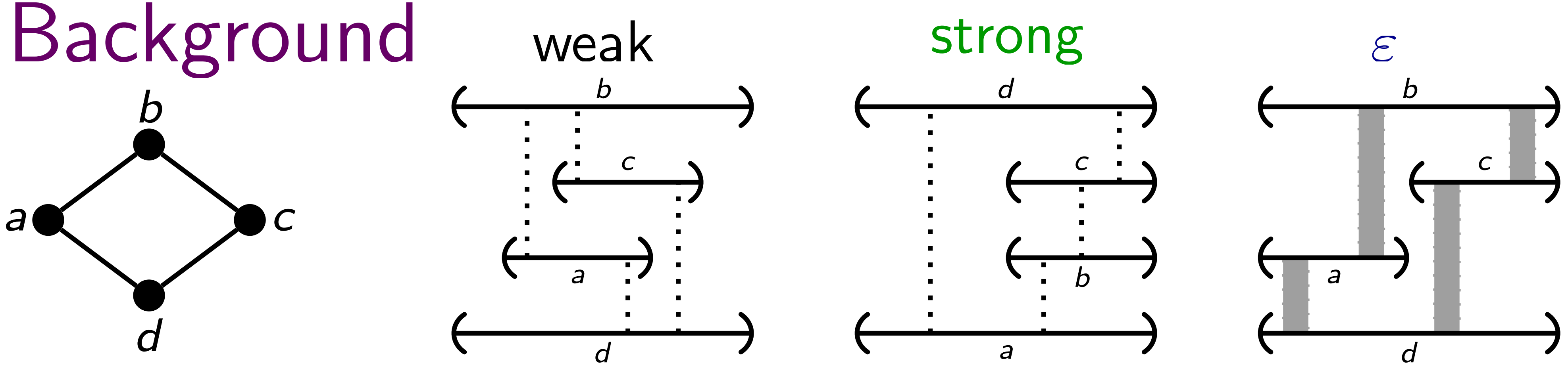
Representation Extension (Construction):
Given a graph G and set of bars ψ' of $V' \subset V(G)$, **decide** if there exists a weak/strong/ε bar visibility representation ψ of G where $\psi|_{V'} = \psi'$ (, and **construct** ψ when it exists).

Background





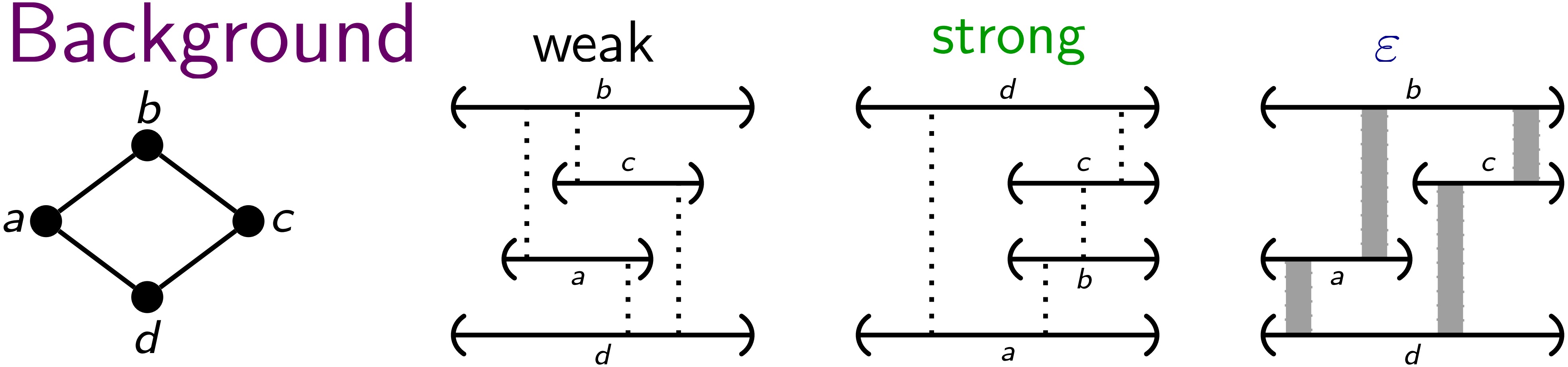
Background



Weak Bar Visibility

- All planar graphs. [Tammasia & Tollis 1986; Wismath 1985]
- Linear time recognition and construction [T&T 1986]
- Representation Extension is NP-complete [**C.**, Dorbec, Kratochvíl, Montassier, Stacho 2014]

Background



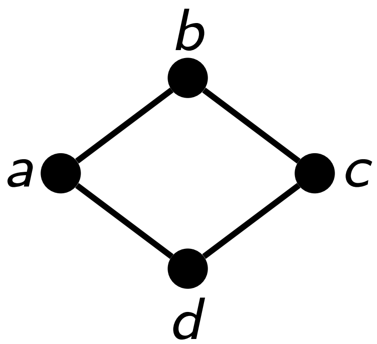
Weak Bar Visibility

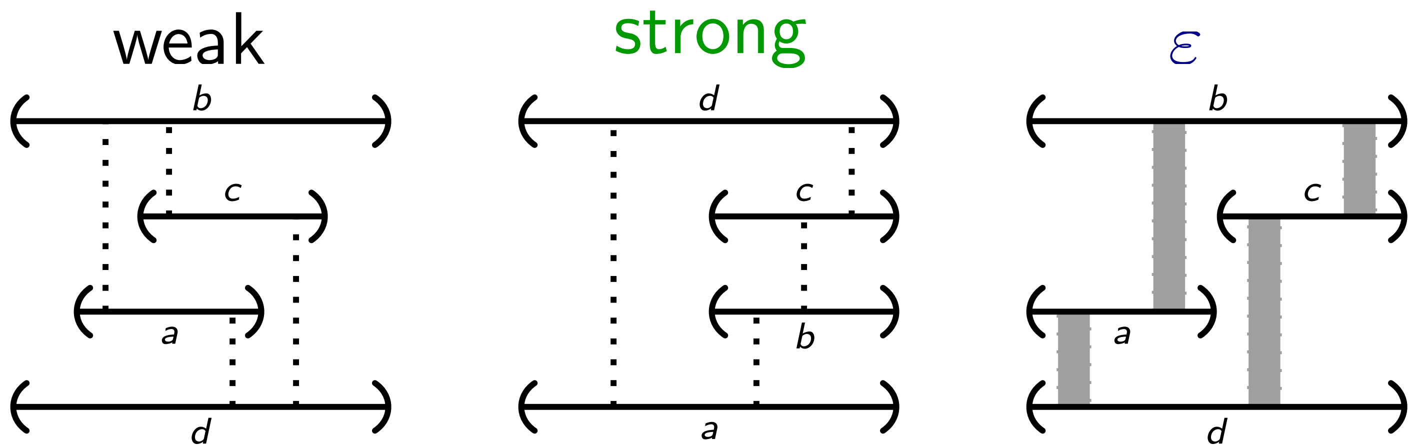
- All planar graphs. [Tammasia & Tollis 1986; Wismath 1985]
- Linear time recognition and construction [T&T 1986]
- Representation Extension is NP-complete [**C.**, Dorbec, Kratochvíl, Montassier, Stacho 2014]

Strong Bar Visibility

- NP-complete to recognize [Andreae 1992]

Background





Weak Bar Visibility

- All planar graphs. [Tammasia & Tollis 1986; Wismath 1985]
- Linear time recognition and construction [T&T 1986]
- Representation Extension is NP-complete [**C.**, Dorbec, Kratochvíl, Montassier, Stacho 2014]

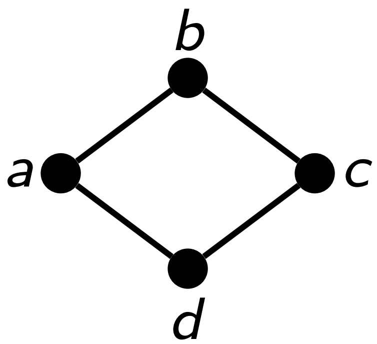
Strong Bar Visibility

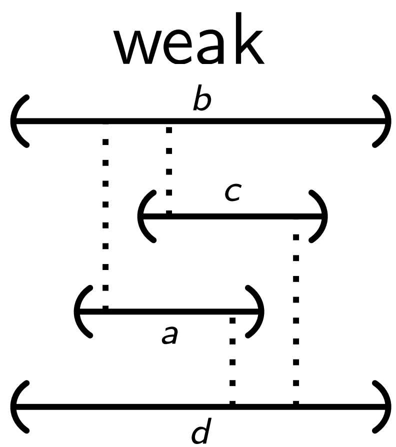
- NP-complete to recognize [Andreae 1992]

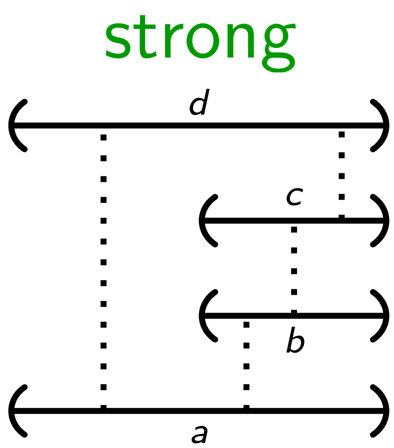
ε-Bar Visibility

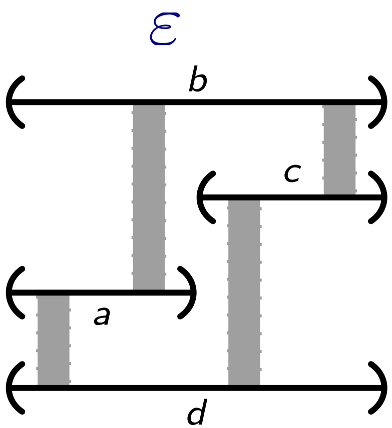
- Planar graphs that can be embedded with all **cut vertices** on the outerface. [T&T 1986, Wismath 1985]
- Linear time recognition and construction [T&T 1986]
- **What about Representation Extension?**

Background









Weak Bar Visibility

- All planar graphs. [Tammasia & Tollis 1986; Wismath 1985]
- Linear time recognition and construction [T&T 1986]
- Representation Extension is NP-complete [**C.**, Dorbec, Kratochvíl, Montassier, Stacho 2014]

Strong Bar Visibility

- NP-complete to recognize [Andreae 1992]

ε-Bar Visibility

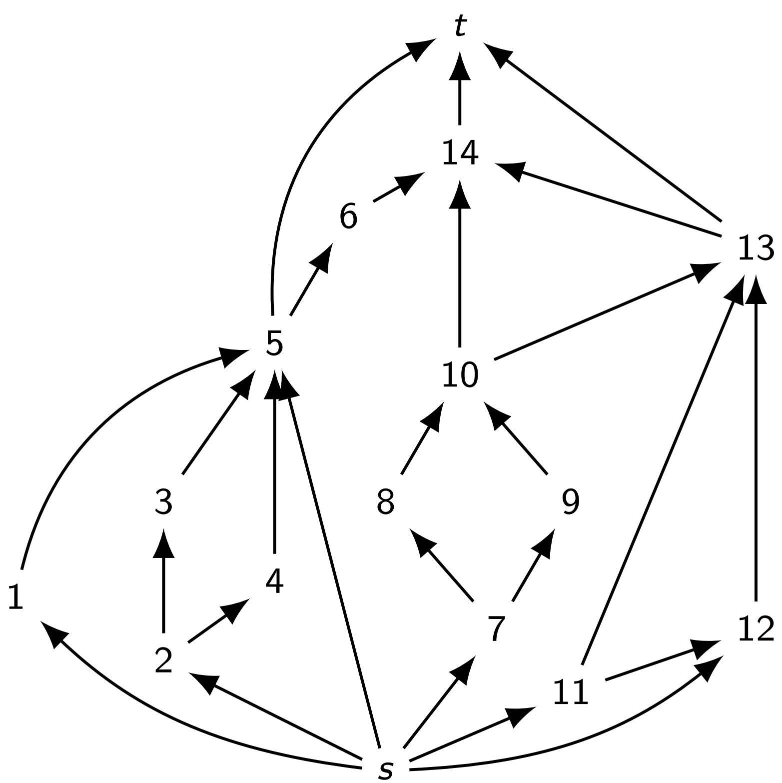
- Planar graphs that can be embedded with all **cut vertices** on the outerface. [T&T 1986, Wismath 1985]
- Linear time recognition and construction [T&T 1986]
- **What about Representation Extension?** ⟵ **This Talk !**

First another definition ..... planar digraphs

Planar st-**graphs**: planar digraph G with exactly one **source** s and one **sink** t where s and t occur on the same *face* (i.e., the **outerface**) of an embedding of G.

First another definition ..... planar digraphs

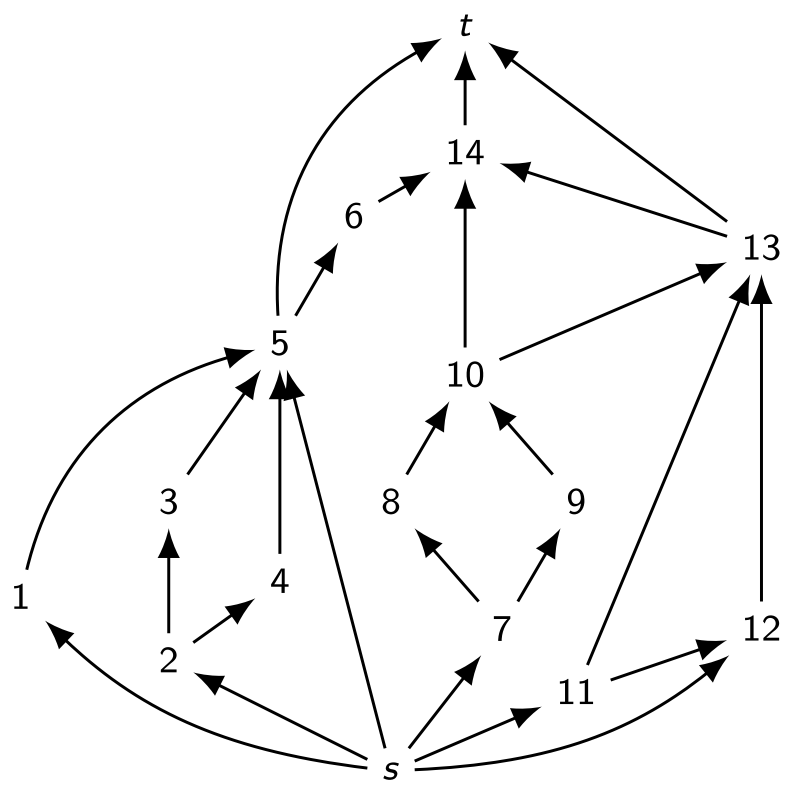
Planar *st*-**graphs**: planar digraph G with exactly one **source** s and one **sink** t where s and t occur on the same *face* (i.e., the **outerface**) of an embedding of G.



First another definition ..... planar digraphs

Planar *st*-**graphs**: planar digraph G with exactly one **source** s and one **sink** t where s and t occur on the same *face* (i.e., the **outerface**) of an embedding of G.

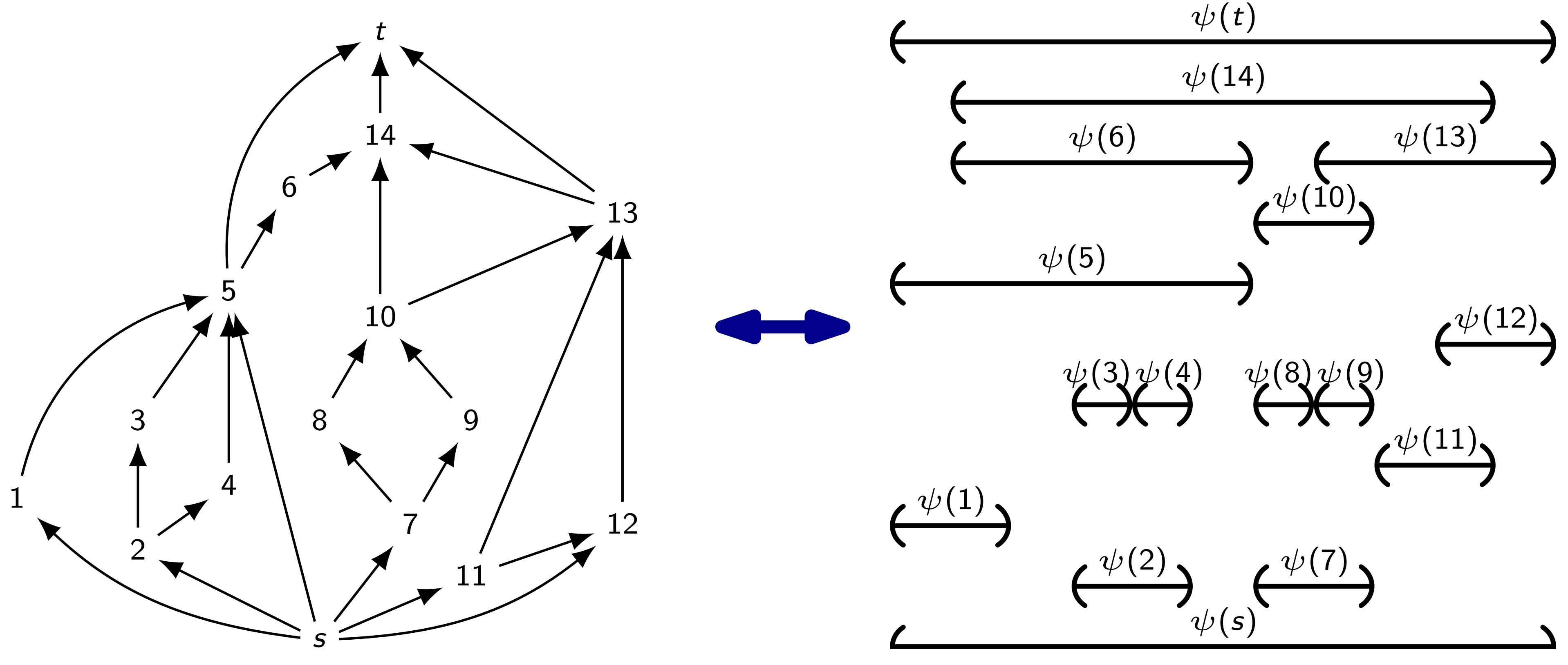
st-orientations correspond to ε-Bar Visibility Representations



First another definition ..... planar digraphs

Planar st**-graphs**: planar digraph G with exactly one **source** s and one **sink** t where s and t occur on the same *face* (i.e., the **outerface**) of an embedding of G.

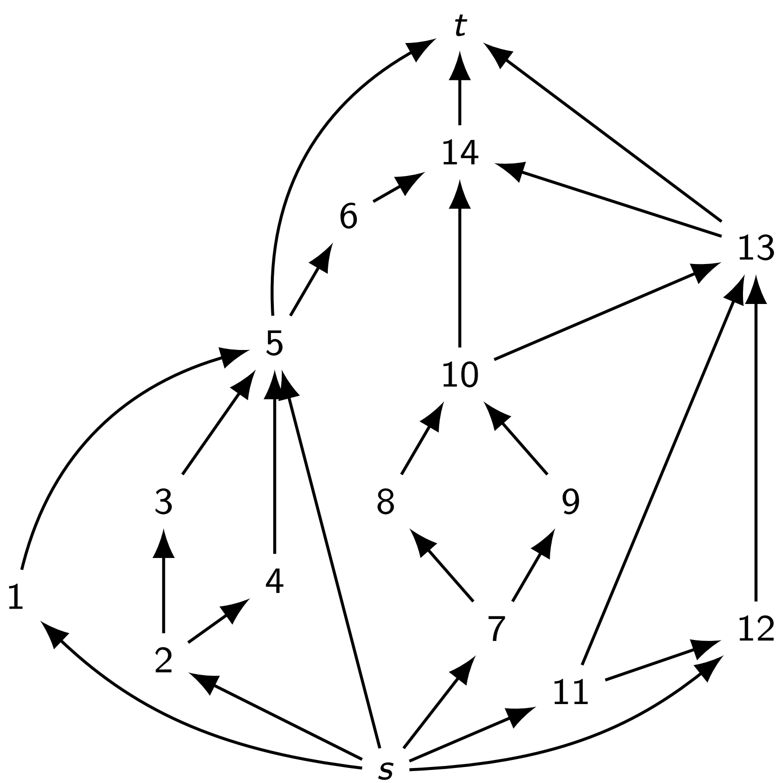
st-orientations correspond to ε-Bar Visibility Representations



First another definition ..... planar digraphs

Planar st-graphs: planar digraph G with exactly one **source** s and one **sink** t where s and t occur on the same *face* (i.e., the **outerface**) of an embedding of G.

st-orientations correspond to ε-Bar Visibility Representations

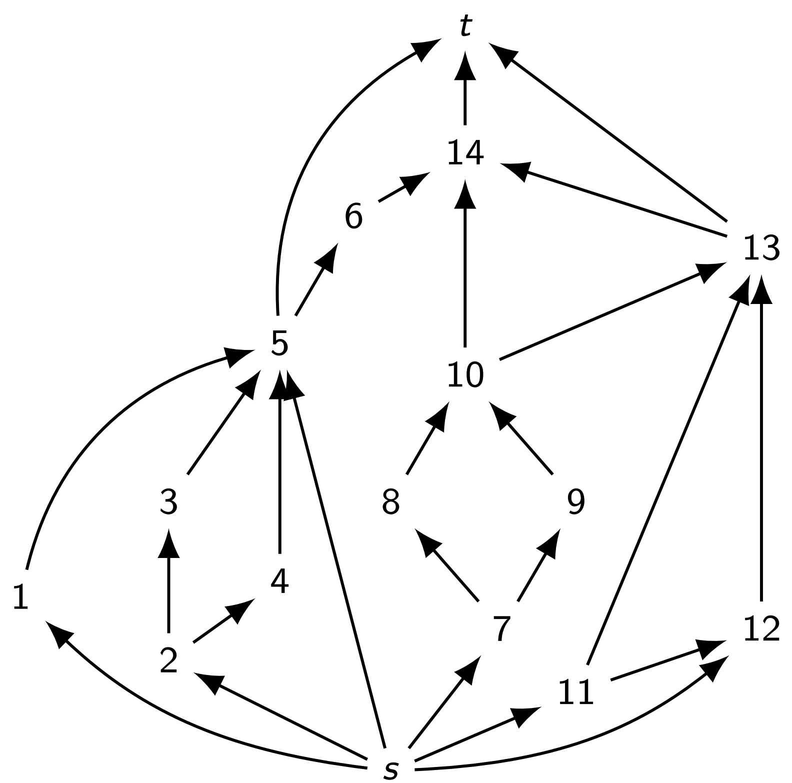


Note: unlike for undirected planar graphs, testing whether a directed acyclic planar graph has a **Weak Bar Visibility representation** is NP-complete → this is *upward planarity testing* [Garg & Tamassia 2001].

First another definition ..... planar digraphs

Planar *st*-graphs: planar digraph G with exactly one **source** s and one **sink** t where s and t occur on the same *face* (i.e., the **outerface**) of an embedding of G.

st-orientations correspond to ε-Bar Visibility Representations



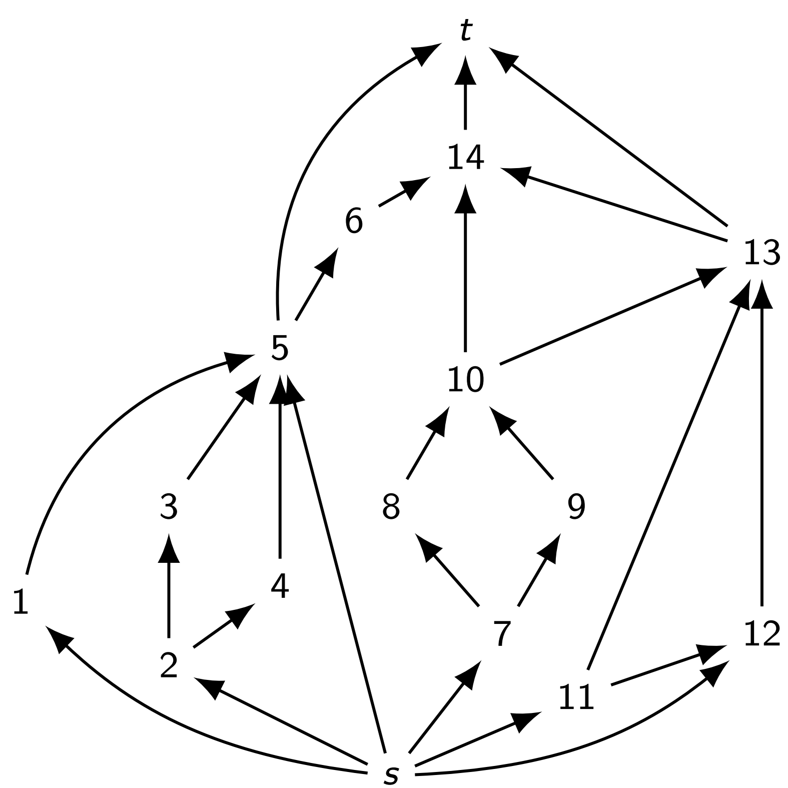
Note: unlike for undirected planar graphs, testing whether a directed acyclic planar graph has a **Weak Bar Visibility representation** is NP-complete $\rightarrow$ this is *upward planarity testing* [Garg & Tamassia 2001].

ε-Bar Visability testing is easily done via *st*-graph recognition.

First another definition ..... planar digraphs

Planar *st*-graphs: planar digraph G with exactly one **source** s and one **sink** t where s and t occur on the same *face* (i.e., the **outerface**) of an embedding of G.

st-orientations correspond to ε-Bar Visibility Representations



Note: unlike for undirected planar graphs, testing whether a directed acyclic planar graph has a **Weak Bar Visibility representation** is NP-complete $\rightarrow$ this is *upward planarity testing* [Garg & Tamassia 2001].

ε-Bar Visability testing is easily done via *st*-graph recognition.

Strong Bar Visibility recognition... open?

Results

ε-Bar Visibility Representation Extension is NP-complete

↳ Reduction: *planar monotone 3-SAT*

Results

ε-Bar Visibility Representation Extension is NP-complete

↳ Reduction: *planar monotone 3-SAT*

rectangular ε-Bar Visibility Representation Extension can be solved in $O(n \log^2 n)$ time for st-graphs

↳ Dynamic Program: via *SPQR-trees*

Results

ε-Bar Visibility Representation Extension is NP-complete

↳ Reduction: *planar monotone 3-SAT*

rectangular ε-Bar Visibility Representation Extension can be solved in $O(n \log^2 n)$ time for st-graphs

↳ Dynamic Program: via *SPQR-trees*

ε-Bar Visibility Representation Extension is NP-complete for (series-parallel) st-graphs when restricted to the *Integer Grid* (or if any fixed $\varepsilon > 0$ is specified).

↳ Reduction: *3-partition*

Results

ε-Bar Visibility Representation Extension is NP-complete
↳ Reduction: *planar monotone 3-SAT*

rectangular ε-Bar Visibility Representation Extension can be solved in $O(n \log^2 n)$ time for st-graphs
↳ Dynamic Program: via *SPQR-trees*
(easier version: $O(n^2)$)

ε-Bar Visibility Representation Extension is NP-complete for (series-parallel) st-graphs when restricted to the *Integer Grid* (or if any fixed $\varepsilon > 0$ is specified).
↳ Reduction: *3-partition*

Results

ε-Bar Visibility Representation Extension is NP-complete

↳ Reduction: *planar monotone 3-SAT*

Focus of this talk

rectangular ε-Bar Visibility Representation Extension can be solved in ~~$O(n \log^2 n)$~~ time for *st*-graphs

↳ Dynamic Program: via *SPQR-trees*

(easier version: $O(n^2)$)

ε-Bar Visibility Representation Extension is NP-complete for (series-parallel) *st*-graphs when restricted to the *Integer Grid* (or if any fixed $\varepsilon > 0$ is specified).

↳ Reduction: *3-partition*

st-graphs : ε-Bar Visibility Representation Extension

Simplifying our life a little: y-**coordinate invariant**

st-graphs : ε-Bar Visibility Representation Extension

Simplifying our life a little: ***y*-coordinate invariant**

Let G be an st-graph, and ψ' be a representation of $V' \subseteq V(G)$.
Let $y_v : V(G) \to \mathbb{R}$ such that

- for each $v \in V'$, $y_v =$ the y-coordinate of $\psi'(v)$.
- for each $u \to v$, $y_u < y_v$.

st-graphs : ε-Bar Visibility Representation Extension

Simplifying our life a little: **y-coordinate invariant**

Let G be an st-graph, and ψ' be a representation of $V' \subseteq V(G)$.
Let $y_v : V(G) \to \mathbb{R}$ such that

- for each $v \in V'$, $y_v =$ the y-coordinate of $\psi'(v)$.
- for each $u \to v$, $y_u < y_v$.

G has a representation extending ψ' if and only if G has a representation ψ extending ψ' where the y-coordinates of the bars are as in y.

st-graphs : ε-Bar Visibility Representation Extension

Simplifying our life a little: ***y*-coordinate invariant**

Let G be an st-graph, and ψ' be a representation of $V' \subseteq V(G)$.
Let $y_v : V(G) \to \mathbb{R}$ such that

- for each $v \in V'$, $y_v =$ the y-coordinate of $\psi'(v)$.
- for each $u \to v$, $y_u < y_v$.

G has a representation extending ψ' if and only if G has a representation ψ extending ψ' where the y-coordinates of the bars are as in y.

Proof idea: the relateive positions of **adjacent** bars must be match the order given by y.
So, we can adjust the y-coordinates of any solution to be as in y by sweeping from bottom-to-top. ■

st-graphs : ε-Bar Visibility Representation Extension

We can now assume all y-coordinates are given!

Simplifying our life a little: ***y*-coordinate invariant**

Let G be an st-graph, and ψ' be a representation of $V' \subseteq V(G)$.
Let $y_v : V(G) \to \mathbb{R}$ such that

- for each $v \in V'$, $y_v =$ the y-coordinate of $\psi'(v)$.
- for each $u \to v$, $y_u < y_v$.

G has a representation extending ψ' if and only if G has a representation ψ extending ψ' where the y-coordinates of the bars are as in y.

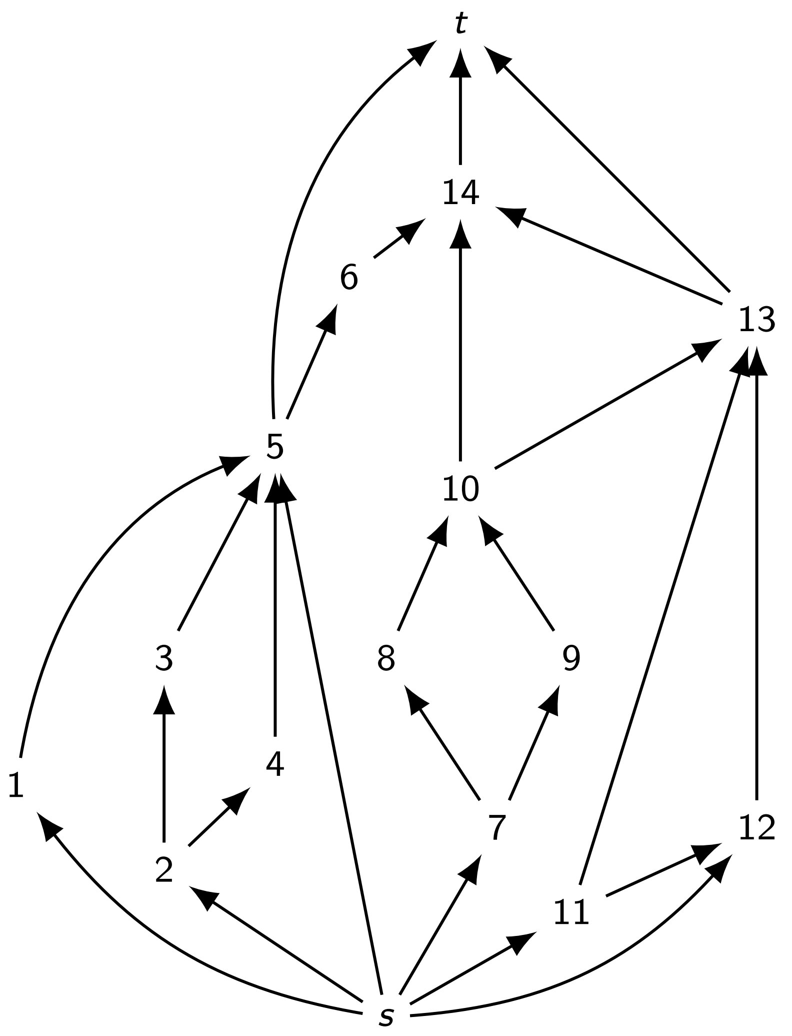
Proof idea: the relateive positions of **adjacent** bars must be match the order given by y.
So, we can adjust the y-coordinates of any solution to be as in y by sweeping from bottom-to-top. ■

SPQR-trees

SPQR-tree : decomposition of a planar graph by *separation pairs*.

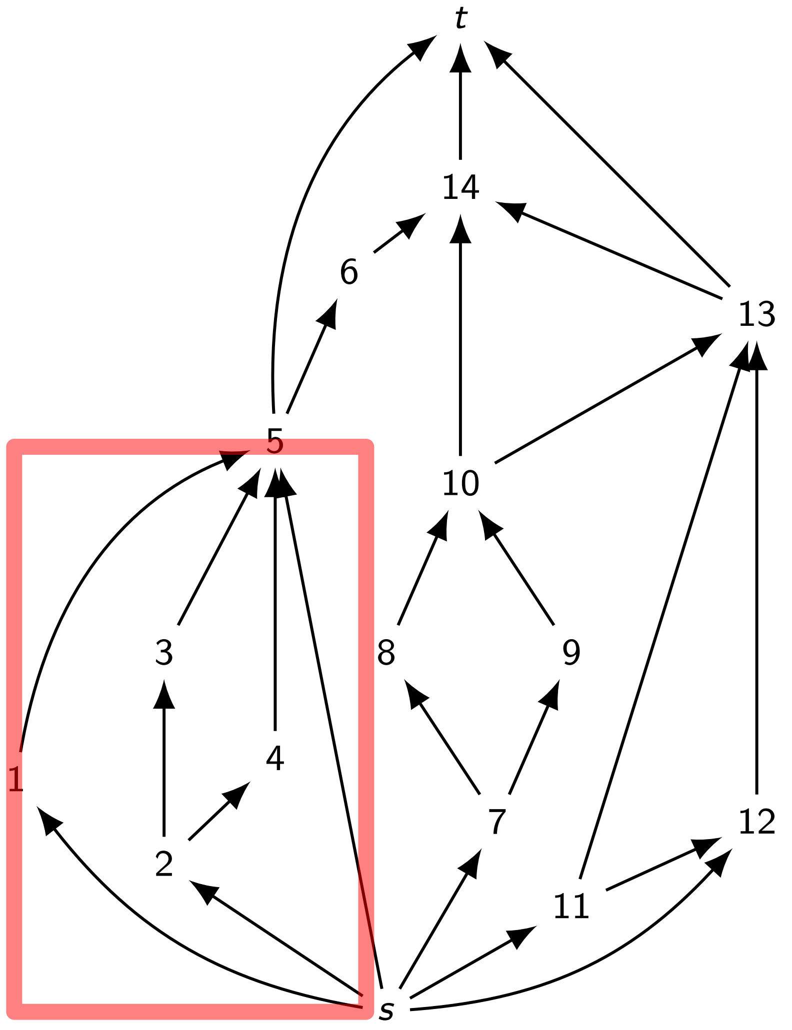
SPQR-trees

SPQR-tree : decomposition of a planar graph by *separation pairs*.



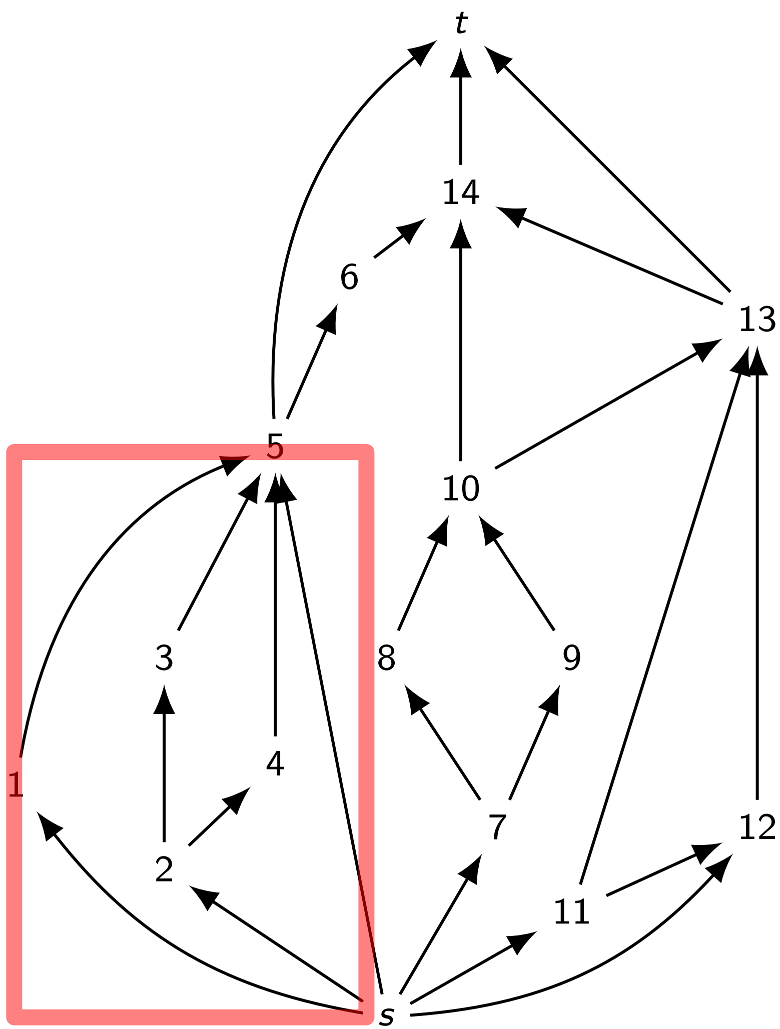
SPQR-trees

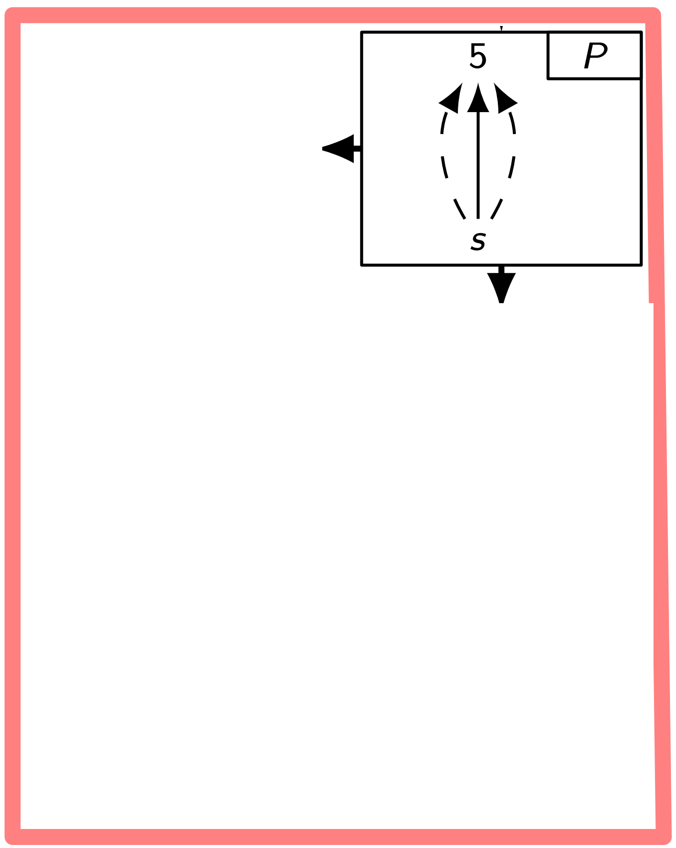
SPQR-tree : decomposition of a planar graph by *separation pairs*.



SPQR-trees

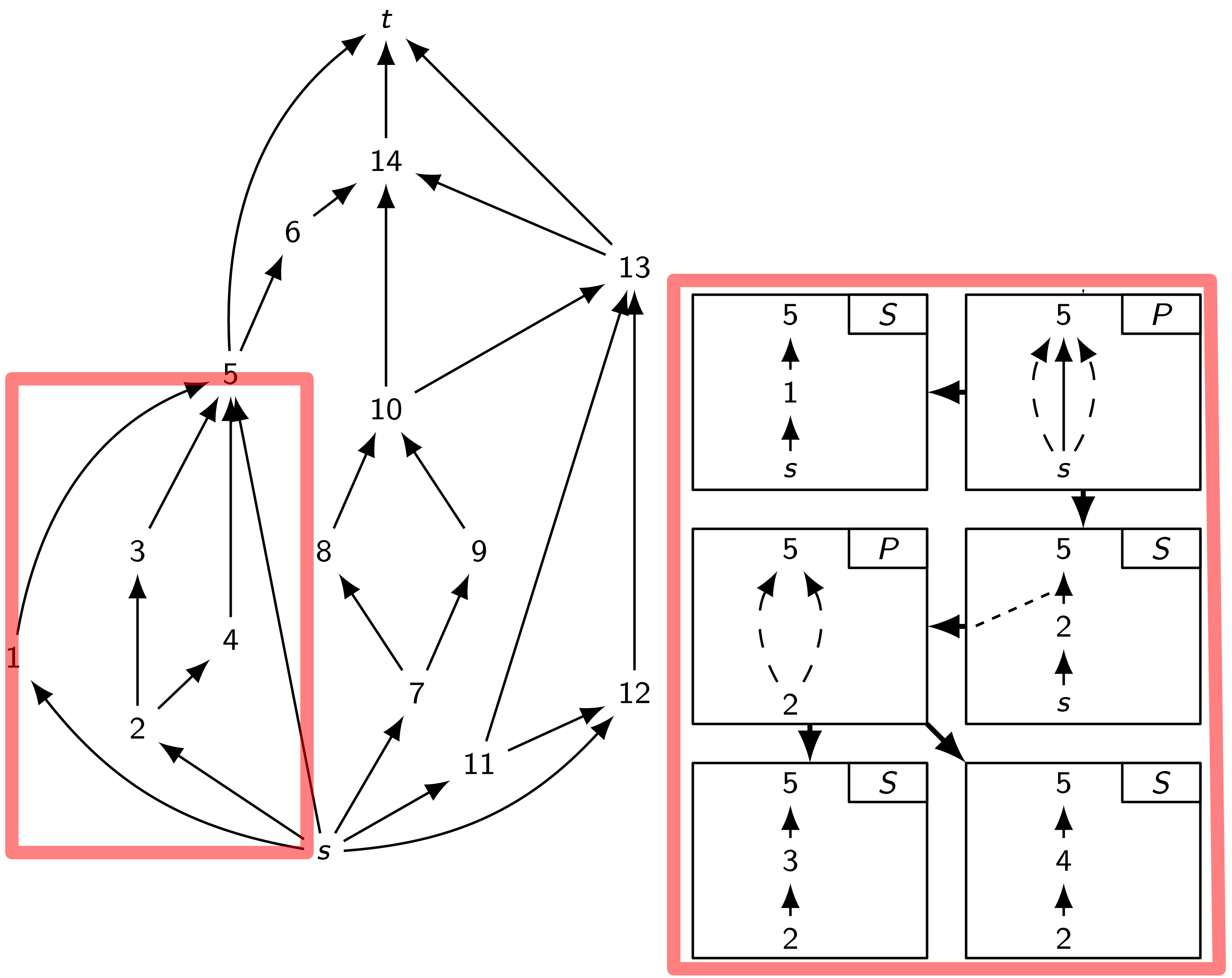
SPQR-tree : decomposition of a planar graph by *separation pairs*.





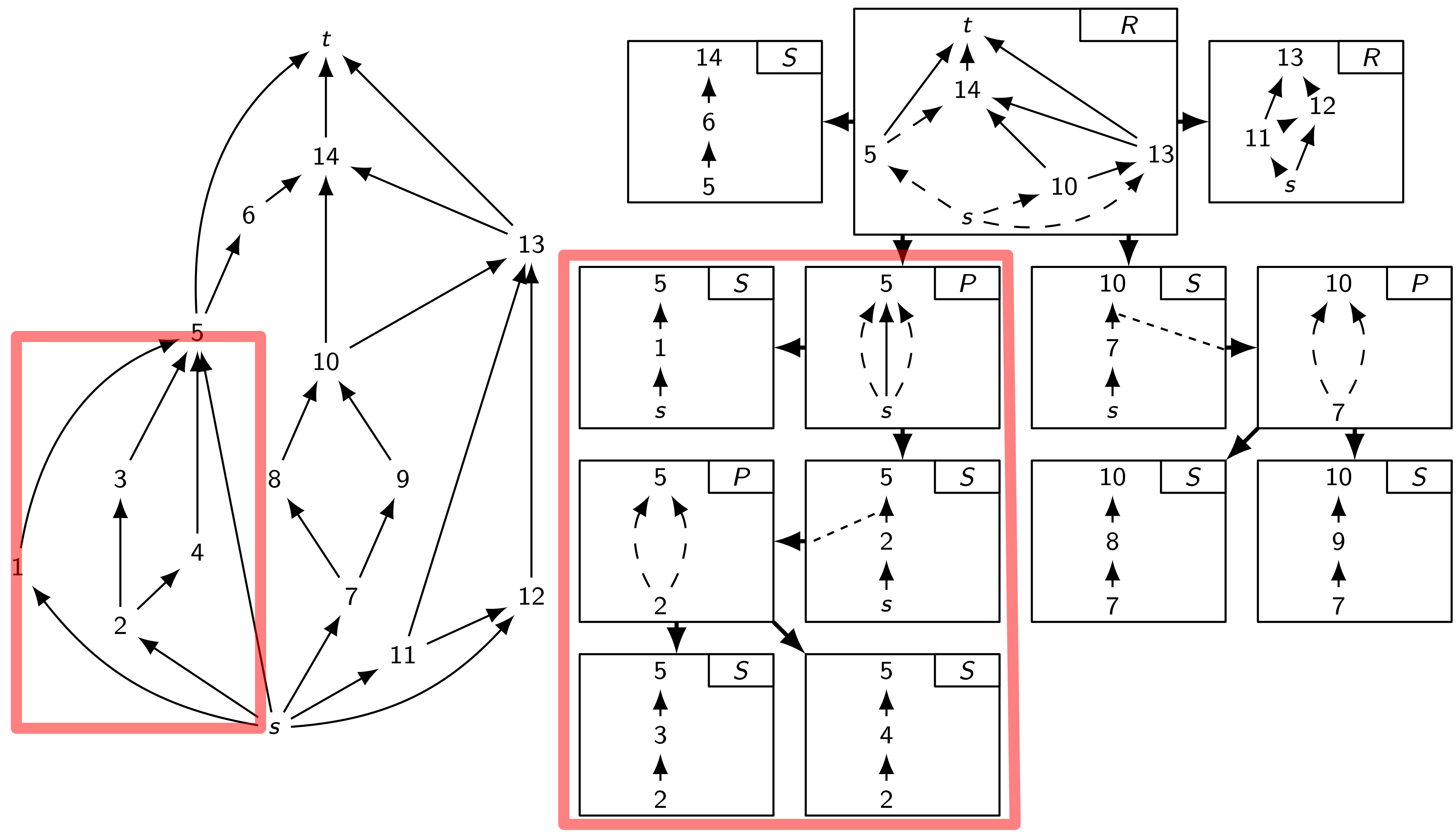
SPQR-trees

SPQR-tree : decomposition of a planar graph by *separation pairs*.

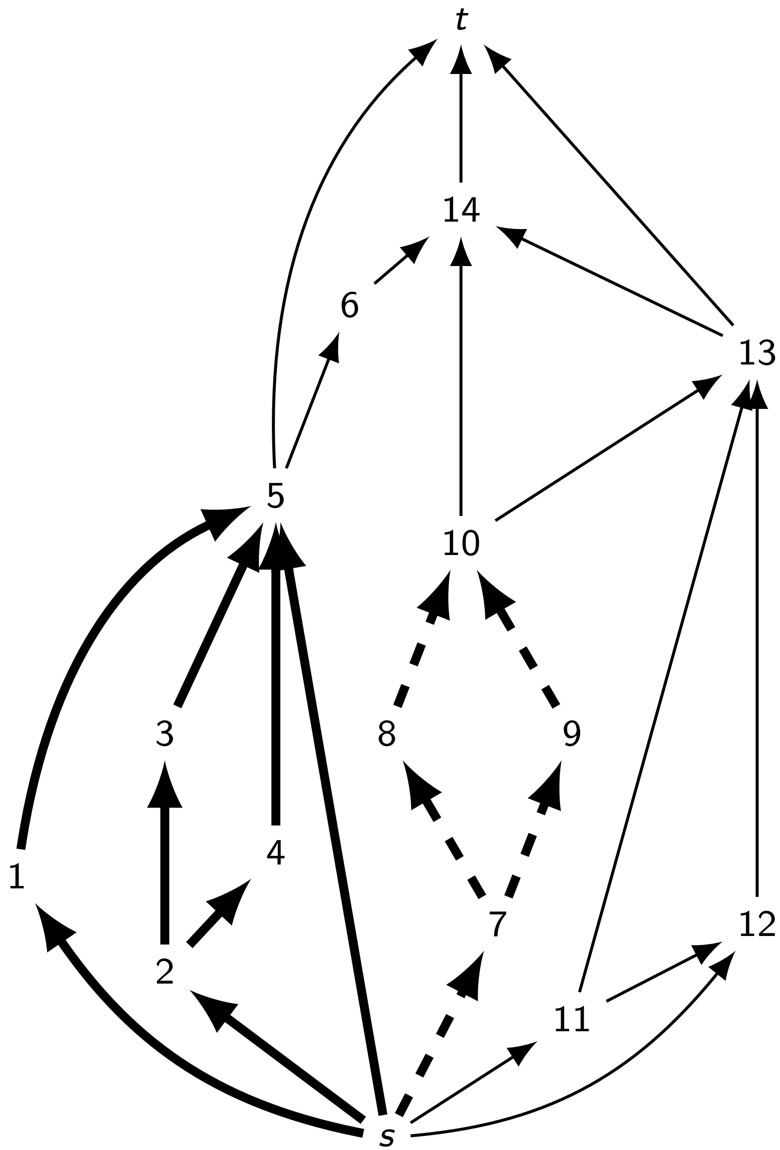


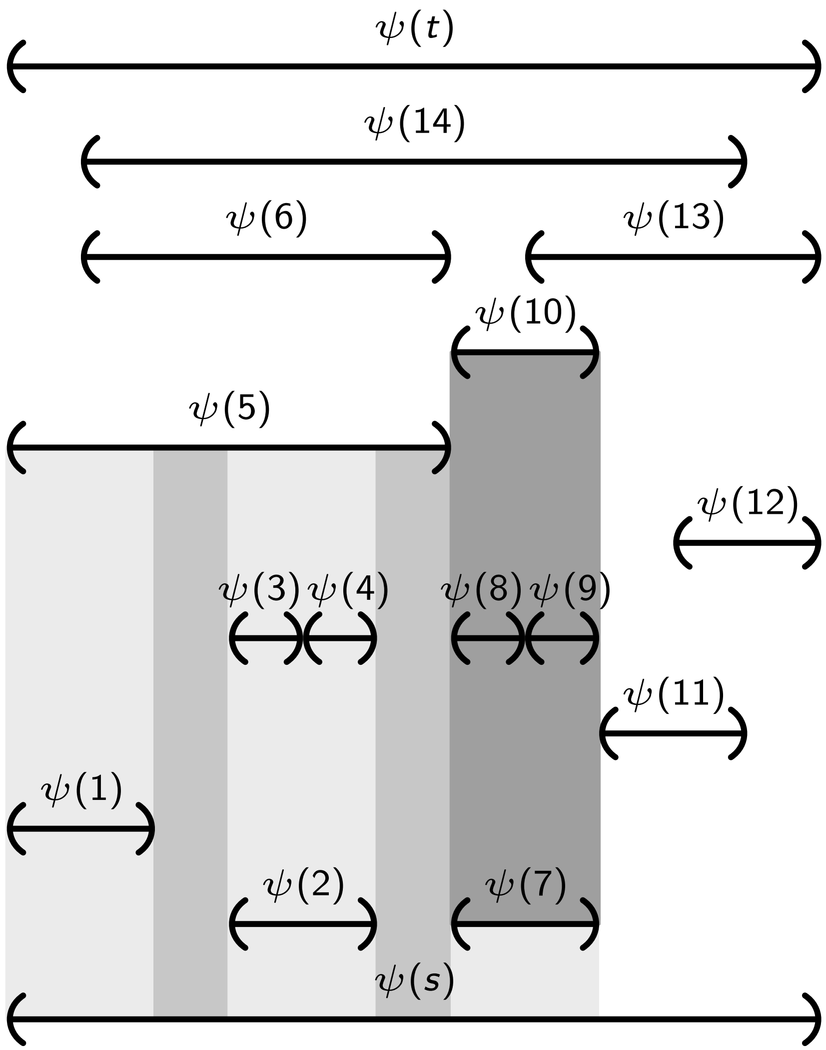
SPQR-trees

SPQR-tree : decomposition of a planar graph by *separation pairs*.

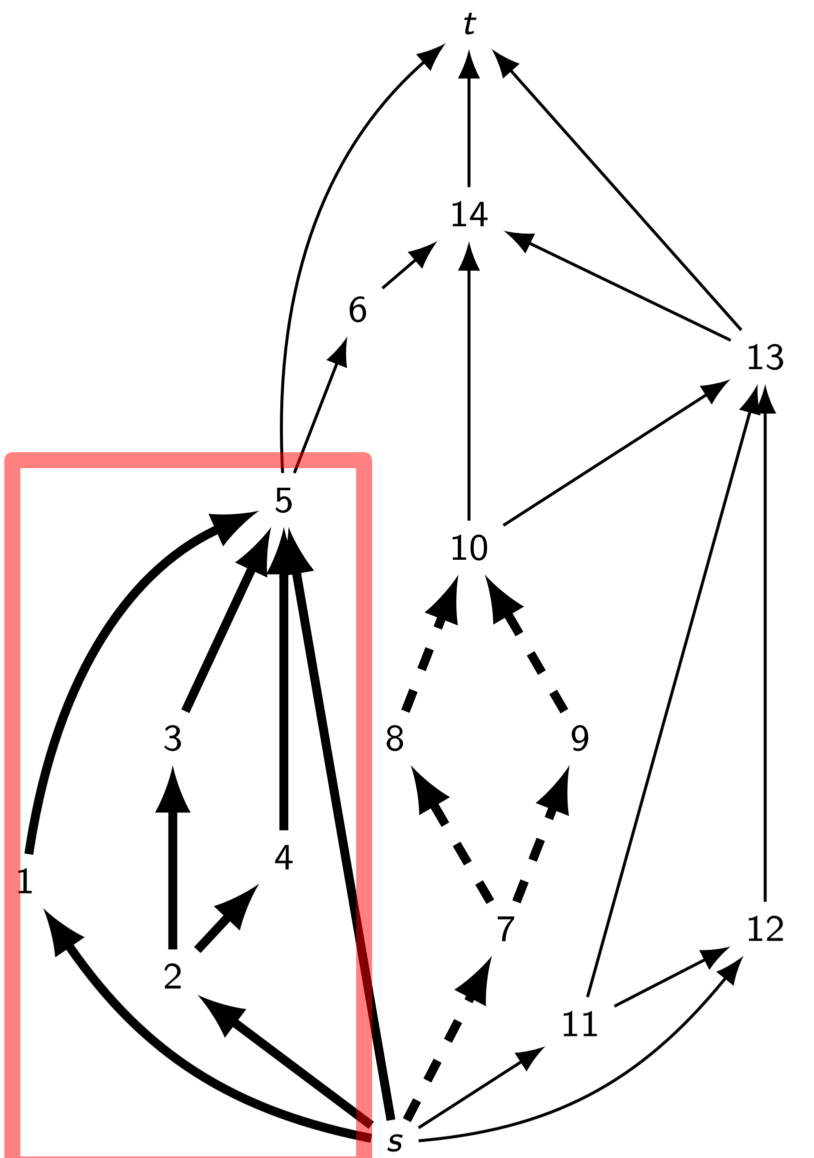


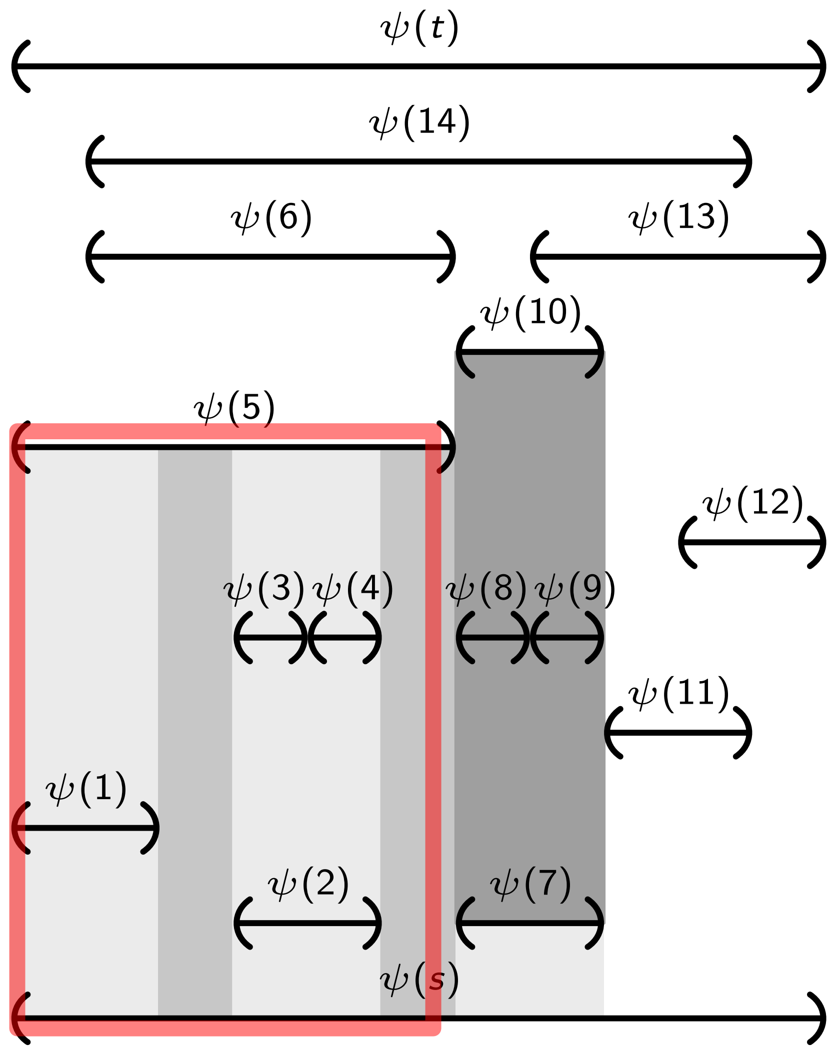
But why do SQPR-trees help?



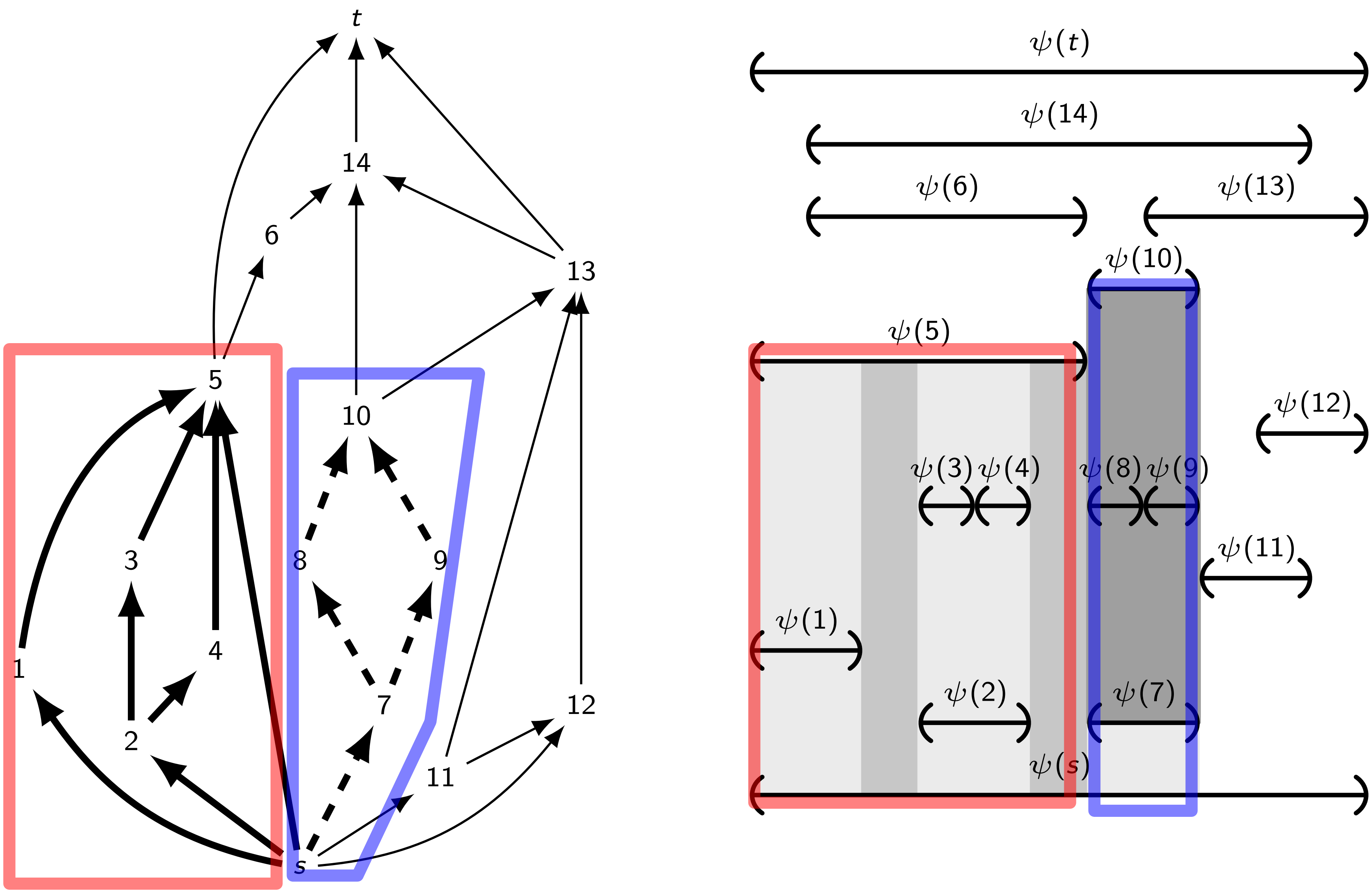


But why do SQPR-trees help?



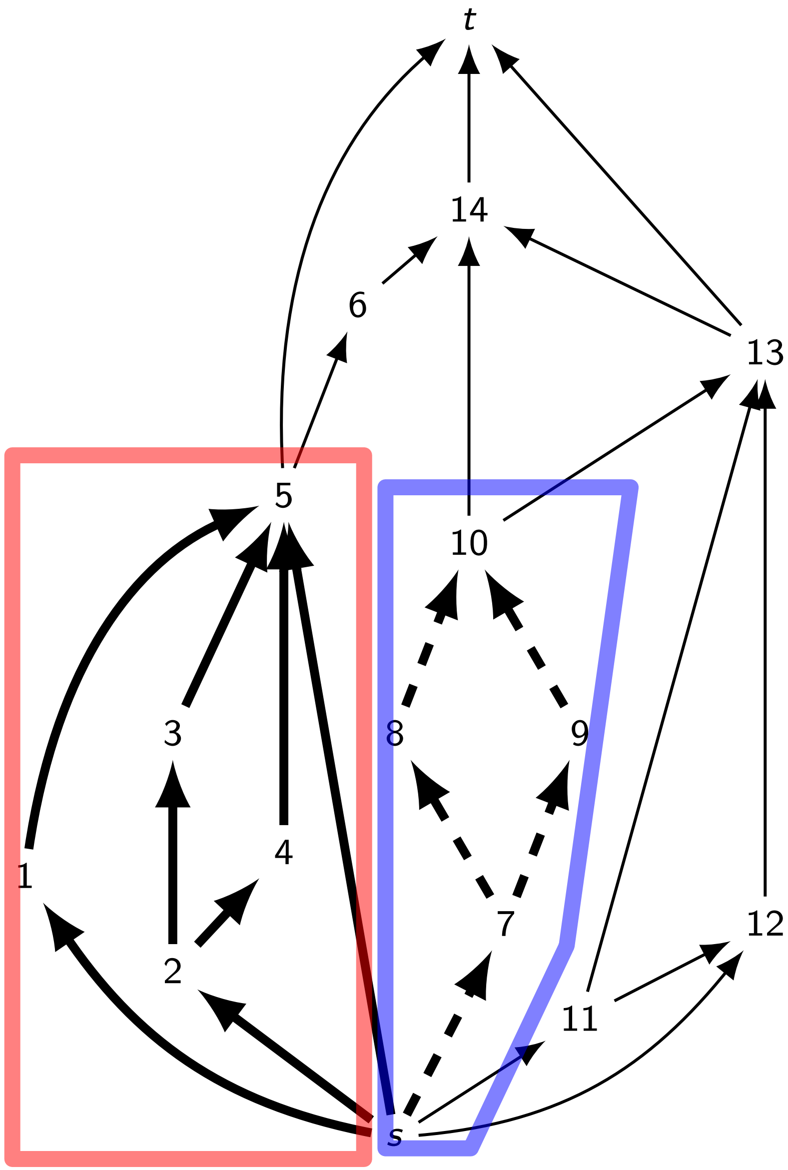


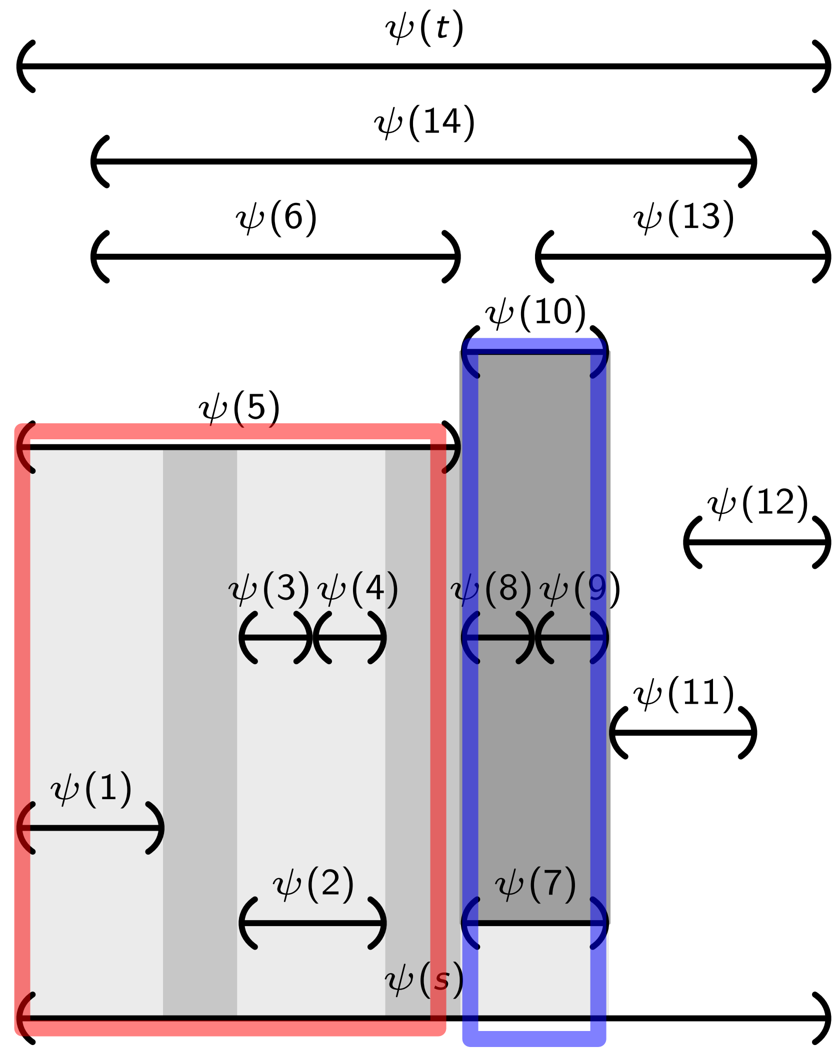
But why do SQPR-trees help?



But why do SQPR-trees help?

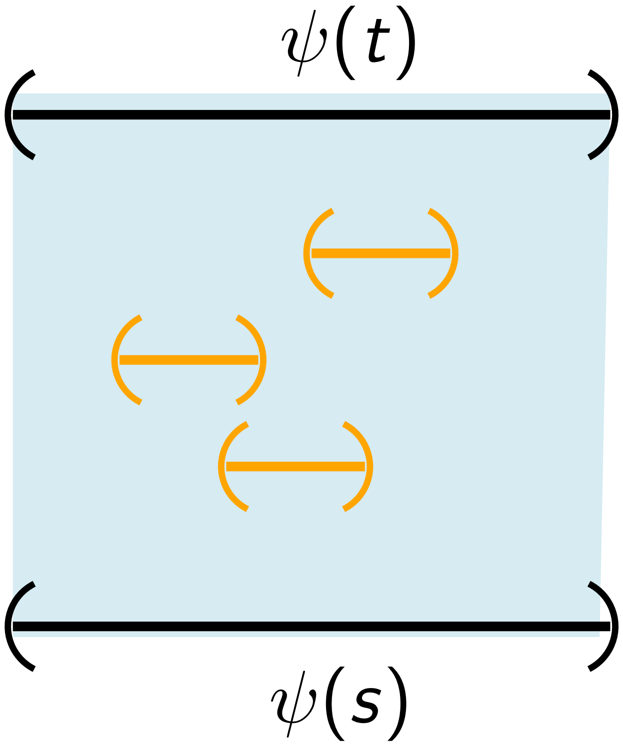
Lemma: The SPQR-tree of an *st*-graph *G* induces a recursive **tiling** of any ε-Bar Visibility Representation of *G*.





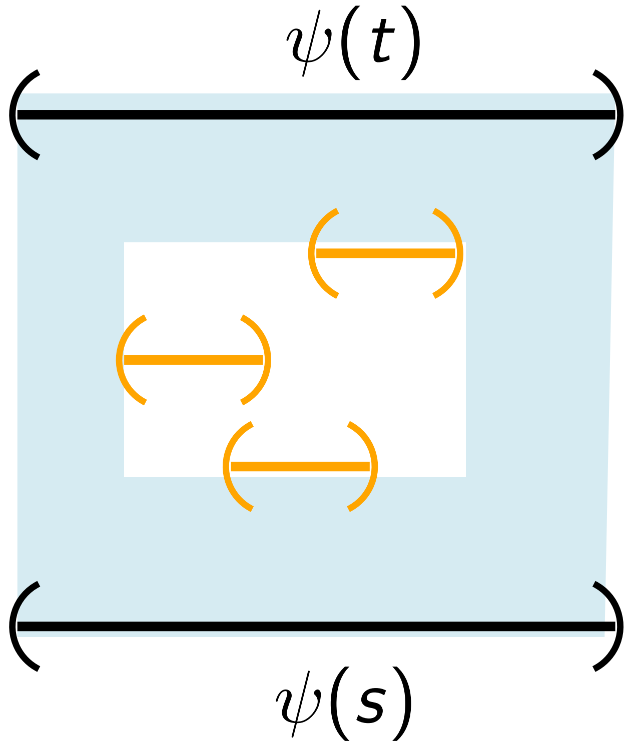
Tiles

Note: orange bars are from the partial representation



Tiles

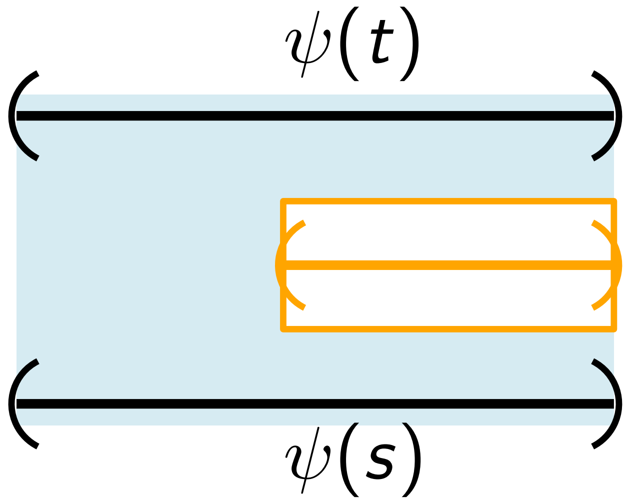
Note: orange bars are from the partial representation



Obs: the bounding box (tile) of any solution ψ, *contains* the bounding box of the partial rep.

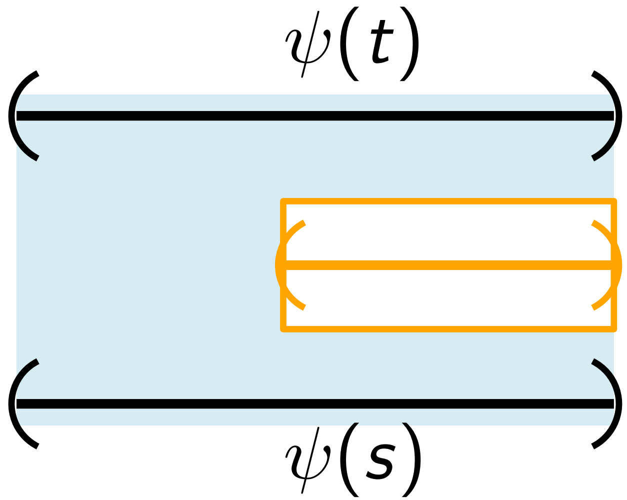
How many **different** tiles can we really have?

Types of Tiles



Right Fixed: due to the orange bar.
Left Loose: due to the orange bar.

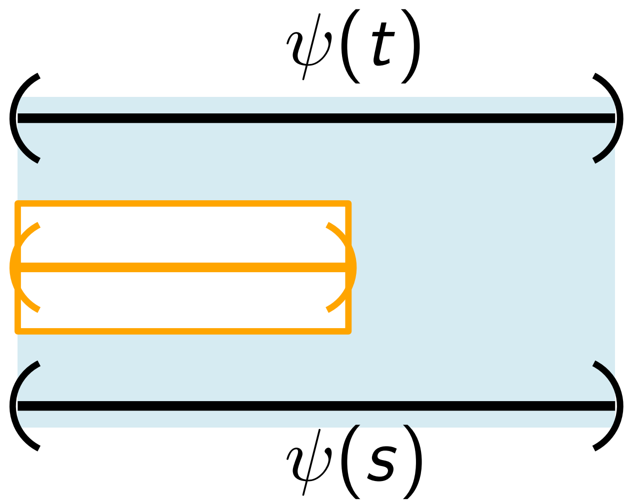
Types of Tiles



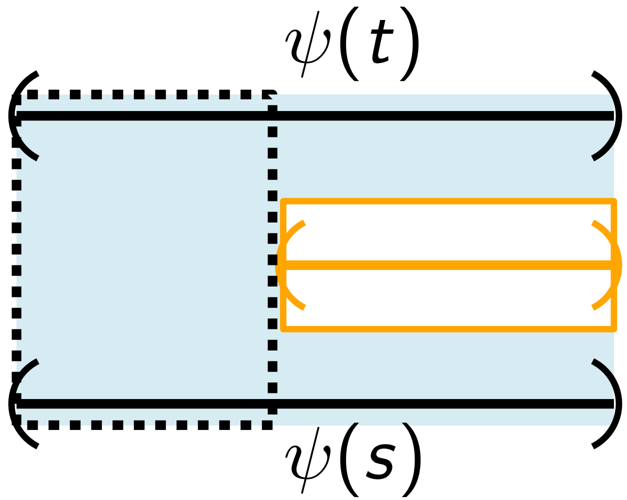
Right Fixed: due to the orange bar.
Left Loose: due to the orange bar.

Left Fixed: due to the orange bar.

Right Loose: due to the orange bar.



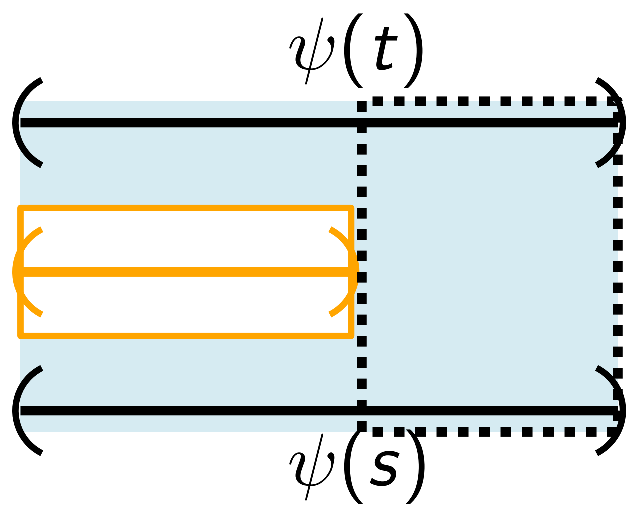
Types of Tiles



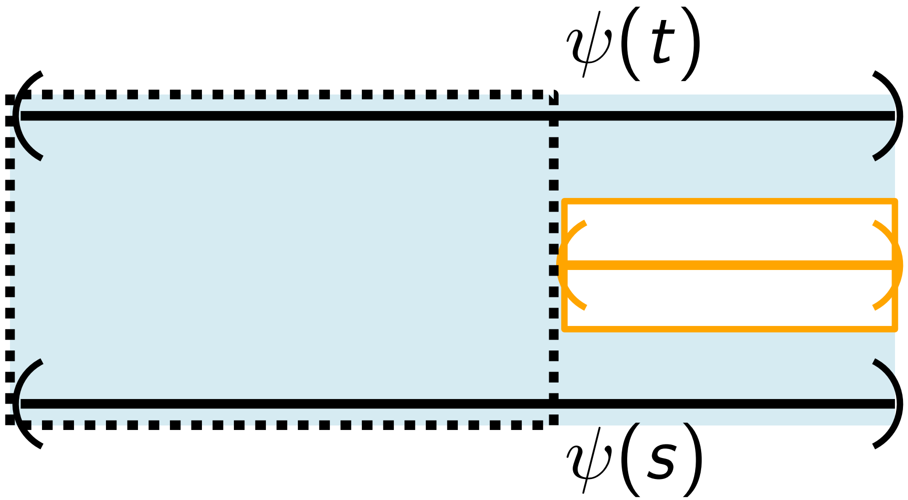
Right Fixed: due to the orange bar.
Left Loose: due to the orange bar.

Left Fixed: due to the orange bar.

Right Loose: due to the orange bar.



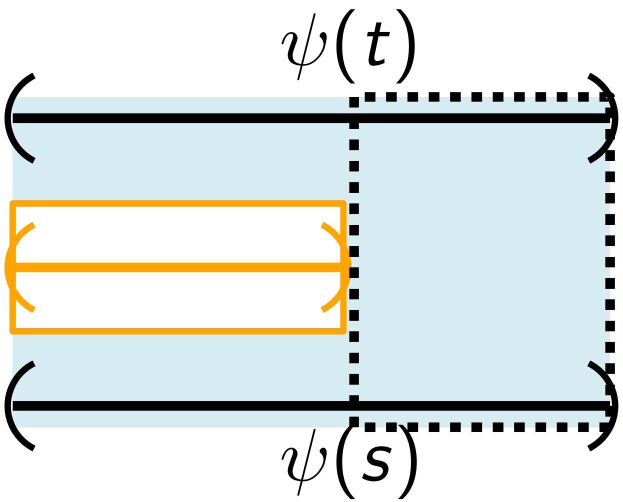
Types of Tiles



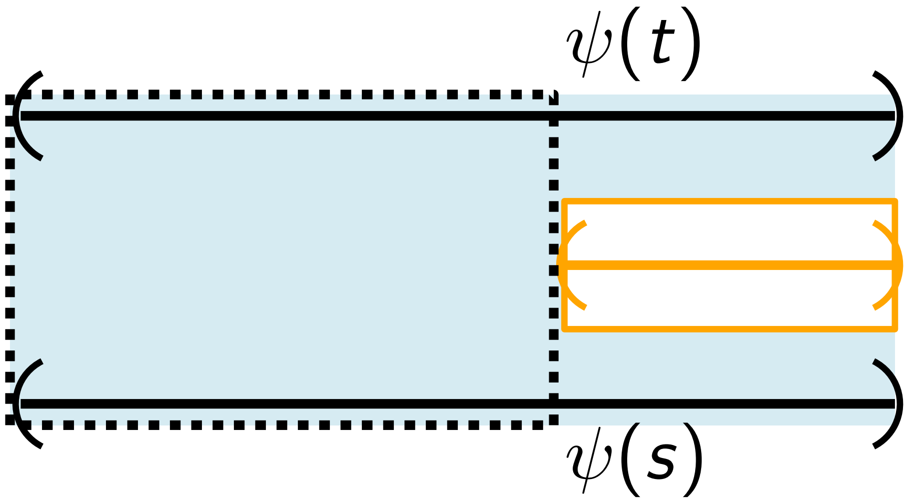
Right Fixed: due to the orange bar.
Left Loose: due to the orange bar.

Left Fixed: due to the orange bar.

Right Loose: due to the orange bar.



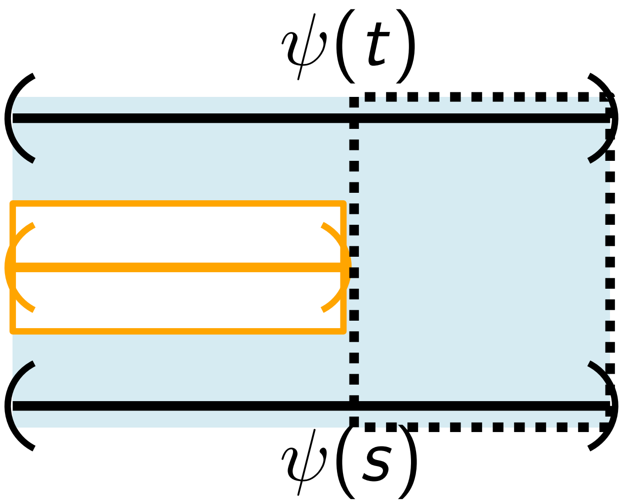
Types of Tiles



Right Fixed: due to the orange bar.
Left Loose: due to the orange bar.

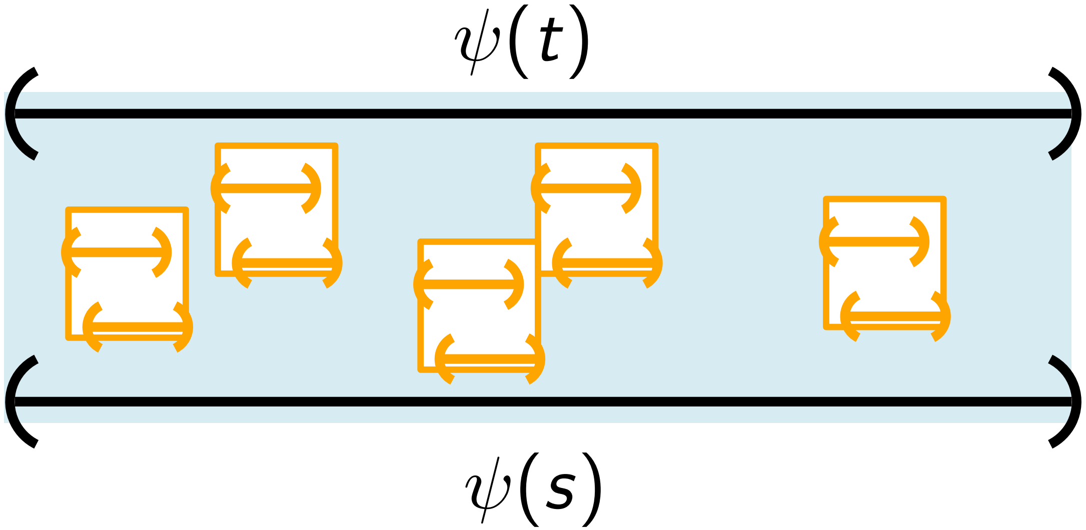
Left Fixed: due to the orange bar.

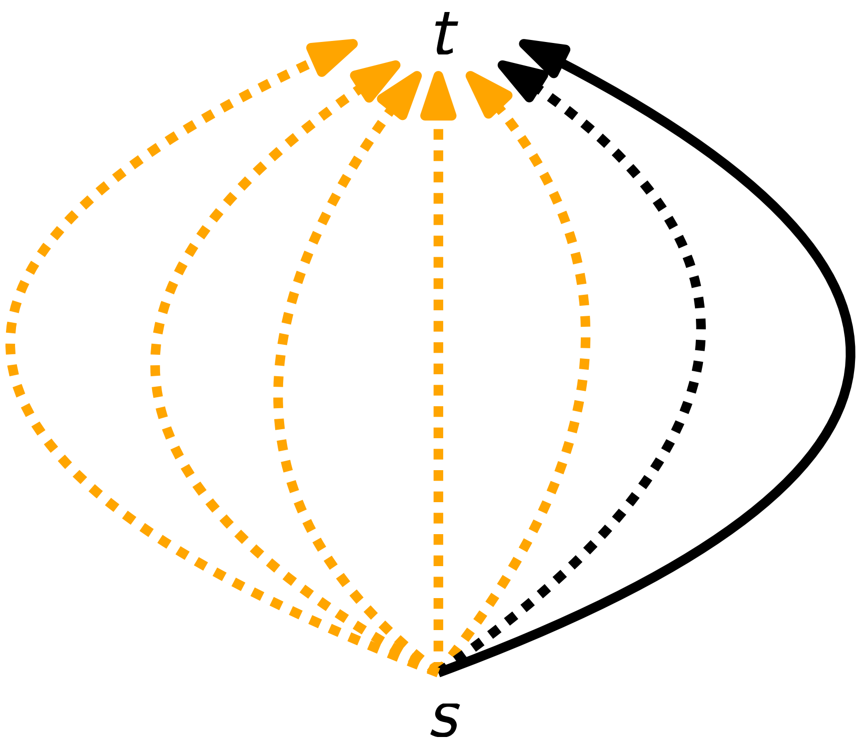
Right Loose: due to the orange bar.



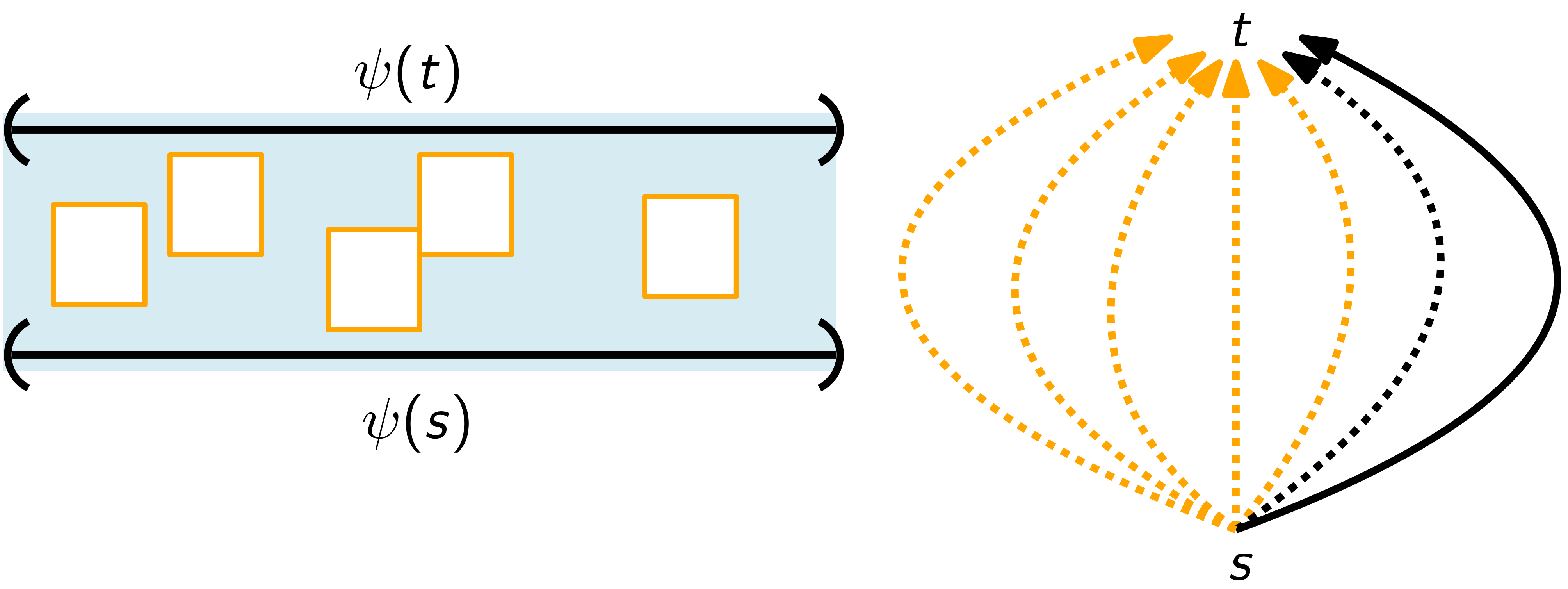
Four different Types: FF, FL, LF, LL

P-nodes

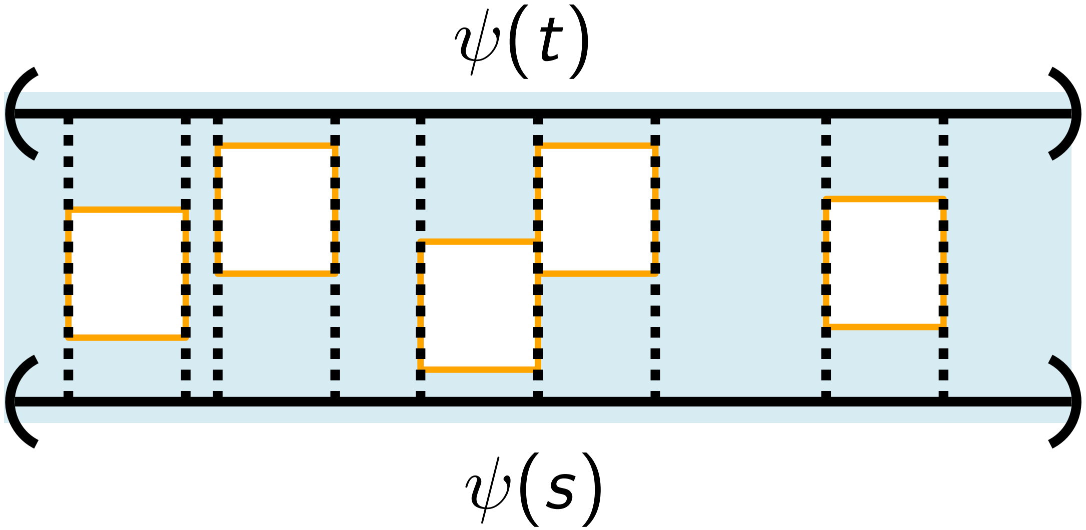


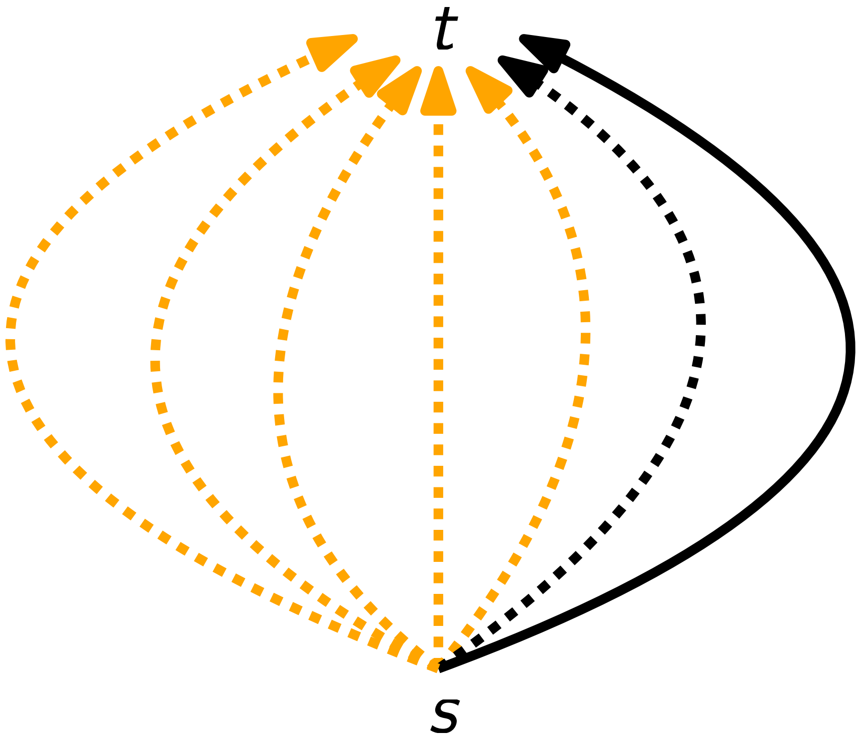


P-nodes

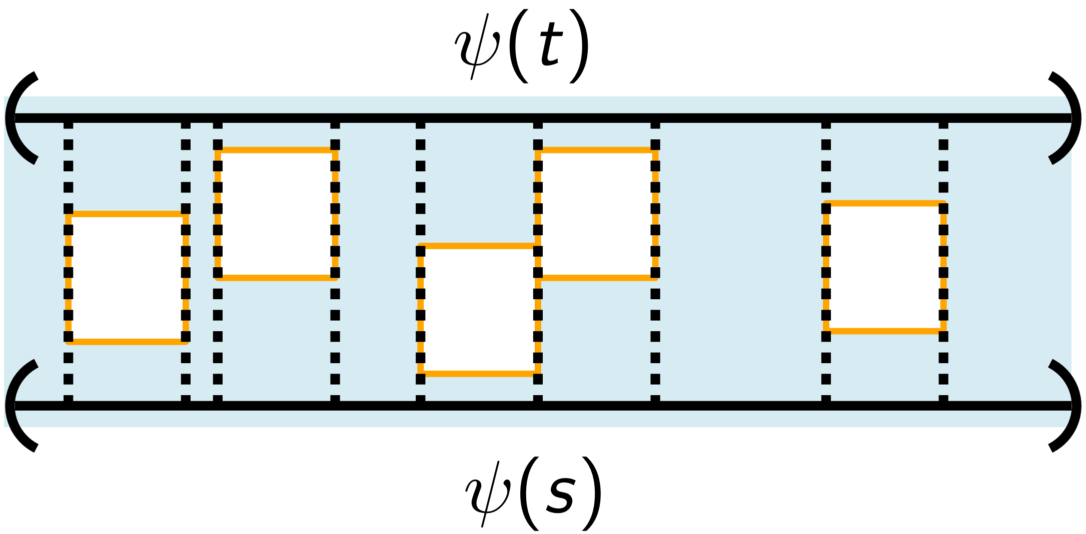


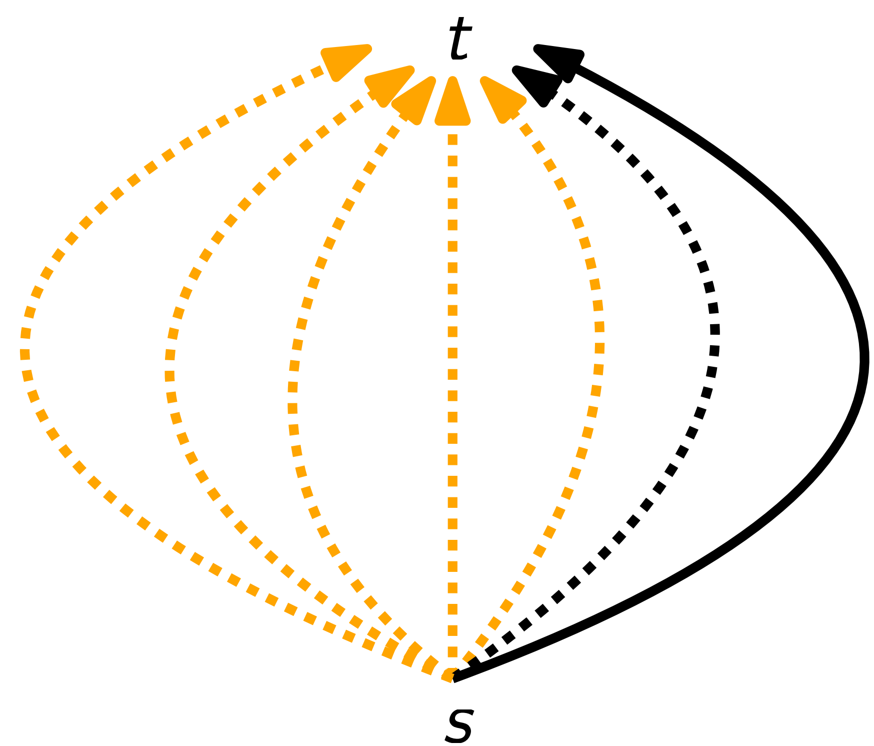
P-nodes





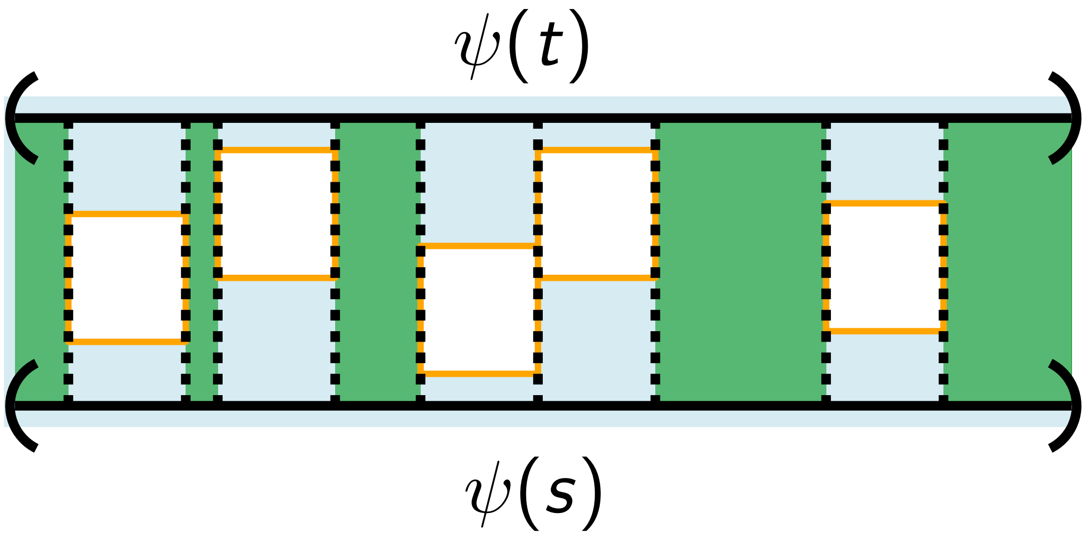
P-nodes

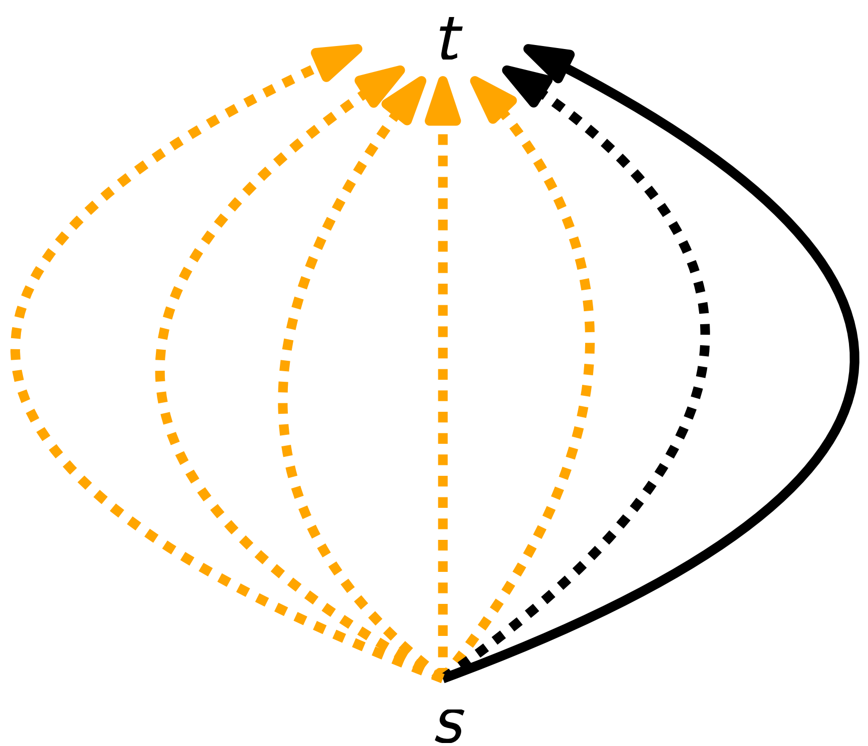




Children with prescribed bars occur in given left-to-right order
But there will be some gaps..

P-nodes

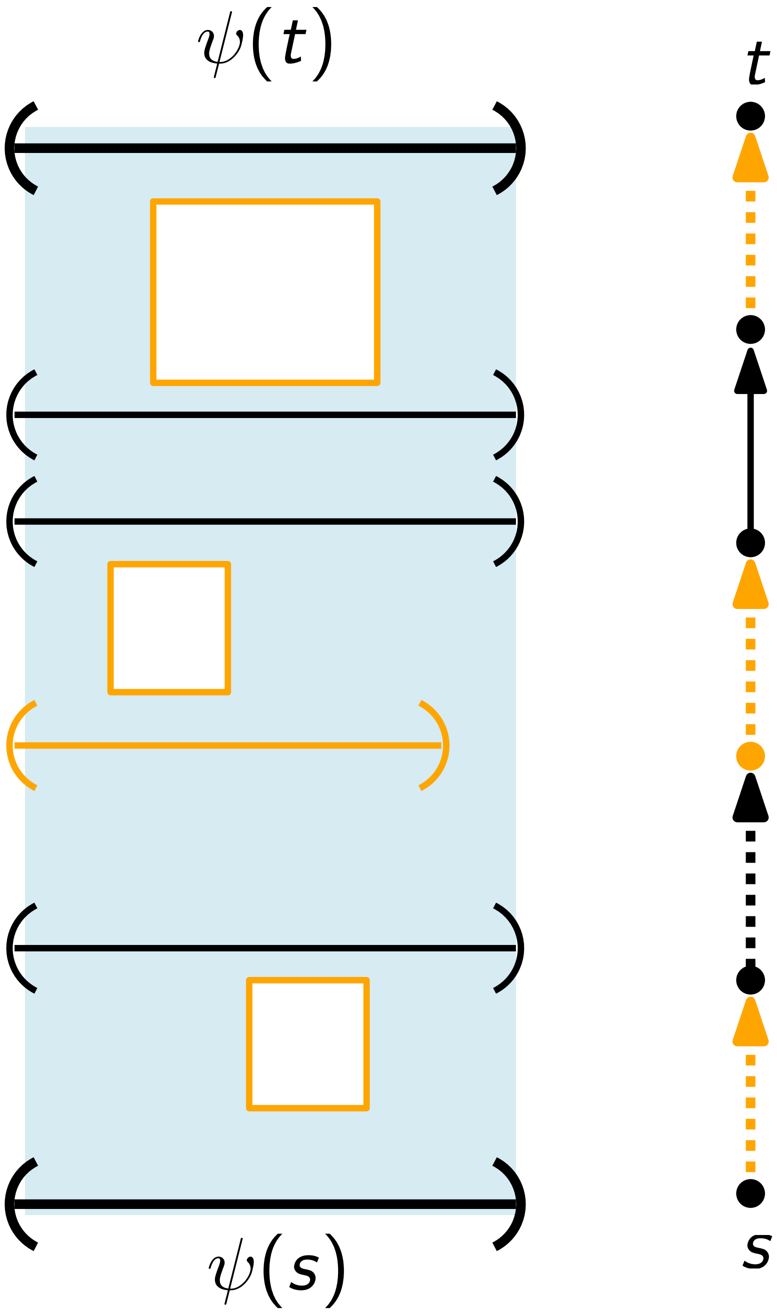




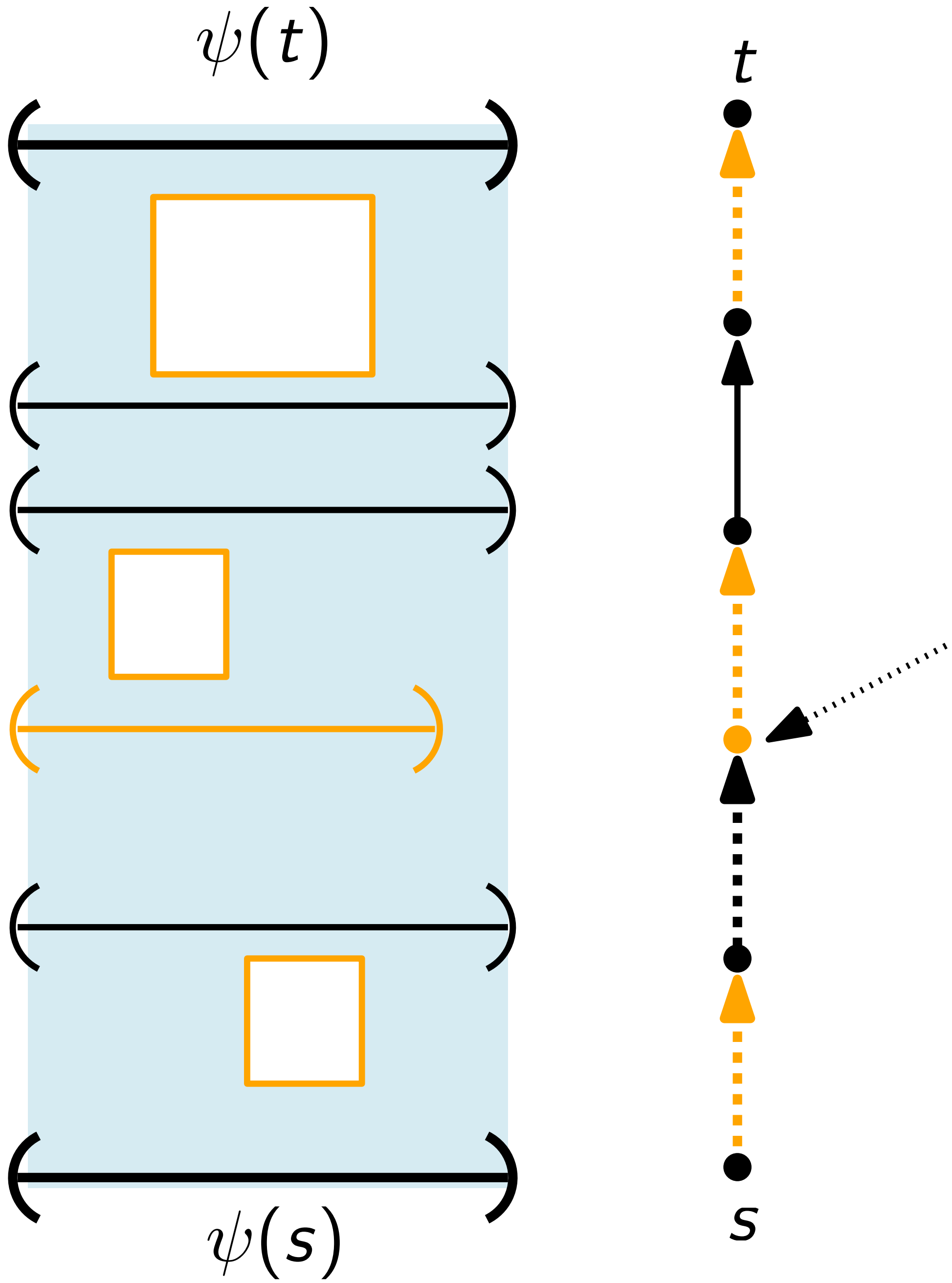
Children with prescribed bars occur in given left-to-right order
But there will be some gaps..

Idea: greedily *fill* the gaps by preferring to “stretch” the children with prescribed bars.

S-nodes

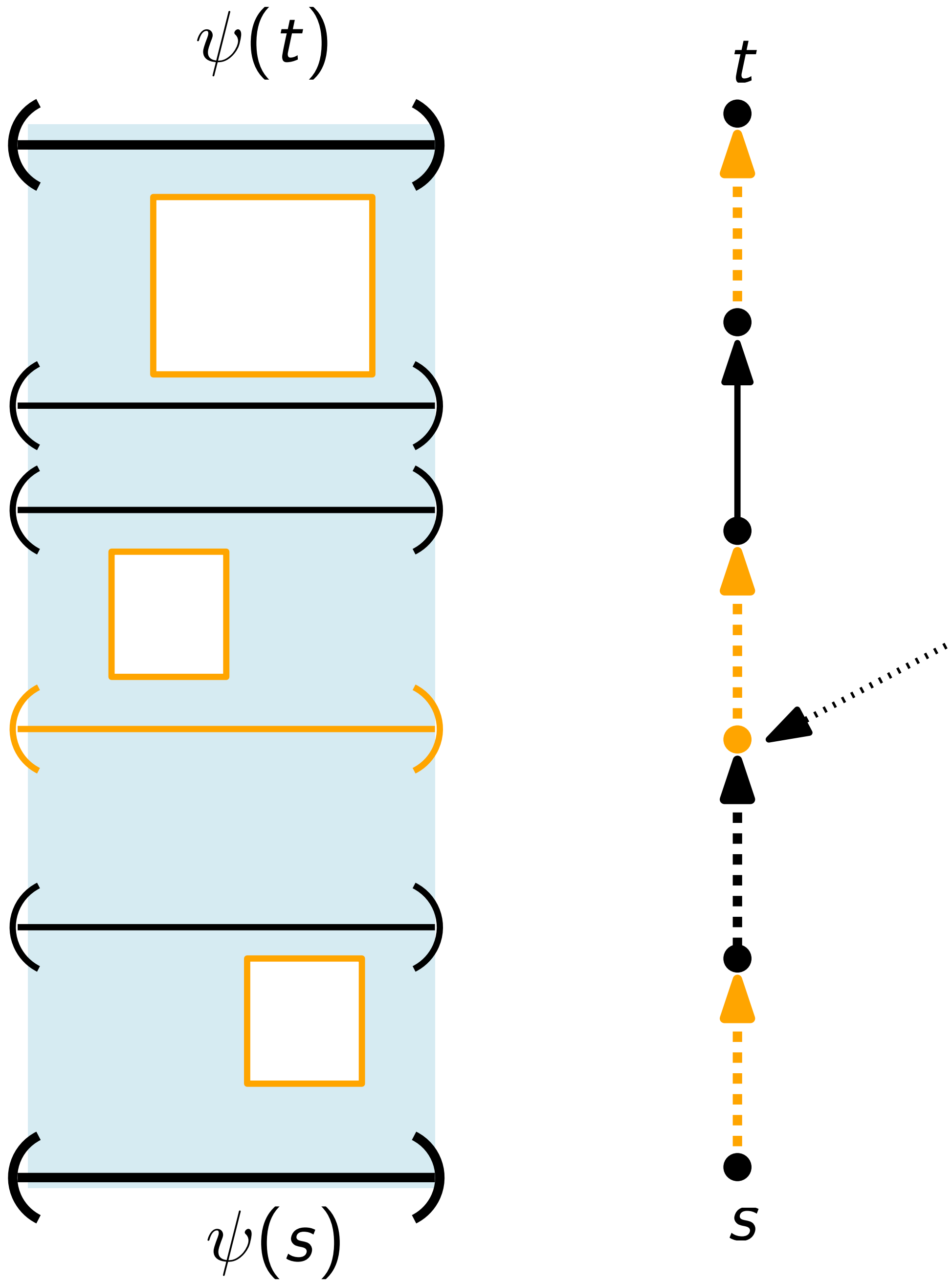


S-nodes



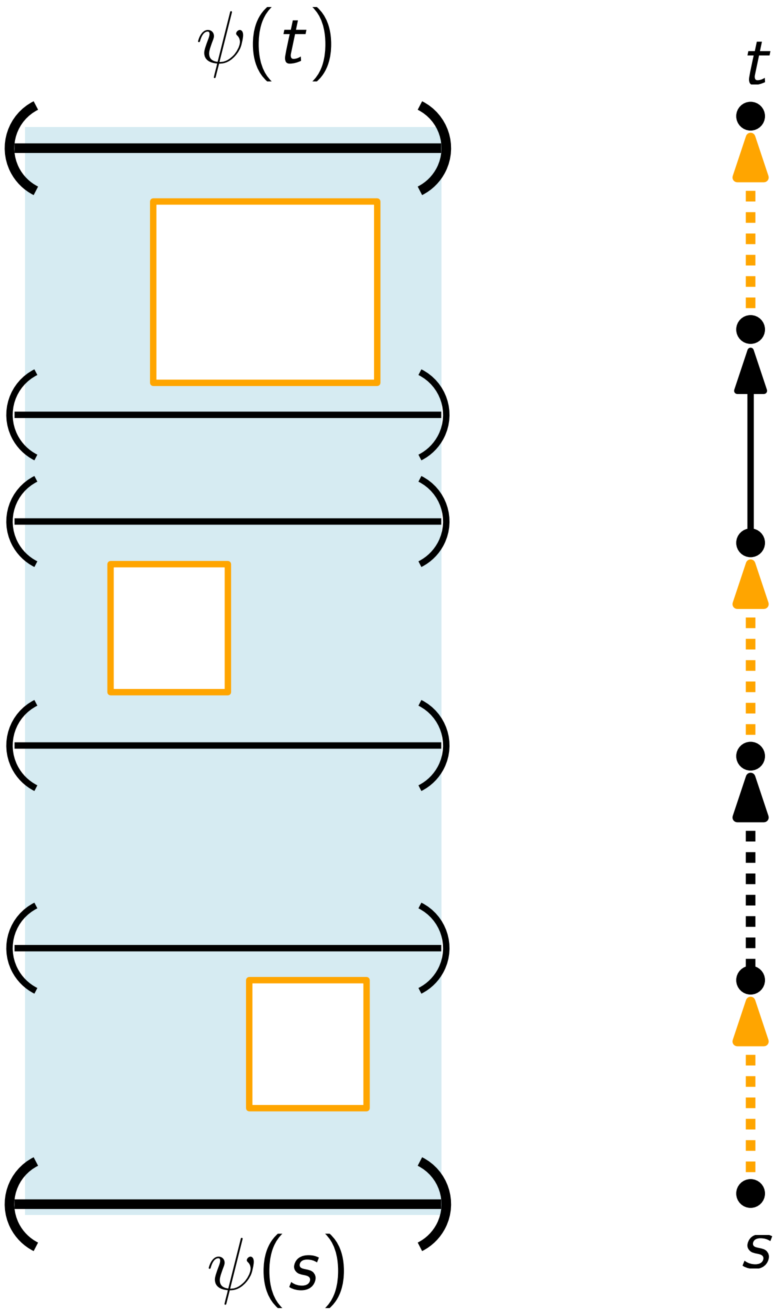
This fixed vertex means we can only make a Fixed-Fixed representation!

S-nodes



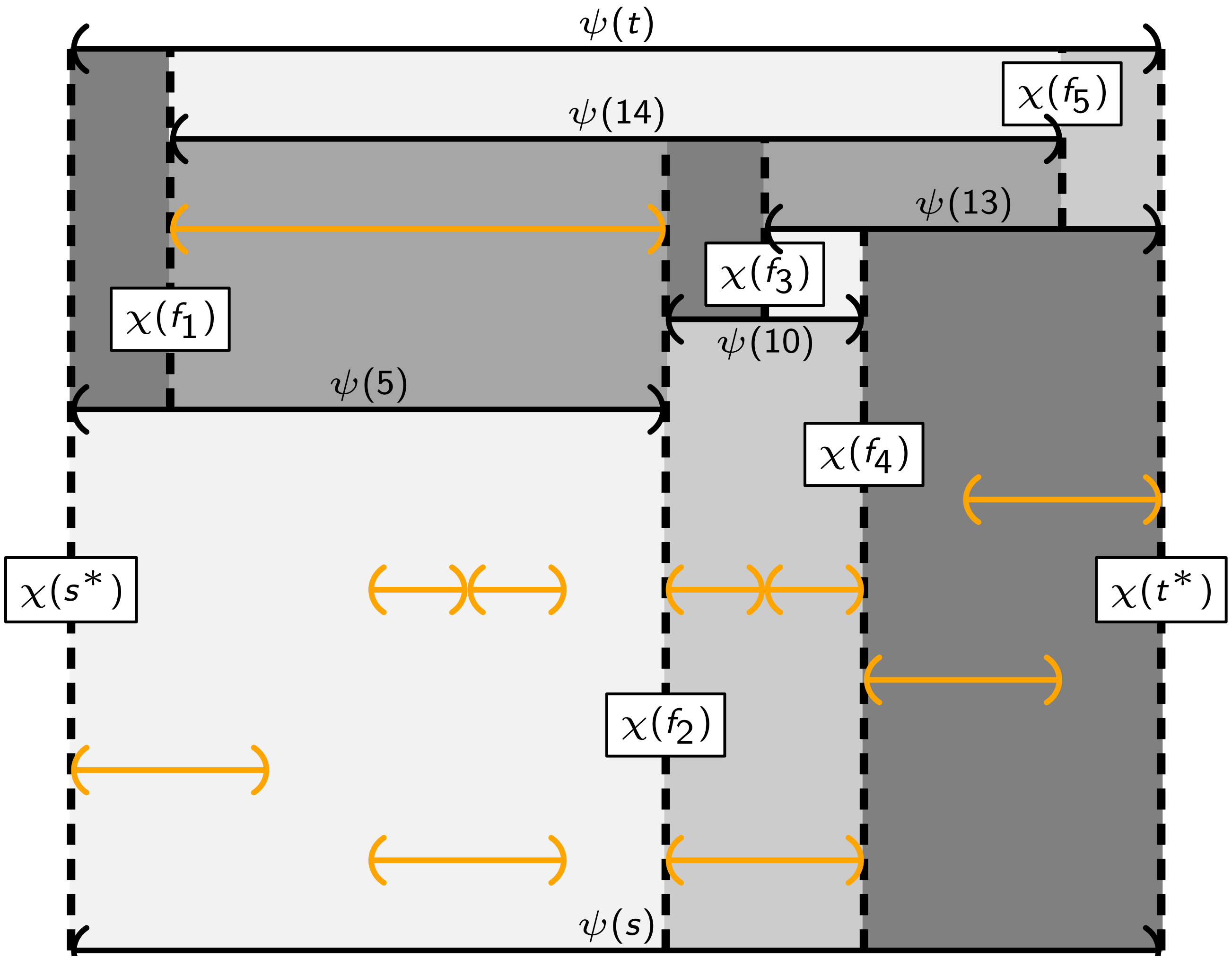
This fixed vertex means we can only make a Fixed-Fixed representation!

S-nodes

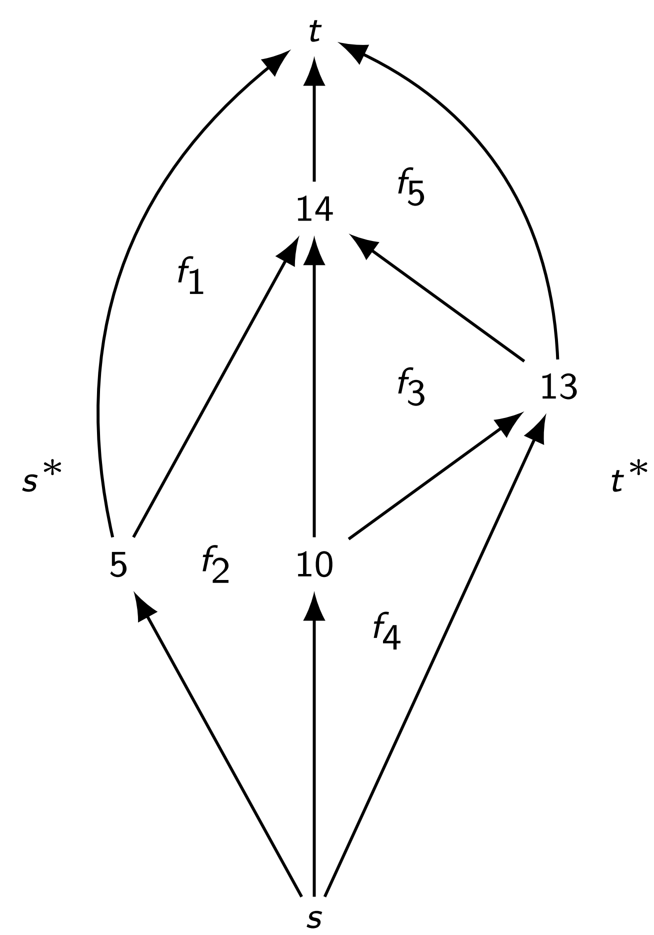


Now, we have a chance to make LL, FL, LF types

R-nodes



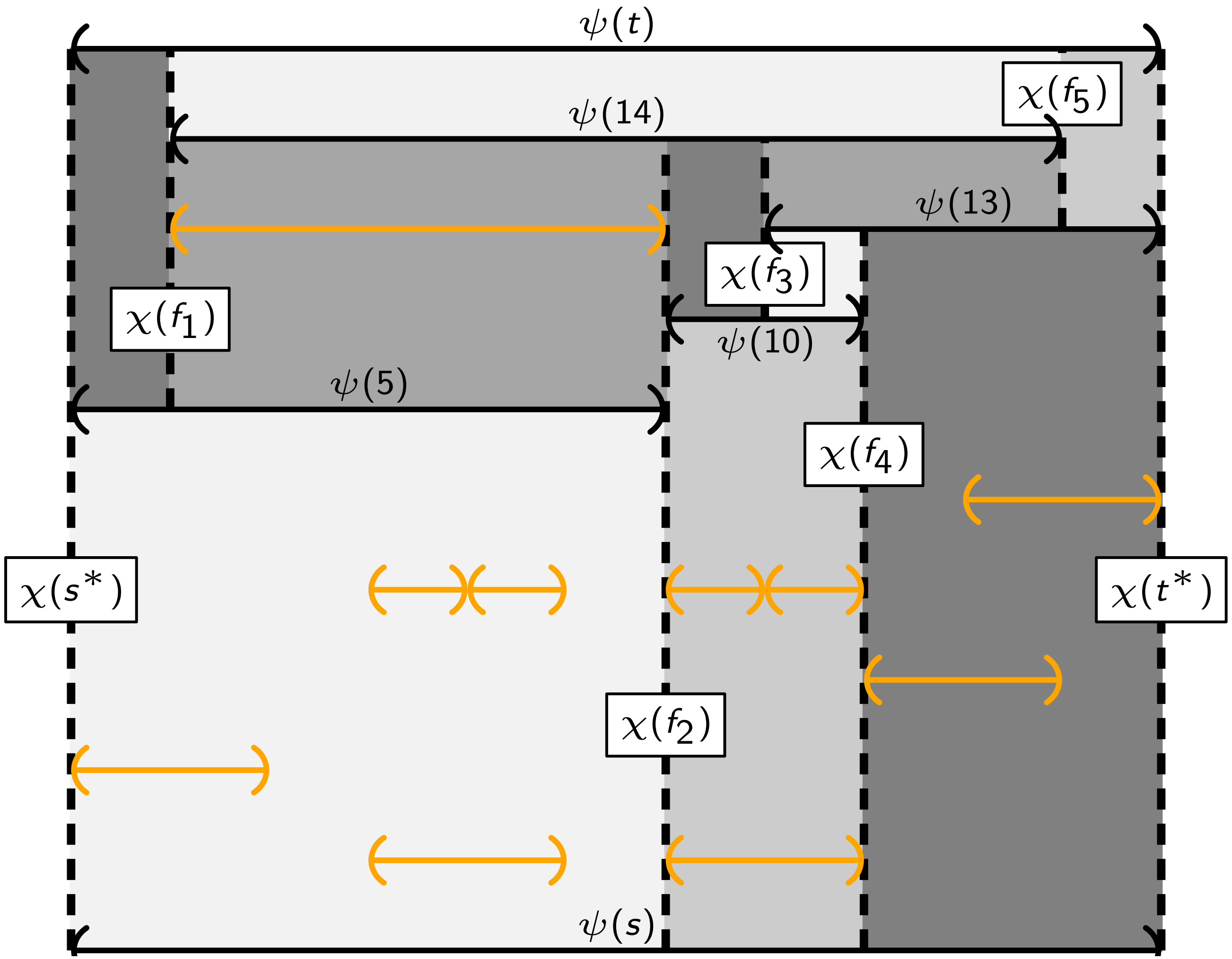
$\psi(t)$
$\chi(f_5)$
$\psi(14)$
$\psi(13)$
$\chi(f_3)$
$\chi(f_1)$
$\psi(10)$
$\psi(5)$
$\chi(f_4)$
$\chi(s^*)$
$\chi(t^*)$
$\chi(f_2)$
$\psi(s)$

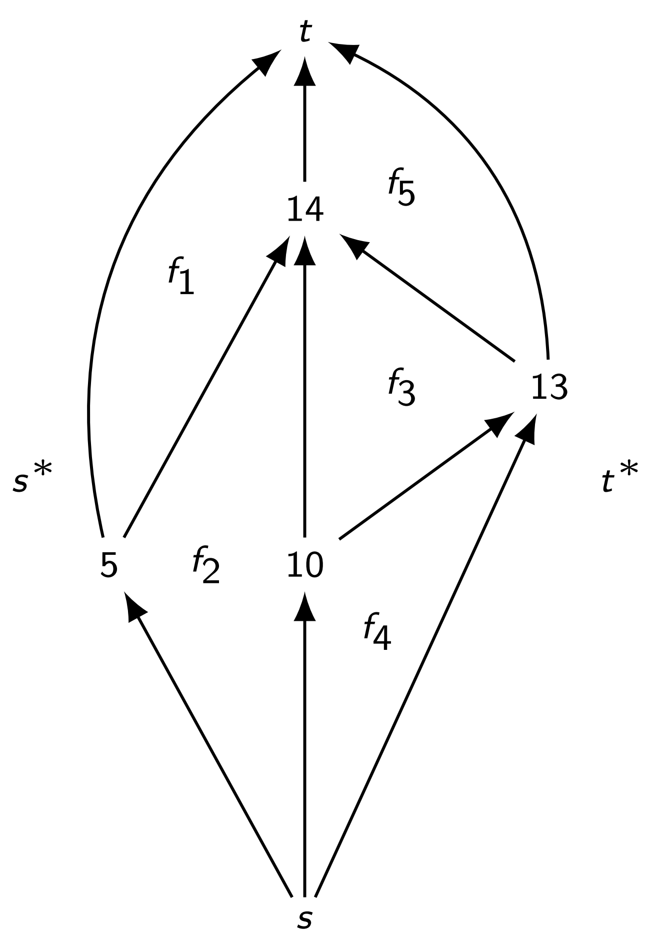


t
f_5
14
f_1
f_3
13
s^*
t^*
5
f_2
10
f_4
s

R-nodes

2-SAT formulation:

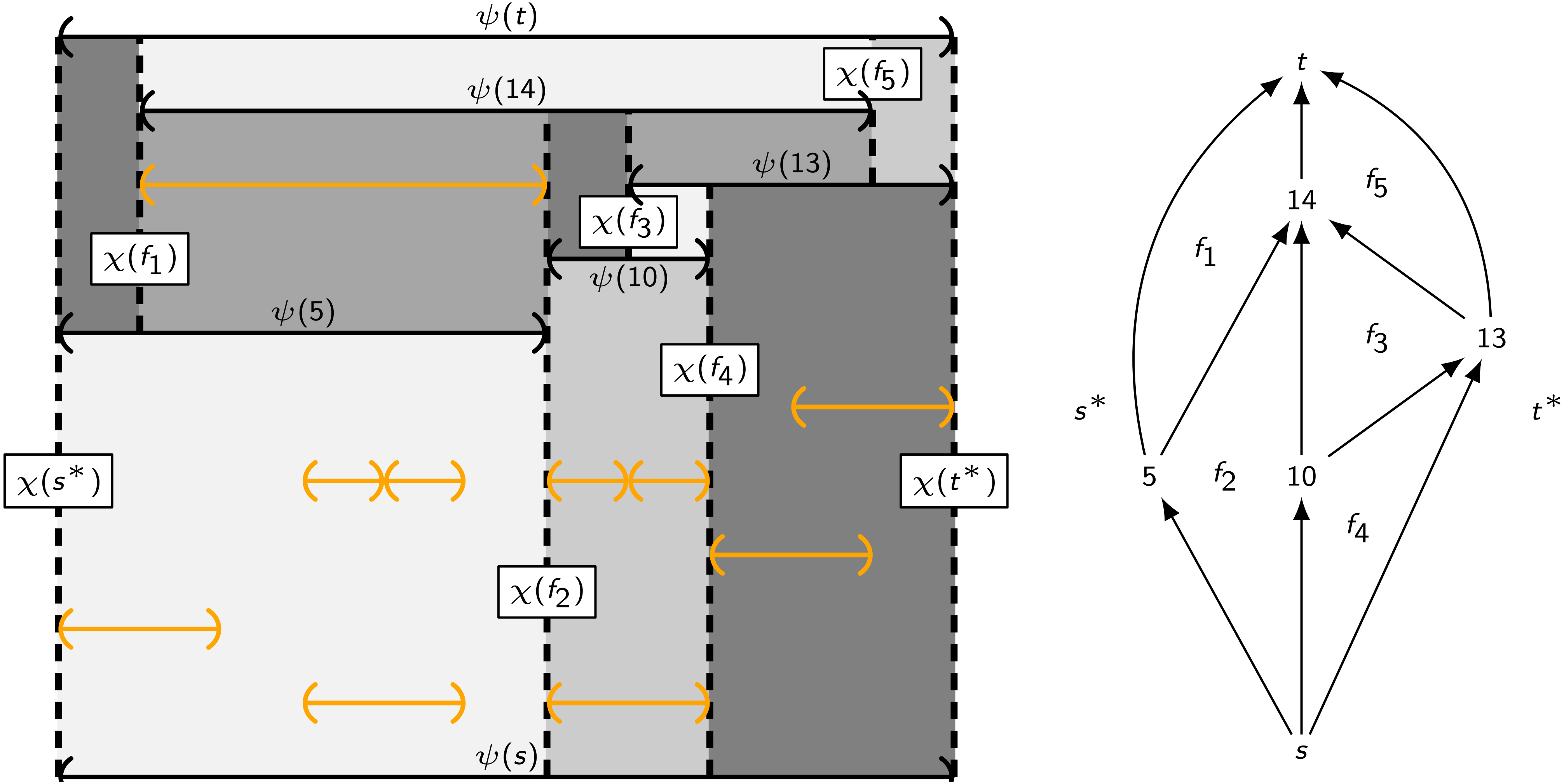




R-nodes

2-SAT formulation:

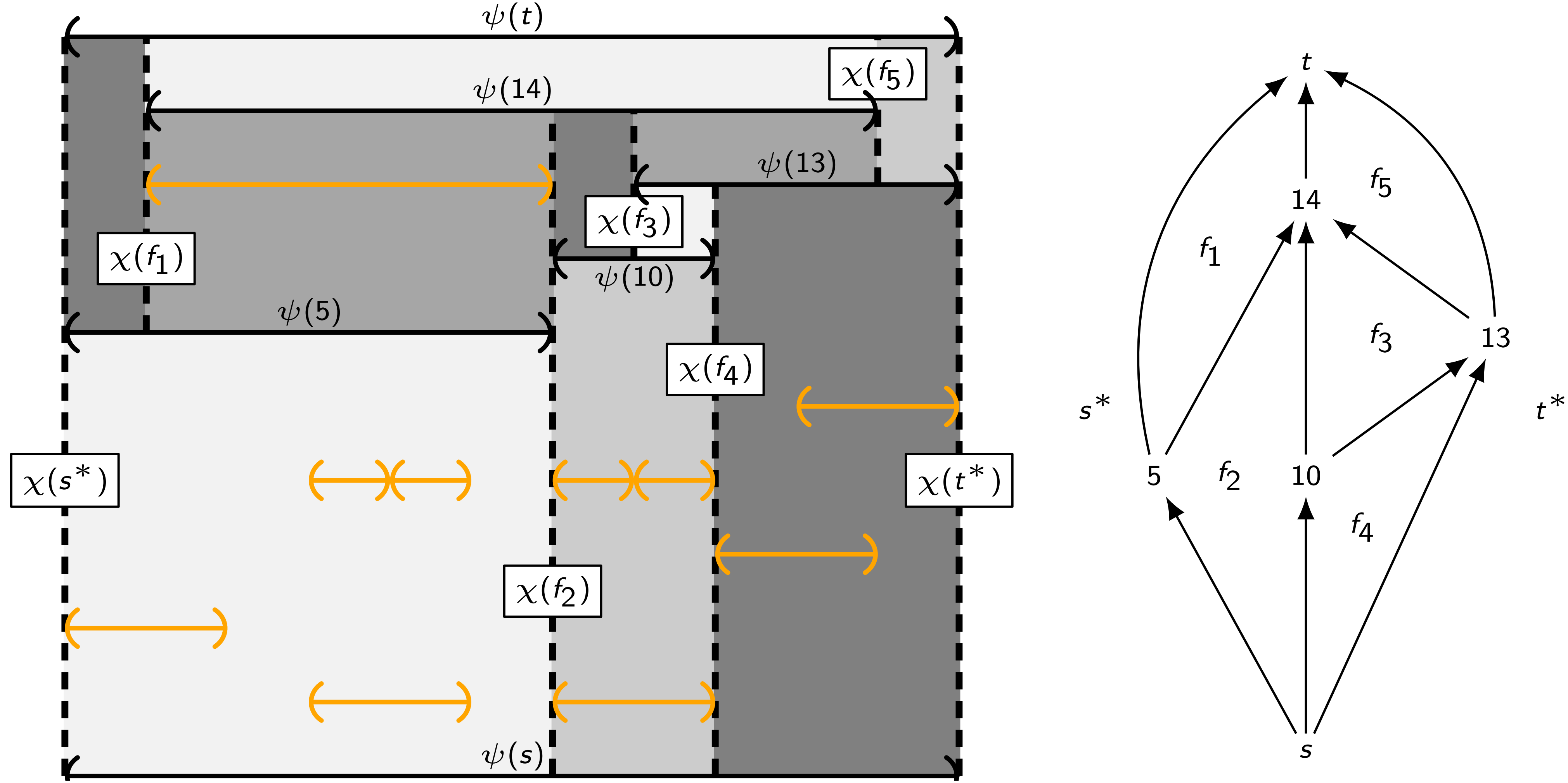
- 2 varaibles for each child: encoding fixed/loose state of its tile.
- 2 variables for each face encoding position of the splitting line.



R-nodes

2-SAT formulation:

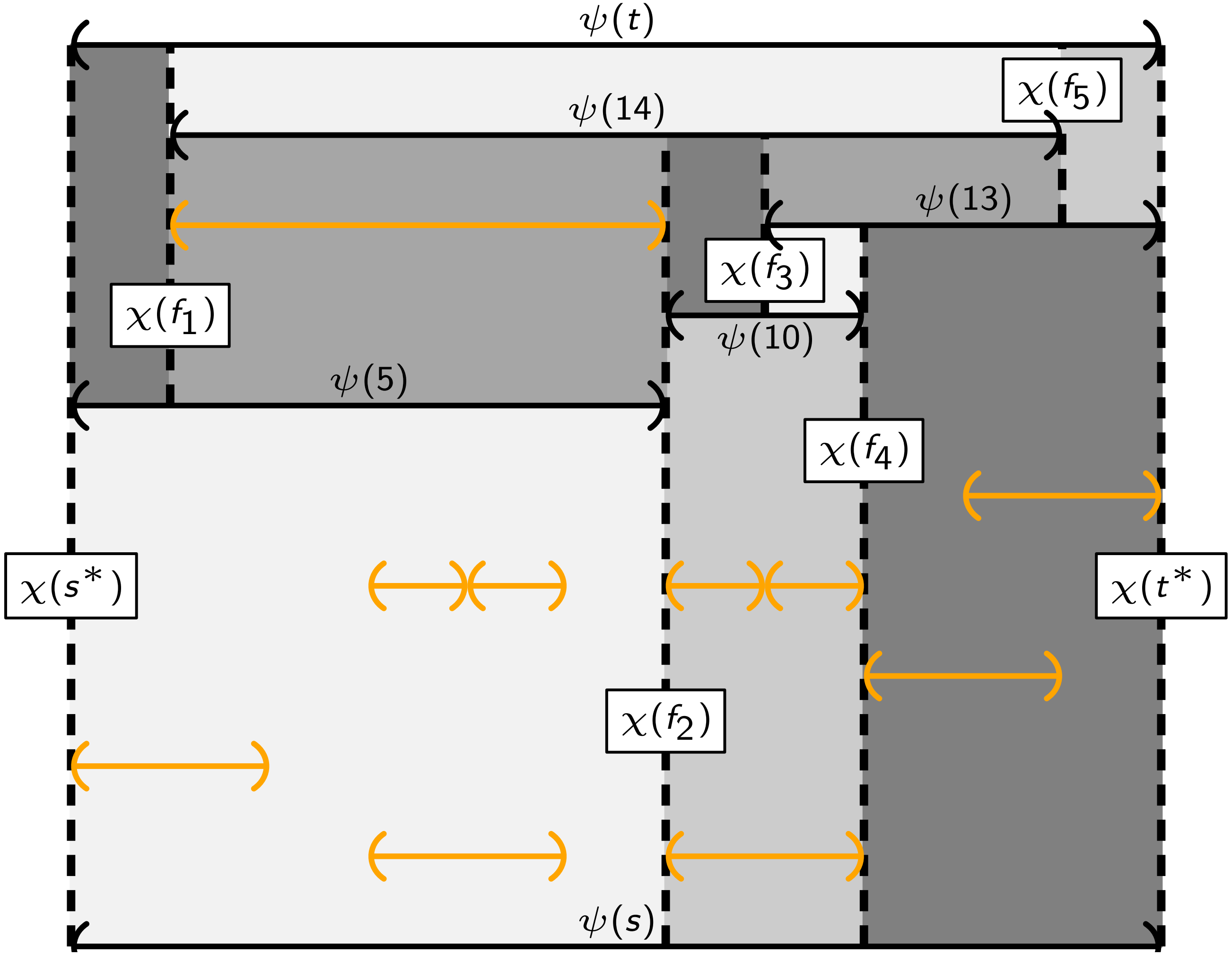
- 2 varaibles for each child: encoding fixed/loose state of its tile.
- 2 variables for each face encoding position of the splitting line.
- restriction clauses for each child to subset of {FF,FL,LF,LL}

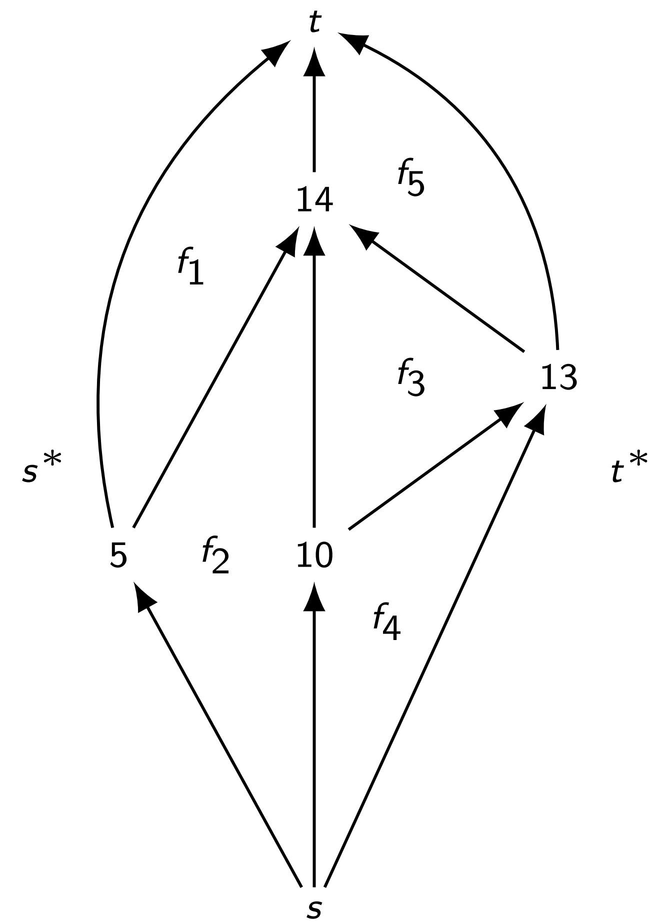


R-nodes

2-SAT formulation:

- 2 varaibles for each child: encoding fixed/loose state of its tile.
- 2 variables for each face encoding position of the splitting line.
- restriction clauses for each child to subset of {FF,FL,LF,LL}
- consistency clauses for each face.

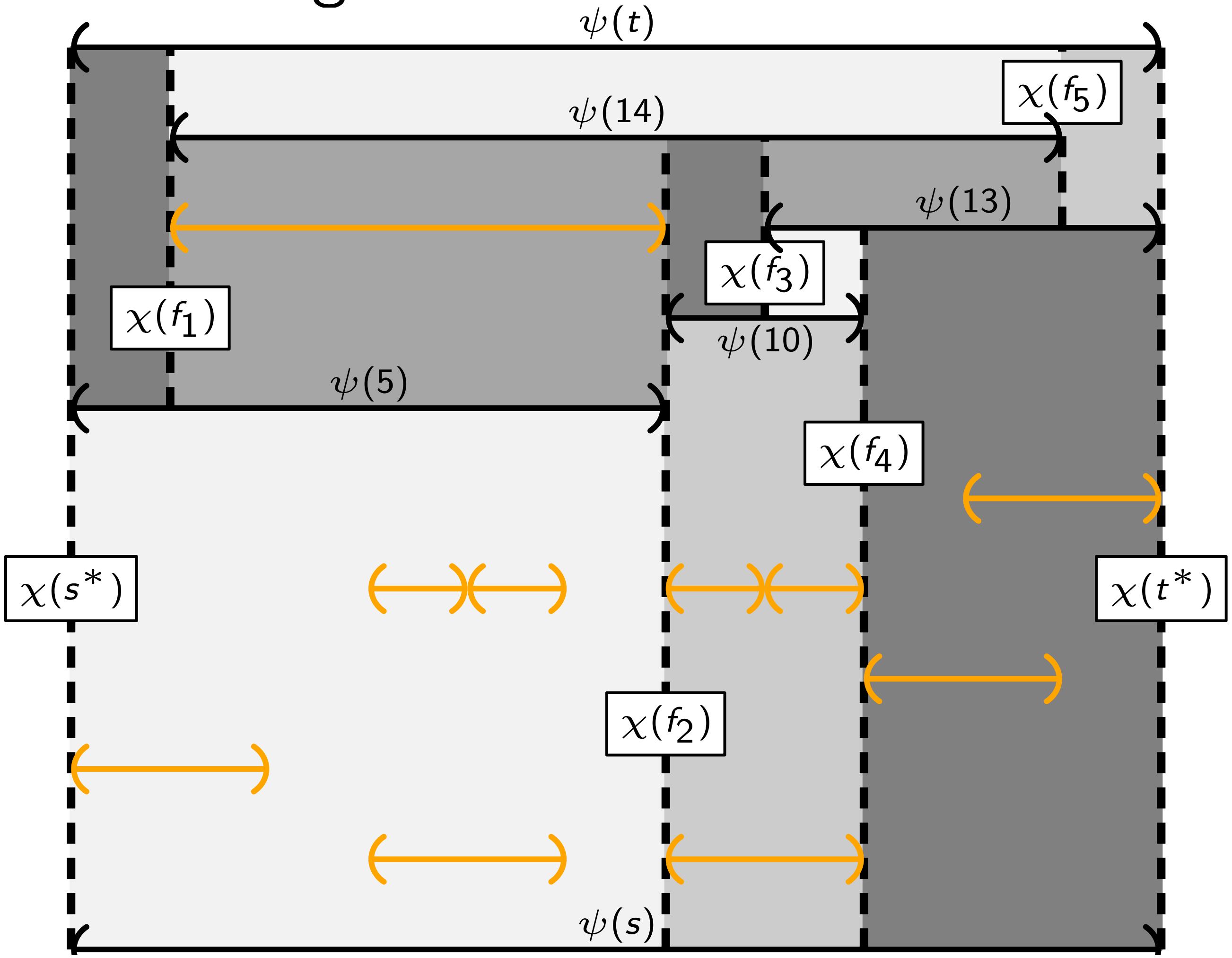


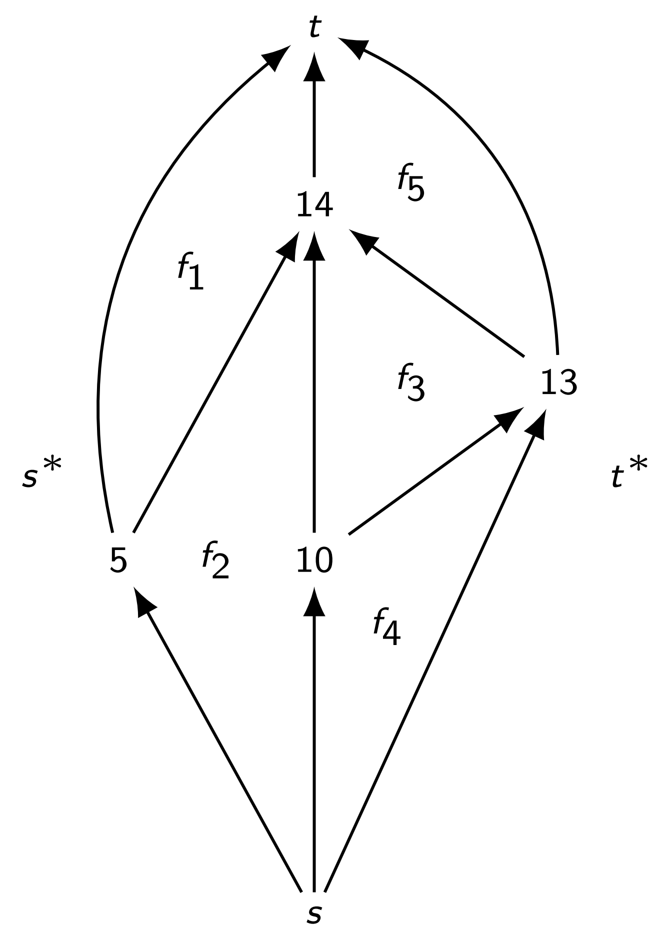


R-nodes

2-SAT formulation:

- 2 varaibles for each child: encoding fixed/loose state of its tile.
- 2 variables for each face encoding position of the splitting line.
- restriction clauses for each child to subset of {FF,FL,LF,LL}
- consistency clauses for each face.
- ordering clauses.

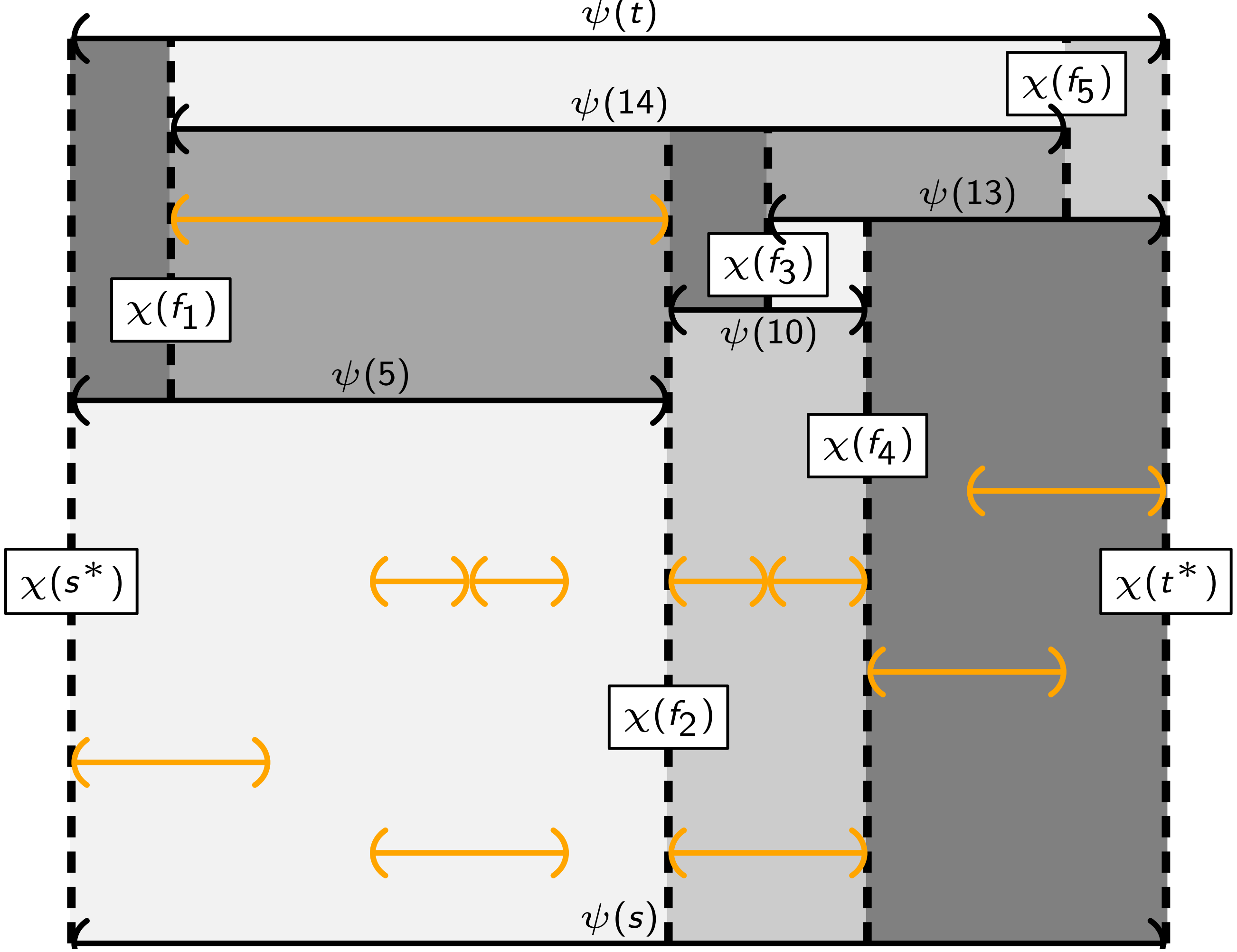


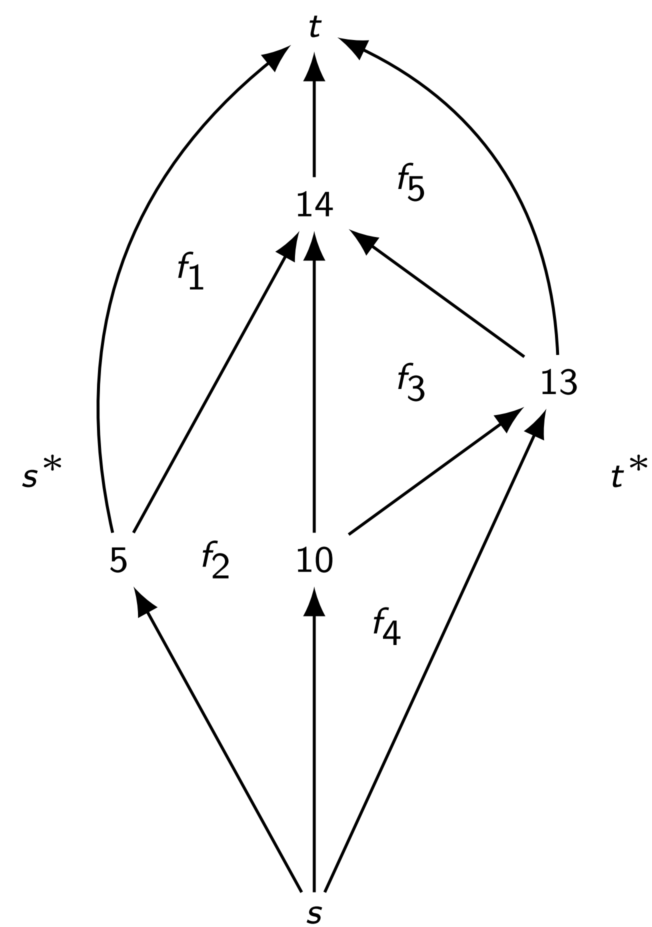


R-nodes

2-SAT formulation:

- 2 varaibles for each child: encoding fixed/loose state of its tile.
- 2 variables for each face encoding position of the splitting line.
- restriction clauses for each child to subset of {FF,FL,LF,LL}
- consistency clauses for each face.
- ordering clauses. ← **Quadratically many :-(**





Hardness Results

ε-Bar Visibility Representation Extension is NP-complete

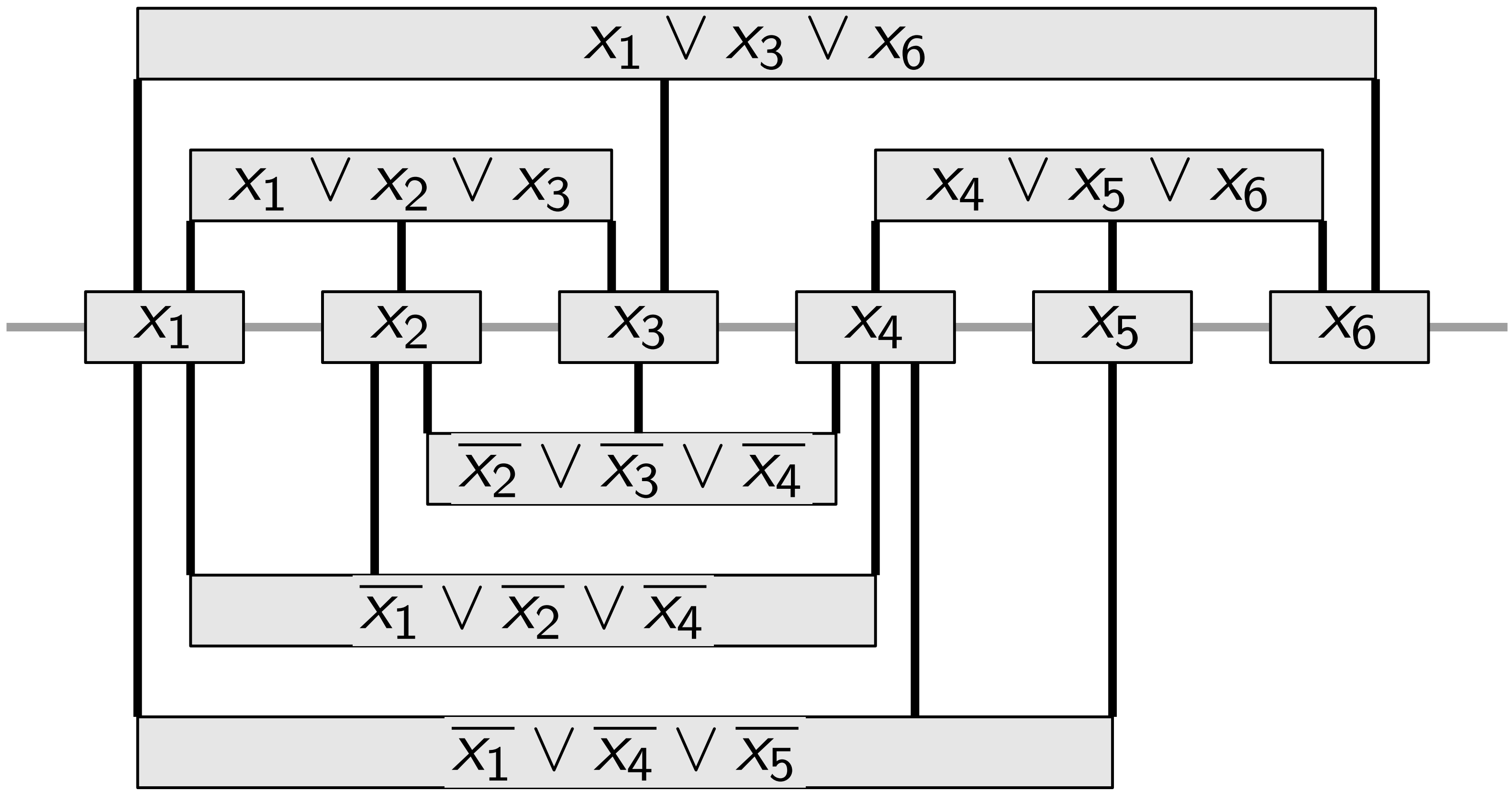
↳ Reduction: *planar monotone 3-SAT*

ε-Bar Visibility Representation Extension is NP-complete for (series-parallel) *st*-graphs when restricted to the *Integer Grid* (or if any fixed $\varepsilon > 0$ is specified).

↳ Reduction: *3-partition*

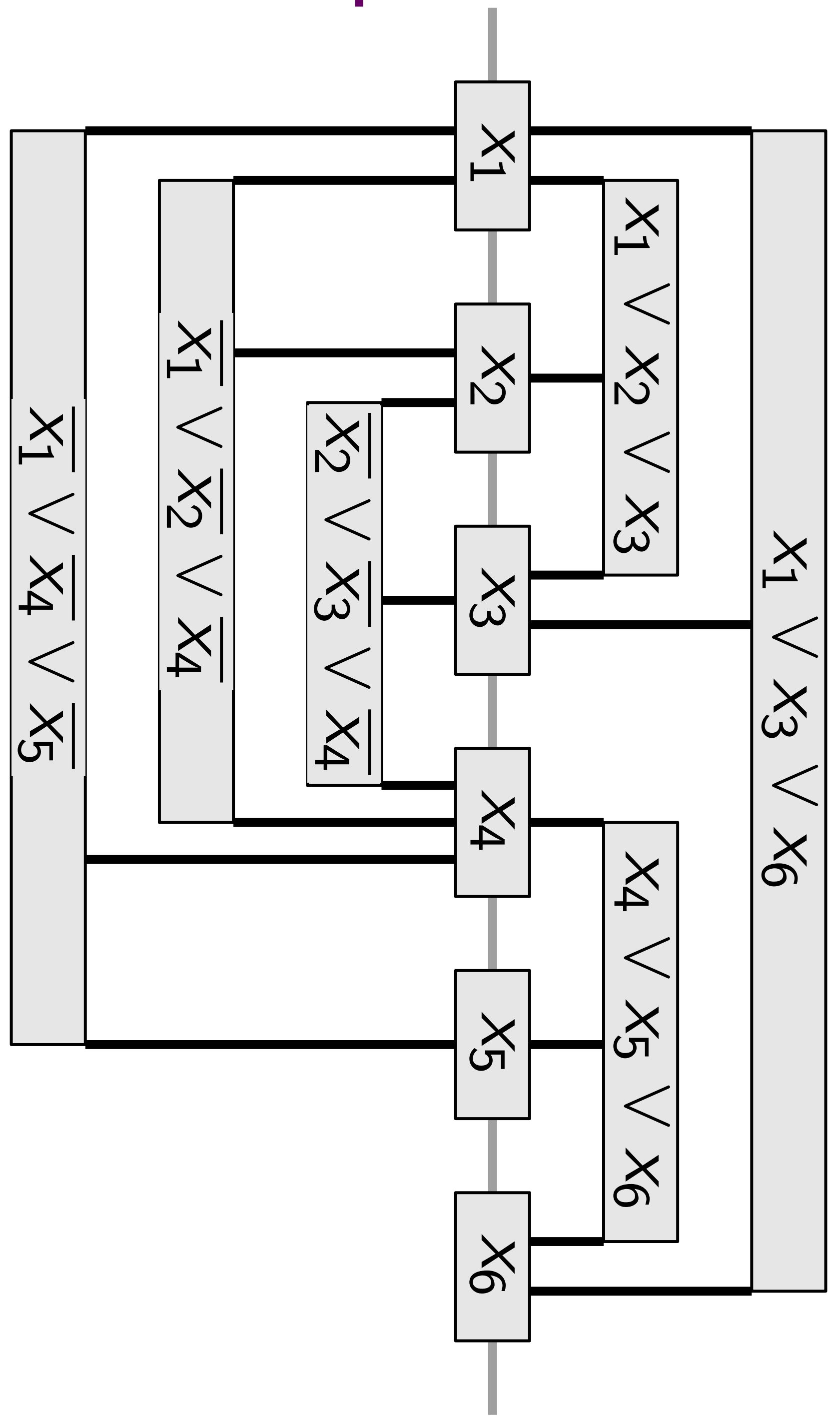
NP-hardness of Representation Extension

Planar Monotone 3-SAT

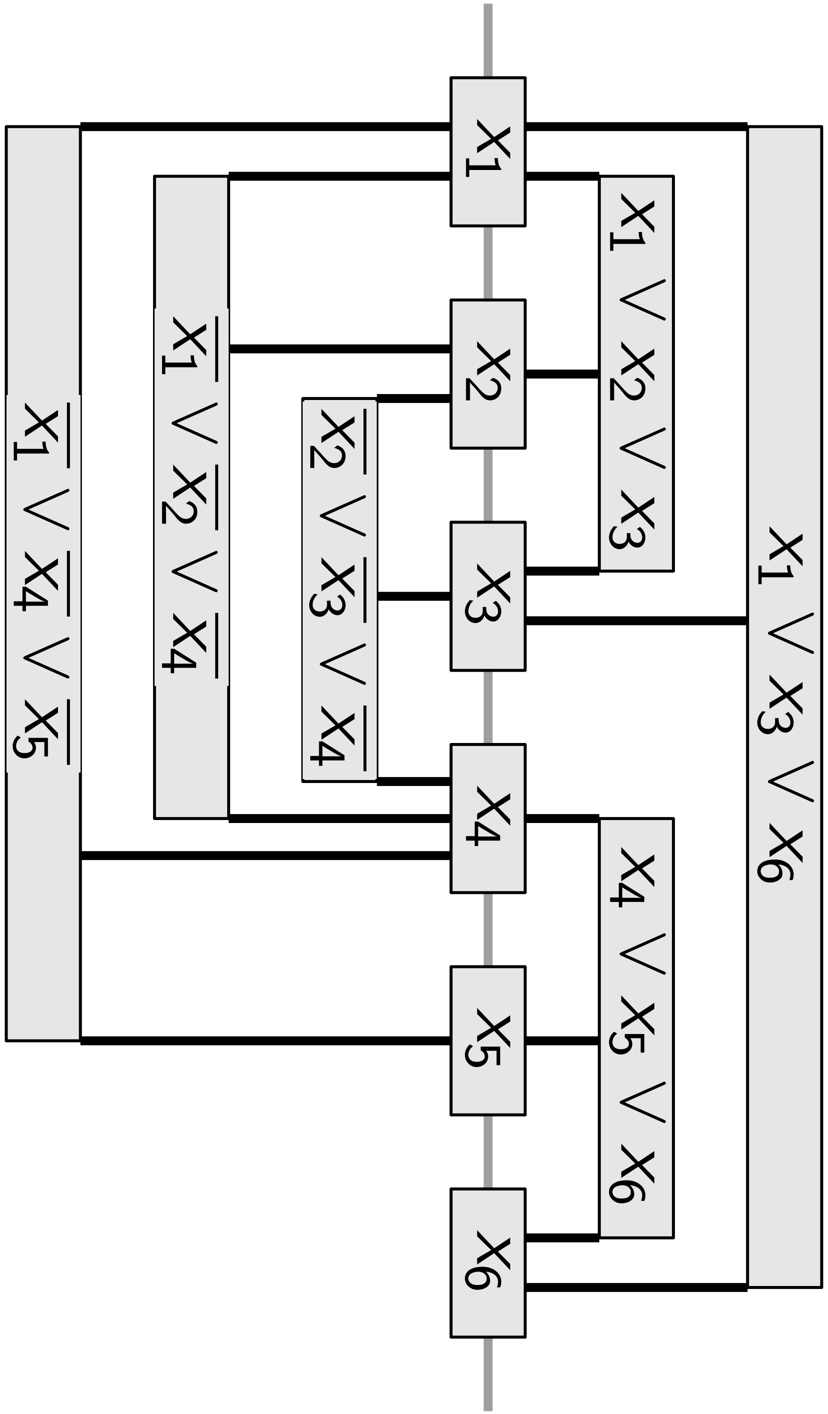


NP-complete [Berg and Khosravi 2010]

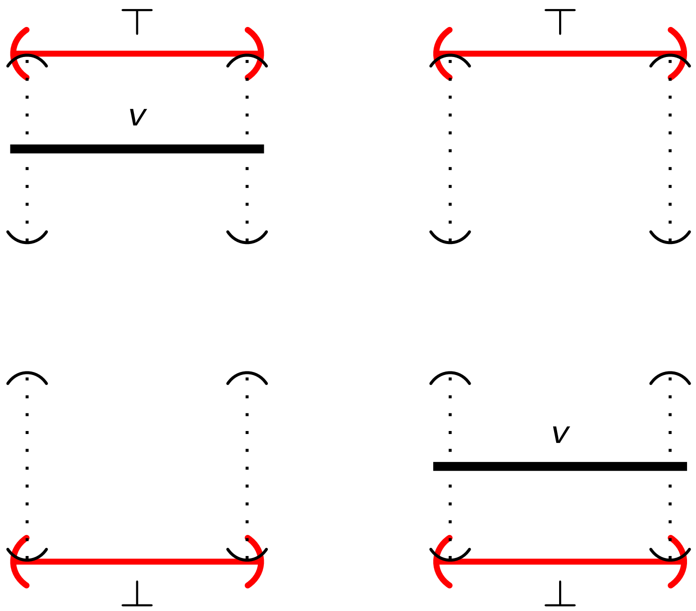
NP-hardness of Representation Extension



NP-hardness of Representation Extension

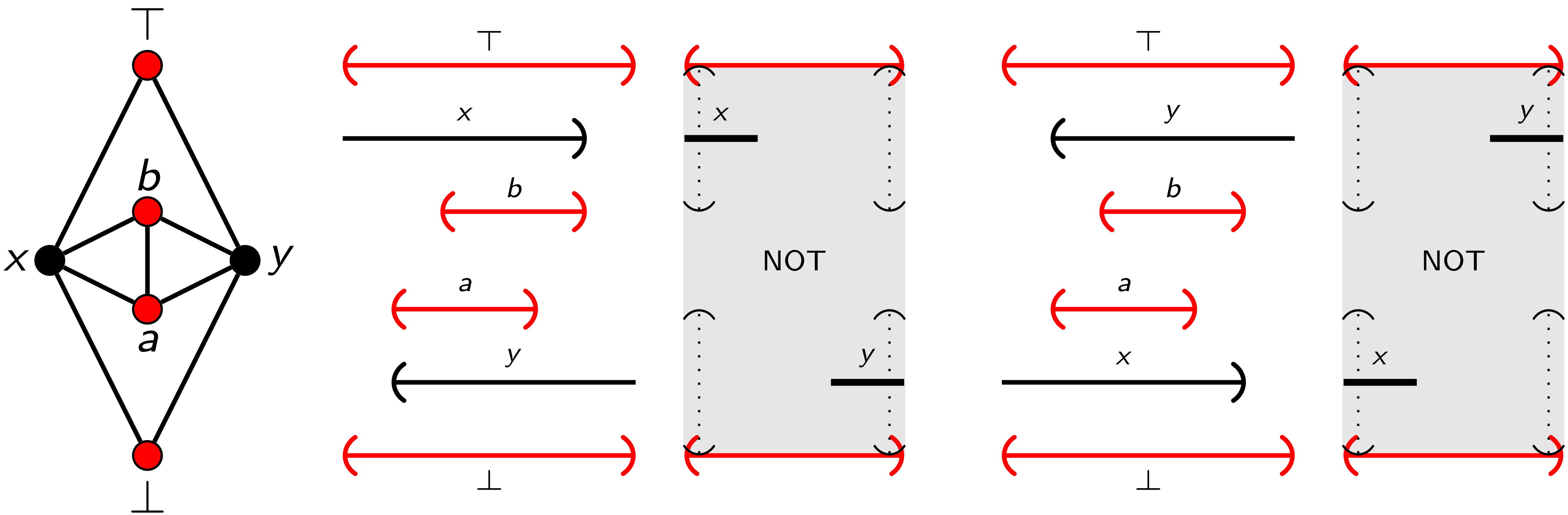


Wire Transmission



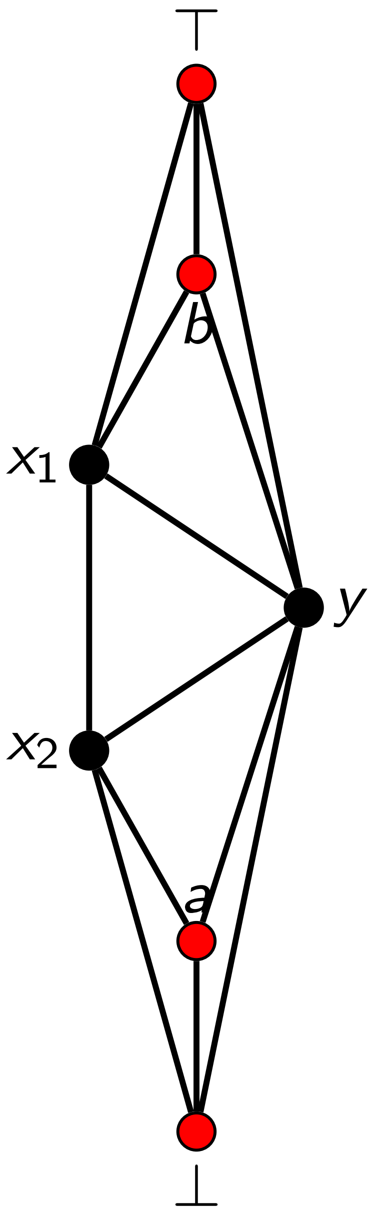
NP-hardness of Representation Extension

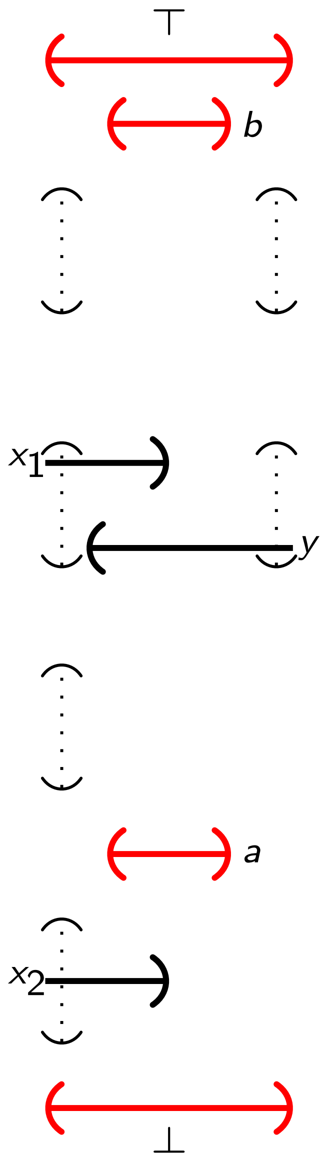
NOT gate

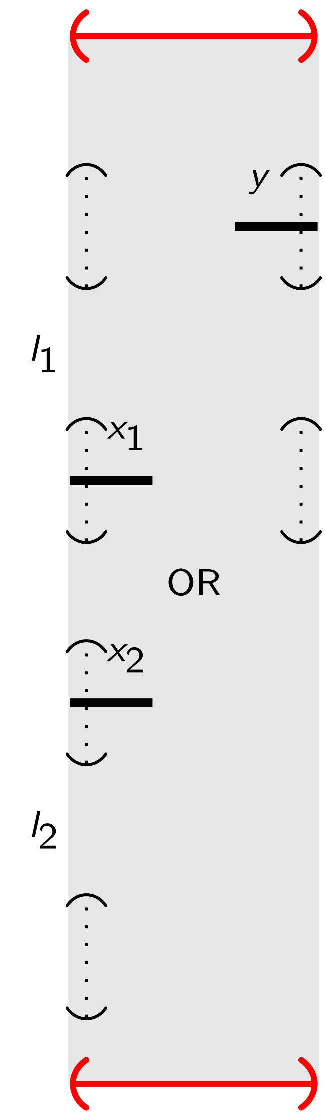


NP-hardness of Representation Extension

OR gate

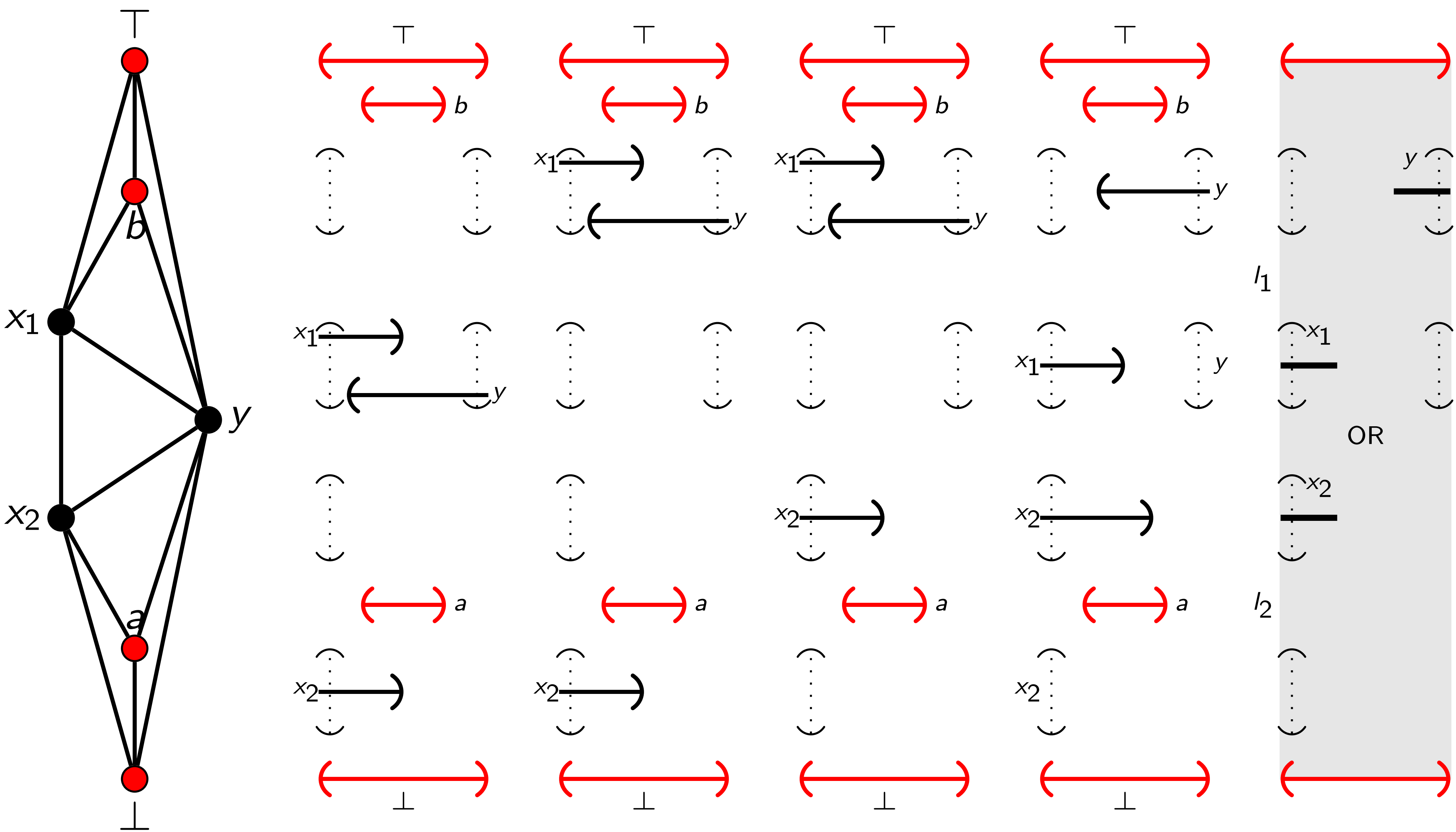




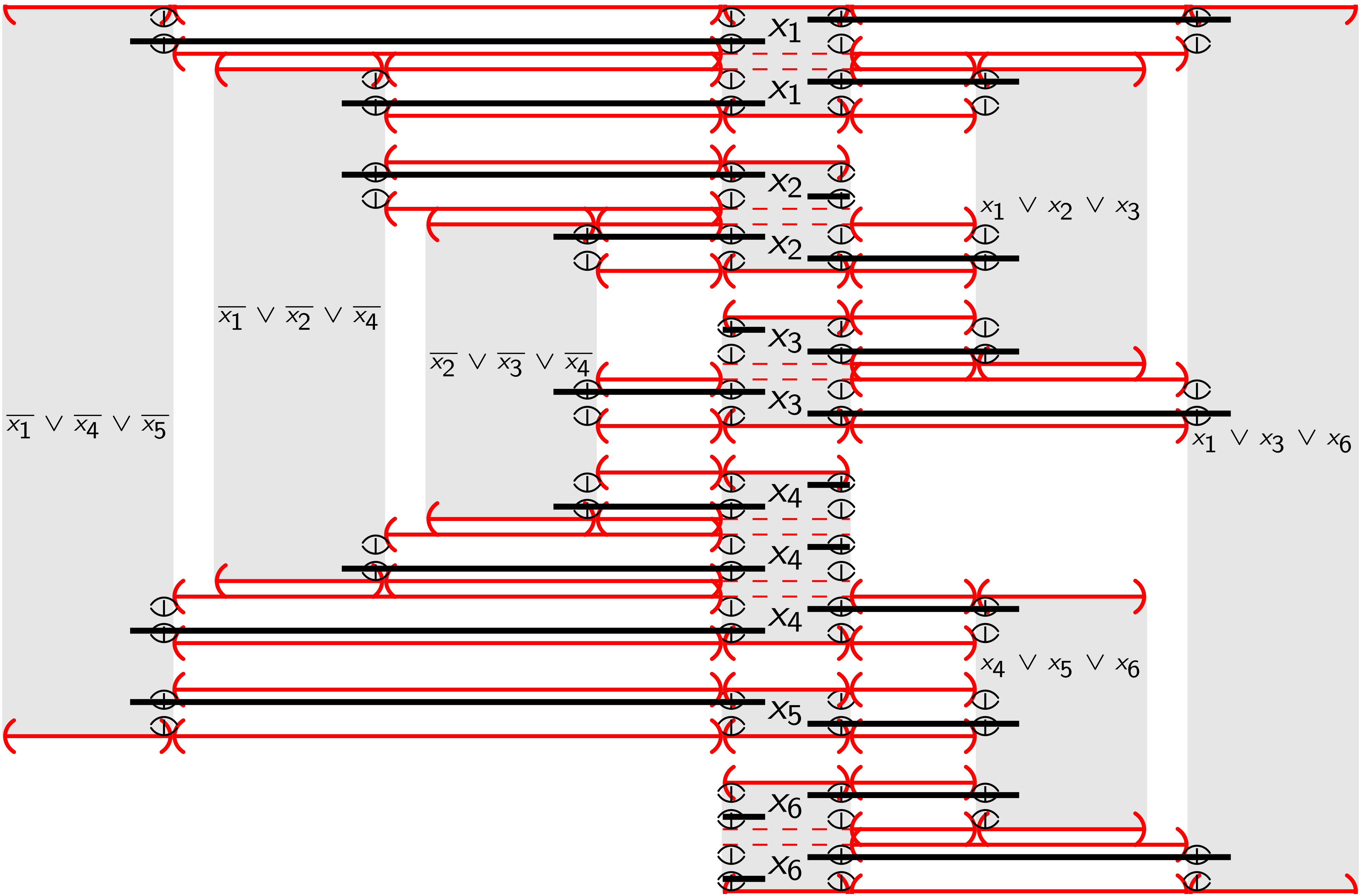


NP-hardness of Representation Extension

OR gate



NP-hardness of Representation Extension



NP-hardness on the Integer Grid (or fixed ε)

Problem: Representation extension in the Integer Grid.

NP-hardness on the Integer Grid (or fixed ε)

Problem: Representation extension in the Integer Grid.

3-Partition

Input: A set of positive integers $w, a_1, a_2, \ldots, a_{3m}$ such that for each $i = 1, \ldots, 3m$, we have $\frac{w}{4} < a_i < \frac{w}{2}$.

Question: Can $\{a_1, \ldots, a_{3m}\}$ be partitioned into m triples, such that the total sum of each triple is exactly w?

Strongly NP-complete [Garey Johnson 1979]

NP-hardness on the Integer Grid (or fixed ε)

Problem: Representation extension in the Integer Grid.

3-Partition

Input: A set of positive integers $w, a_1, a_2, \ldots, a_{3m}$ such that for each $i = 1, \ldots, 3m$, we have $\frac{w}{4} < a_i < \frac{w}{2}$.

Question: Can $\{a_1, \ldots, a_{3m}\}$ be partitioned into m triples, such that the total sum of each triple is exactly w?

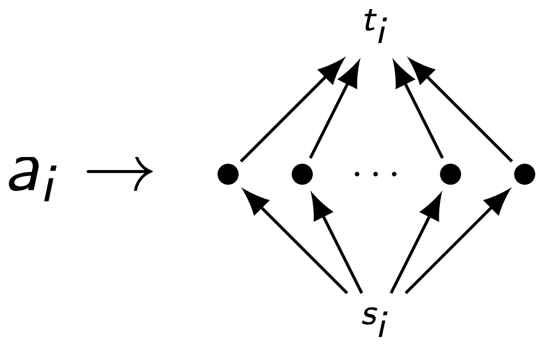
$a_i \longrightarrow$

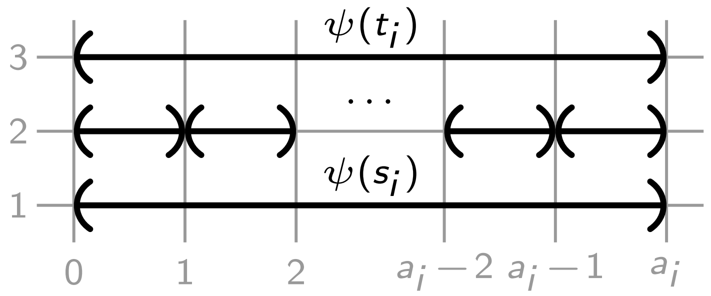
NP-hardness on the Integer Grid (or fixed ε)

Problem: Representation extension in the Integer Grid.

3-Partition

Input: A set of positive integers $w, a_1, a_2, \ldots, a_{3m}$ such that for each $i = 1, \ldots, 3m$, we have $\frac{w}{4} < a_i < \frac{w}{2}$.
Question: Can $\{a_1, \ldots, a_{3m}\}$ be partitioned into m triples, such that the total sum of each triple is exactly w?





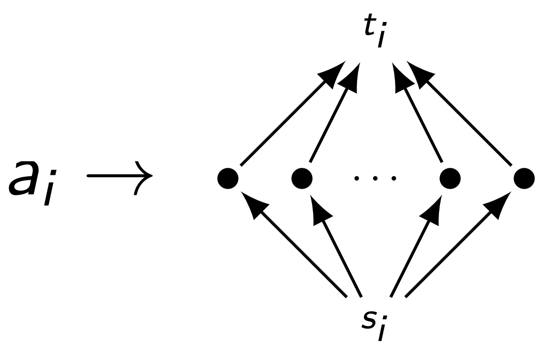
NP-hardness on the Integer Grid (or fixed ε)

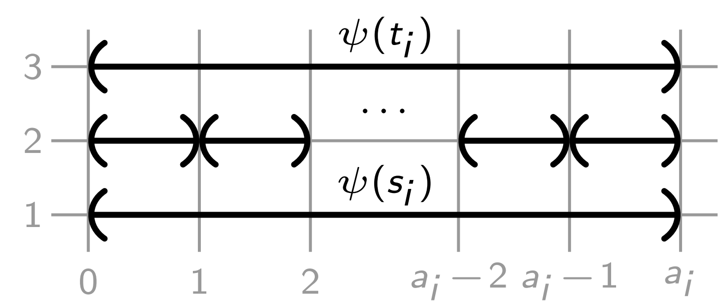
Problem: Representation extension in the Integer Grid.

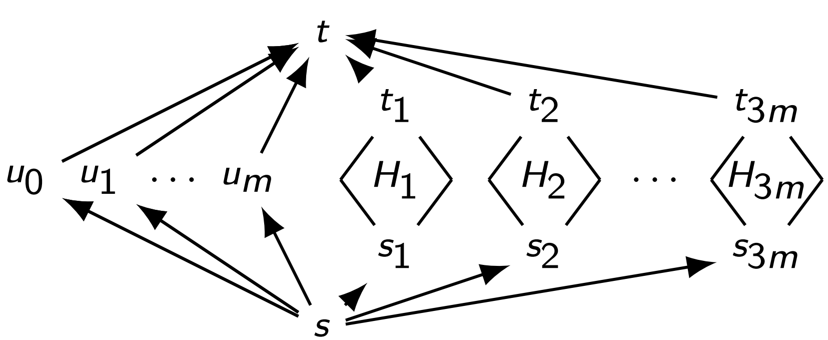
3-Partition

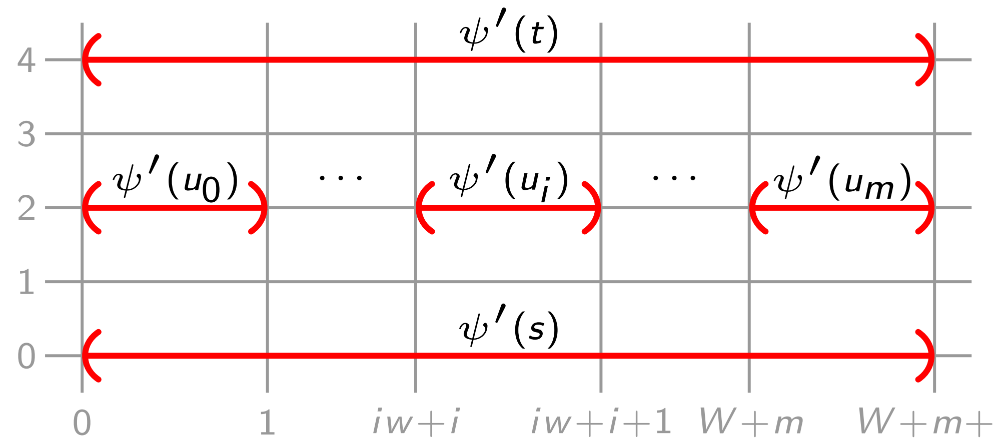
Input: A set of positive integers $w, a_1, a_2, \ldots, a_{3m}$ such that for each $i = 1, \ldots, 3m$, we have $\frac{w}{4} < a_i < \frac{w}{2}$.

Question: Can $\{a_1, \ldots, a_{3m}\}$ be partitioned into m triples, such that the total sum of each triple is exactly w?









Conclusion

Conclusion

- ε-Bar Visibility Representation Extension is NP-complete.
- *rectangular* ε-Bar Visibility Representation Extension can be solved in $O(n \log^2 n)$ time for st-graphs.
- ε-Bar Visibility Representation Extension is NP-complete for (series-parallel) st-graphs when restricted to the *Integer Grid* (or if any fixed $\varepsilon > 0$ is specified).

Conclusion

- ε-Bar Visibility Representation Extension is NP-complete.
- *rectangular* ε-Bar Visibility Representation Extension can be solved in $O(n \log^2 n)$ time for *st*-graphs.
- ε-Bar Visibility Representation Extension is NP-complete for (series-parallel) *st*-graphs when restricted to the *Integer Grid* (or if any fixed $\varepsilon > 0$ is specified).

Open Problems:

- Can ~~*rectangular*~~ ε-Bar Visibility Representation Extension can be solved in polynomial time on *st*-graphs? DAGs?
- Can **Strong** Bar Visibility Recognition / Representation Extension can be solved in polynomial time on *st*-graphs?

Conclusion

- ε-Bar Visibility Representation Extension is NP-complete.
- *rectangular* ε-Bar Visibility Representation Extension can be solved in $O(n \log^2 n)$ time for *st*-graphs.
- ε-Bar Visibility Representation Extension is NP-complete for (series-parallel) *st*-graphs when restricted to the *Integer Grid* (or if any fixed $\varepsilon > 0$ is specified).

Open Problems:

- Can ~~*rectangular*~~ ε-Bar Visibility Representation Extension can be solved in polynomial time on *st*-graphs? DAGs?
- Can **Strong** Bar Visibility Recognition / Representation Extension can be solved in polynomial time on *st*-graphs?

Thank you for your attention :-)