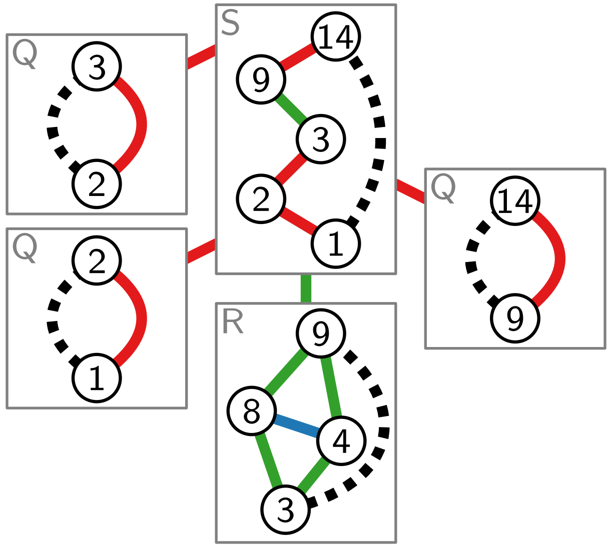
Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

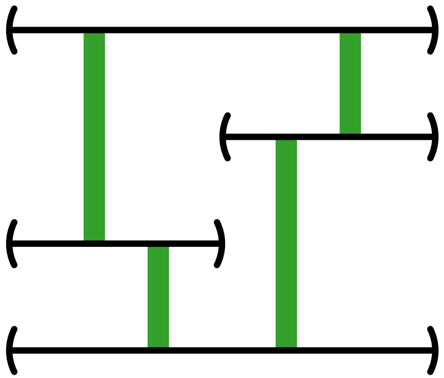
Visualization of Graphs

Lecture 9:
Partial Visibility Representation Extension



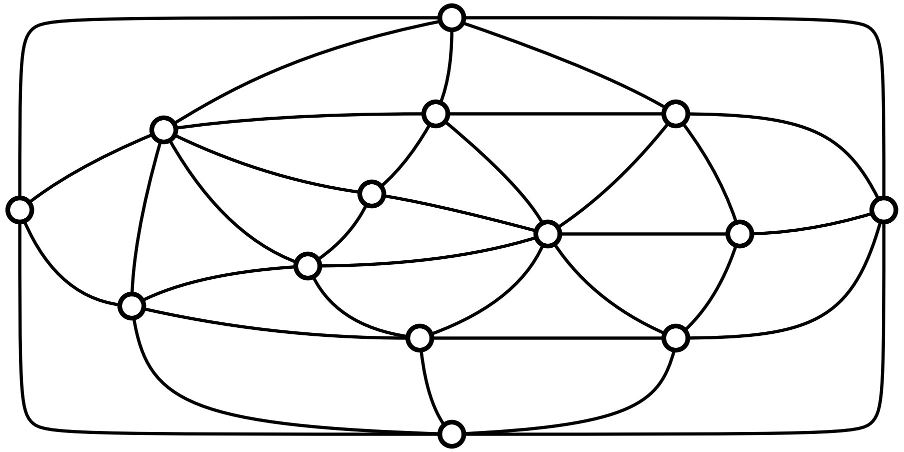
Part I:
Problem Definition

Alexander Wolff



Partial Representation Extension Problem

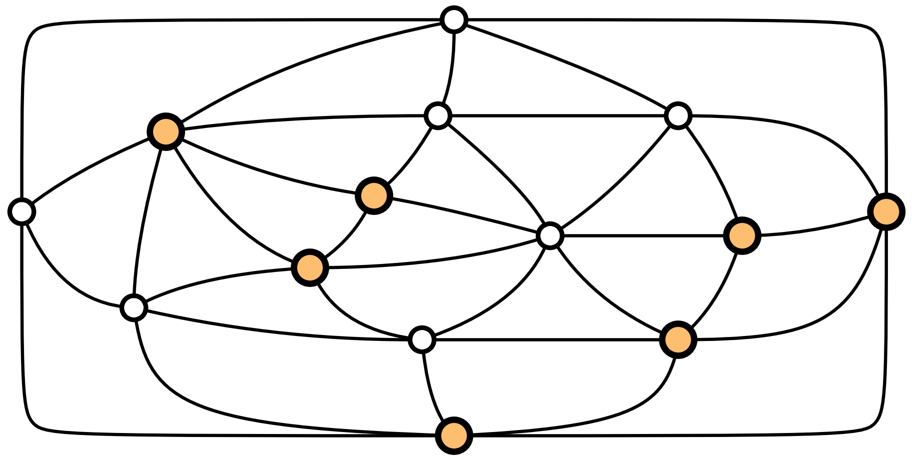
Let $G = (V, E)$ be a graph.



Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.

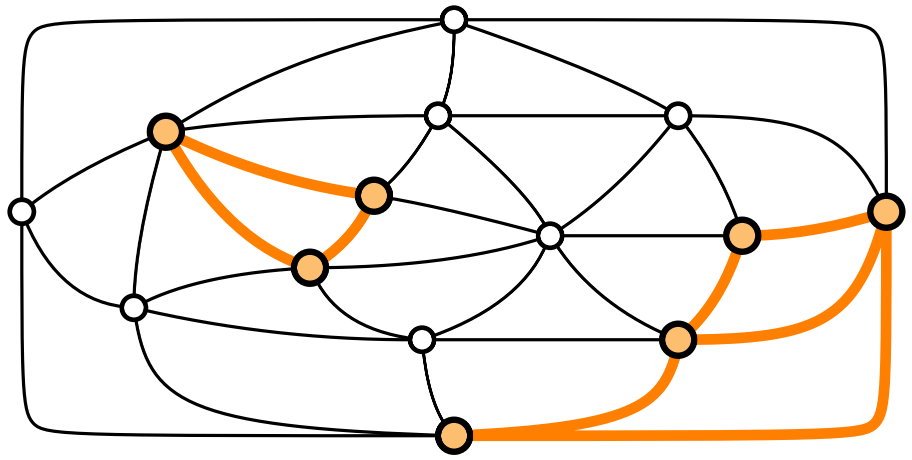
Let $V' \subseteq V$



Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.

Let $V' \subseteq V$ and $H = G[V']$

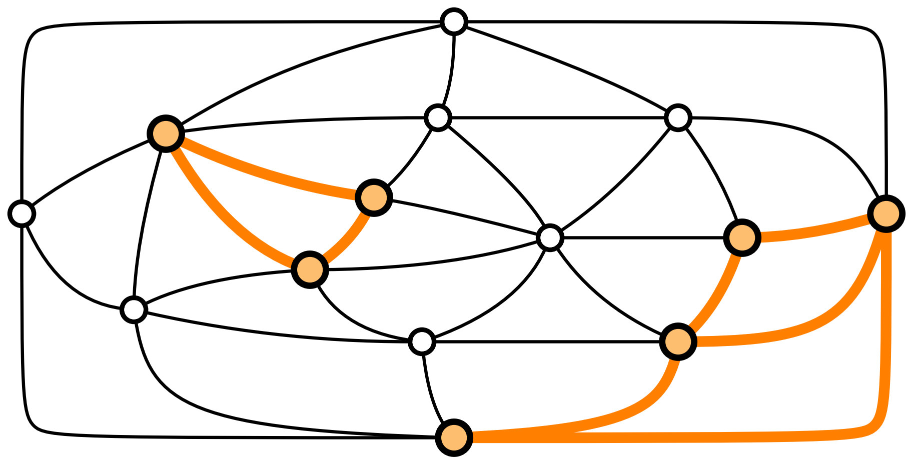


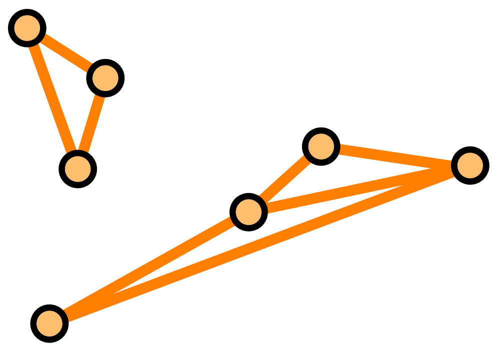
Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.

Let $V' \subseteq V$ and $H = G[V']$

Let Γ_H be a representation of H.





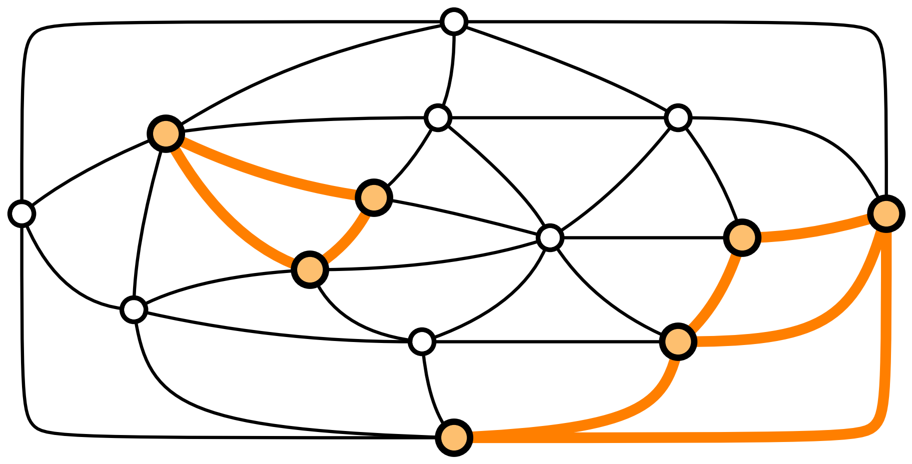
Partial Representation Extension Problem

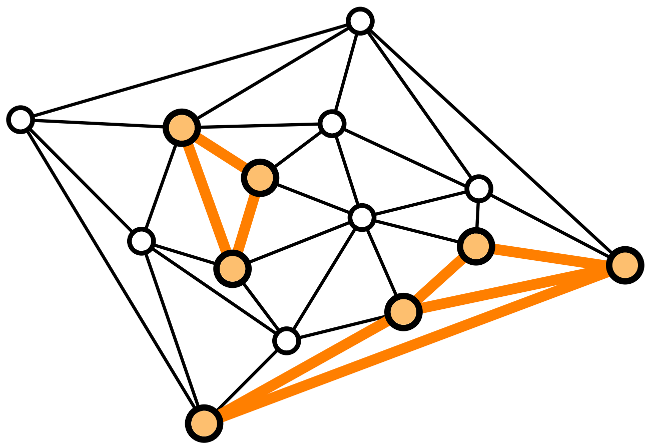
Let $G = (V, E)$ be a graph.

Let $V' \subseteq V$ and $H = G[V']$

Let Γ_H be a representation of H.

Find a representation Γ_G of G that *extends* Γ_H





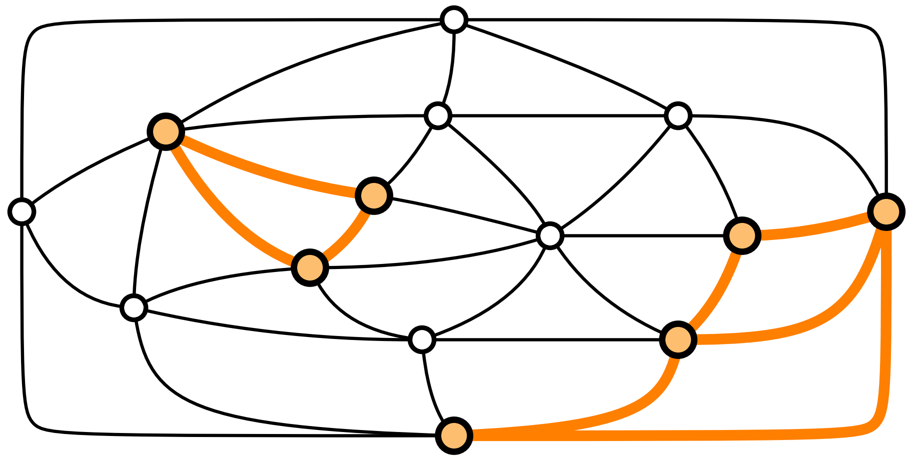
Partial Representation Extension Problem

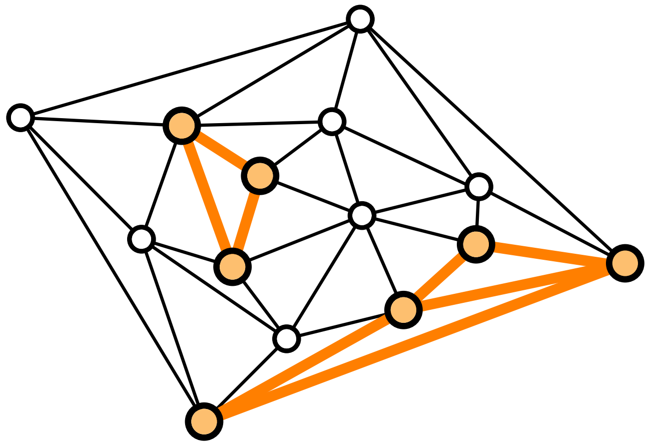
Let $G = (V, E)$ be a graph.

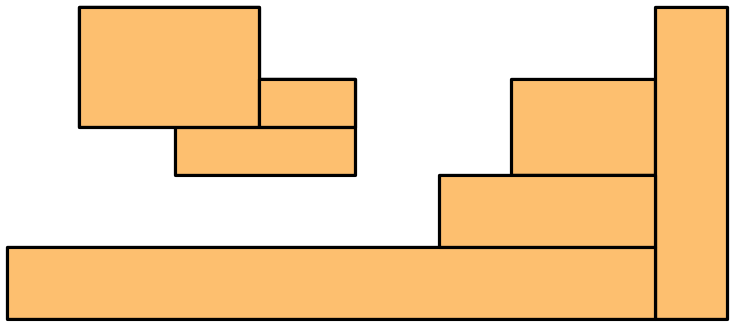
Let $V' \subseteq V$ and $H = G[V']$

Let Γ_H be a representation of H.

Find a representation Γ_G of G that *extends* Γ_H







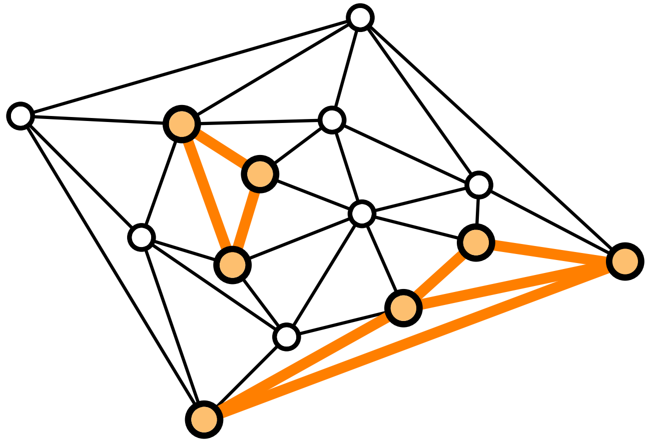
Partial Representation Extension Problem

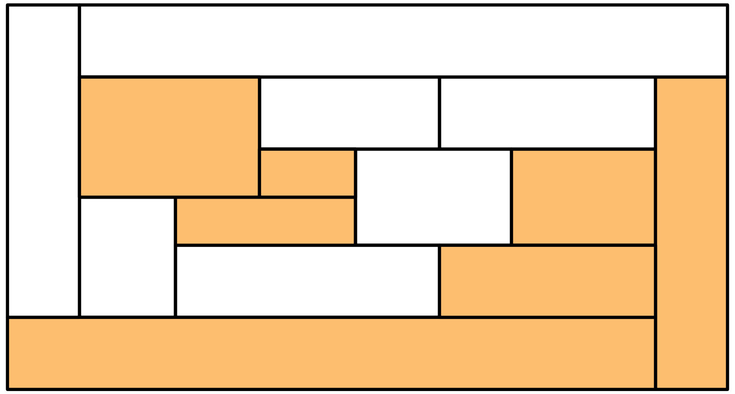
Let $G = (V, E)$ be a graph.

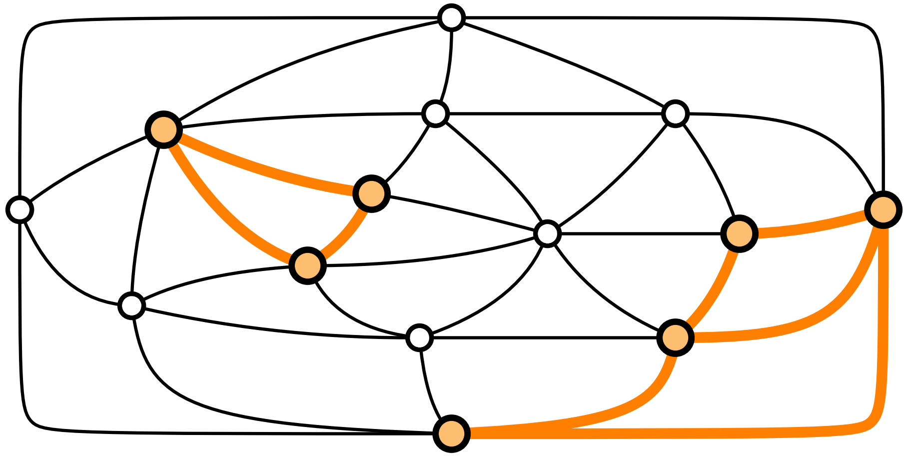
Let $V' \subseteq V$ and $H = G[V']$

Let Γ_H be a representation of H.

Find a representation Γ_G of G that *extends* Γ_H







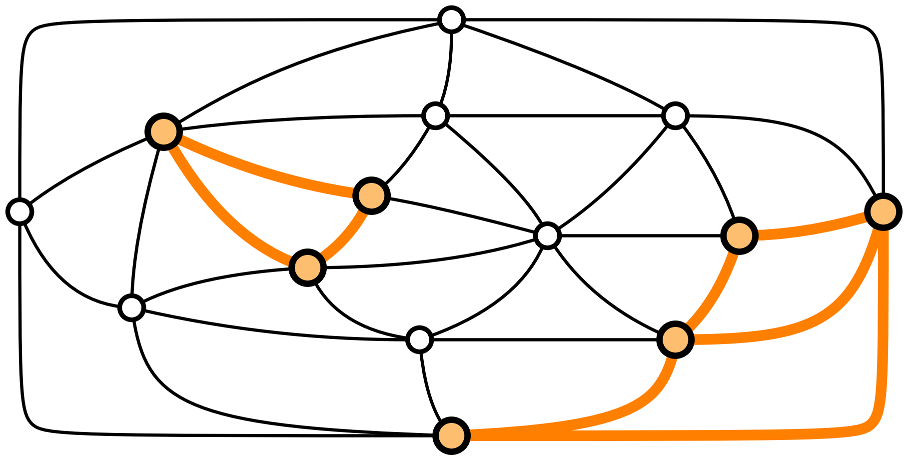
Partial Representation Extension Problem

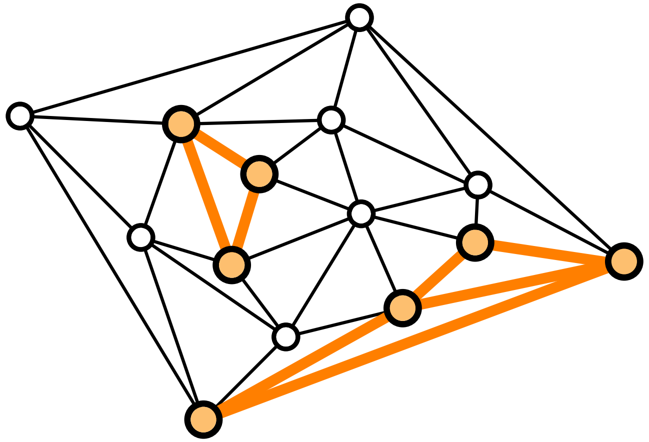
Let $G = (V, E)$ be a graph.

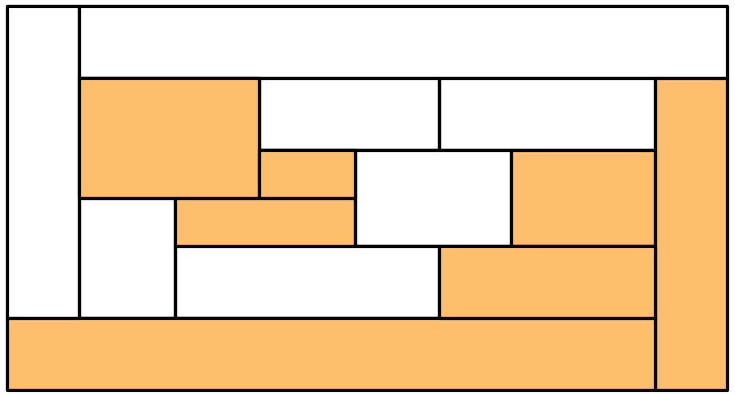
Let $V' \subseteq V$ and $H = G[V']$

Let Γ_H be a representation of H.

Find a representation Γ_G of G that *extends* Γ_H







Polytime for:

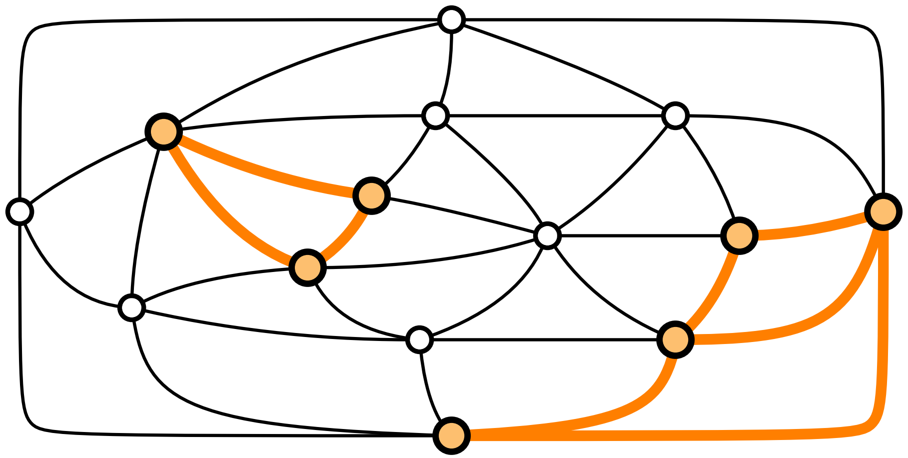
Partial Representation Extension Problem

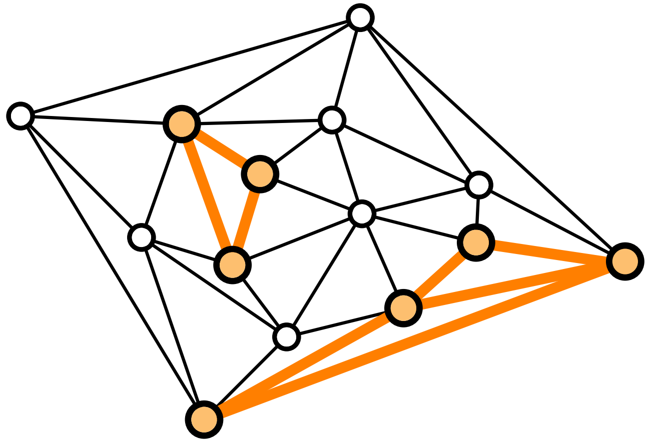
Let $G = (V, E)$ be a graph.

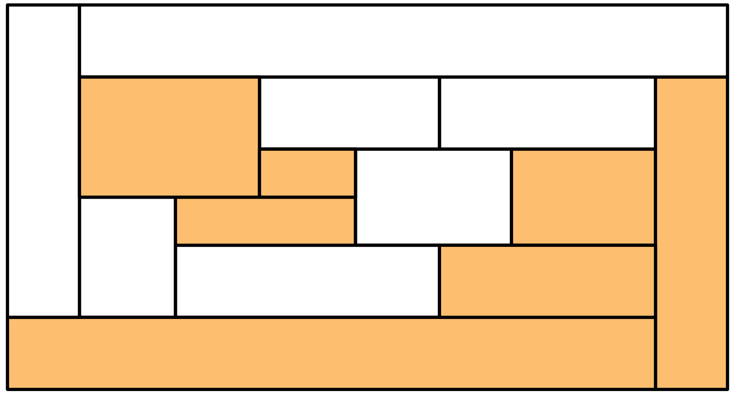
Let $V' \subseteq V$ and $H = G[V']$

Let Γ_H be a representation of H.

Find a representation Γ_G of G that *extends* Γ_H

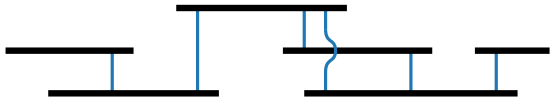




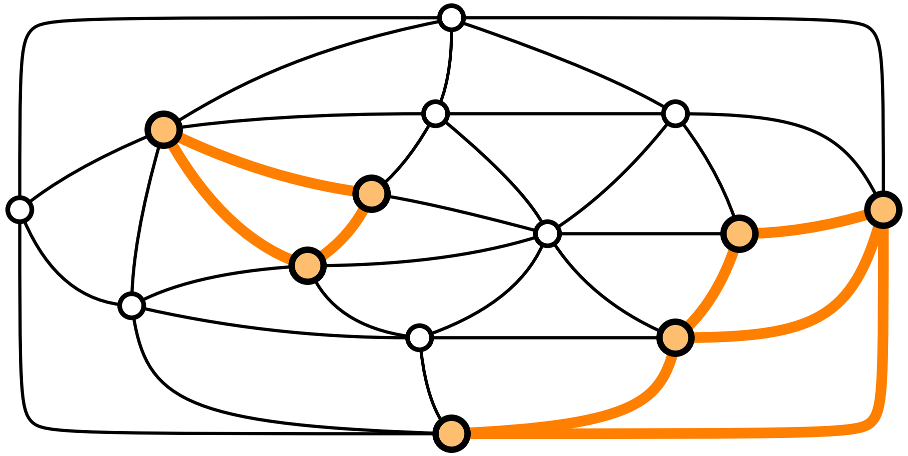


Polytime for:

- (unit) interval graphs



Partial Representation Extension Problem

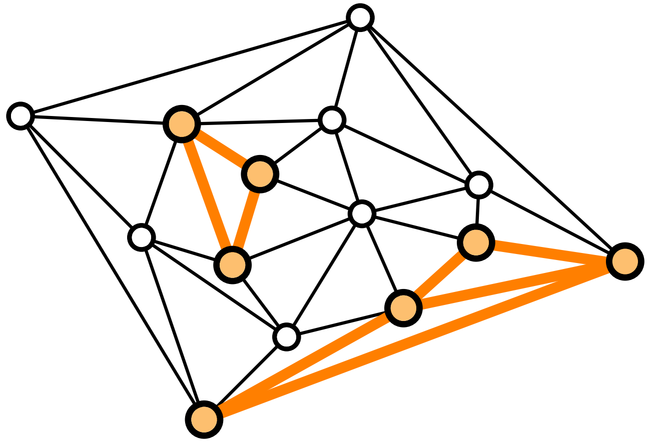


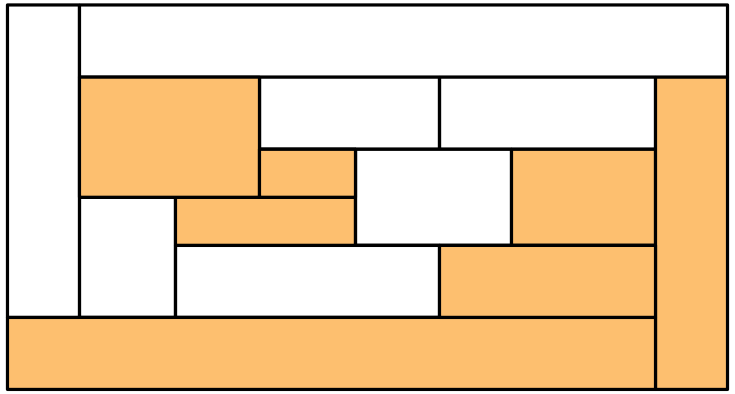
Let $G = (V, E)$ be a graph.

Let $V' \subseteq V$ and $H = G[V']$

Let Γ_H be a representation of H.

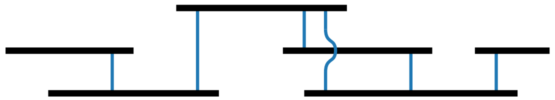
Find a representation Γ_G of G that *extends* Γ_H



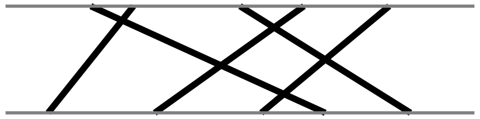


Polytime for:

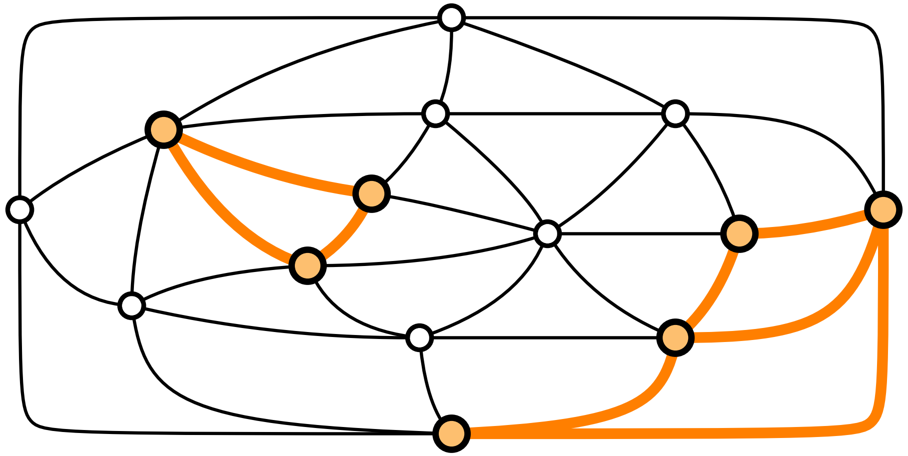
- (unit) interval graphs



- permutation graphs



Partial Representation Extension Problem

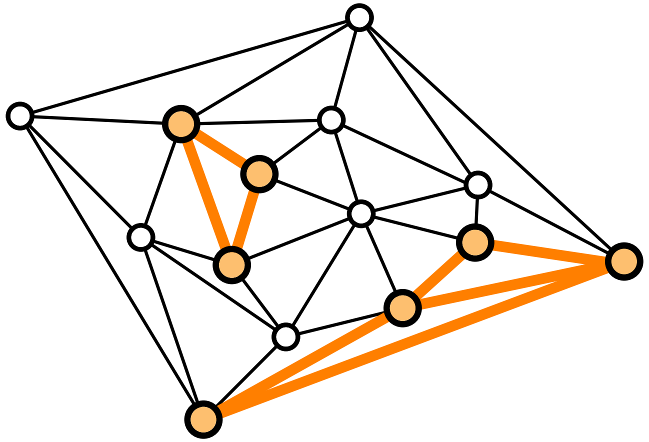


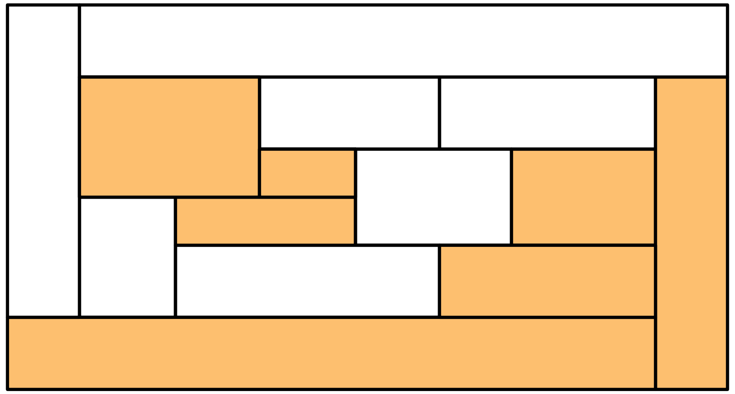
Let $G = (V, E)$ be a graph.

Let $V' \subseteq V$ and $H = G[V']$

Let Γ_H be a representation of H.

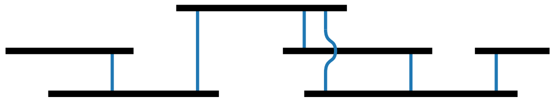
Find a representation Γ_G of G that *extends* Γ_H



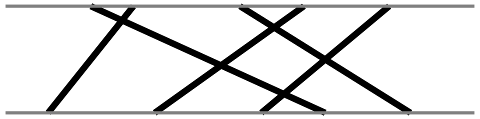


Polytime for:

- (unit) interval graphs

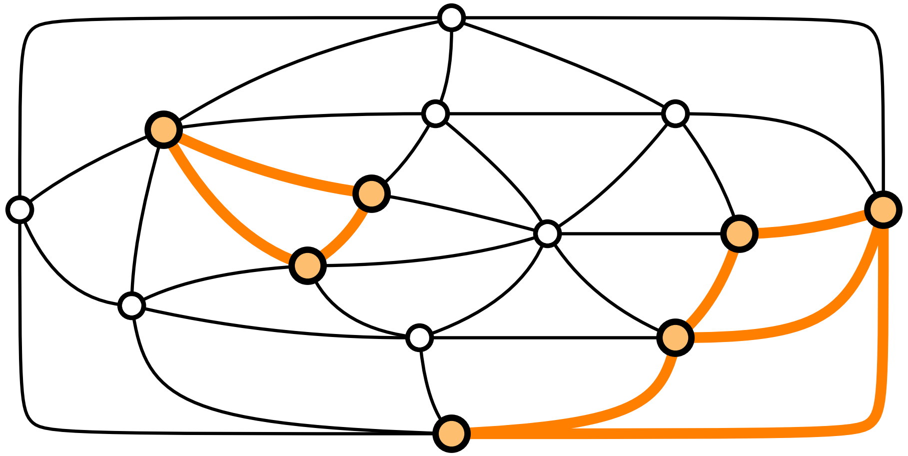


- permutation graphs



- circle graphs

Partial Representation Extension Problem

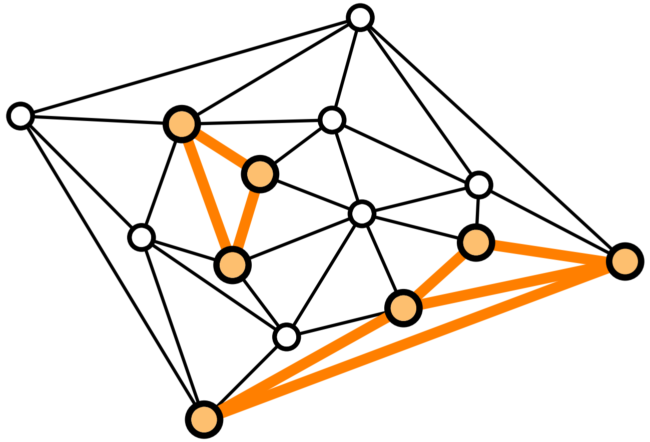


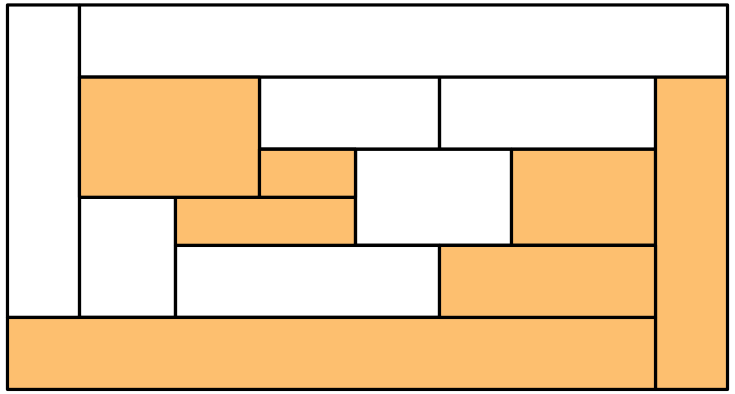
Let $G = (V, E)$ be a graph.

Let $V' \subseteq V$ and $H = G[V']$

Let Γ_H be a representation of H.

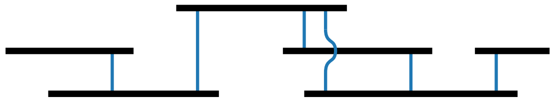
Find a representation Γ_G of G that *extends* Γ_H



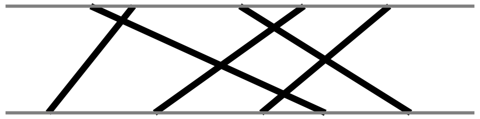


Polytime for:

- (unit) interval graphs



- permutation graphs



- circle graphs

NP-hard for:

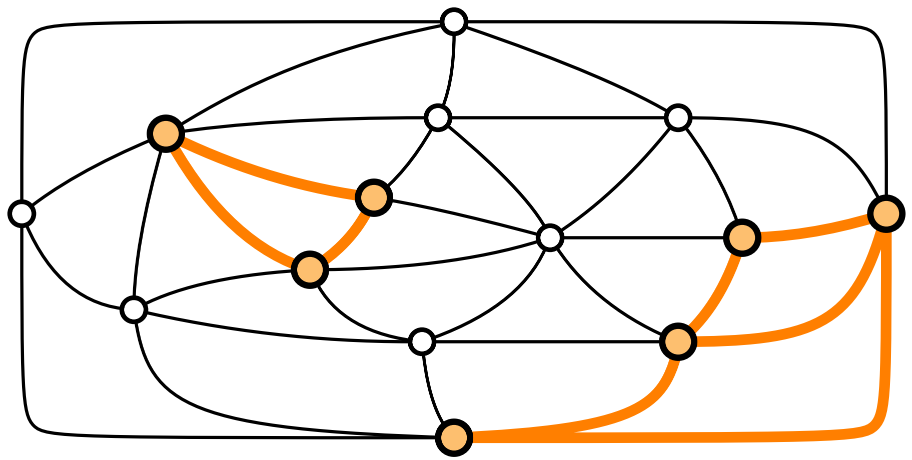
Partial Representation Extension Problem

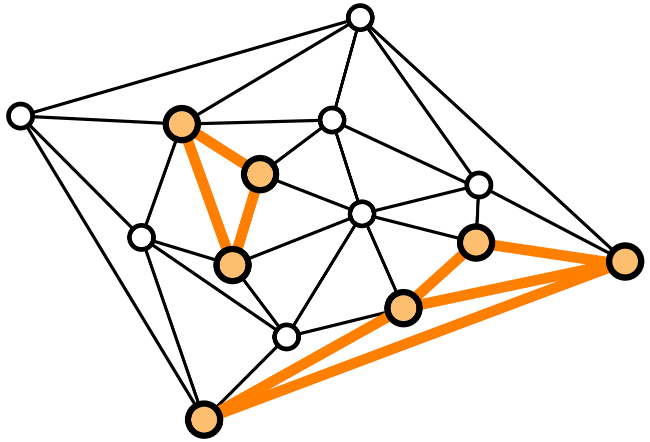
Let $G = (V, E)$ be a graph.

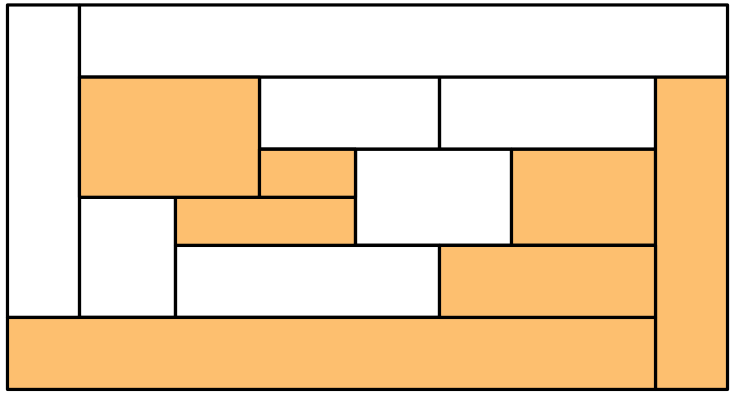
Let $V' \subseteq V$ and $H = G[V']$

Let Γ_H be a representation of H.

Find a representation Γ_G of G that *extends* Γ_H

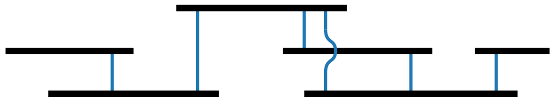




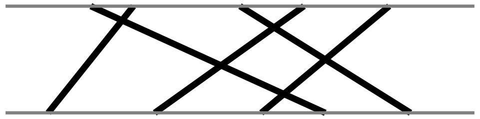


Polytime for:

- (unit) interval graphs



- permutation graphs



- circle graphs

NP-hard for:

- planar straight-line drawings

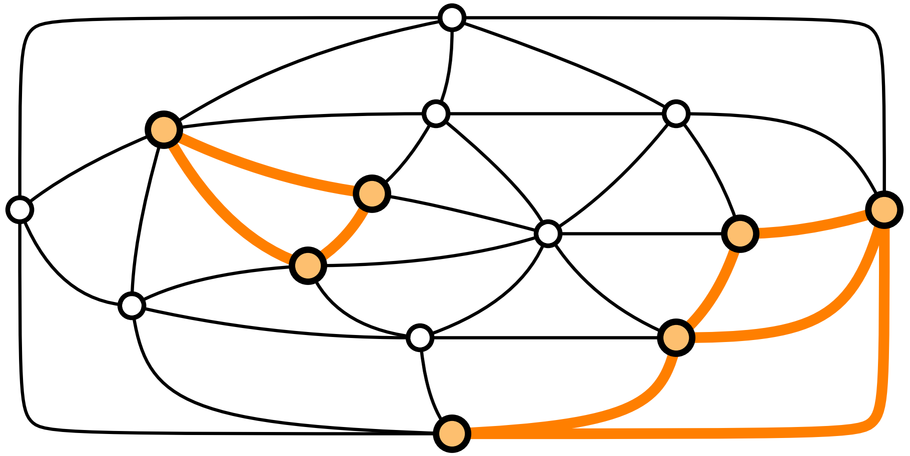
Partial Representation Extension Problem

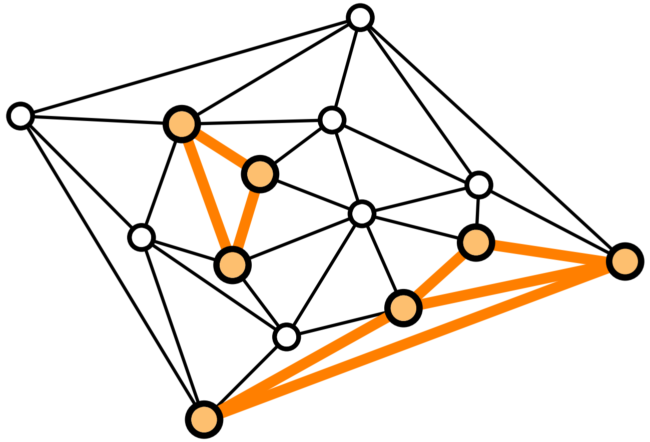
Let $G = (V, E)$ be a graph.

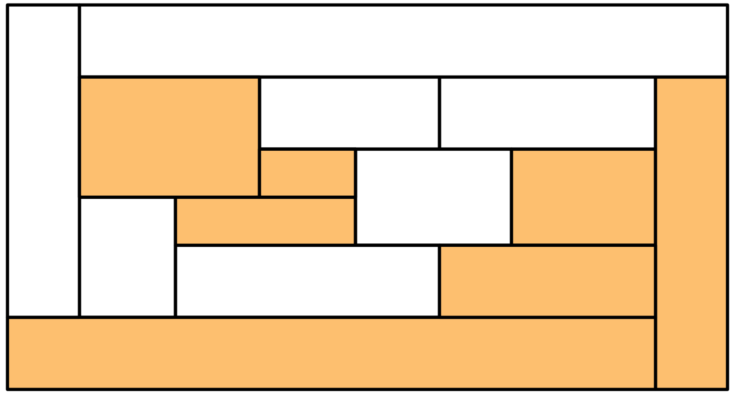
Let $V' \subseteq V$ and $H = G[V']$

Let Γ_H be a representation of H.

Find a representation Γ_G of G that *extends* Γ_H

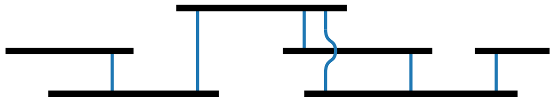




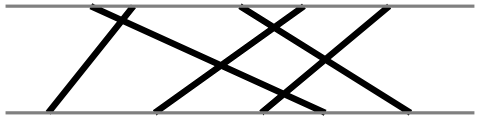


Polytime for:

- (unit) interval graphs



- permutation graphs



- circle graphs

NP-hard for:

- planar straight-line drawings
- contacts of

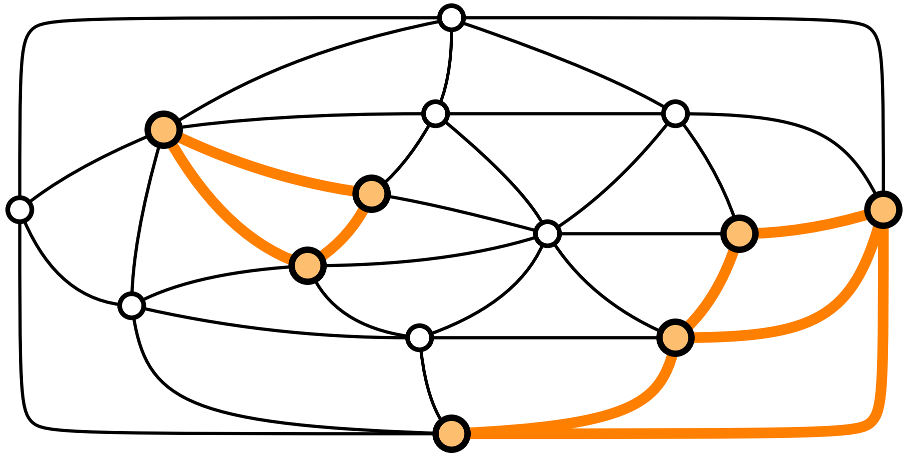
Partial Representation Extension Problem

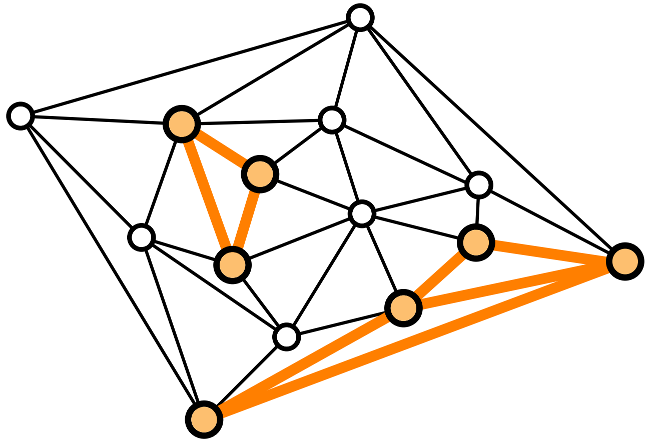
Let $G = (V, E)$ be a graph.

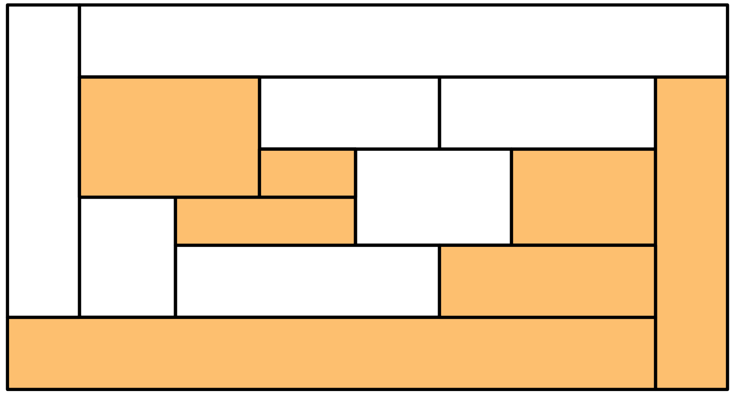
Let $V' \subseteq V$ and $H = G[V']$

Let Γ_H be a representation of H.

Find a representation Γ_G of G that *extends* Γ_H

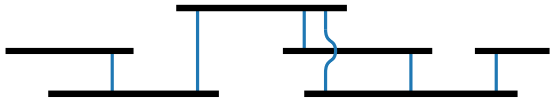




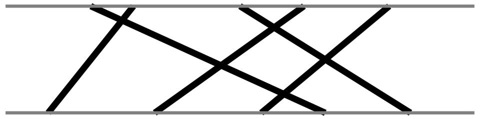


Polytime for:

- (unit) interval graphs



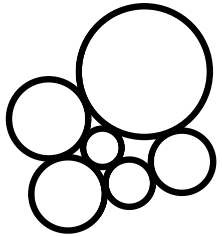
- permutation graphs



- circle graphs

NP-hard for:

- planar straight-line drawings
- contacts of
  - disks



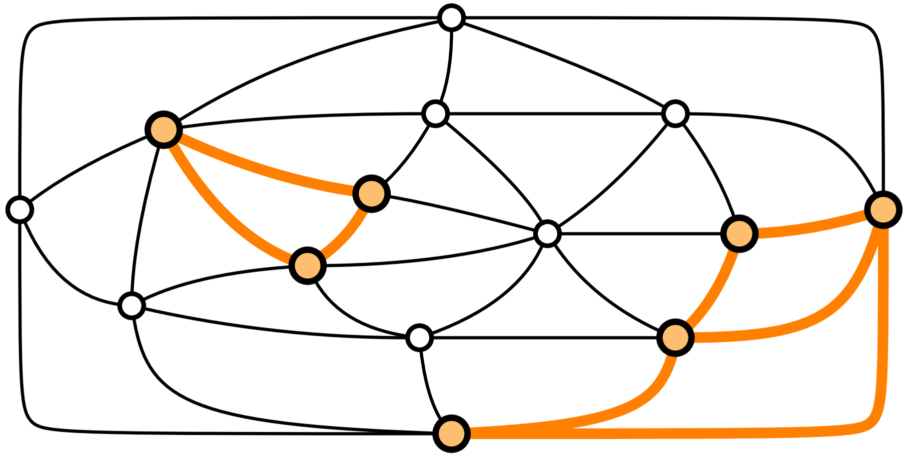
Partial Representation Extension Problem

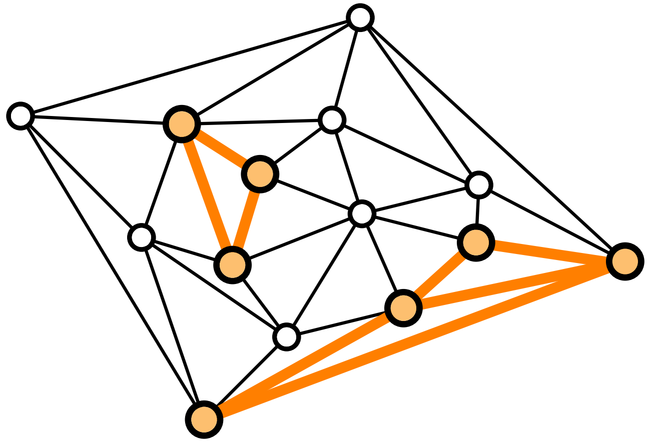
Let $G = (V, E)$ be a graph.

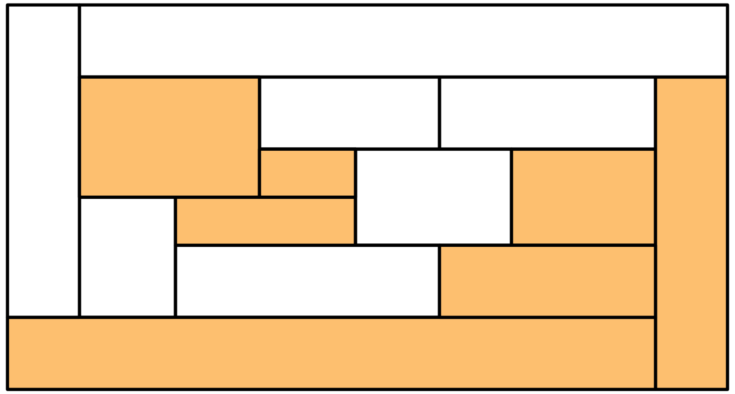
Let $V' \subseteq V$ and $H = G[V']$

Let Γ_H be a representation of H.

Find a representation Γ_G of G that *extends* Γ_H

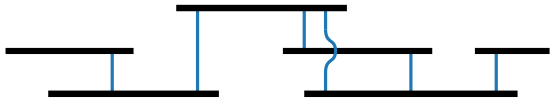




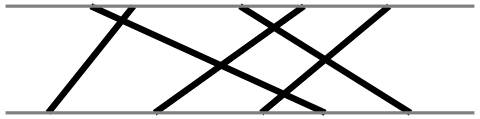


Polytime for:

- (unit) interval graphs



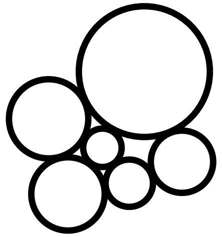
- permutation graphs

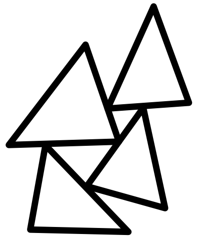


- circle graphs

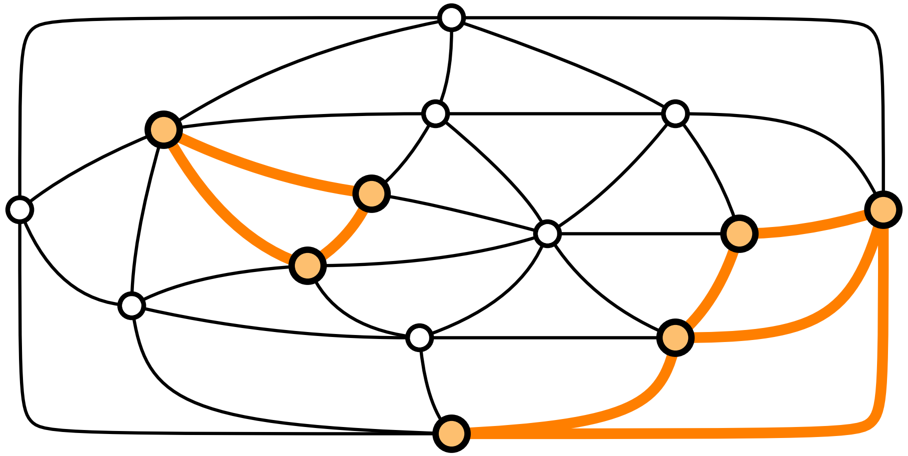
NP-hard for:

- planar straight-line drawings
- contacts of
  - disks
  - triangles





Partial Representation Extension Problem

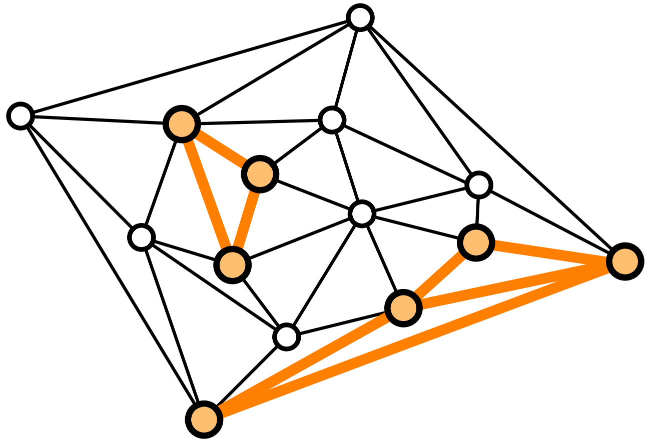


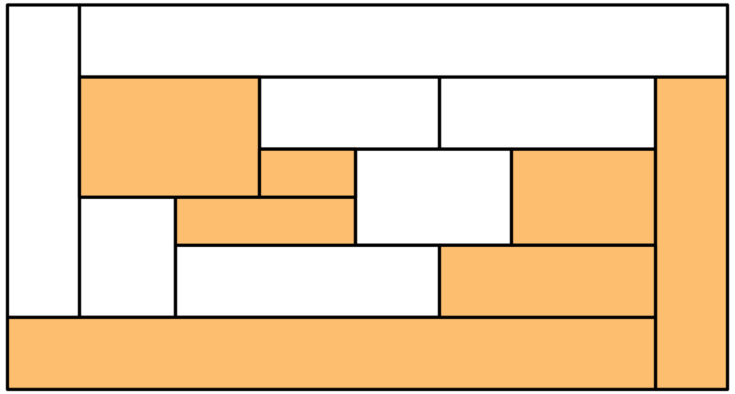
Let $G = (V, E)$ be a graph.

Let $V' \subseteq V$ and $H = G[V']$

Let Γ_H be a representation of H.

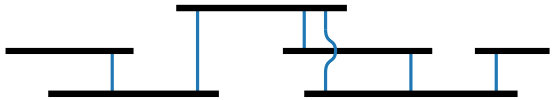
Find a representation Γ_G of G that *extends* Γ_H



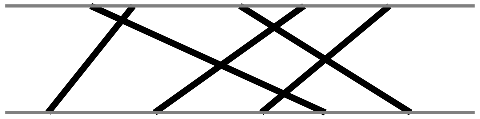


Polytime for:

- (unit) interval graphs



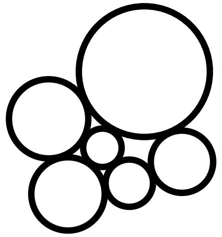
- permutation graphs

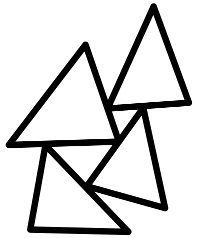


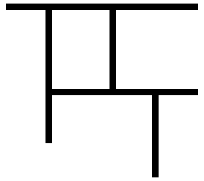
- circle graphs

NP-hard for:

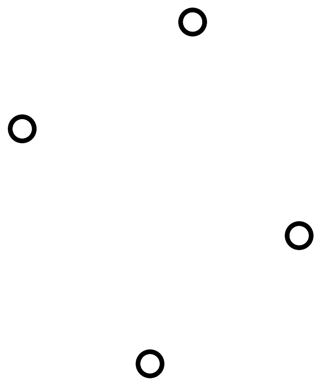
- planar straight-line drawings
- contacts of
  - disks
  - triangles
  - orthogonal segments





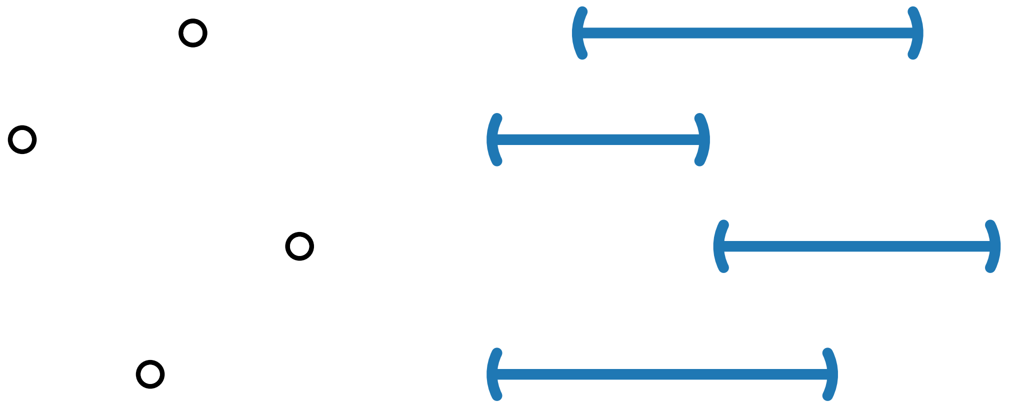


Bar Visibility Representation



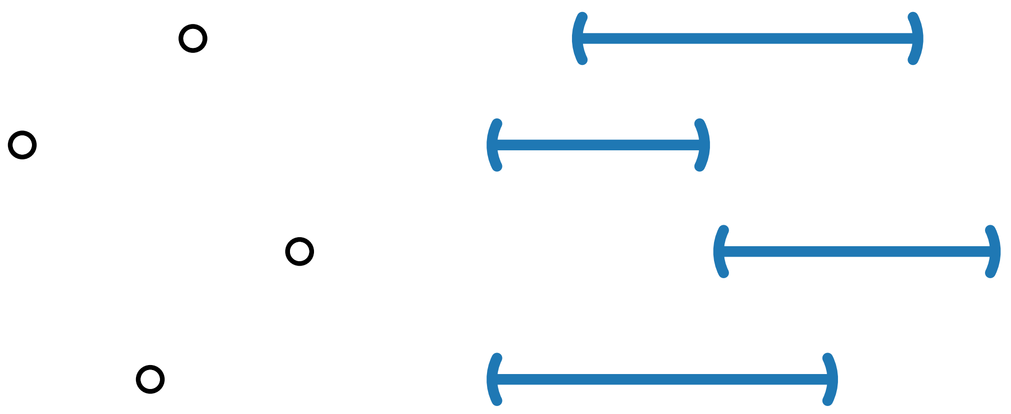
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.



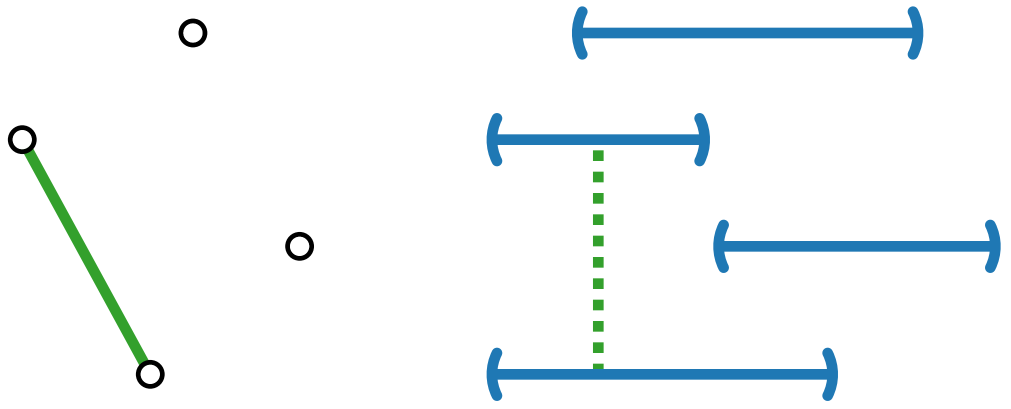
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.



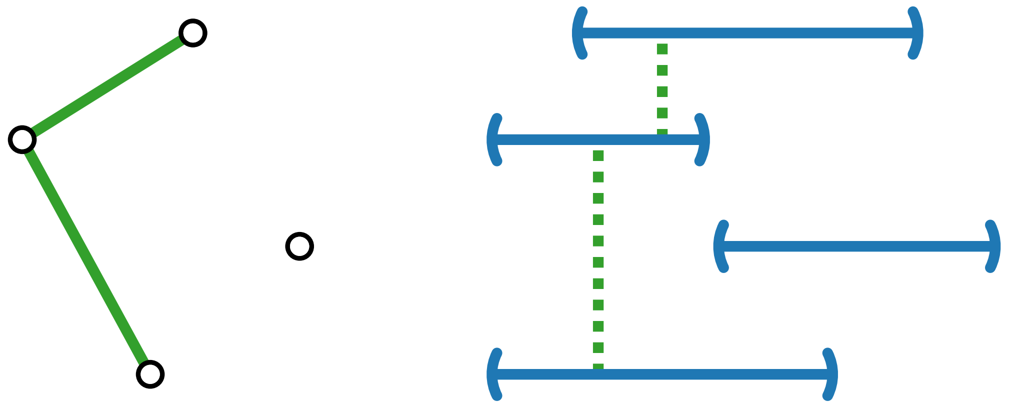
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.



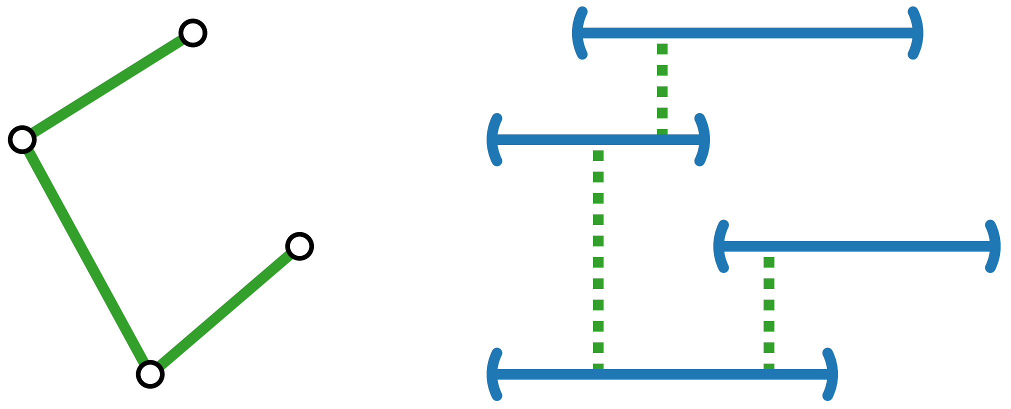
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.



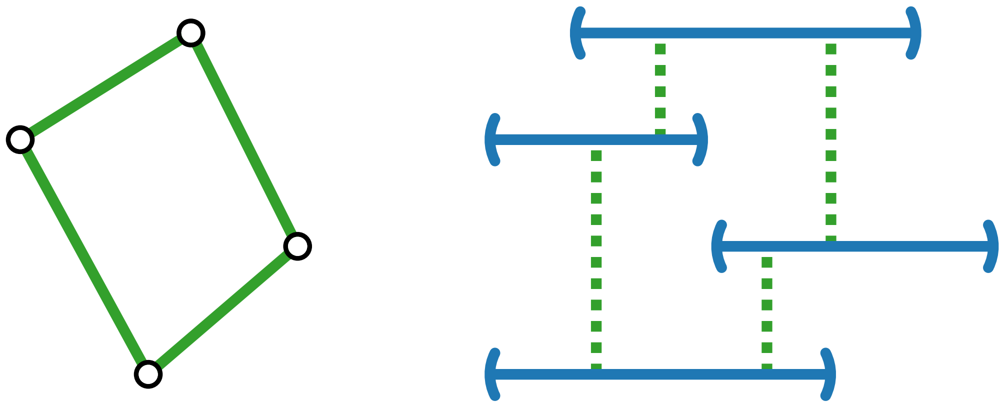
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.



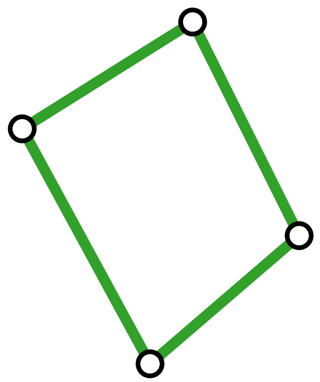
Bar Visibility Representation

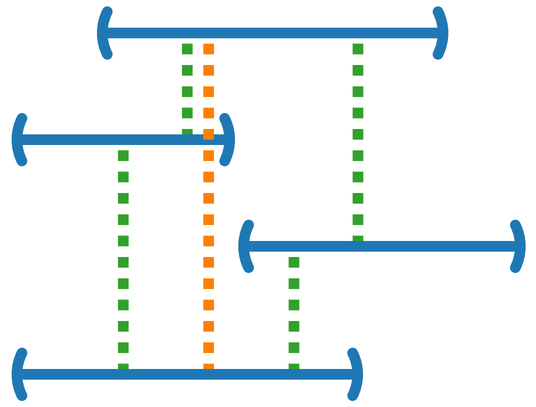
- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.



Bar Visibility Representation

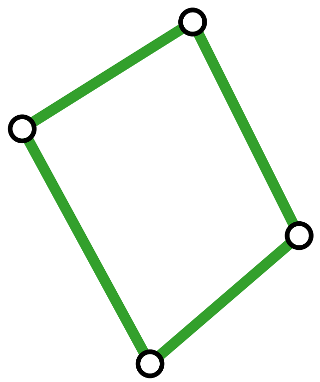
- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.

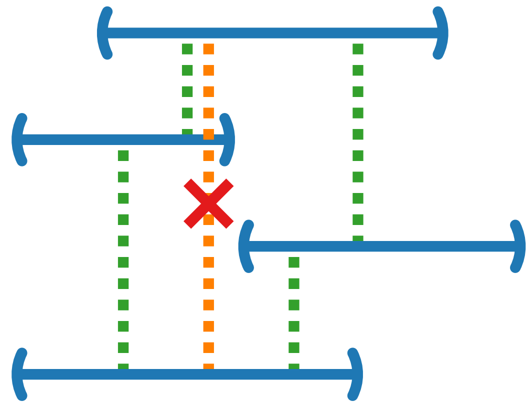




Bar Visibility Representation

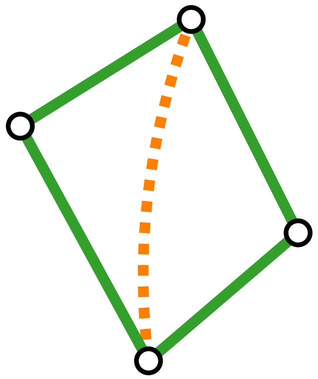
- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.

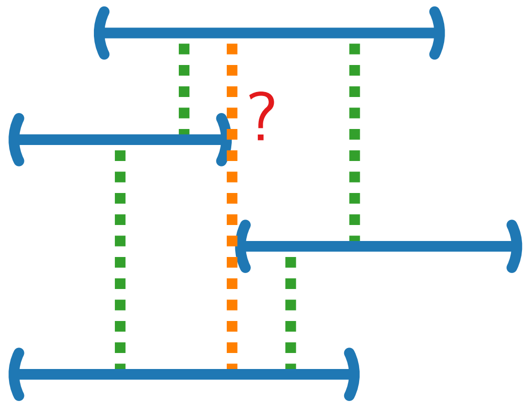




Bar Visibility Representation

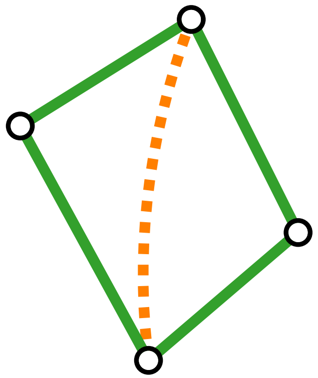
- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.

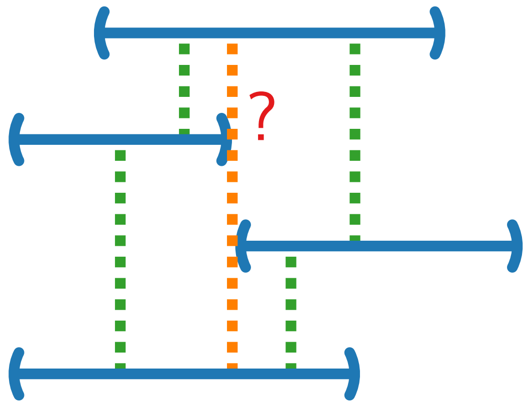




Bar Visibility Representation

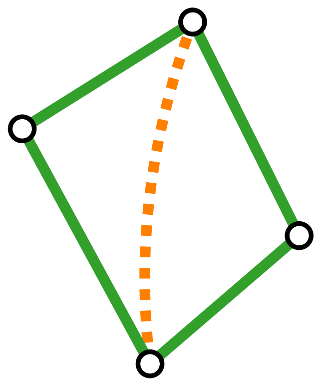
- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
- What about unobstructed **0-width** vertical lines of sight? Do all visibilities induce edges?

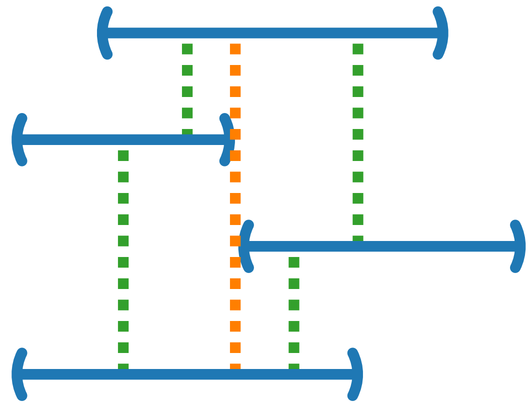




Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
- What about unobstructed **0-width** vertical lines of sight? Do all visibilities induce edges?

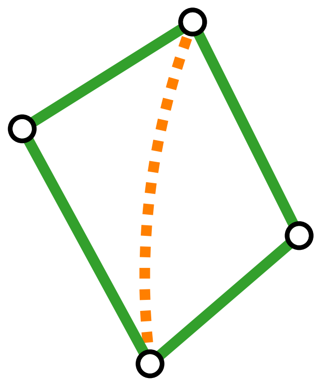


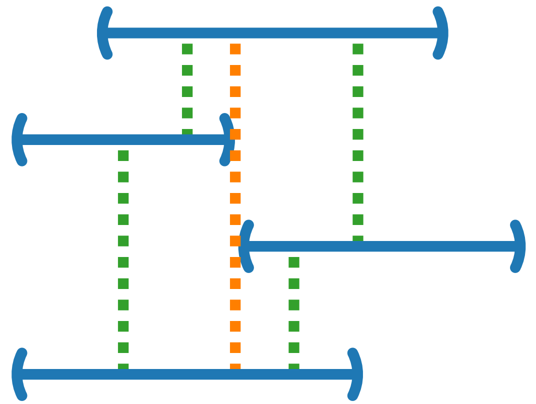


Models.

Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
- What about unobstructed **0-width** vertical lines of sight? Do all visibilities induce edges?



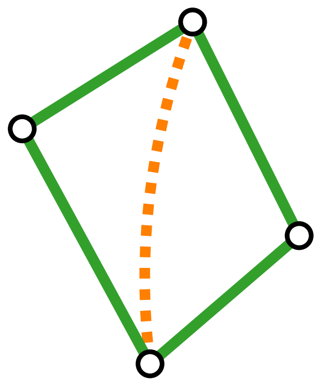


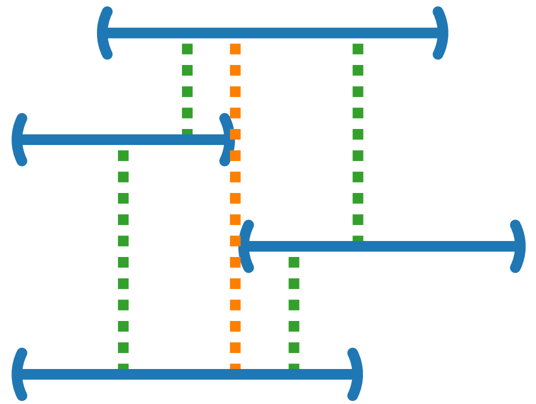
Models.

- **Strong:**
  Edge $uv \Leftrightarrow$ unobstructed **0-width** vertical lines of sight.

Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
- What about unobstructed **0-width** vertical lines of sight? Do all visibilities induce edges?



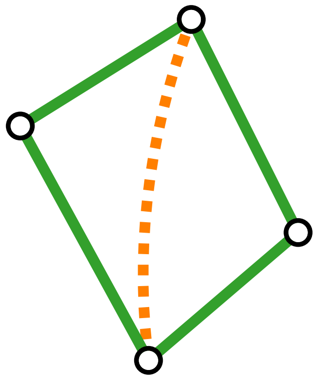


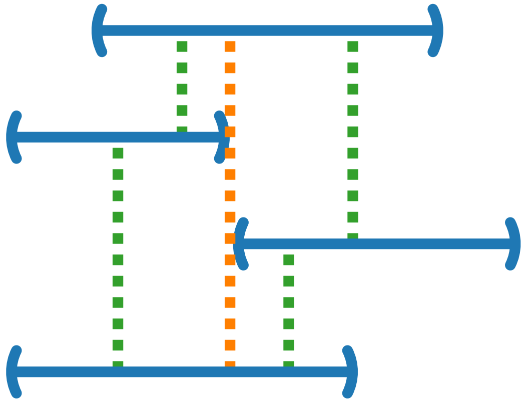
Models.

- **Strong:**
  Edge $uv \Leftrightarrow$ unobstructed **0-width** vertical lines of sight.
- **Epsilon:**
  Edge $uv \Leftrightarrow$ ε-wide vertical lines of sight for $\varepsilon > 0$.

Bar Visibility Representation

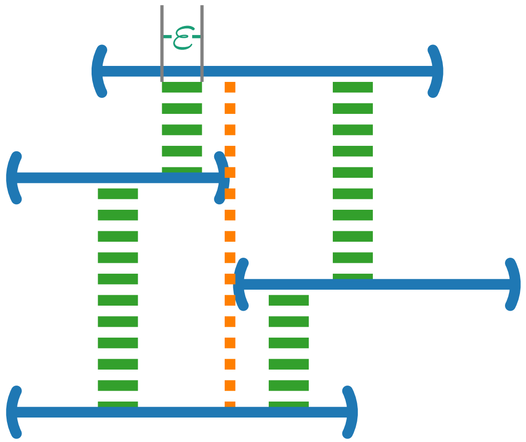
- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
- What about unobstructed **0-width** vertical lines of sight? Do all visibilities induce edges?





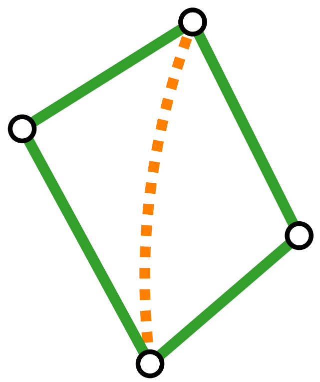
Models.

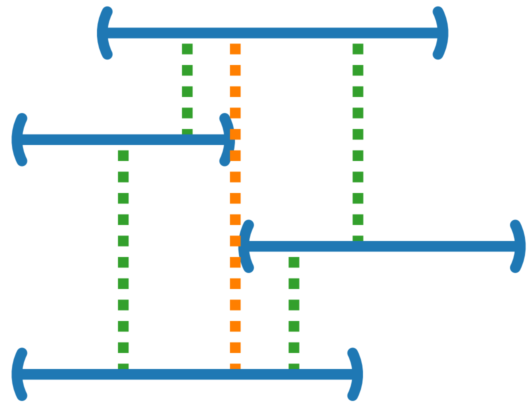
- **Strong:**
  Edge $uv \Leftrightarrow$ unobstructed **0-width** vertical lines of sight.
- **Epsilon:**
  Edge $uv \Leftrightarrow$ ε-wide vertical lines of sight for $\varepsilon > 0$.



Bar Visibility Representation

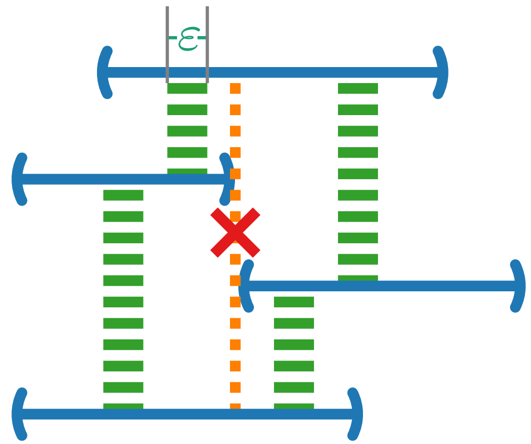
- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
- What about unobstructed **0-width** vertical lines of sight? Do all visibilities induce edges?





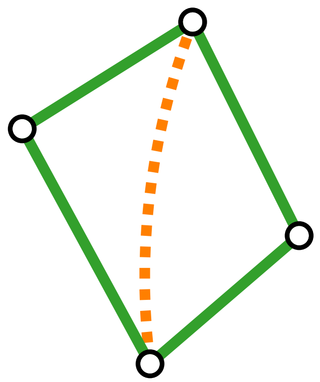
Models.

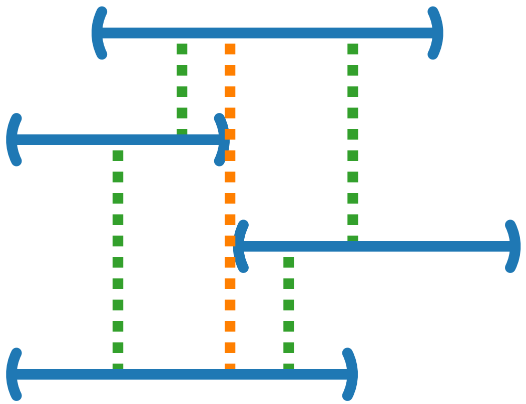
- **Strong:**
  Edge $uv \Leftrightarrow$ unobstructed **0-width** vertical lines of sight.
- **Epsilon:**
  Edge $uv \Leftrightarrow$ ε-wide vertical lines of sight for $\varepsilon > 0$.



Bar Visibility Representation

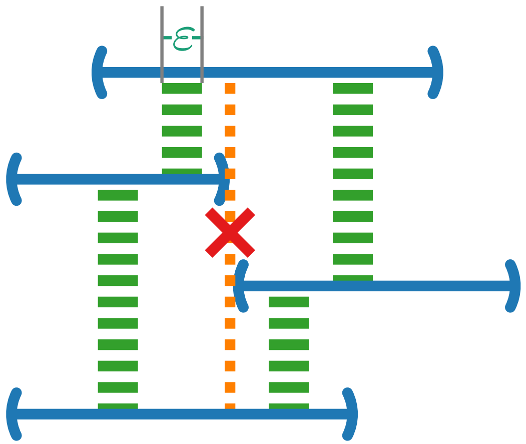
- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
- What about unobstructed **0-width** vertical lines of sight? Do all visibilities induce edges?





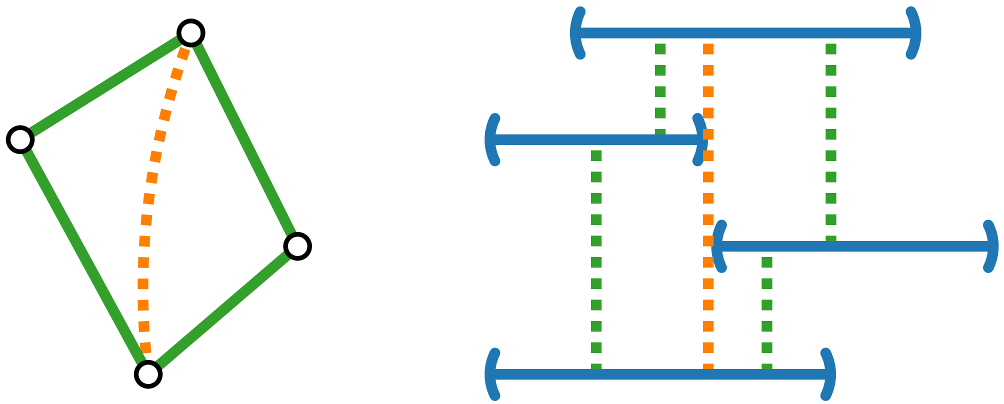
Models.

- **Strong:**
  Edge $uv \Leftrightarrow$ unobstructed **0-width** vertical lines of sight.
- **Epsilon:**
  Edge $uv \Leftrightarrow$ ε-wide vertical lines of sight for $\varepsilon > 0$.
- **Weak:**
  Edge $uv \Rightarrow$ unobstructed vertical sightlines exists, i.e., any subset of *visible* pairs



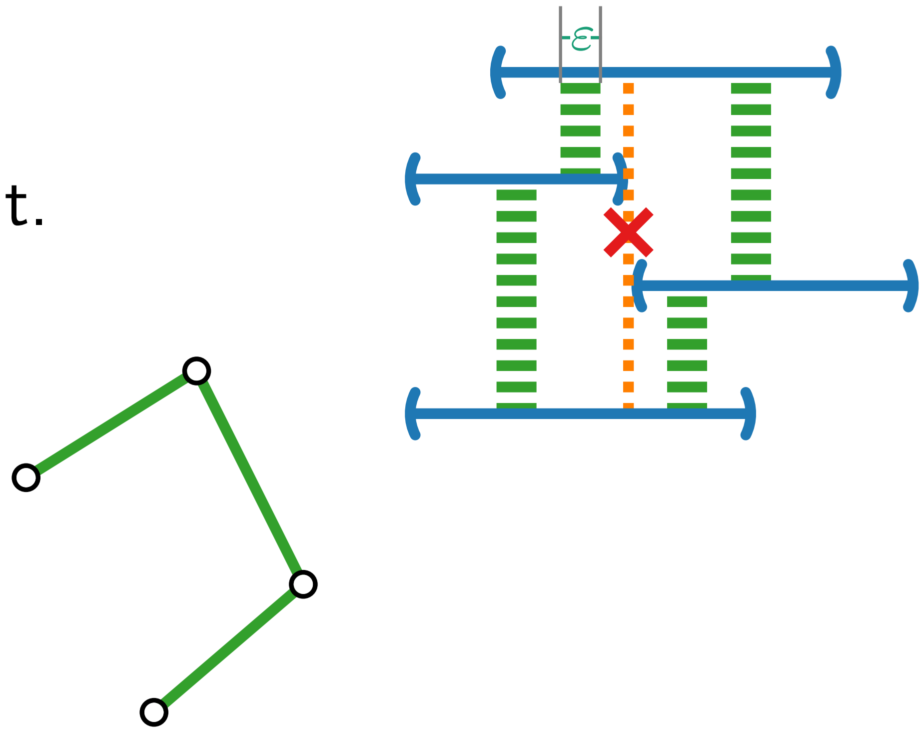
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
- What about unobstructed **0-width** vertical lines of sight? Do all visibilities induce edges?

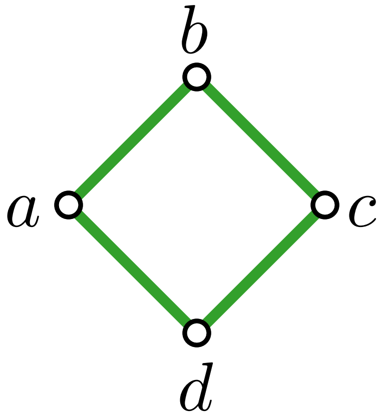


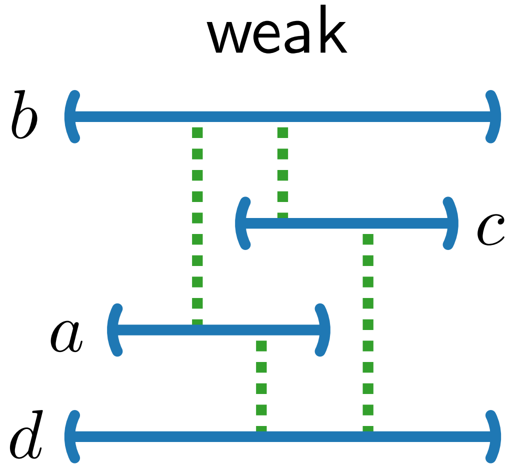
Models.

- **Strong:**
  Edge $uv \Leftrightarrow$ unobstructed **0-width** vertical lines of sight.
- **Epsilon:**
  Edge $uv \Leftrightarrow$ ε-wide vertical lines of sight for $\varepsilon > 0$.
- **Weak:**
  Edge $uv \Rightarrow$ unobstructed vertical sightlines exists, i.e., any subset of *visible* pairs

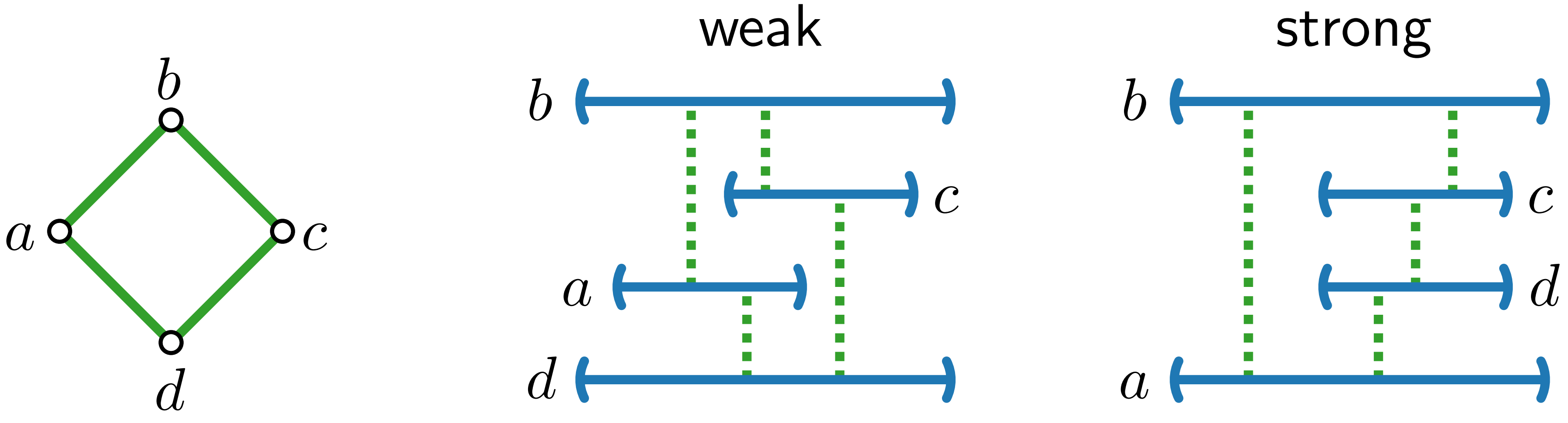


Problems

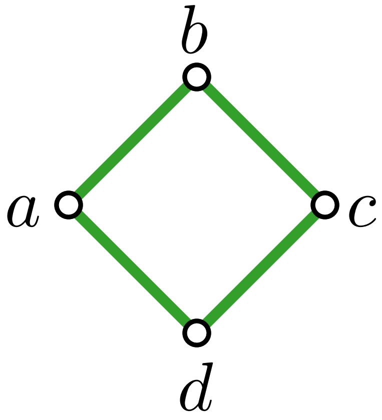


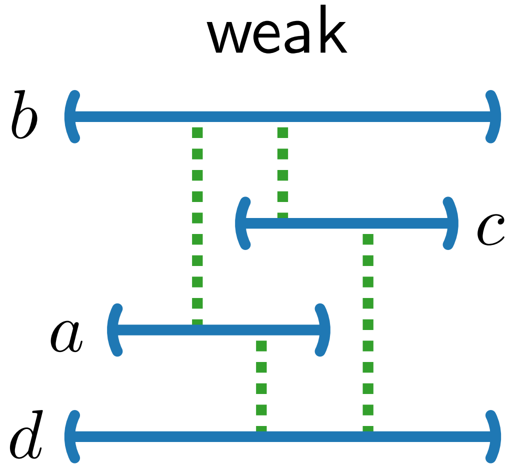


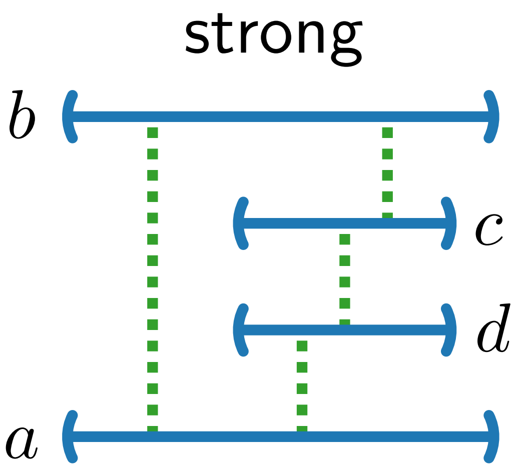
Problems

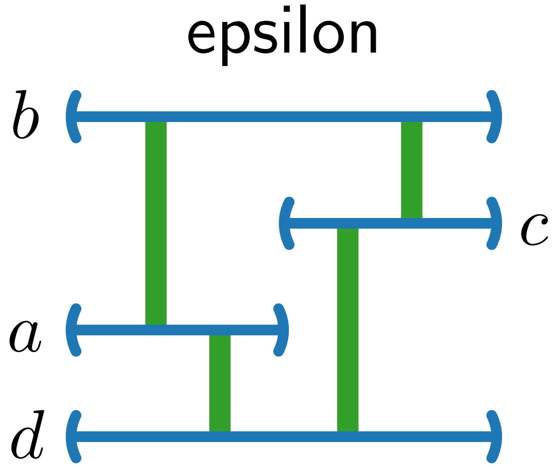


Problems

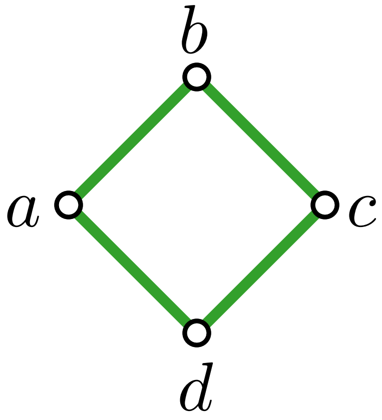


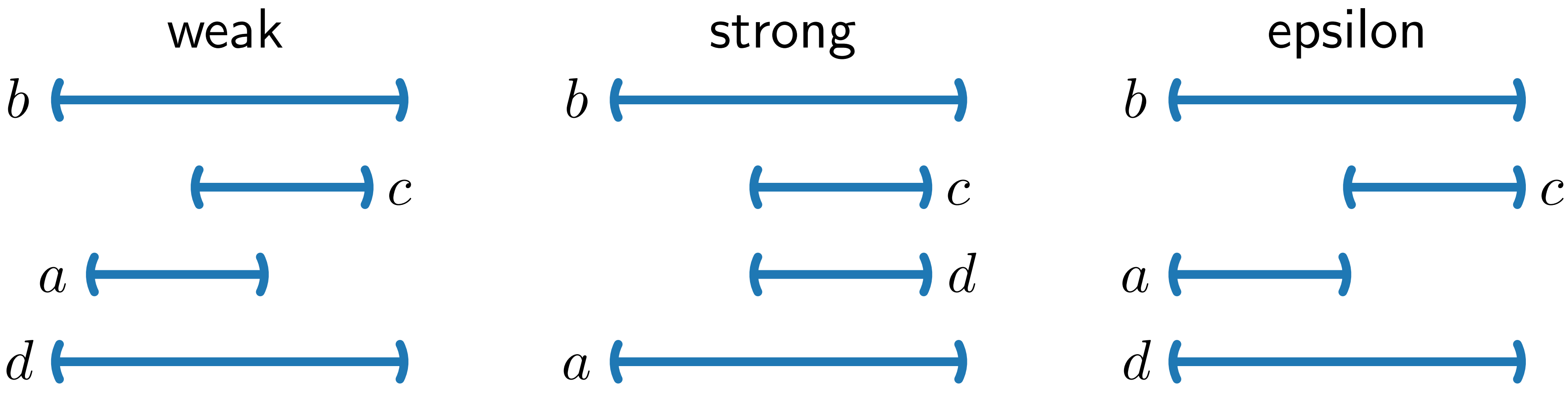




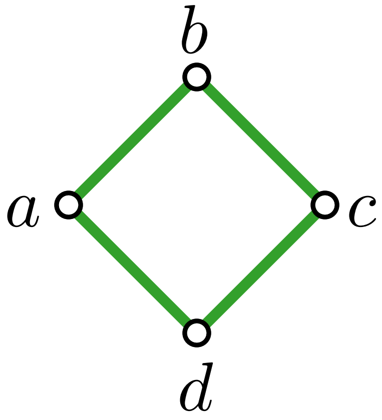


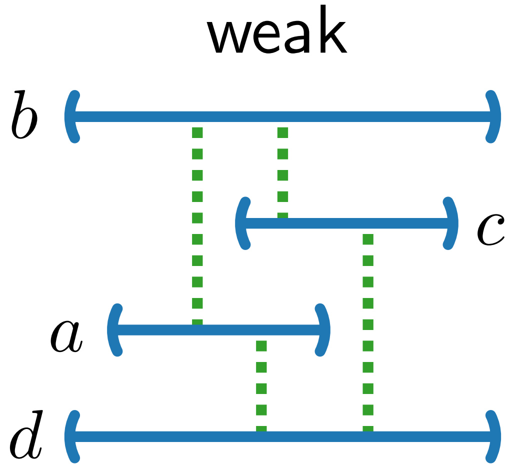
Problems

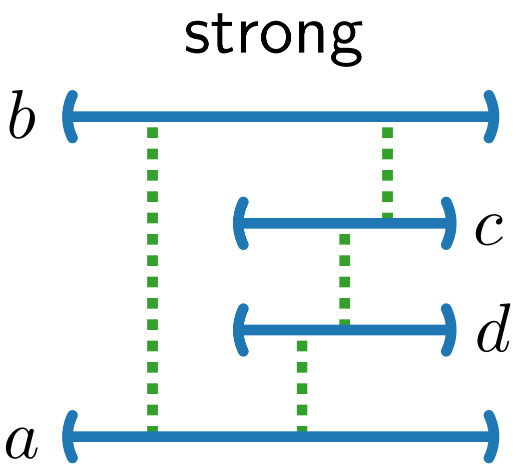


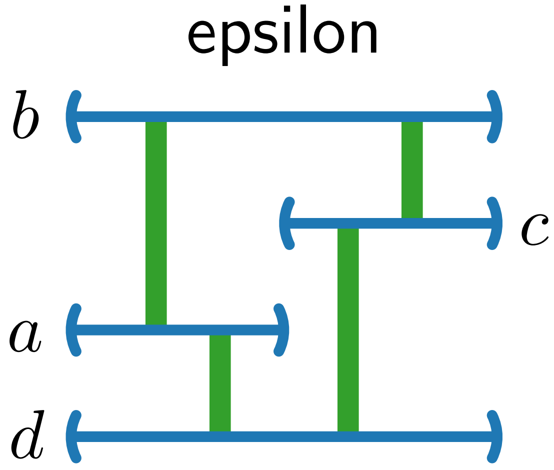


Problems

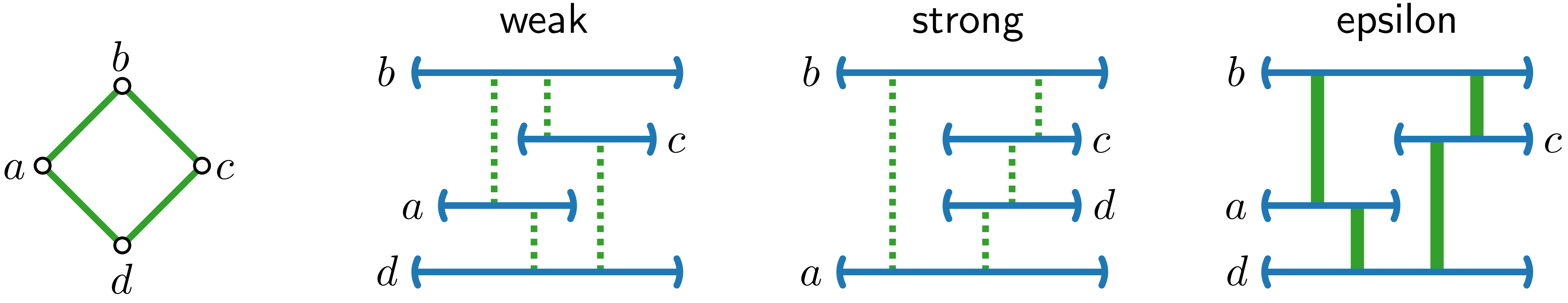






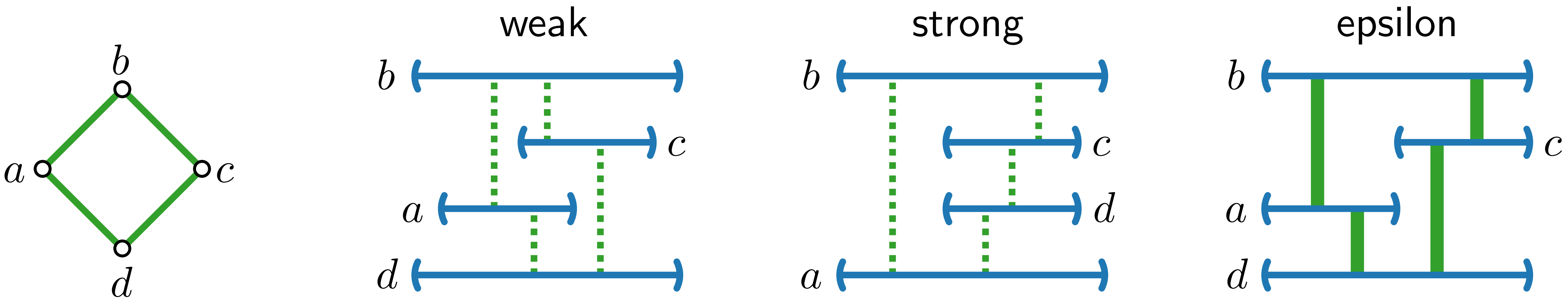


Problems



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

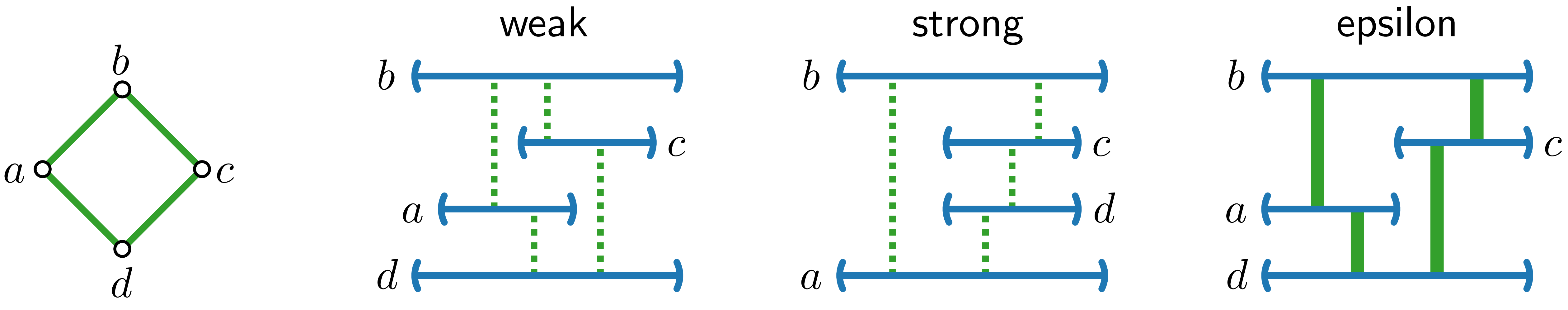
Problems



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

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

Problems

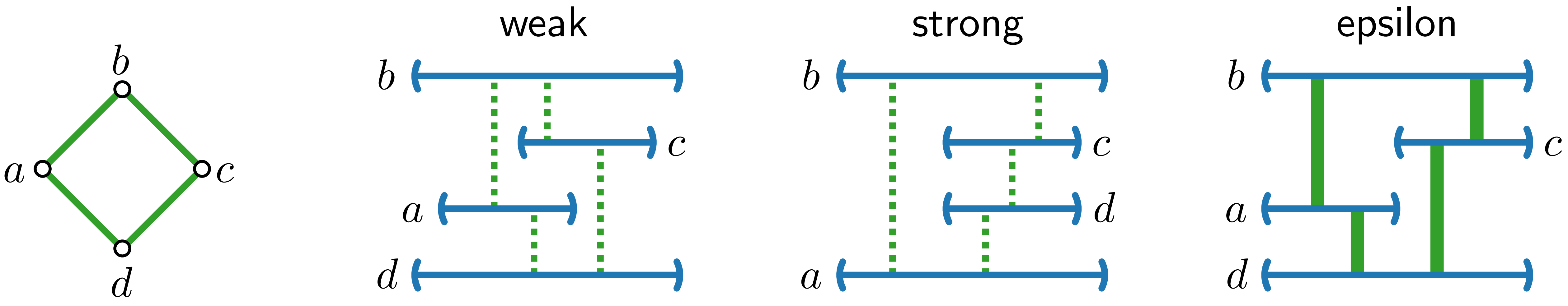


Recognition Problem.
Given a graph G, **decide** whether there exists a weak/strong/ε bar visibility representation ψ of G.

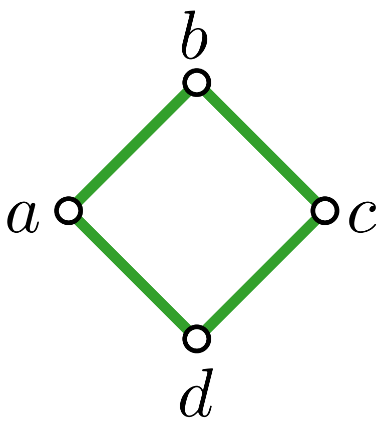
Construction Problem.
Given a graph G, **construct** a weak/strong/ε bar visibility representation ψ of G – if one exists.

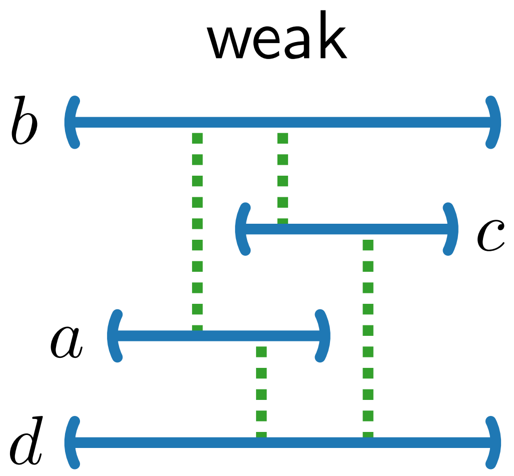
Partial Representation Extension Problem.
Given a graph G and a **set of bars** ψ' **of** $V' \subset V(G)$, **decide** whether there exists a weak/strong/ε bar visibility representation ψ of G **where** $\psi|_{V'} = \psi'$ (and **construct** ψ if a representation exists).

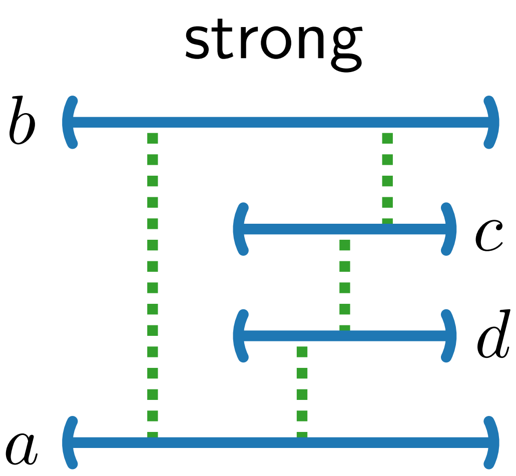
Background

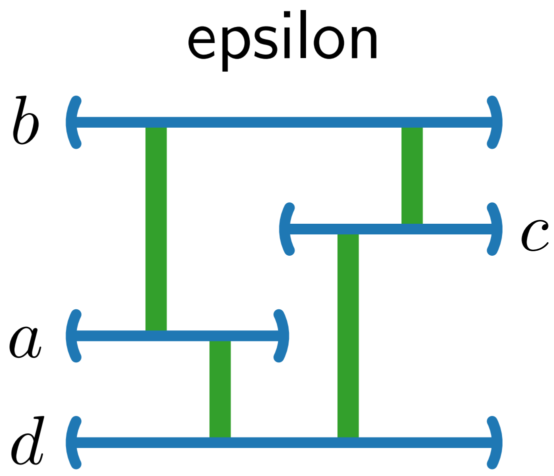


Background



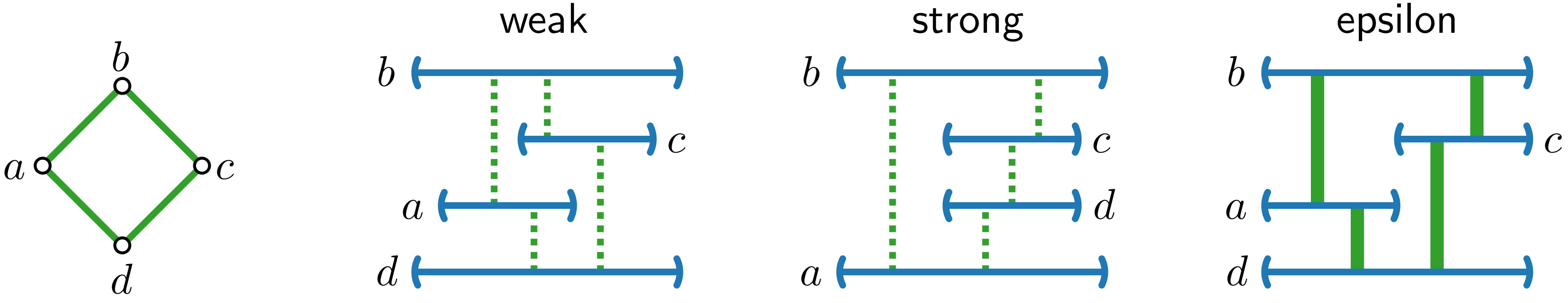






Weak Bar Visibility.

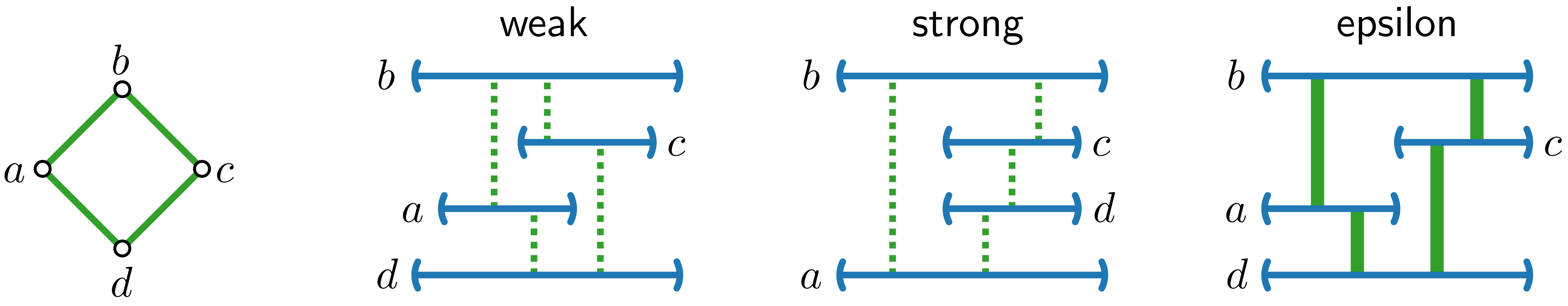
Background



Weak Bar Visibility.

- All planar graphs. [Tamassia & Tollis '86; Wismath '85]

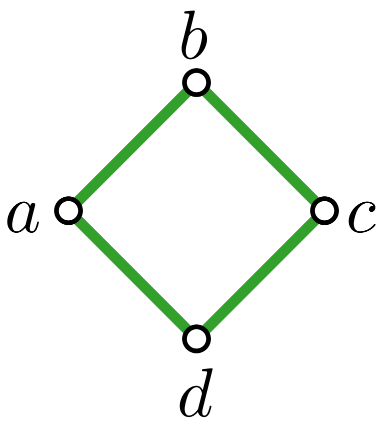
Background

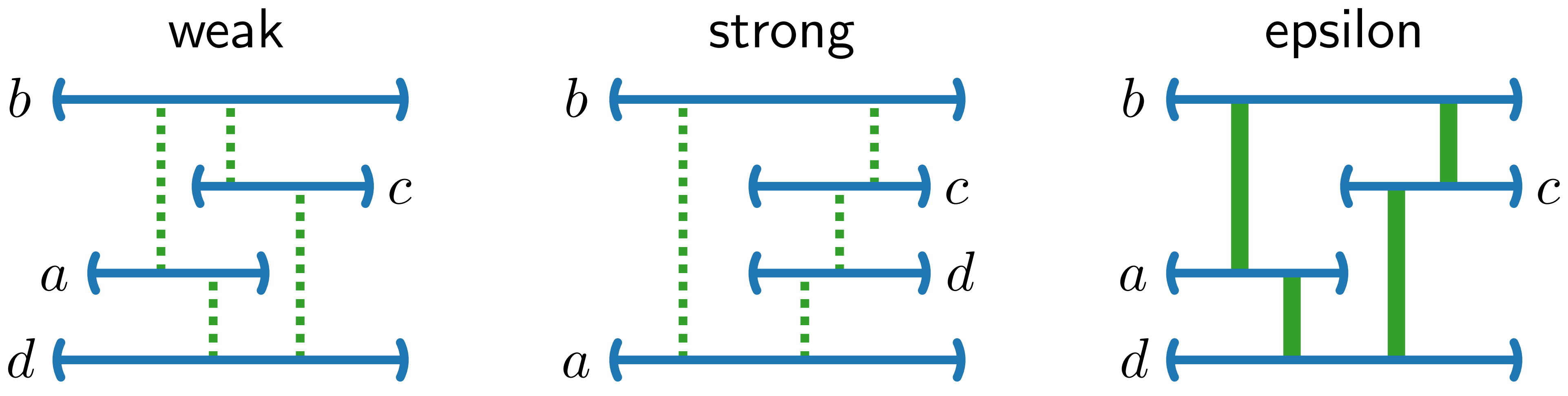


Weak Bar Visibility.

- All planar graphs. [Tamassia & Tollis ’86; Wismath ’85]
- Linear time recognition and construction [T&T ’86]

Background

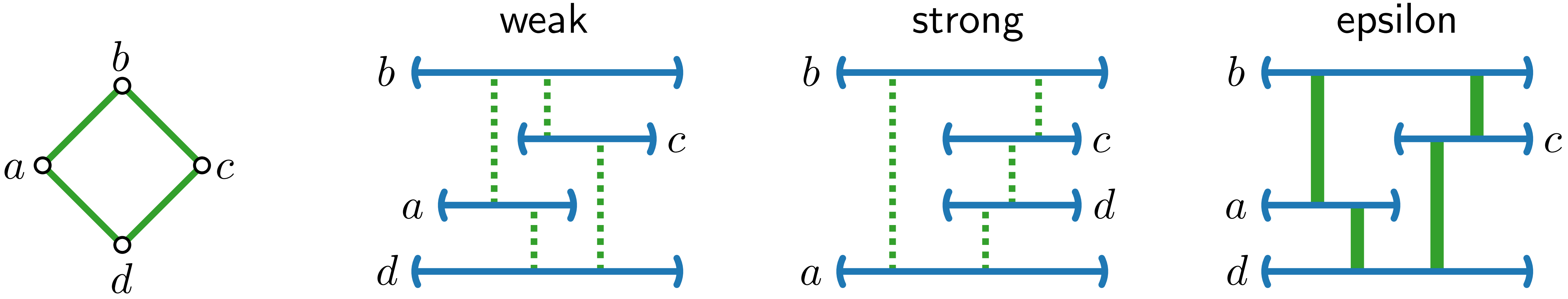




Weak Bar Visibility.

- All planar graphs. [Tamassia & Tollis '86; Wismath '85]
- Linear time recognition and construction [T&T '86]
- Representation Extension is NP-complete [Chaplick et al. '14]

Background

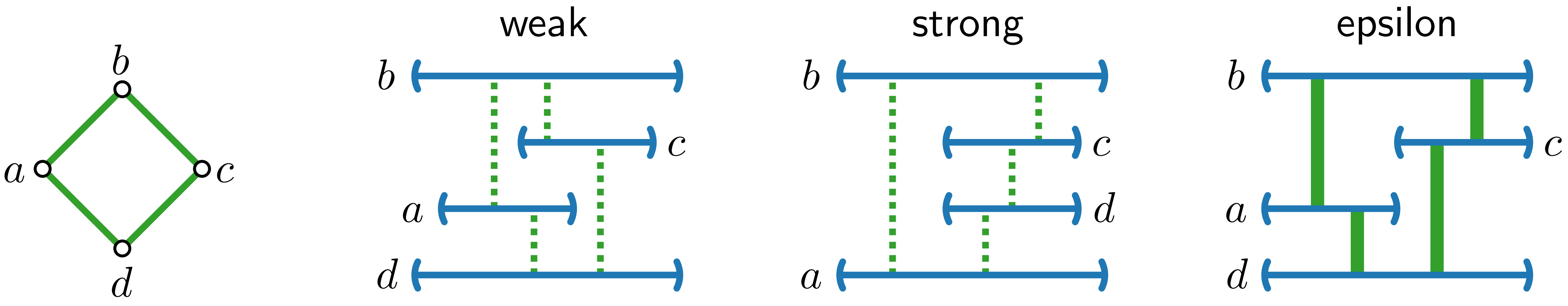


Weak Bar Visibility.

- All planar graphs. [Tamassia & Tollis '86; Wismath '85]
- Linear time recognition and construction [T&T '86]
- Representation Extension is NP-complete [Chaplick et al. '14]

Strong Bar Visibility.

Background



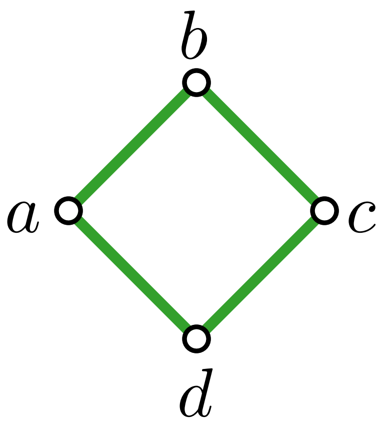
Weak Bar Visibility.

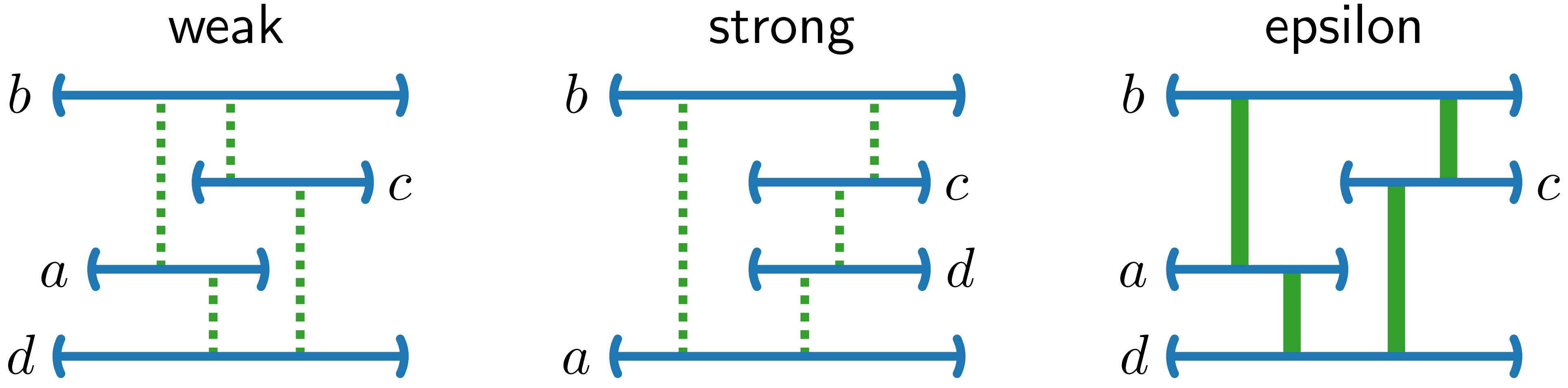
- All planar graphs. [Tamassia & Tollis '86; Wismath '85]
- Linear time recognition and construction [T&T '86]
- Representation Extension is NP-complete [Chaplick et al. '14]

Strong Bar Visibility.

- NP-complete to recognize [Andreae '92]

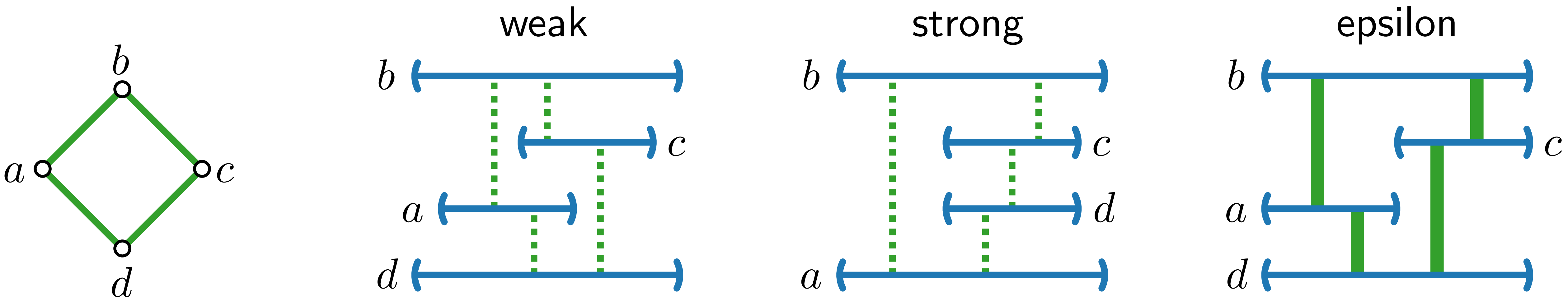
Background





ε-Bar Visibility.

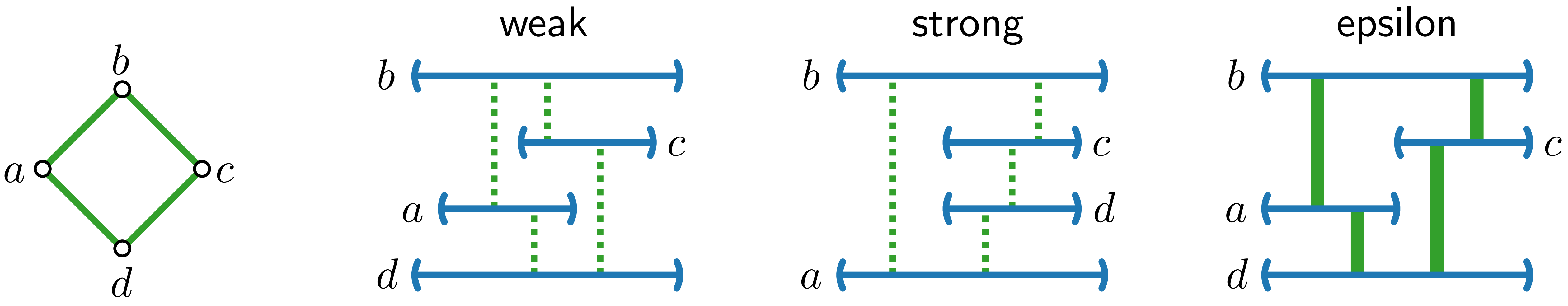
Background



ε-Bar Visibility.

- Planar graphs that can be embedded with all **cut vertices** on the outerface. [T&T '86, Wismath '85]

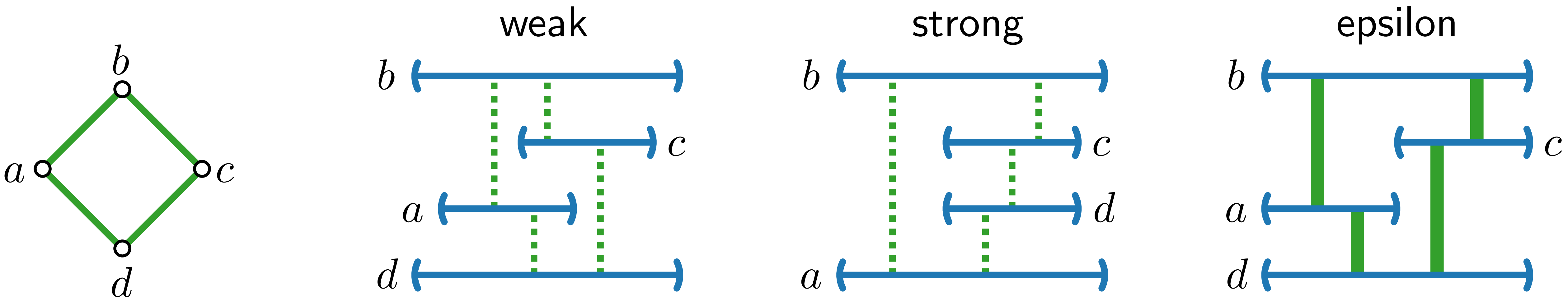
Background



ε-Bar Visibility.

- Planar graphs that can be embedded with all **cut vertices** on the outerface. [T&T '86, Wismath '85]
- Linear-time recognition and construction [T&T '86]

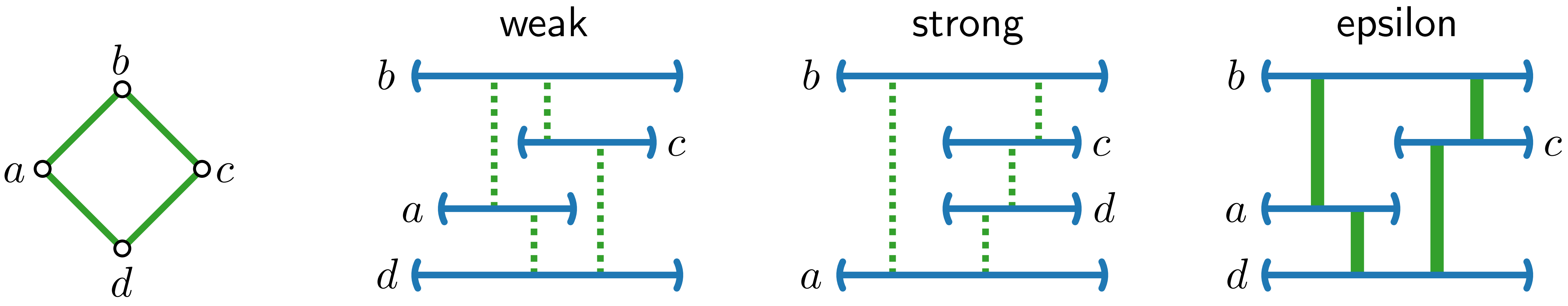
Background



ε-Bar Visibility.

- Planar graphs that can be embedded with all **cut vertices** on the outerface. [T&T '86, Wismath '85]
- Linear-time recognition and construction [T&T '86]
- Representation extension?

Background



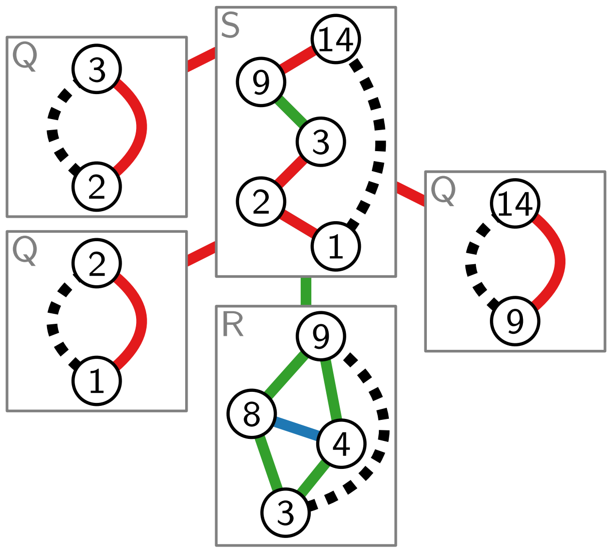
ε-Bar Visibility.

- Planar graphs that can be embedded with all **cut vertices** on the outerface. [T&T '86, Wismath '85]
- Linear-time recognition and construction [T&T '86]
- Representation extension? This Lecture!

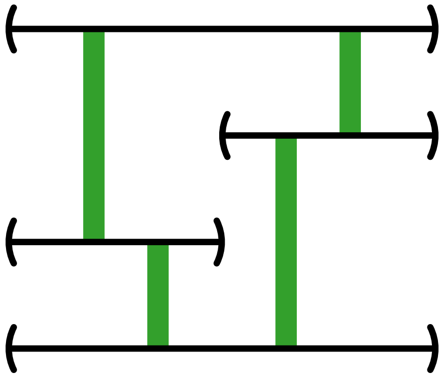
Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Visualization of Graphs

Lecture 9:
Partial Visibility Representation Extension



Part II:
Recognition & Construction



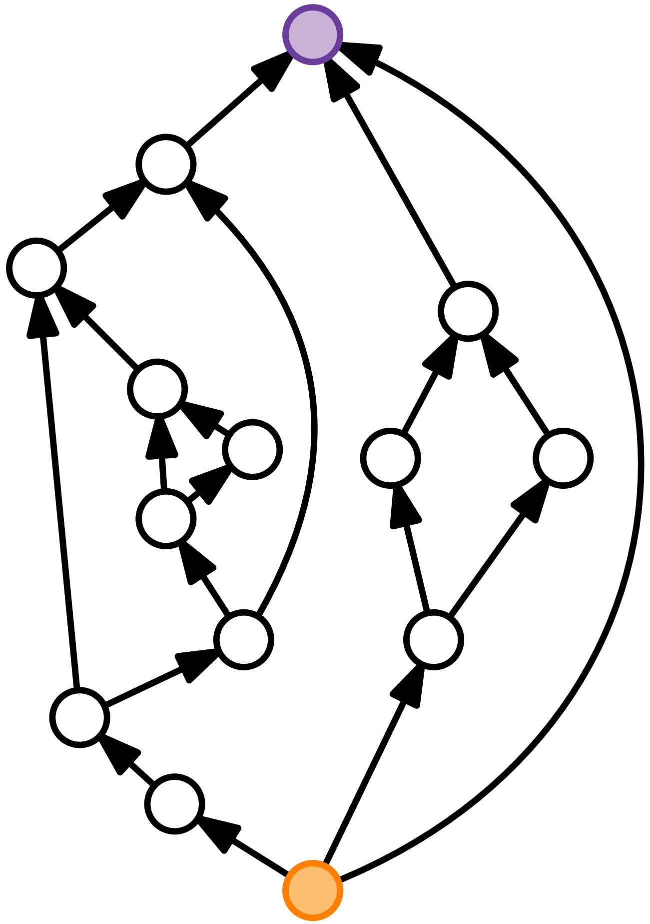
Alexander Wolff

ε-bar Visibility and st-Graphs

Recall that an **st-graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

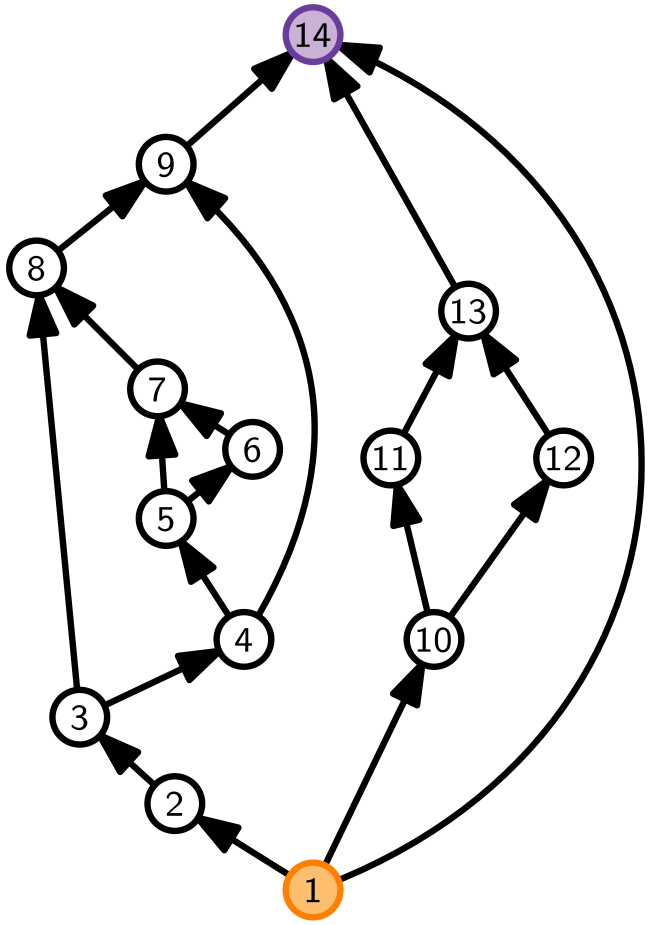
ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.



ε-bar Visibility and st-Graphs

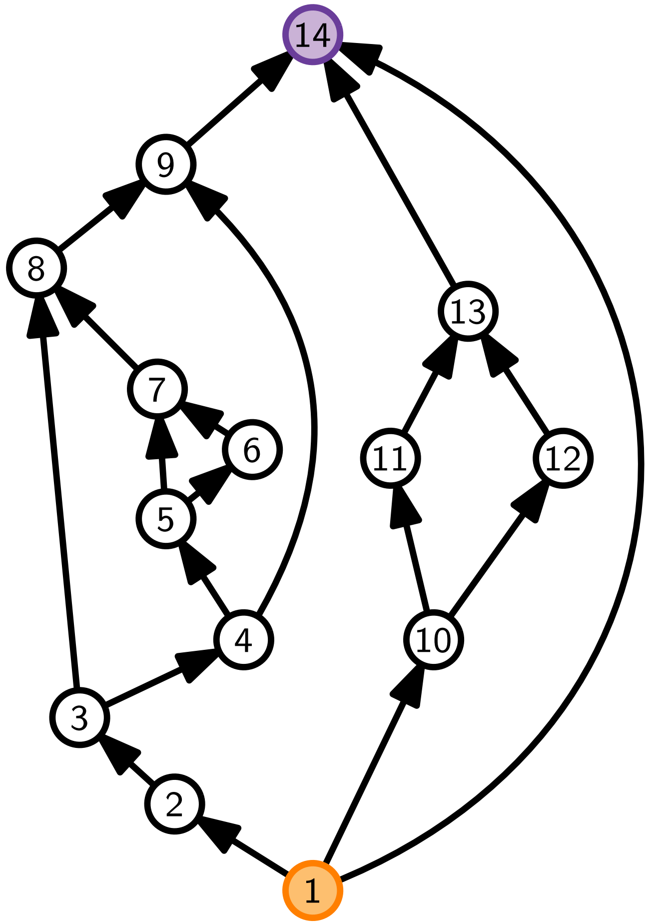
Recall that an **st-graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.



ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

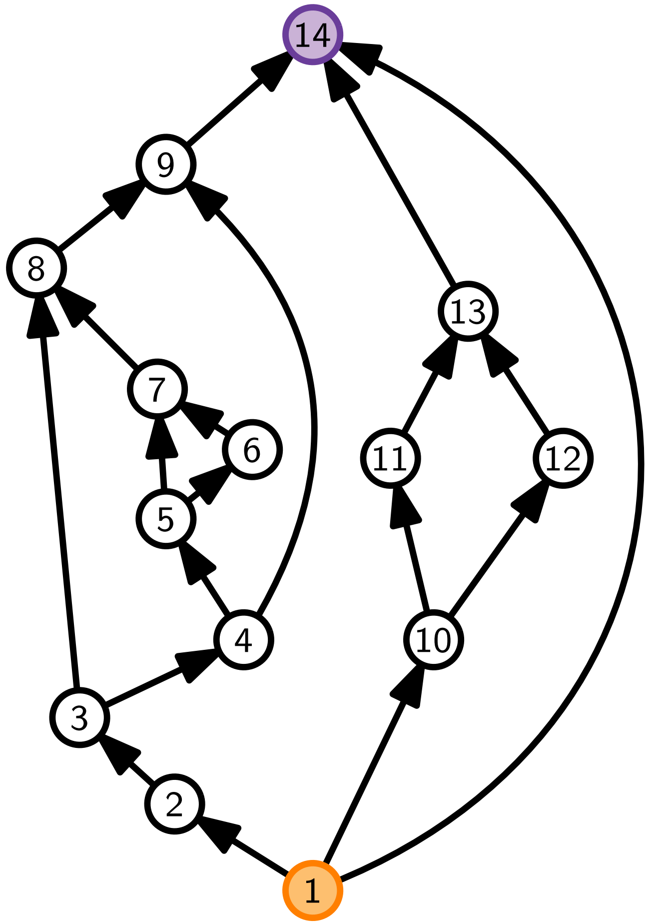
Observation.
st-orientations correspond to ε-bar visibility representations.



ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

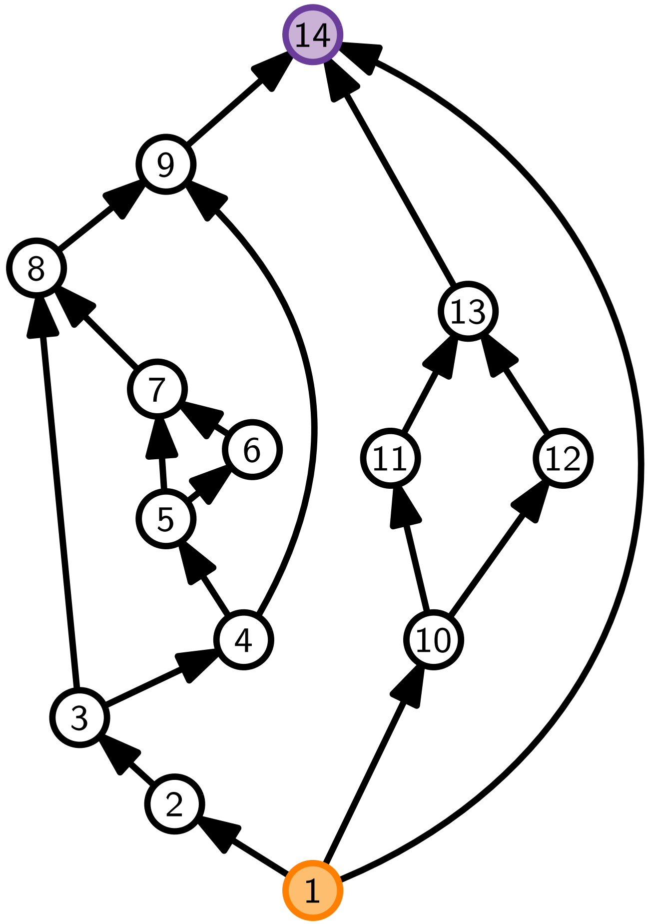
Observation.
st-orientations correspond to ε-bar visibility representations.

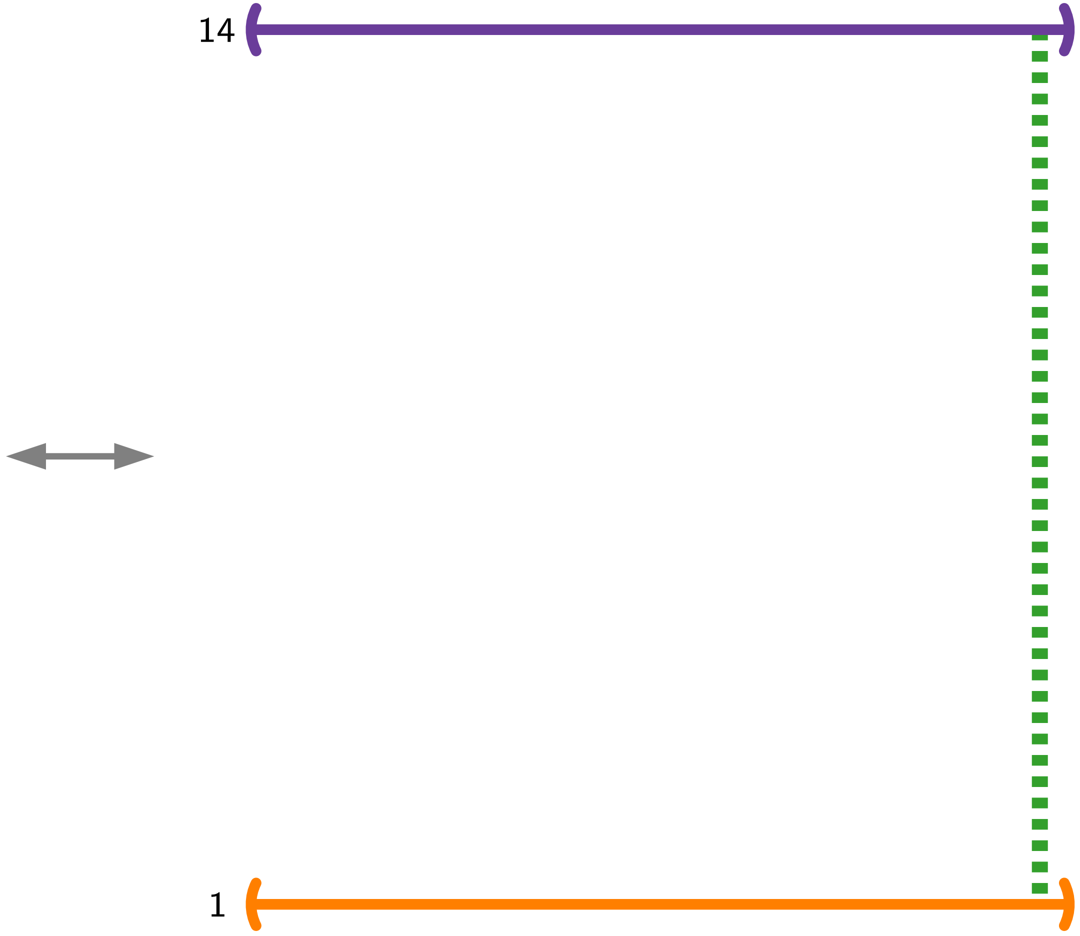


ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

Observation.
st-orientations correspond to ε-bar visibility representations.

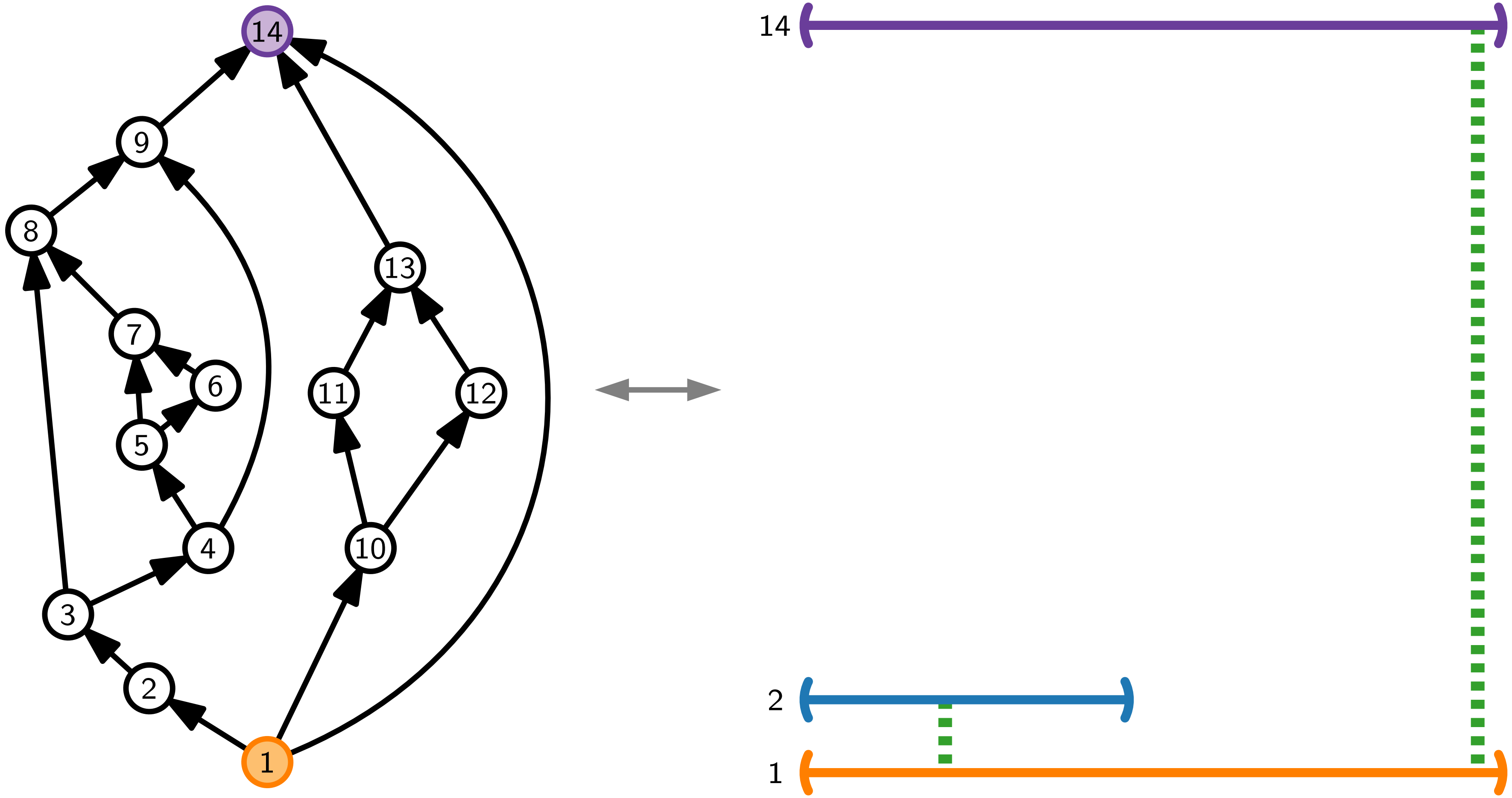




ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

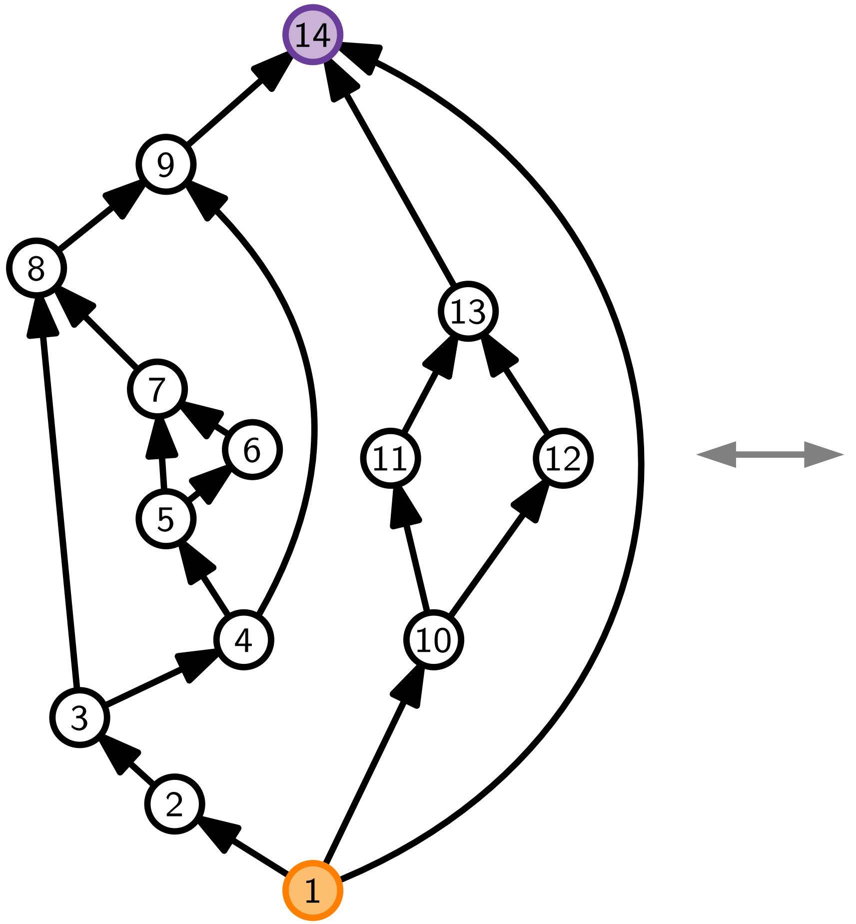
Observation.
st-orientations correspond to ε-bar visibility representations.

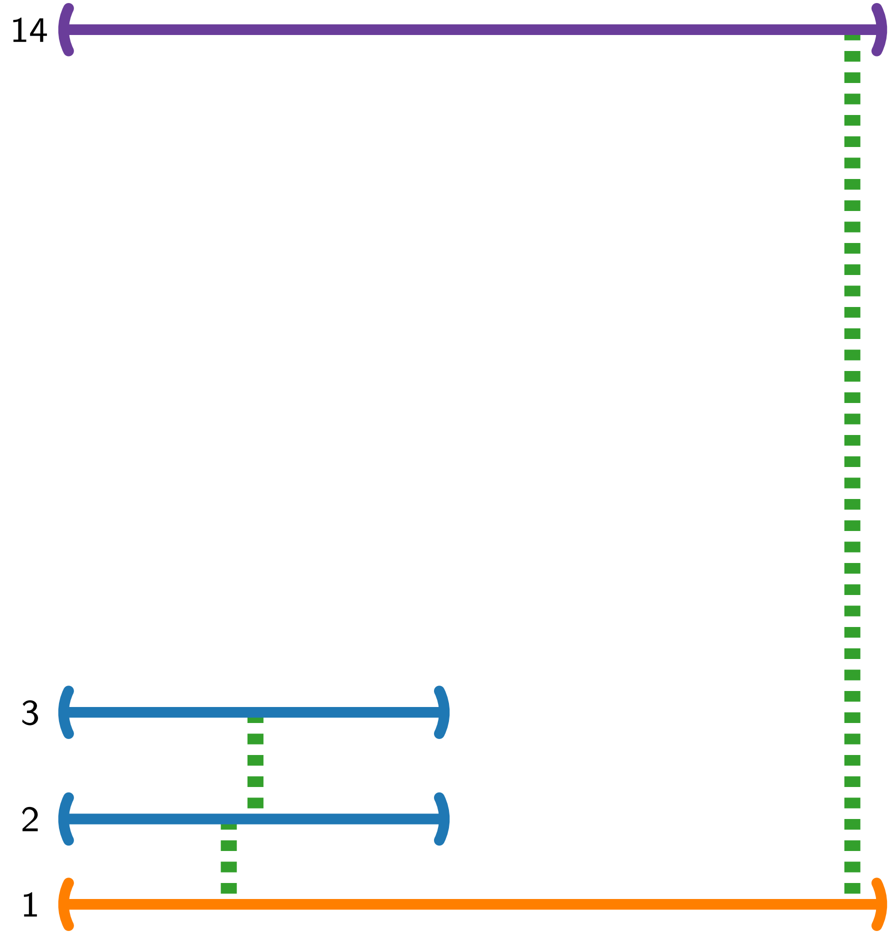


ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

Observation.
st-orientations correspond to ε-bar visibility representations.

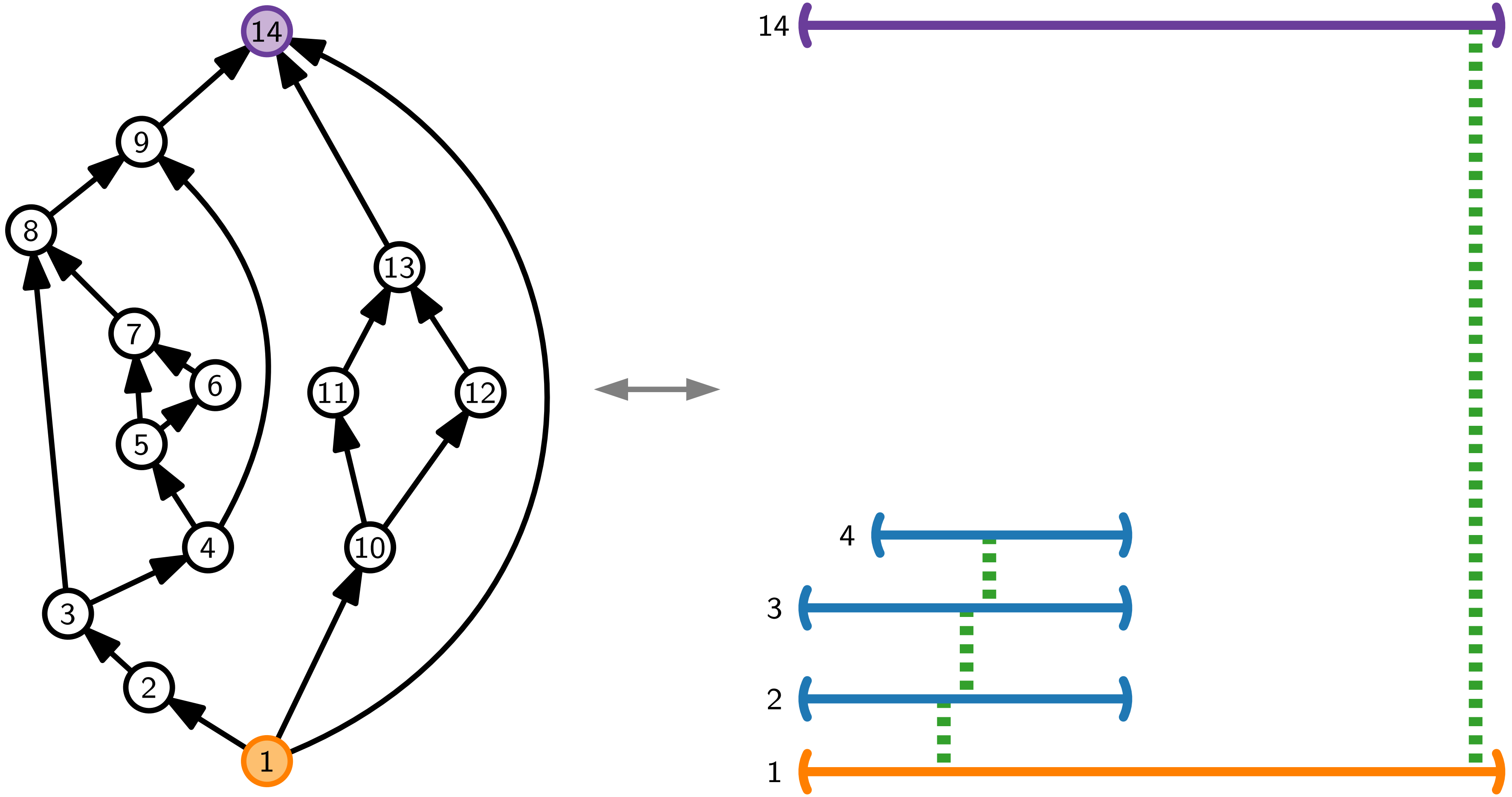




ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

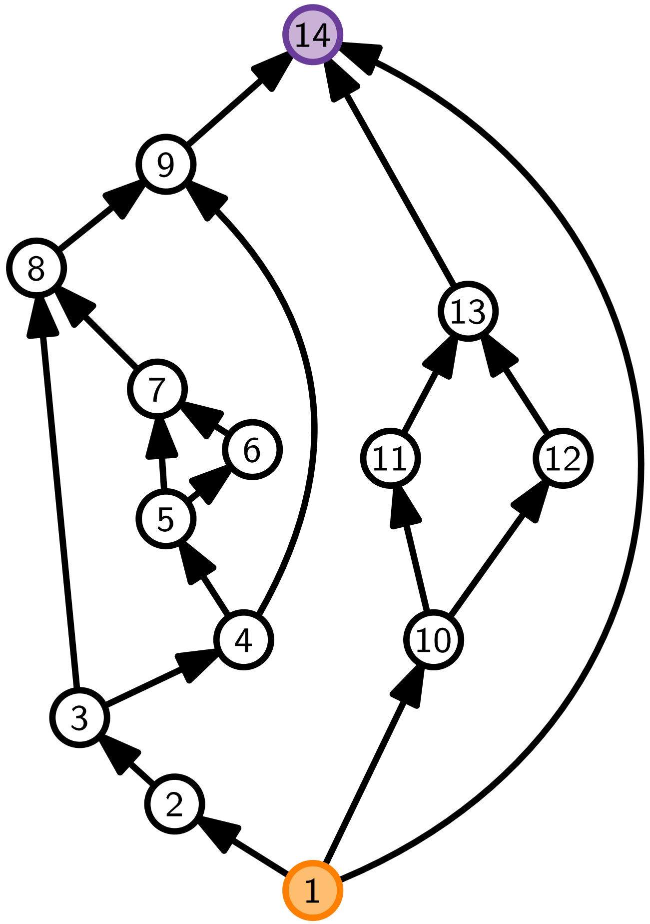
Observation.
st-orientations correspond to ε-bar visibility representations.

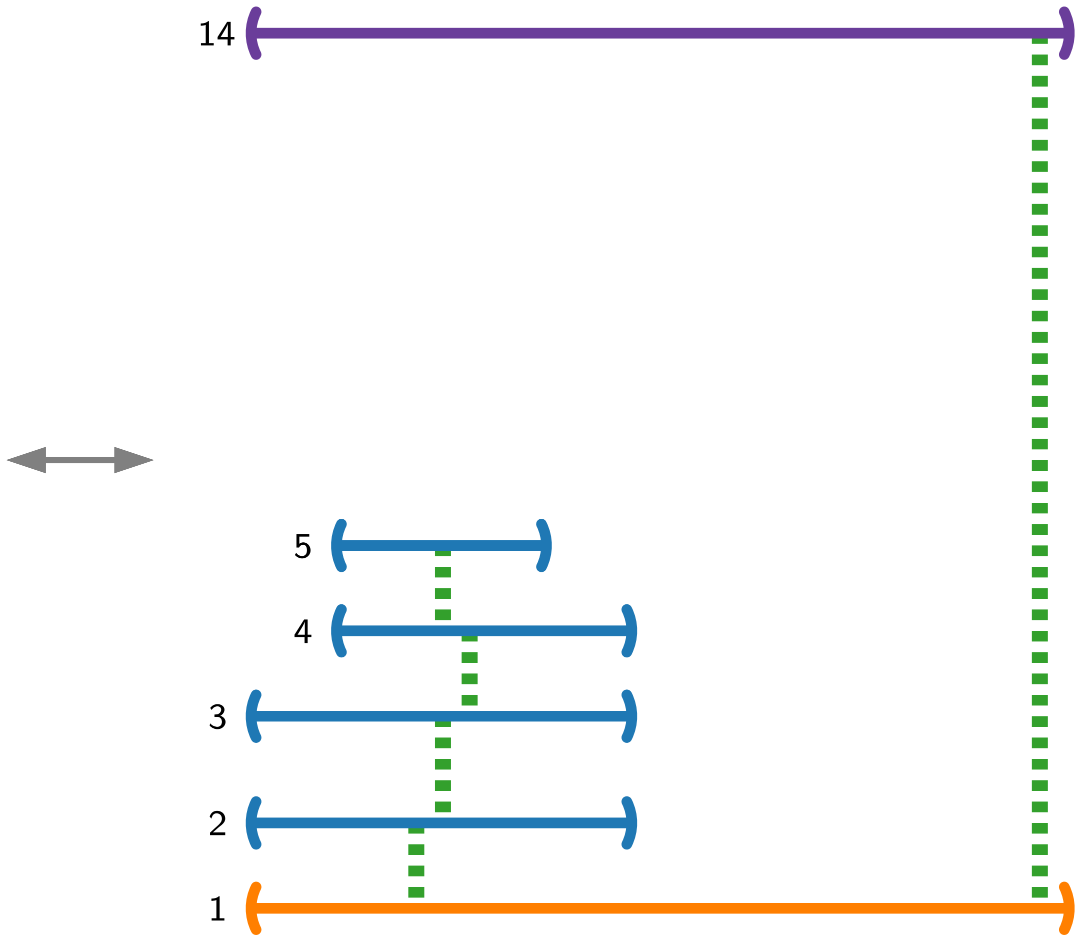


ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

Observation.
st-orientations correspond to ε-bar visibility representations.

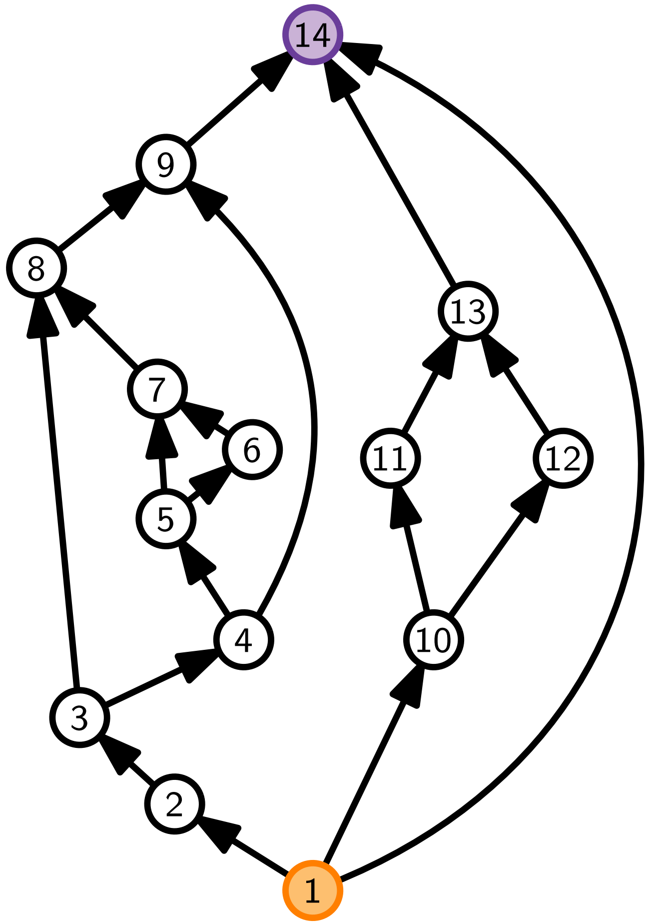


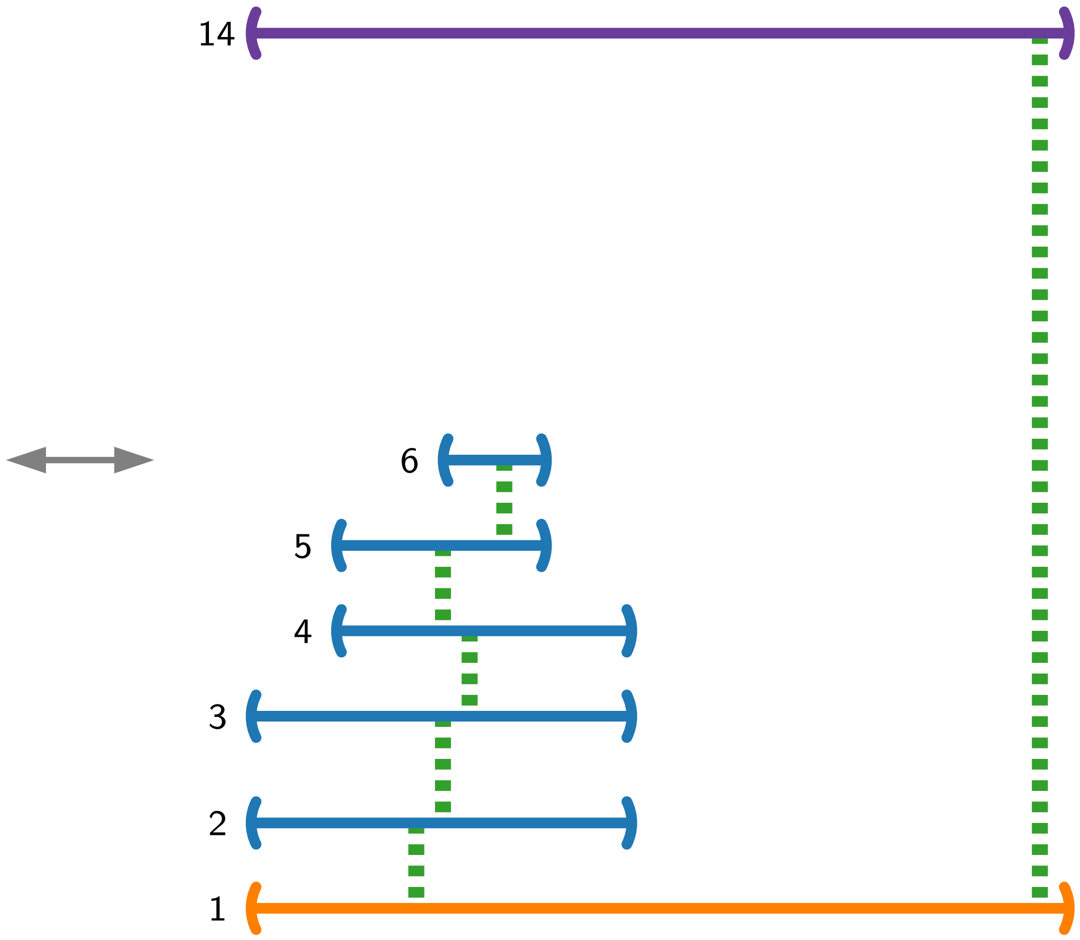


ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

Observation.
st-orientations correspond to ε-bar visibility representations.

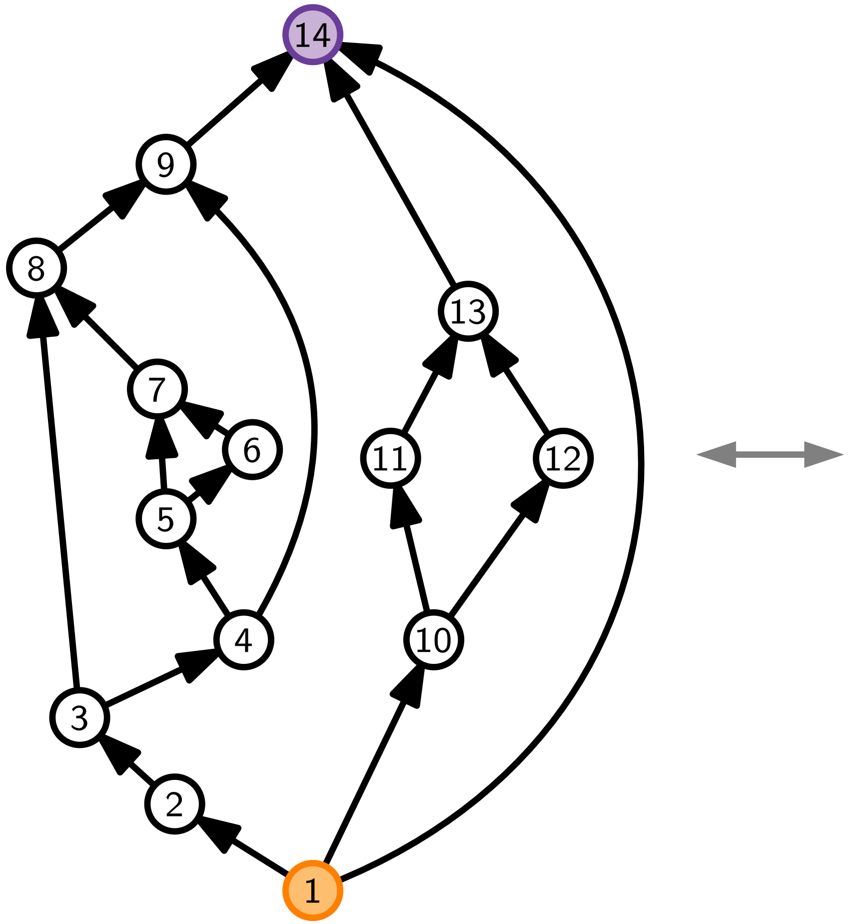


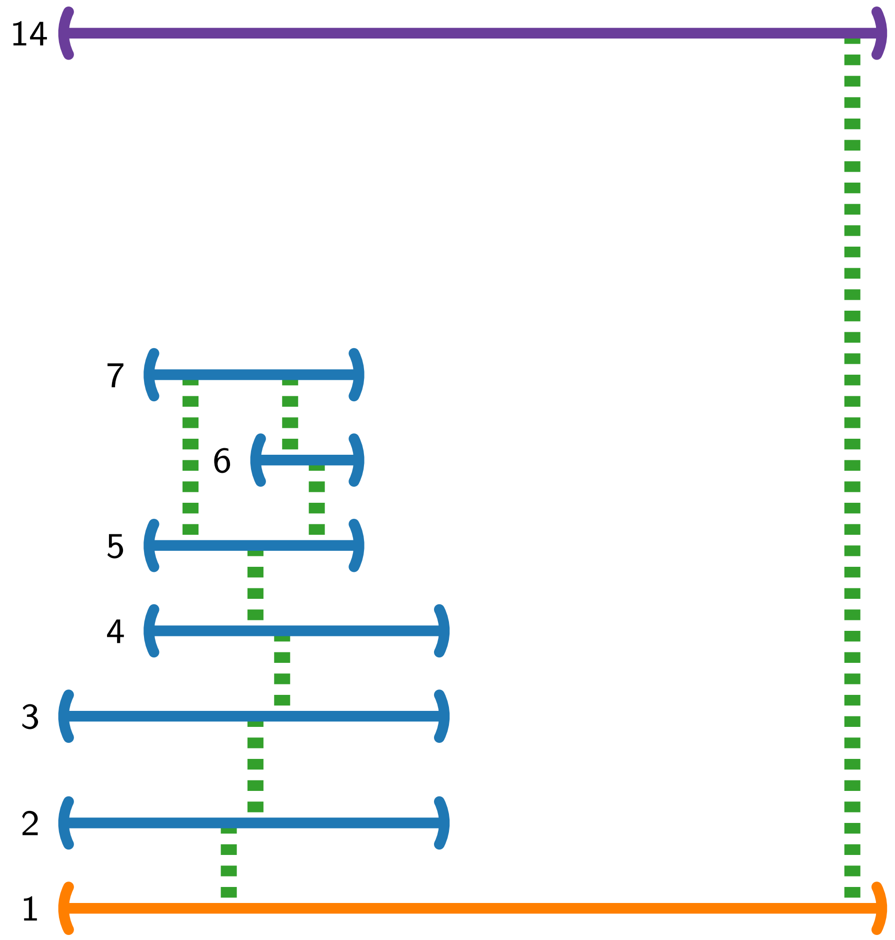


ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

Observation.
st-orientations correspond to ε-bar visibility representations.

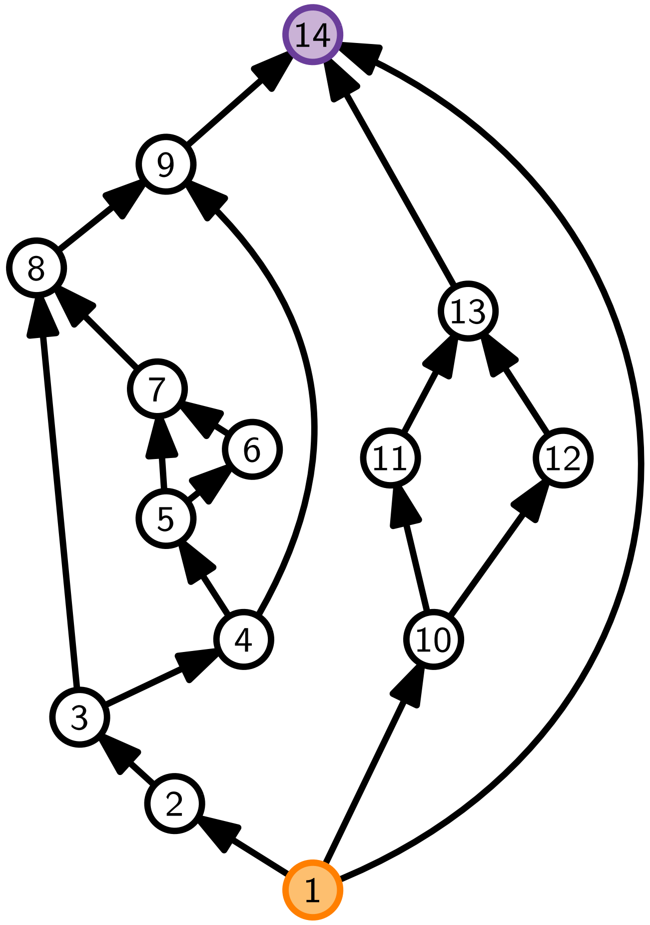


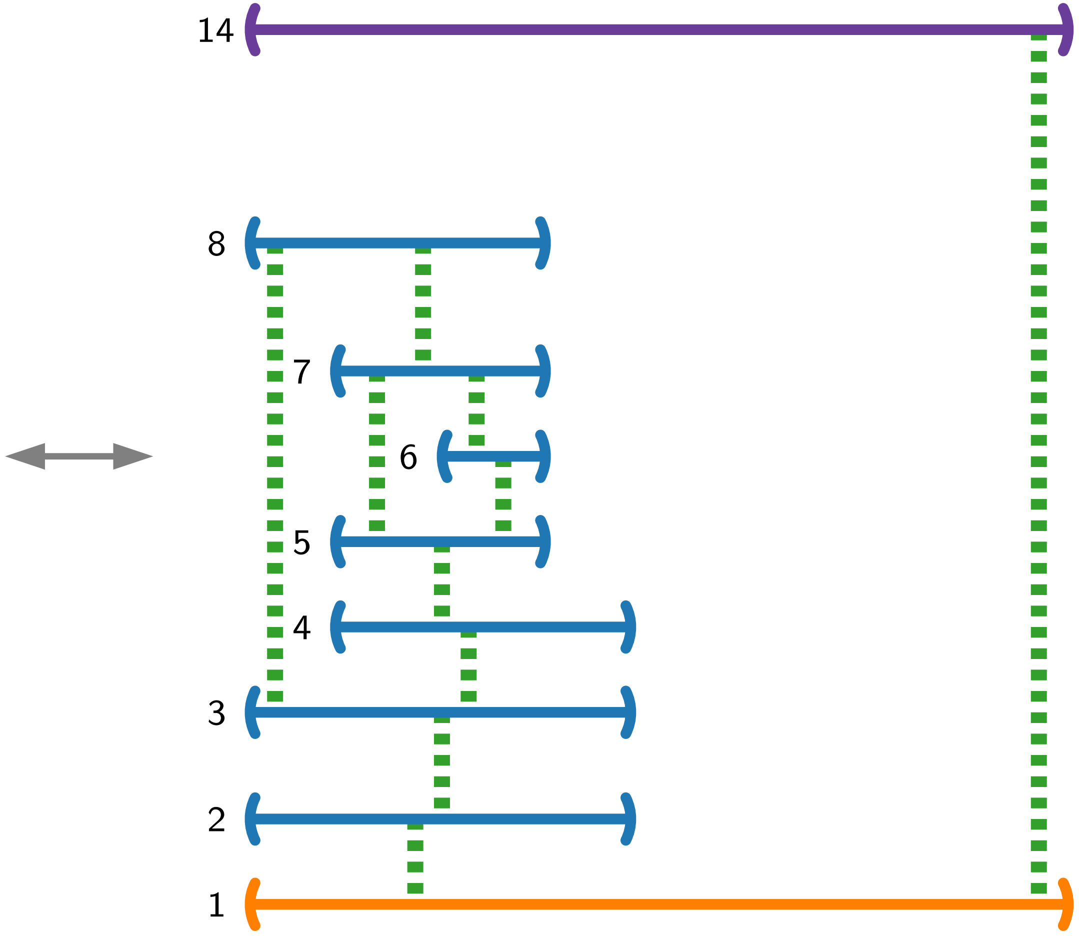


ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

Observation.
st-orientations correspond to ε-bar visibility representations.

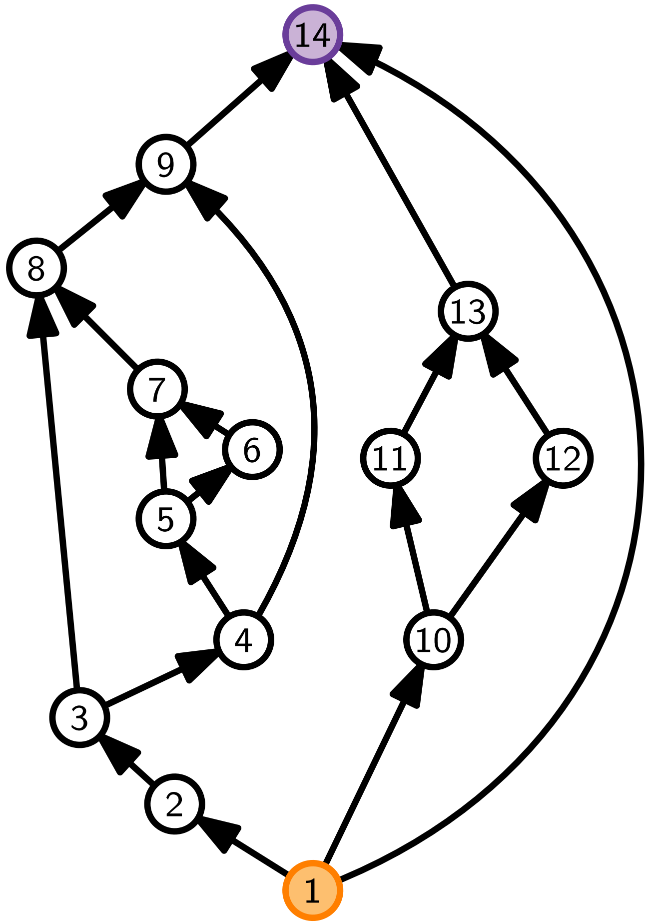


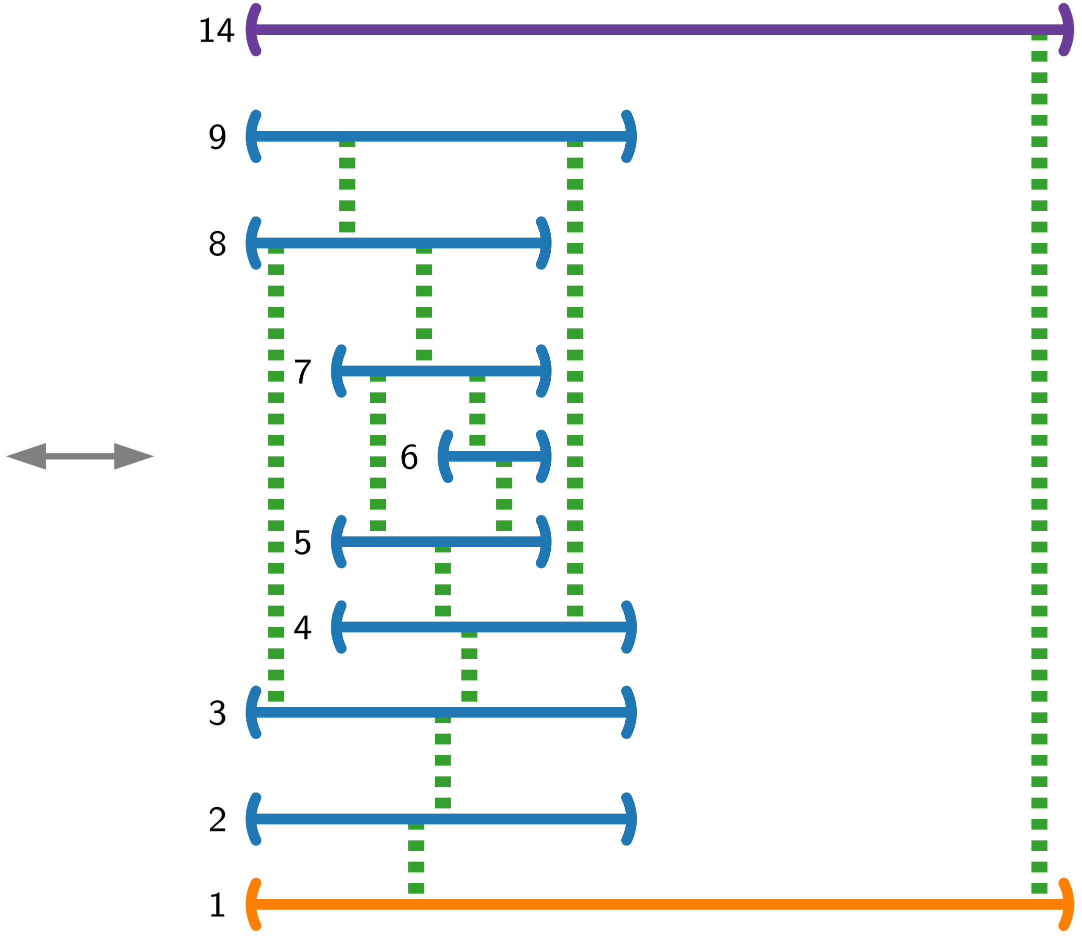


ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

Observation.
st-orientations correspond to ε-bar visibility representations.

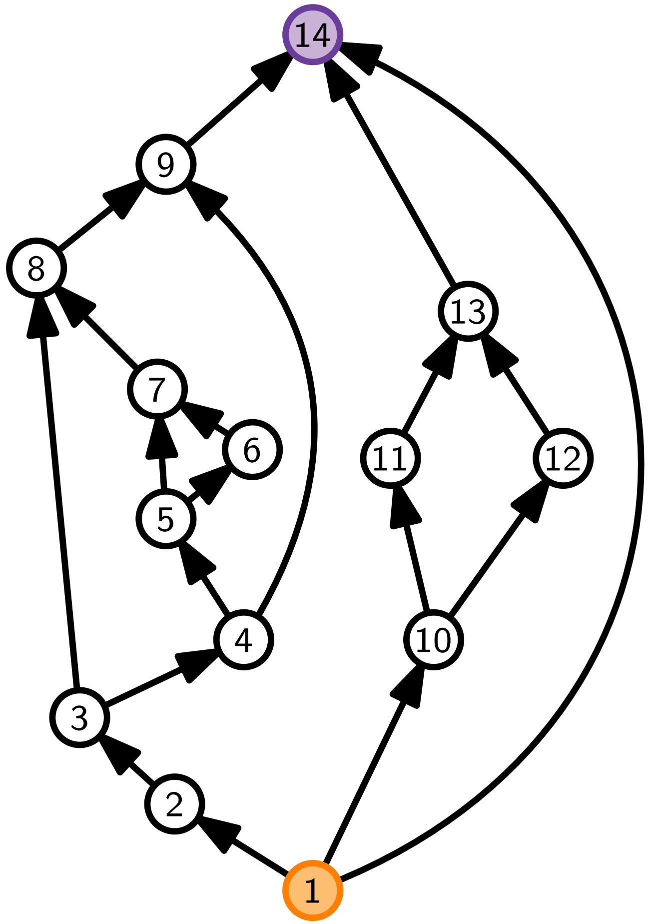


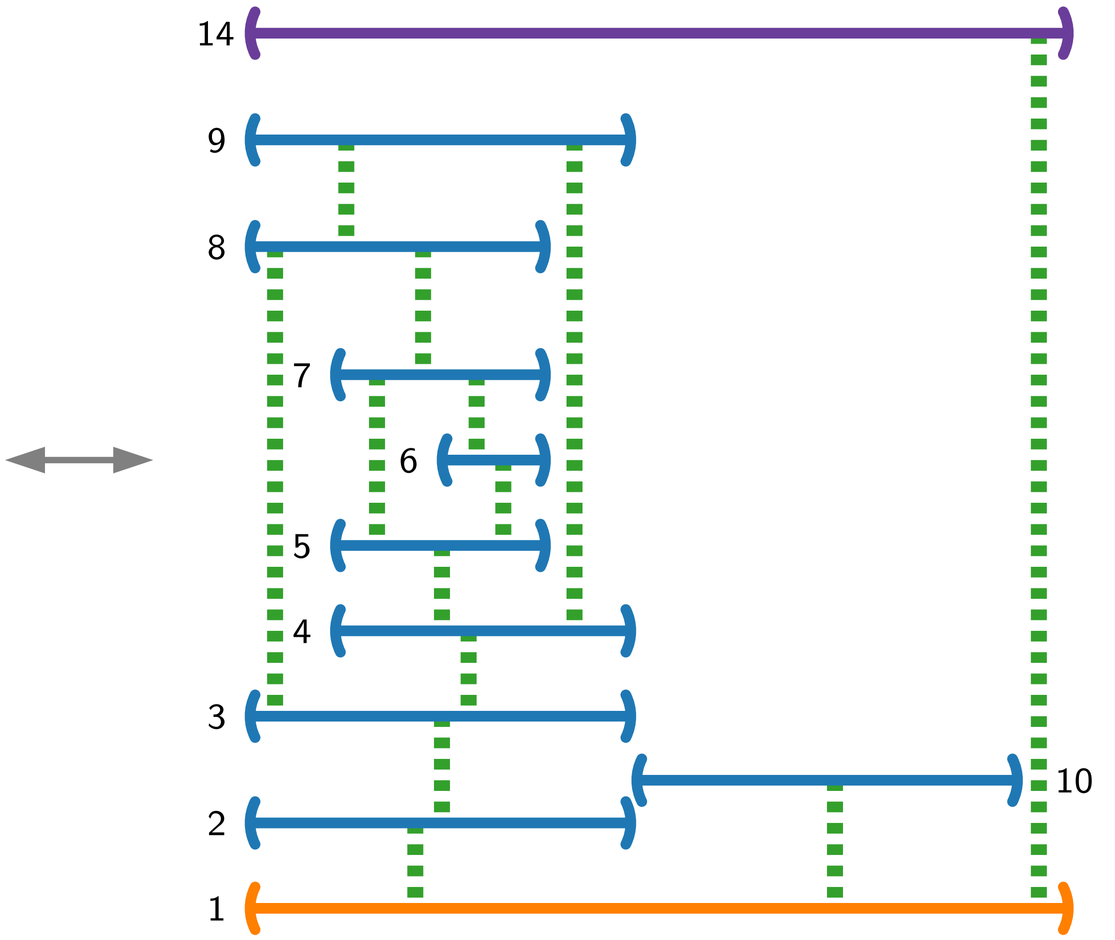


ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

Observation.
st-orientations correspond to ε-bar visibility representations.

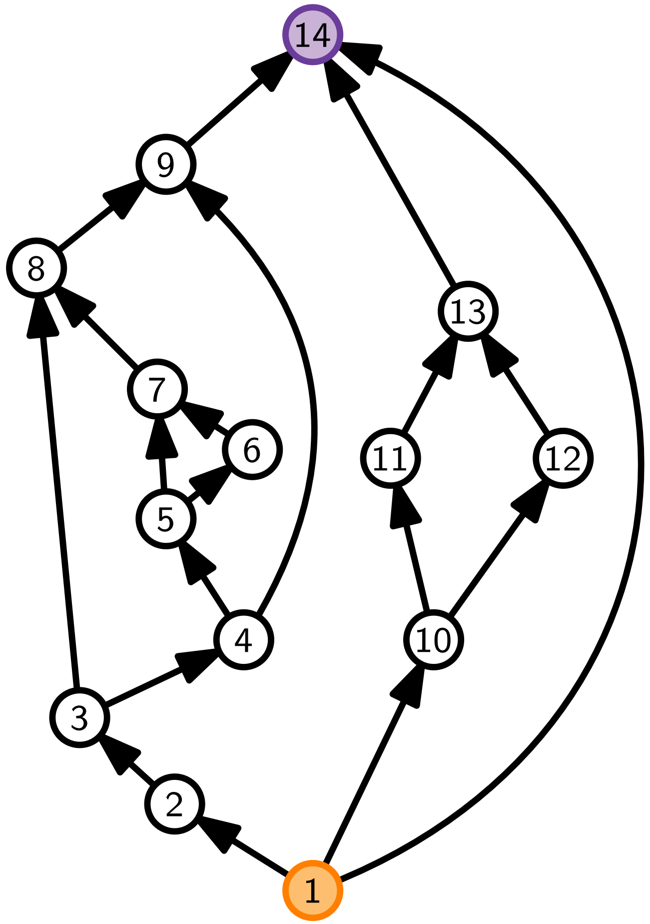


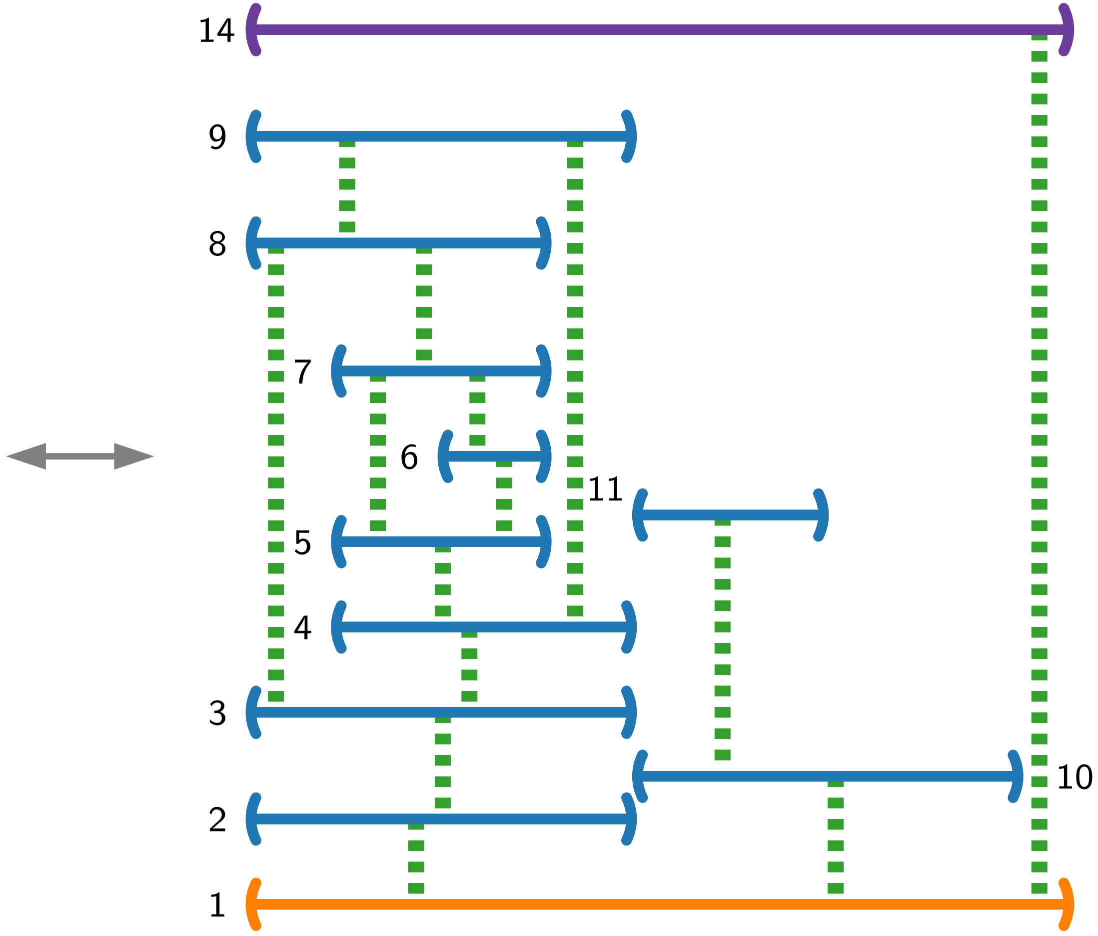


ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

Observation.
st-orientations correspond to ε-bar visibility representations.

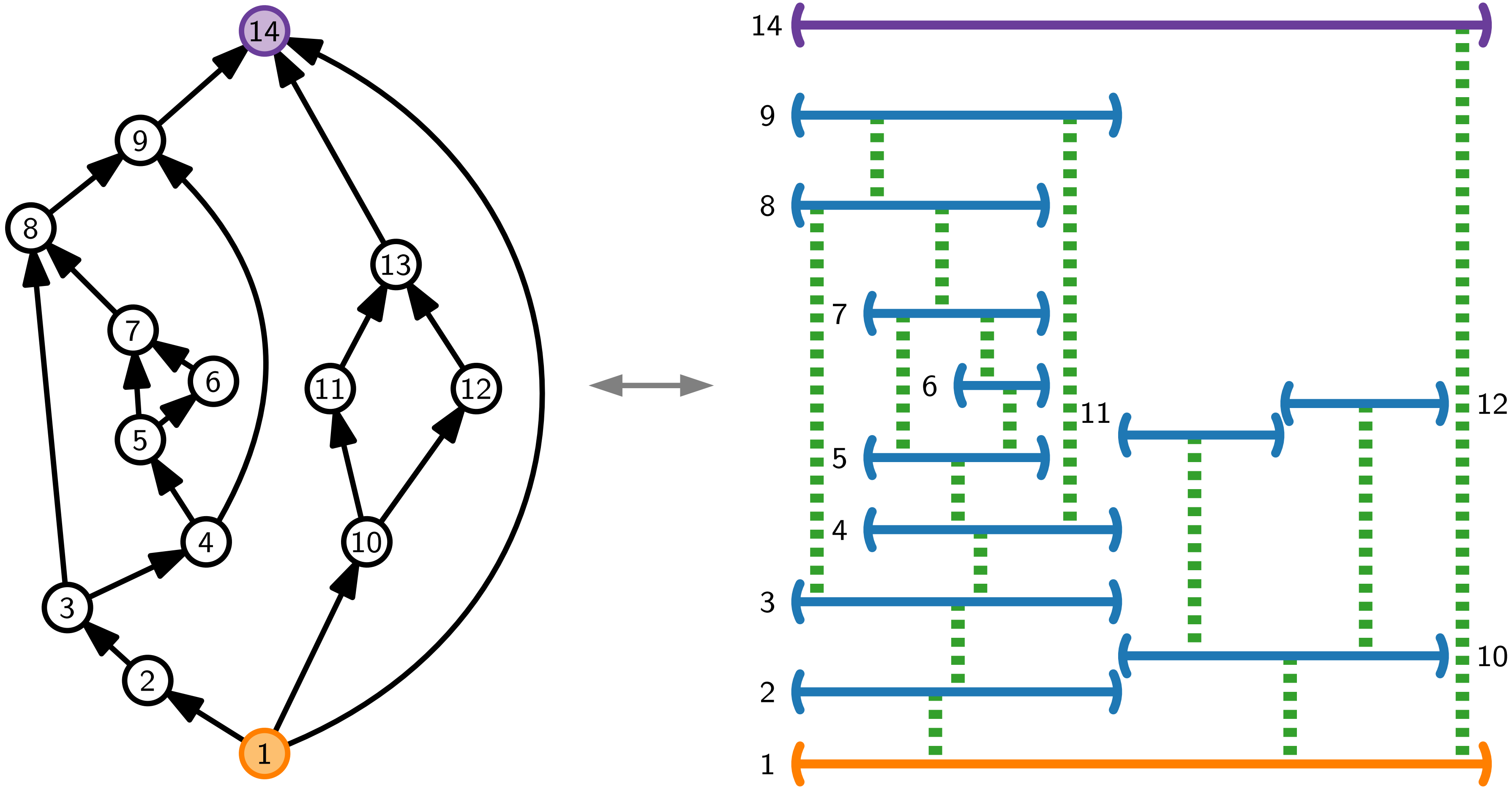




ε-bar Visibility and st-Graphs

Recall that an **st-graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

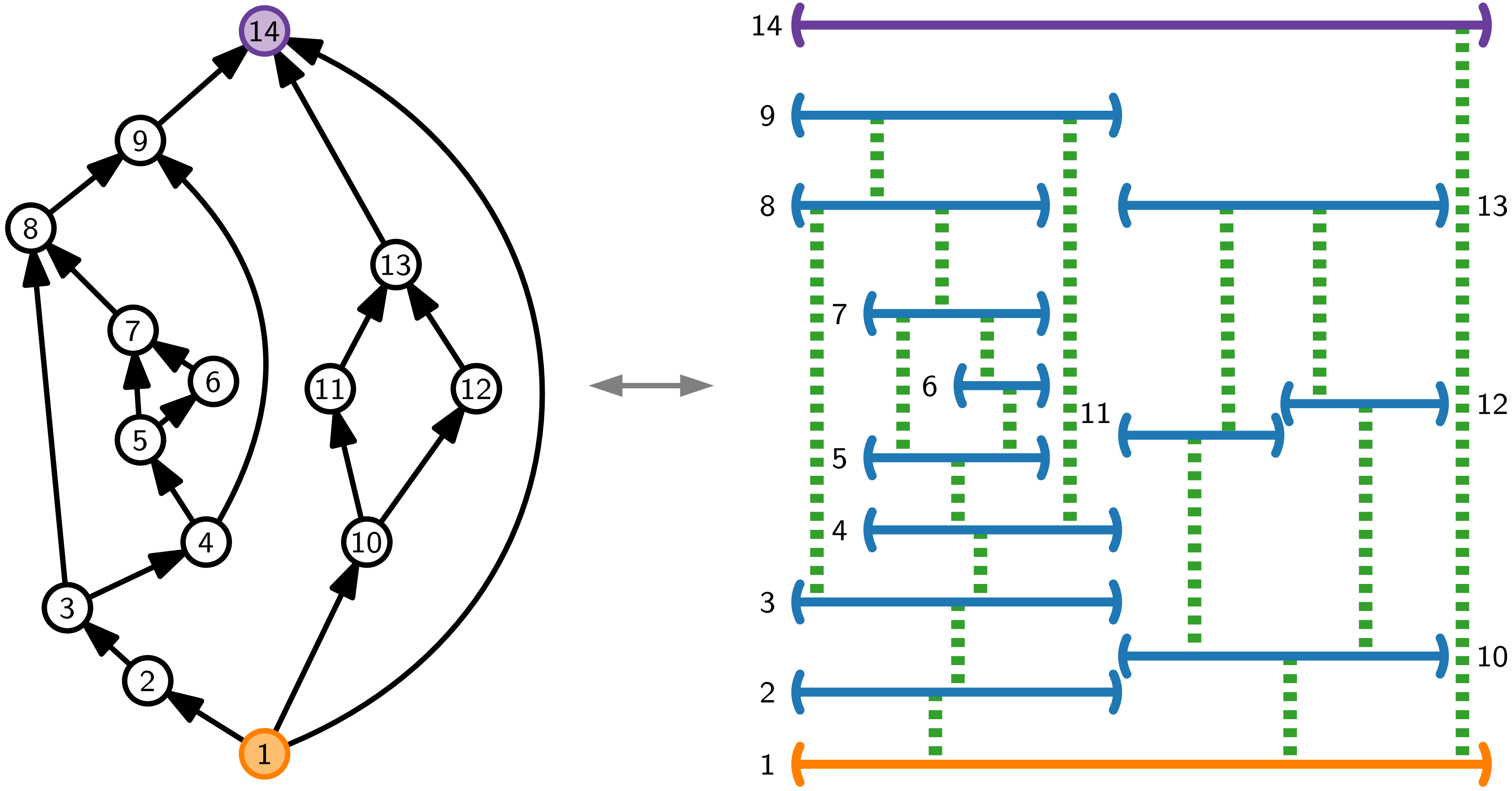
Observation.
st-orientations correspond to ε-bar visibility representations.



ε-bar Visibility and st-Graphs

Recall that an ***st*-graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

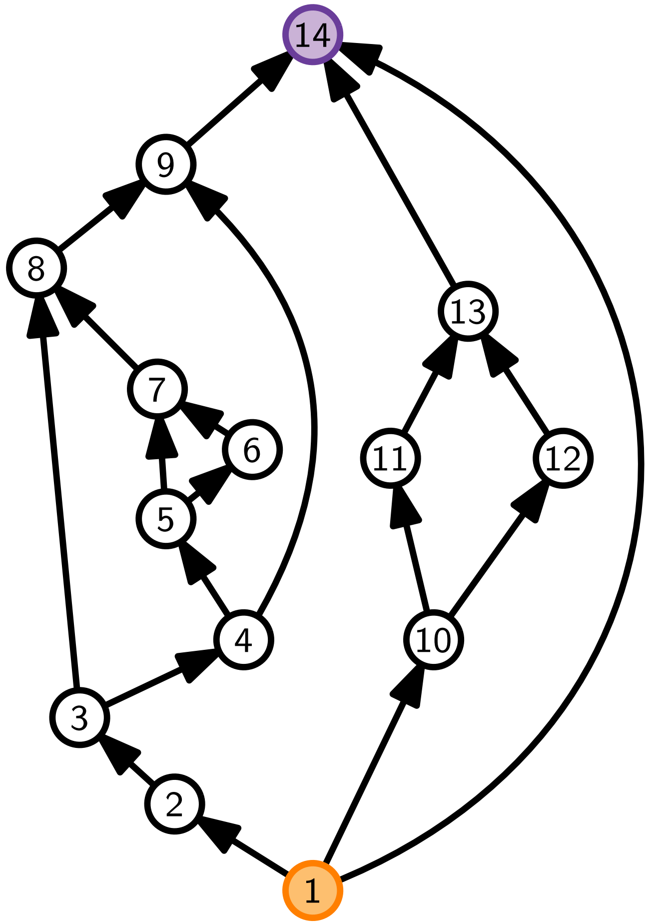
Observation.
st-orientations correspond to ε-bar visibility representations.

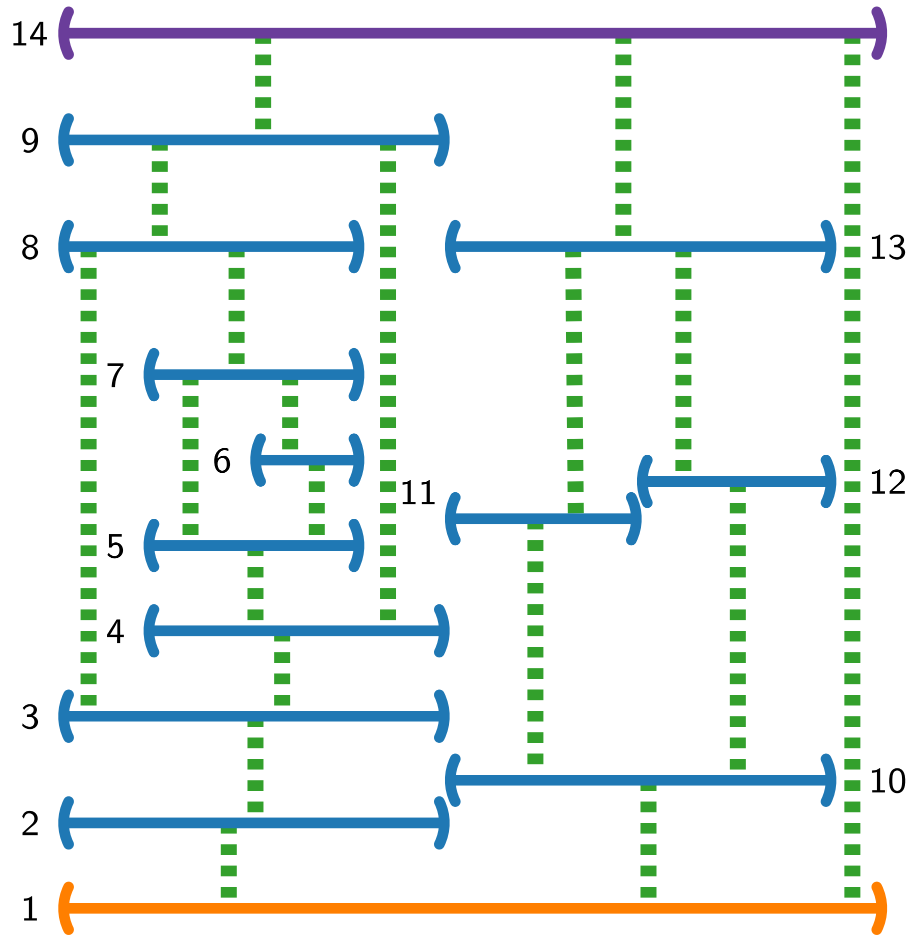


ε-bar Visibility and st-Graphs

Recall that an **st-graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

Observation.
st-orientations correspond to ε-bar visibility representations.



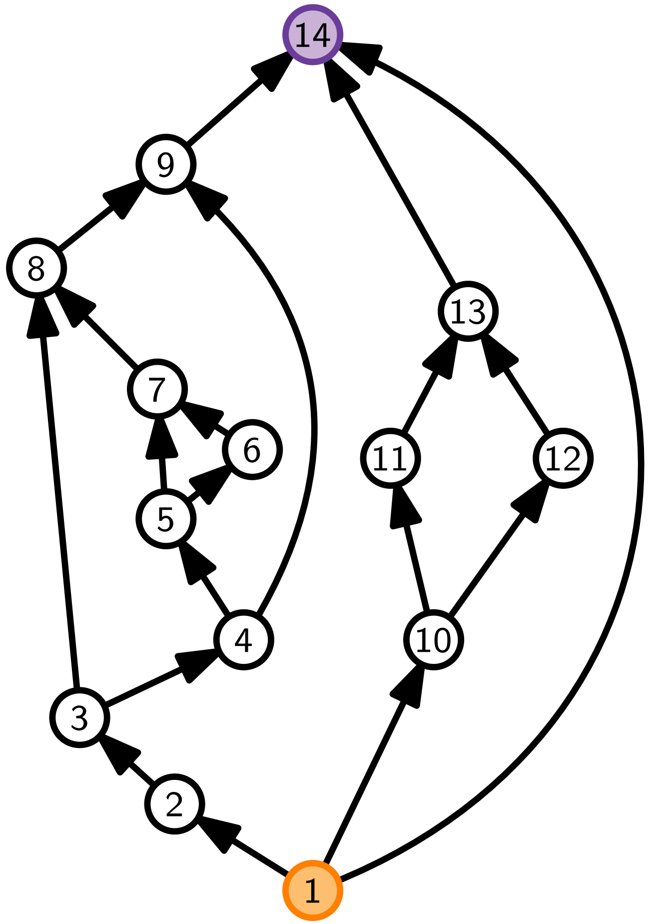


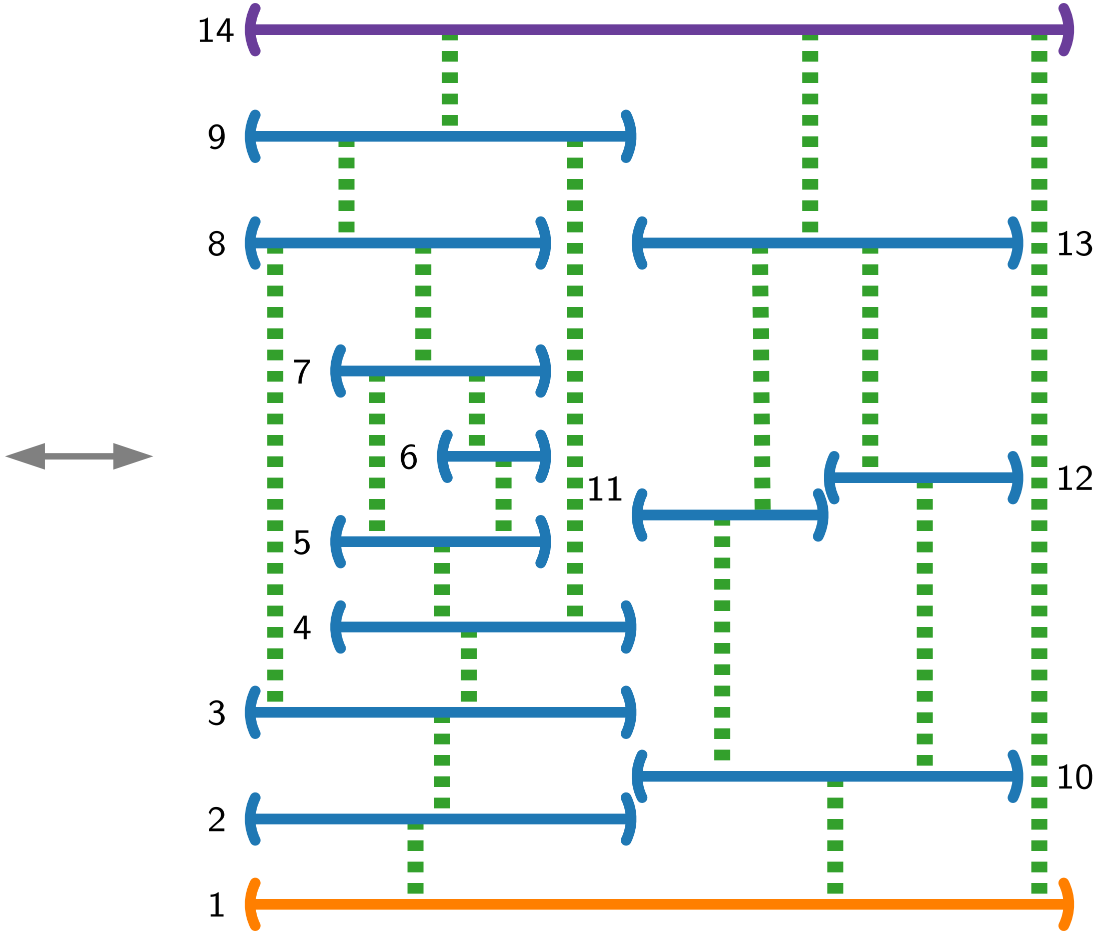
ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

Observation.
st-orientations correspond to ε-bar visibility representations.

Testing whether an acyclic planar **di**graph has a weak bar visibility representation is NP-complete.





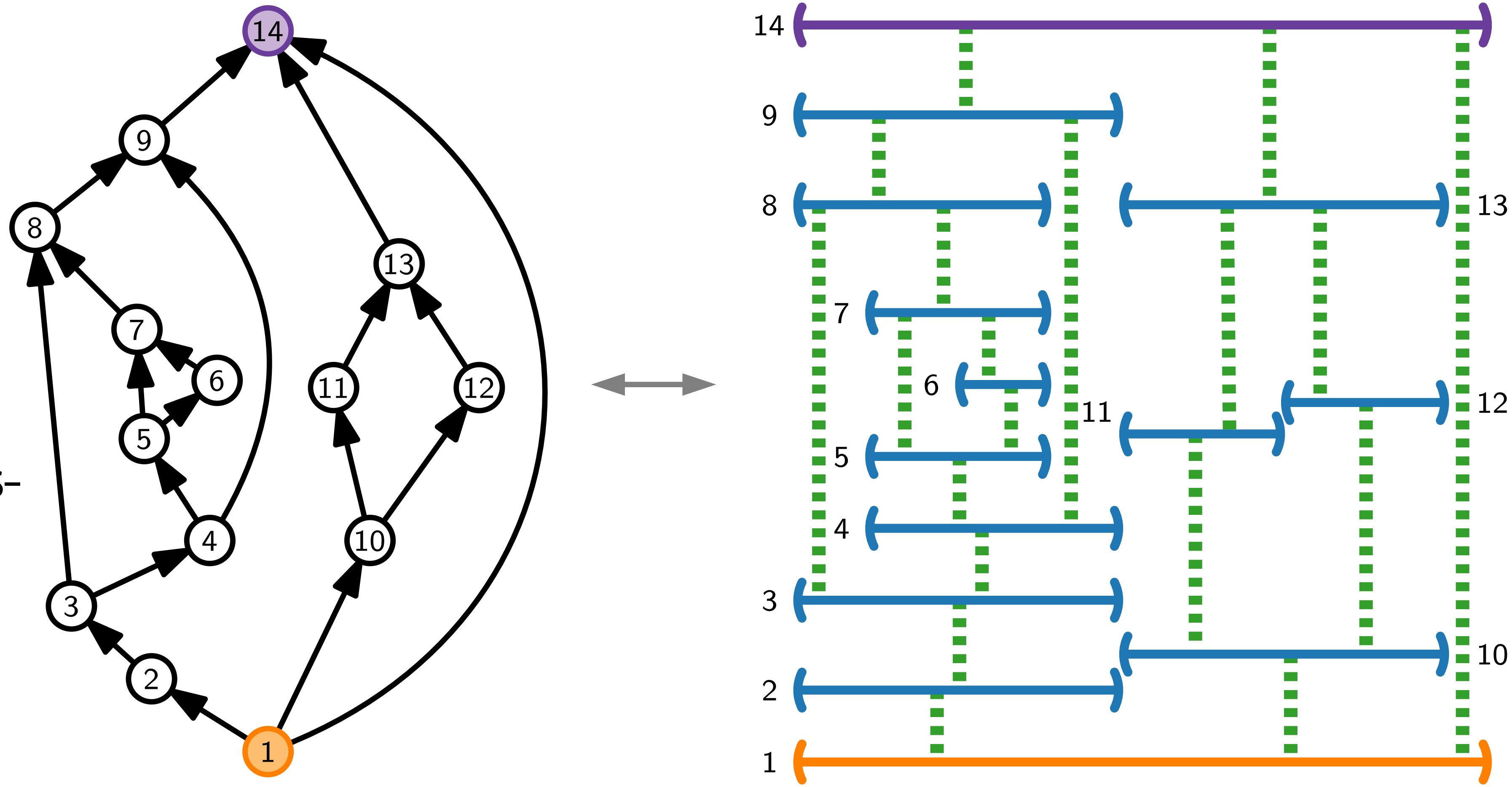
ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

Observation.
st-orientations correspond to ε-bar visibility representations.

Testing whether an acyclic planar **di**graph has a weak bar visibility representation is NP-complete.

- This is upward planarity testing! [Garg & Tamassia '01]

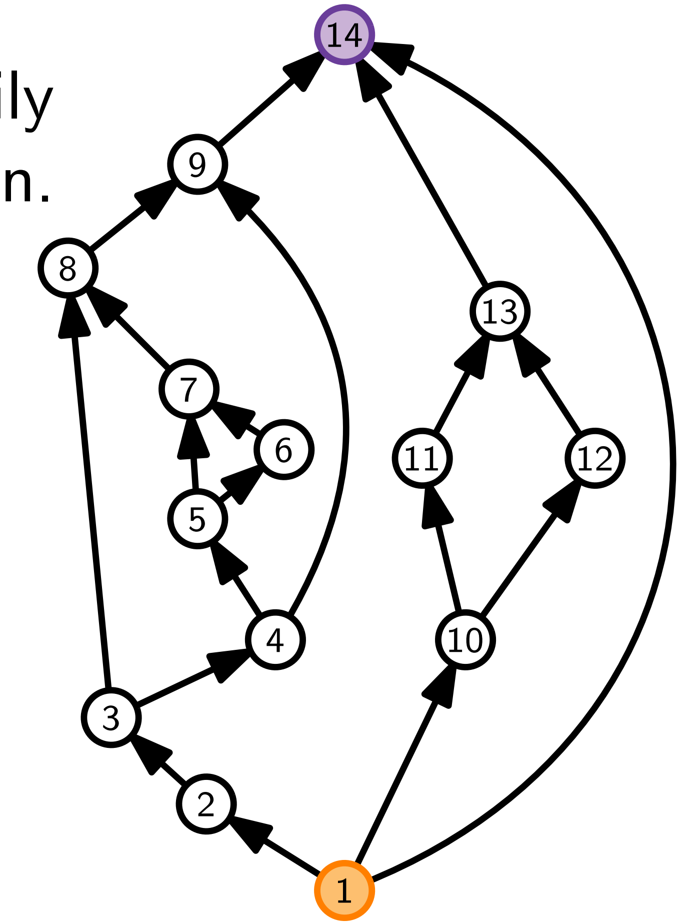


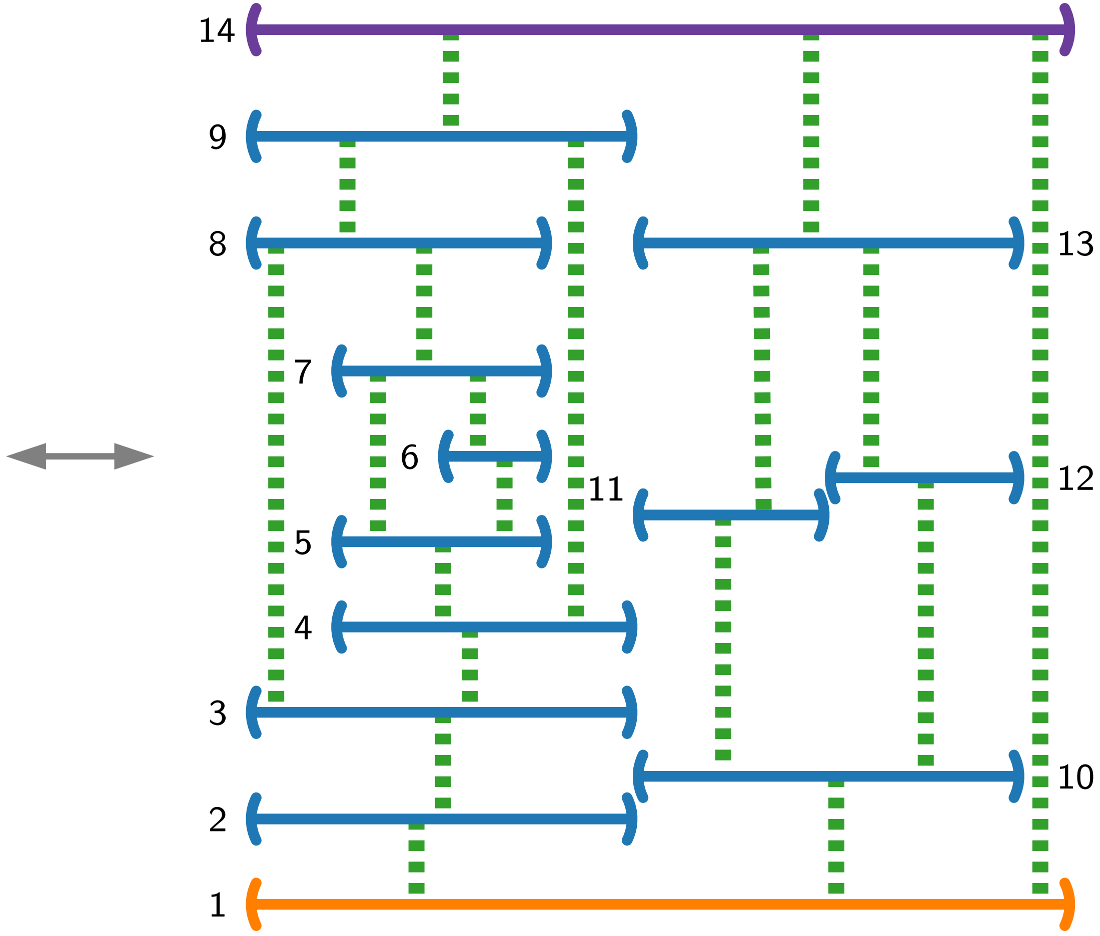
ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

Observation.
st-orientations correspond to ε-bar visibility representations.

- ε-bar visibility testing is easily done via st-graph recognition.



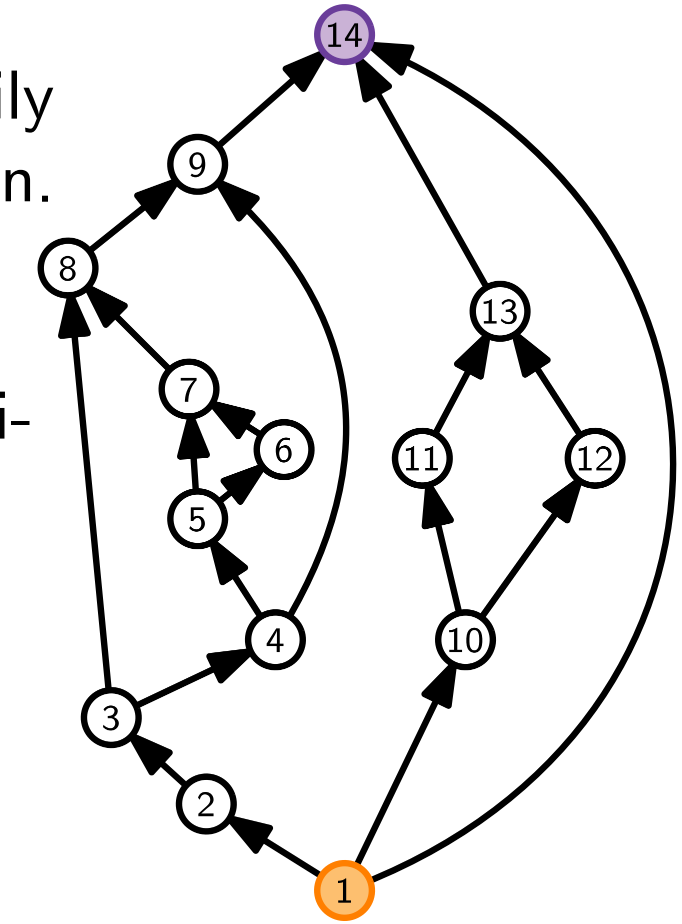


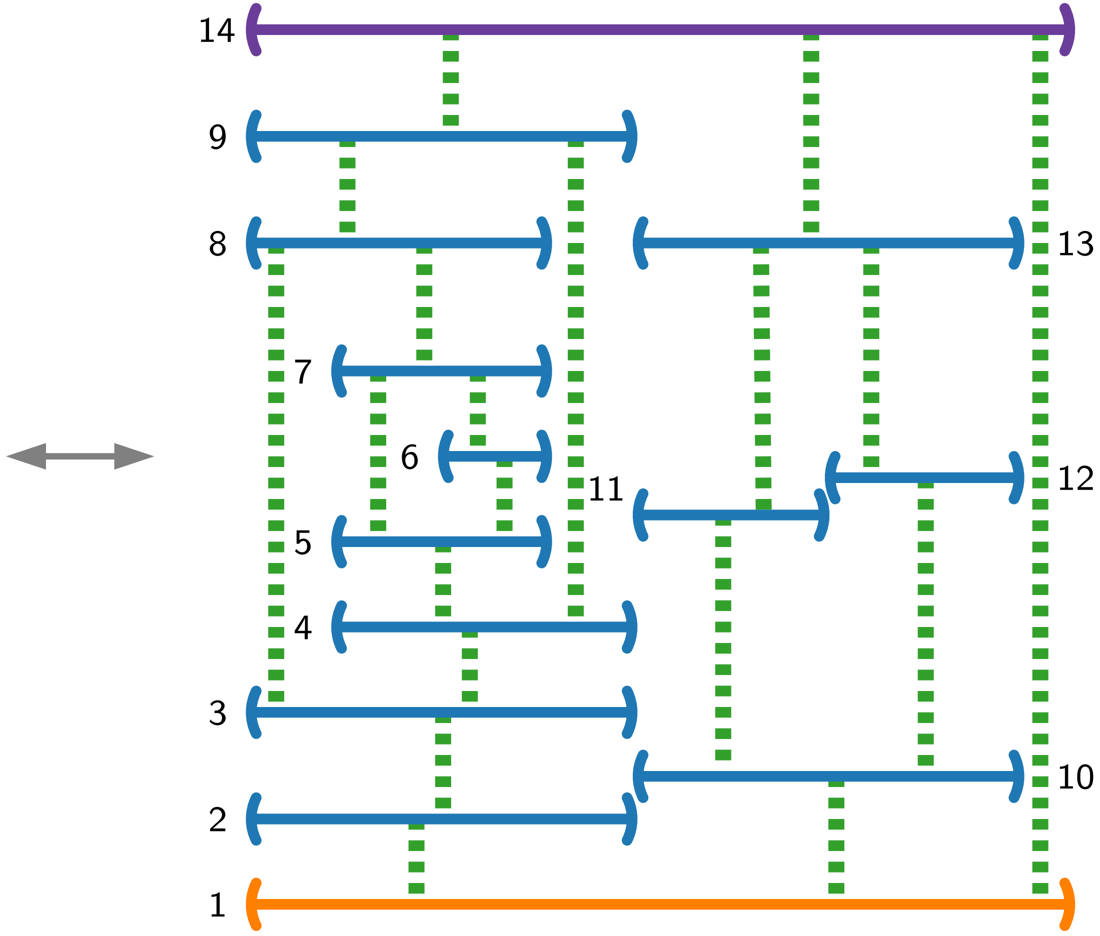
ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

Observation.
st-orientations correspond to ε-bar visibility representations.

- ε-bar visibility testing is easily done via st-graph recognition.
- Strong bar visibility recognition. . . open!



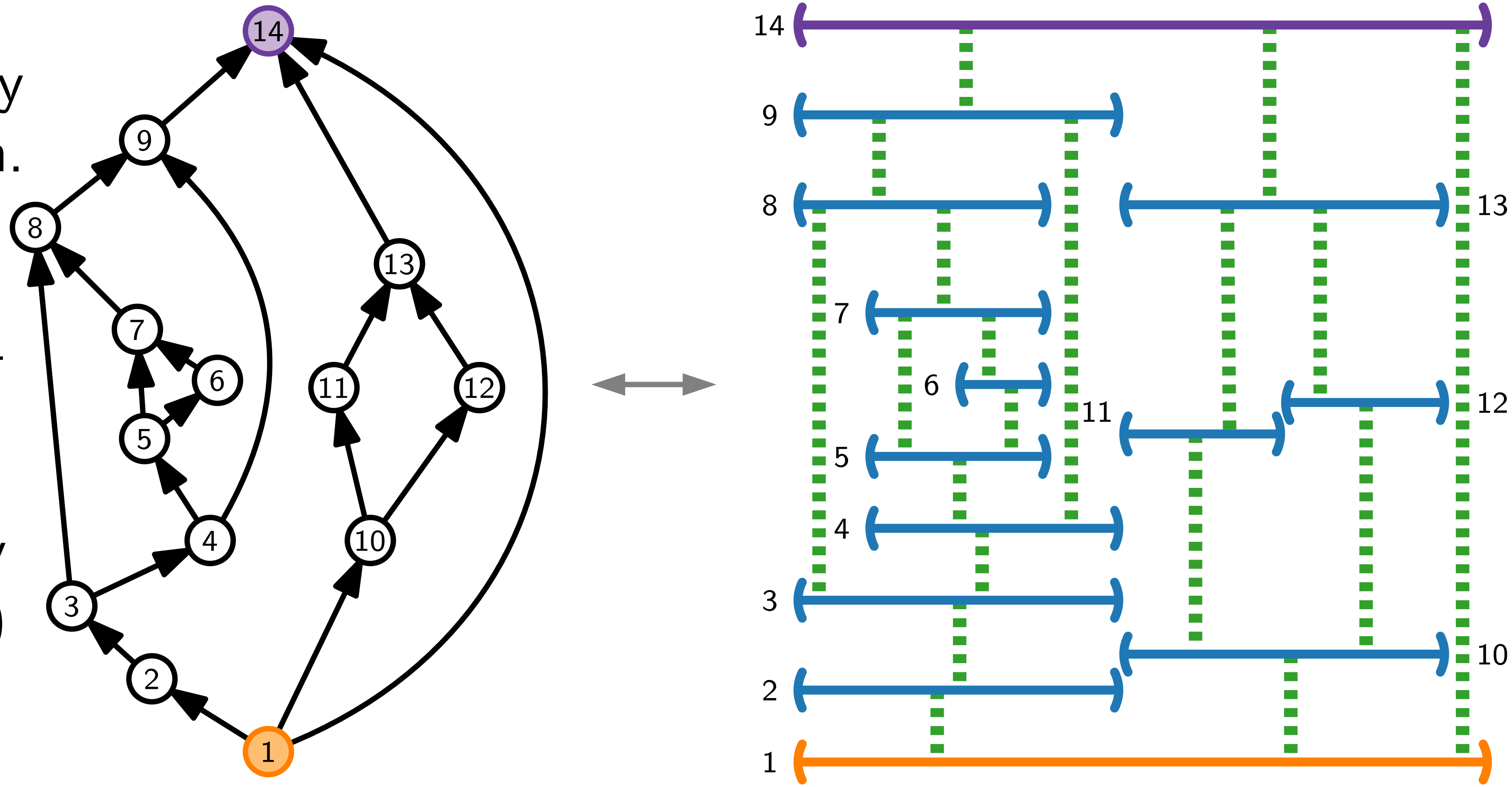


ε-bar Visibility and st-Graphs

Recall that an st-**graph** is a planar digraph G with exactly one source s and one sink t where s and t occur on the outer face of an embedding of G.

Observation.
st-orientations correspond to ε-bar visibility representations.

- ε-bar visibility testing is easily done via st-graph recognition.
- Strong bar visibility recognition. . . open!
- In a **rectangular** bar visibility representation $\psi(s)$ and $\psi(t)$ span an enclosing rectangle.



Results and Outline

Theorem 1. [Chaplick et al. '18]
Rectangular ε-Bar Visibility Representation Extension can be solved in $\mathcal{O}(n \log^2 n)$ time for st-graphs.

Results and Outline

Theorem 1. [Chaplick et al. '18]
Rectangular ε-Bar Visibility Representation Extension can be solved in $\mathcal{O}(n \log^2 n)$ time for st-graphs.

- Dynamic program via SPQR-trees

Results and Outline

Theorem 1. [Chaplick et al. '18]
Rectangular ε-Bar Visibility Representation Extension can be solved in $\mathcal{O}(n \log^2 n)$ time for st-graphs.

- Dynamic program via SPQR-trees
- Easier version: $\mathcal{O}(n^2)$

Results and Outline

Theorem 1. [Chaplick et al. '18]
Rectangular ε-Bar Visibility Representation Extension can be solved in $\mathcal{O}(n \log^2 n)$ time for st-graphs.

- Dynamic program via SPQR-trees
- Easier version: $\mathcal{O}(n^2)$

Theorem 2. [Chaplick et al. '18]
ε-Bar Visibility Representation Extension is NP-complete.

Results and Outline

Theorem 1. [Chaplick et al. '18]
Rectangular ε-Bar Visibility Representation Extension can be solved in $\mathcal{O}(n \log^2 n)$ time for st-graphs.

- Dynamic program via SPQR-trees
- Easier version: $\mathcal{O}(n^2)$

Theorem 2. [Chaplick et al. '18]
ε-Bar Visibility Representation Extension is NP-complete.

- Reduction from Planar Monotone 3-SAT

Results and Outline

Theorem 1. [Chaplick et al. '18]
Rectangular ε-Bar Visibility Representation Extension can be solved in $\mathcal{O}(n \log^2 n)$ time for st-graphs.

- Dynamic program via SPQR-trees
- Easier version: $\mathcal{O}(n^2)$

Theorem 2. [Chaplick et al. '18]
ε-Bar Visibility Representation Extension is NP-complete.

- Reduction from Planar Monotone 3-SAT

Theorem 3. [Chaplick et al. '18]
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).

Results and Outline

Theorem 1. [Chaplick et al. '18]
Rectangular ε-Bar Visibility Representation Extension can be solved in $\mathcal{O}(n \log^2 n)$ time for st-graphs.

- Dynamic program via SPQR-trees
- Easier version: $\mathcal{O}(n^2)$

Theorem 2. [Chaplick et al. '18]
ε-Bar Visibility Representation Extension is NP-complete.

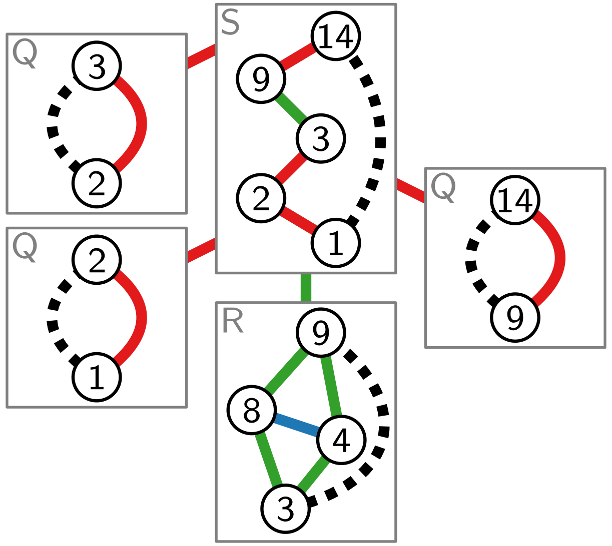
- Reduction from Planar Monotone 3-SAT

Theorem 3. [Chaplick et al. '18]
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).

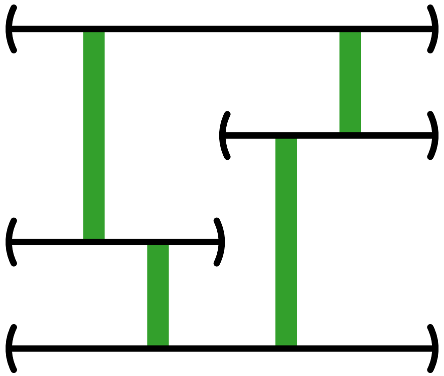
- Reduction from 3-Partition

Visualization of Graphs

Lecture 9:
Partial Visibility Representation Extension



Part III:
SPQR-Trees



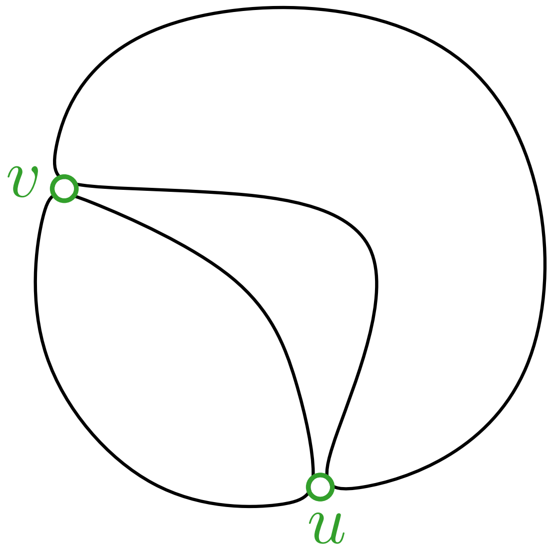
Alexander Wolff

SPQR-Tree

- An **SPQR-tree** T is a decomposition of a planar graph G by **separation pairs**.

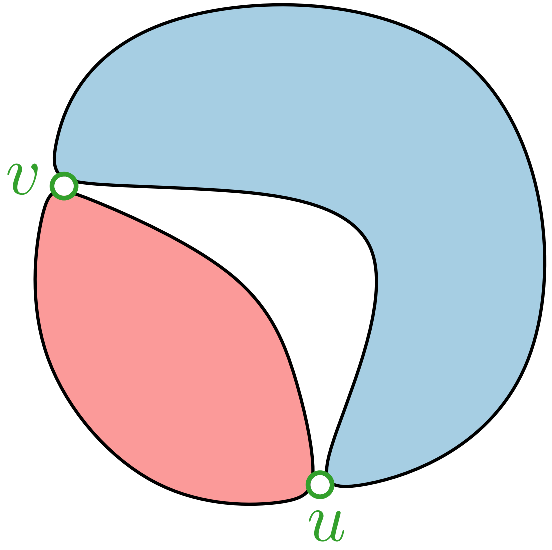
SPQR-Tree

- An **SPQR-tree** T is a decomposition of a planar graph G by **separation pairs**.



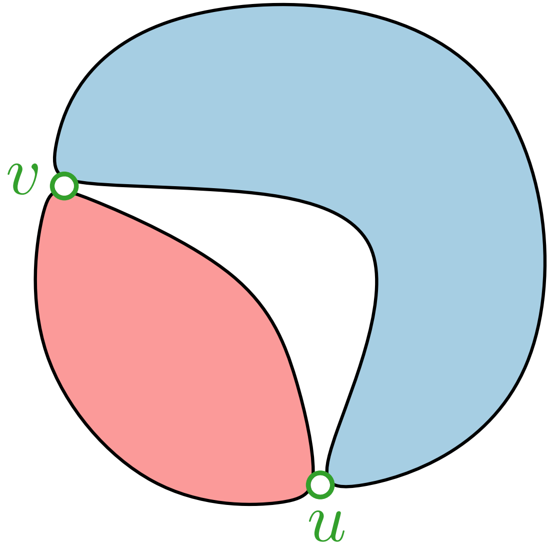
SPQR-Tree

- An **SPQR-tree** T is a decomposition of a planar graph G by **separation pairs**.



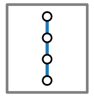
SPQR-Tree

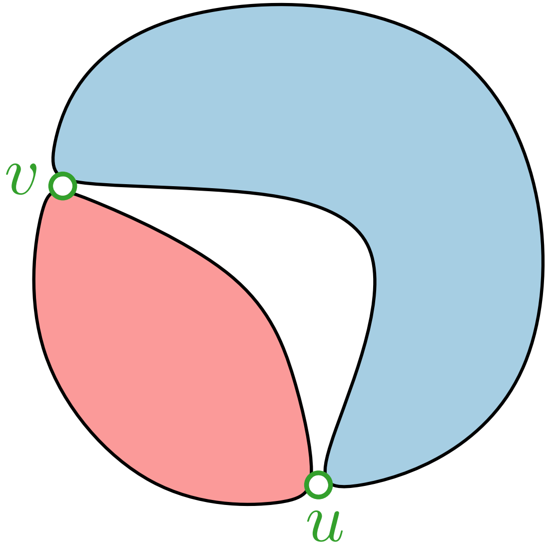
- An **SPQR-tree** T is a decomposition of a planar graph G by **separation pairs**.
- The nodes of T are of four types:



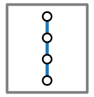
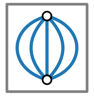
SPQR-Tree

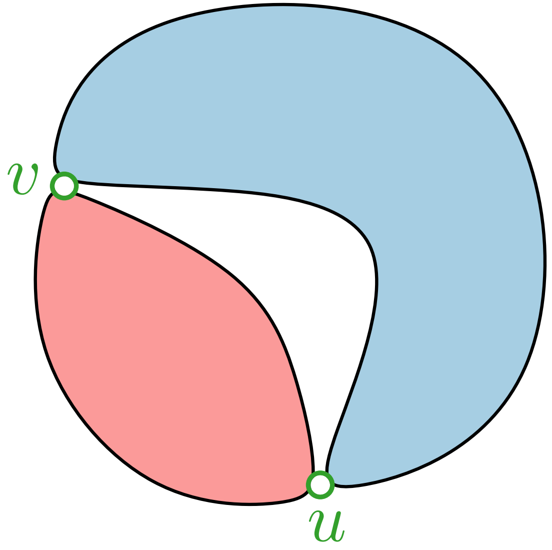
- An **SPQR-tree** T is a decomposition of a planar graph G by **separation pairs**.
- The nodes of T are of four types:
  - **S**-nodes represent a series composition



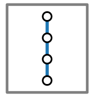
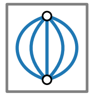
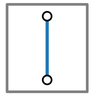


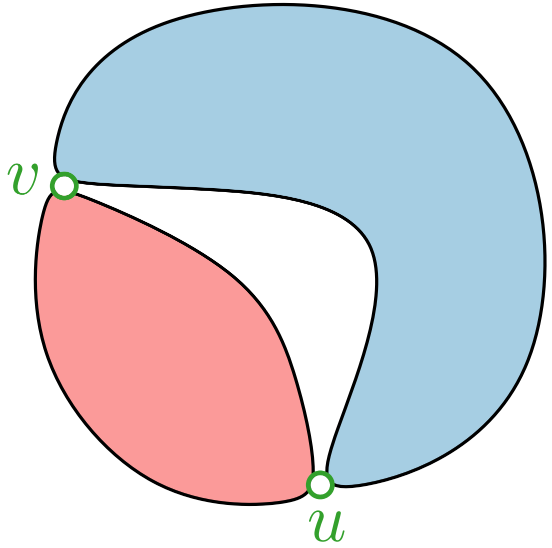
SPQR-Tree

- An **SPQR-tree** T is a decomposition of a planar graph G by **separation pairs**.
- The nodes of T are of four types:
  - **S**-nodes represent a series composition 
  - **P**-nodes represent a parallel composition 



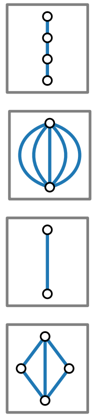
SPQR-Tree

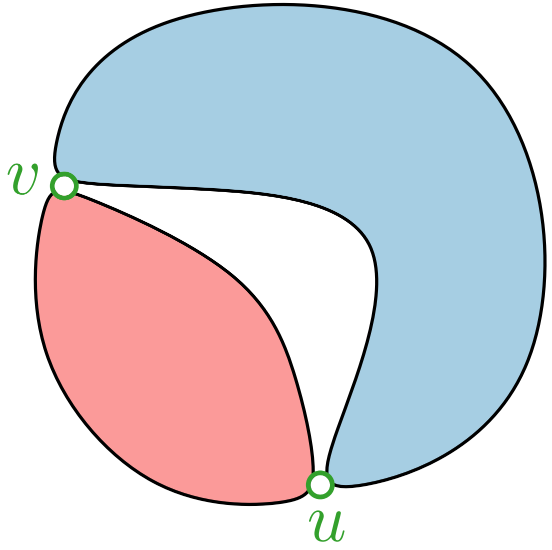
- An **SPQR-tree** T is a decomposition of a planar graph G by **separation pairs**.
- The nodes of T are of four types:
  - **S**-nodes represent a series composition 
  - **P**-nodes represent a parallel composition 
  - **Q**-nodes represent a single edge 



SPQR-Tree

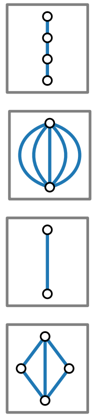
- An **SPQR-tree** T is a decomposition of a planar graph G by **separation pairs**.
- The nodes of T are of four types:
  - **S**-nodes represent a series composition
  - **P**-nodes represent a parallel composition
  - **Q**-nodes represent a single edge
  - **R**-nodes represent 3-connected (*rigid*) subgraphs

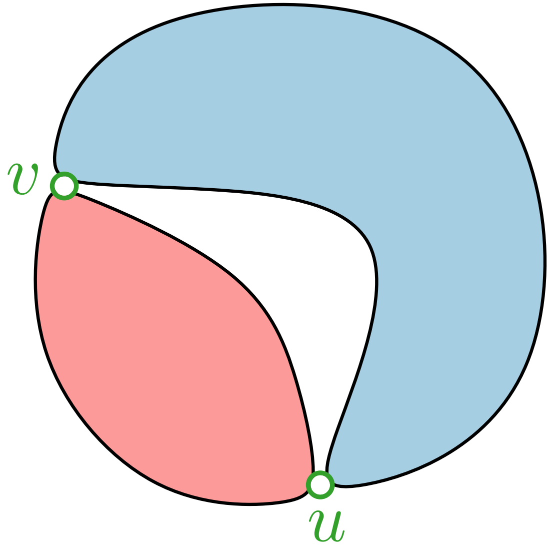




SPQR-Tree

- An **SPQR-tree** T is a decomposition of a planar graph G by **separation pairs**.
- The nodes of T are of four types:
  - **S**-nodes represent a series composition
  - **P**-nodes represent a parallel composition
  - **Q**-nodes represent a single edge
  - **R**-nodes represent 3-connected (*rigid*) subgraphs

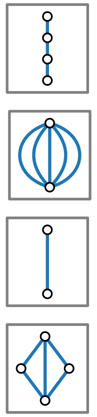


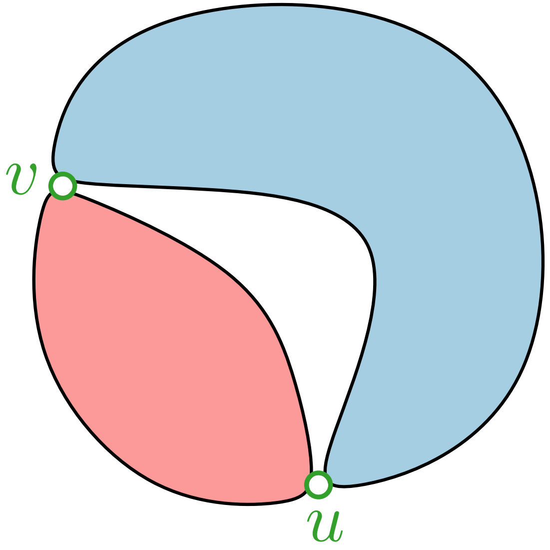


- A decomposition tree of a series-parallel graph is an SPQR-tree without **R**-nodes.

SPQR-Tree

- An **SPQR-tree** T is a decomposition of a planar graph G by **separation pairs**.
- The nodes of T are of four types:
  - **S**-nodes represent a series composition
  - **P**-nodes represent a parallel composition
  - **Q**-nodes represent a single edge
  - **R**-nodes represent 3-connected (*rigid*) subgraphs

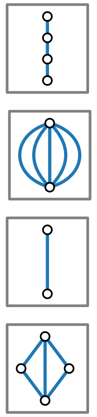




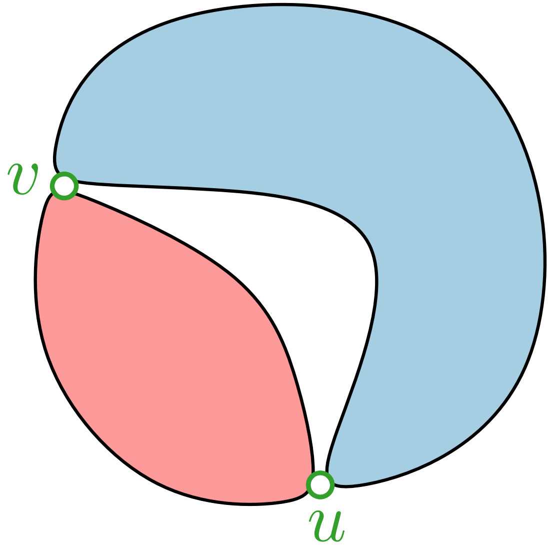
- A decomposition tree of a series-parallel graph is an SPQR-tree without **R**-nodes.
- T represents all planar embeddings of G.

SPQR-Tree

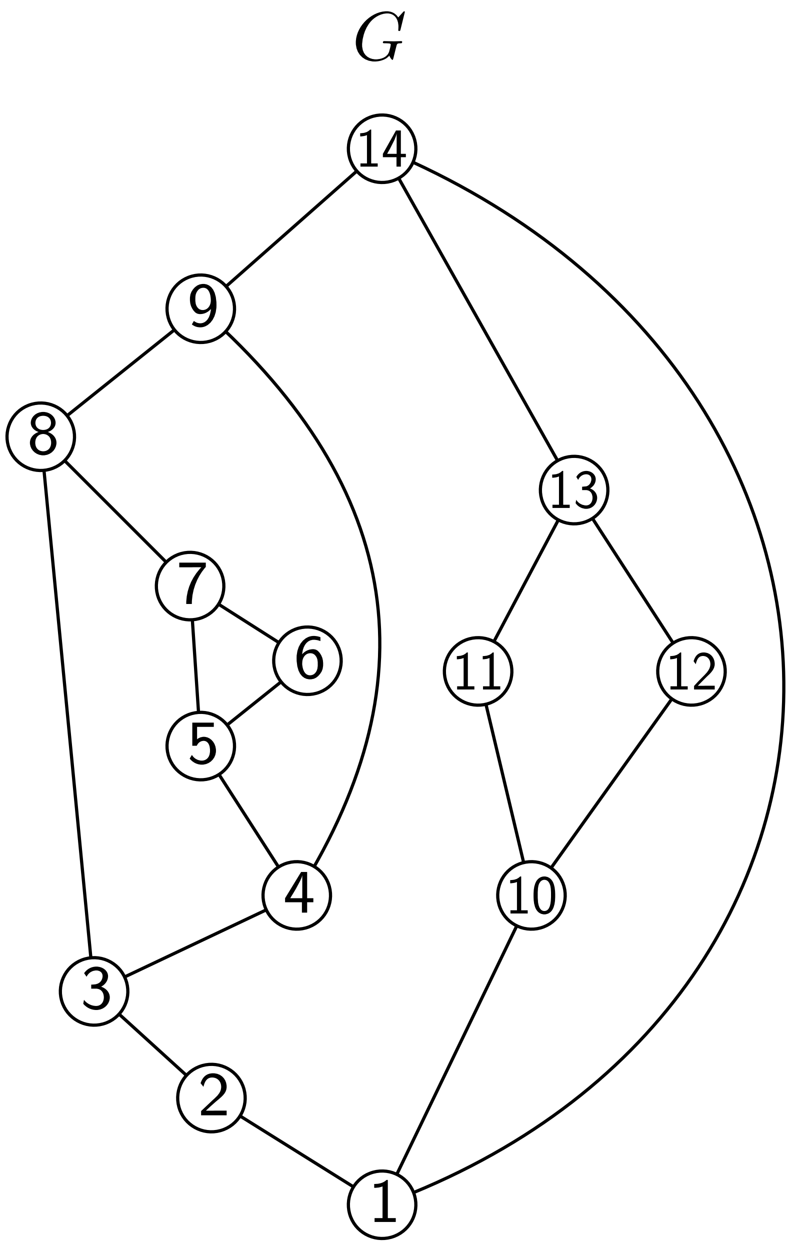
- An **SPQR-tree** T is a decomposition of a planar graph G by **separation pairs**.
- The nodes of T are of four types:
  - **S**-nodes represent a series composition
  - **P**-nodes represent a parallel composition
  - **Q**-nodes represent a single edge
  - **R**-nodes represent 3-connected (*rigid*) subgraphs



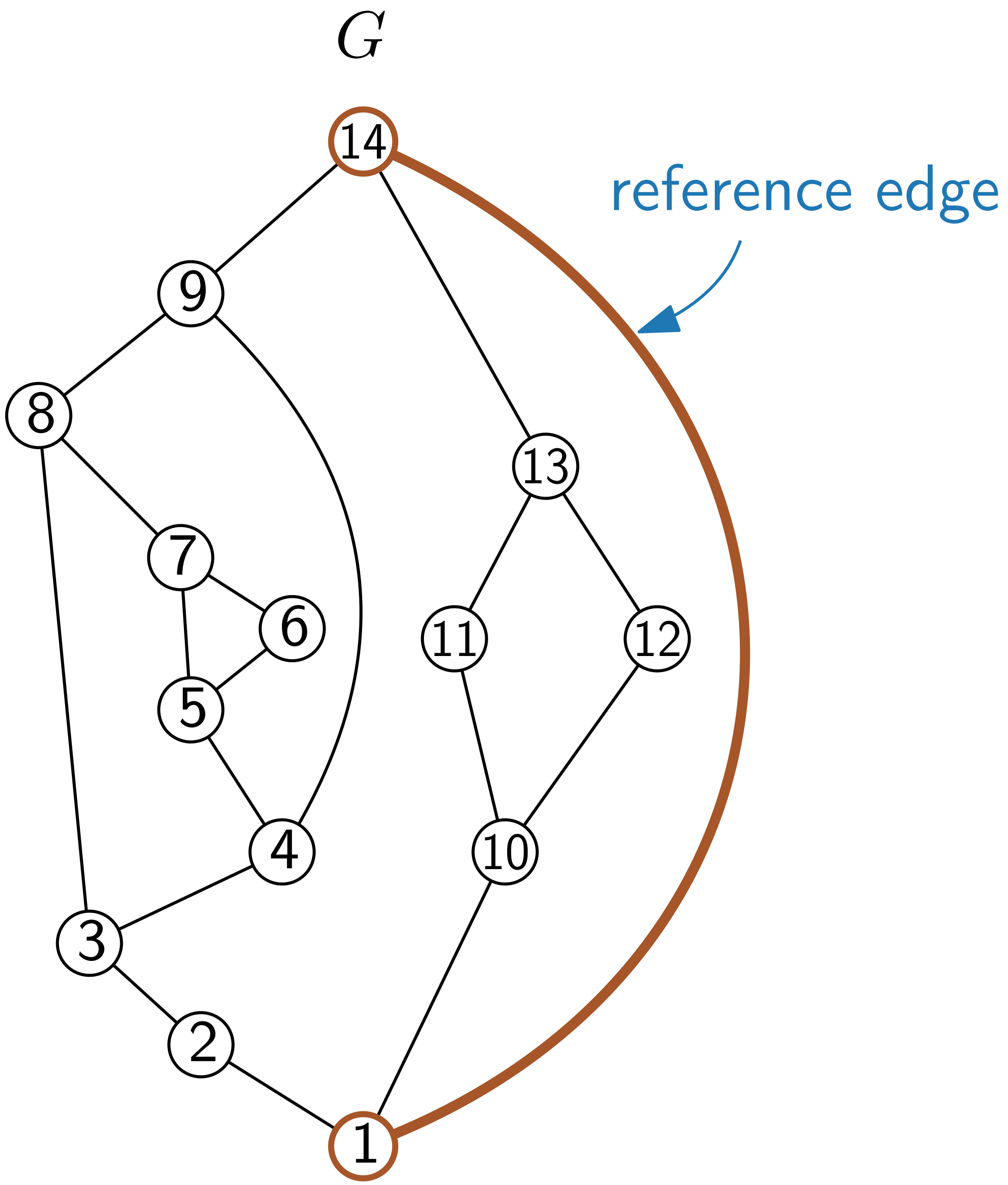
- A decomposition tree of a series-parallel graph is an SPQR-tree without **R**-nodes.
- T represents all planar embeddings of G.
- T can be computed in $\mathcal{O}(n)$ time. [Gutwenger, Mutzel '01]

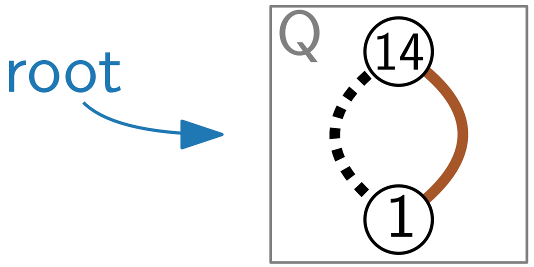


SPQR-Tree Example

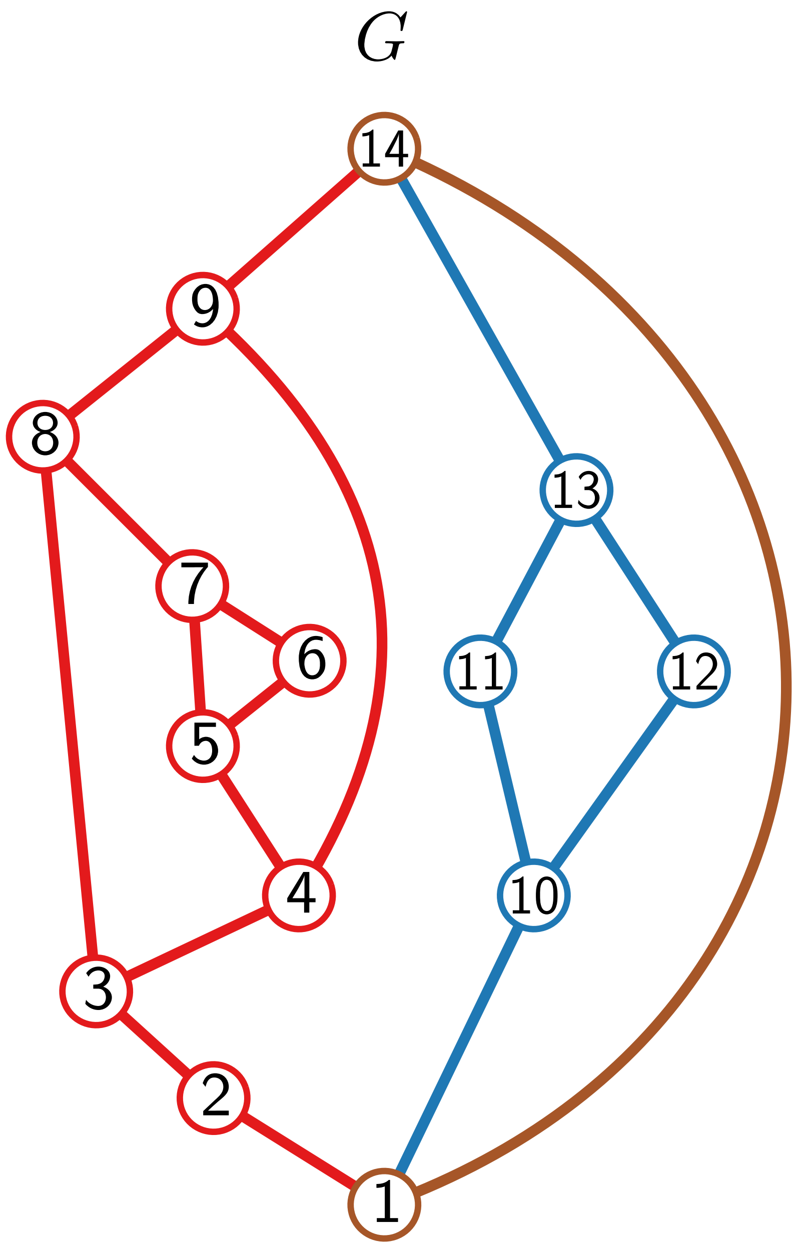


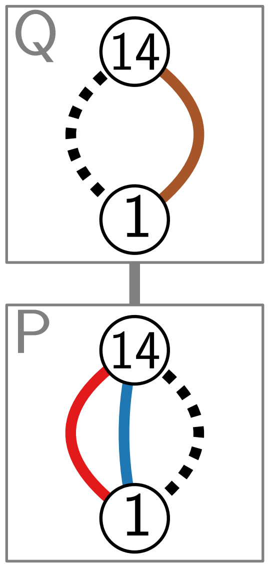
SPQR-Tree Example



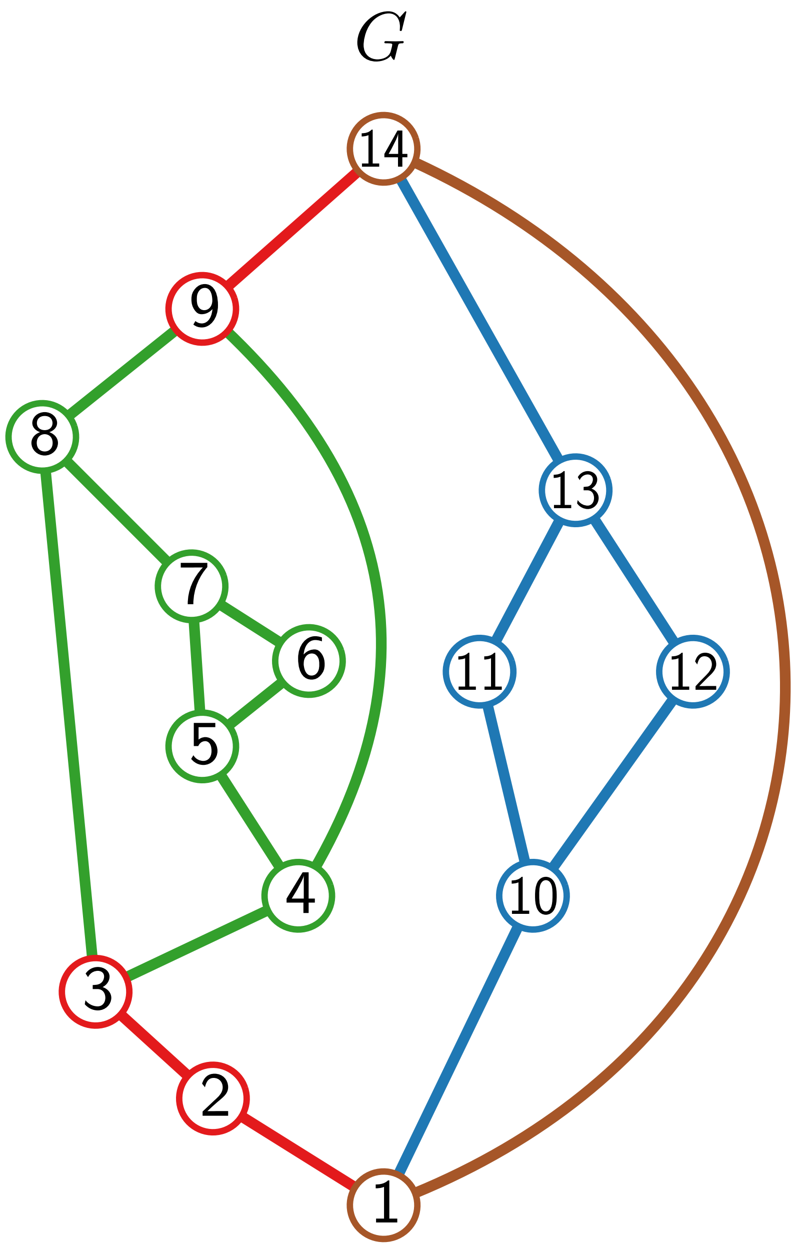


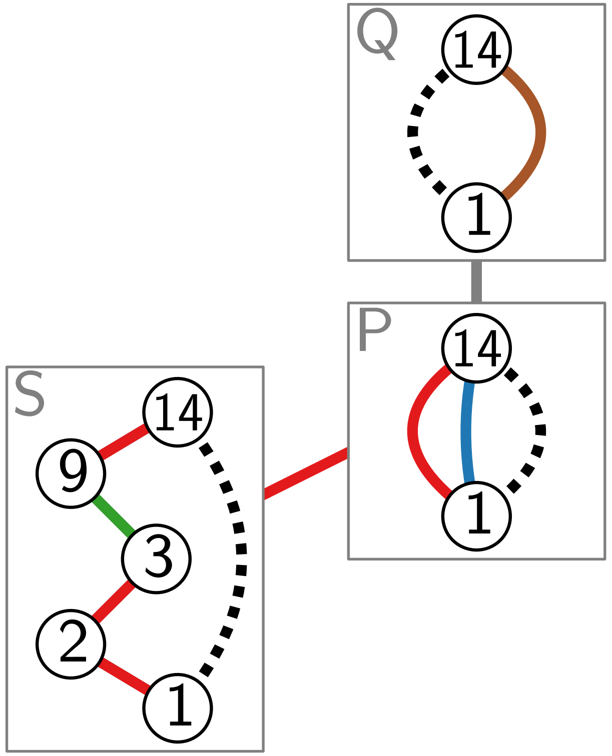
SPQR-Tree Example



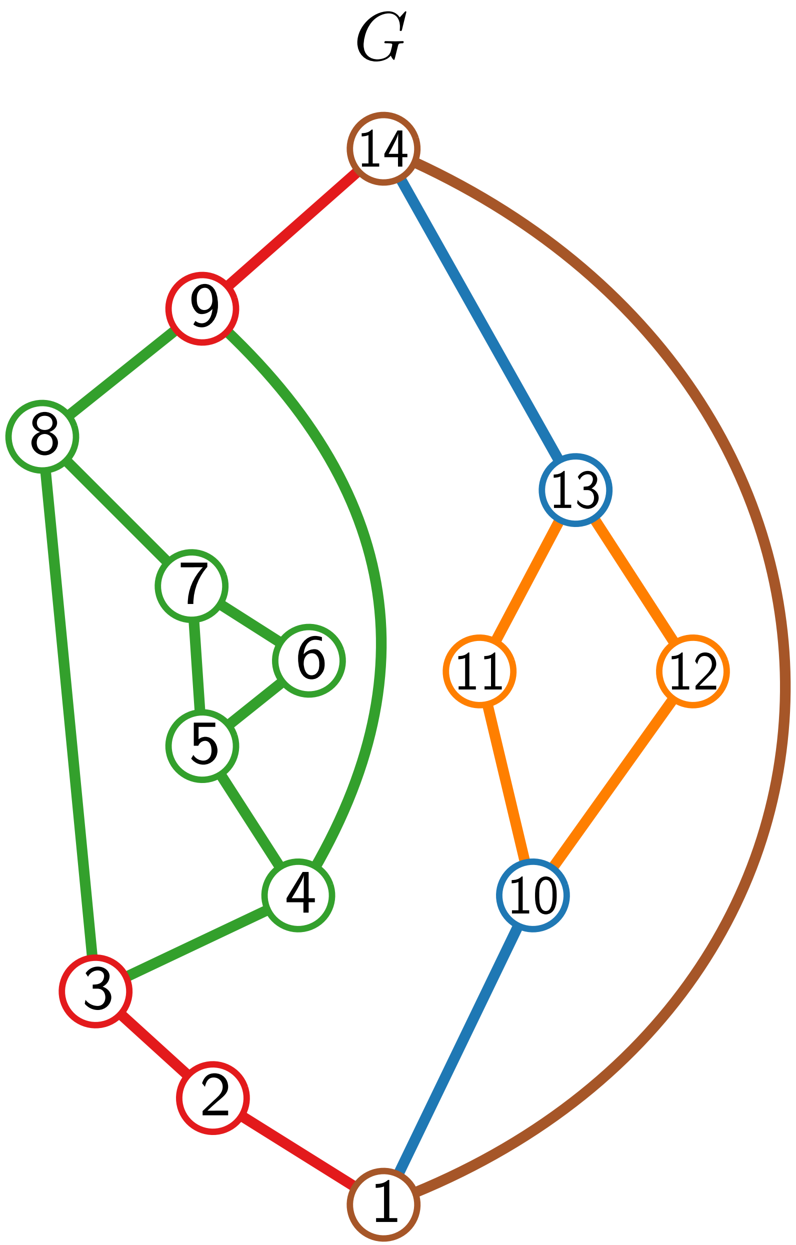


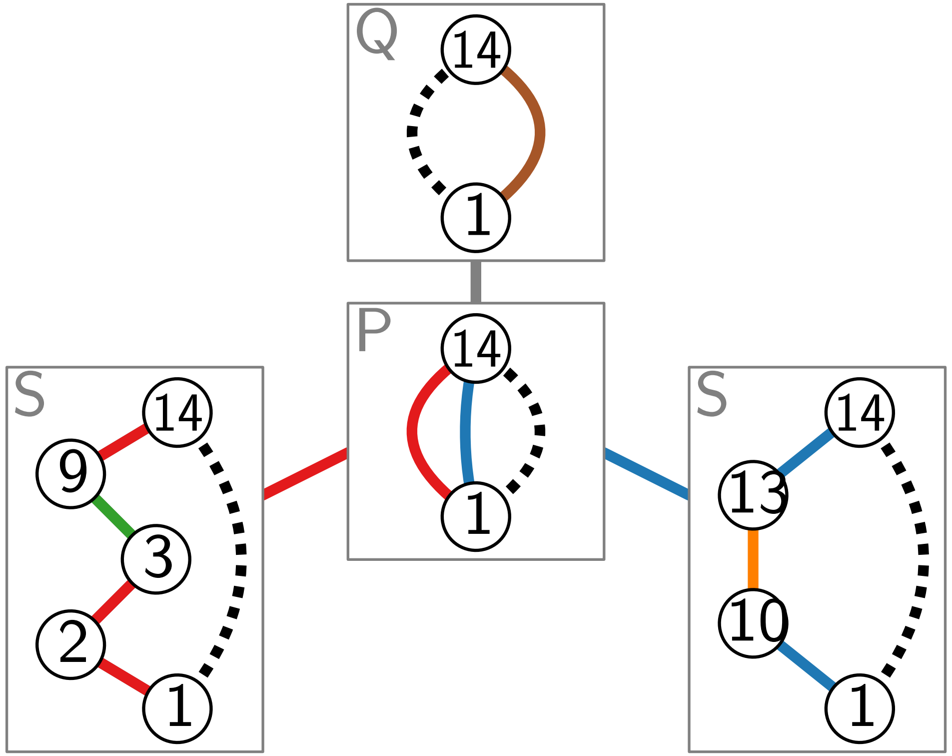
SPQR-Tree Example



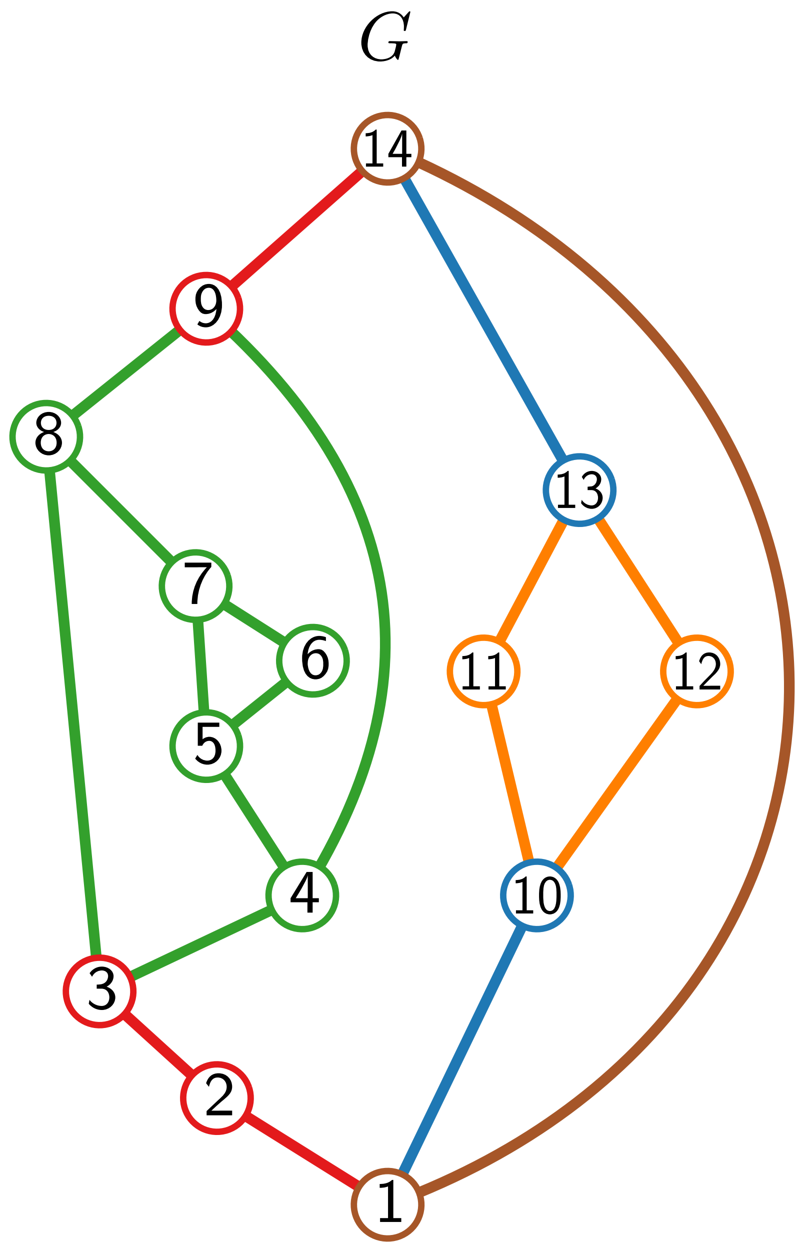


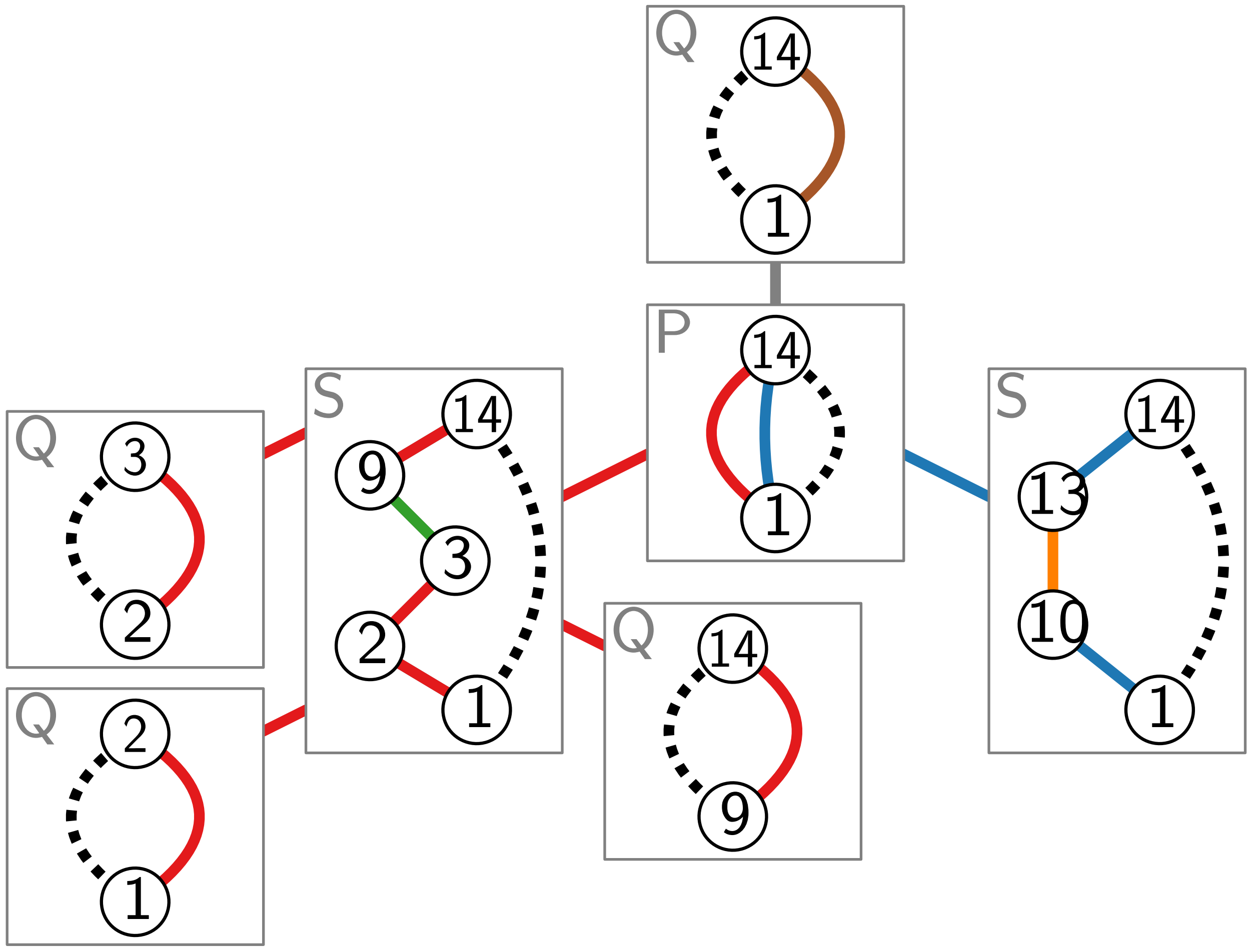
SPQR-Tree Example



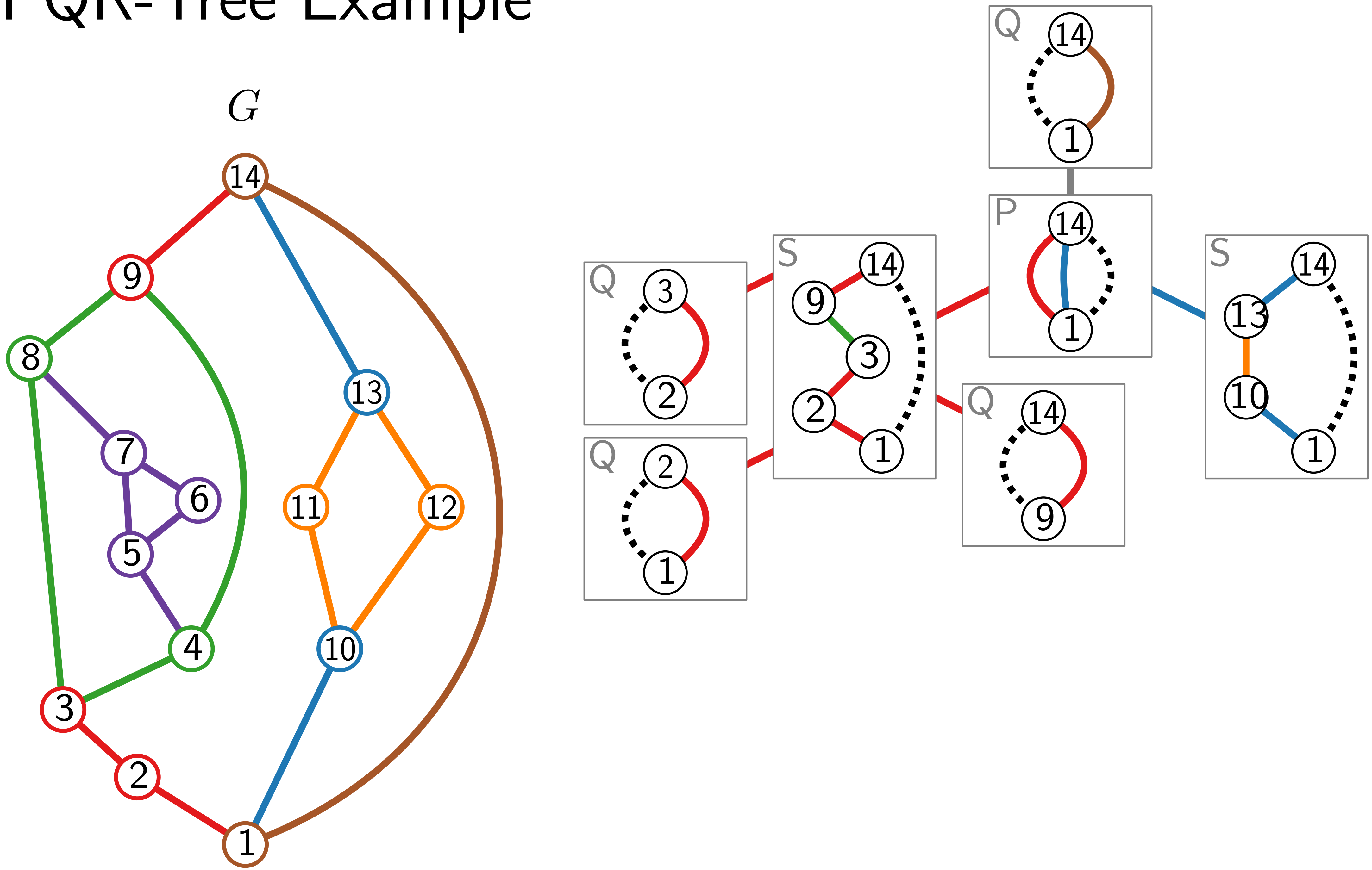


SPQR-Tree Example

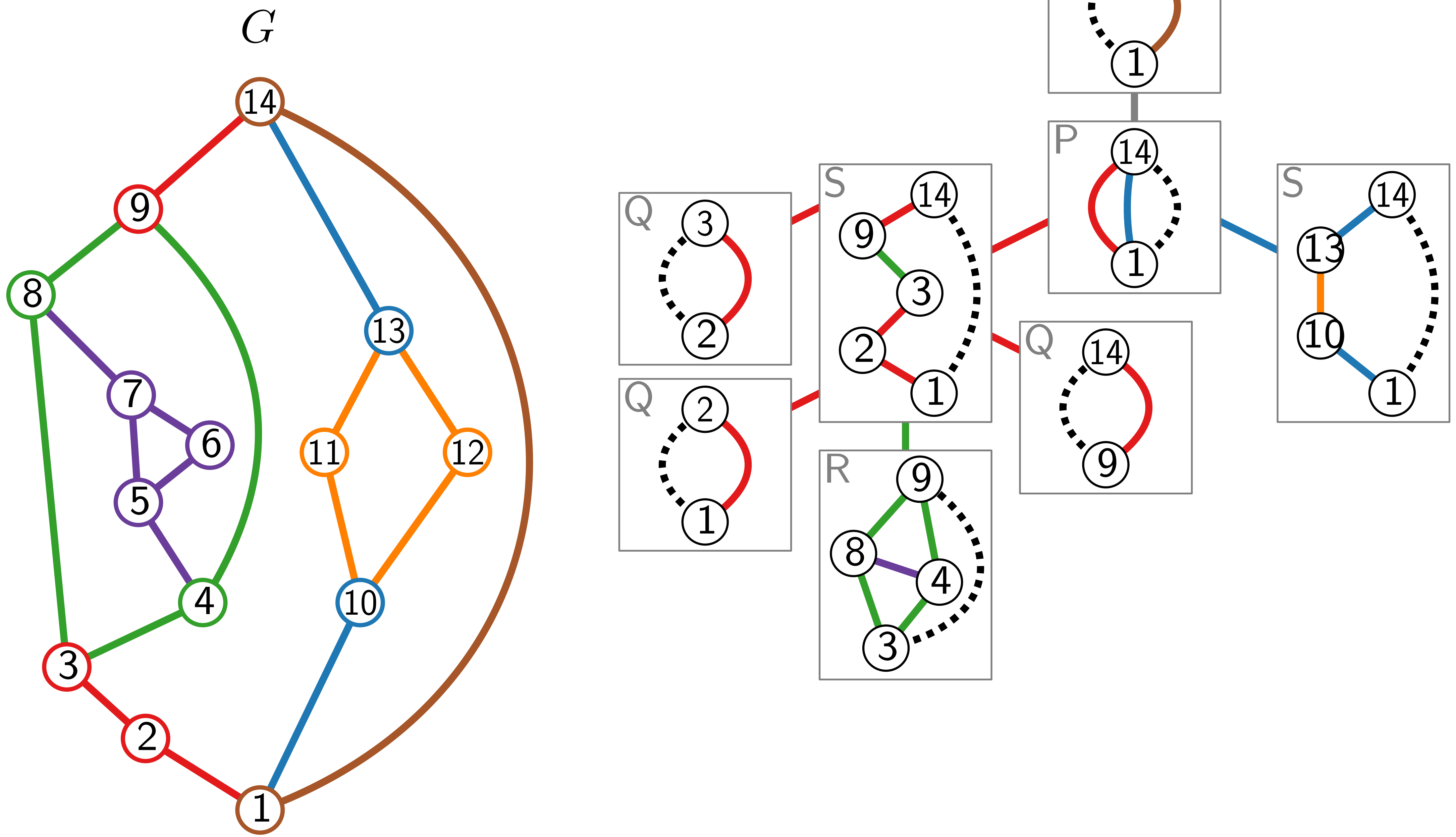




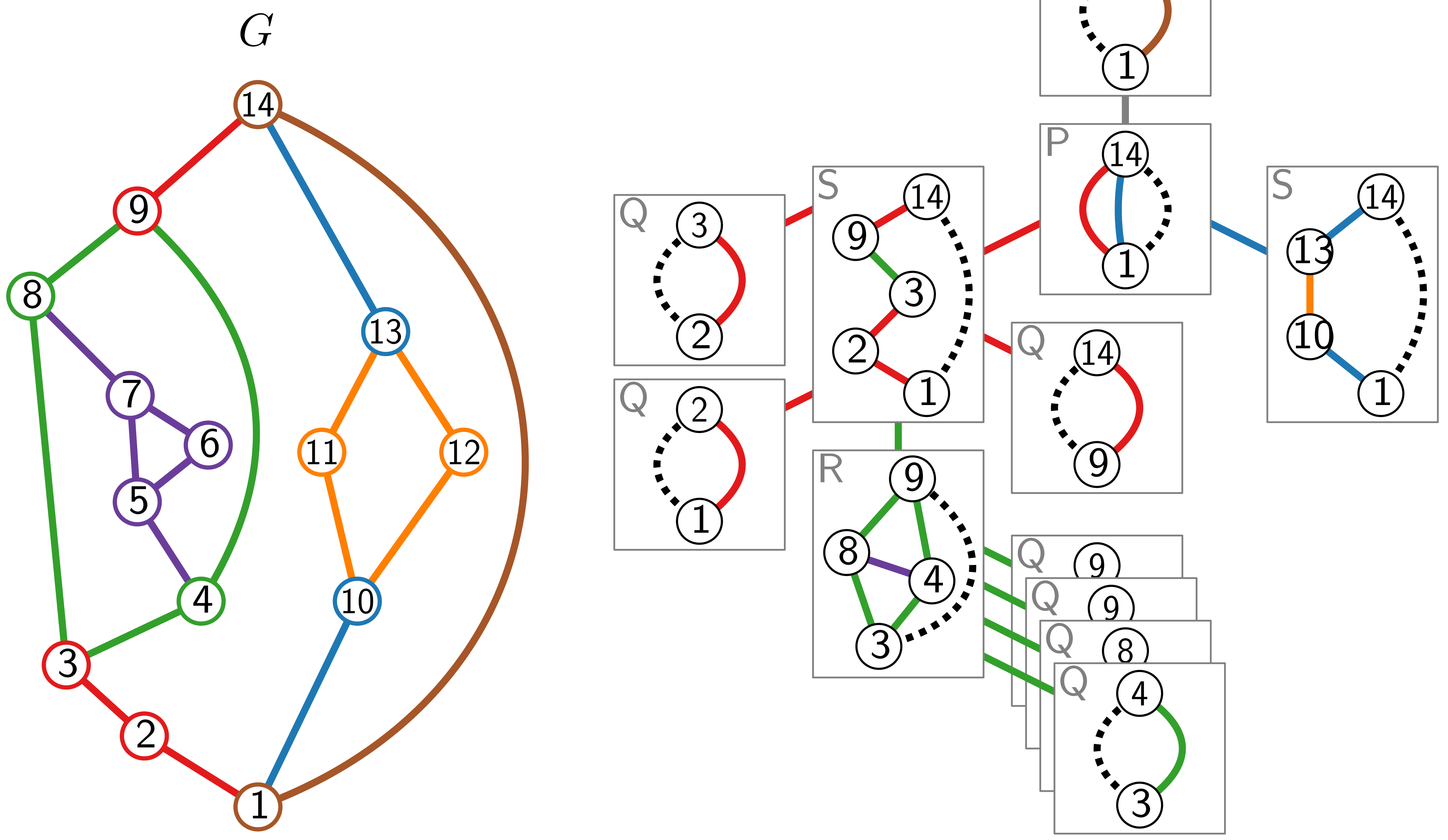
SPQR-Tree Example



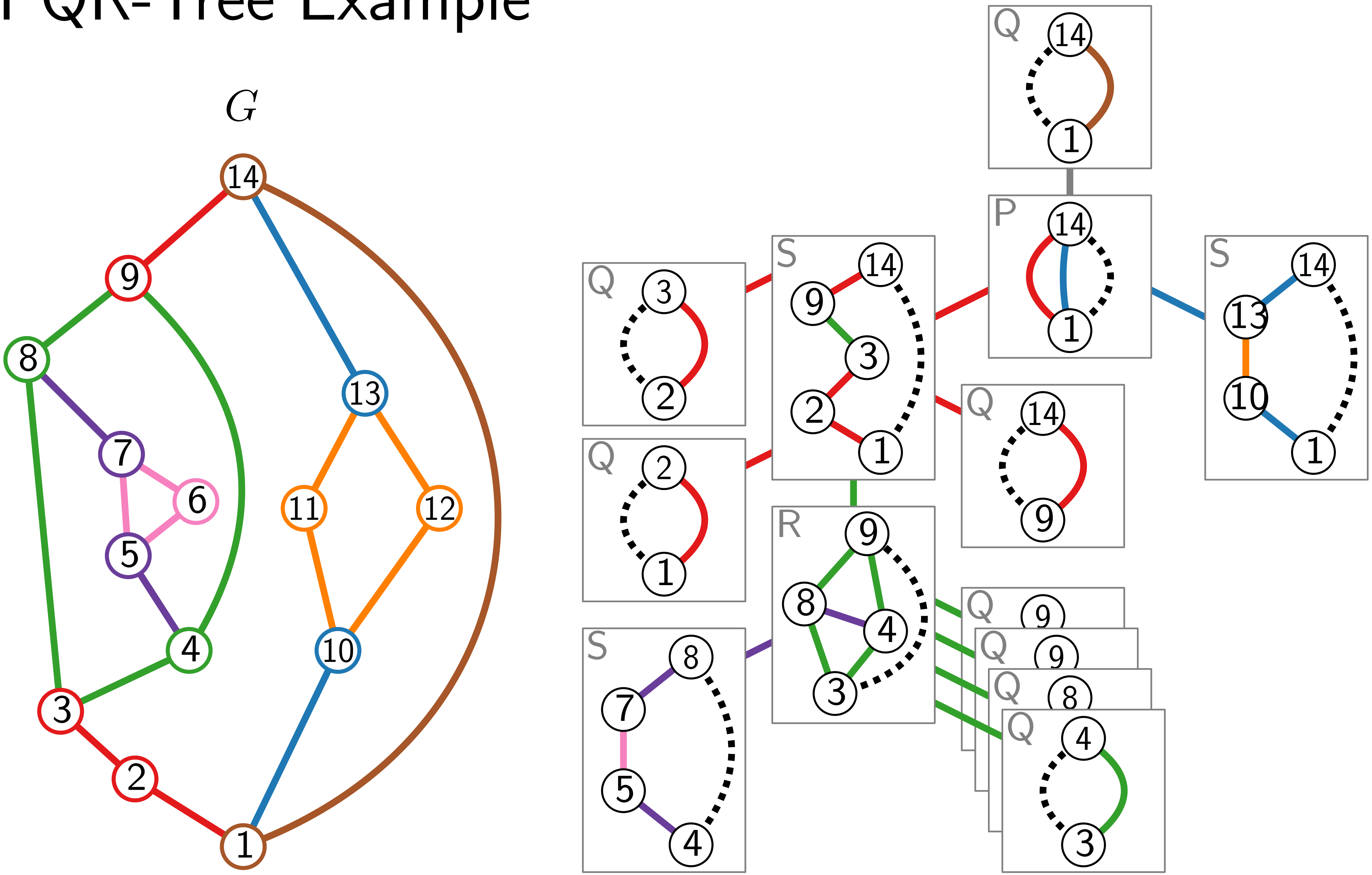
SPQR-Tree Example



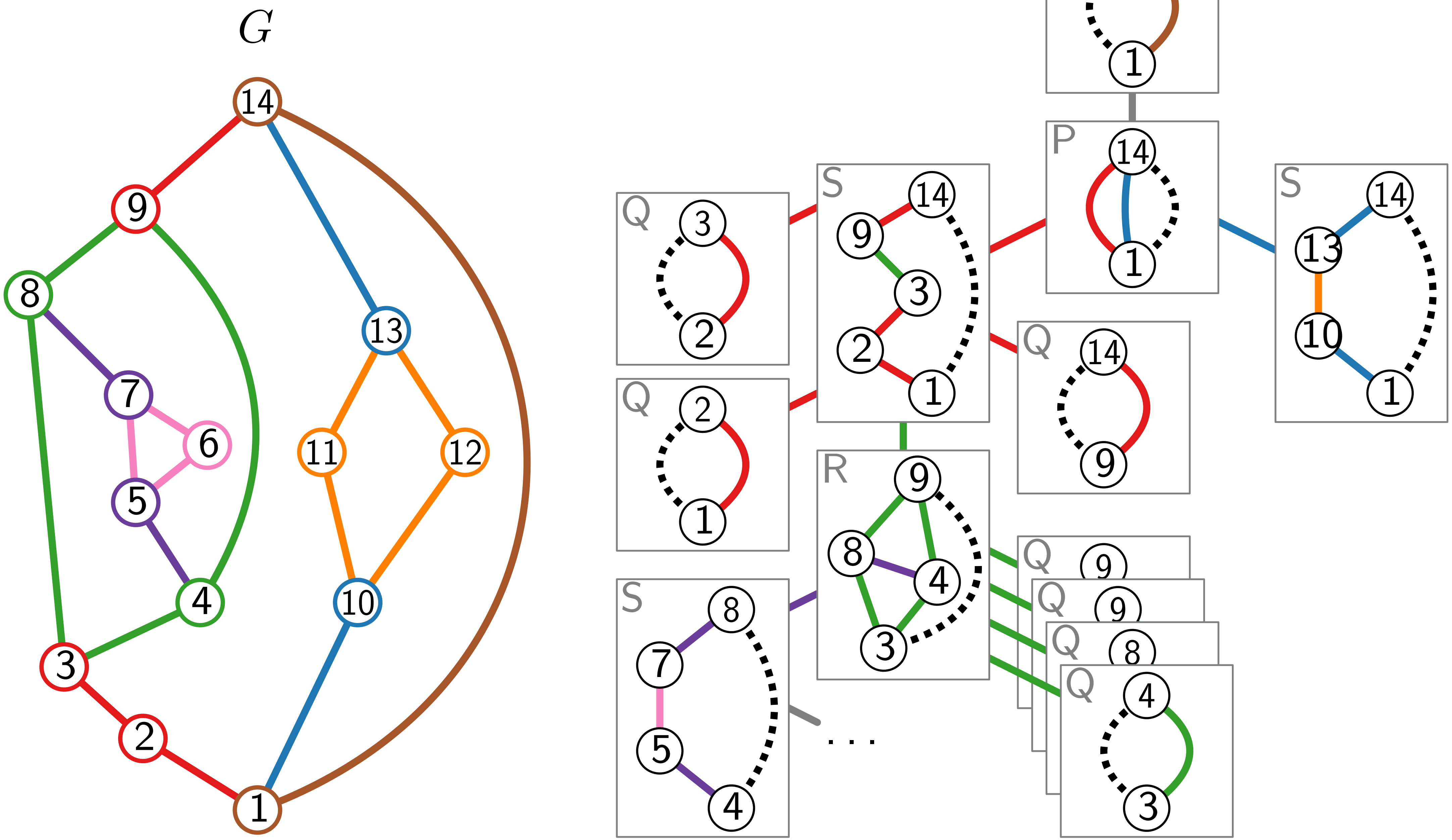
SPQR-Tree Example



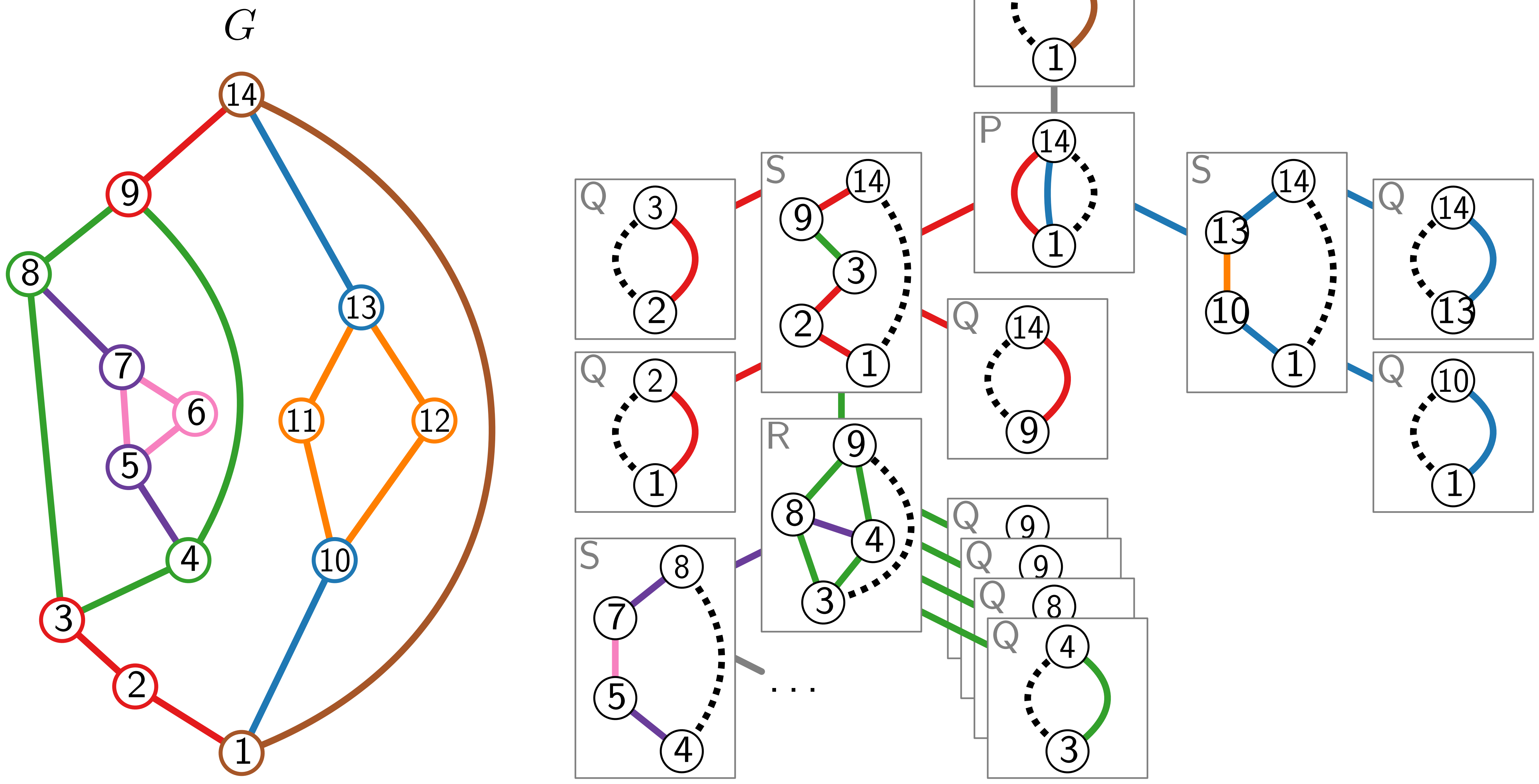
SPQR-Tree Example



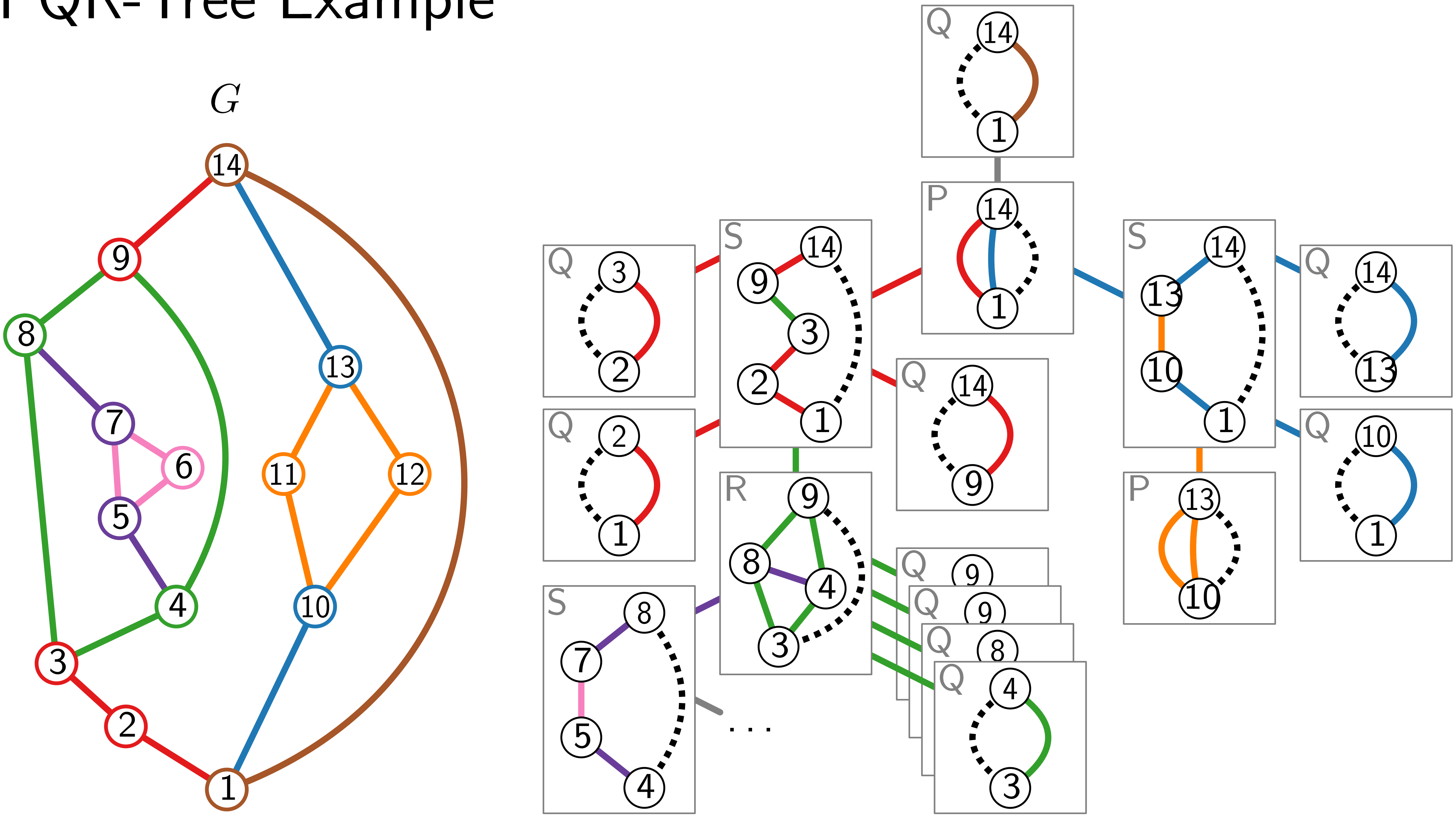
SPQR-Tree Example



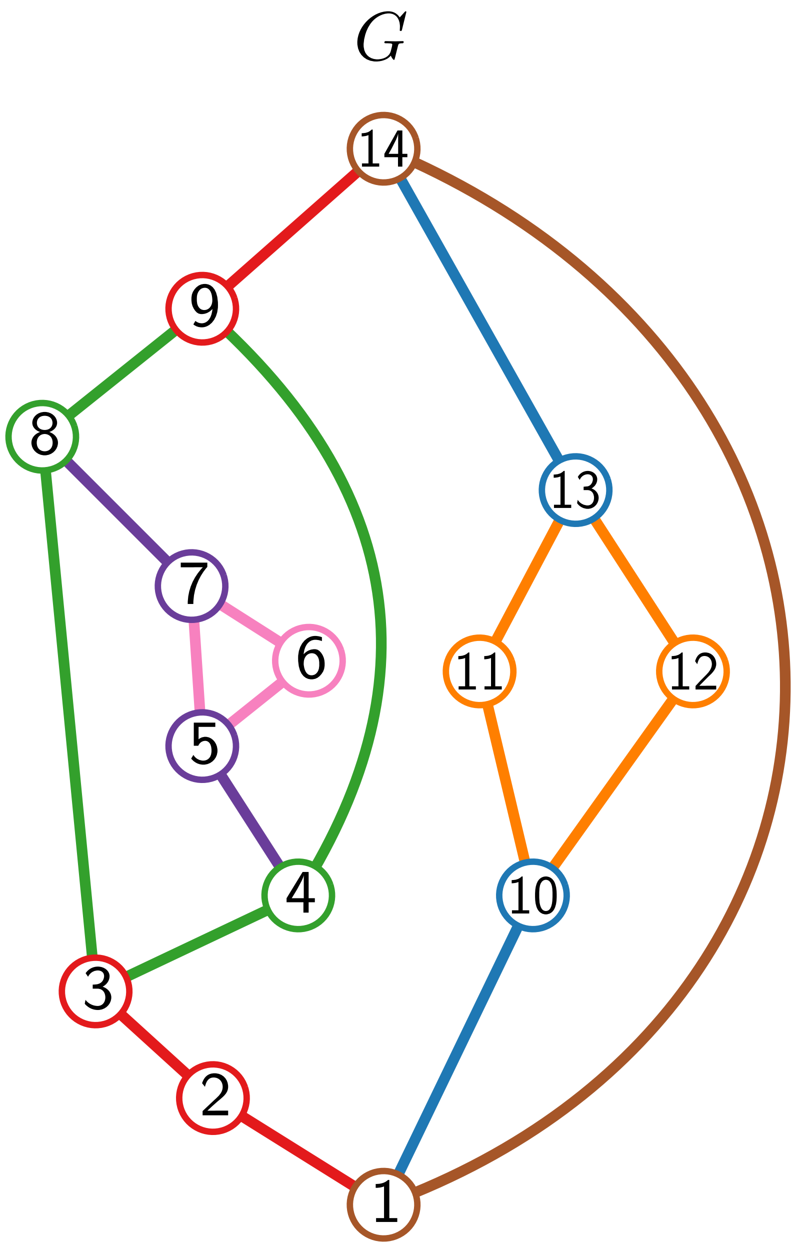
SPQR-Tree Example

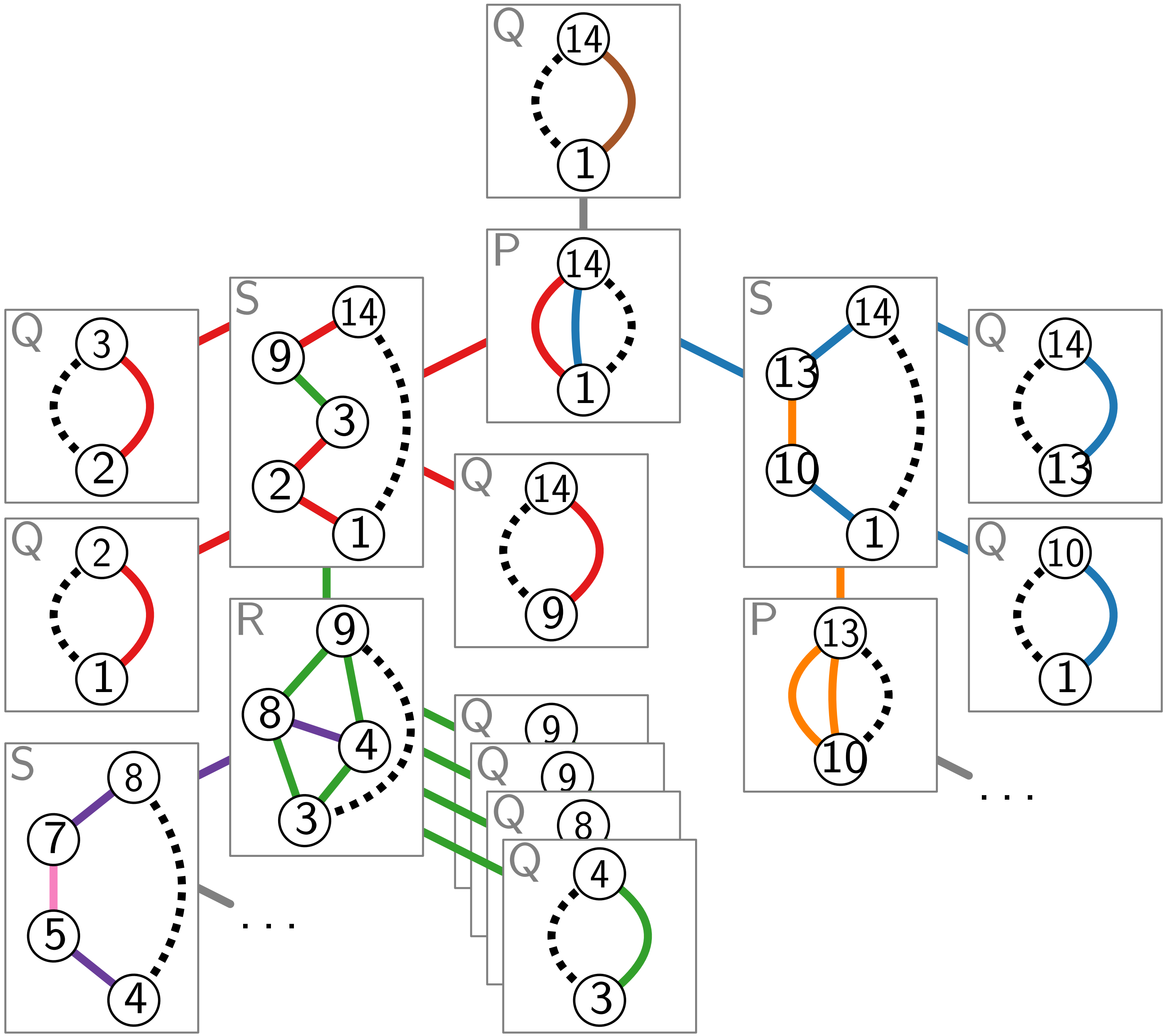


SPQR-Tree Example



SPQR-Tree Example

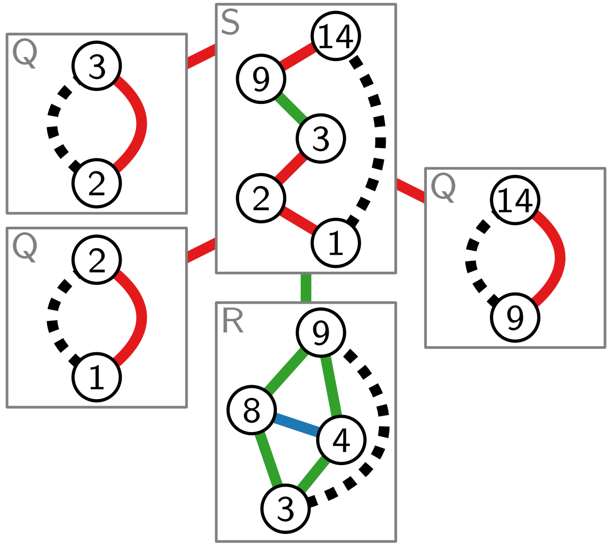




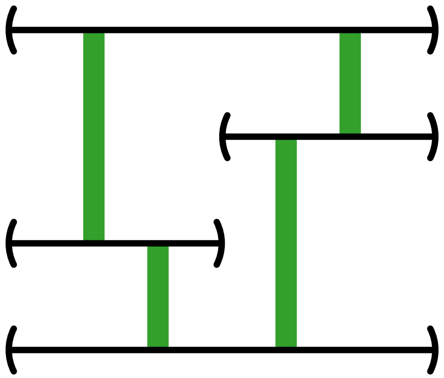
Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Visualization of Graphs

Lecture 9:
Partial Visibility Representation Extension



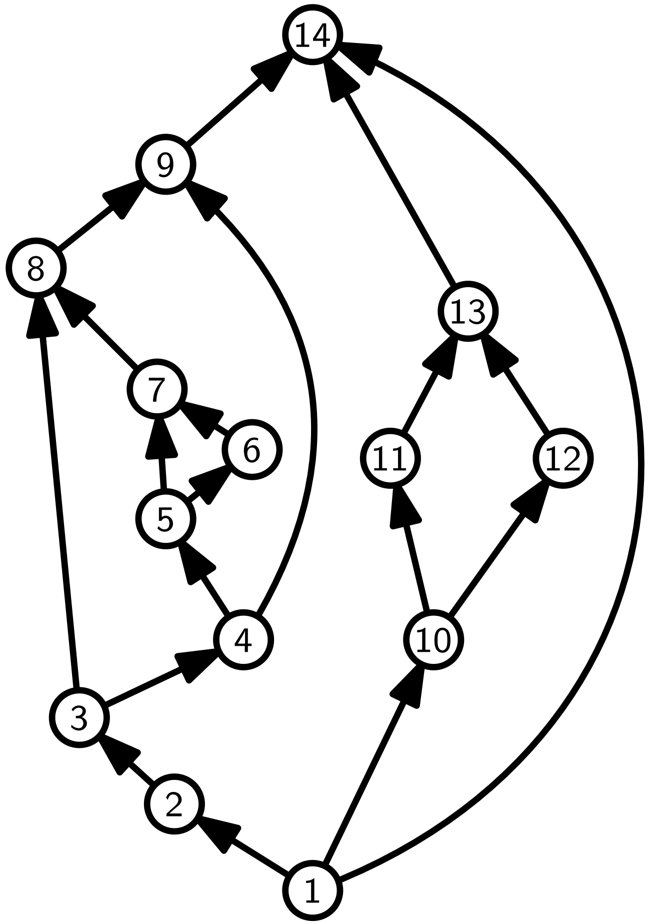
Part IV:
Rectangular
Representation Extension



Alexander Wolff

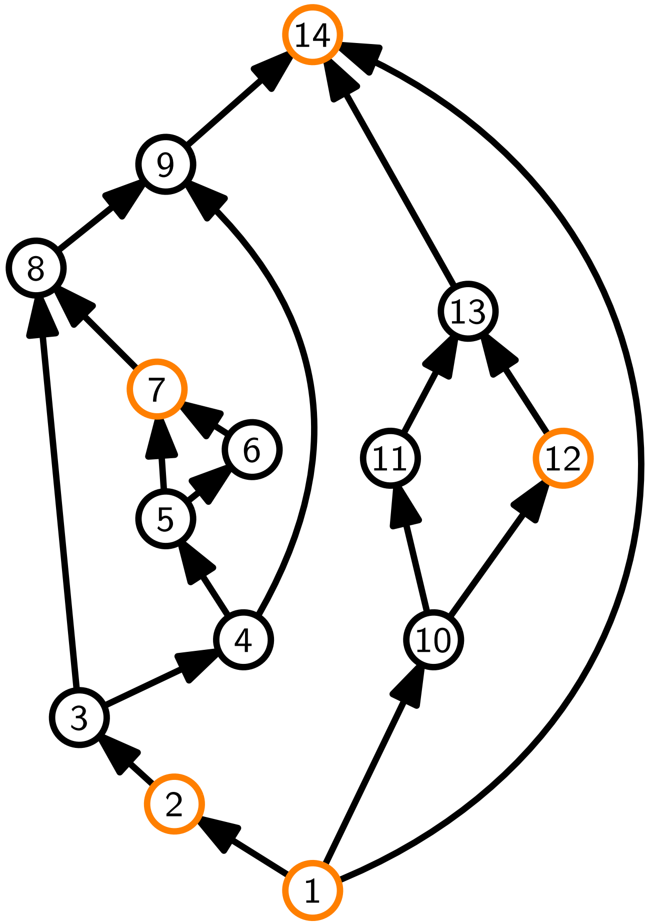
Representation Extension for st-Graphs

Theorem 1'.
Rectangular ε-Bar Visibility Representation Extension can be solved in $\mathcal{O}(n^2)$ time for st-graphs.



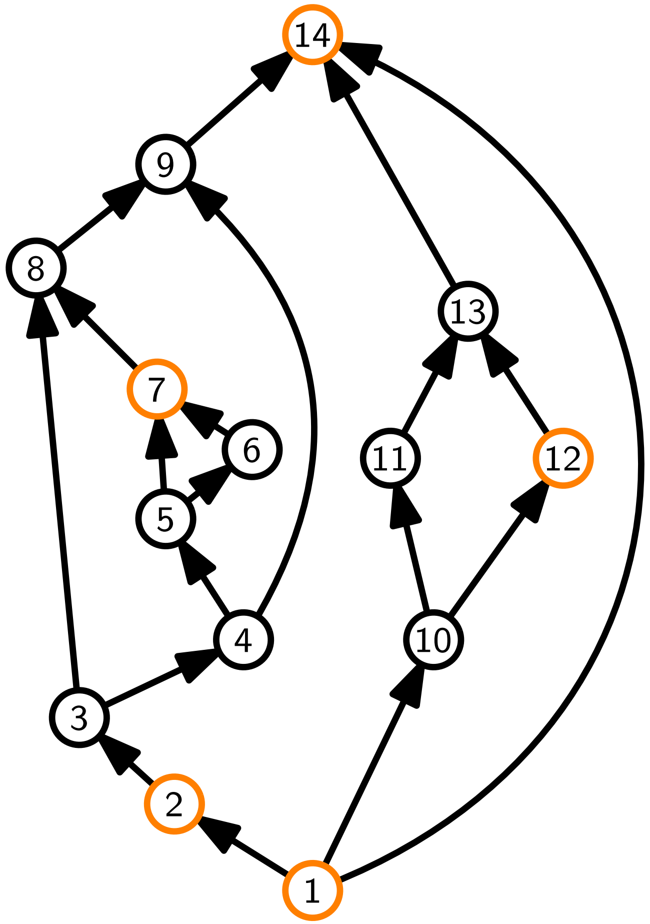
Representation Extension for st-Graphs

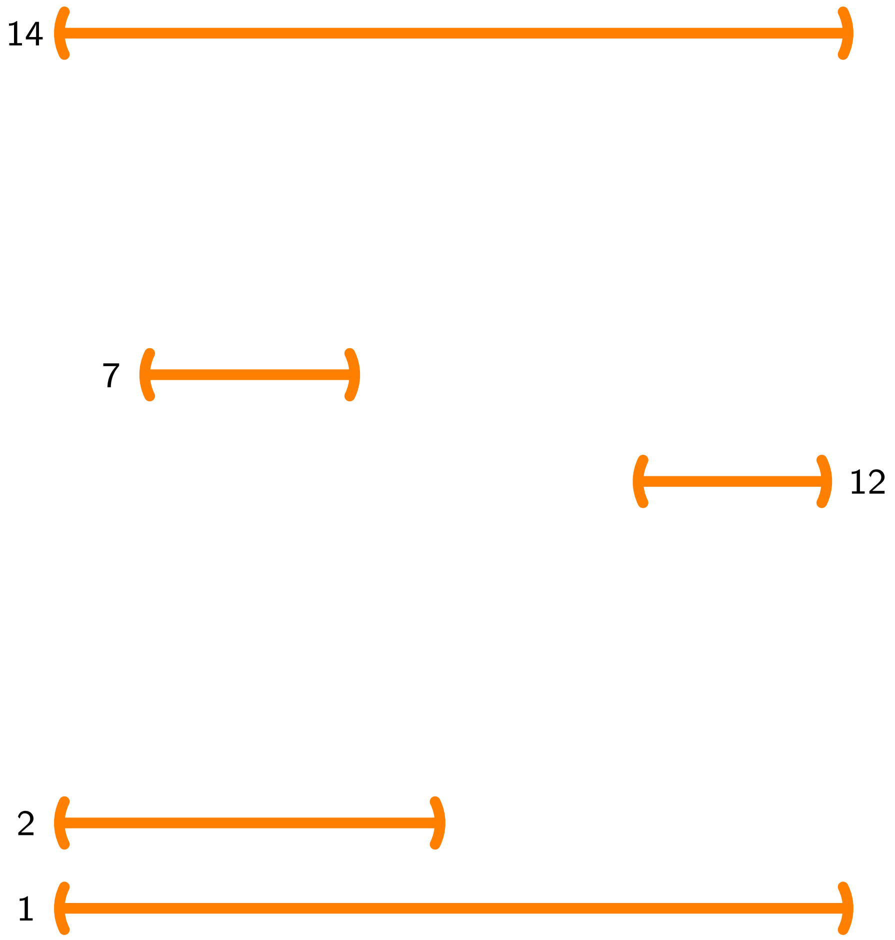
Theorem 1'.
Rectangular ε-Bar Visibility Representation Extension can be solved in $\mathcal{O}(n^2)$ time for st-graphs.



Representation Extension for st-Graphs

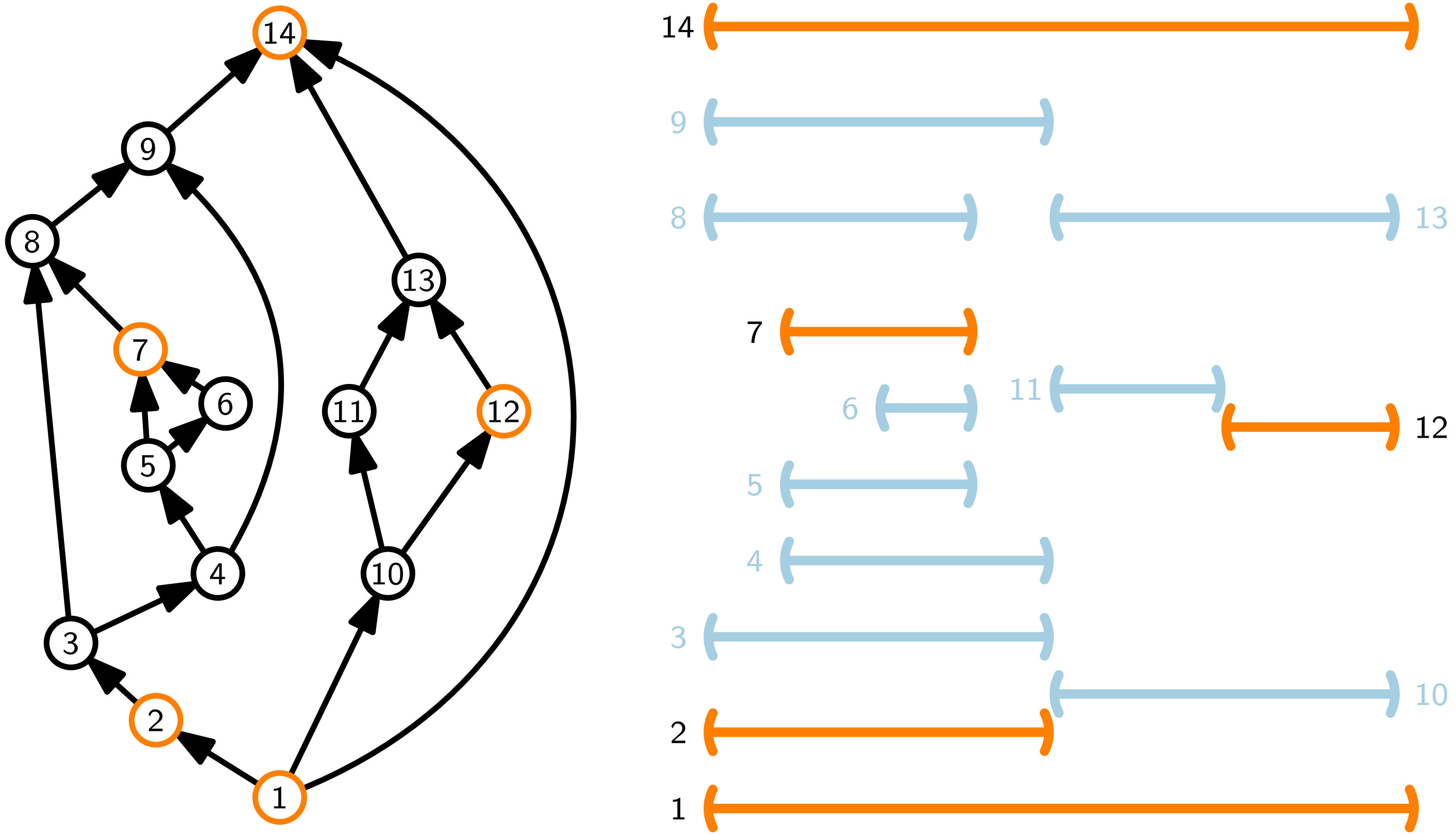
Theorem 1'.
Rectangular ε-Bar Visibility Representation Extension can be solved in $\mathcal{O}(n^2)$ time for st-graphs.





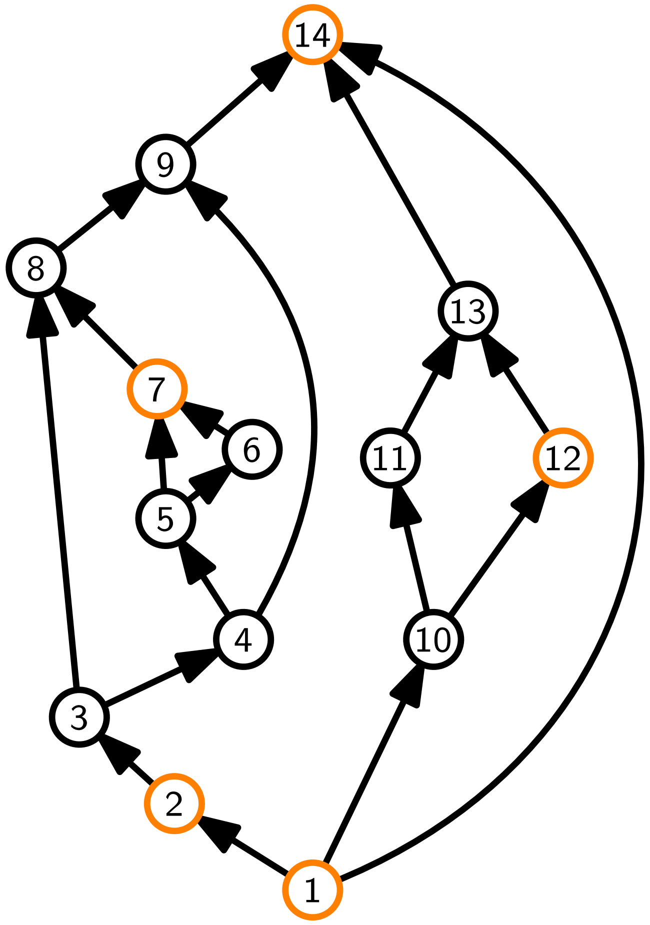
Representation Extension for st-Graphs

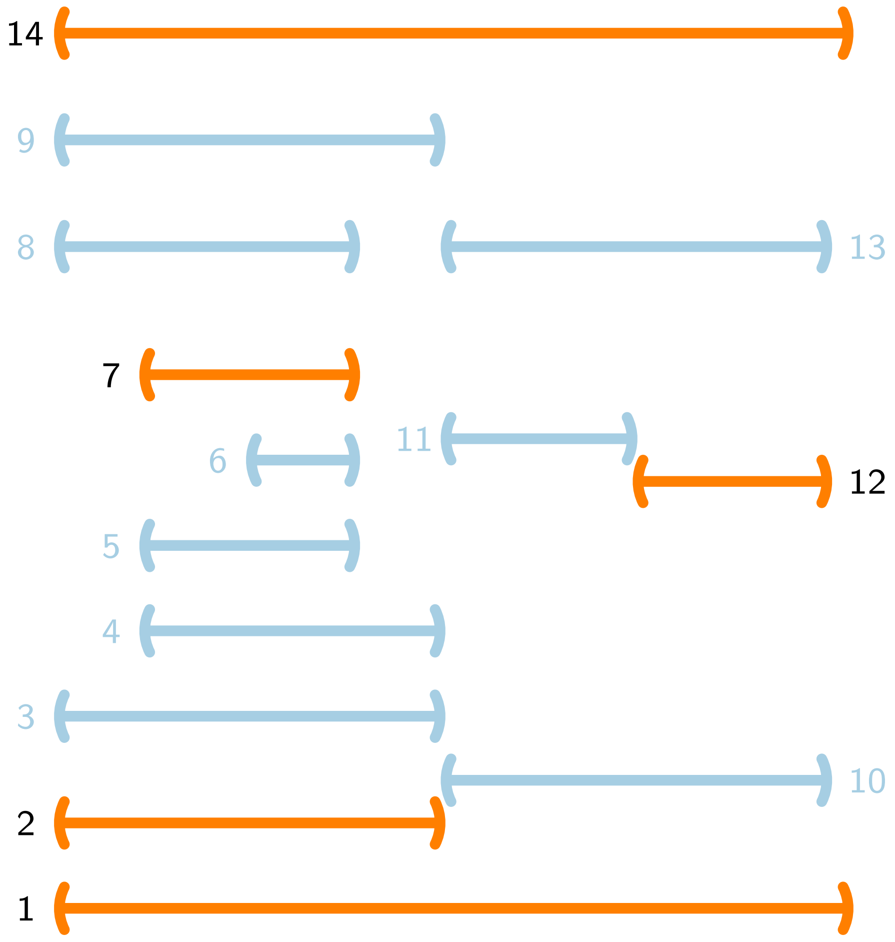
Theorem 1'.
Rectangular ε-Bar Visibility Representation Extension can be solved in $\mathcal{O}(n^2)$ time for st-graphs.



Representation Extension for st-Graphs

Theorem 1'.
Rectangular ε-Bar Visibility Representation Extension can be solved in $\mathcal{O}(n^2)$ time for st-graphs.

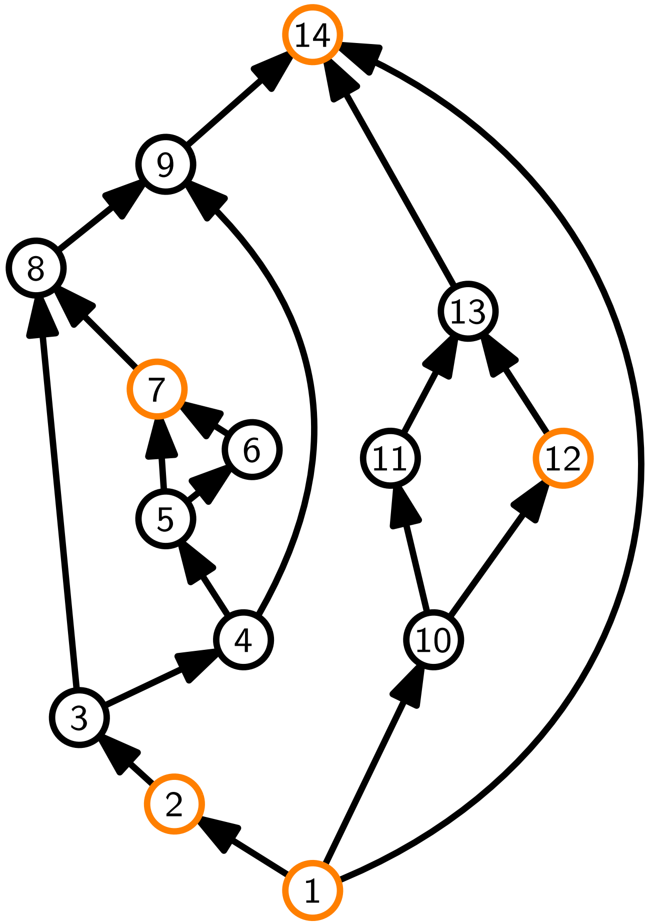


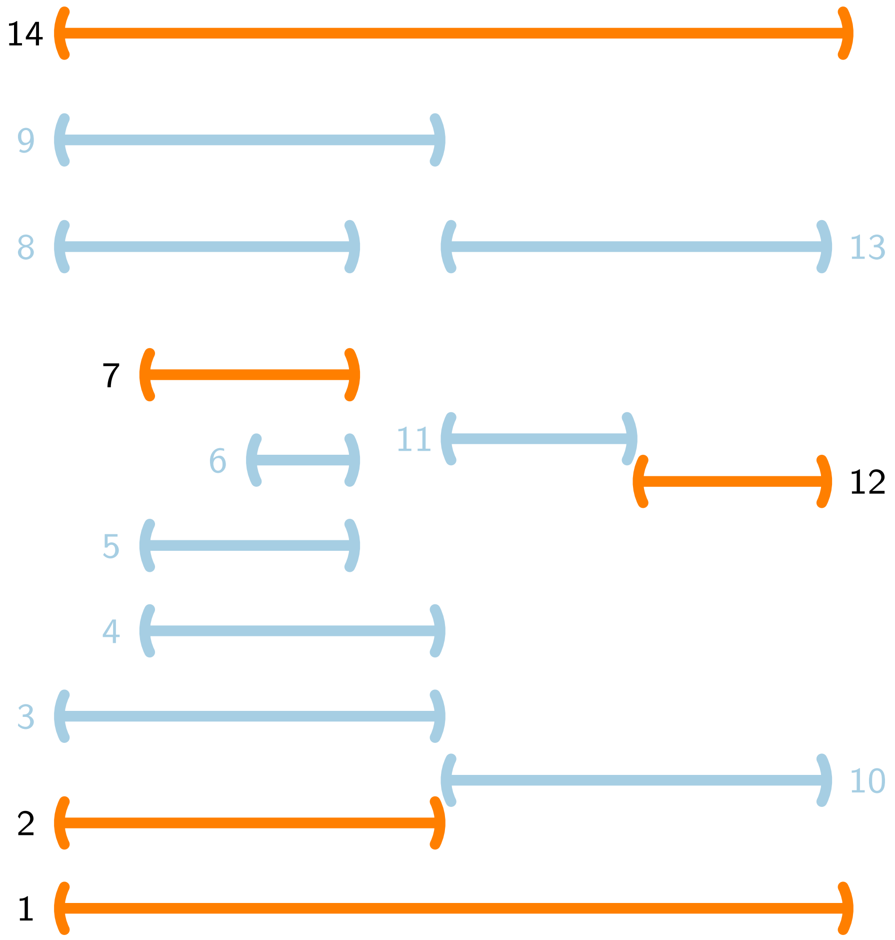


- Simplify with assumption on y-coordinates

Representation Extension for st-Graphs

Theorem 1'.
Rectangular ε-Bar Visibility Representation Extension can be solved in $\mathcal{O}(n^2)$ time for st-graphs.

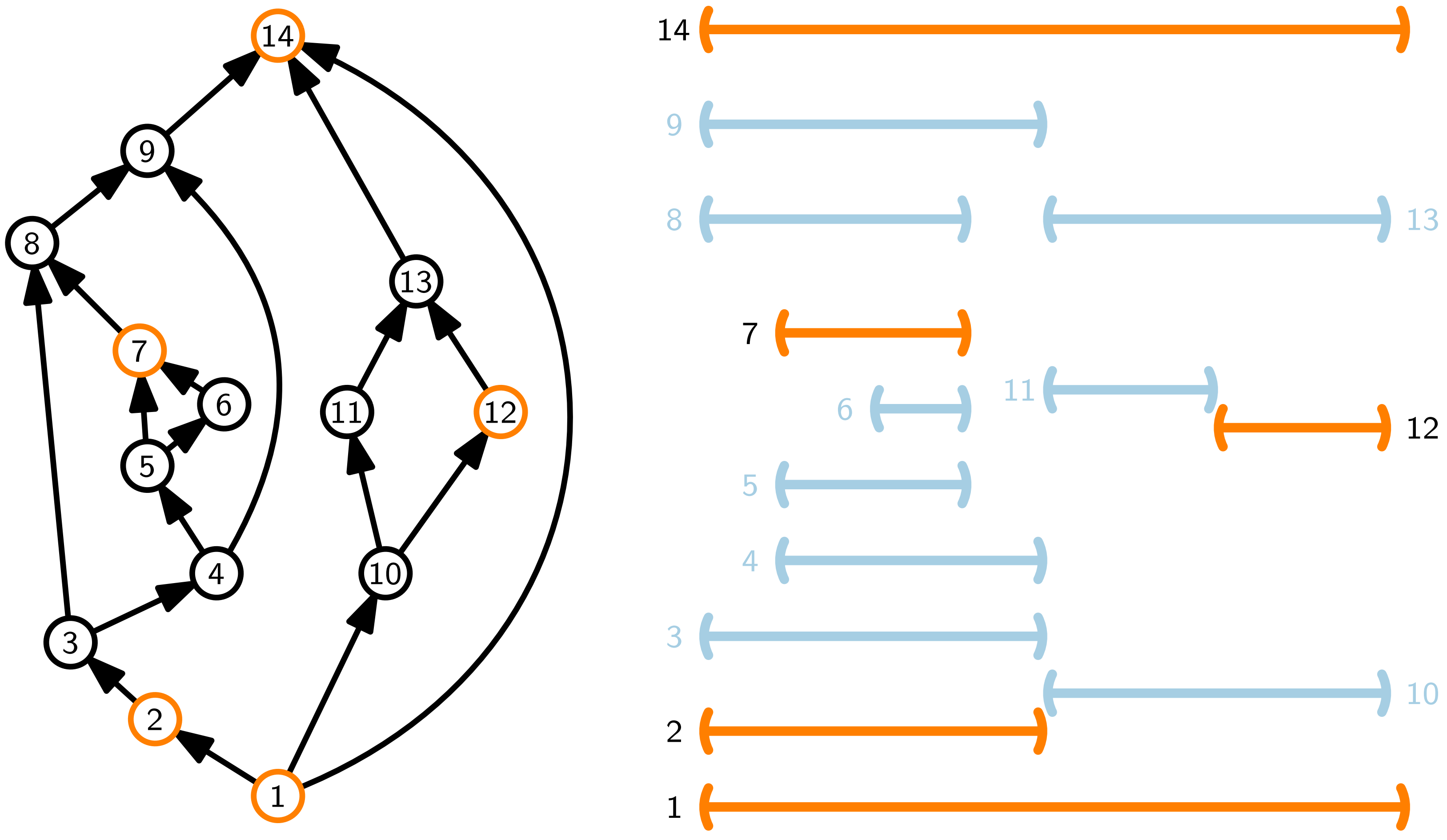




- Simplify with assumption on y-coordinates
- Look at connection to SPQR-trees – tiling

Representation Extension for st-Graphs

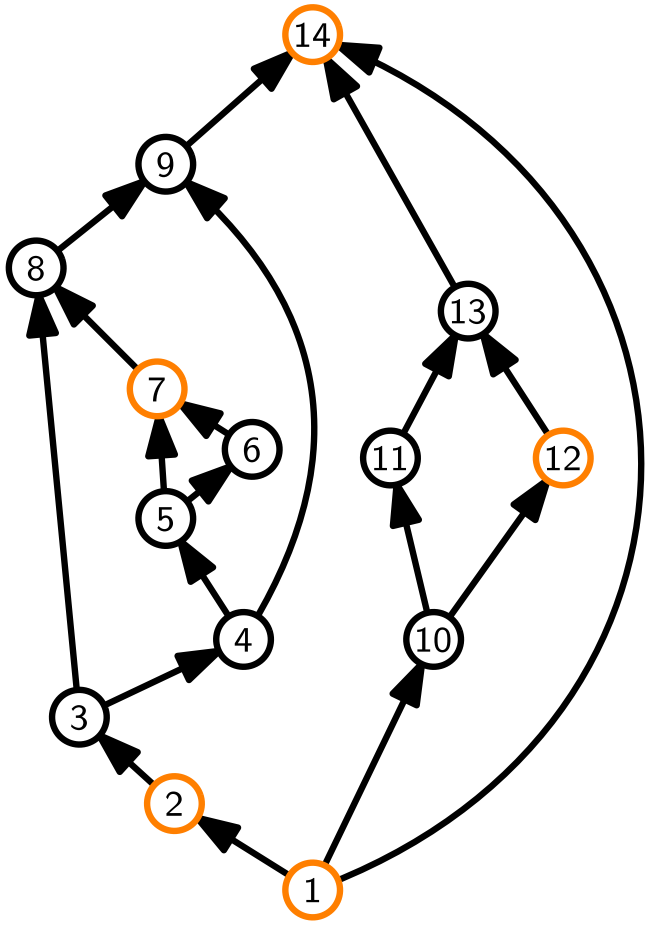
Theorem 1'.
Rectangular ε-Bar Visibility Representation Extension can be solved in $\mathcal{O}(n^2)$ time for st-graphs.

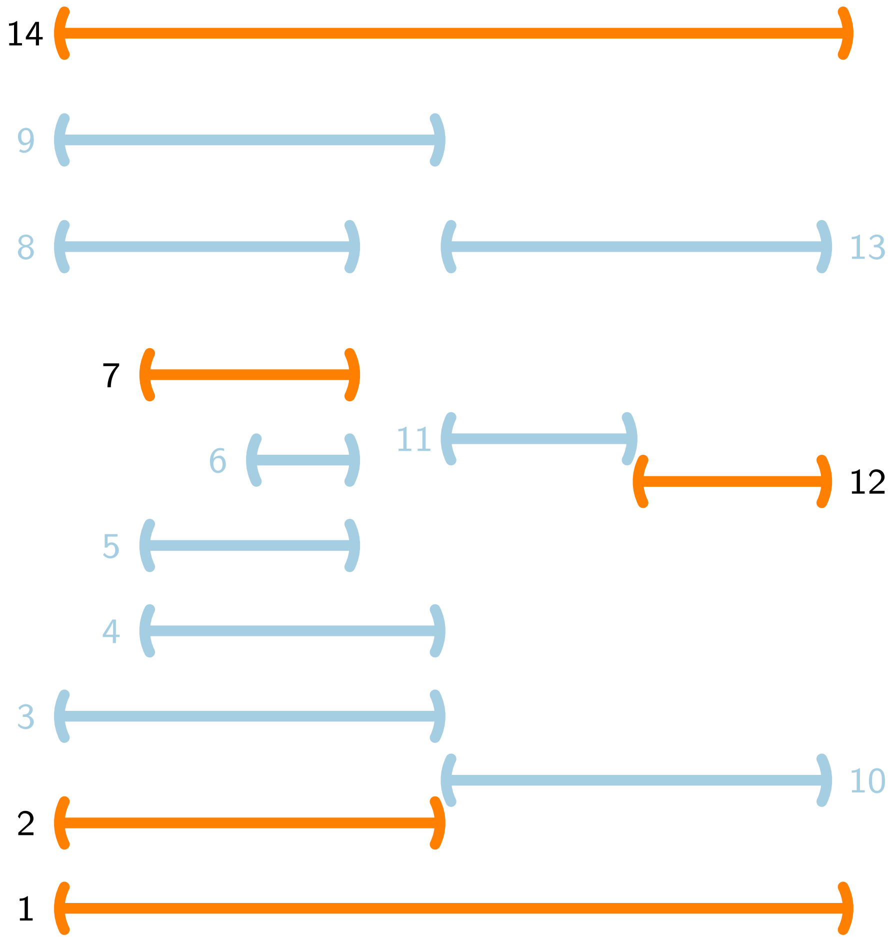


- Simplify with assumption on y-coordinates
- Look at connection to SPQR-trees – tiling
- Solve problems for **S**-, **P**-, and **R**-nodes

Representation Extension for st-Graphs

Theorem 1'.
Rectangular ε-Bar Visibility Representation Extension can be solved in $\mathcal{O}(n^2)$ time for st-graphs.





- Simplify with assumption on y-coordinates
- Look at connection to SPQR-trees – tiling
- Solve problems for **S**-, **P**-, and **R**-nodes
- Dynamic program via SPQR-tree

y-Coordinate Invariant

- Let $G = (V, E)$ be an st-graph, and let ψ' be a representation of $V' \subseteq V$.

y-Coordinate Invariant

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

y-Coordinate Invariant

- Let $G = (V, E)$ be an st-graph, and let ψ' be a representation of $V' \subseteq V$.
- Let $y\colon V \to \mathbb{R}$ such that
  - for each $v \in V'$, $y(v) =$ the y-coordinate of $\psi'(v)$.

y-Coordinate Invariant

- Let $G = (V, E)$ be an st-graph, and let ψ' be a representation of $V' \subseteq V$.
- Let $y\colon V \to \mathbb{R}$ such that
  - for each $v \in V'$, $y(v) =$ the y-coordinate of $\psi'(v)$.
  - for each edge (u, v), $y(u) < y(v)$.

y-Coordinate Invariant

- Let $G = (V, E)$ be an st-graph, and let ψ' be a representation of $V' \subseteq V$.
- Let $y \colon V \to \mathbb{R}$ such that
  - for each $v \in V'$, $y(v) =$ the y-coordinate of $\psi'(v)$.
  - for each edge (u, v), $y(u) < y(v)$.

Lemma 1.
G has a representation extending ψ' $\Leftrightarrow$
G has a representation extending ψ'
where the y-coordinates of the bars are as in y.

y-Coordinate Invariant

- Let $G = (V, E)$ be an st-graph, and let ψ' be a representation of $V' \subseteq V$.
- Let $y\colon V \to \mathbb{R}$ such that
  - for each $v \in V'$, $y(v) =$ the y-coordinate of $\psi'(v)$.
  - for each edge (u, v), $y(u) < y(v)$.

Lemma 1.
G has a representation extending ψ' $\Leftrightarrow$
G has a representation extending ψ'
where the y-coordinates of the bars are as in y.

Proof idea. The relative positions of **adjacent** bars must 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.

y-Coordinate Invariant

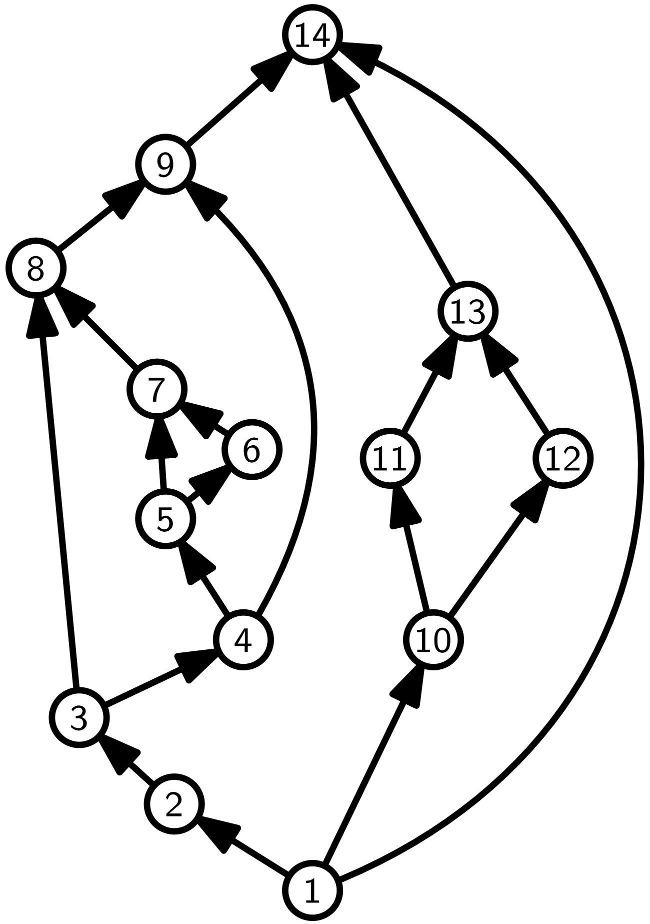
- Let $G = (V, E)$ be an st-graph, and let ψ' be a representation of $V' \subseteq V$.
- Let $y \colon V \to \mathbb{R}$ such that
  - for each $v \in V'$, $y(v) =$ the y-coordinate of $\psi'(v)$.
  - for each edge (u, v), $y(u) < y(v)$.

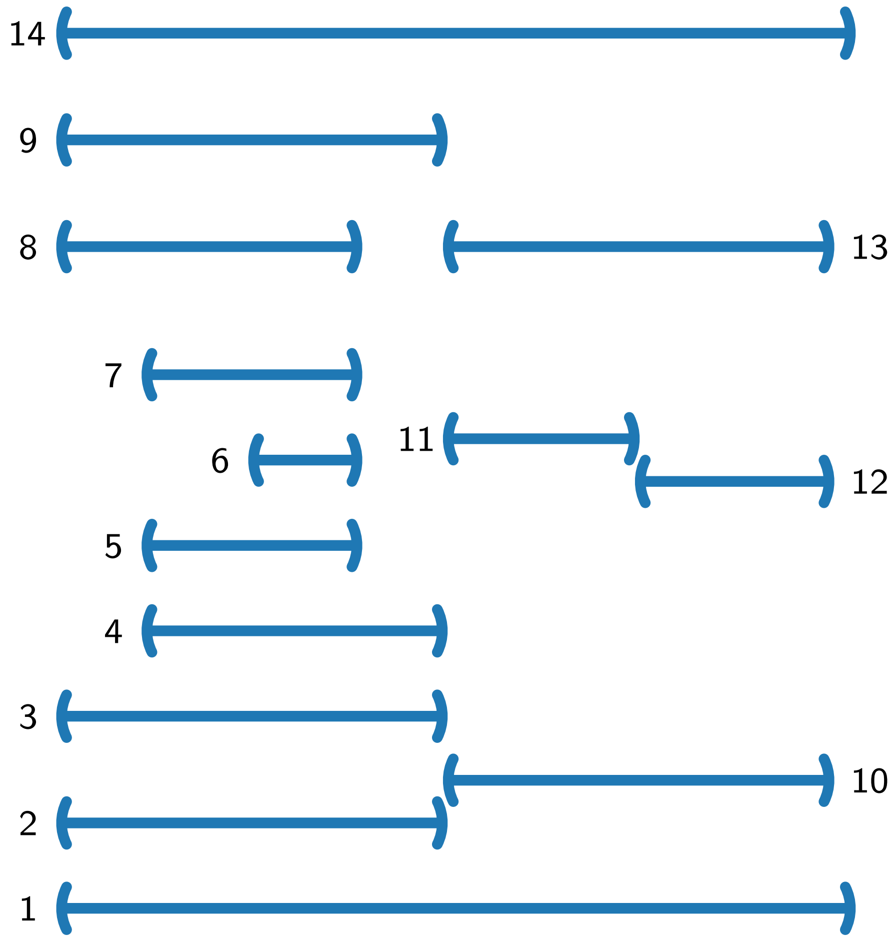
Lemma 1.
G has a representation extending ψ' $\Leftrightarrow$
G has a representation extending ψ'
where the y-coordinates of the bars are as in y.

We can now assume that all y-coordinates are given!

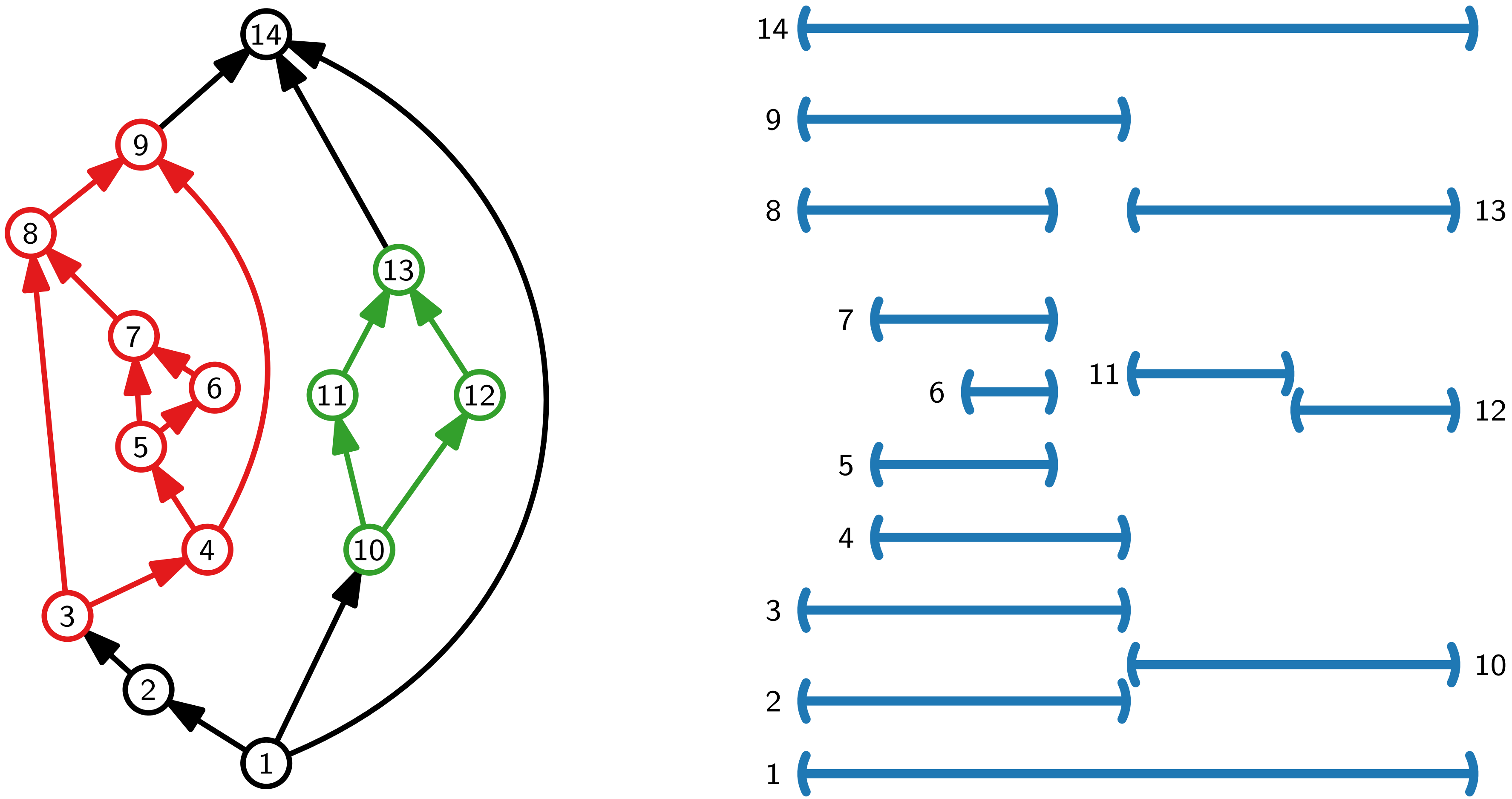
Proof idea. The relative positions of **adjacent** bars must match the order given by y.
So, we can adjust the y-coordinates of any solution to be as in y by sweeping from bottom to top.

But why do SPQR-Trees help?

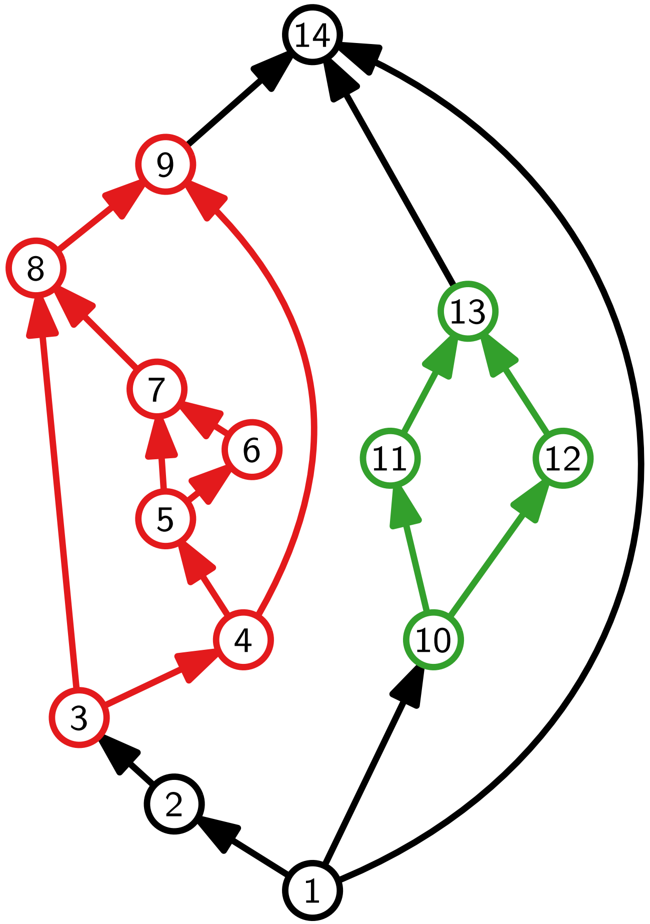


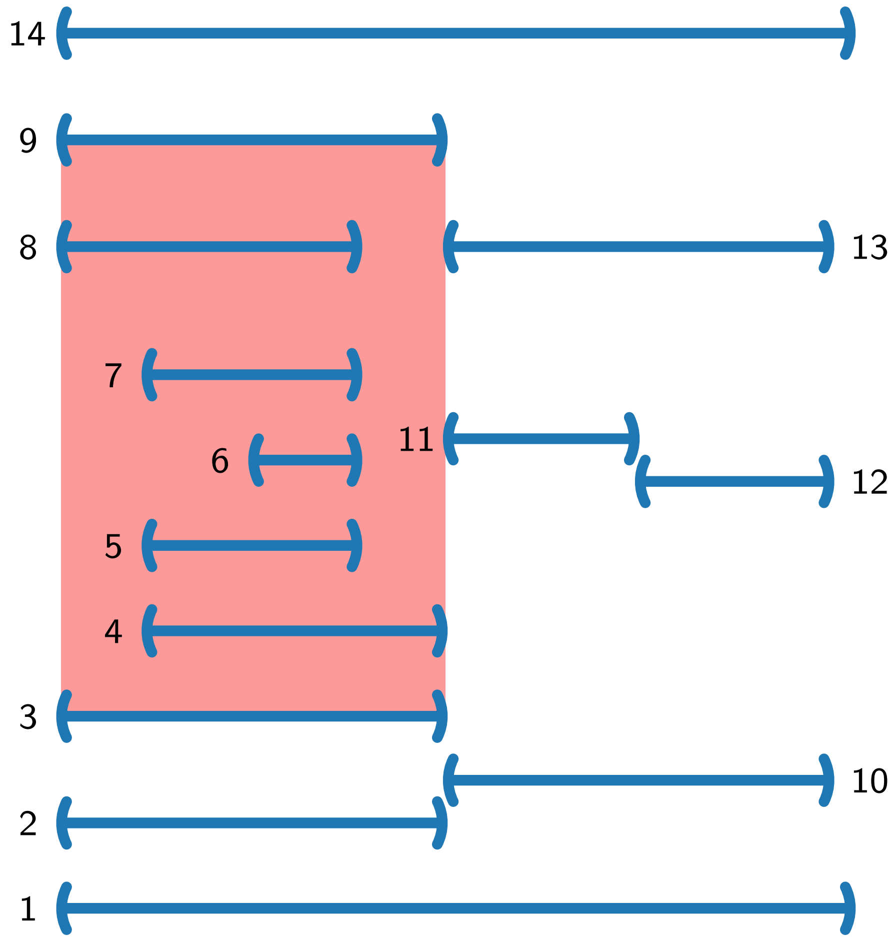


But why do SPQR-Trees help?

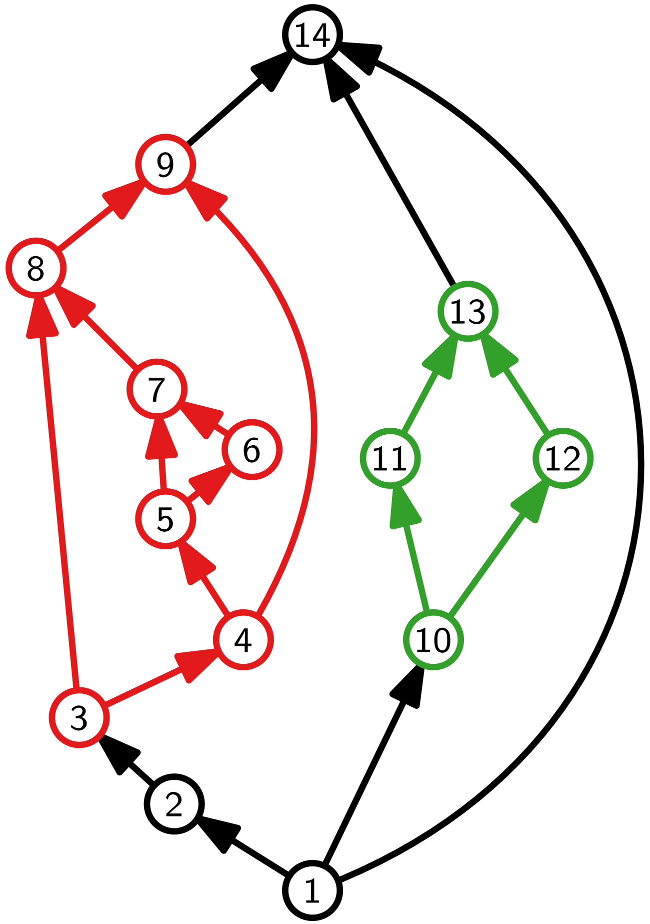


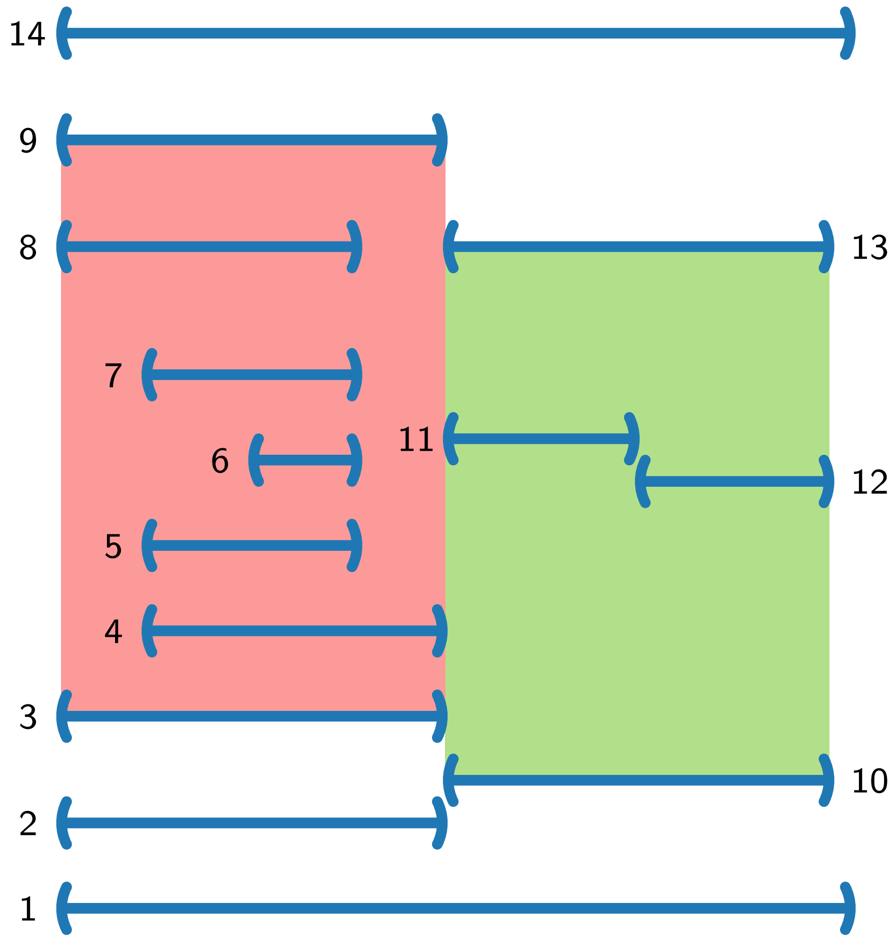
But why do SPQR-Trees help?





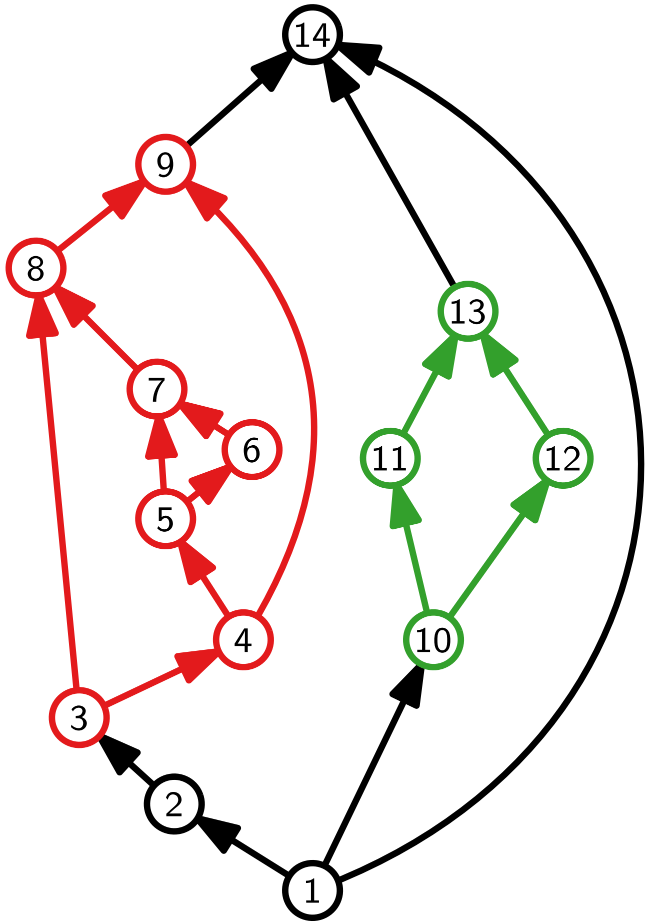
But why do SPQR-Trees help?

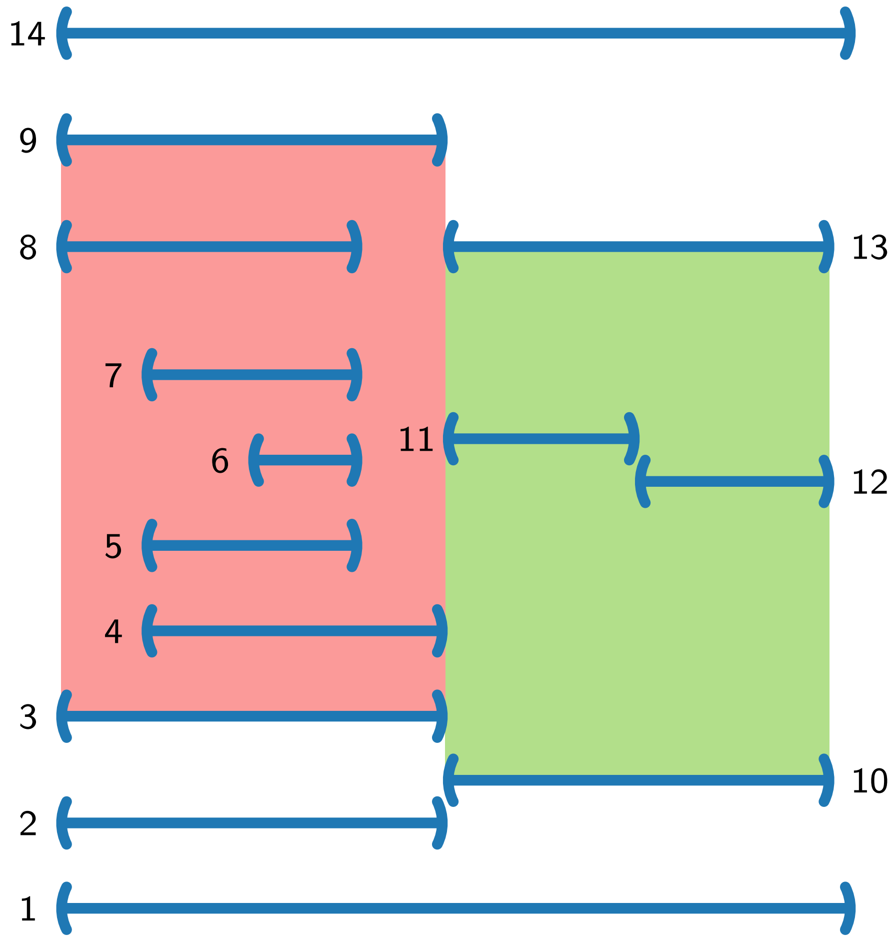




But why do SPQR-Trees help?

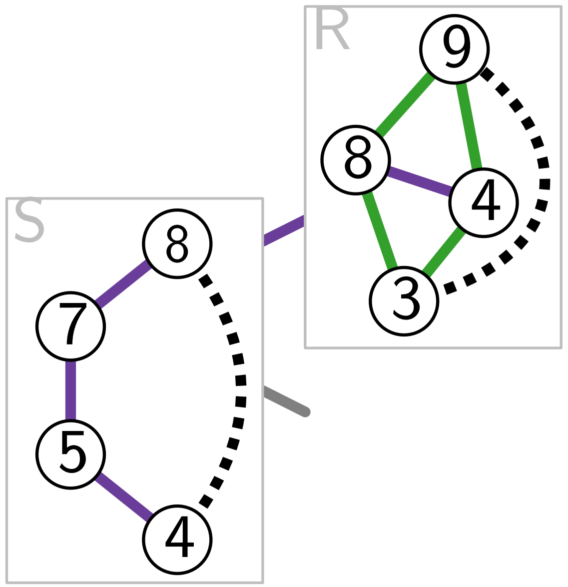
Lemma 2.
The SPQR-tree of an st-graph G induces a recursive **tiling** of any ε-bar visibility representation of G.



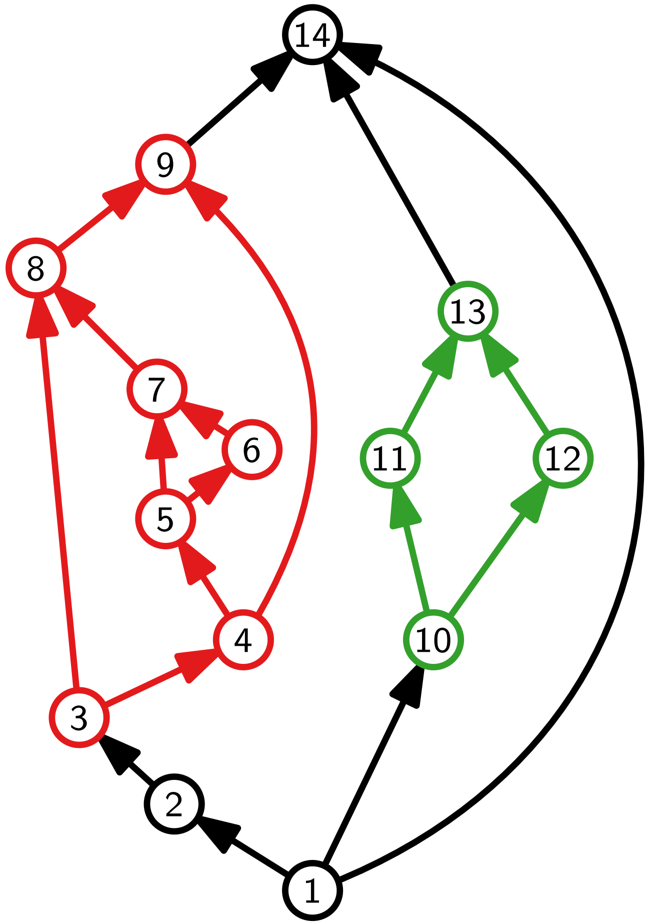


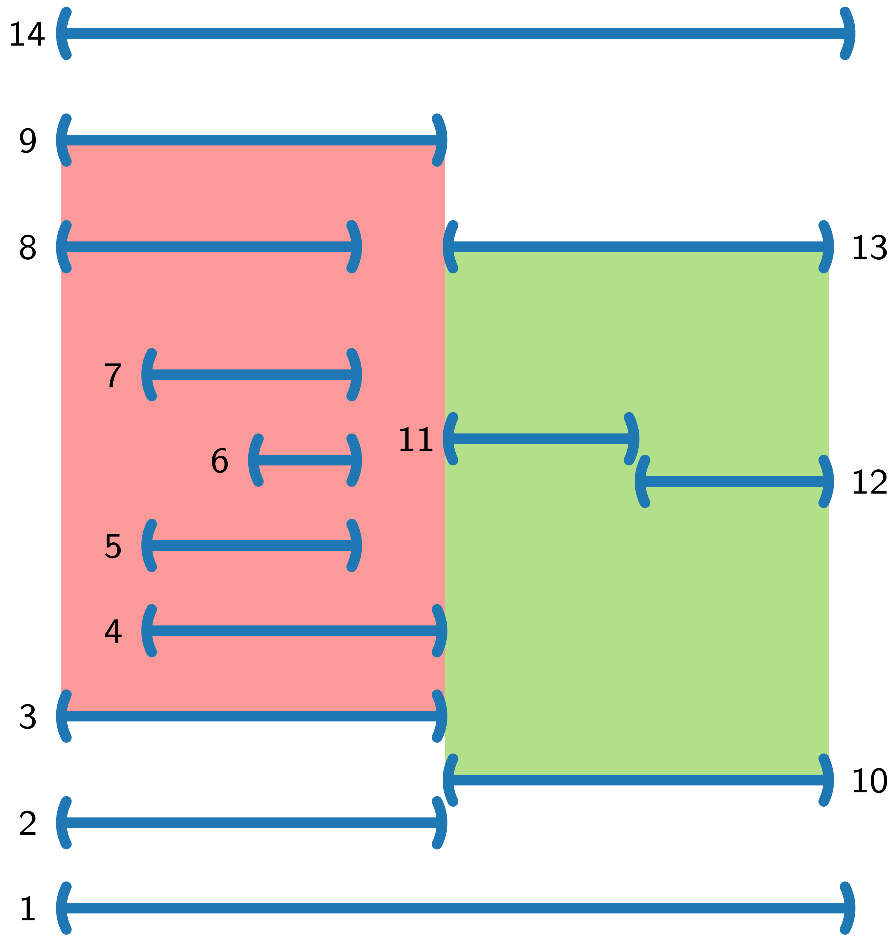
But why do SPQR-Trees help?

Lemma 2.
The SPQR-tree of an st-graph G induces a recursive **tiling** of any ε-bar visibility representation of G.



Solve tiles bottom-up

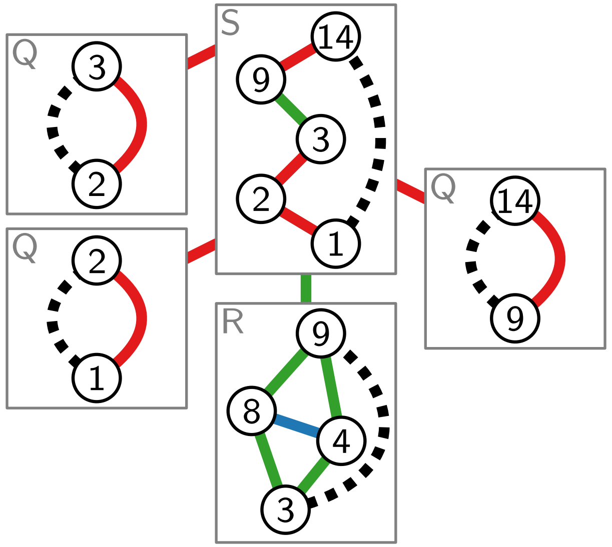




Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

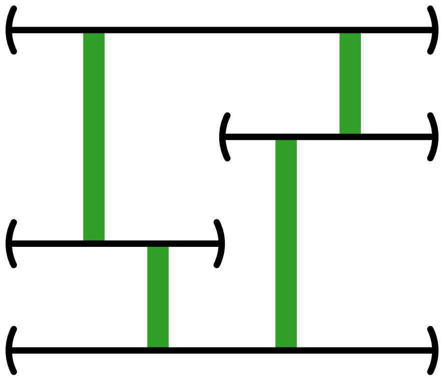
Visualization of Graphs

Lecture 9:
Partial Visibility Representation Extension



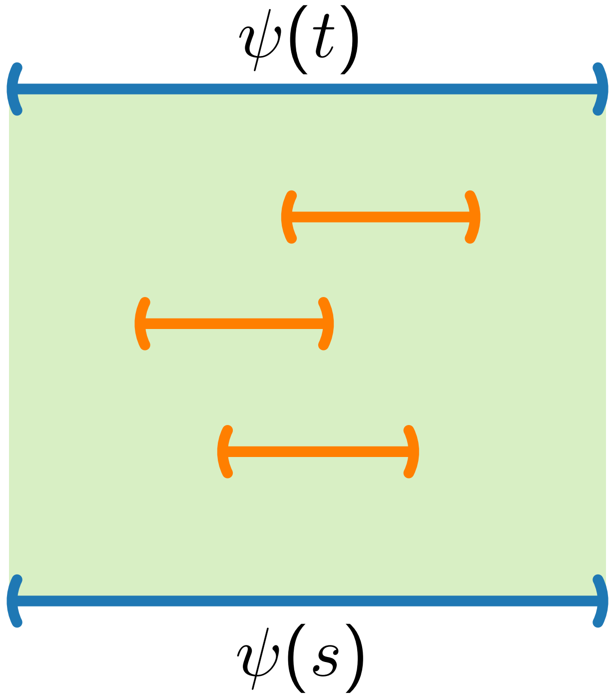
Part V:
Dynamic Program

Alexander Wolff



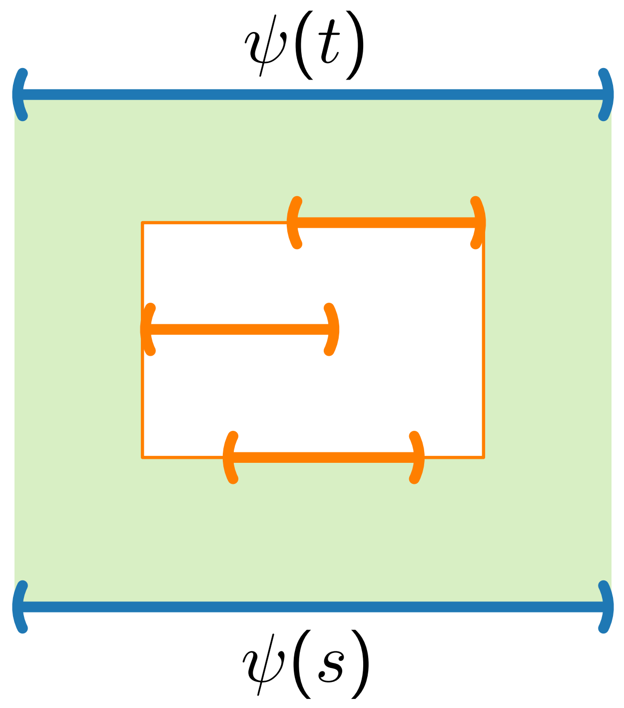
Tiles

Convention. **Orange** bars are from the partial representation



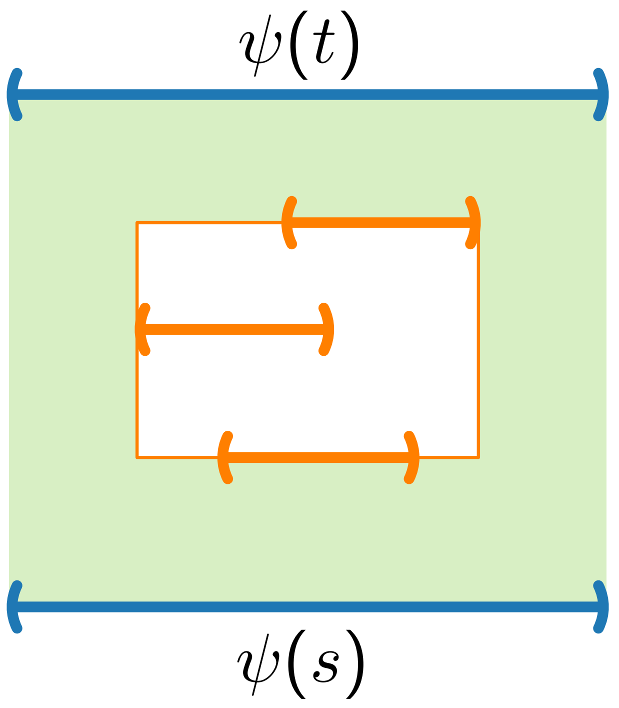
Tiles

Convention. **Orange** bars are from the partial representation



Tiles

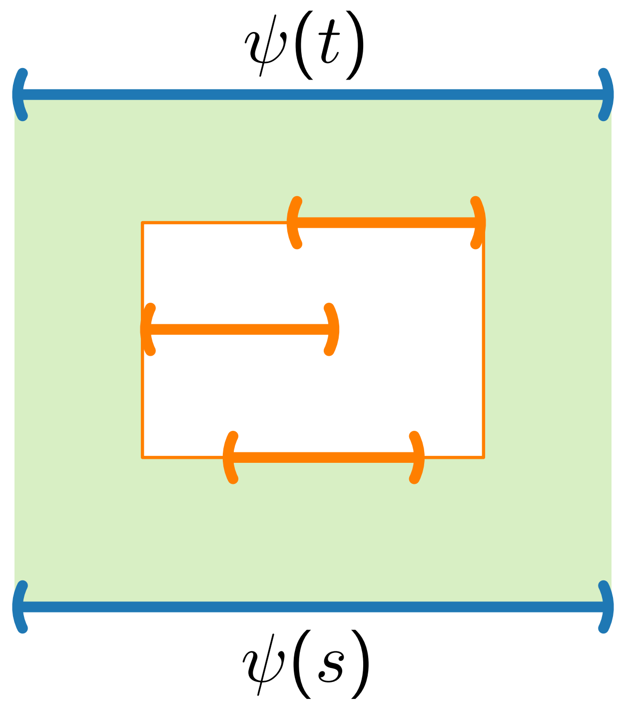
Convention. **Orange** bars are from the partial representation



Observation.
The bounding box (tile) of any solution ψ **contains** the bounding box of the partial representation.

Tiles

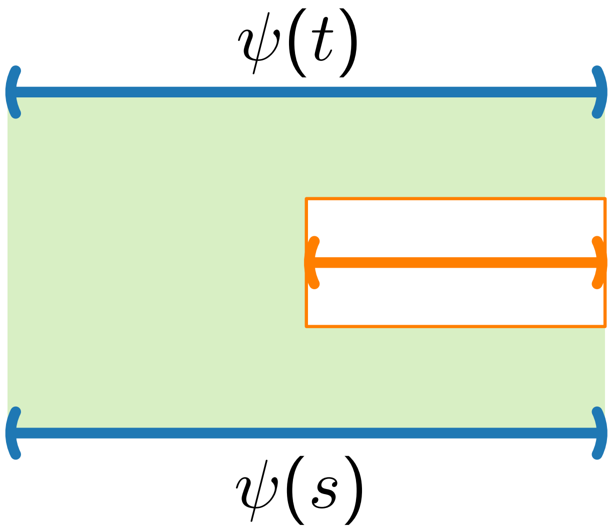
Convention. **Orange** bars are from the partial representation



Observation.
The bounding box (tile) of any solution ψ **contains** the bounding box of the partial representation.

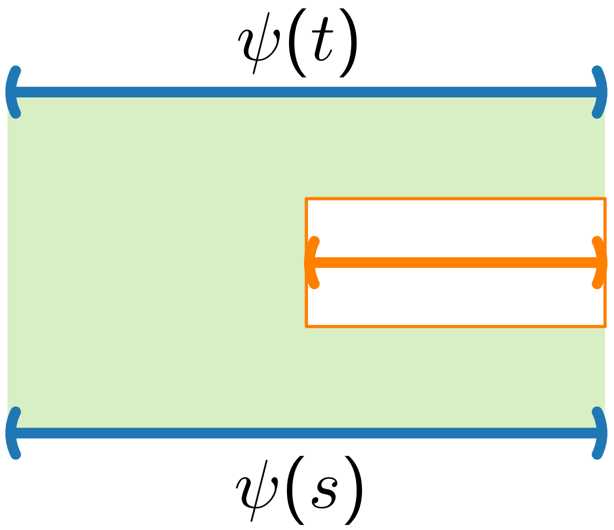
How many different **types** of tiles are there?

Types of Tiles

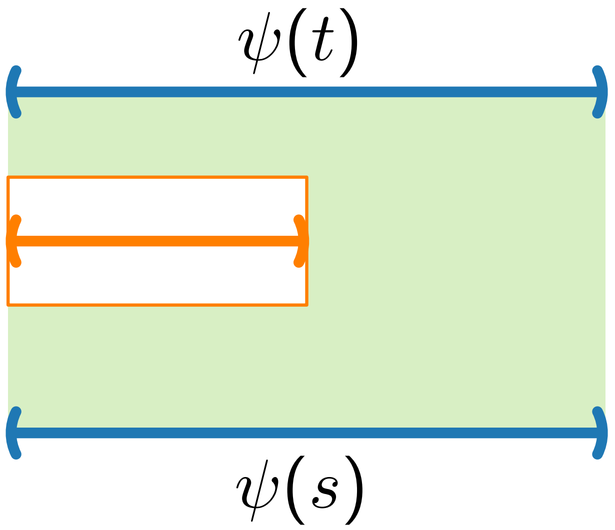


- Right **F**ixed – due to the orange bar
- Left **L**oose – due to the orange bar

Types of Tiles

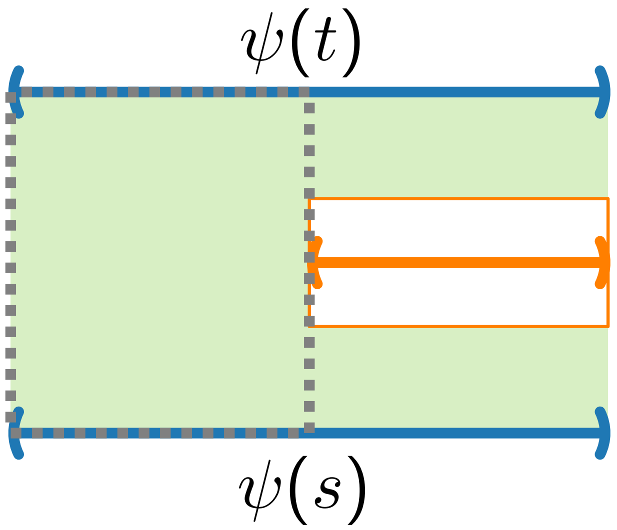


- Right **F**ixed – due to the orange bar
- Left **L**oose – due to the orange bar

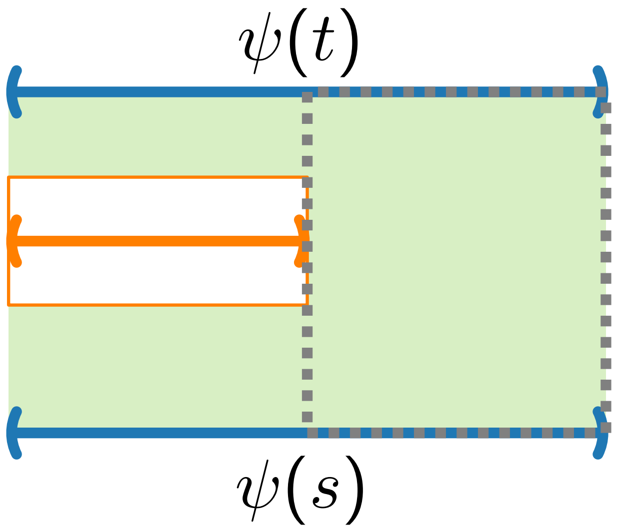


- Left **F**ixed – due to the orange bar
- Right **L**oose – due to the orange bar

Types of Tiles

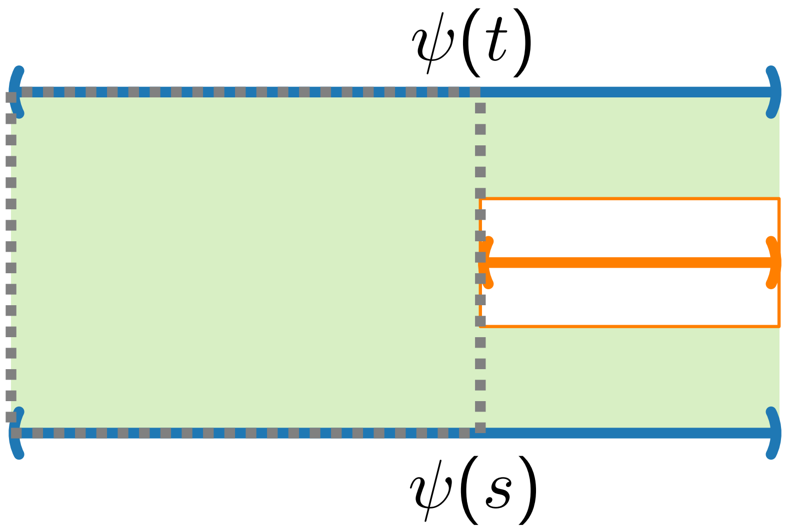


- Right **F**ixed – due to the orange bar
- Left **L**oose – due to the orange bar

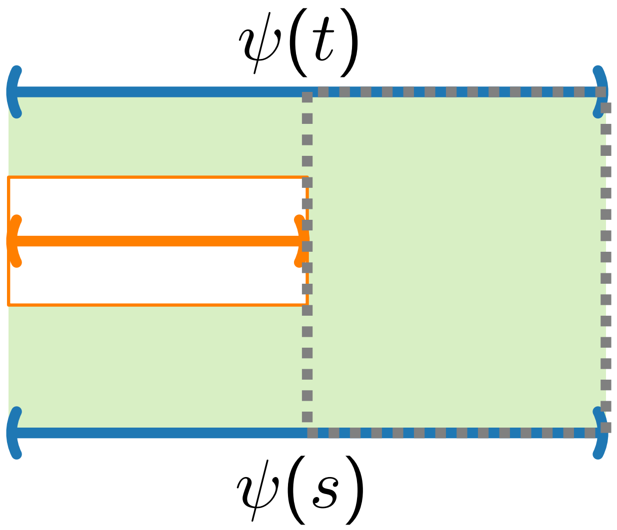


- Left **F**ixed – due to the orange bar
- Right **L**oose – due to the orange bar

Types of Tiles

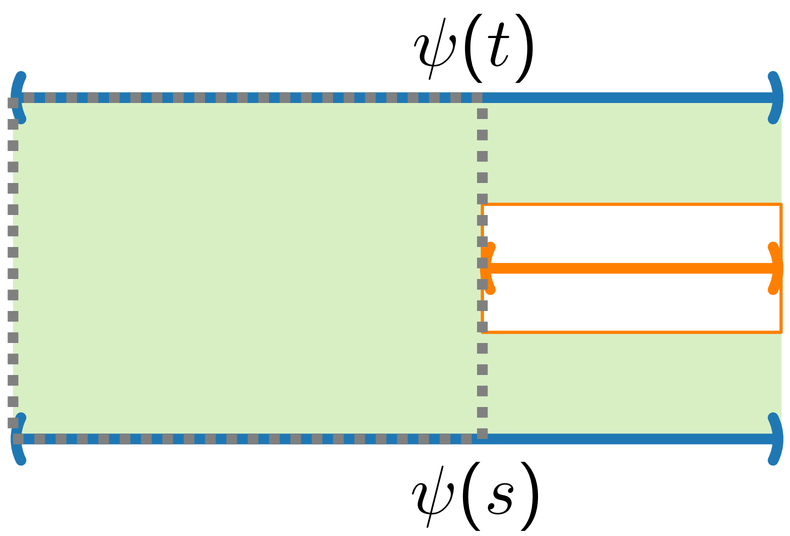


- Right **F**ixed – due to the orange bar
- Left **L**oose – due to the orange bar

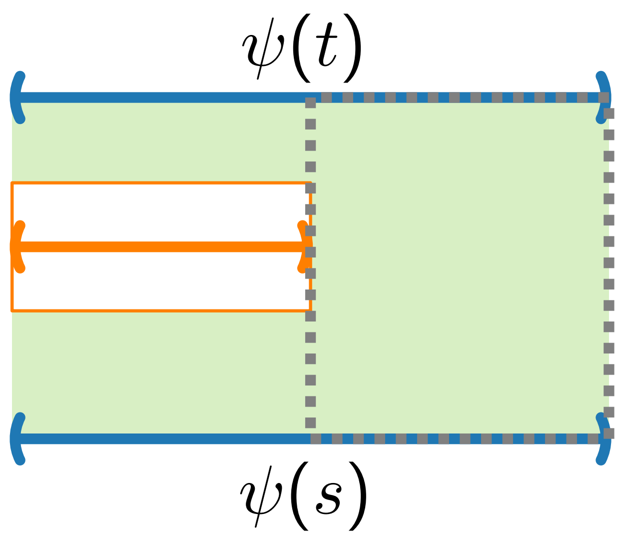


- Left **F**ixed – due to the orange bar
- Right **L**oose – due to the orange bar

Types of Tiles



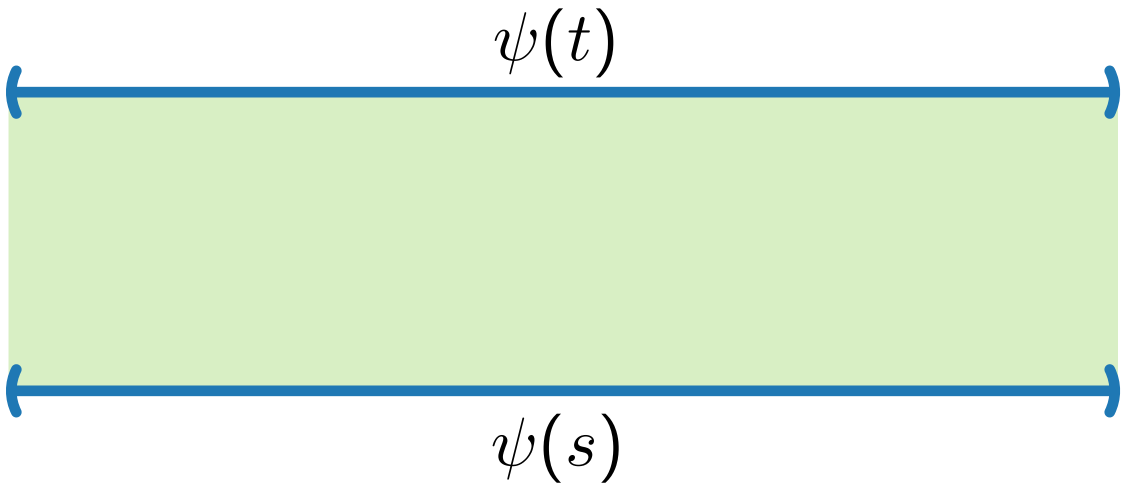
- Right **F**ixed – due to the orange bar
- Left **L**oose – due to the orange bar

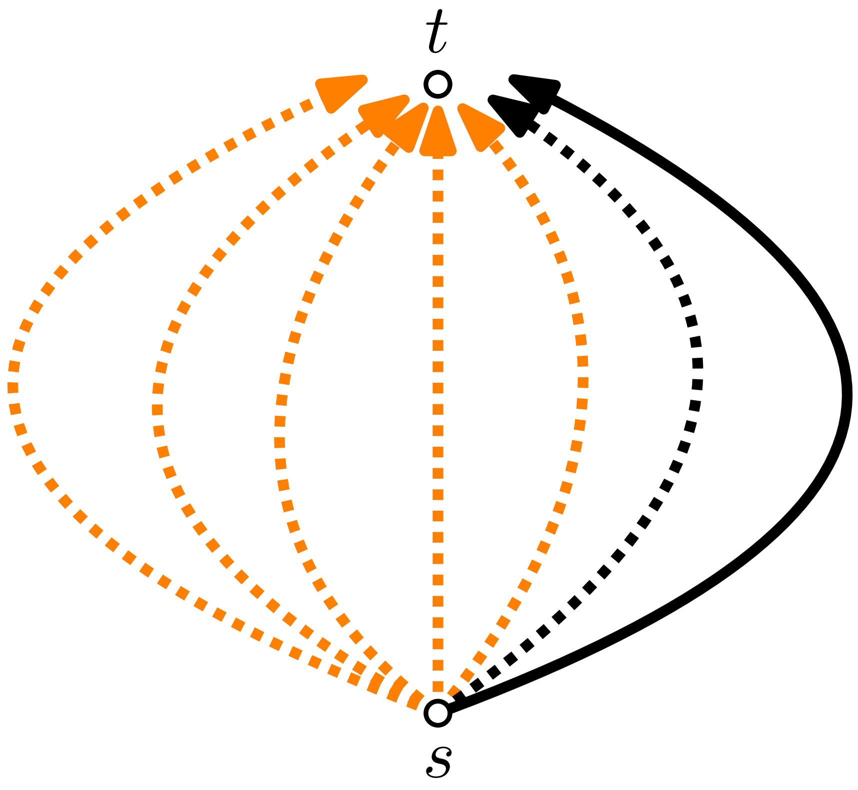


- Left **F**ixed – due to the orange bar
- Right **L**oose – due to the orange bar

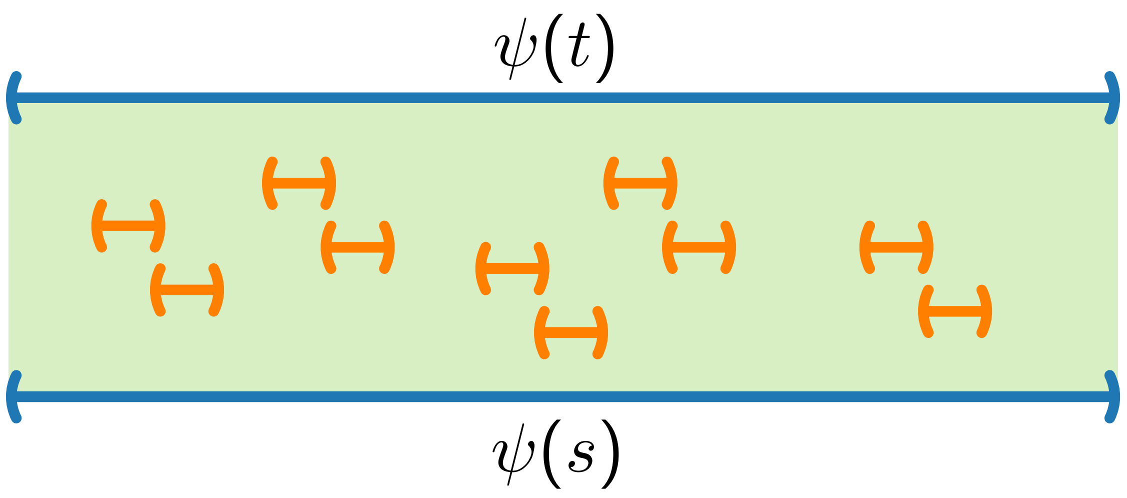
Four different types: **FF, FL, LF, LL**

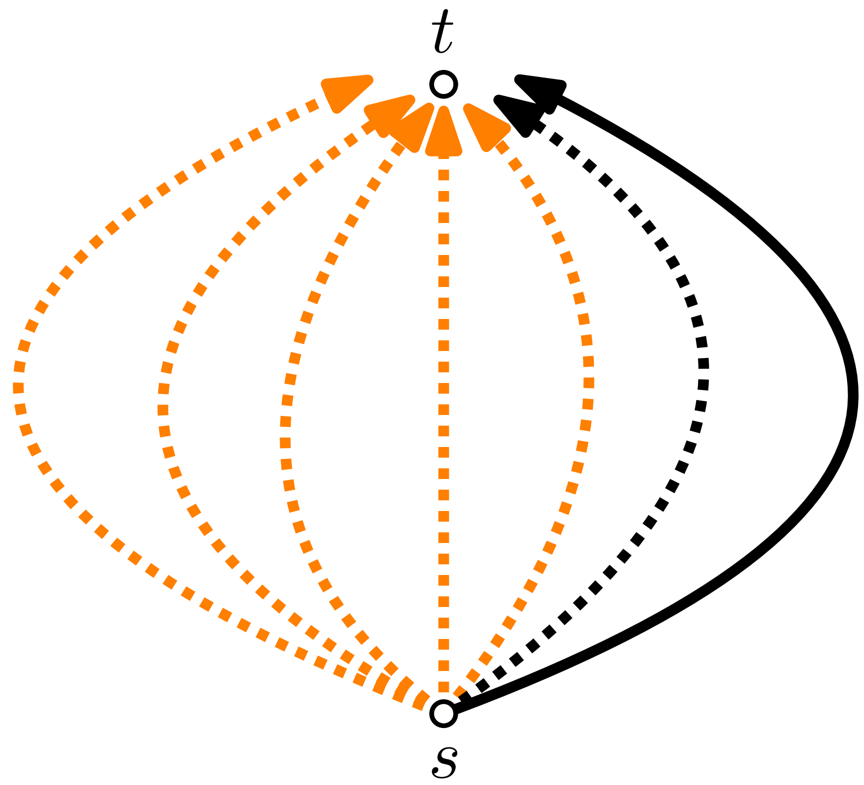
P-Nodes



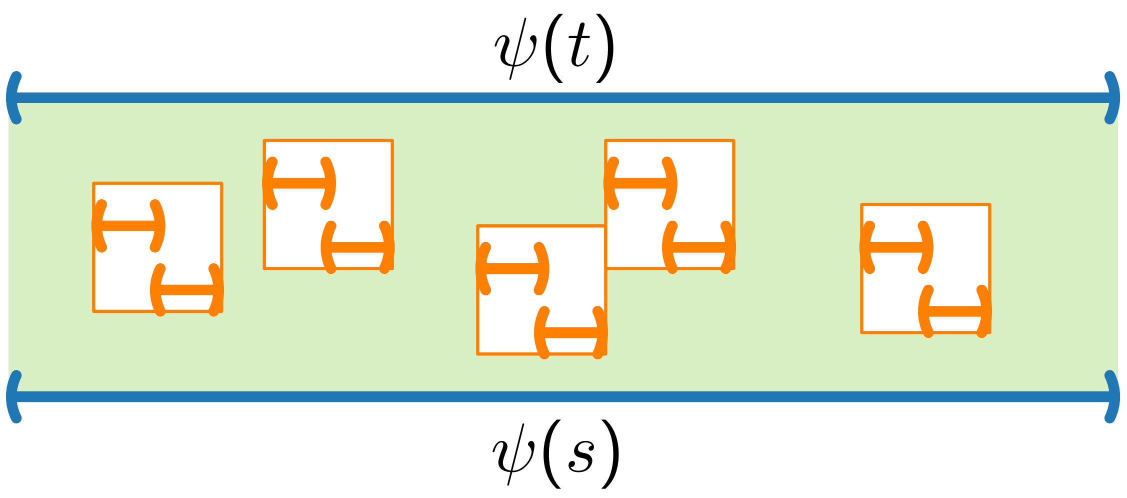


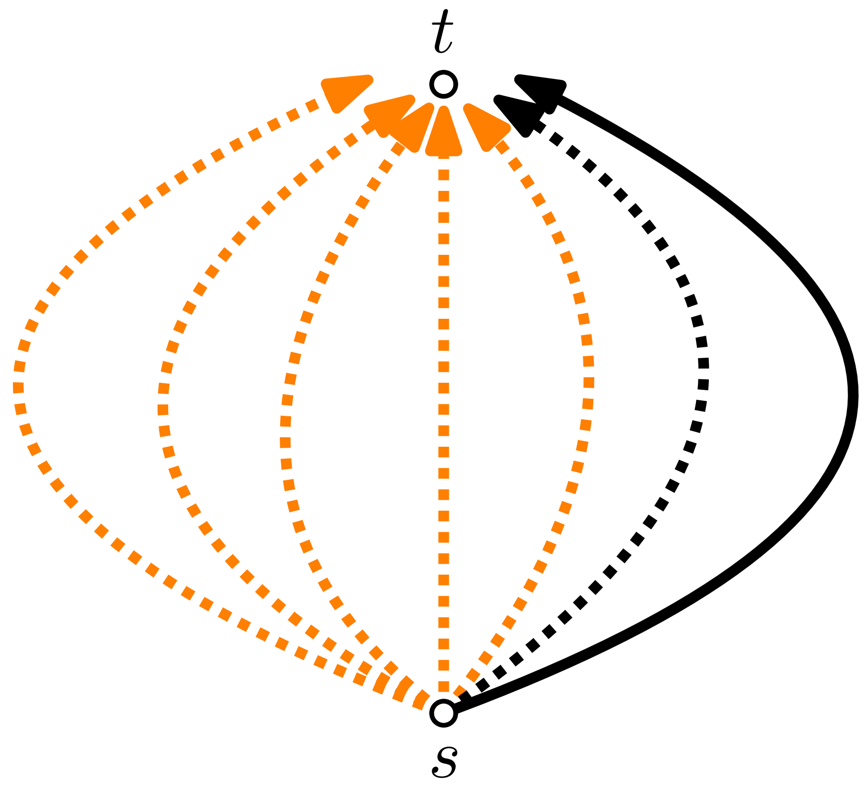
P-Nodes



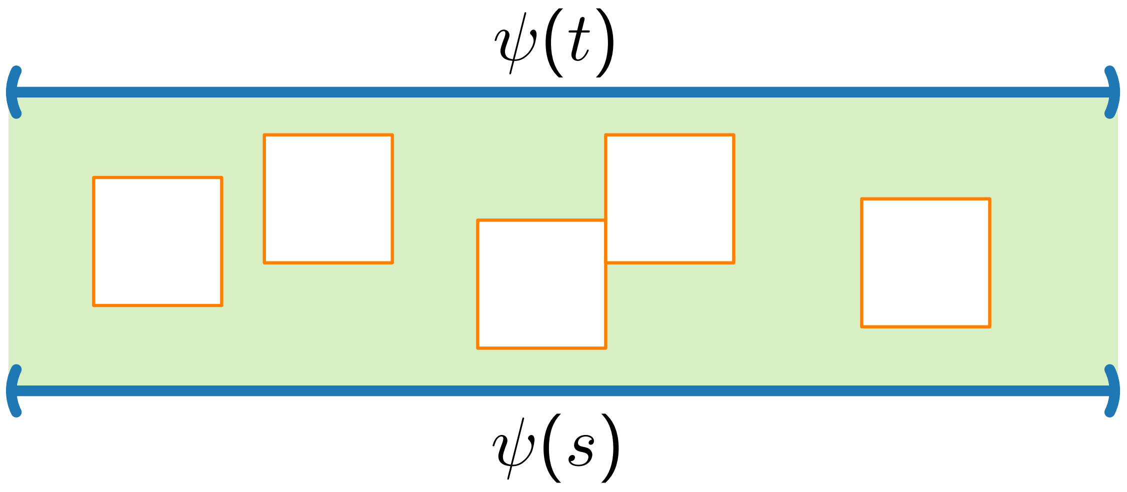


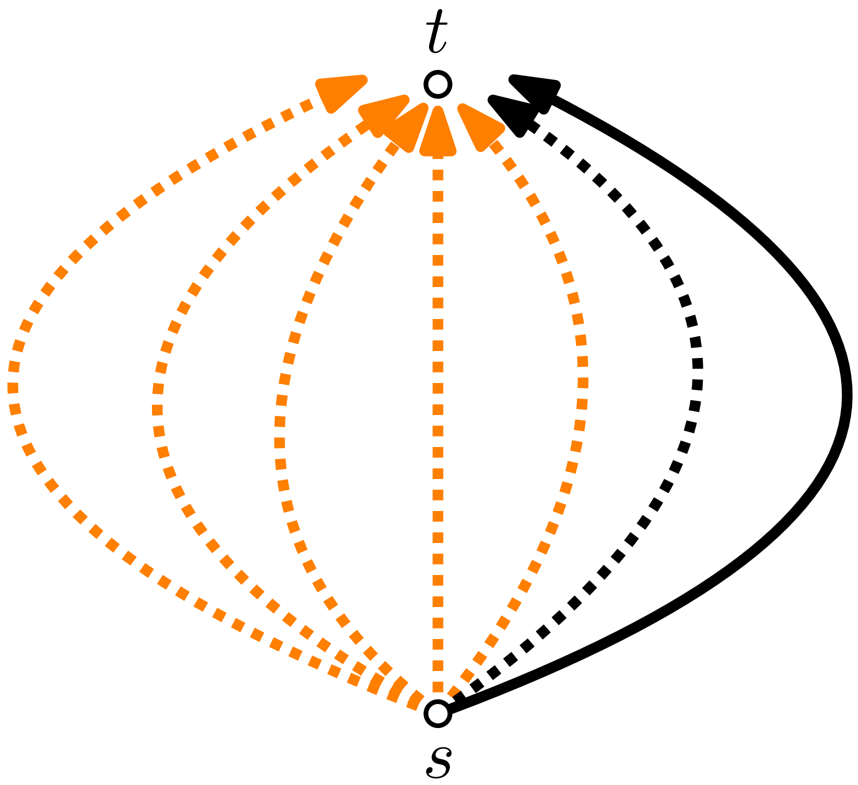
P-Nodes



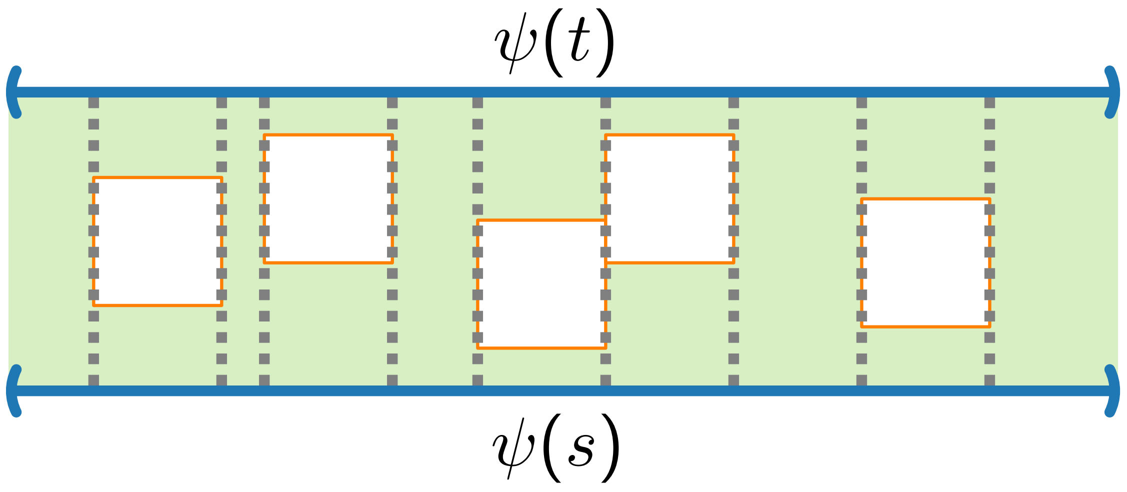


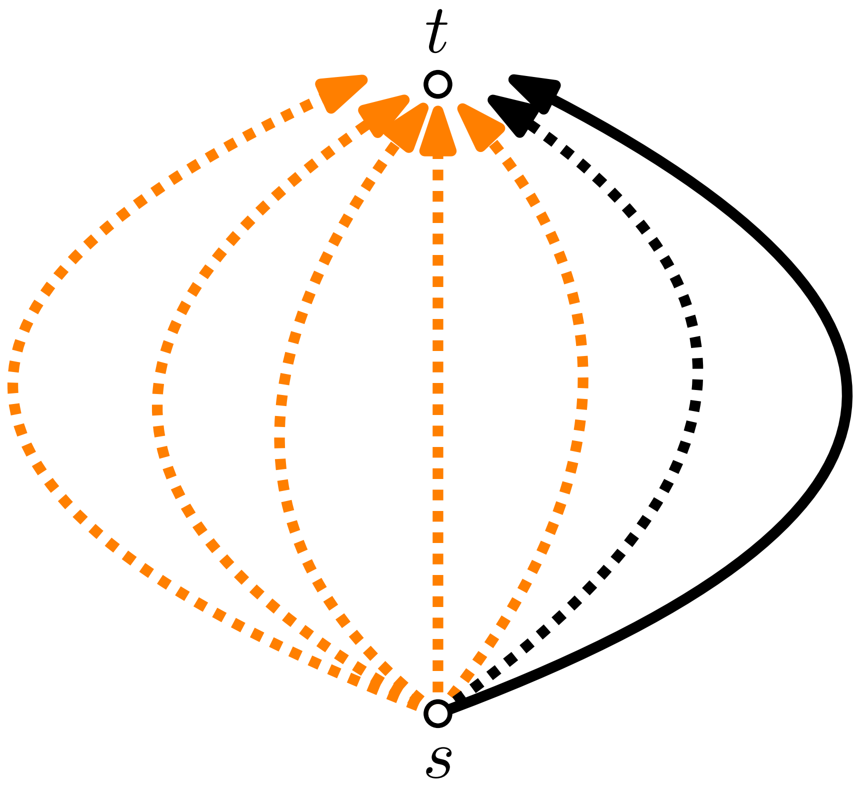
P-Nodes



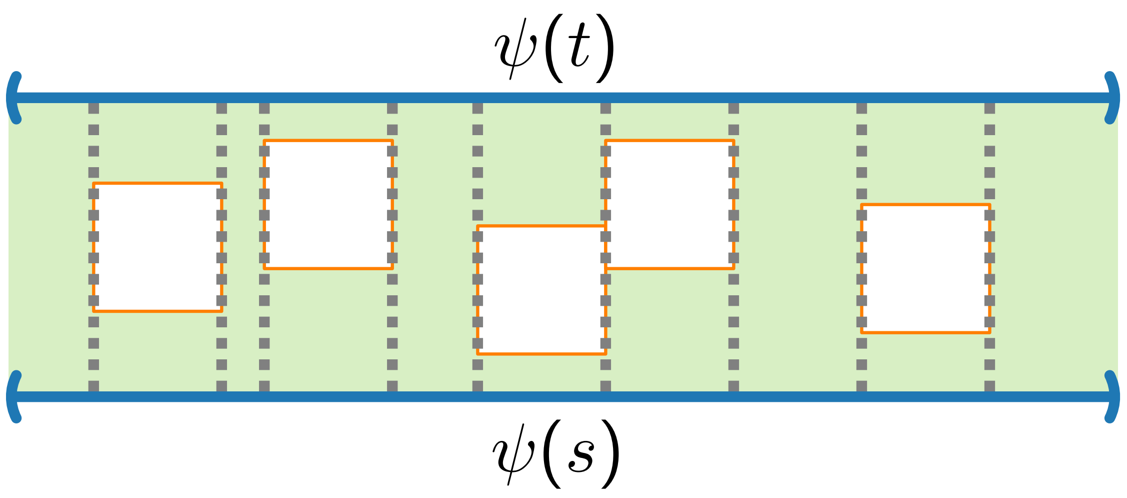


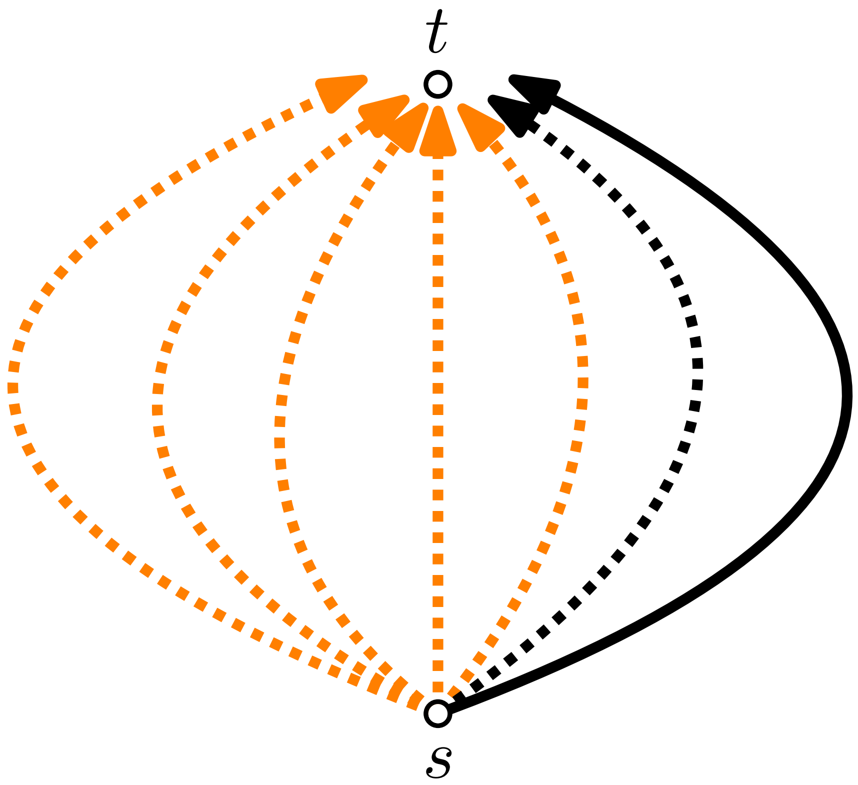
P-Nodes





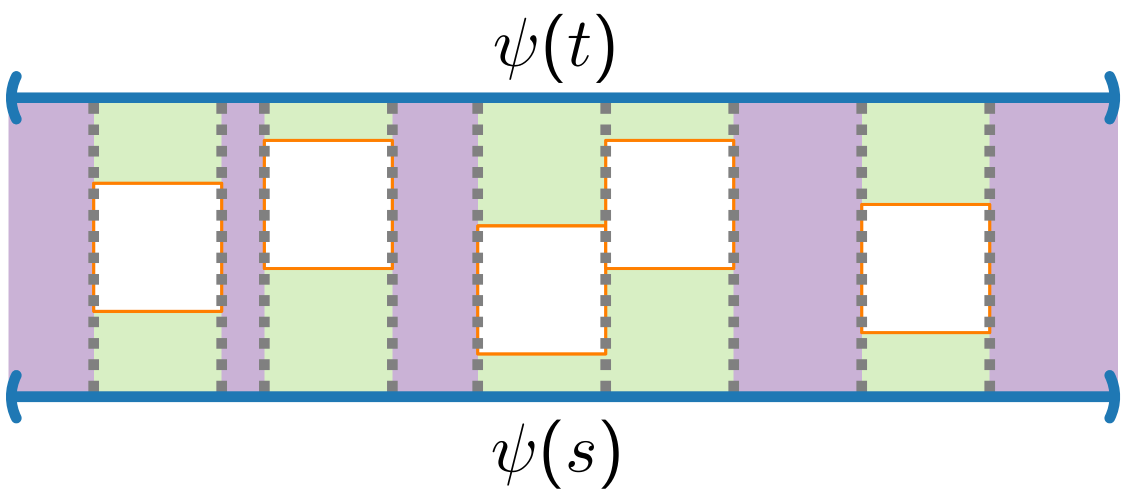
P-Nodes

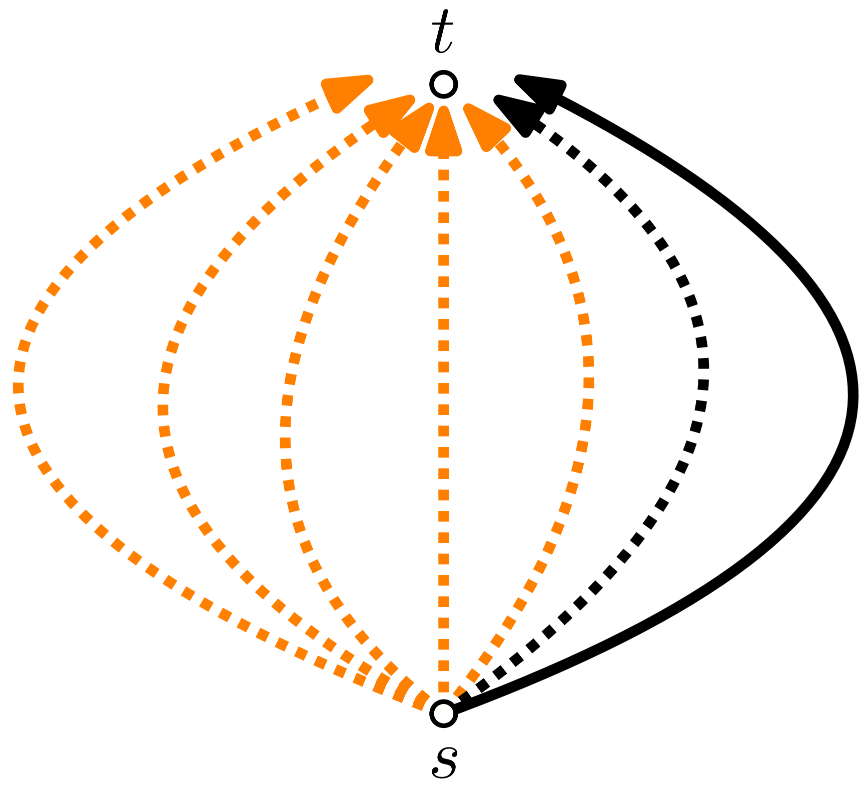




- Children of **P**-node with prescribed bars occur in given left-to-right order

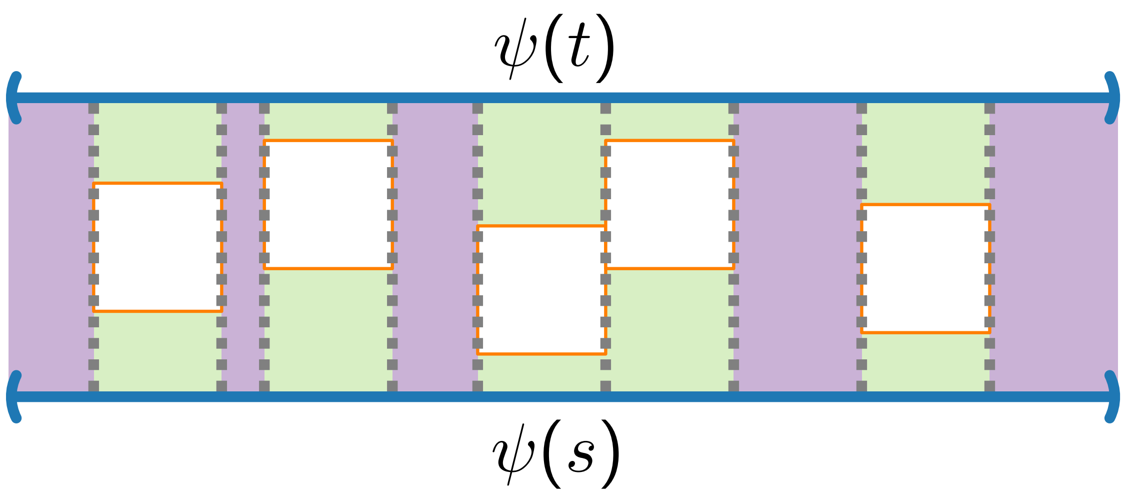
P-Nodes

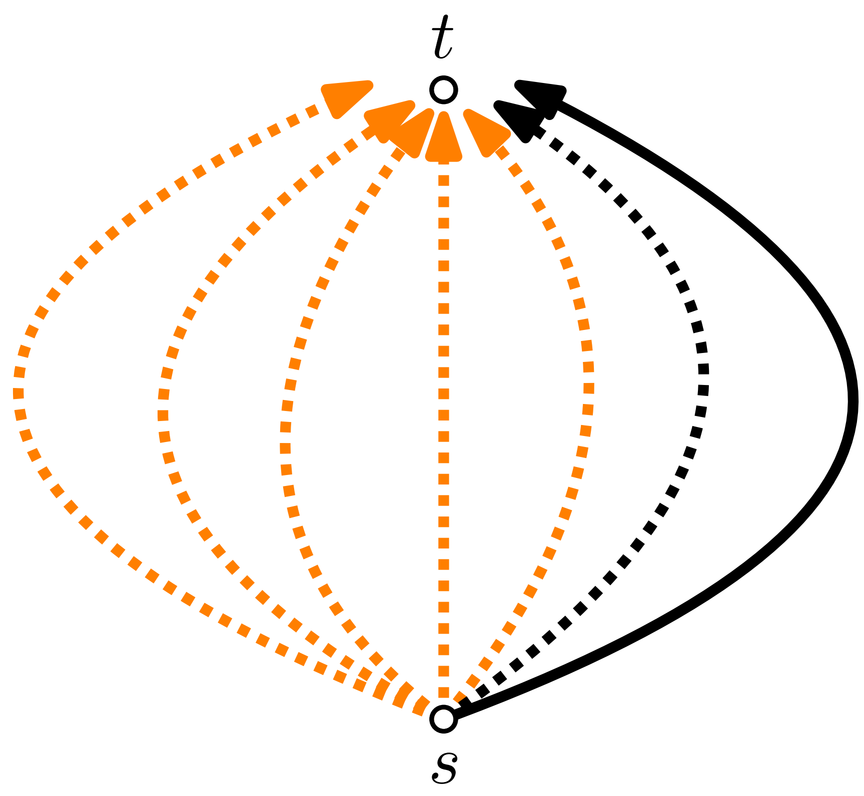




- Children of **P**-node with prescribed bars occur in given left-to-right order
- But there might be some gaps. . .

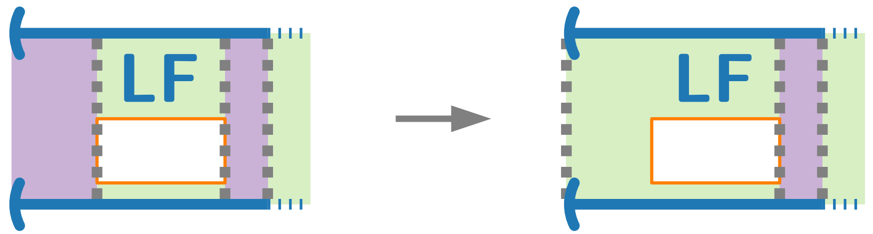
P-Nodes



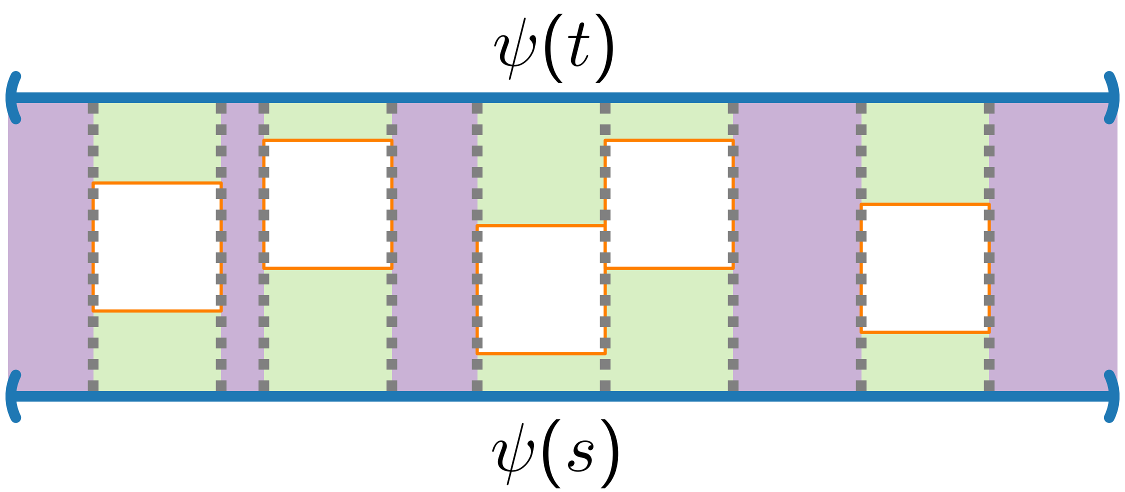


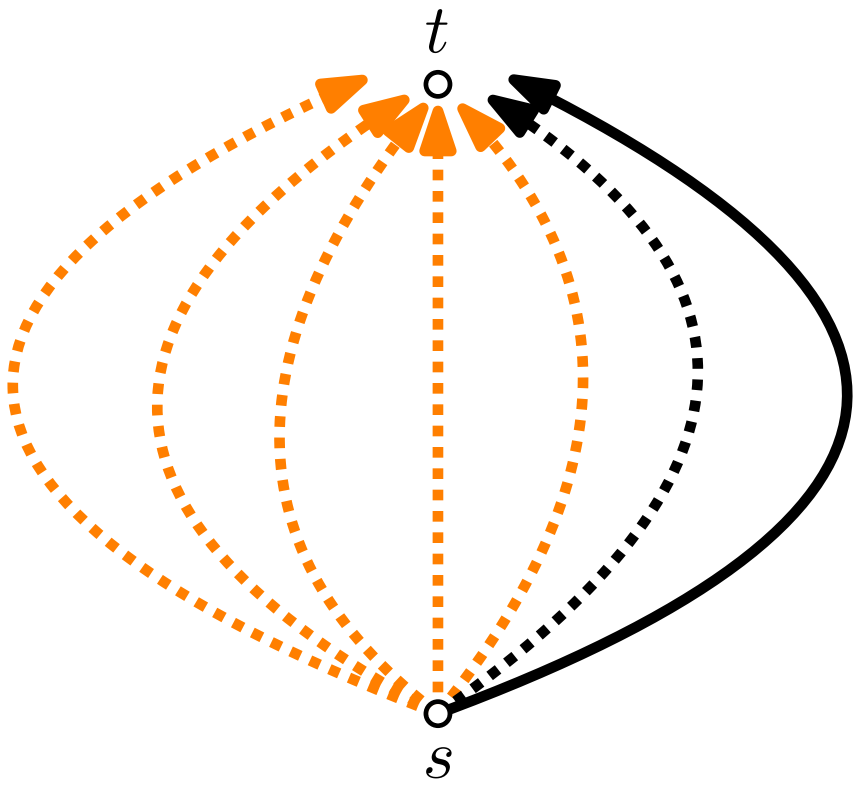
- Children of **P**-node with prescribed bars occur in given left-to-right order
- But there might be some gaps. . .

Idea.
Greedily *fill* the gaps by preferring to "stretch" the children with prescribed bars.



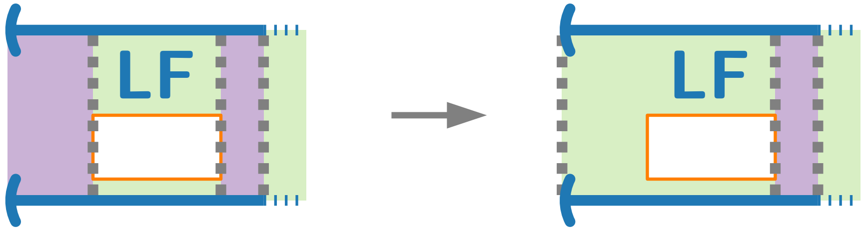
P-Nodes





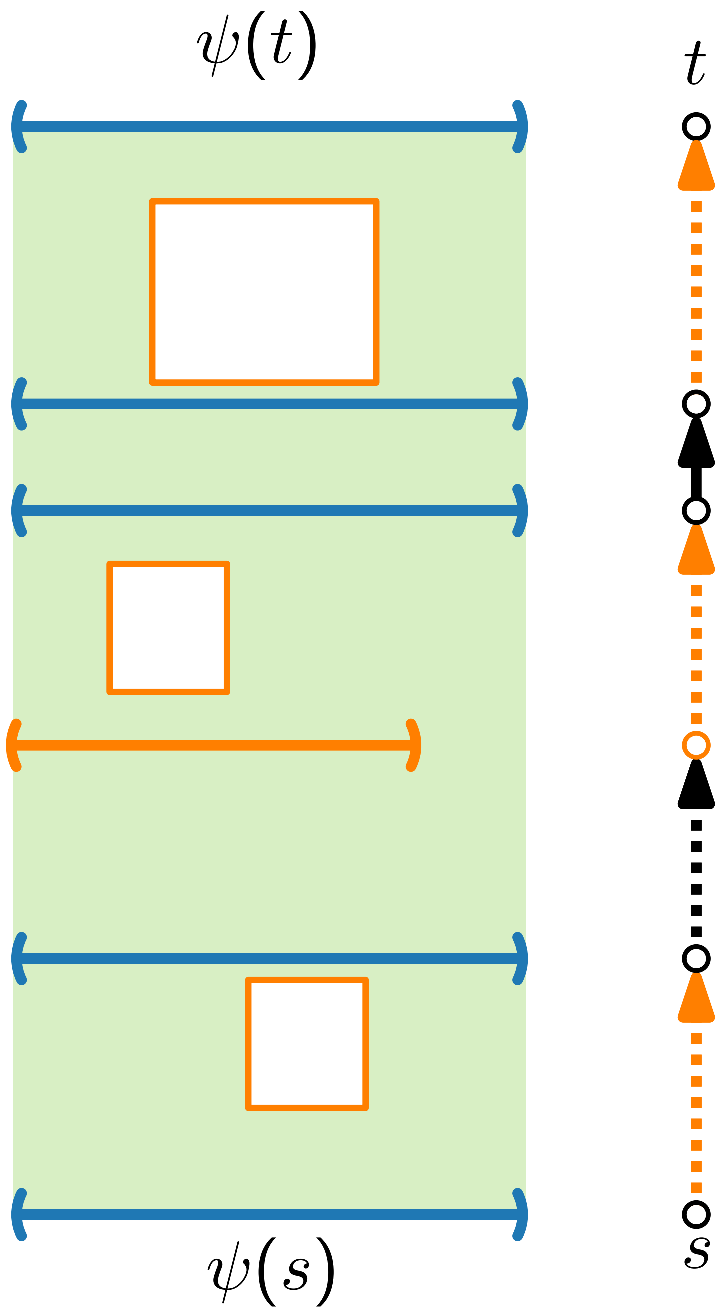
- Children of **P**-node with prescribed bars occur in given left-to-right order
- But there might be some gaps. . .

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

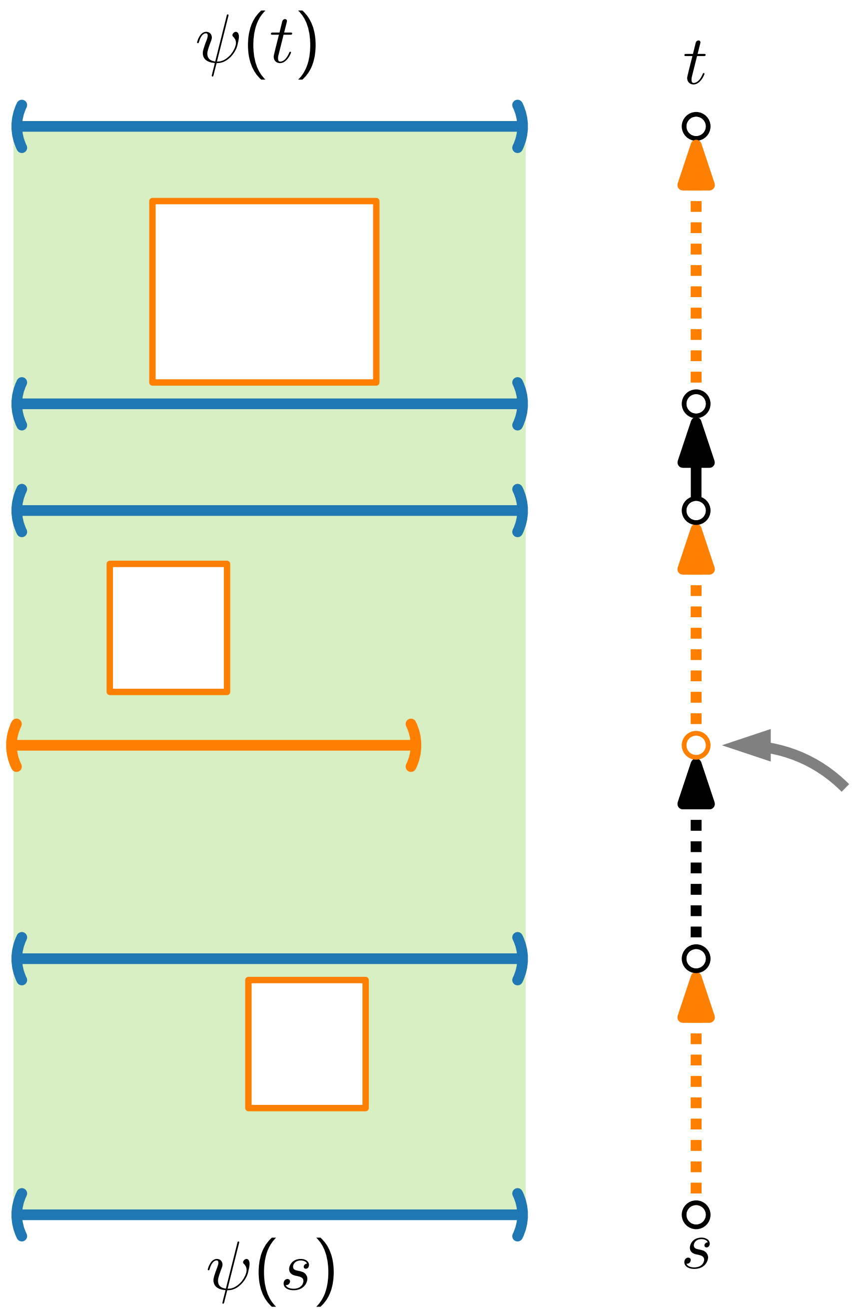


Outcome.
After processing, we must know the valid types for the corresponding subgraphs.

S-Nodes

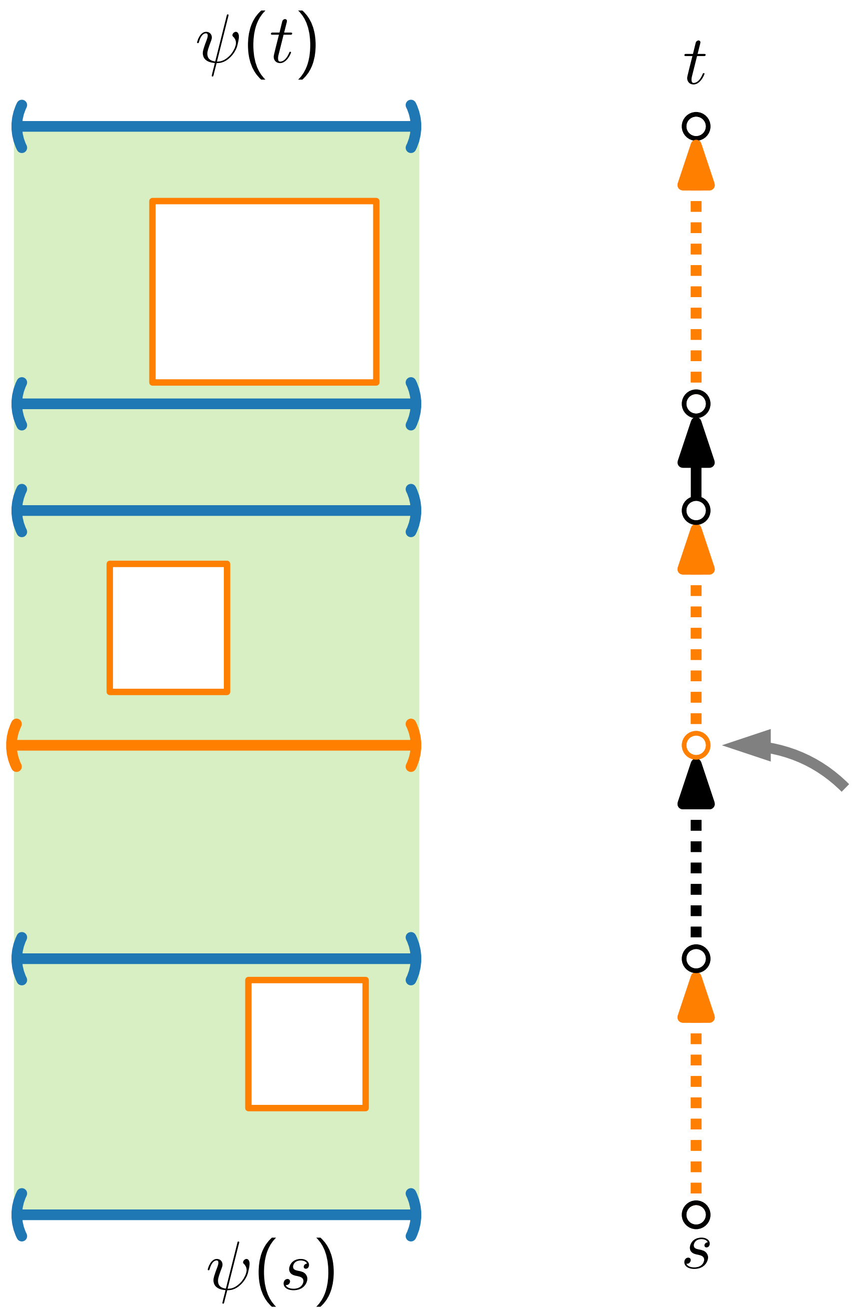


S-Nodes



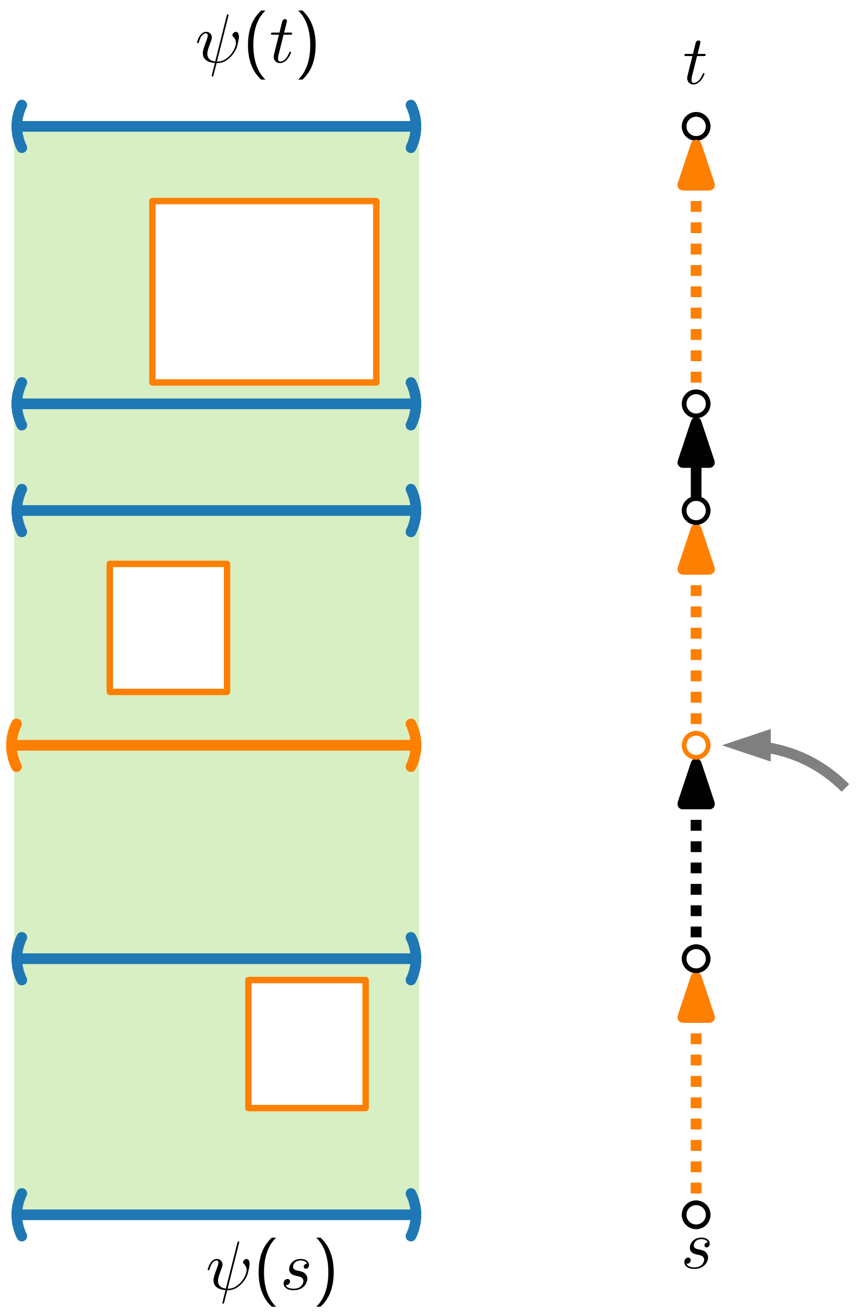
This fixed vertex means we can only make a Fixed-Fixed representation!

S-Nodes

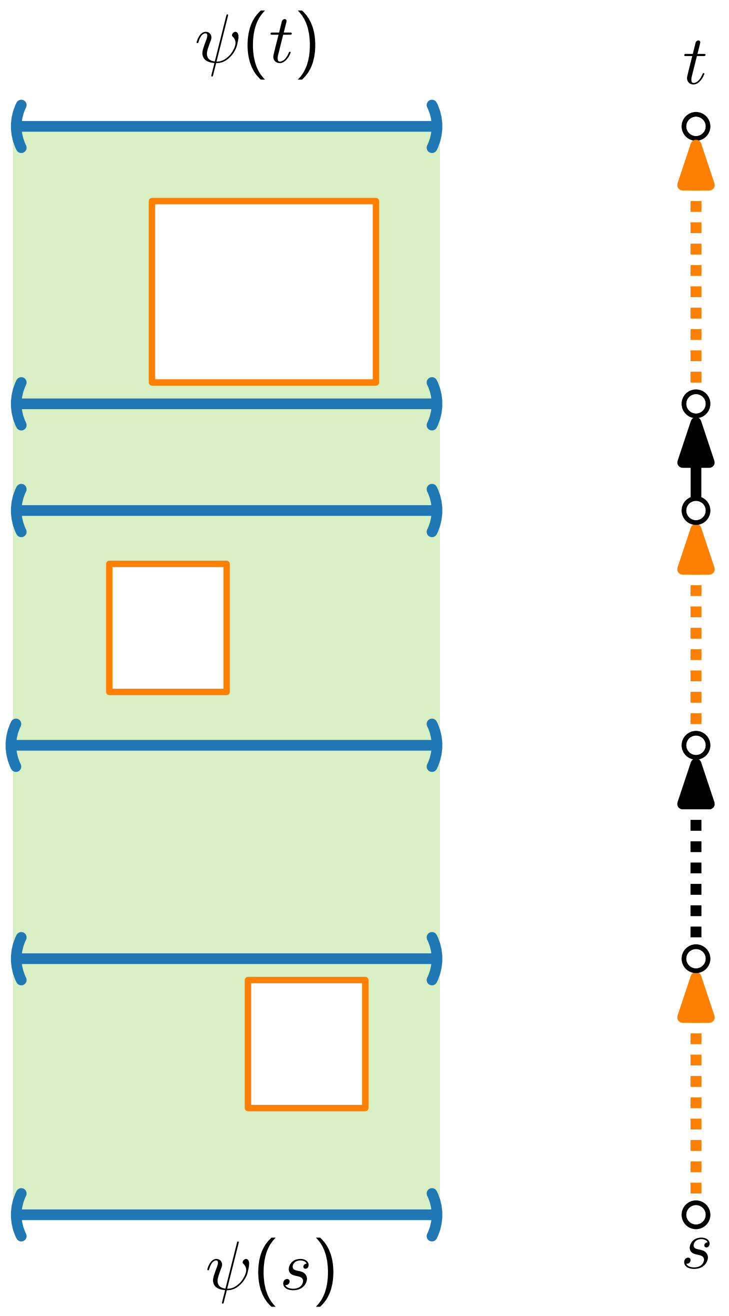


This fixed vertex means we can only make a Fixed-Fixed representation!

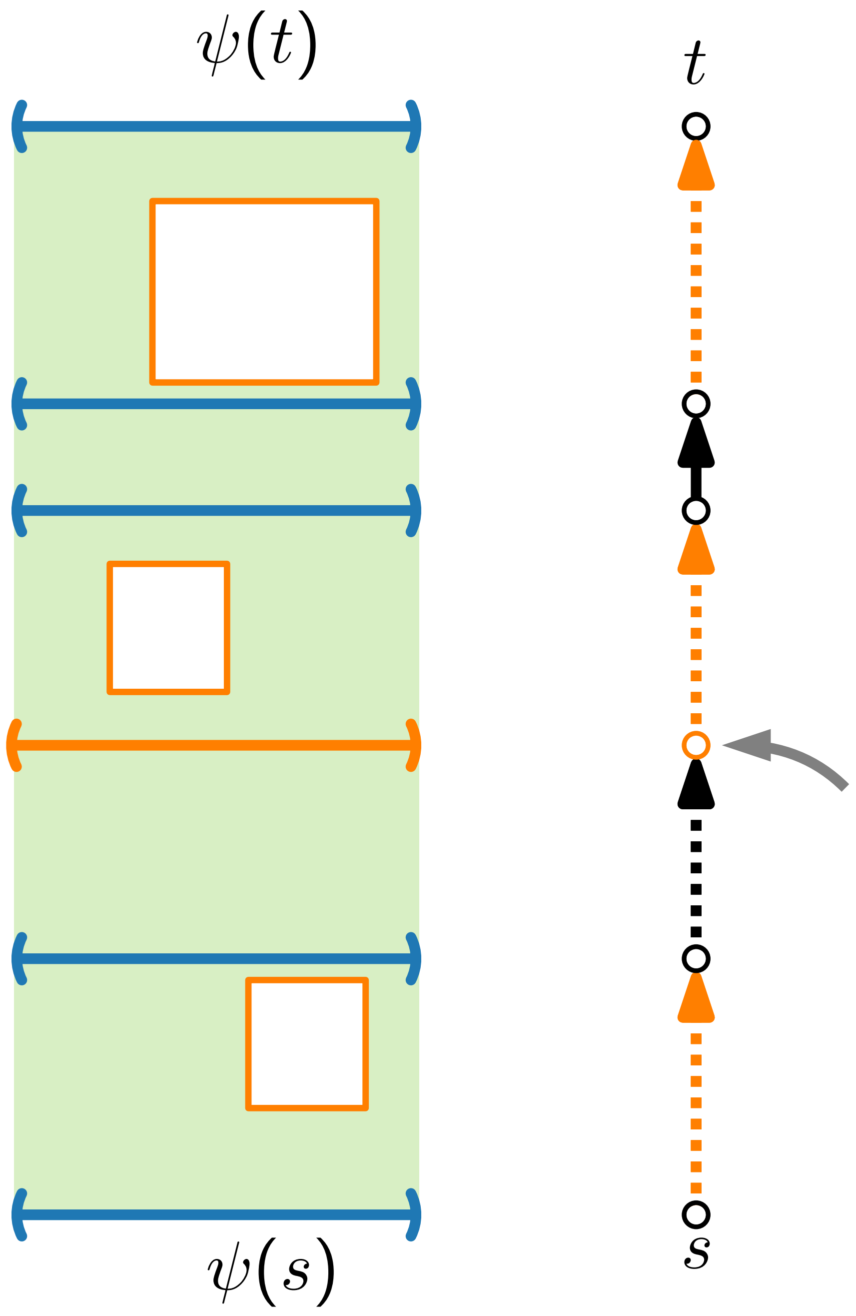
S-Nodes



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

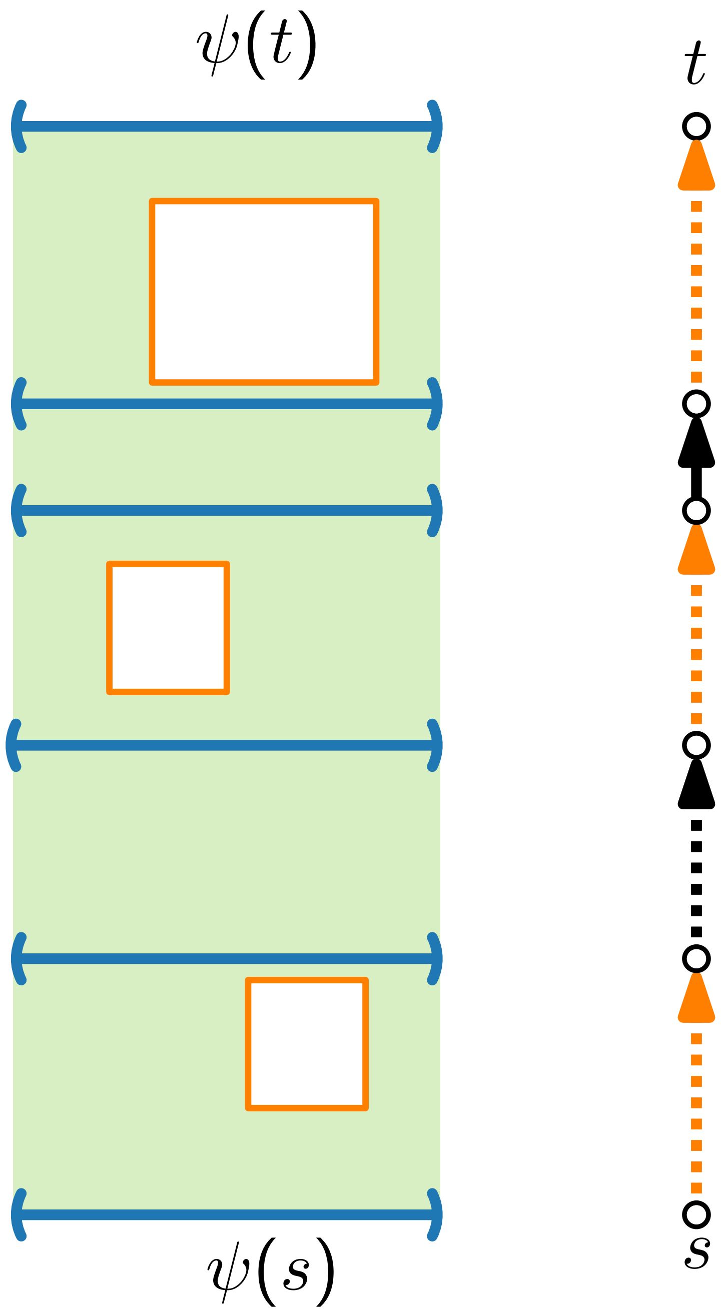


S-Nodes

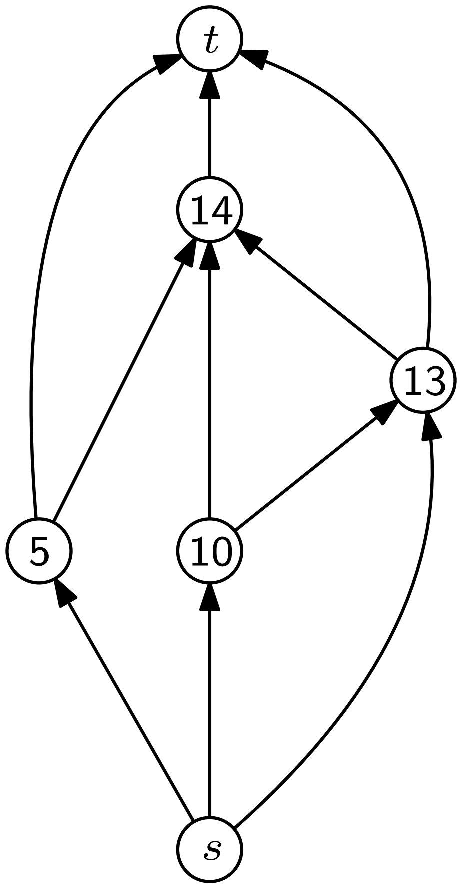


Here we have a chance to make all (**LL, FL, LF, FF**) types.

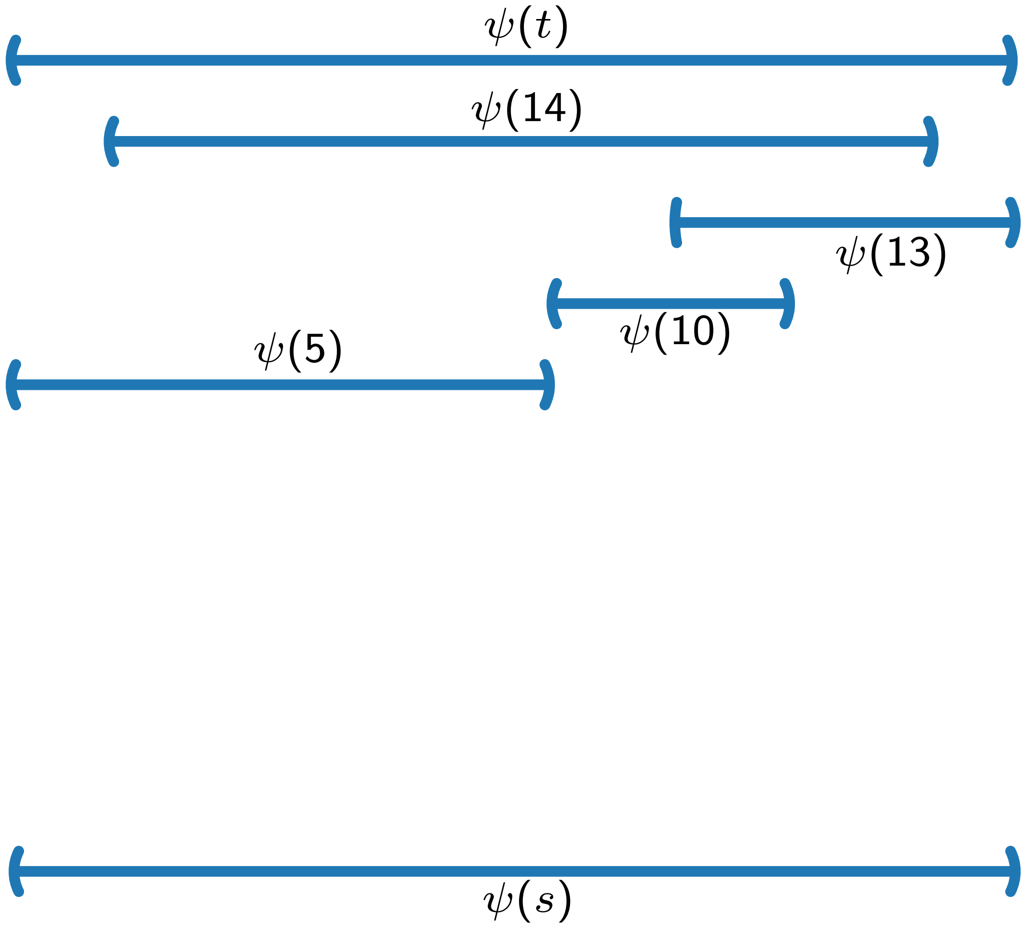
This fixed vertex means we can only make a Fixed-Fixed representation!

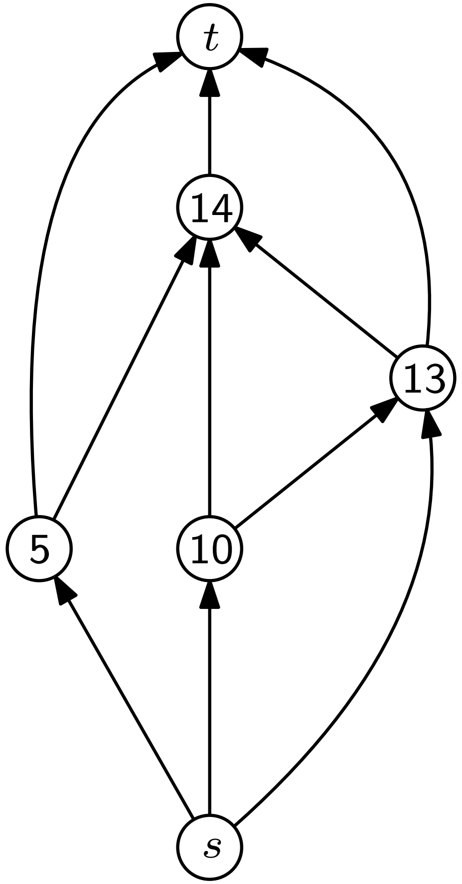


R-Nodes

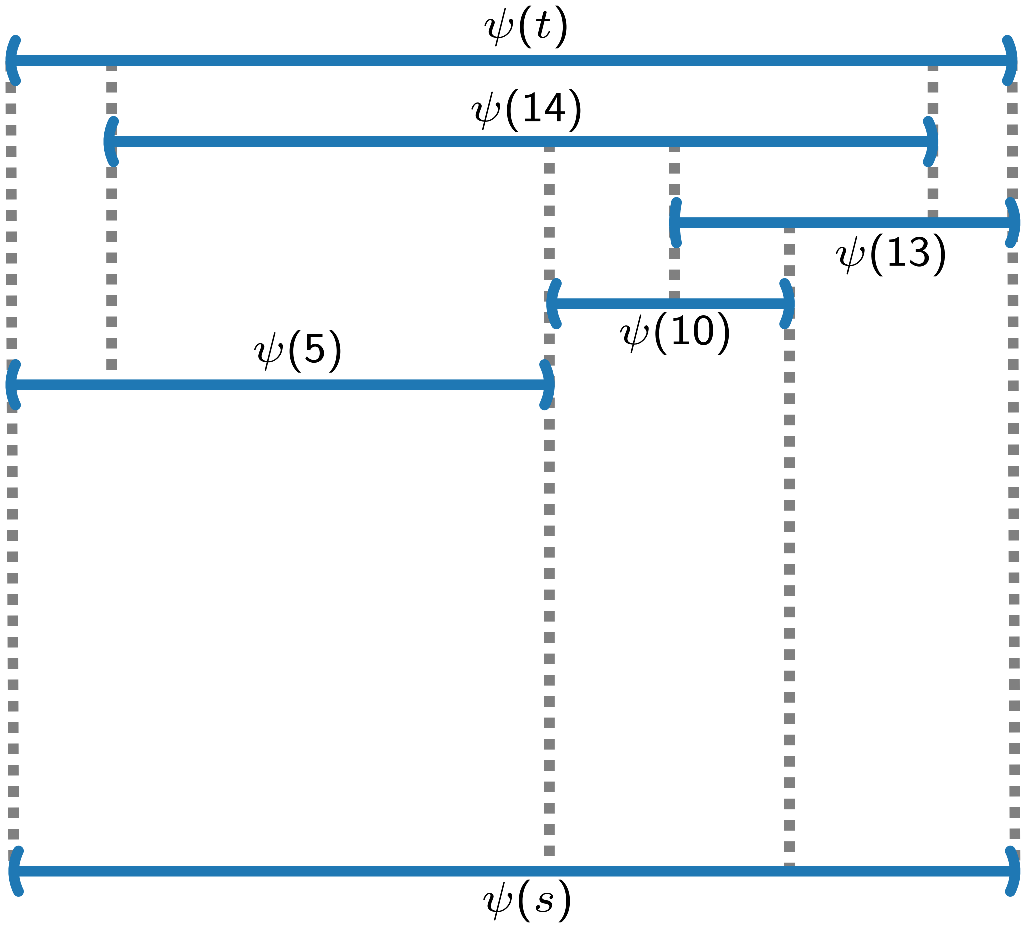


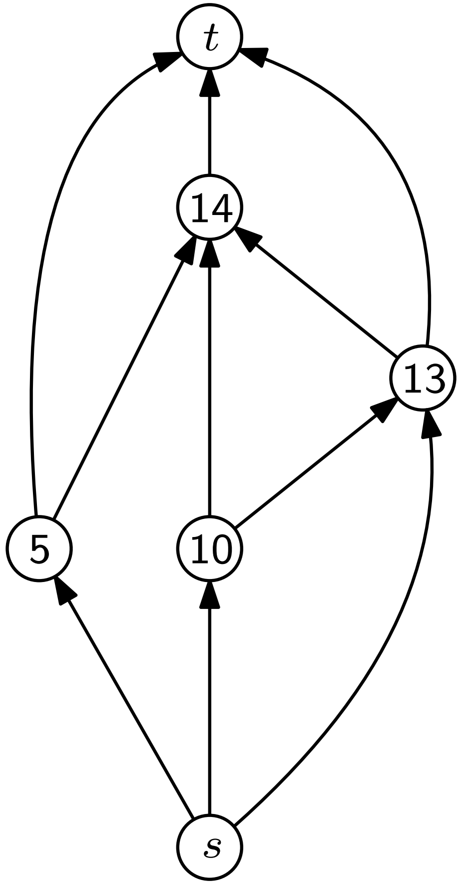
R-Nodes



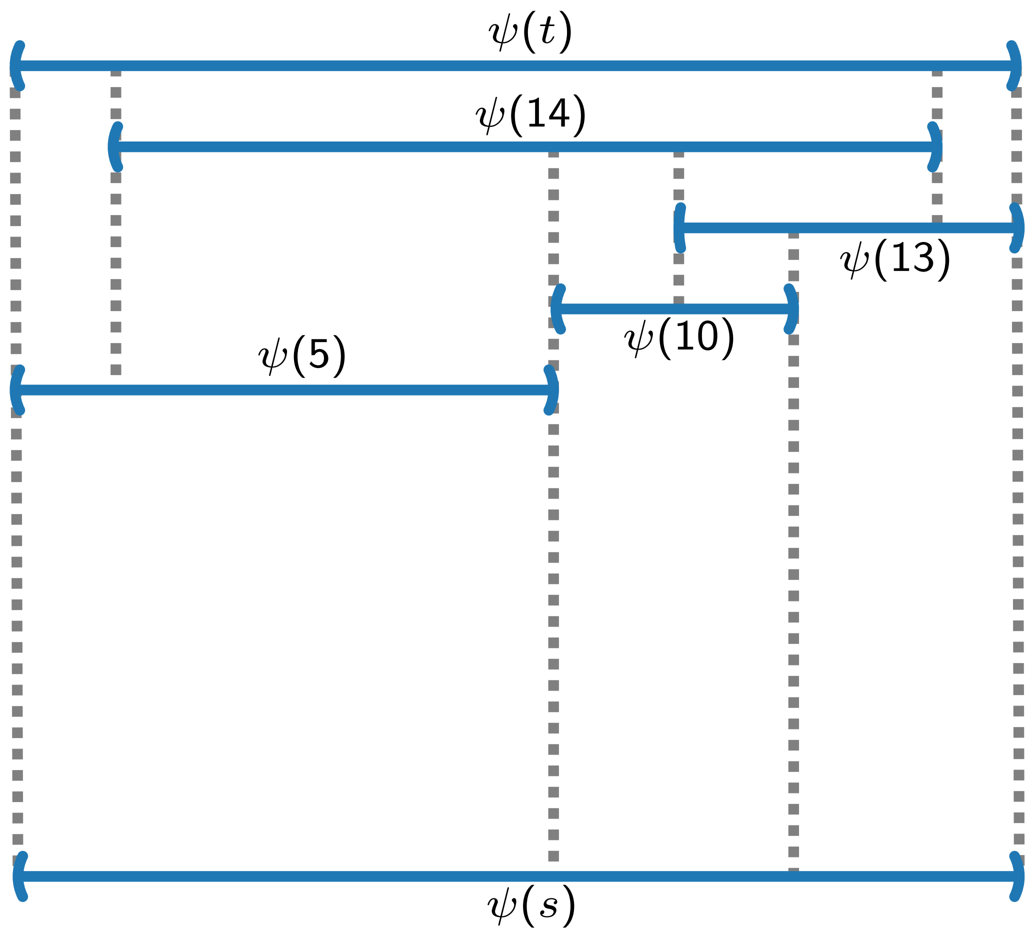


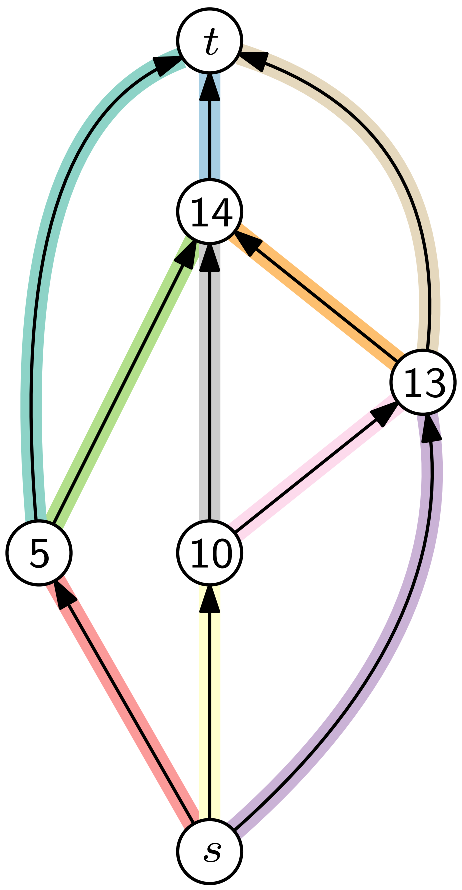
R-Nodes



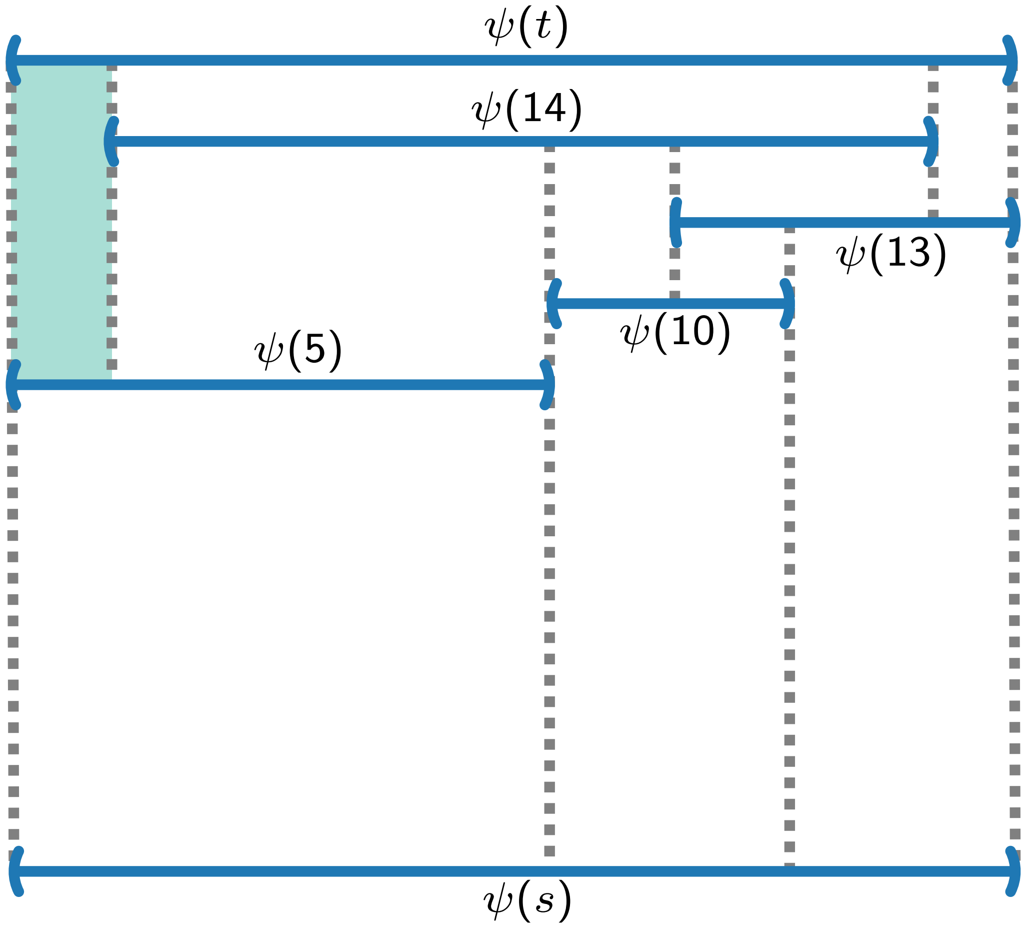


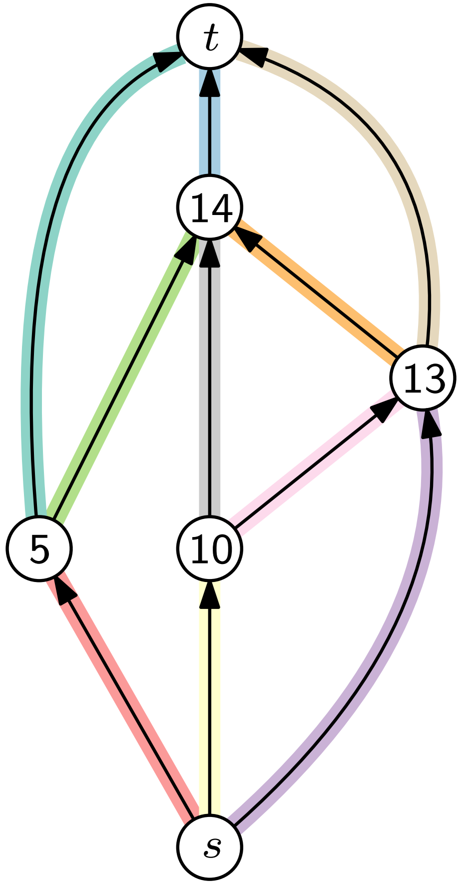
R-Nodes



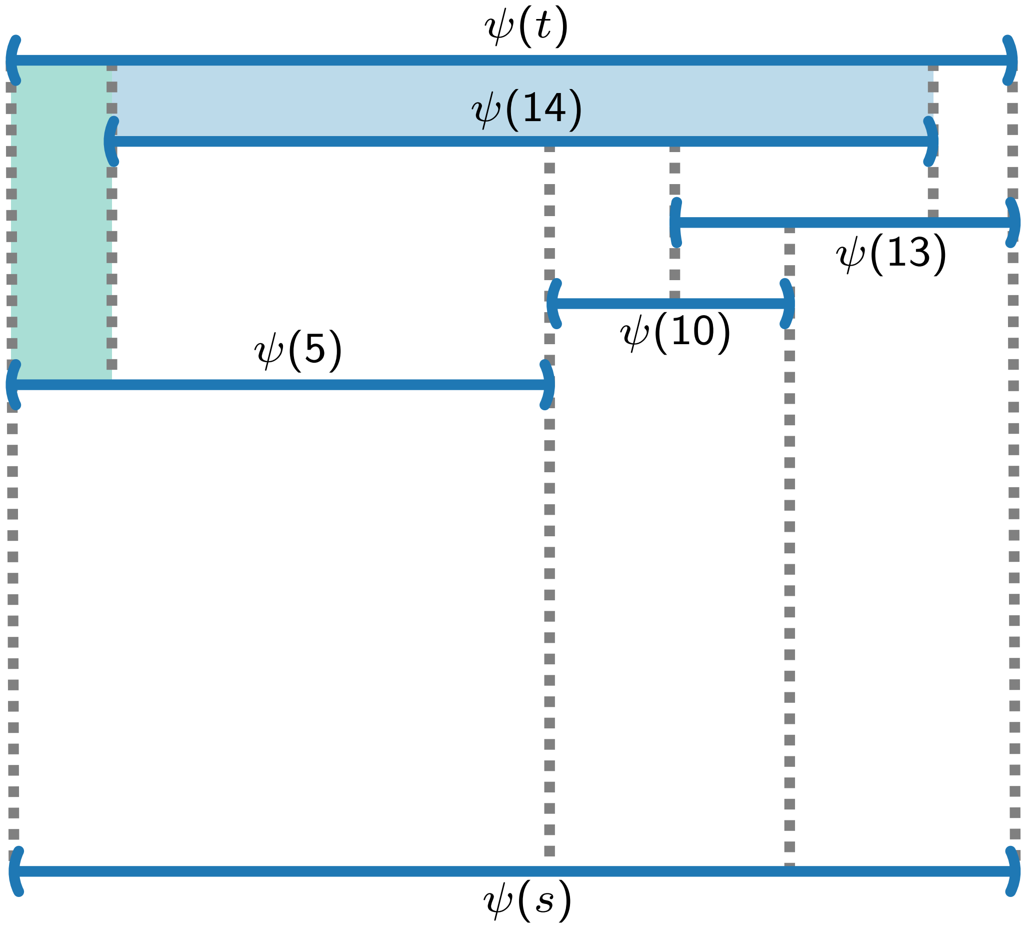


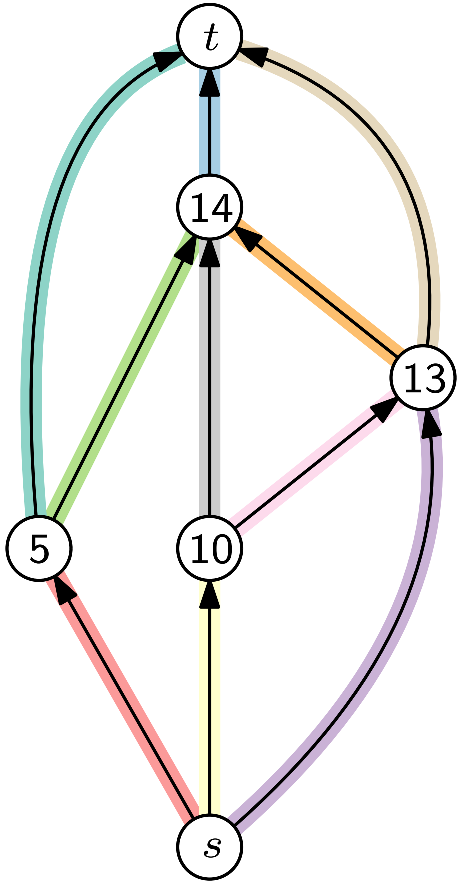
R-Nodes



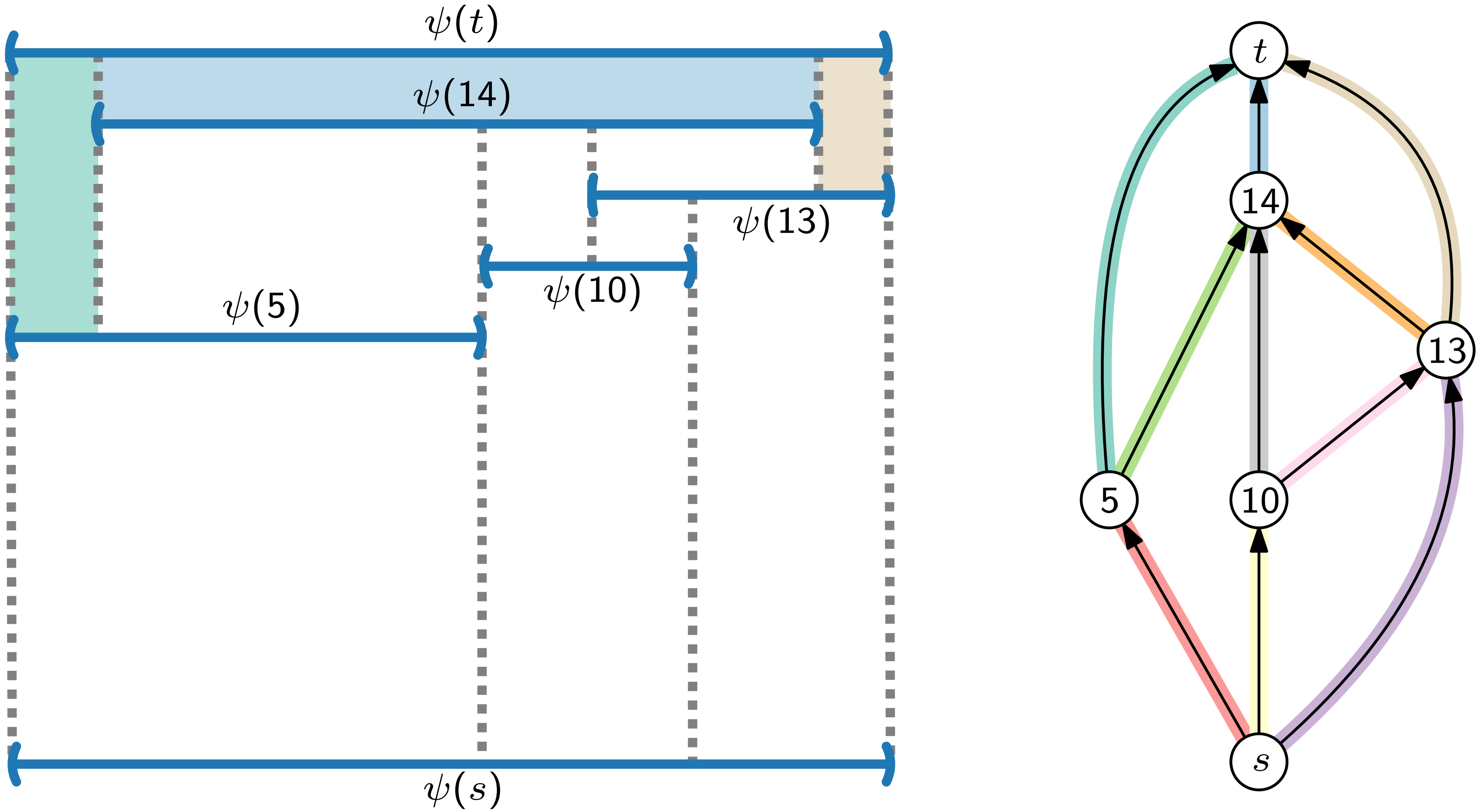


R-Nodes

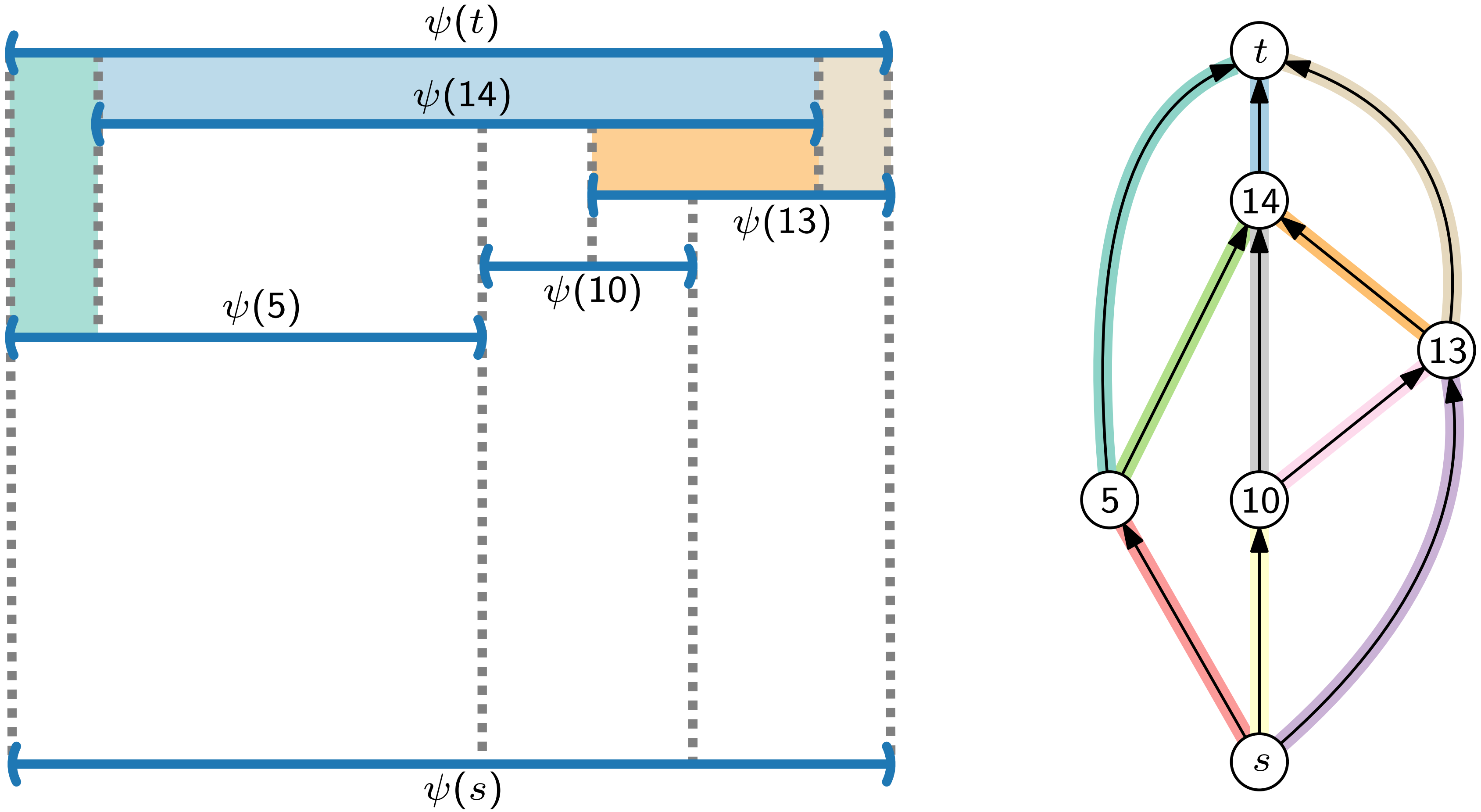




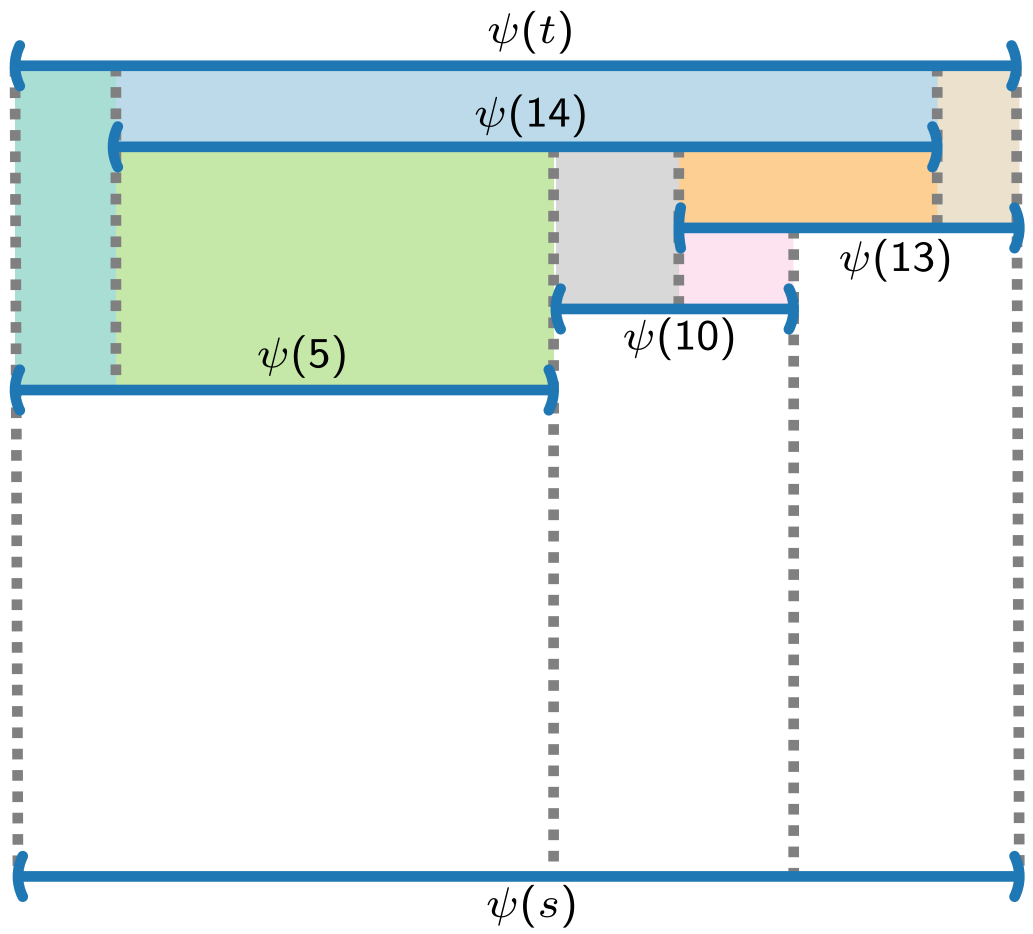
R-Nodes

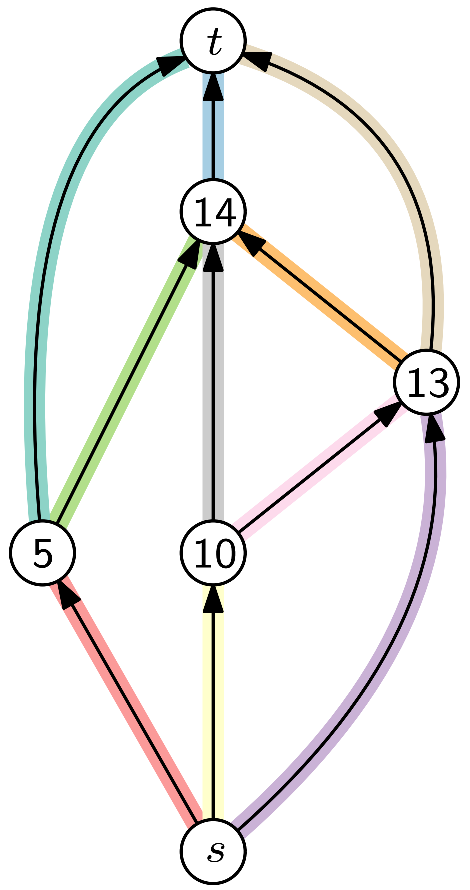


R-Nodes

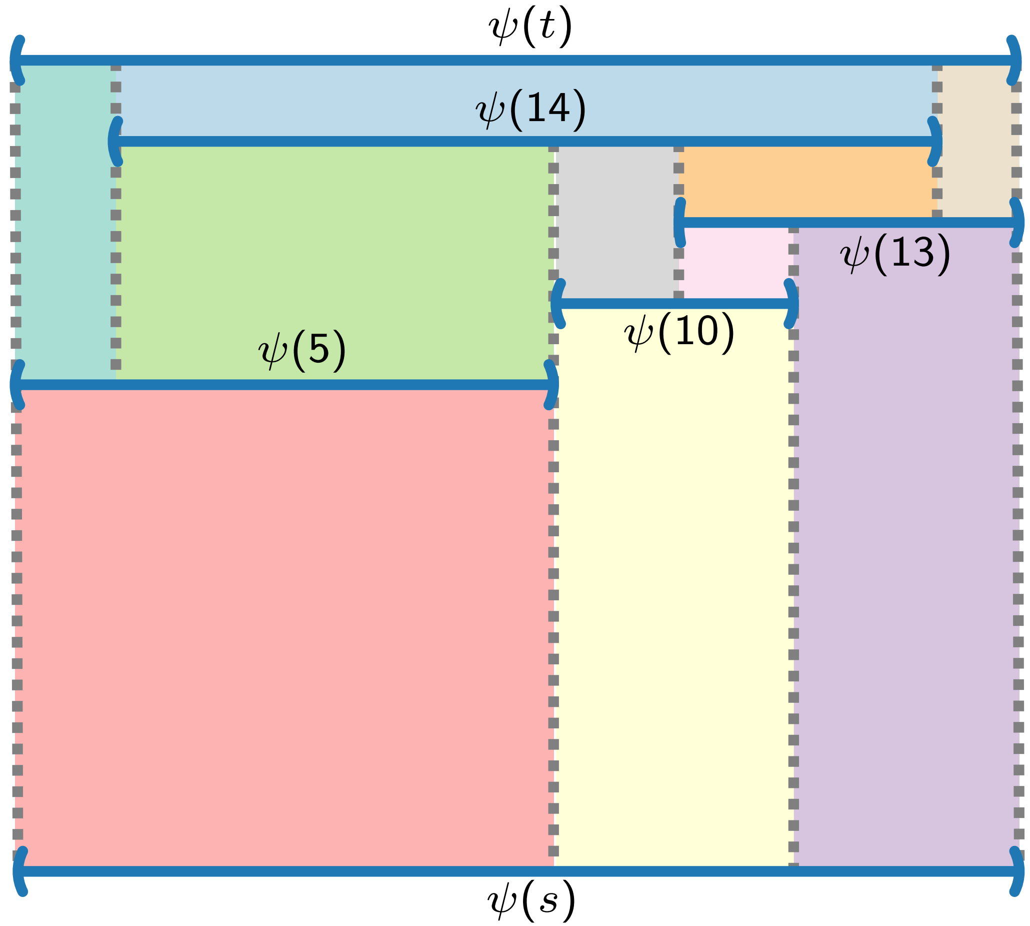


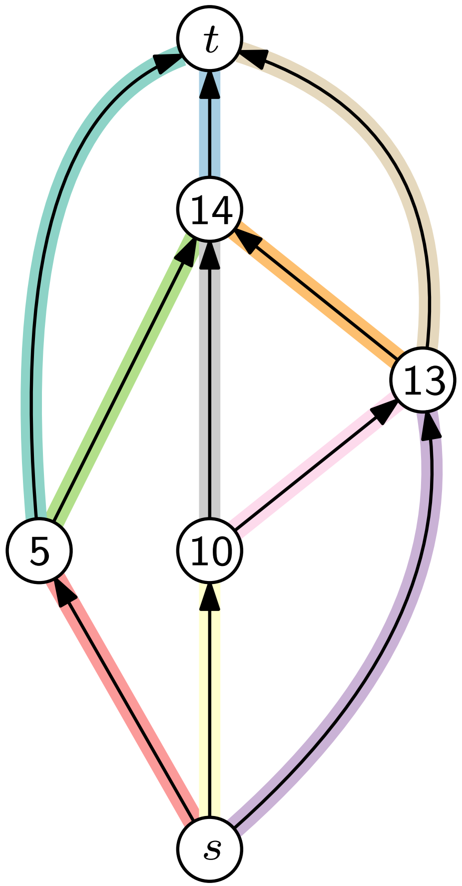
R-Nodes



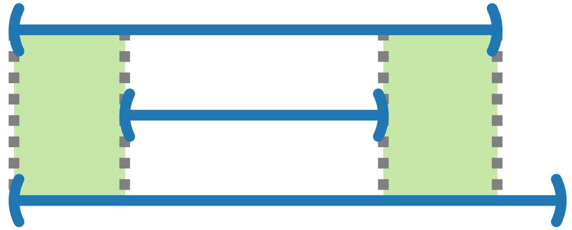


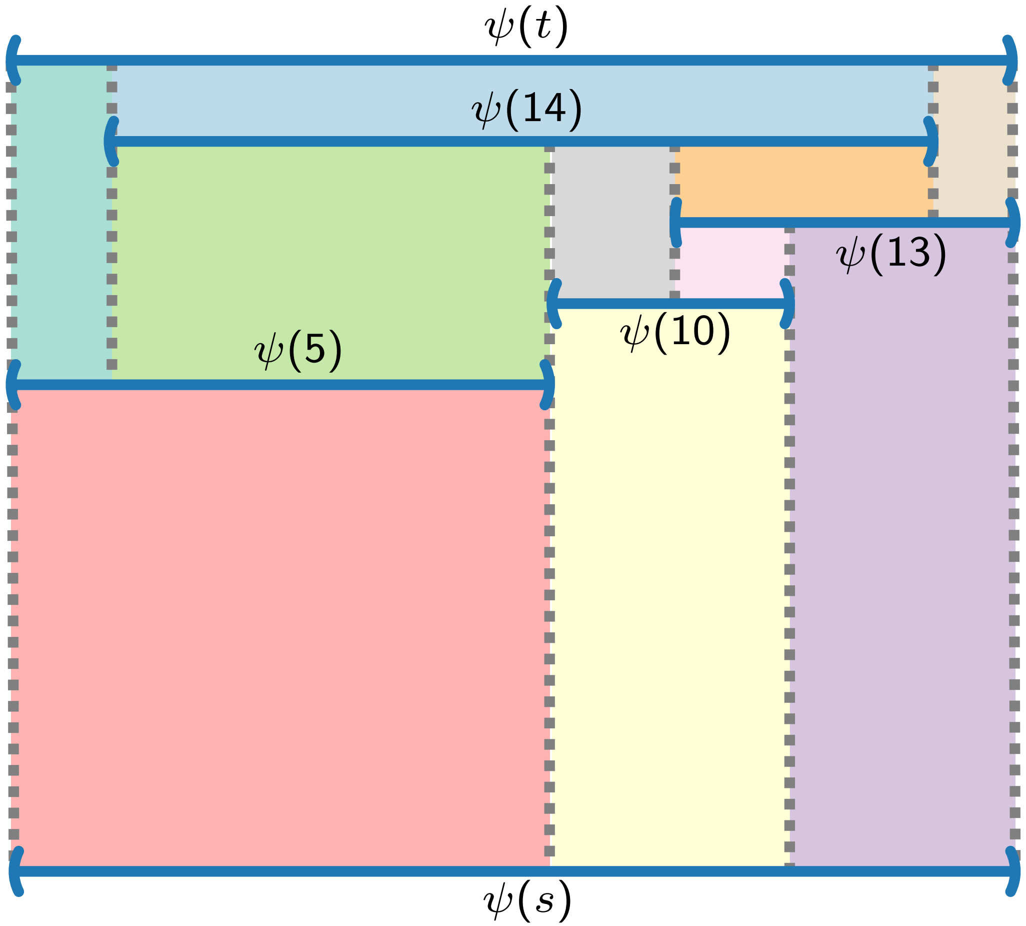
R-Nodes

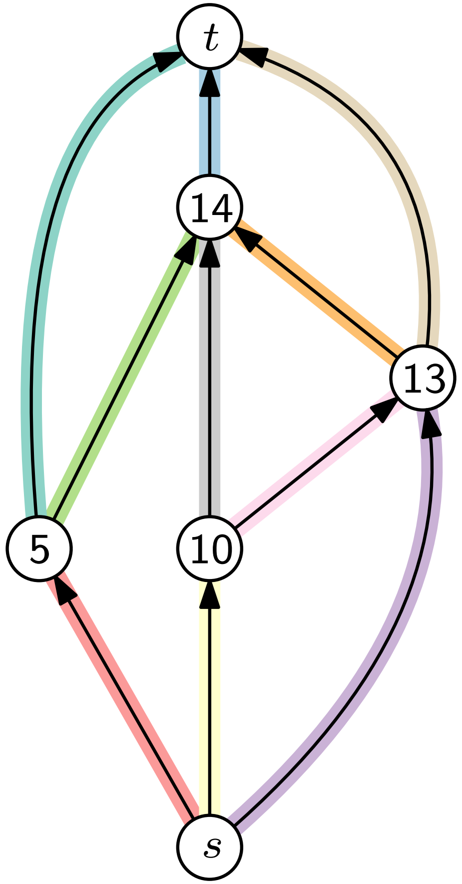




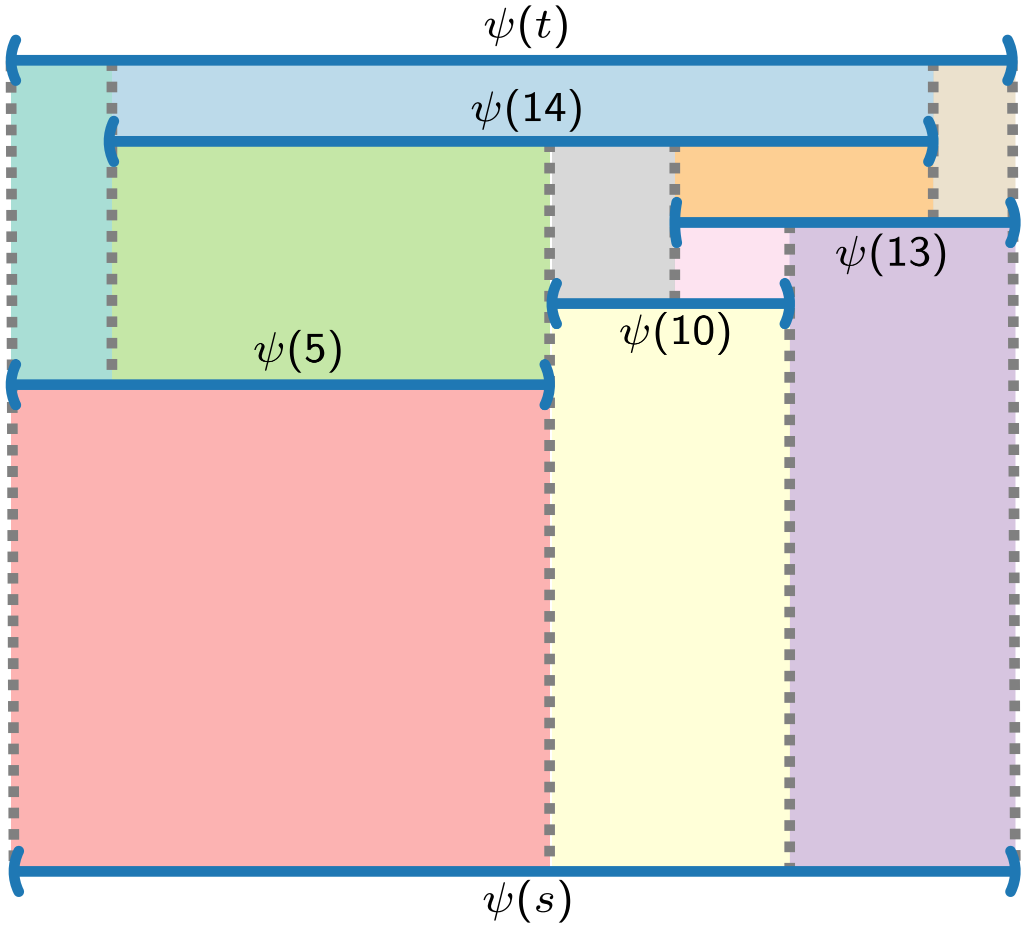
R-Nodes

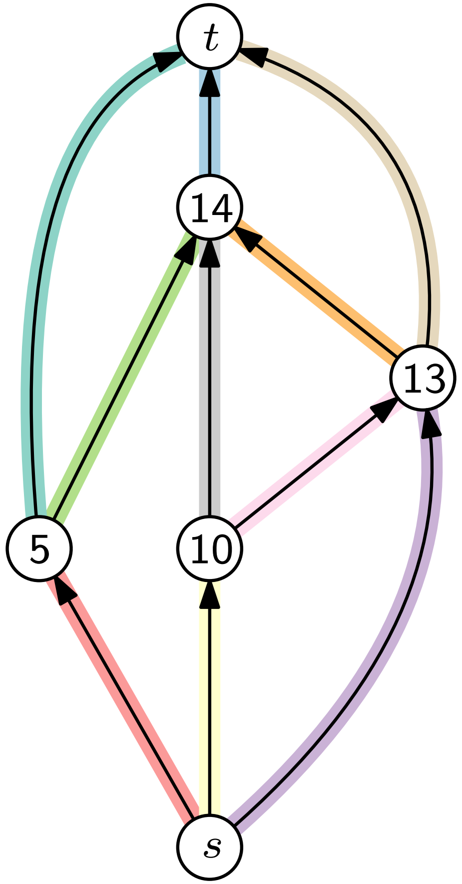


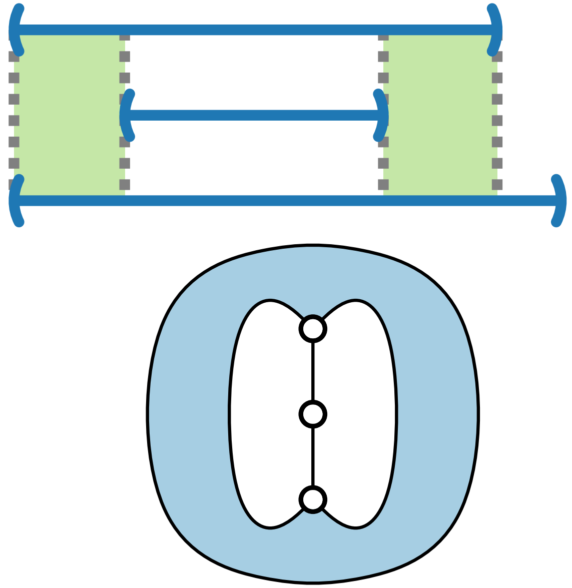




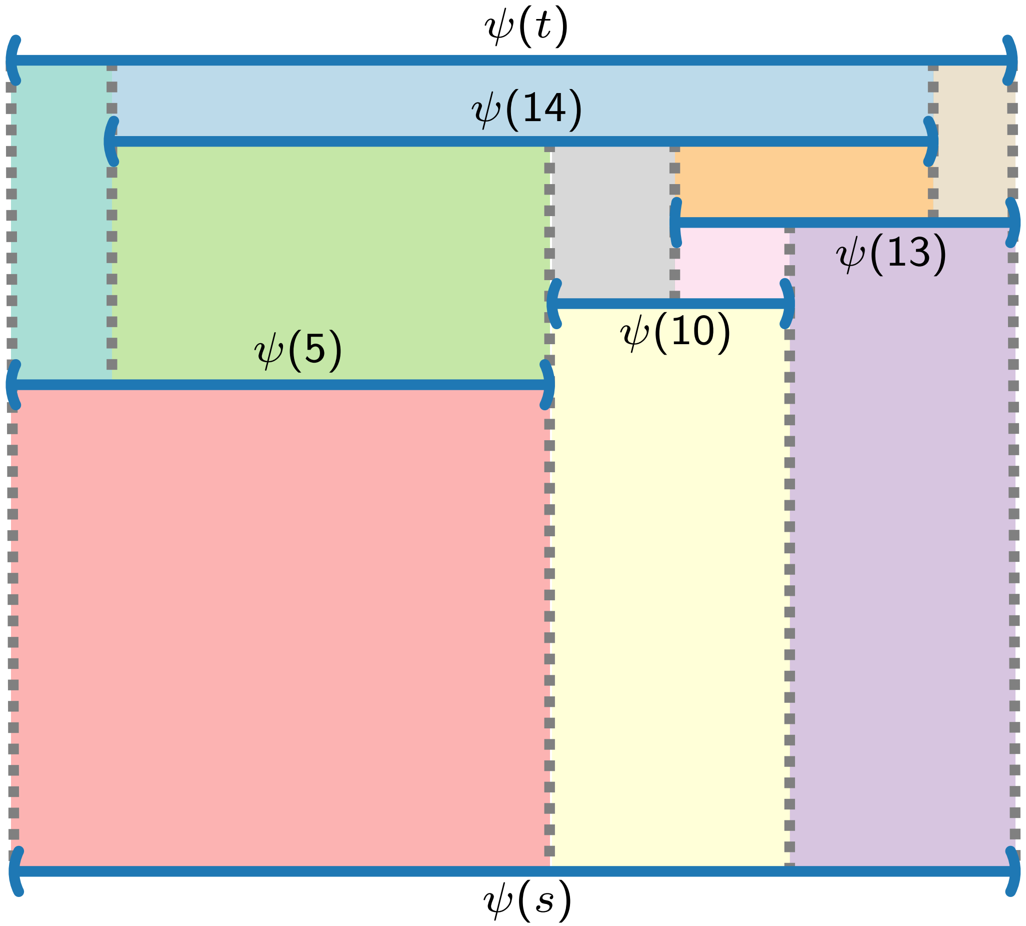
R-Nodes

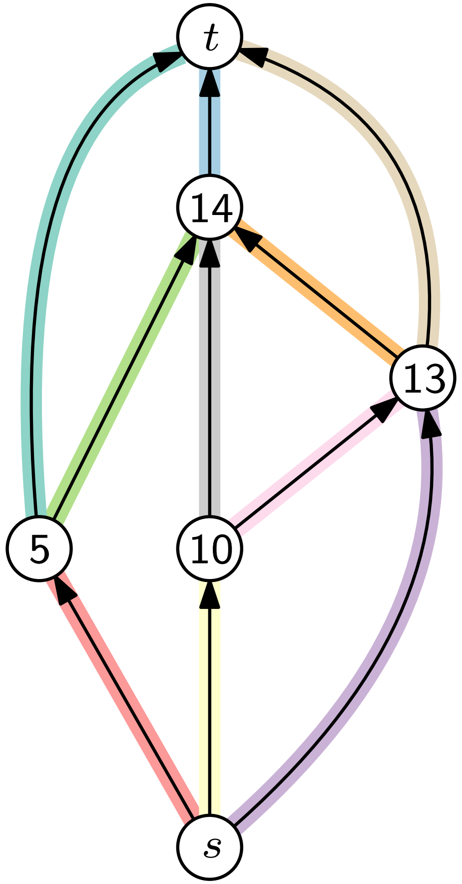


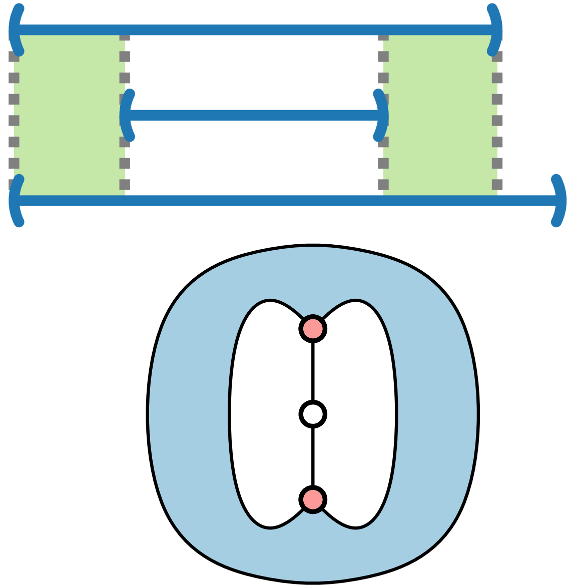




R-Nodes

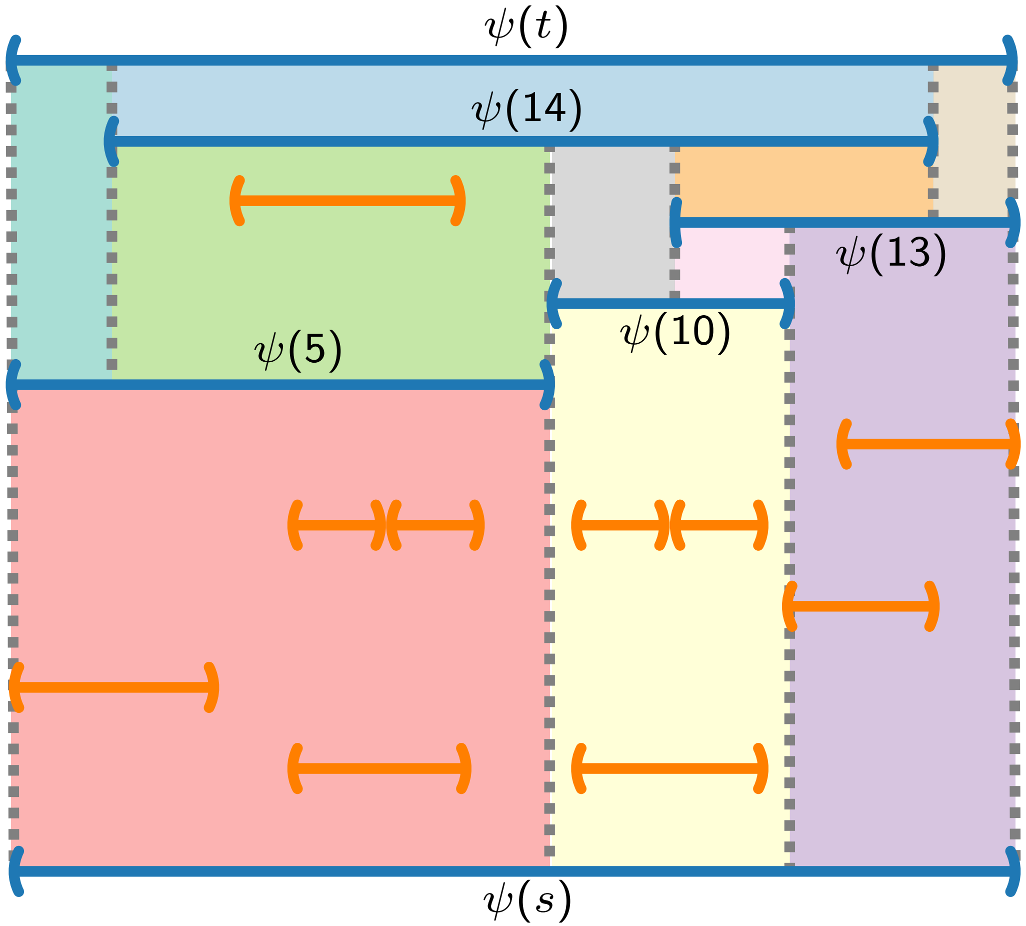


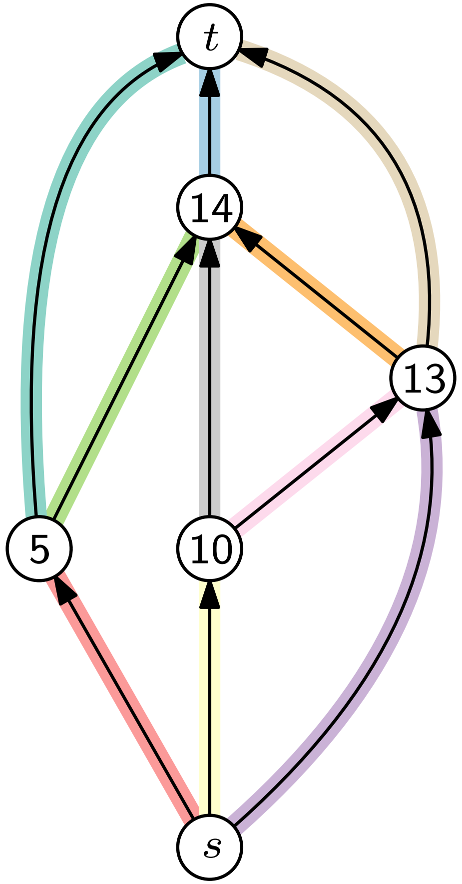


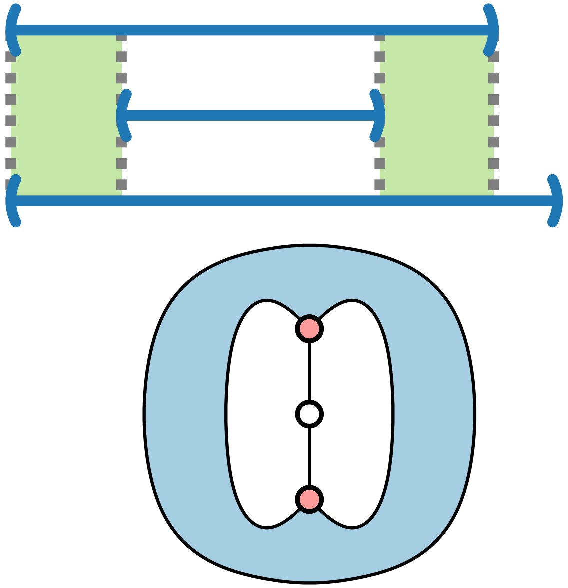


separation pair!

R-Nodes



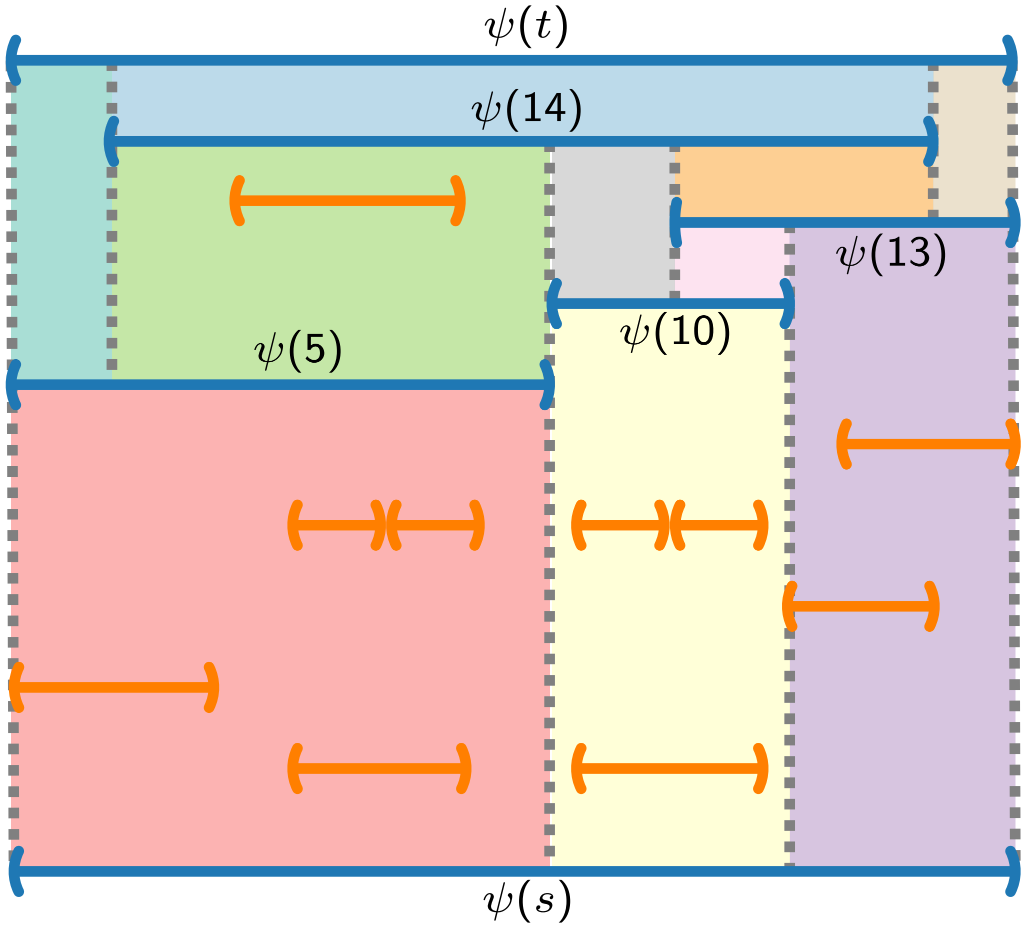


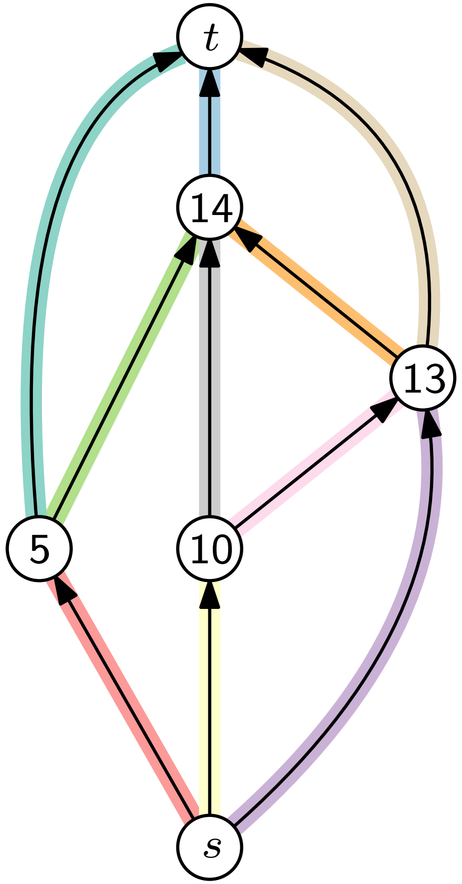


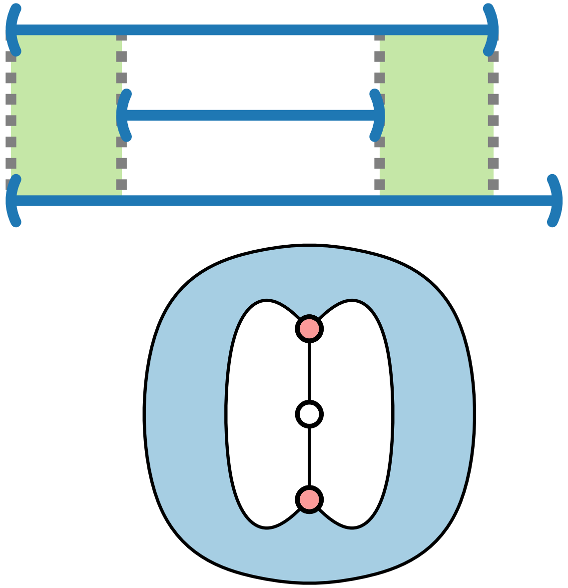
separation pair!

R-Nodes

- for each child (edge) e:

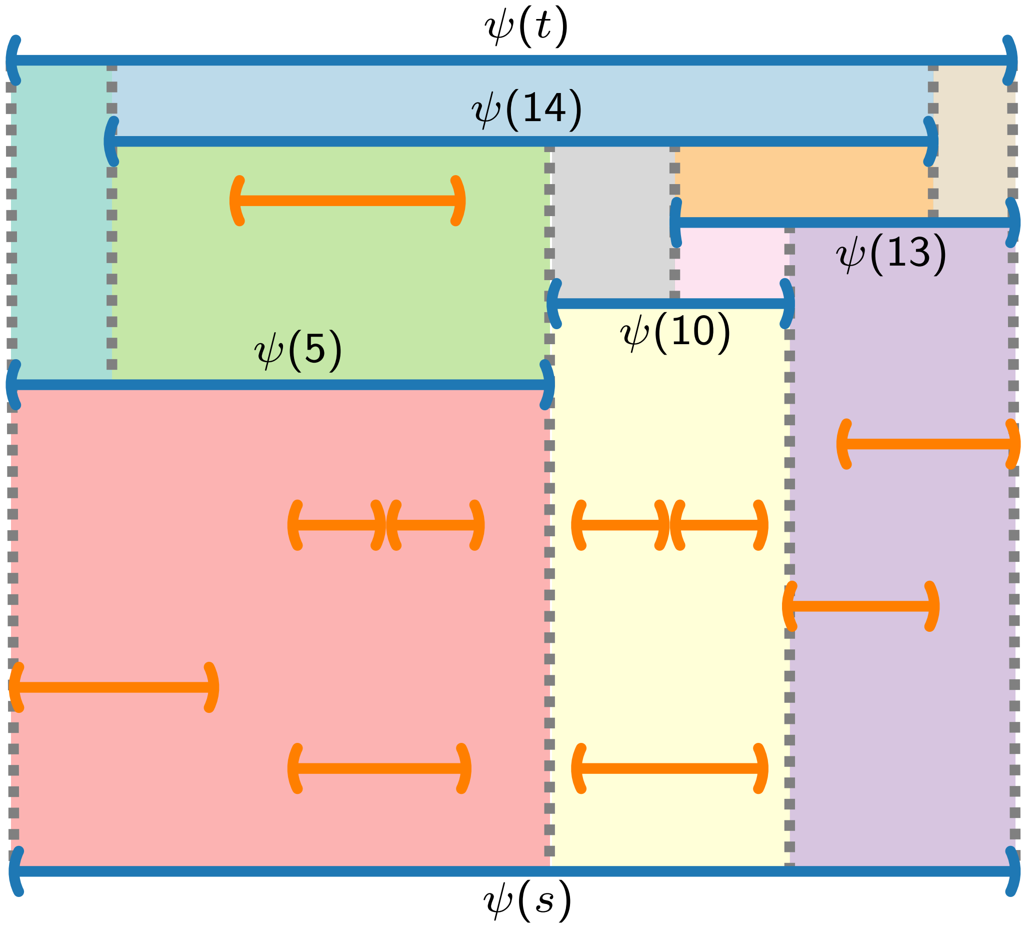


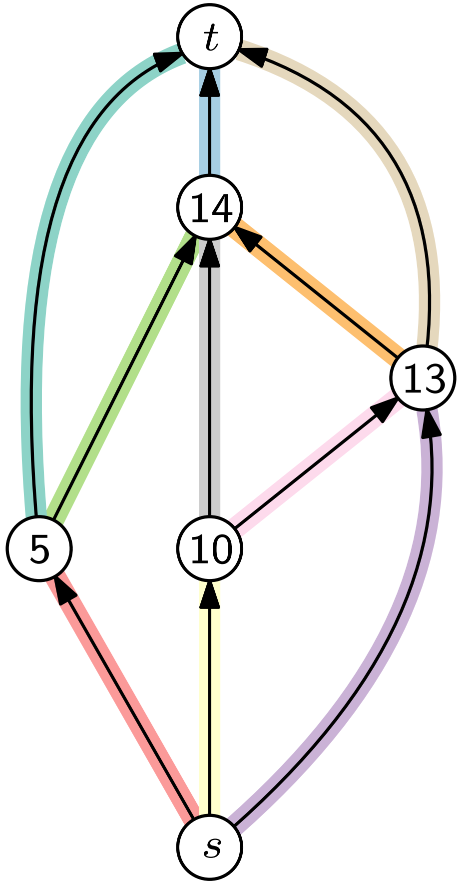


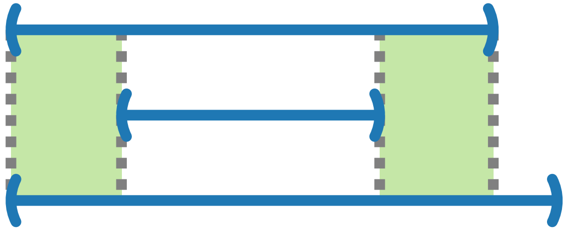


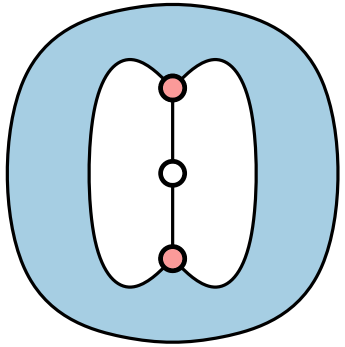
R-Nodes

- for each child (edge) e:
  - find all types of {**FF**,**FL**,**LF**,**LL**} that admit a drawing





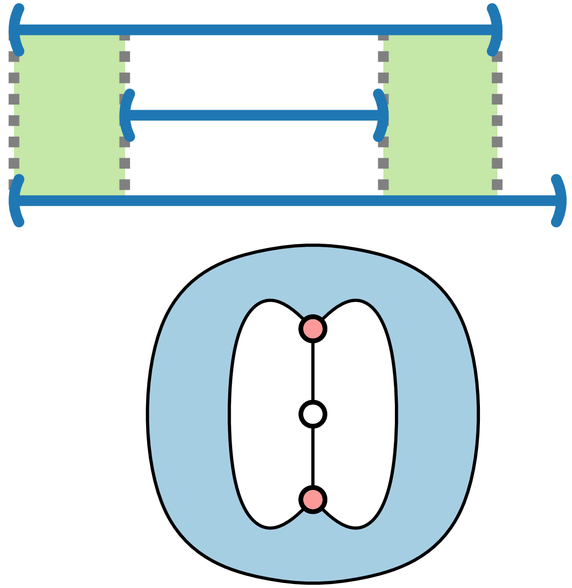


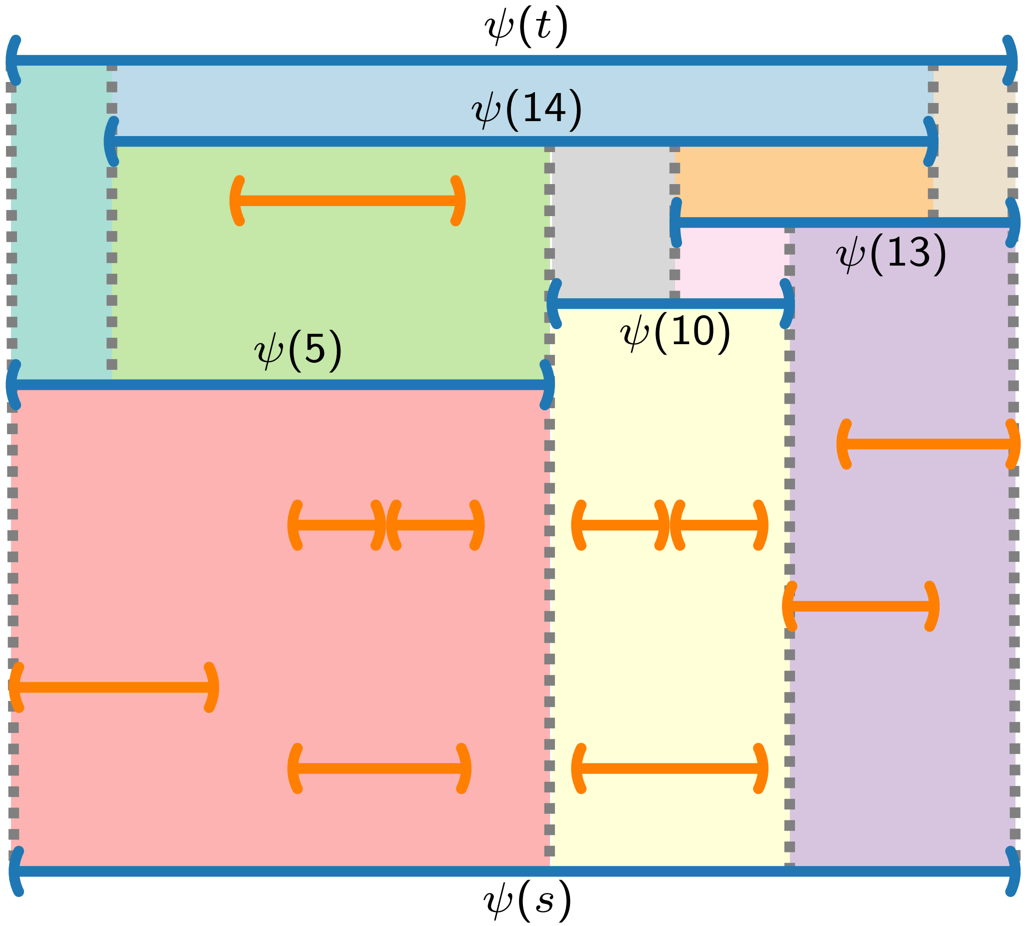


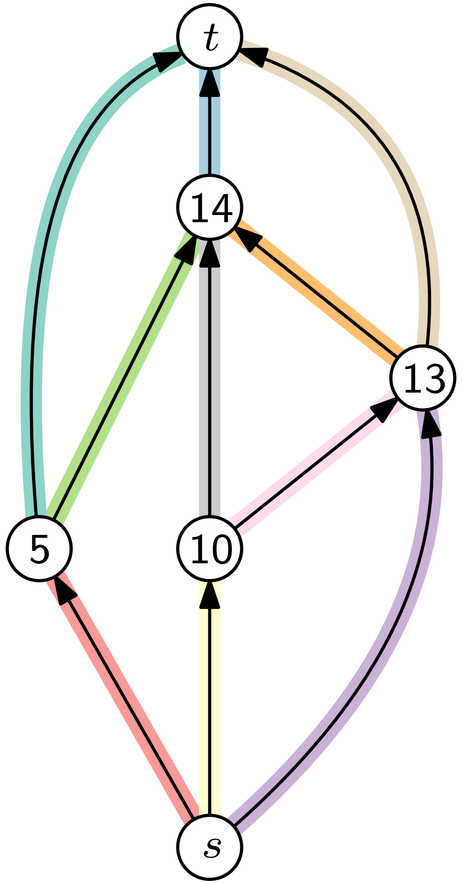
separation pair!

R-Nodes with 2-SAT Formulation

- for each child (edge) e:
  - find all types of {**FF**,**FL**,**LF**,**LL**} that admit a drawing



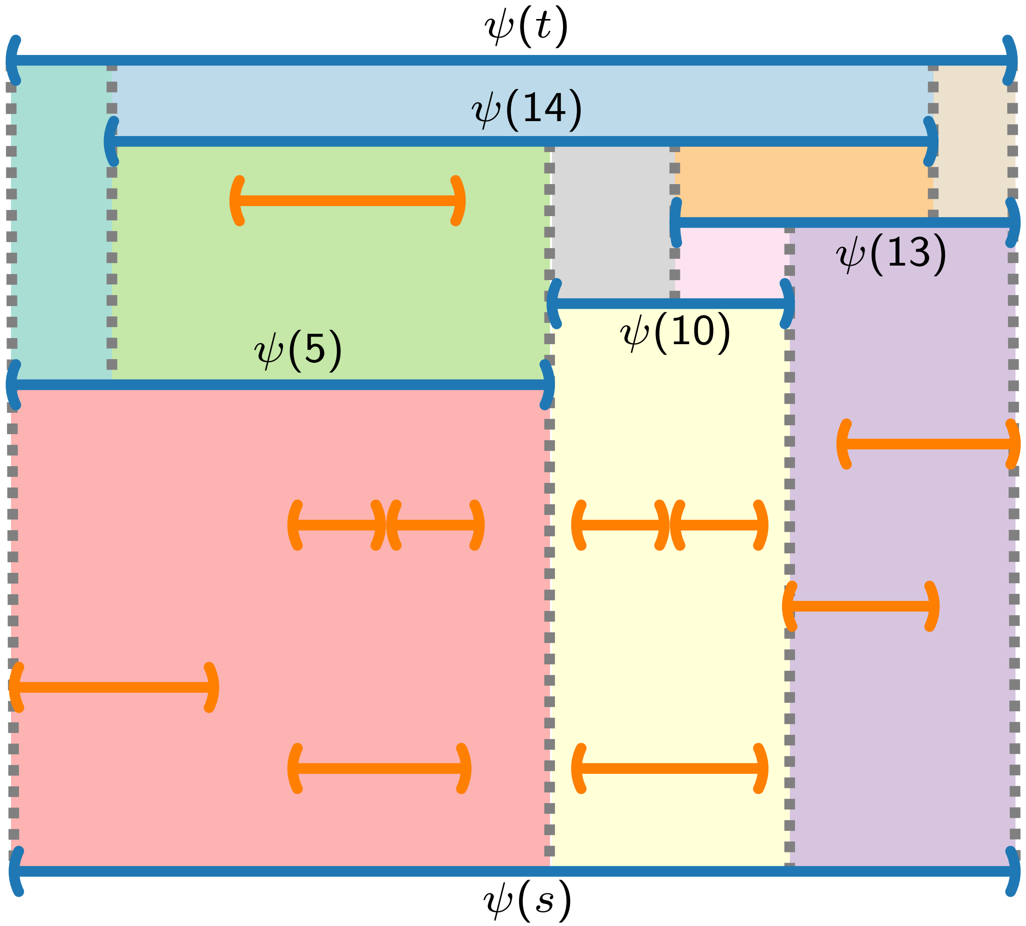


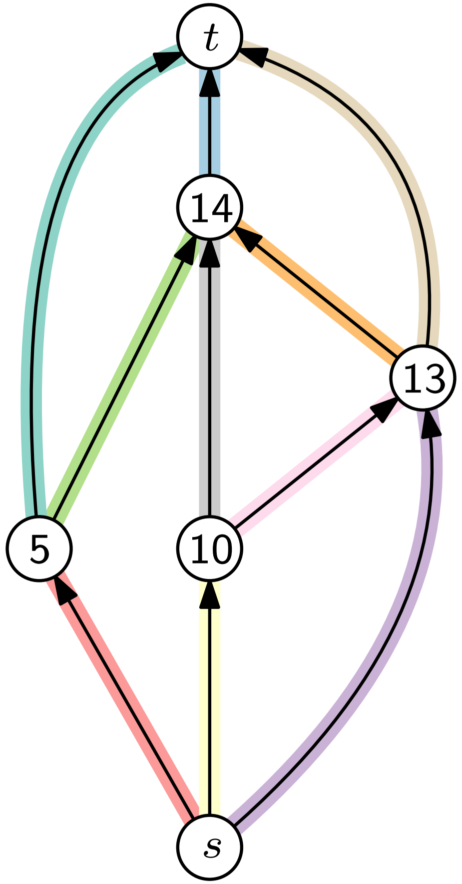


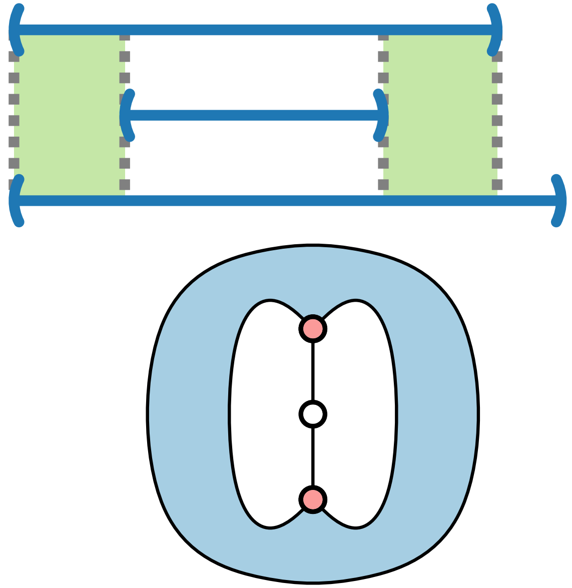
separation pair!

R-Nodes with 2-SAT Formulation

- for each child (edge) e:
  - find all types of {**FF,FL,LF,LL**} that admit a drawing
  - 2 variables l_e, r_e encoding fixed/loose type of its tile



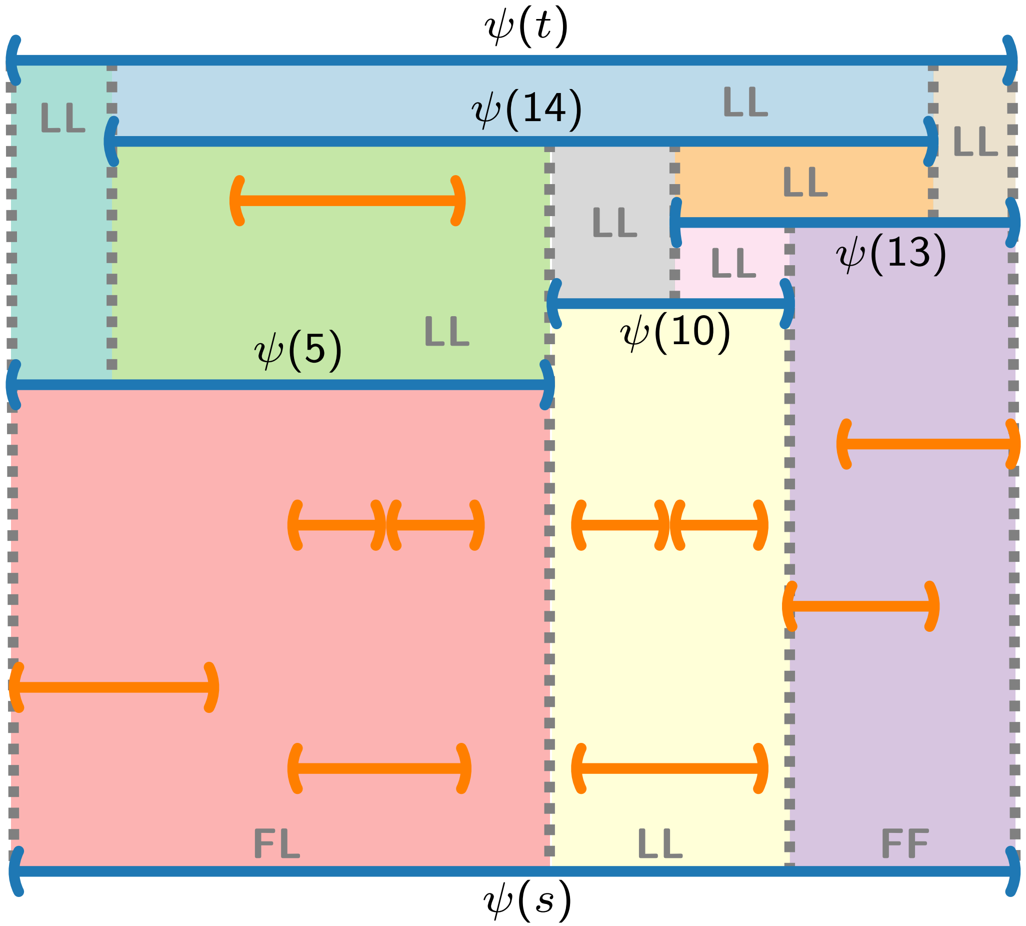


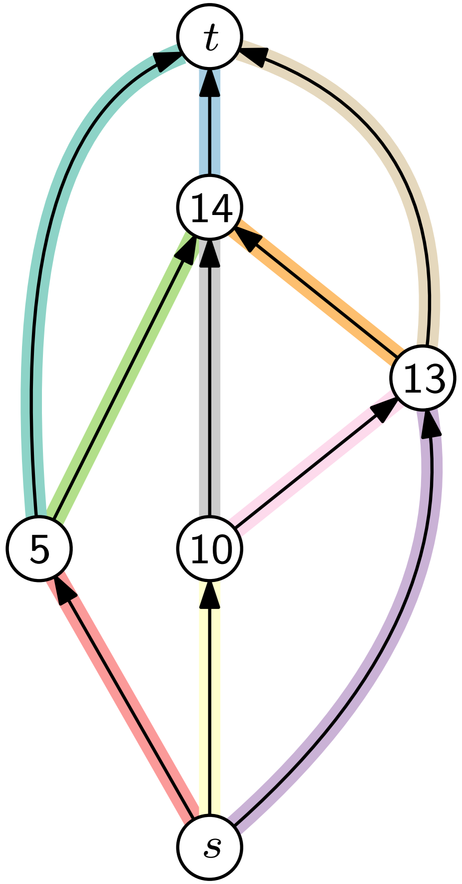


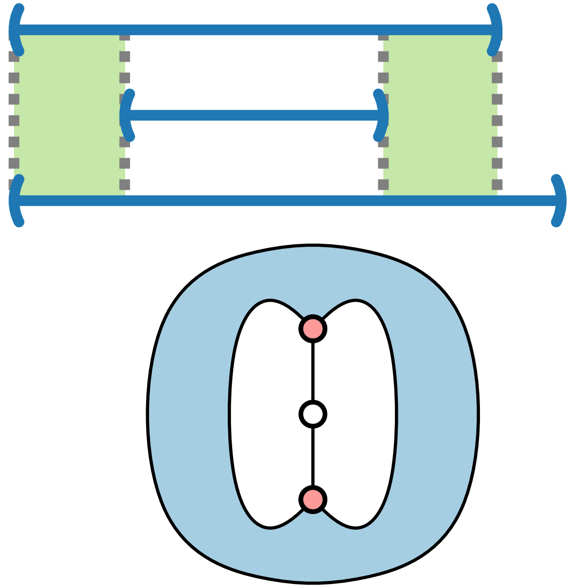
separation pair!

R-Nodes with 2-SAT Formulation

- for each child (edge) e:
  - find all types of {**FF**,**FL**,**LF**,**LL**} that admit a drawing
  - 2 variables l_e, r_e encoding fixed/loose type of its tile



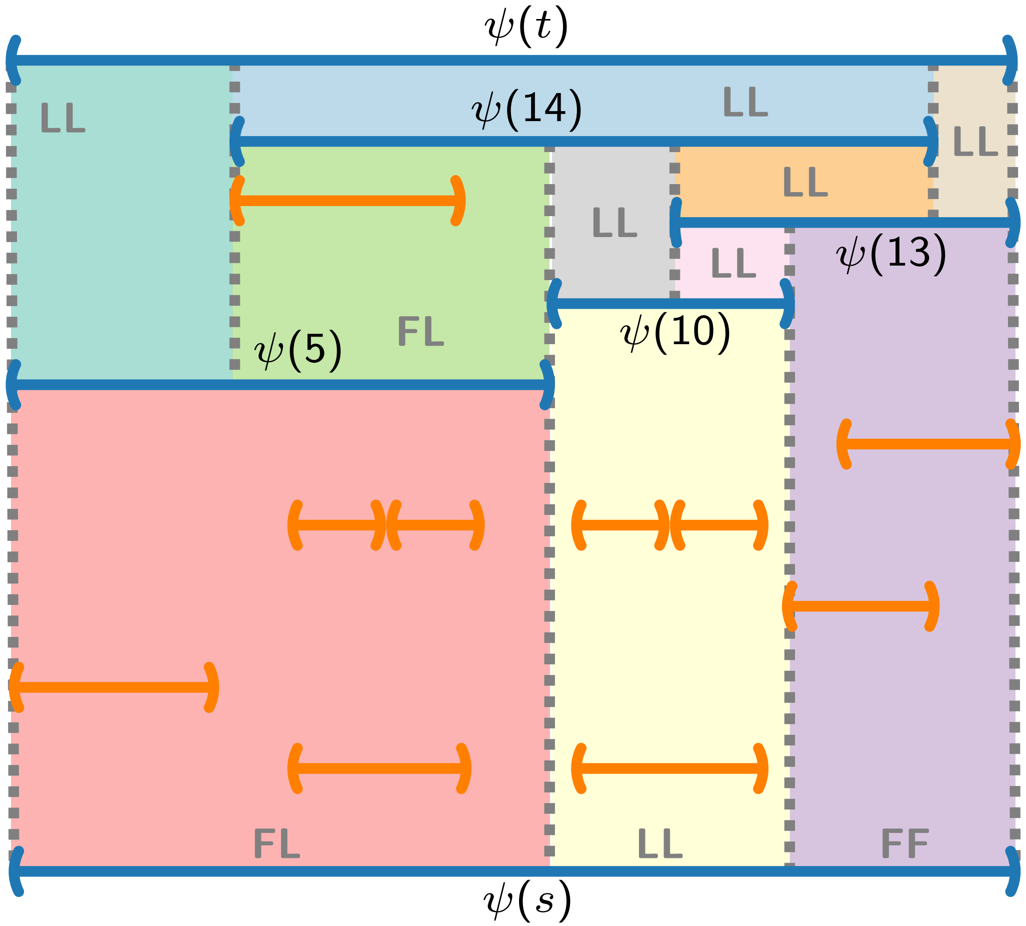


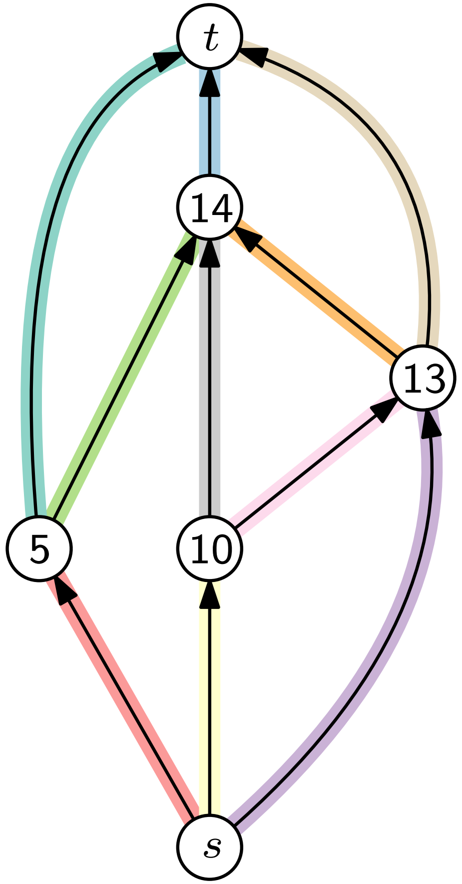


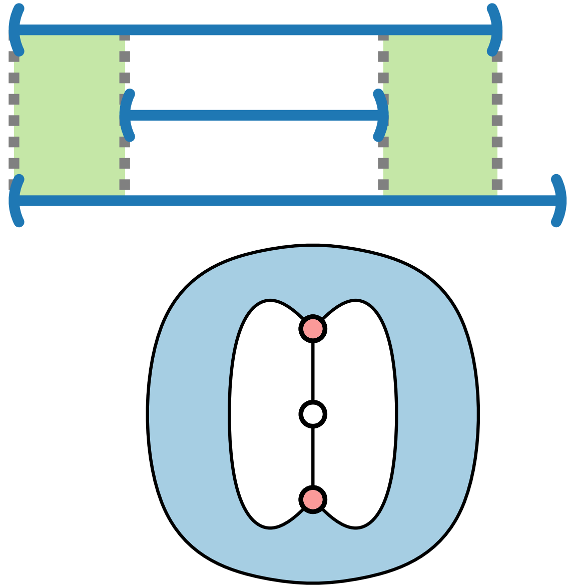
separation pair!

R-Nodes with 2-SAT Formulation

- for each child (edge) e:
  - find all types of {**FF**,**FL**,**LF**,**LL**} that admit a drawing
  - 2 variables l_e, r_e encoding fixed/loose type of its tile



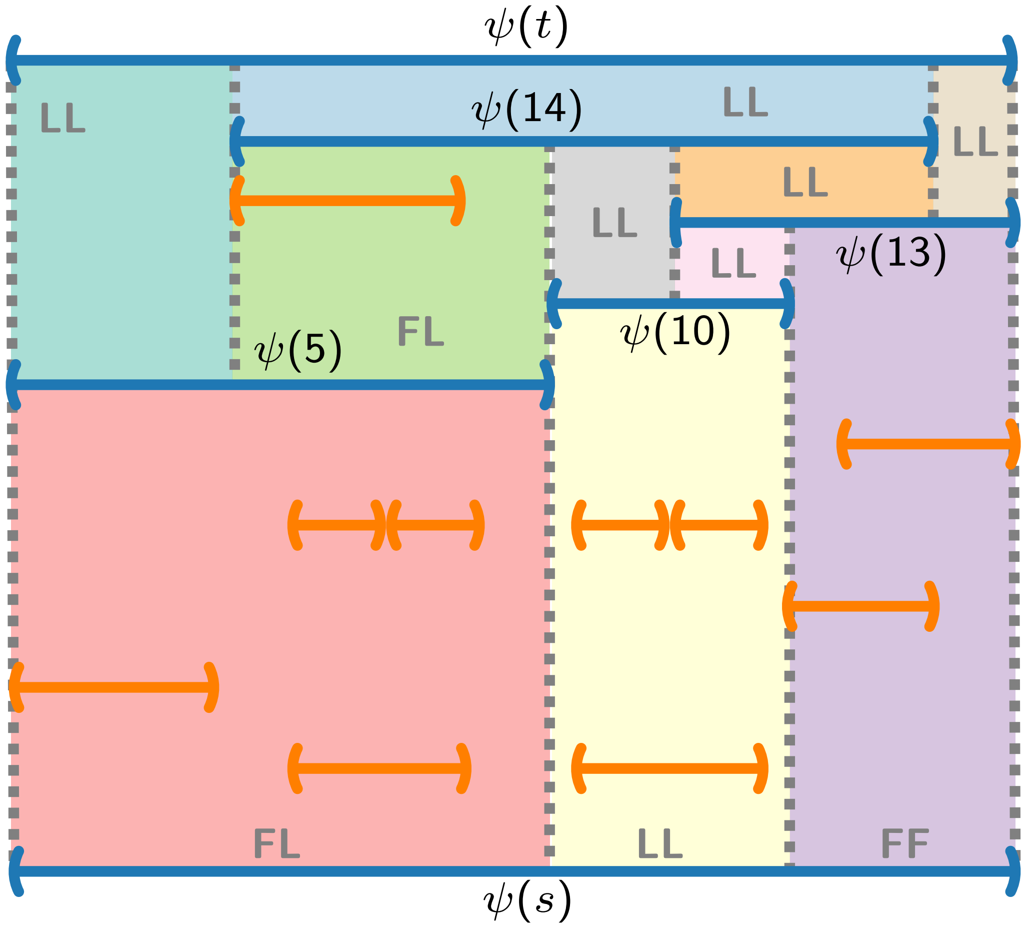


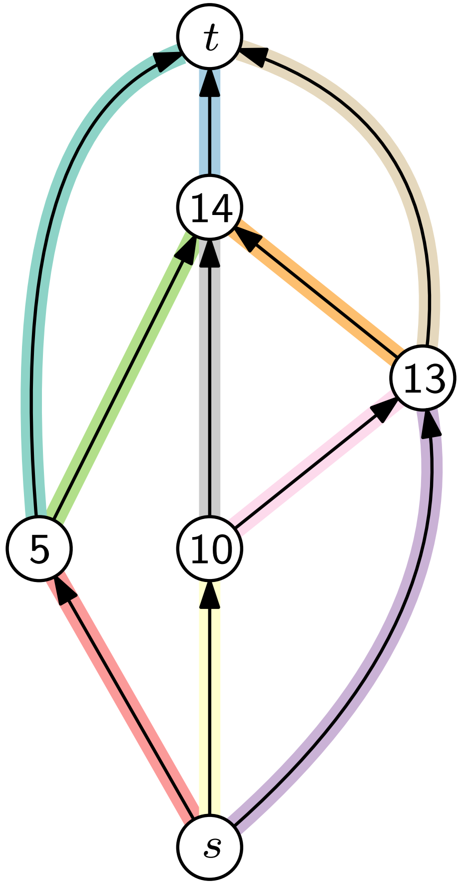


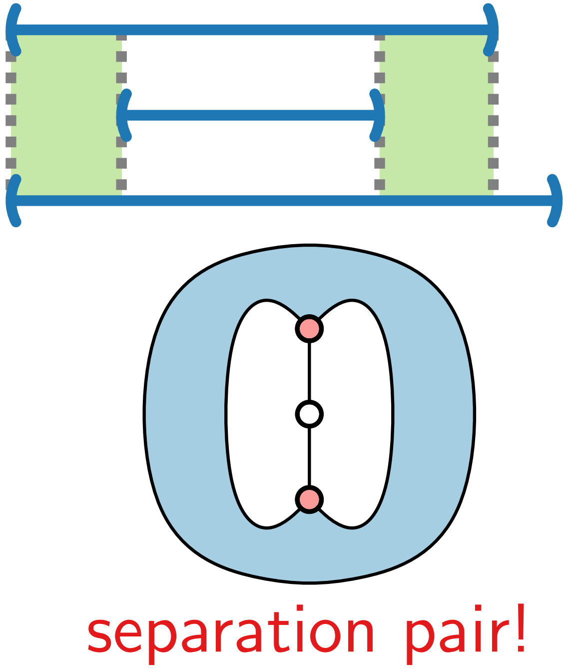
separation pair!

R-Nodes with 2-SAT Formulation

- for each child (edge) e:
  - find all types of {**FF,FL,LF,LL**} that admit a drawing
  - 2 variables l_e, r_e encoding fixed/loose type of its tile
  - consistency clauses

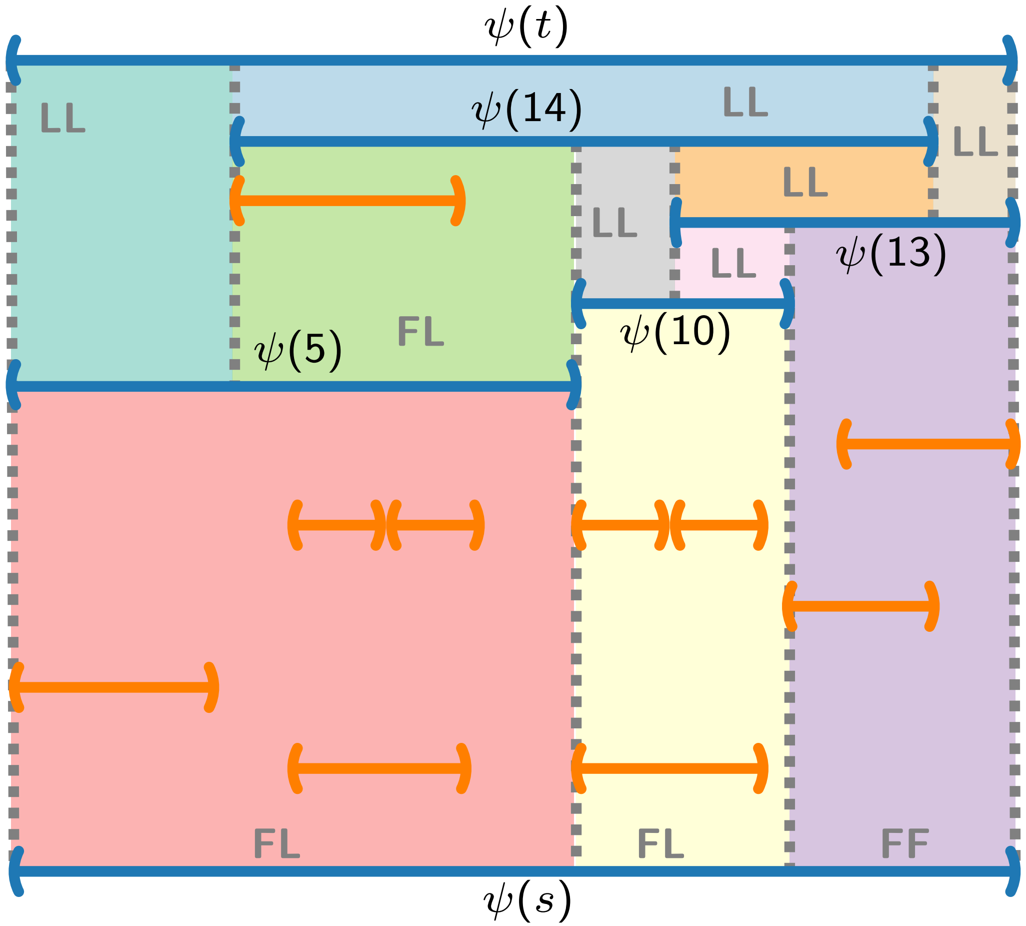


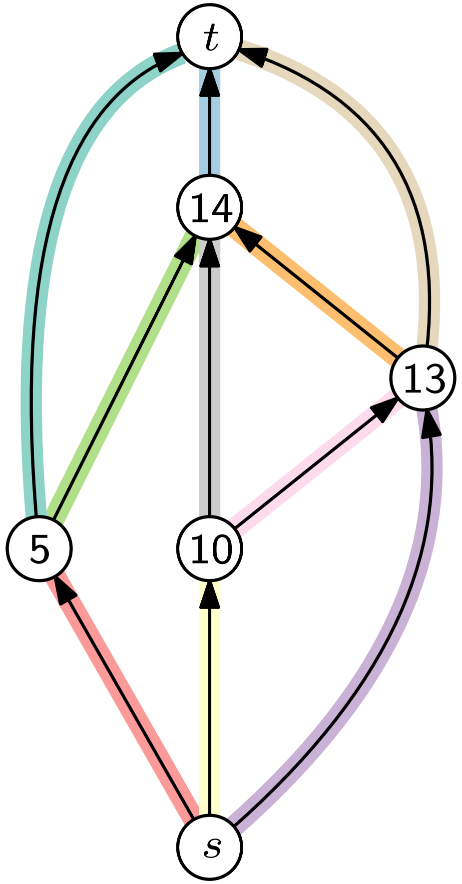


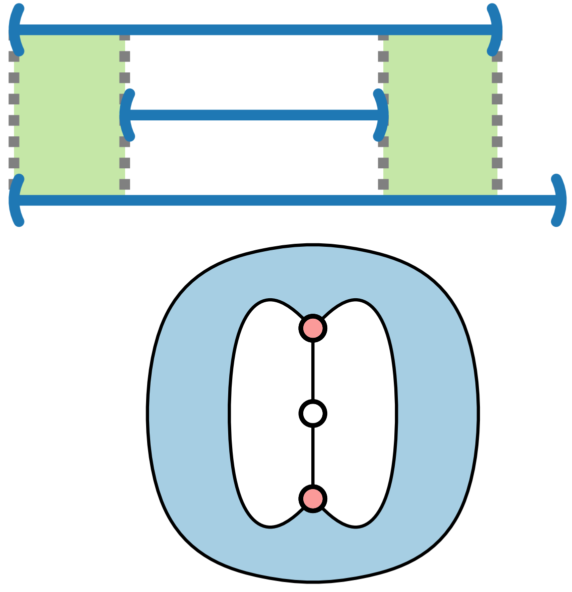


R-Nodes with 2-SAT Formulation

- for each child (edge) e:
  - find all types of {**FF**,**FL**,**LF**,**LL**} that admit a drawing
  - 2 variables l_e, r_e encoding fixed/loose type of its tile
  - consistency clauses

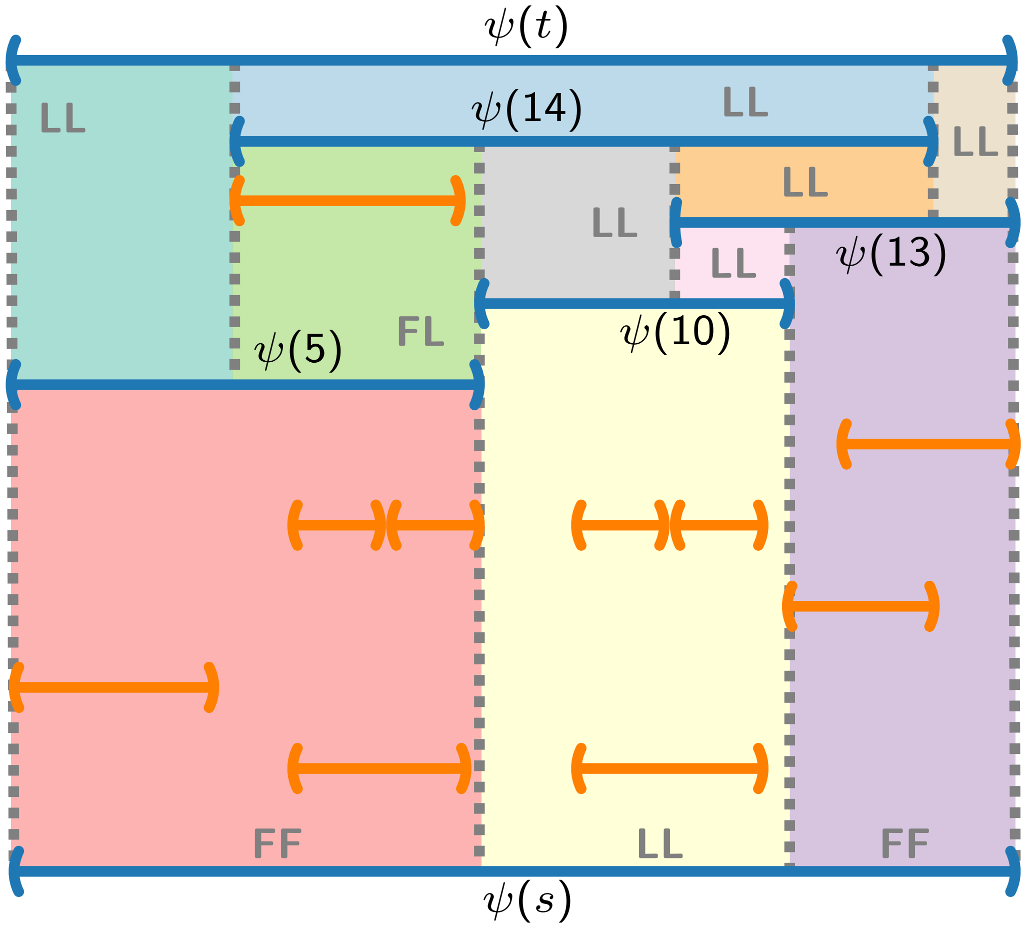


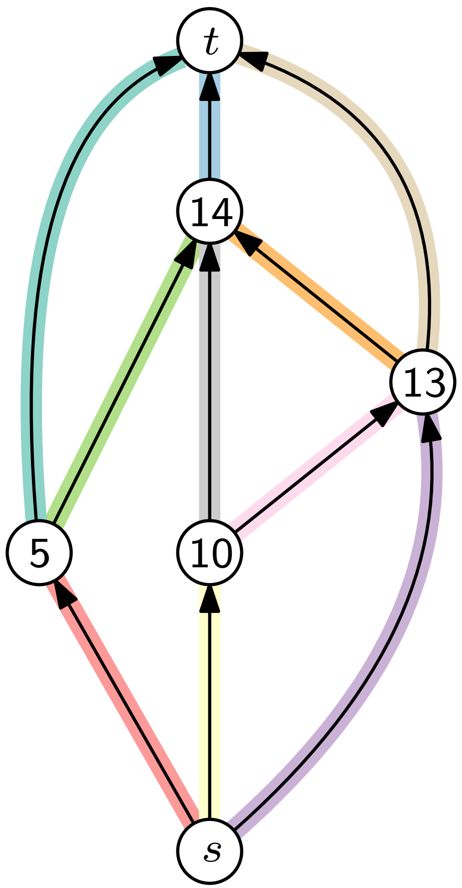


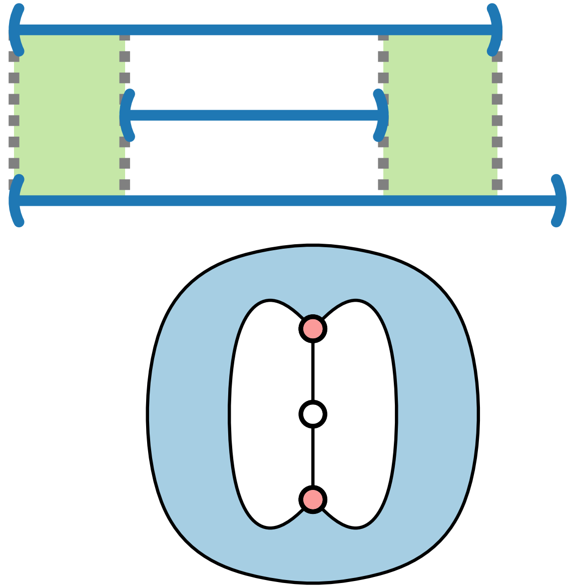


R-Nodes with 2-SAT Formulation

- for each child (edge) e:
  - find all types of {**FF,FL,LF,LL**} that admit a drawing
  - 2 variables l_e, r_e encoding fixed/loose type of its tile
  - consistency clauses



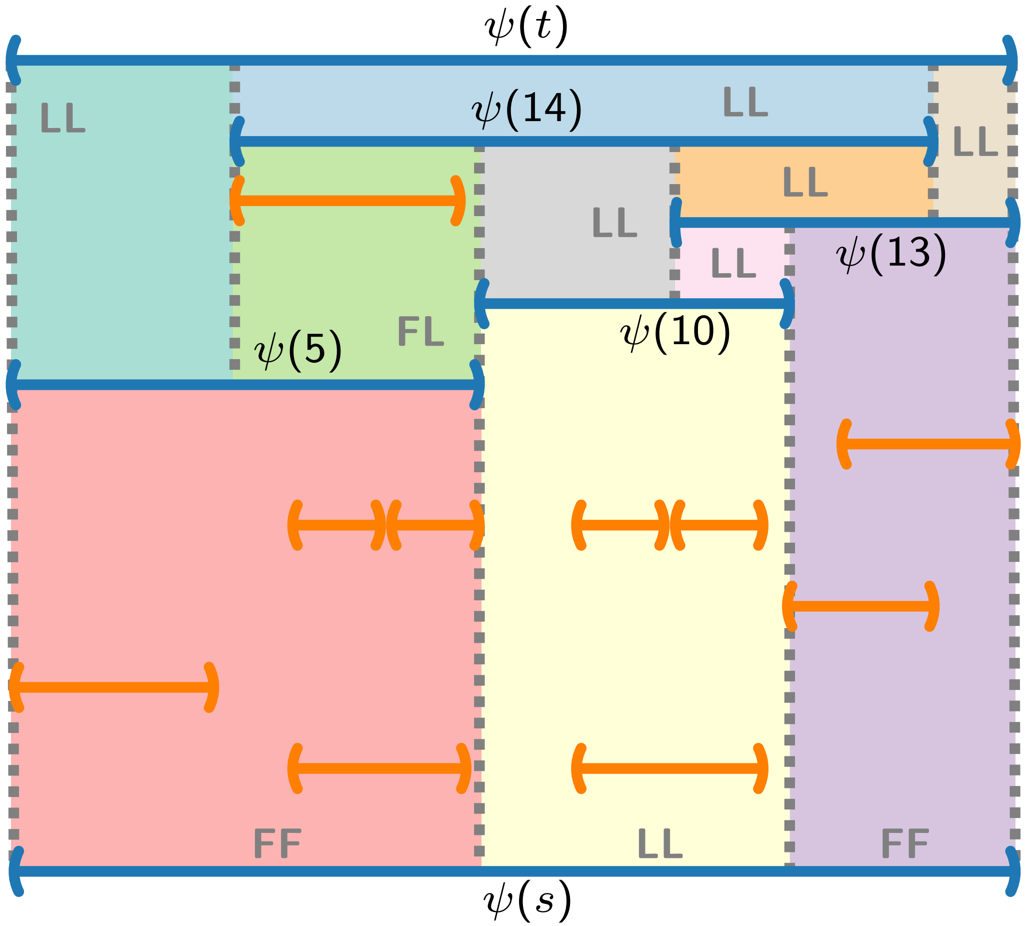


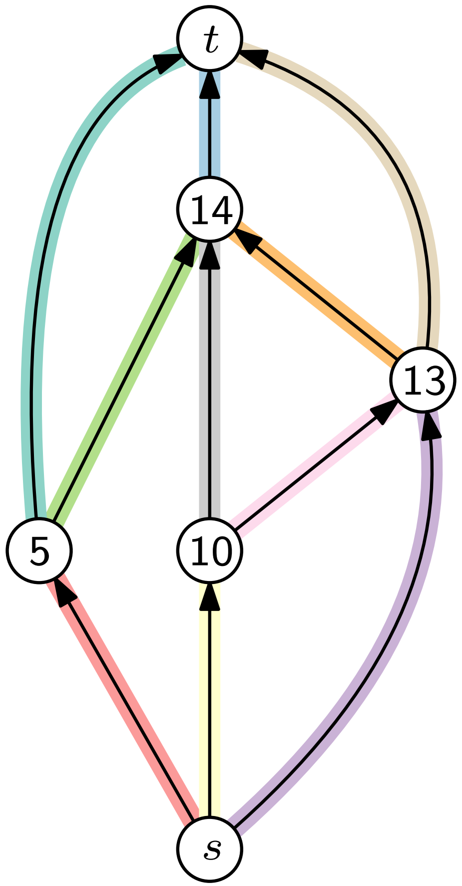


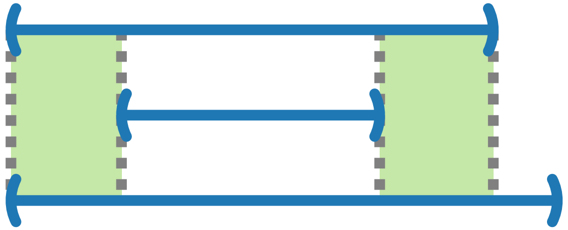
separation pair!

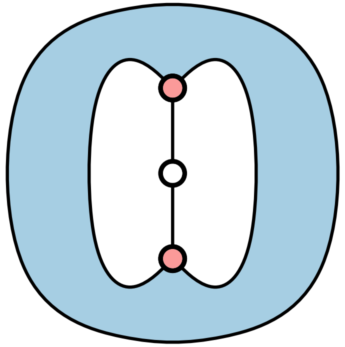
R-Nodes with 2-SAT Formulation

- for each child (edge) e:
  - find all types of $\{$**FF**,**FL**,**LF**,**LL**$\}$ that admit a drawing
  - 2 variables l_e, r_e encoding fixed/loose type of its tile
  - consistency clauses – $O(n^2)$ many,





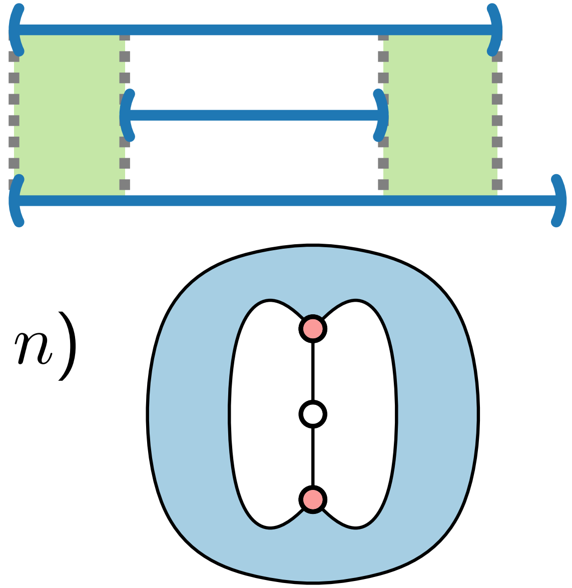


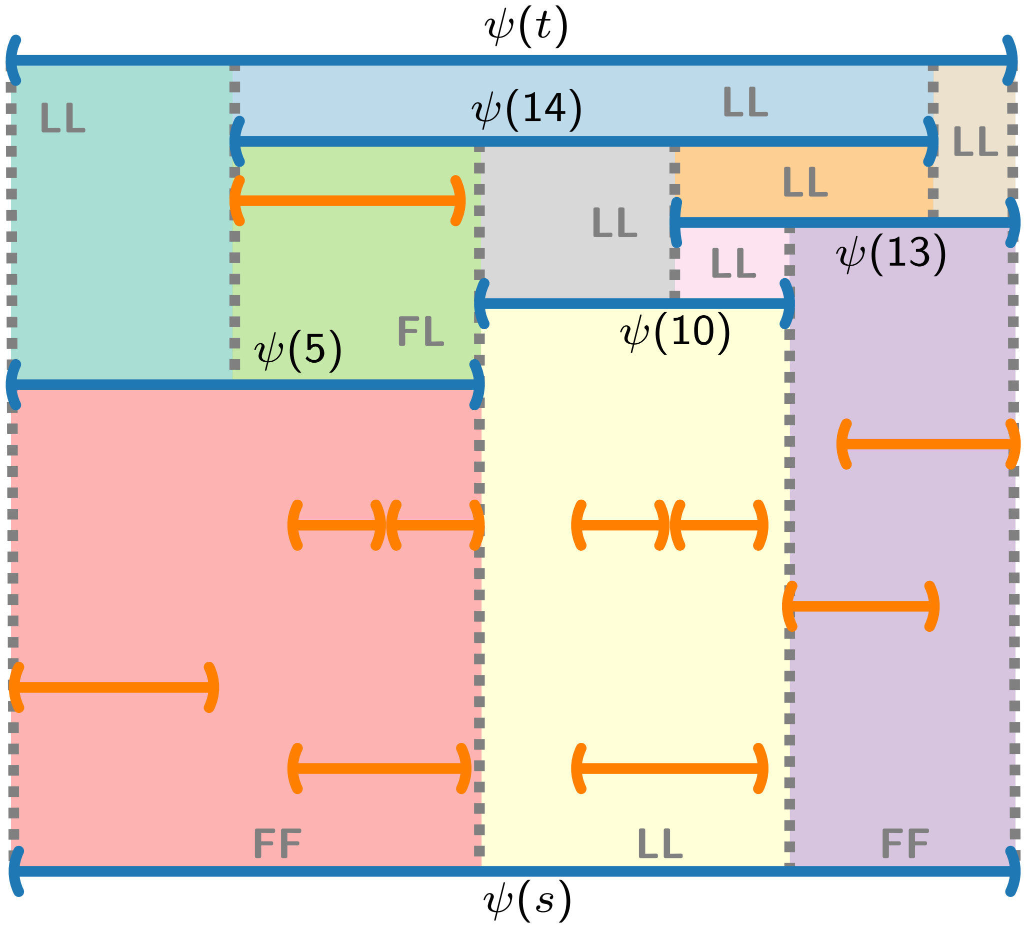


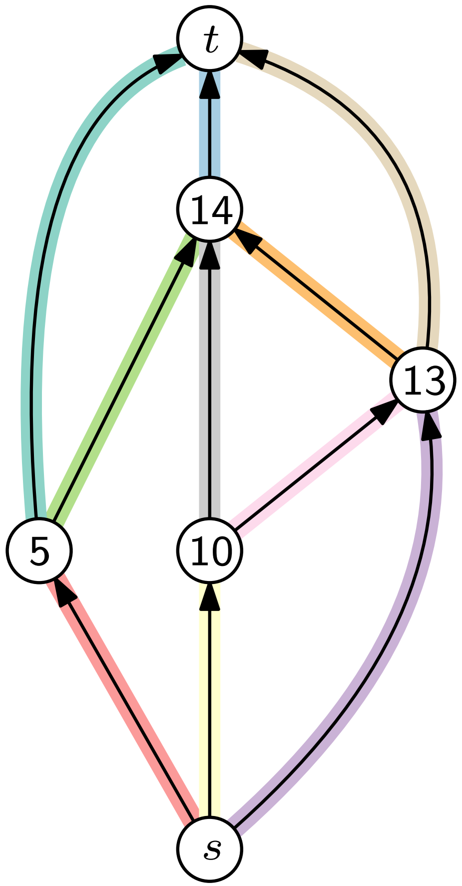
separation pair!

R-Nodes with 2-SAT Formulation

- for each child (edge) e:
  - find all types of {**FF,FL,LF,LL**} that admit a drawing
  - 2 variables l_e, r_e encoding fixed/loose type of its tile
  - consistency clauses – $O(n^2)$ many, but can be reduced to $O(n \log^2 n)$

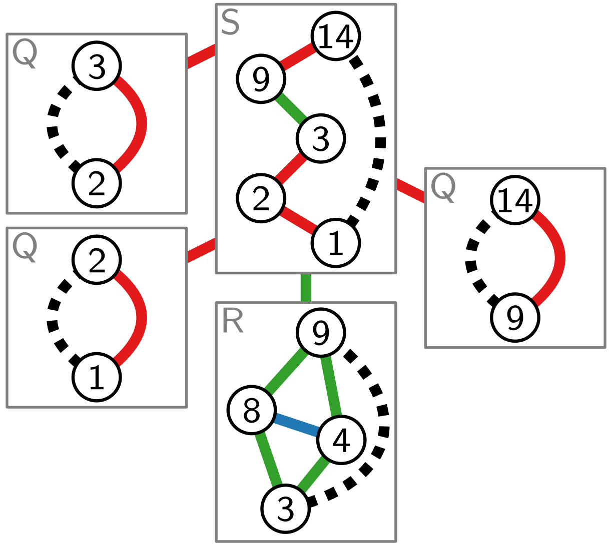




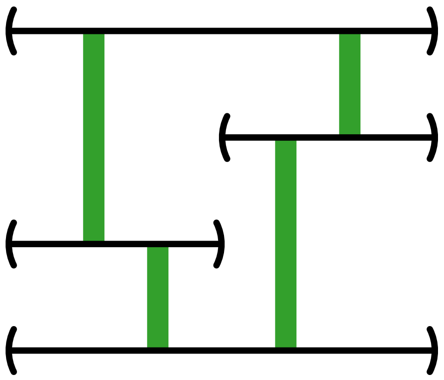


Visualization of Graphs

Lecture 9:
Partial Visibility Representation Extension



Part VI:
NP-Hardness of the General Case



Alexander Wolff

NP-Hardness of RepExt in the General Case

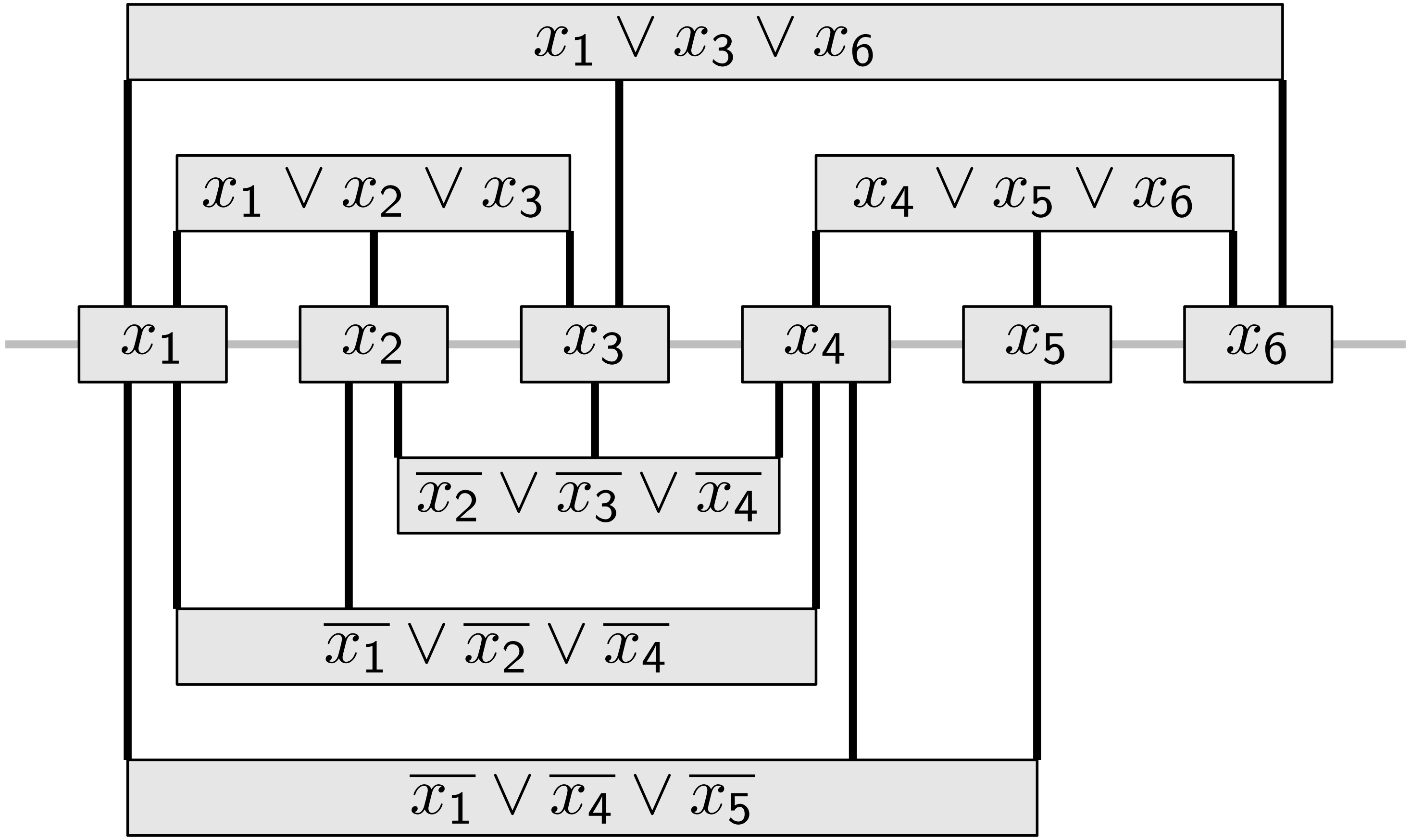
Theorem 2.
ε-Bar Visibility Representation Extension is NP-complete.

- Reduction from Planar Monotone 3-SAT

NP-Hardness of RepExt in the General Case

Theorem 2.
ε-Bar Visibility Representation Extension is NP-complete.

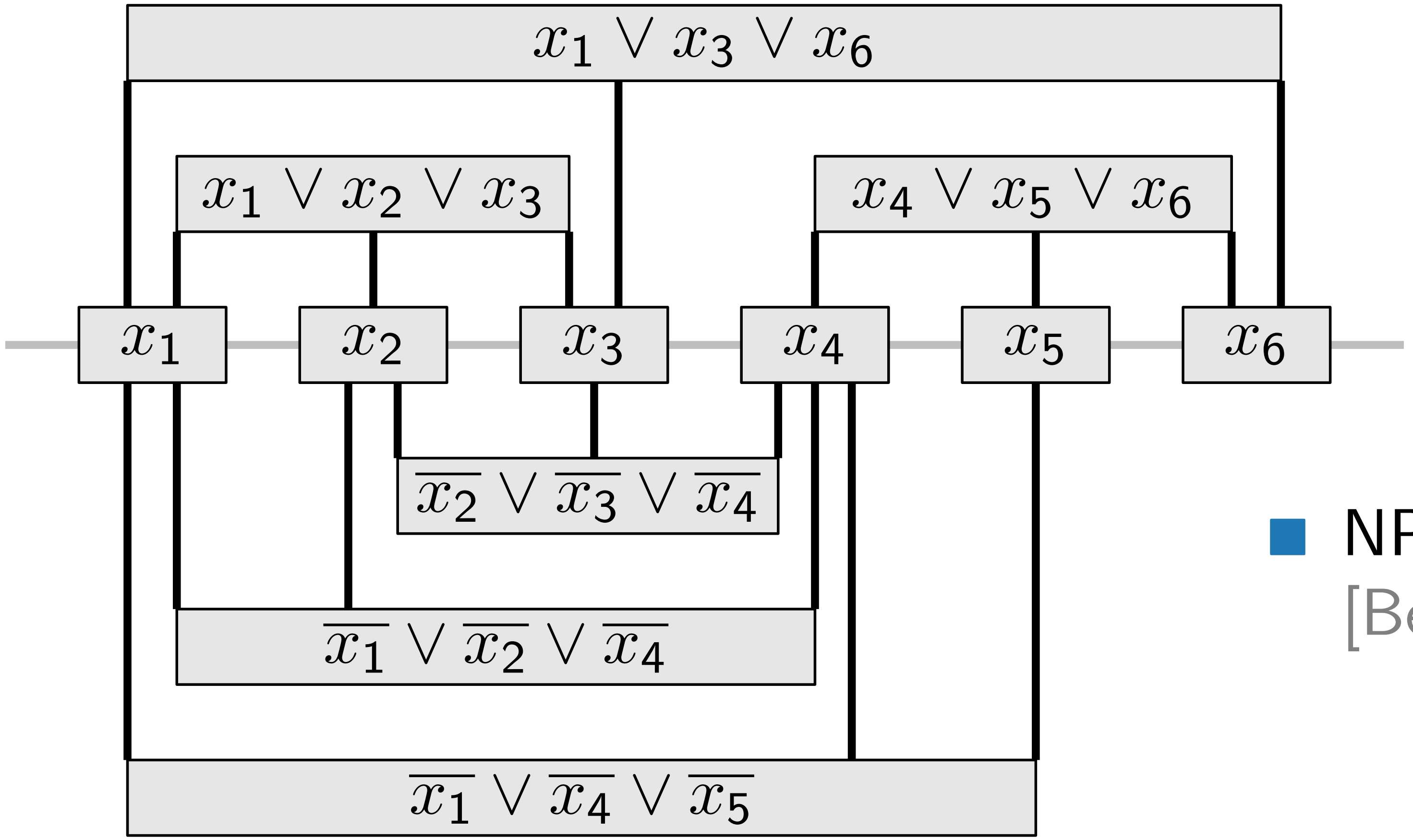
- Reduction from Planar Monotone 3-SAT



NP-Hardness of RepExt in the General Case

Theorem 2.
ε-Bar Visibility Representation Extension is NP-complete.

- Reduction from Planar Monotone 3-SAT

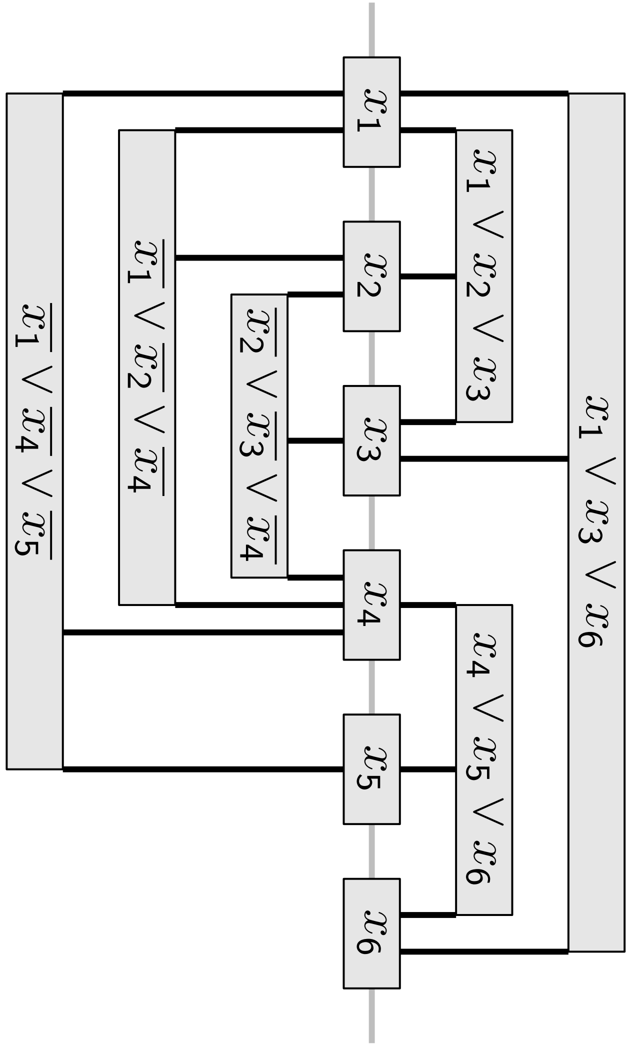


- NP-complete
  [Berg & Khosravi '10]

NP-Hardness of RepExt in the General Case

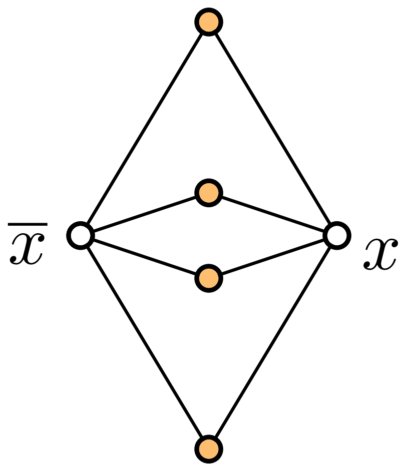
Theorem 2.
ε-Bar Visibility Representation Extension is NP-complete.

- Reduction from Planar Monotone 3-SAT

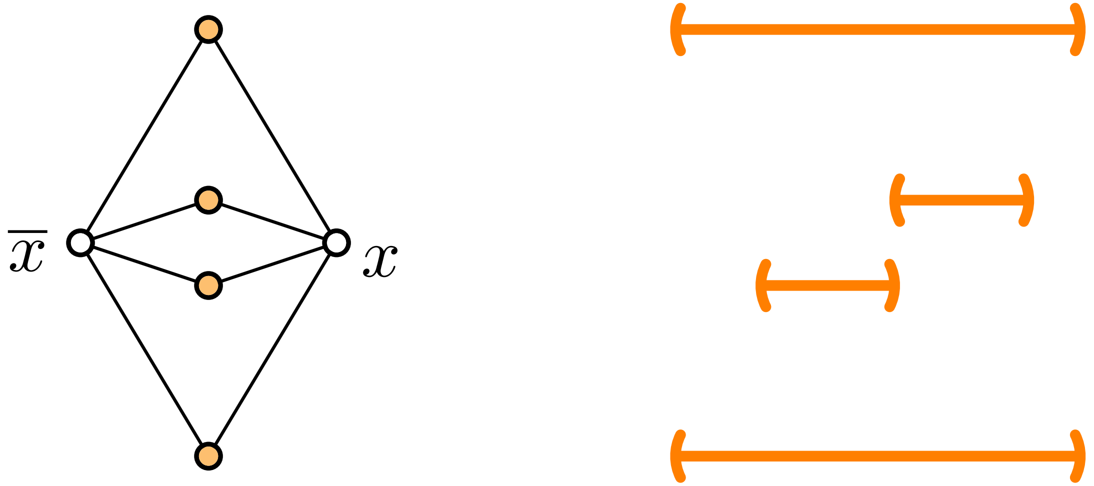


- NP-complete
  [Berg & Khosravi '10]

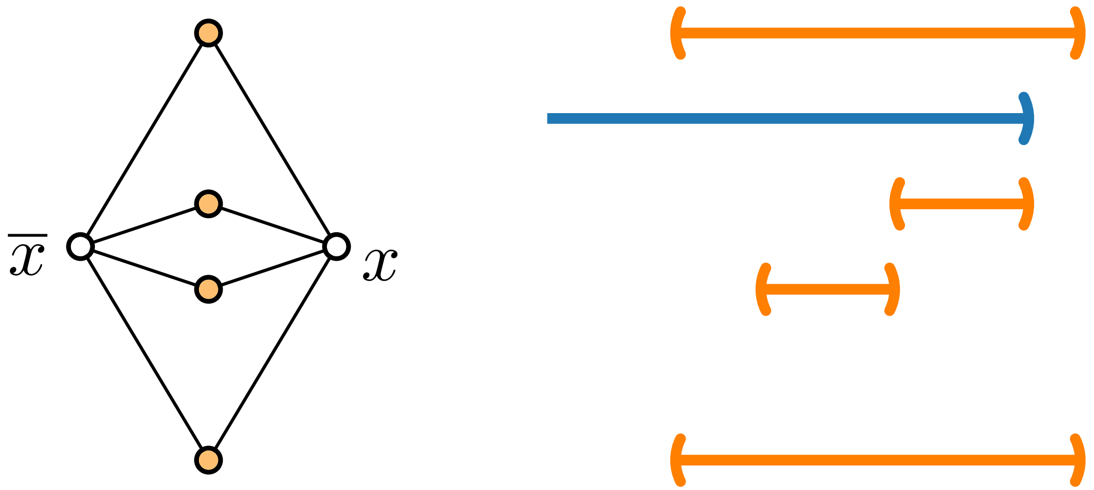
Variable Gadget



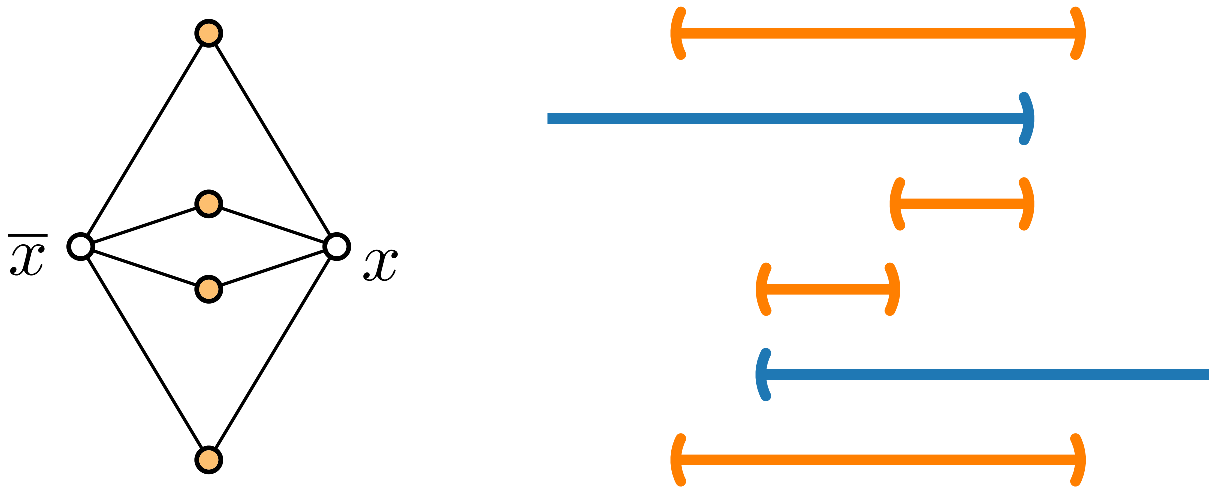
Variable Gadget



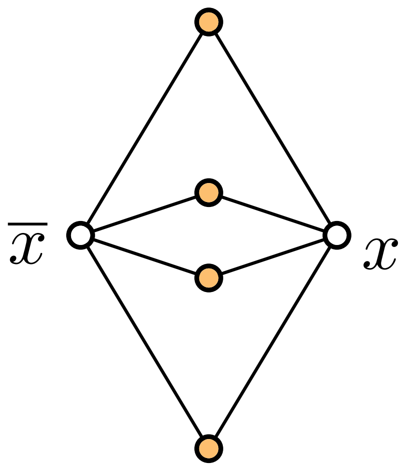
Variable Gadget



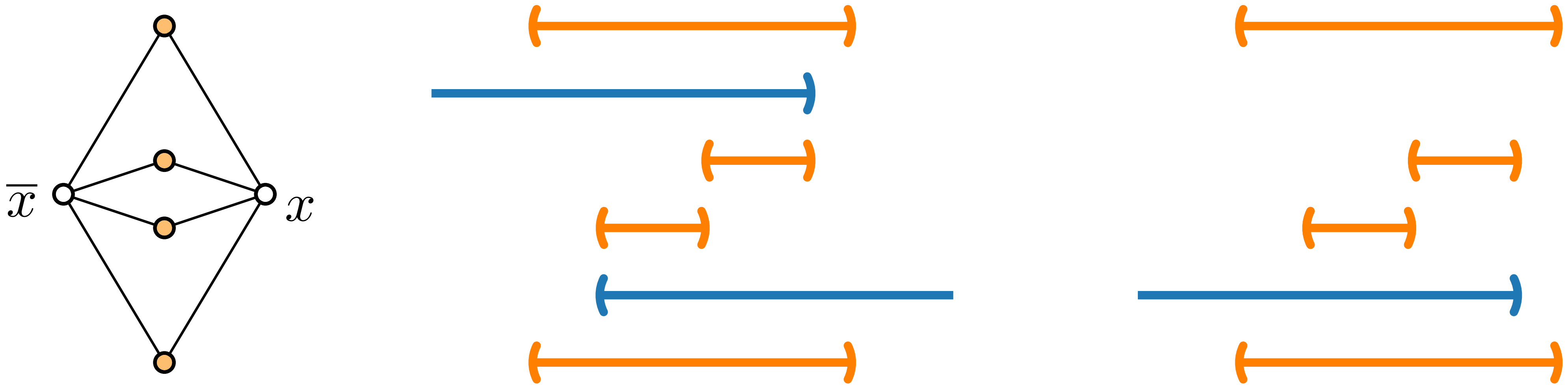
Variable Gadget



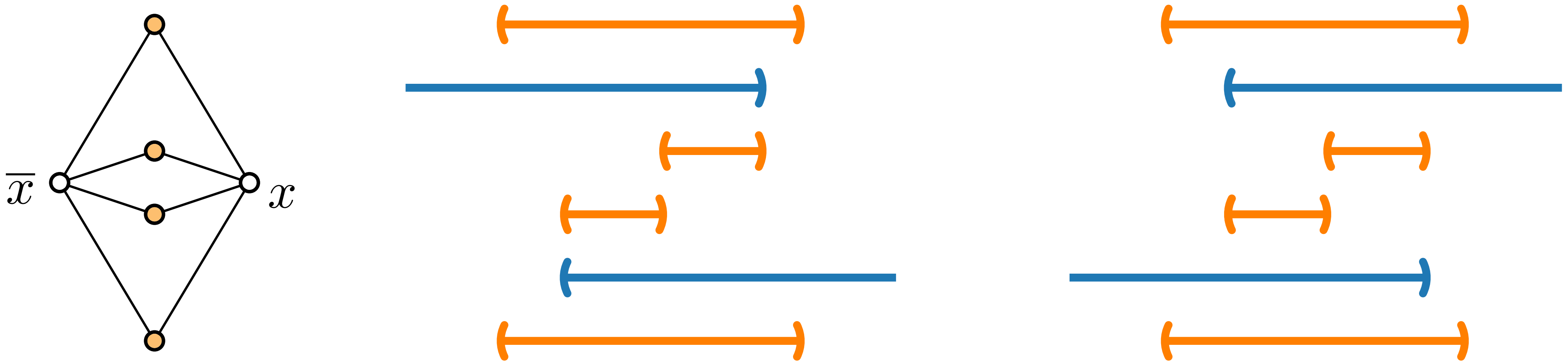
Variable Gadget



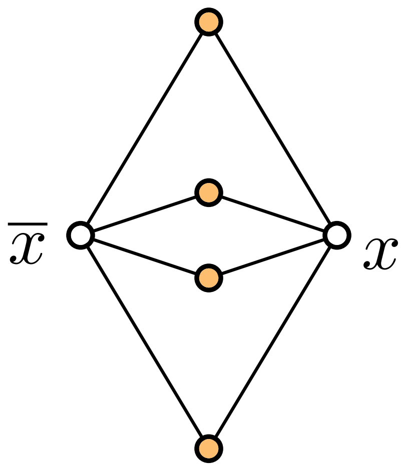
Variable Gadget

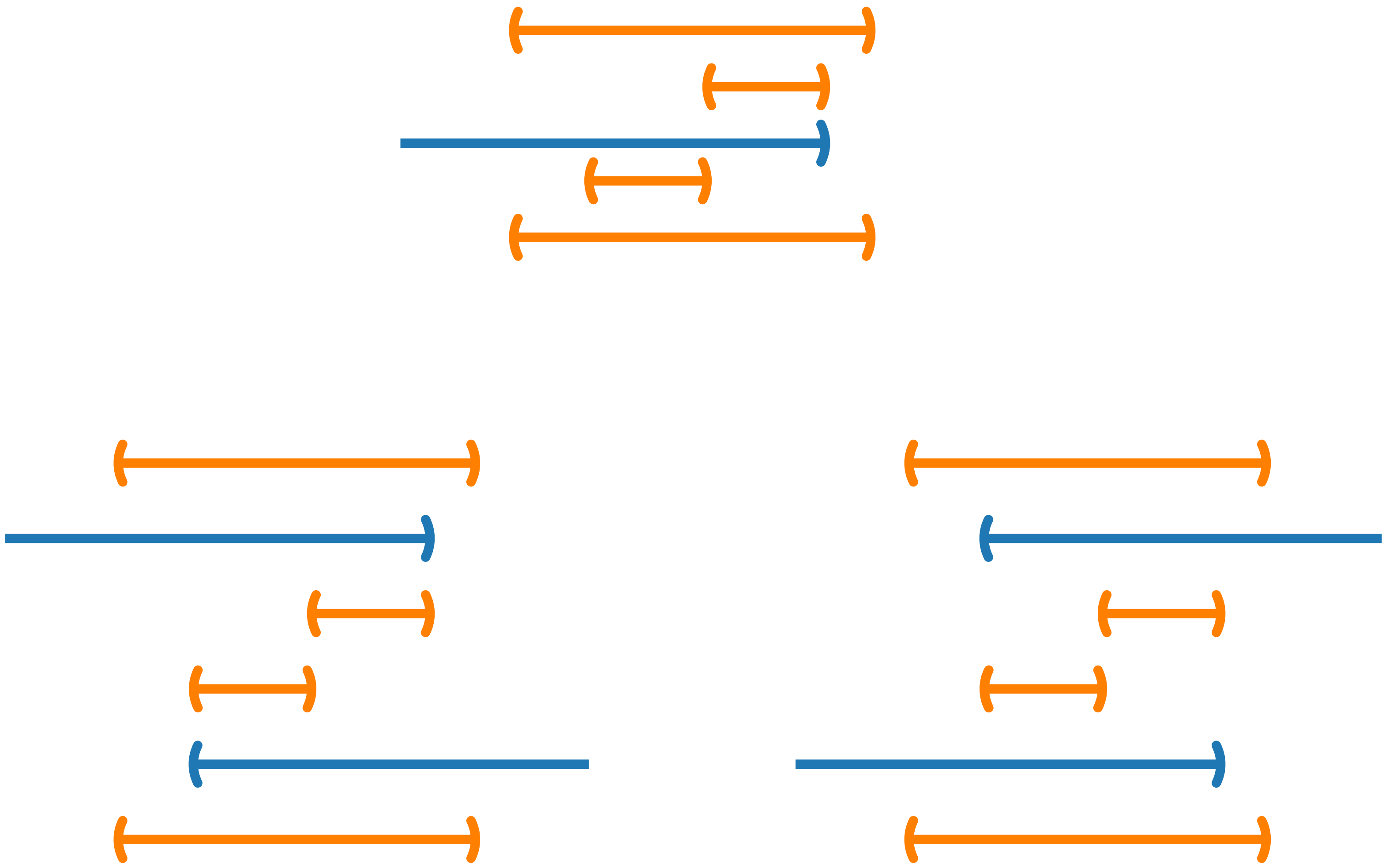


Variable Gadget

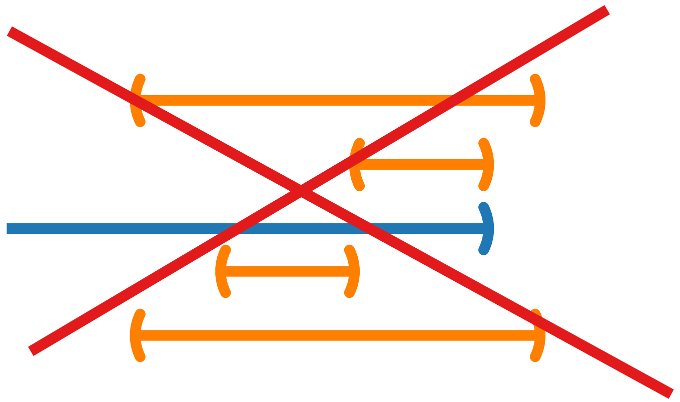


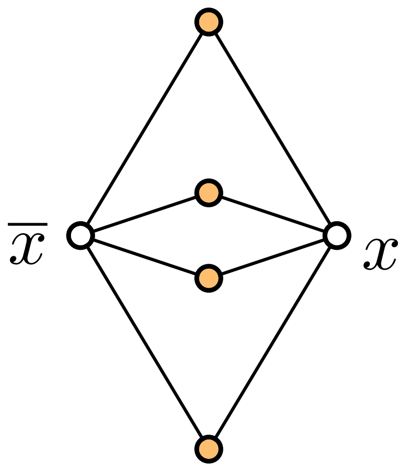
Variable Gadget

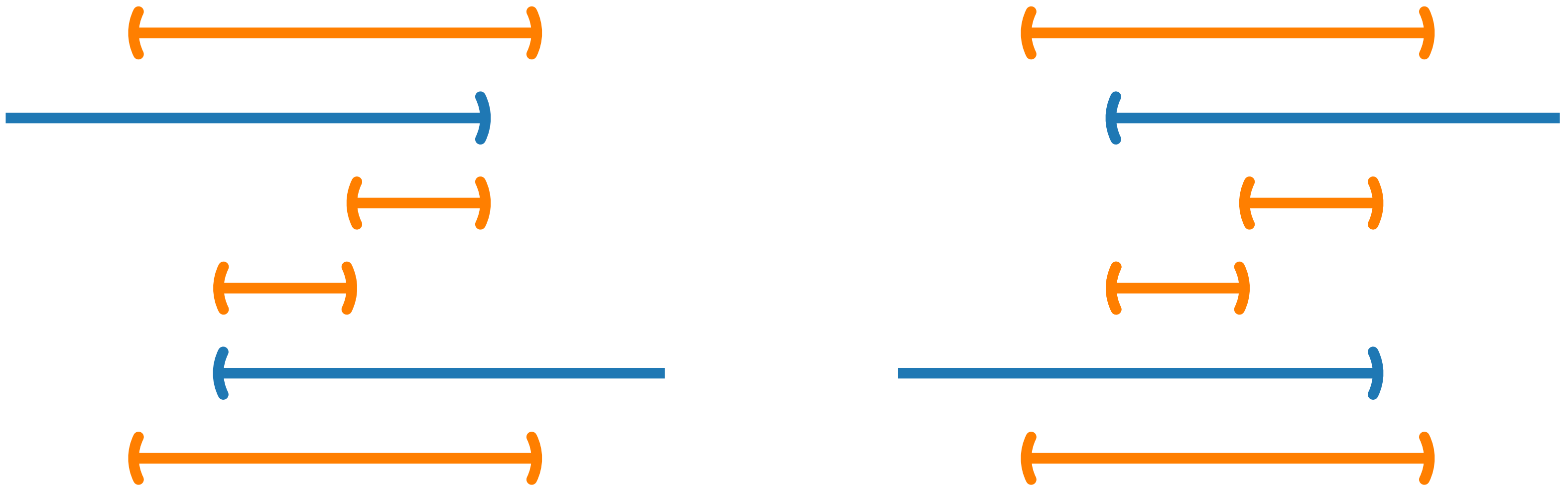




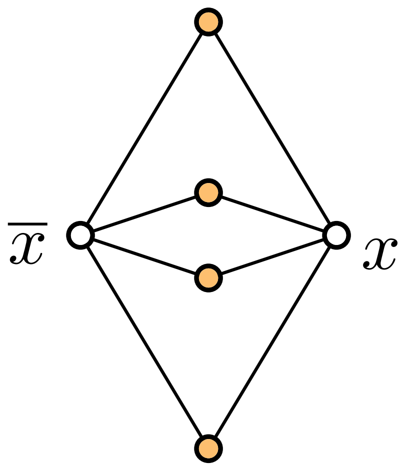
Variable Gadget

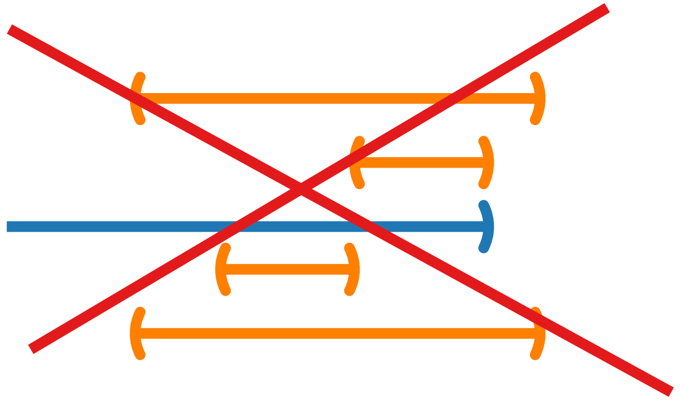


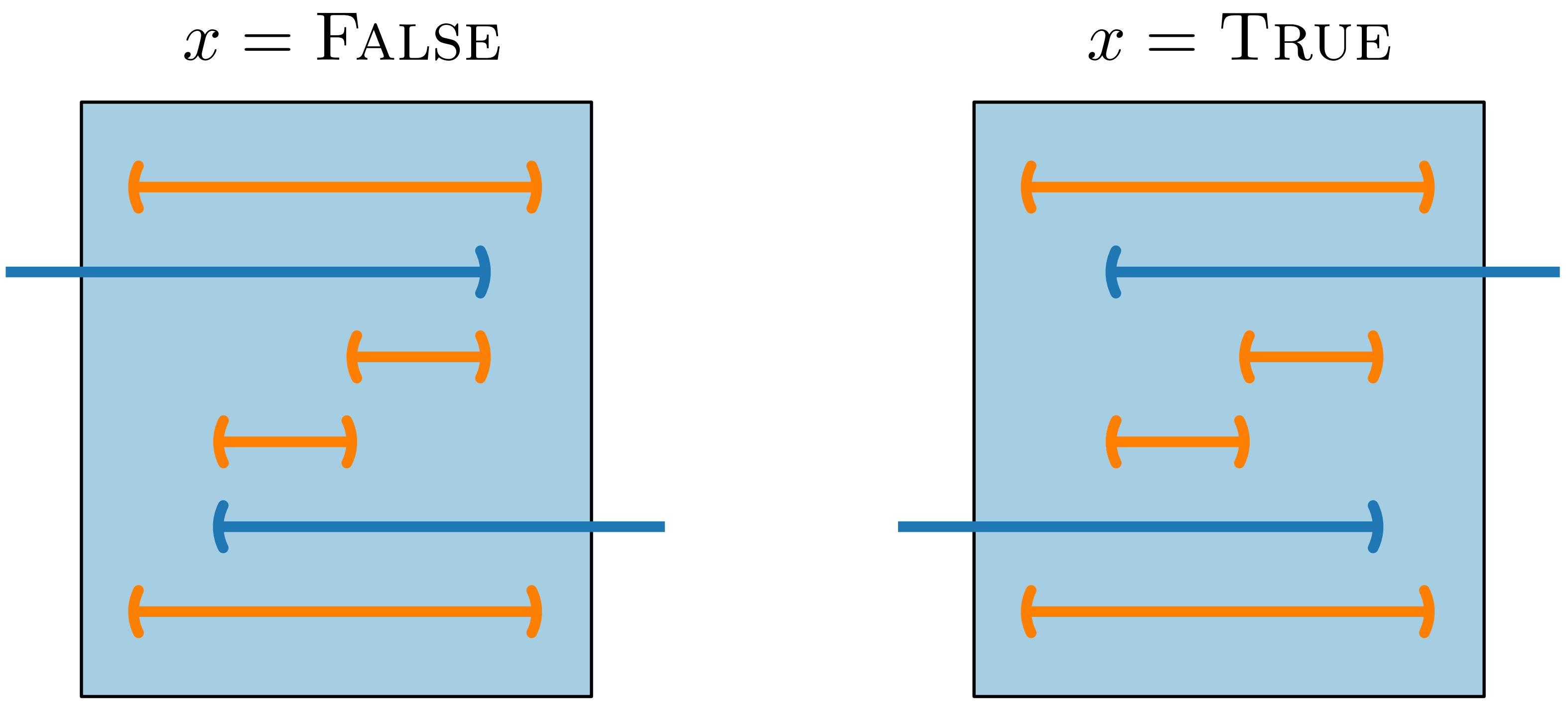




Variable Gadget







Clause Gadget

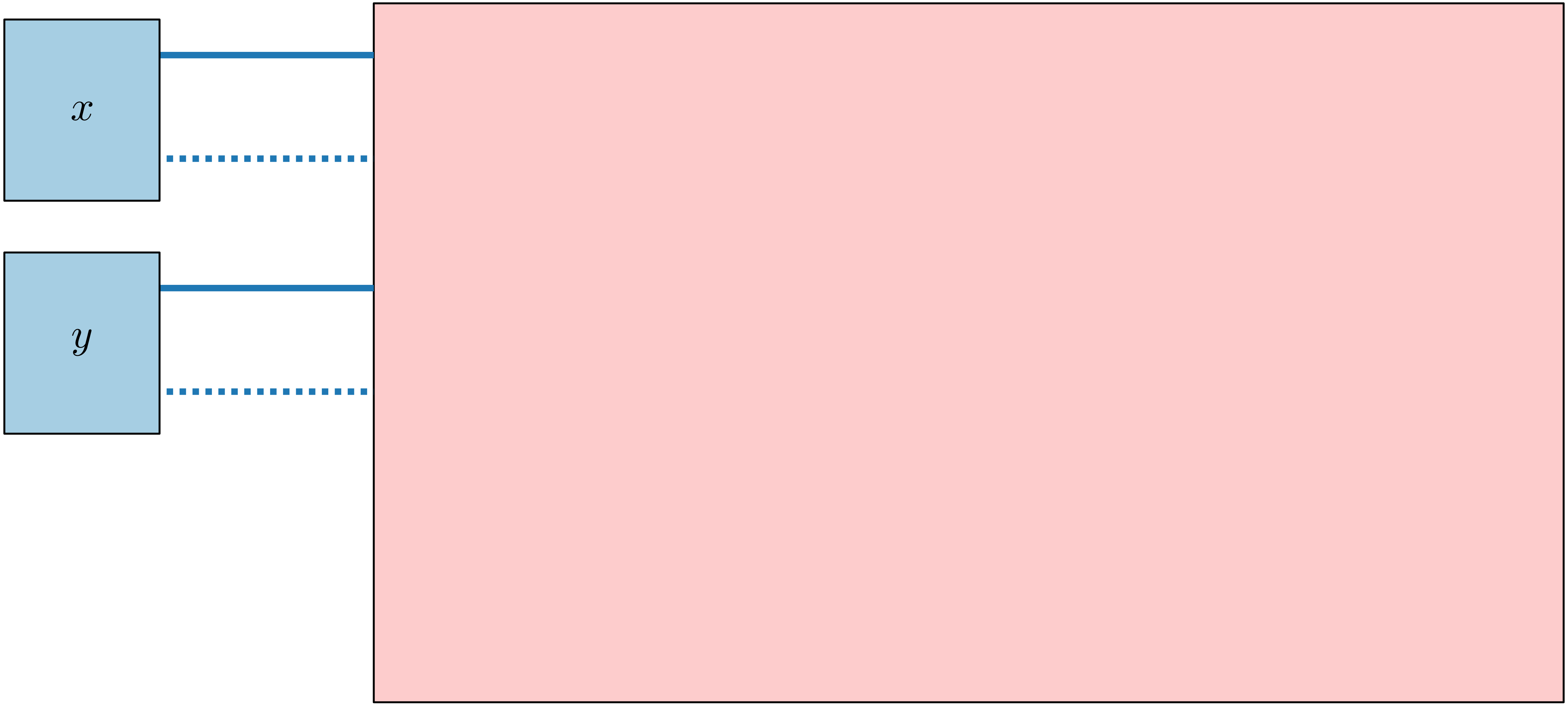
$x \vee y \vee z$

Clause Gadget

$x \vee y \vee z$

Clause Gadget

$x \vee y \vee z$

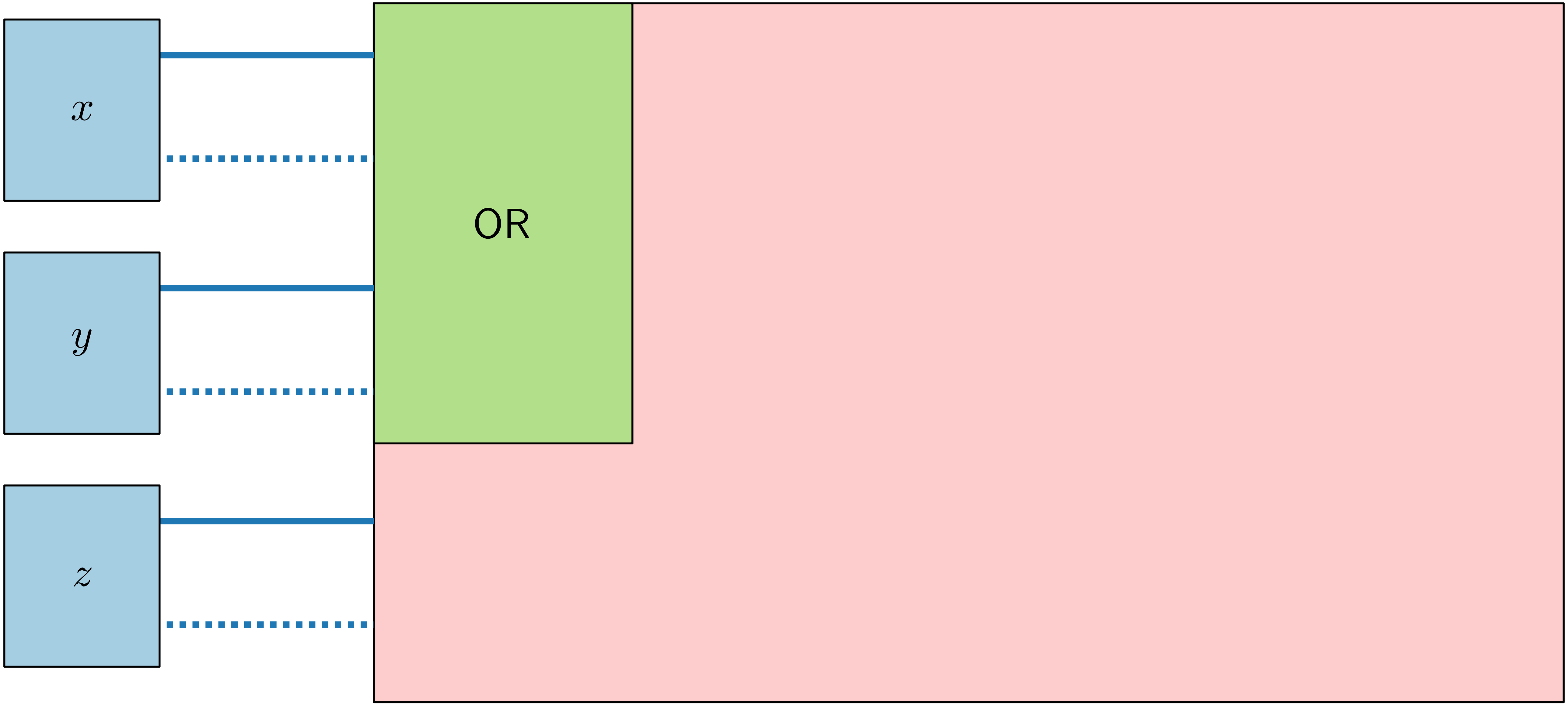


Clause Gadget

$x \vee y \vee z$

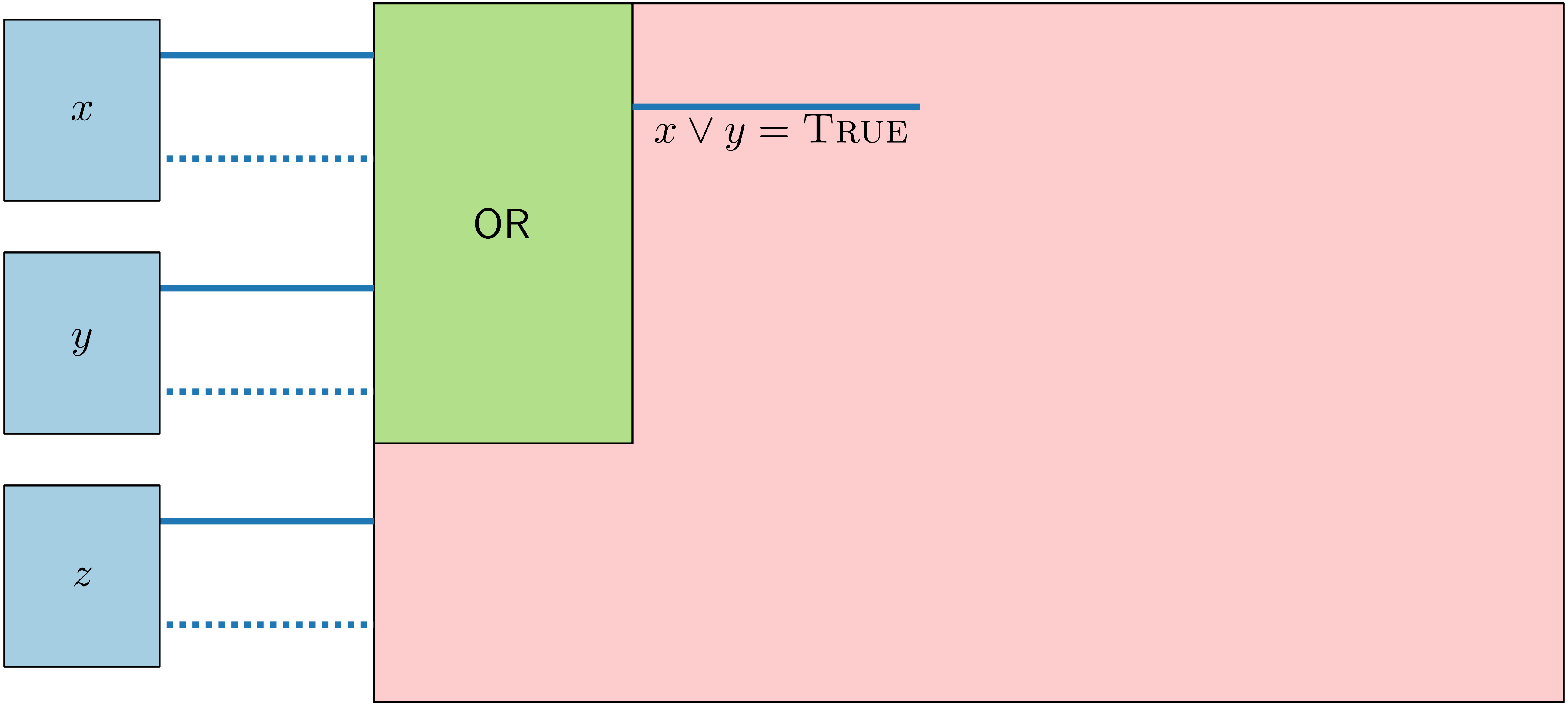
Clause Gadget

$x \vee y \vee z$

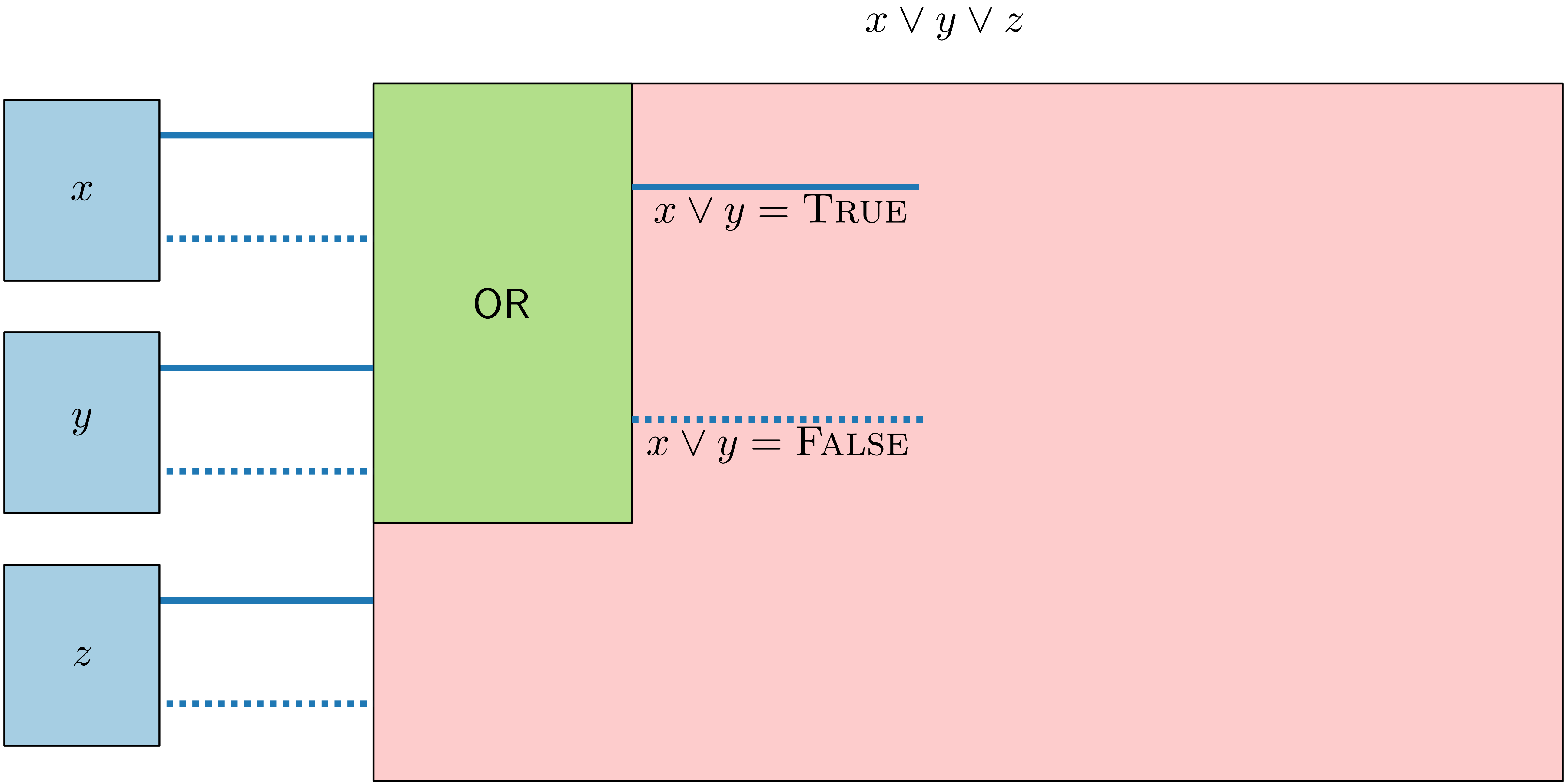


Clause Gadget

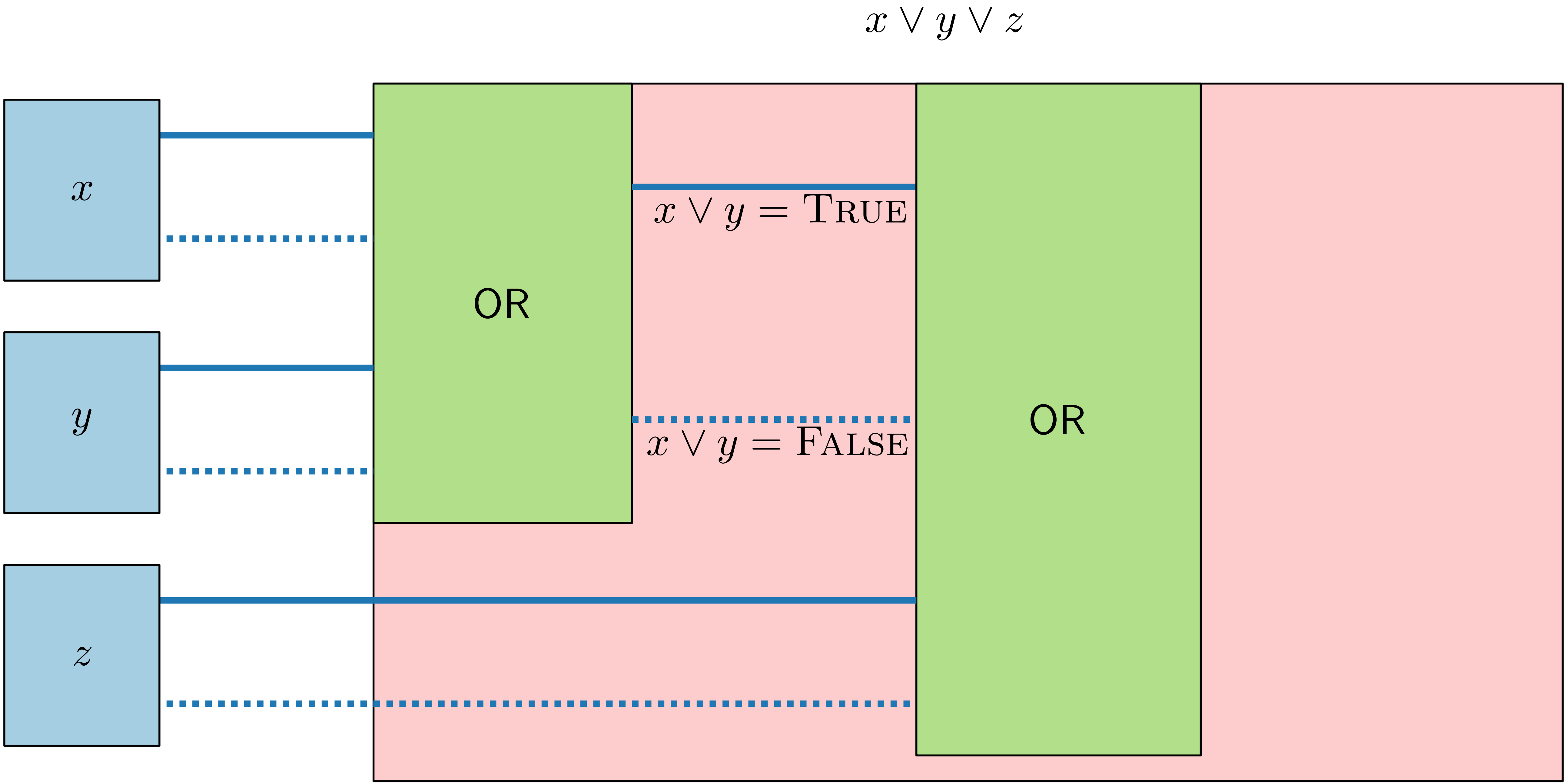
$x \vee y \vee z$



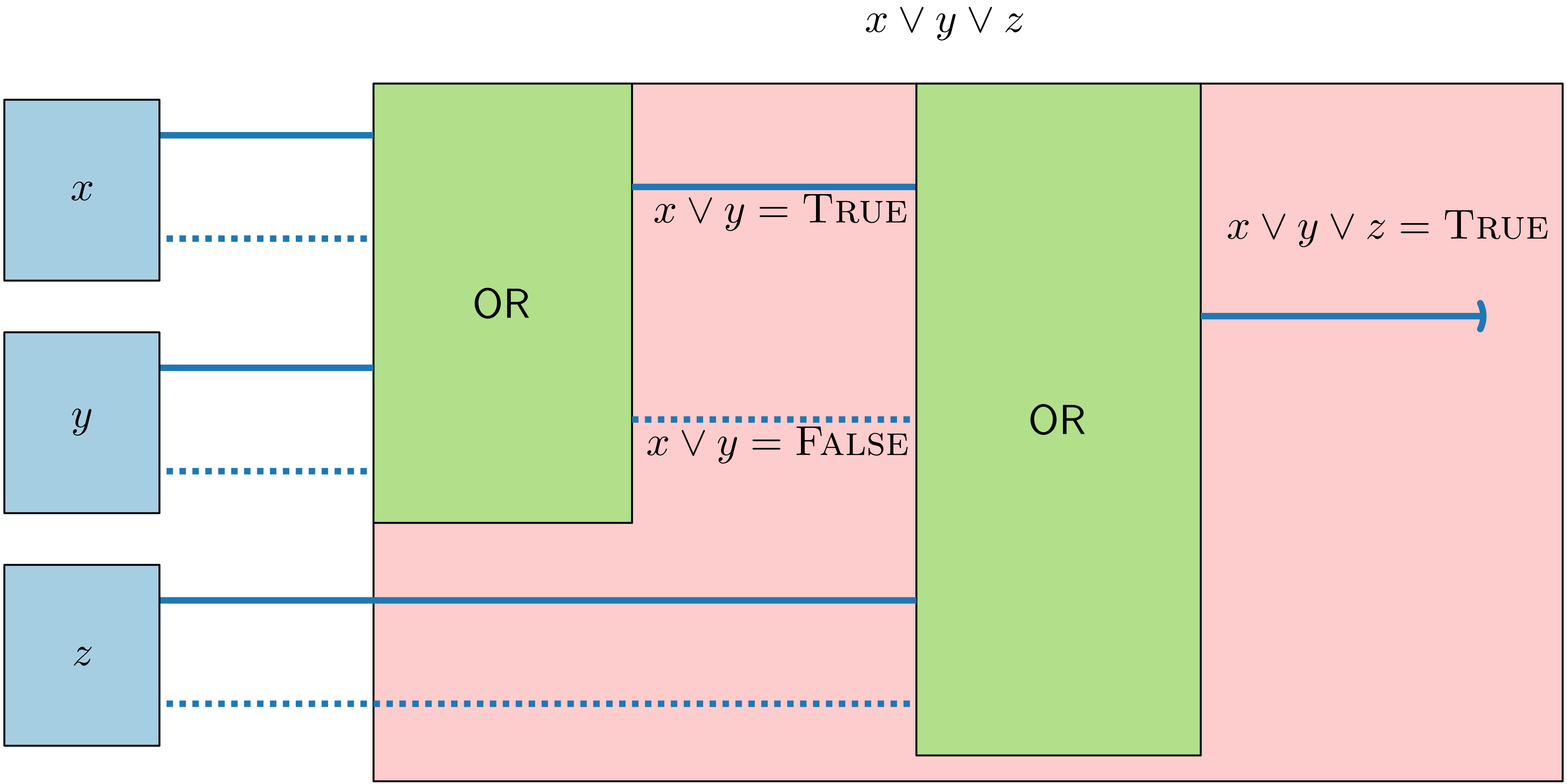
Clause Gadget



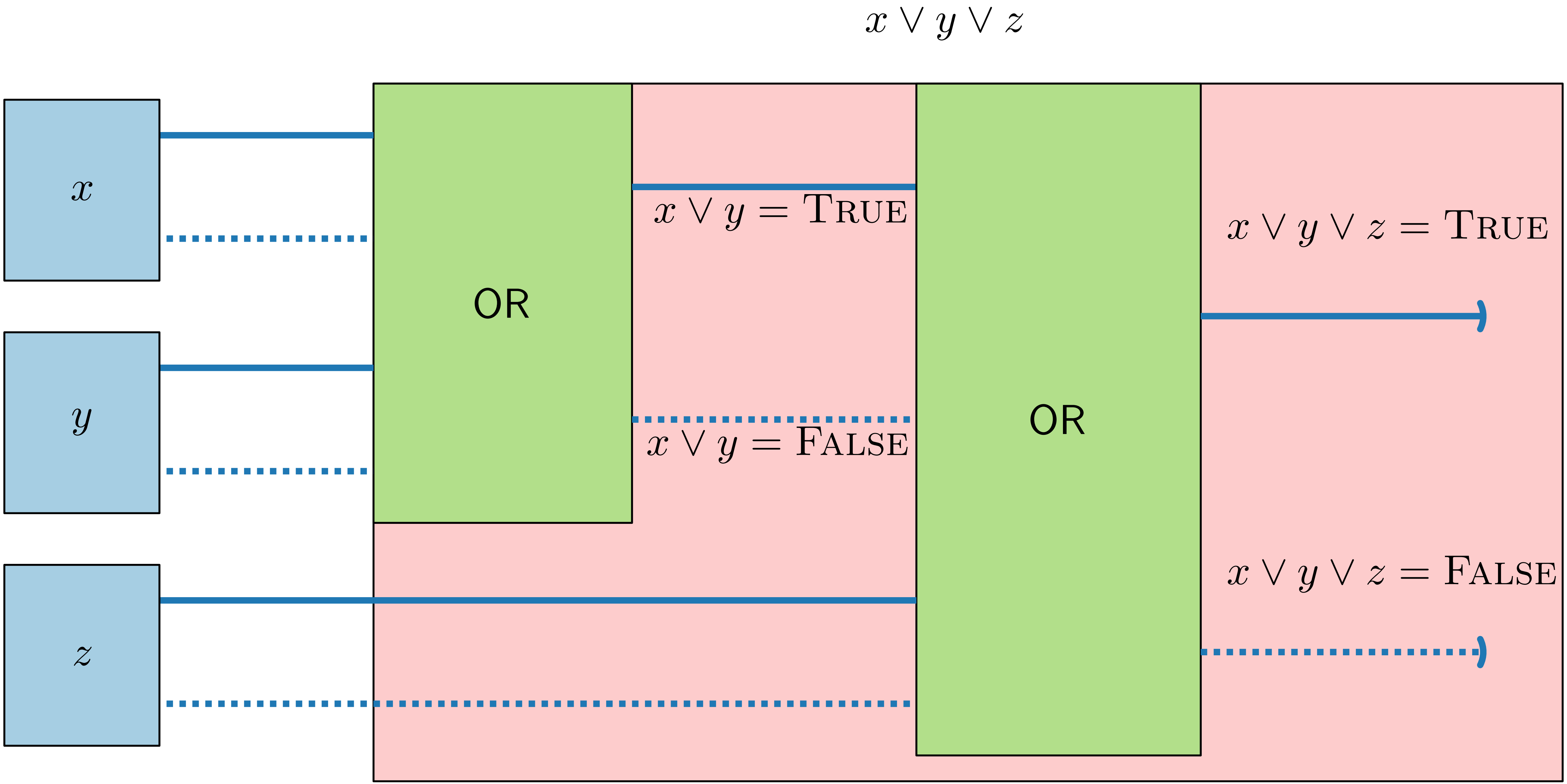
Clause Gadget



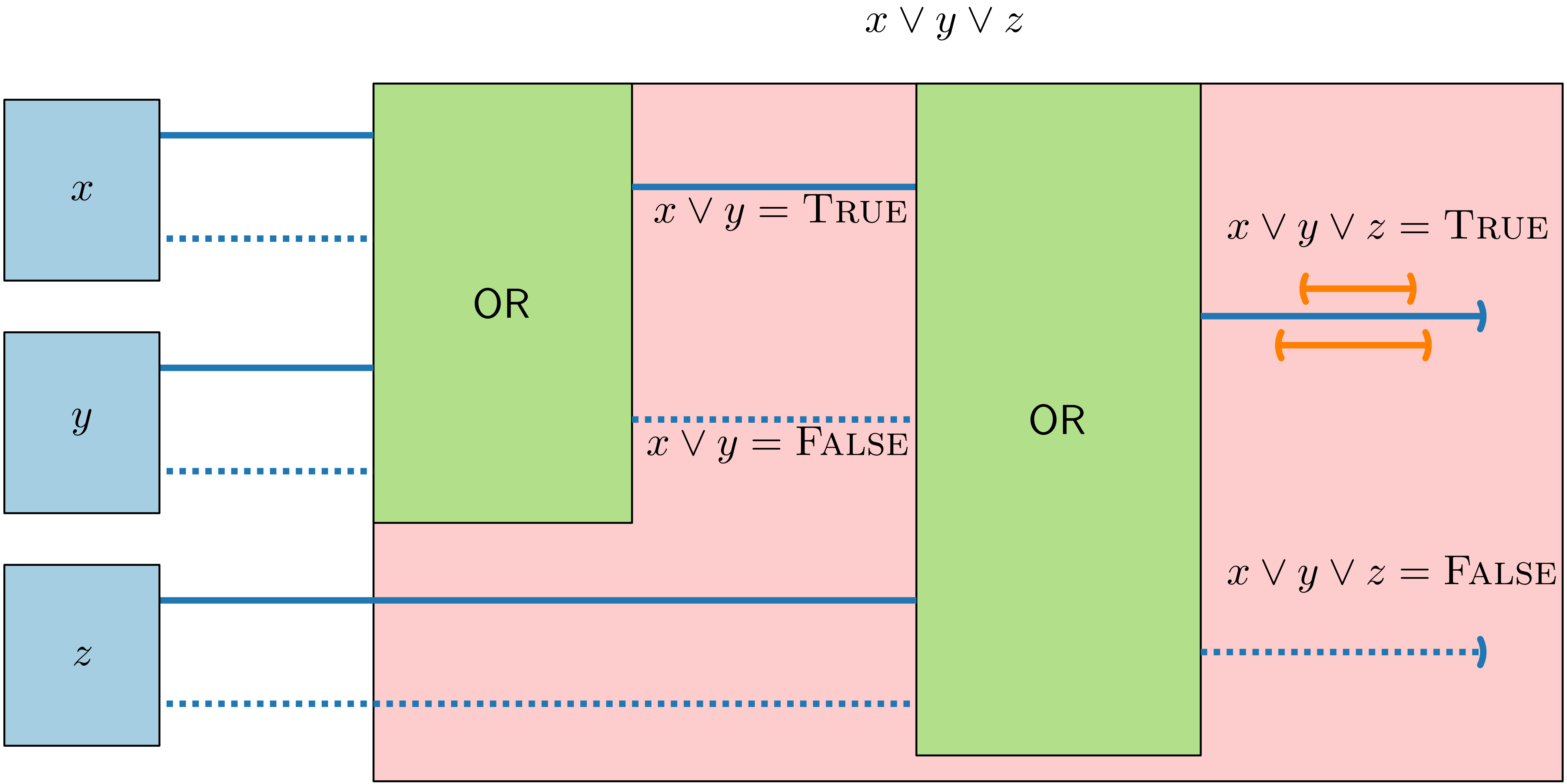
Clause Gadget



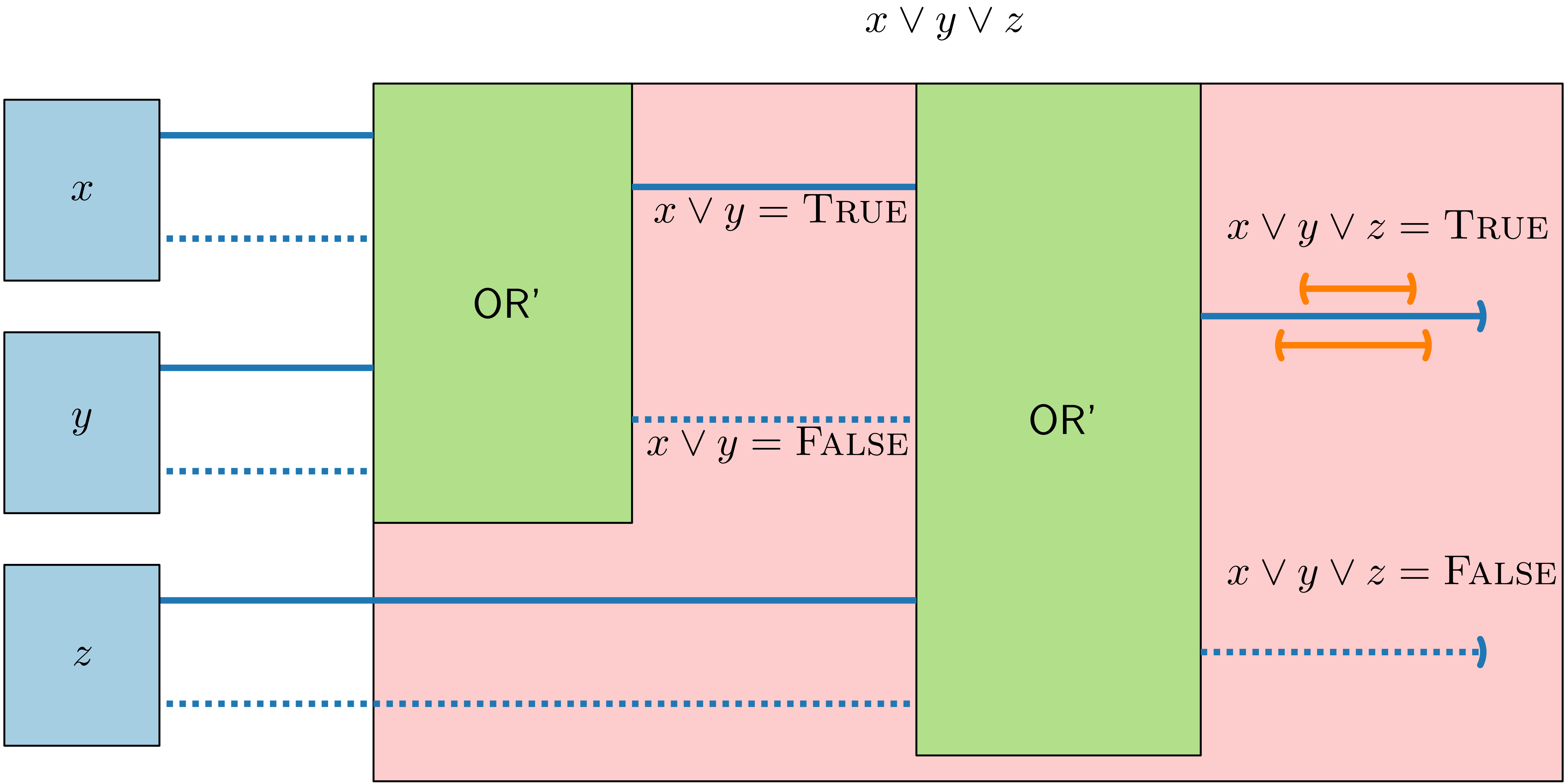
Clause Gadget



Clause Gadget

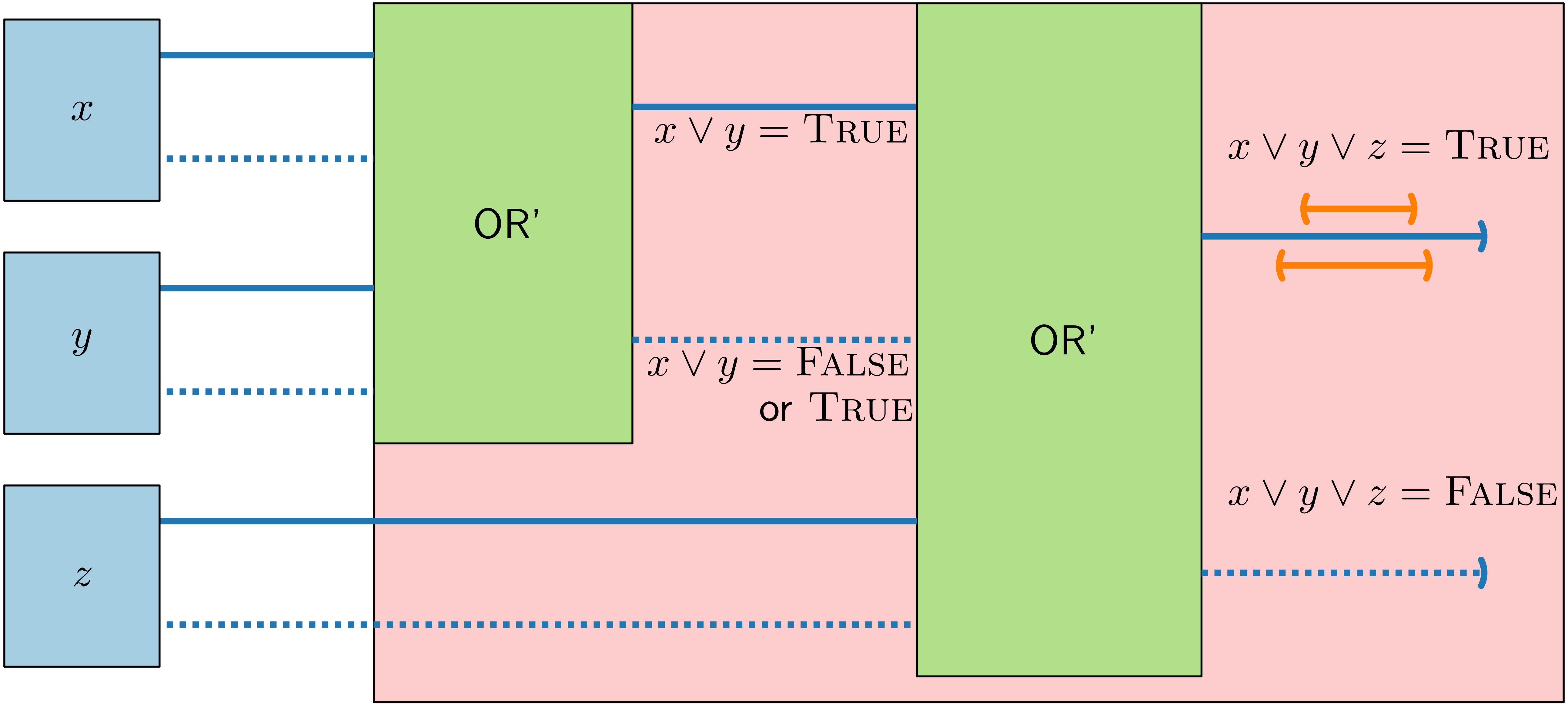


Clause Gadget

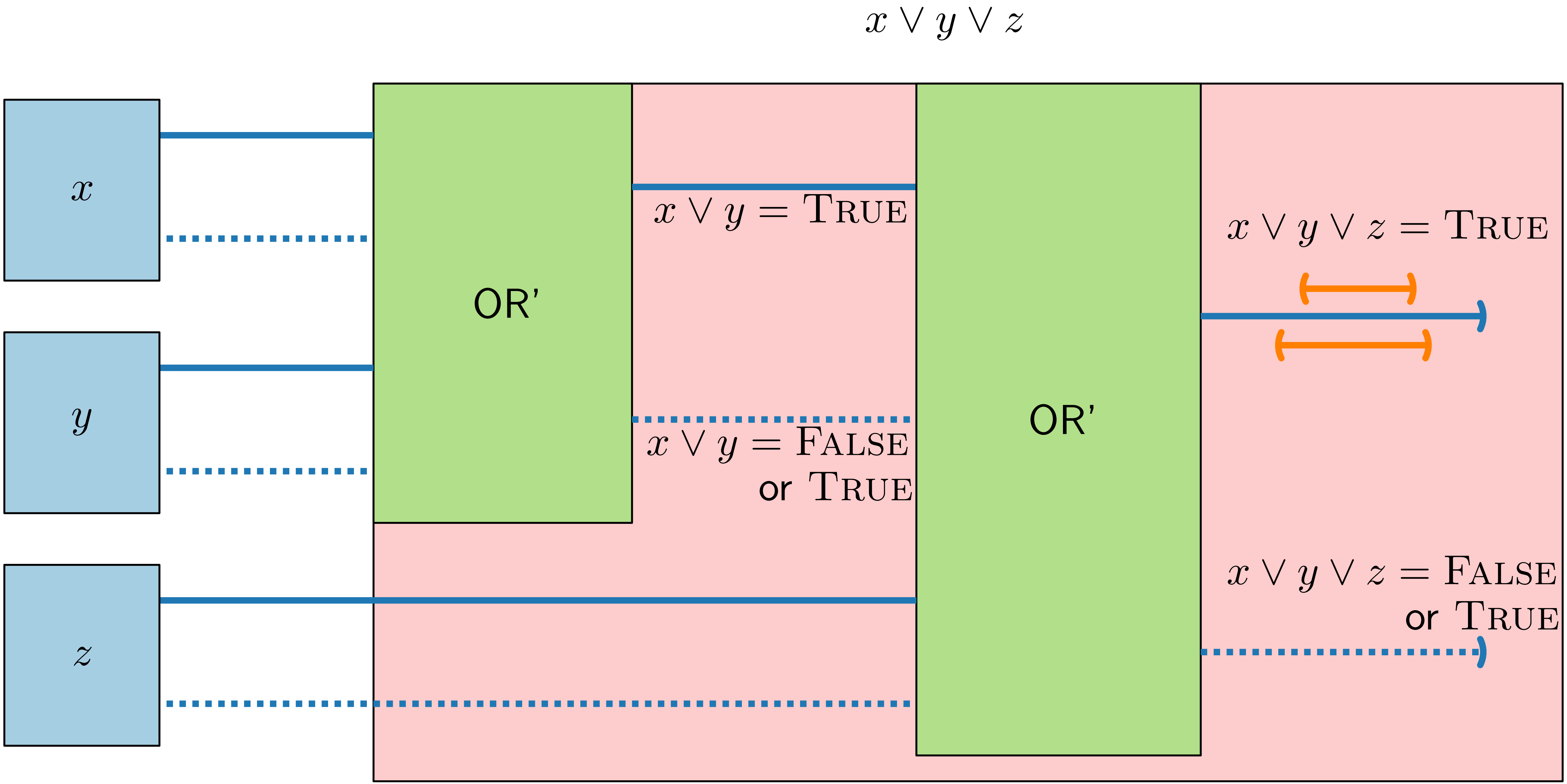


Clause Gadget

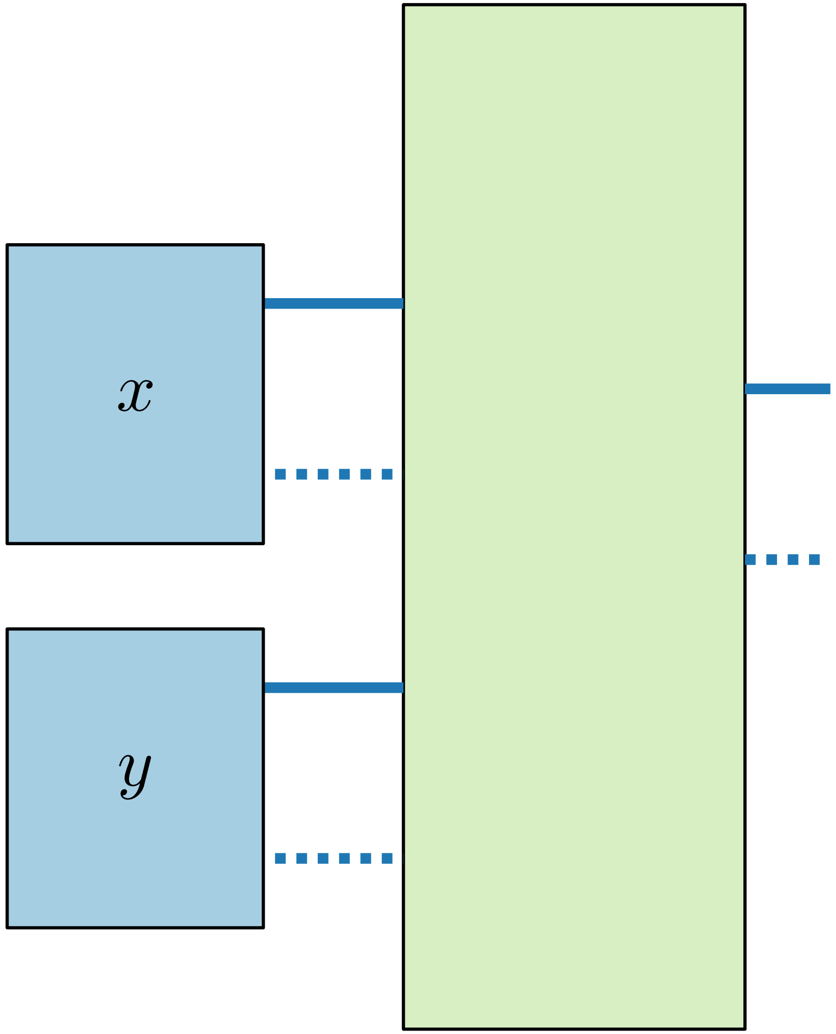
$x \vee y \vee z$



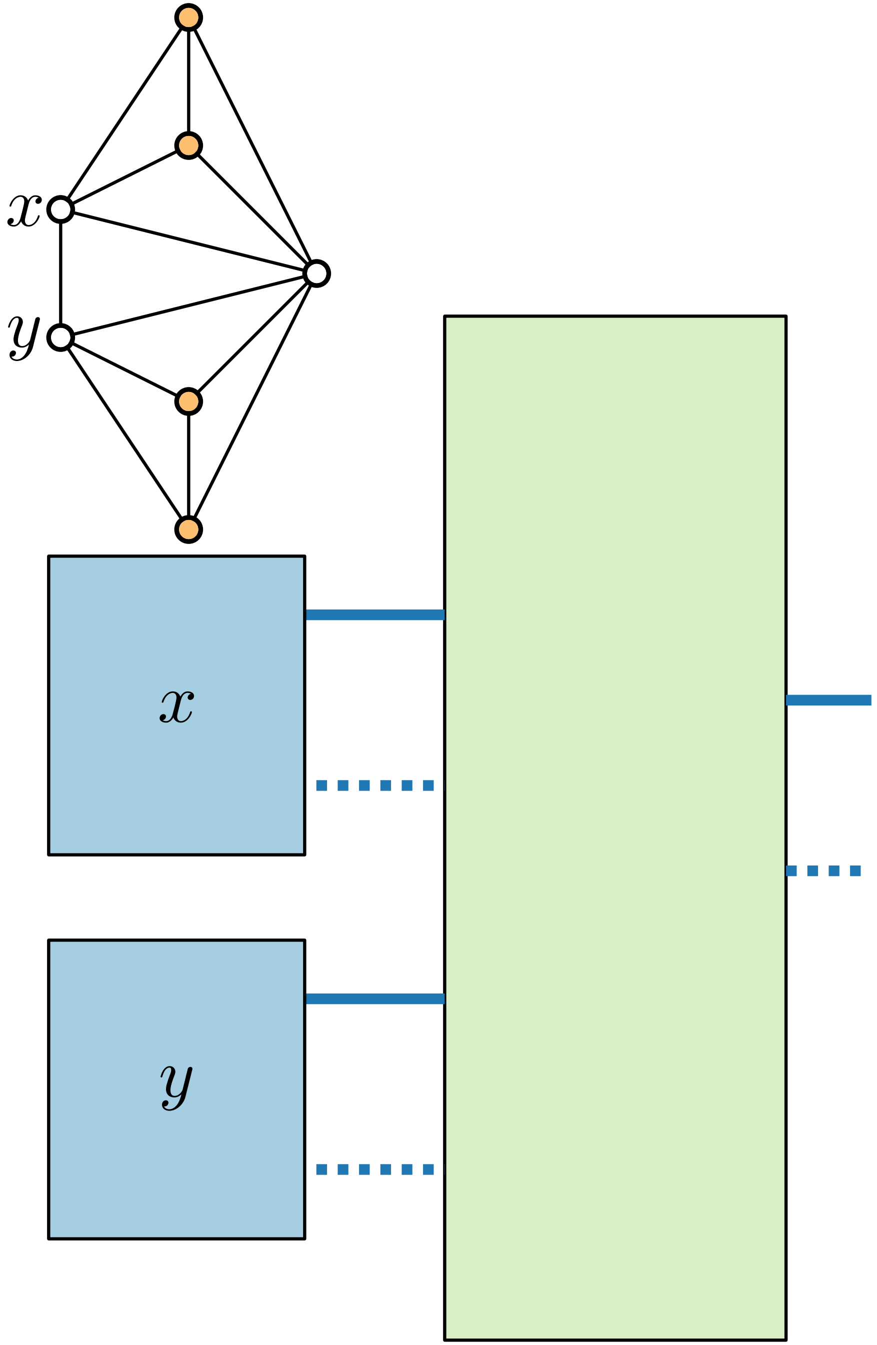
Clause Gadget



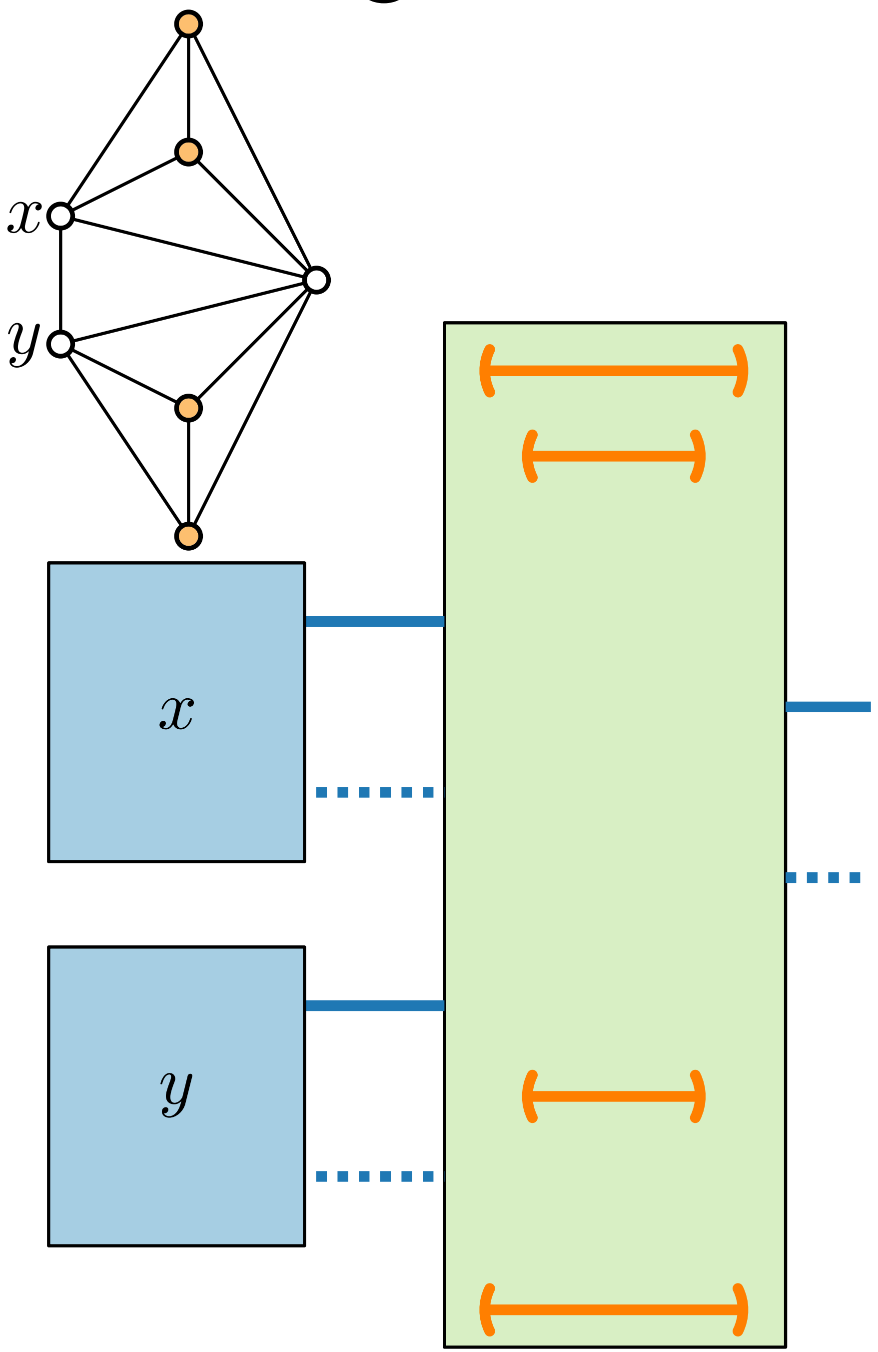
OR' Gadget



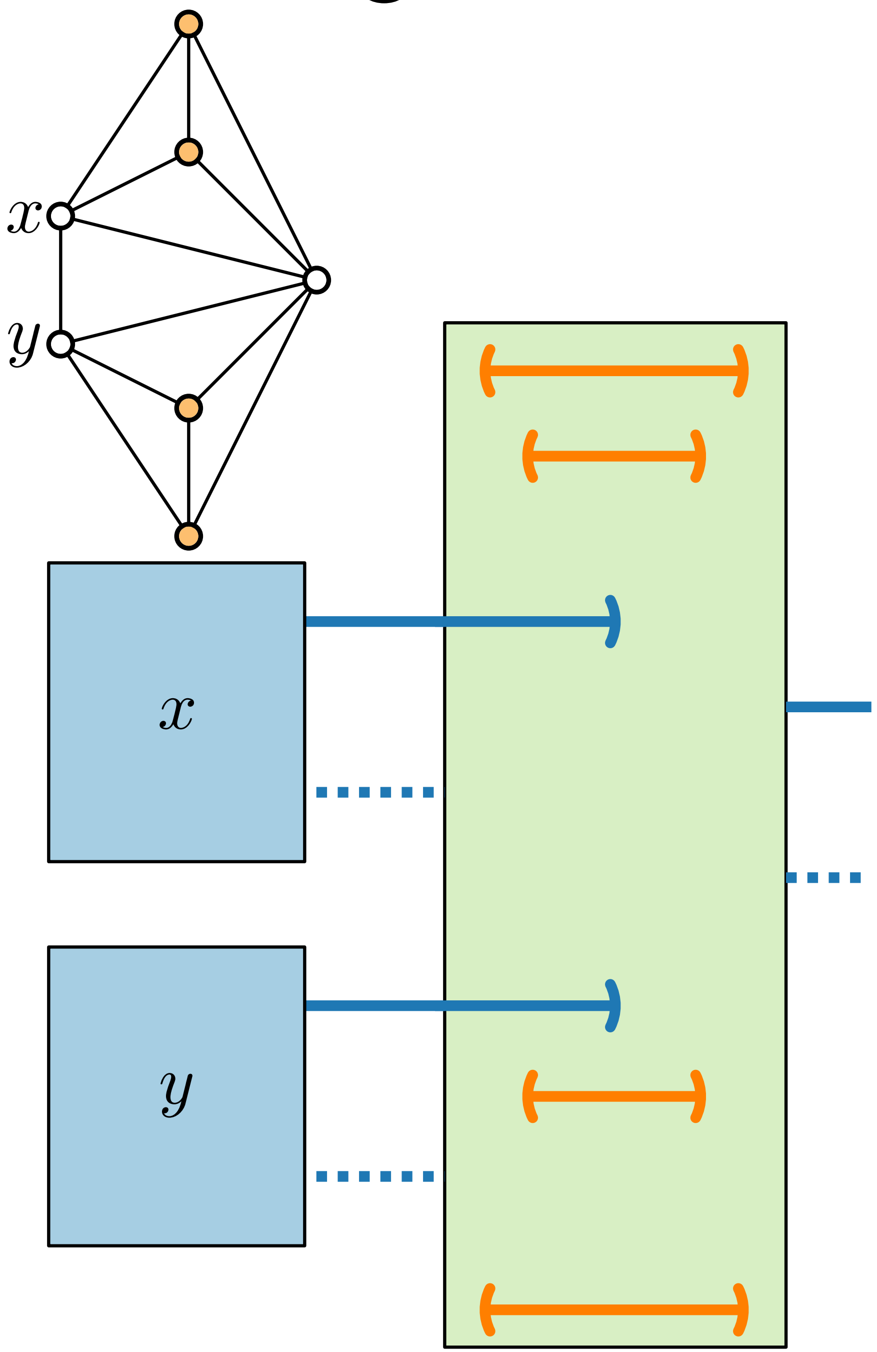
OR' Gadget



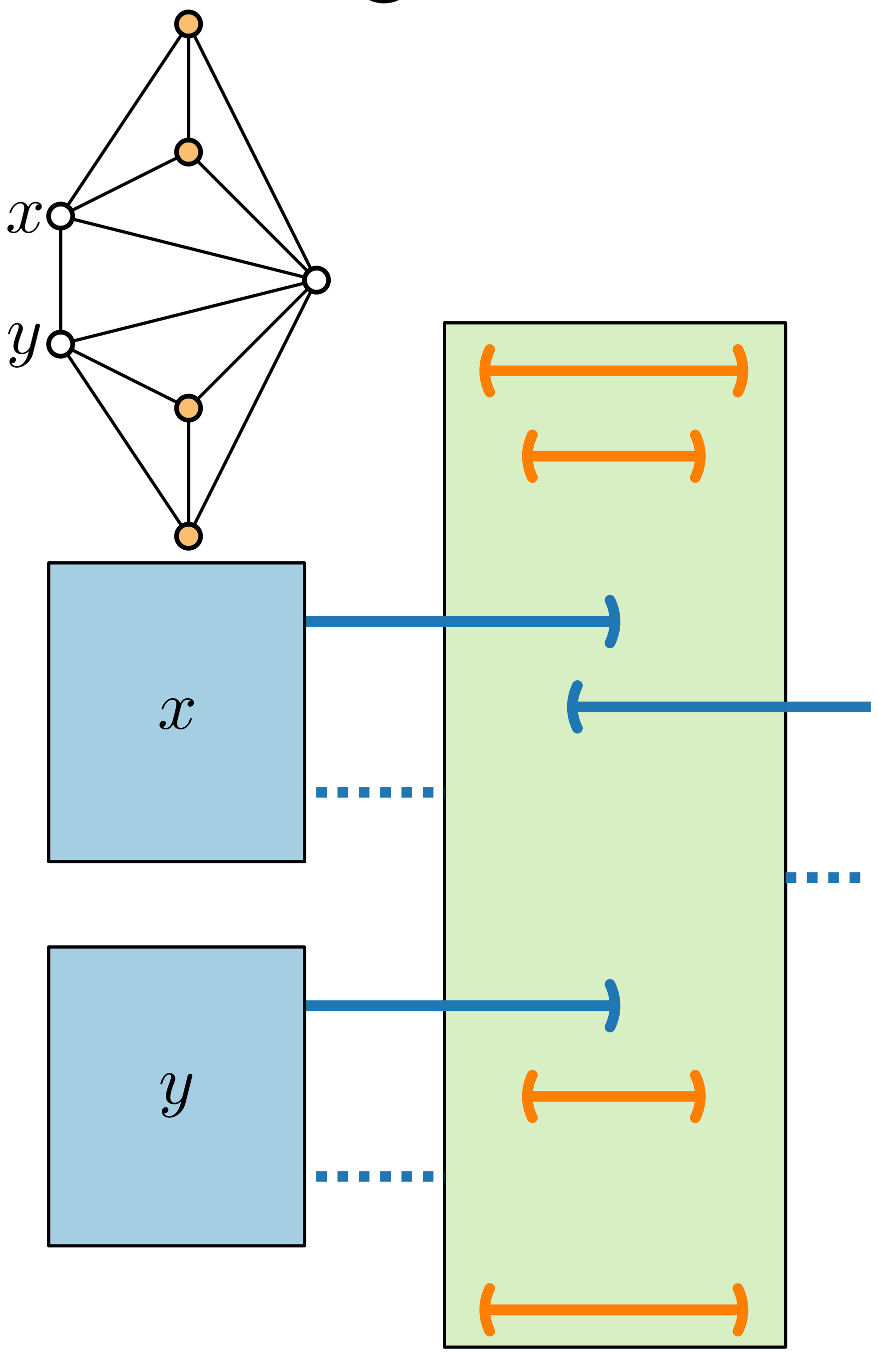
OR' Gadget



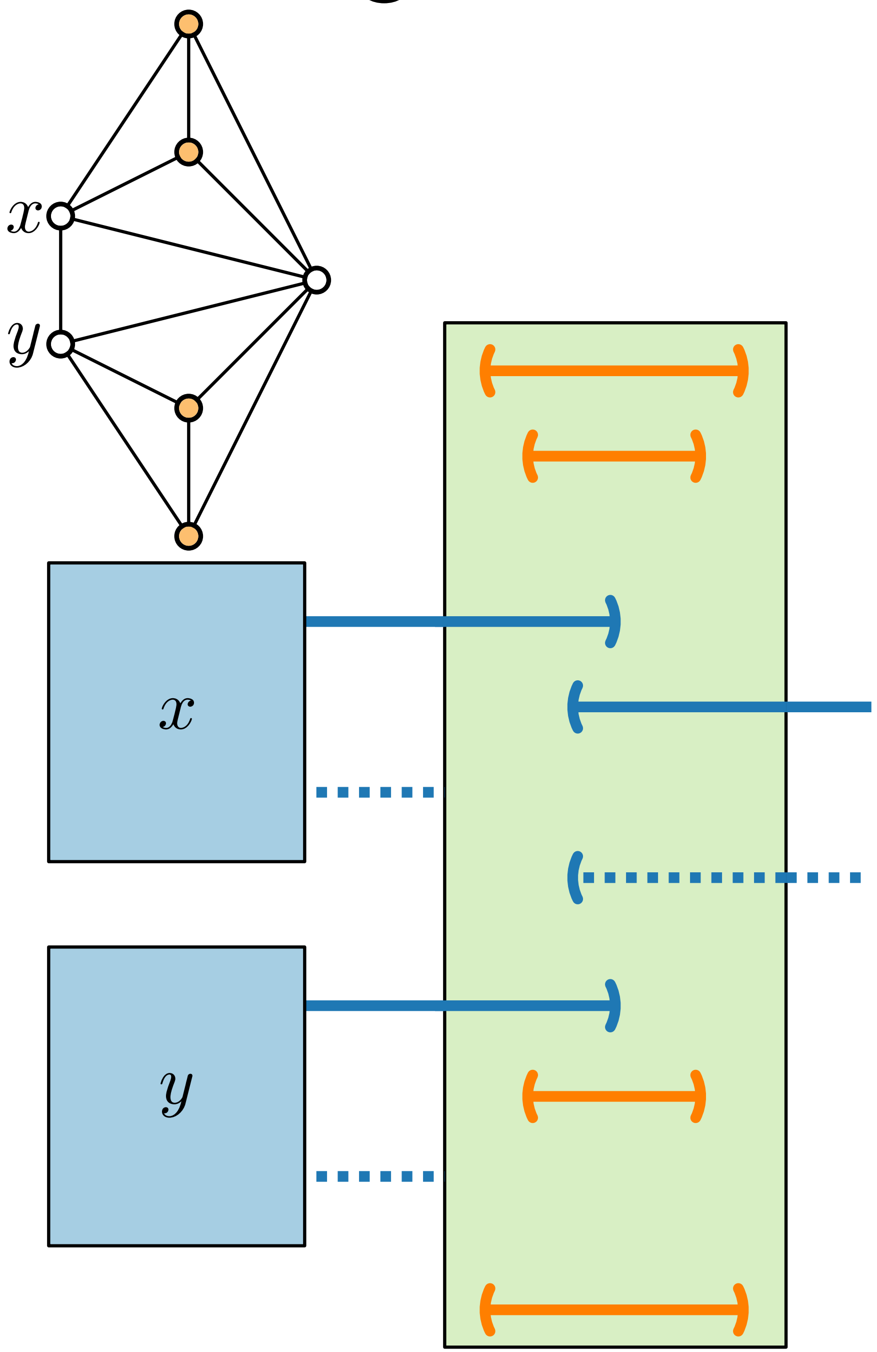
OR' Gadget



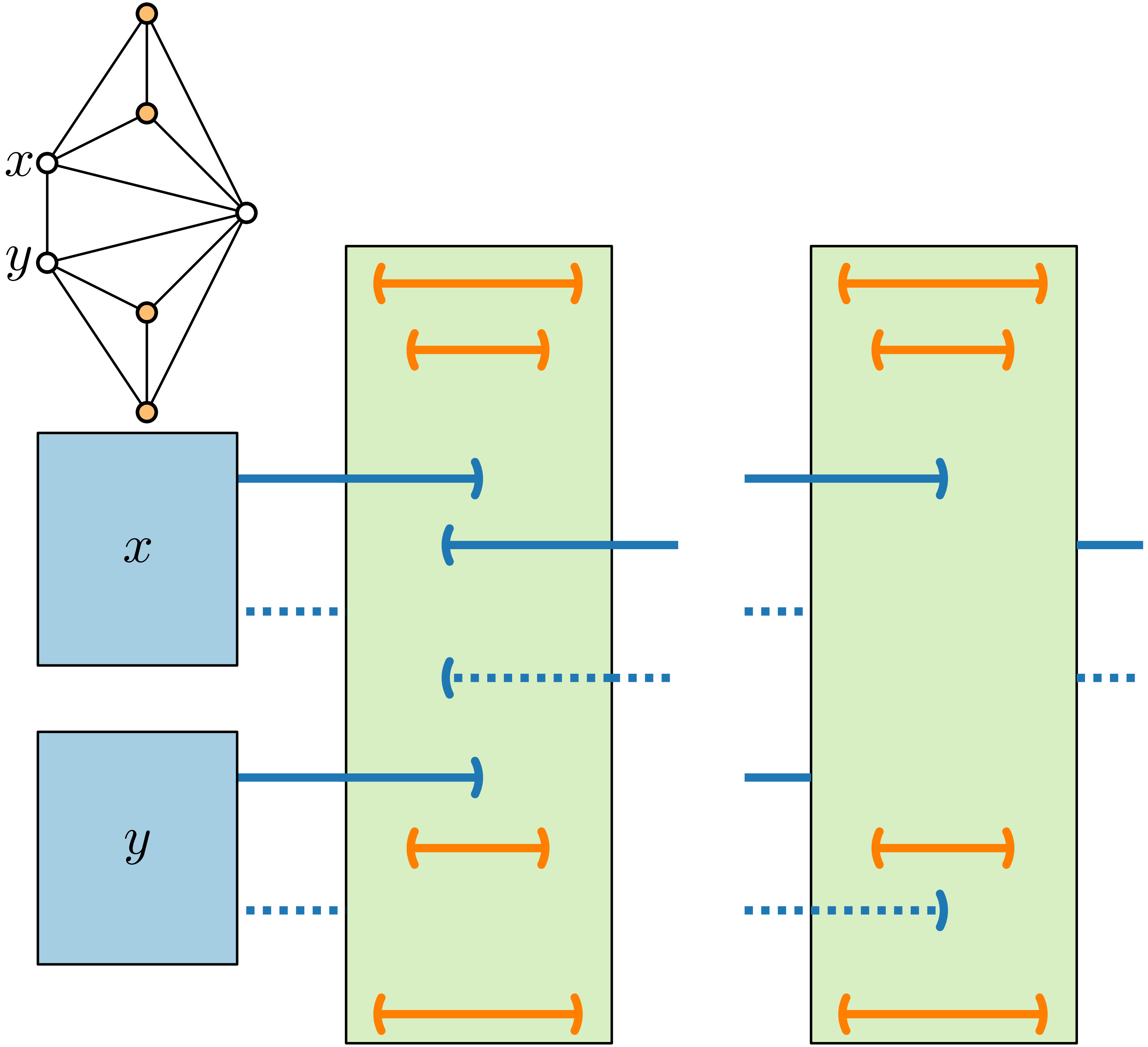
OR' Gadget



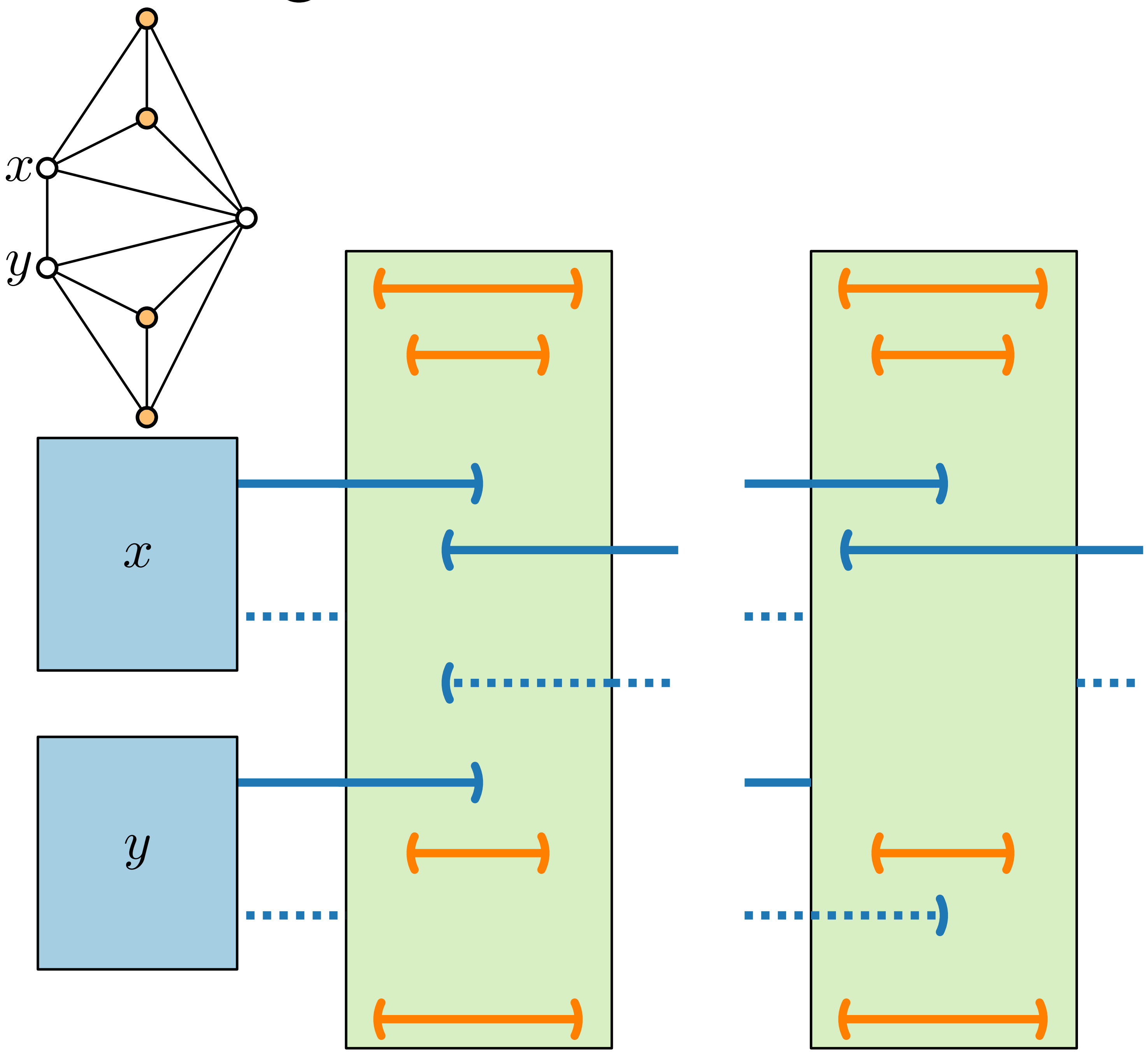
OR' Gadget



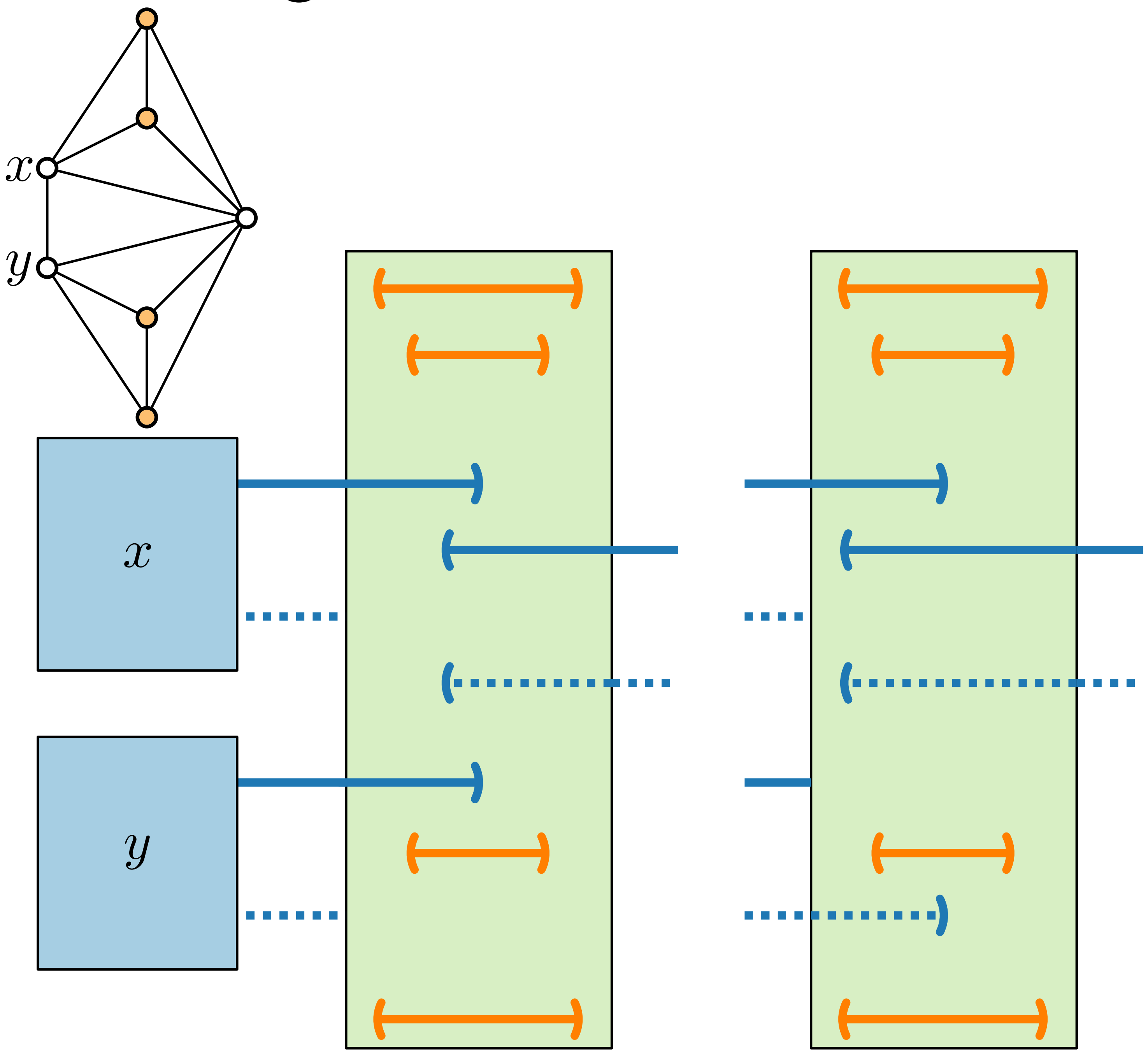
OR' Gadget



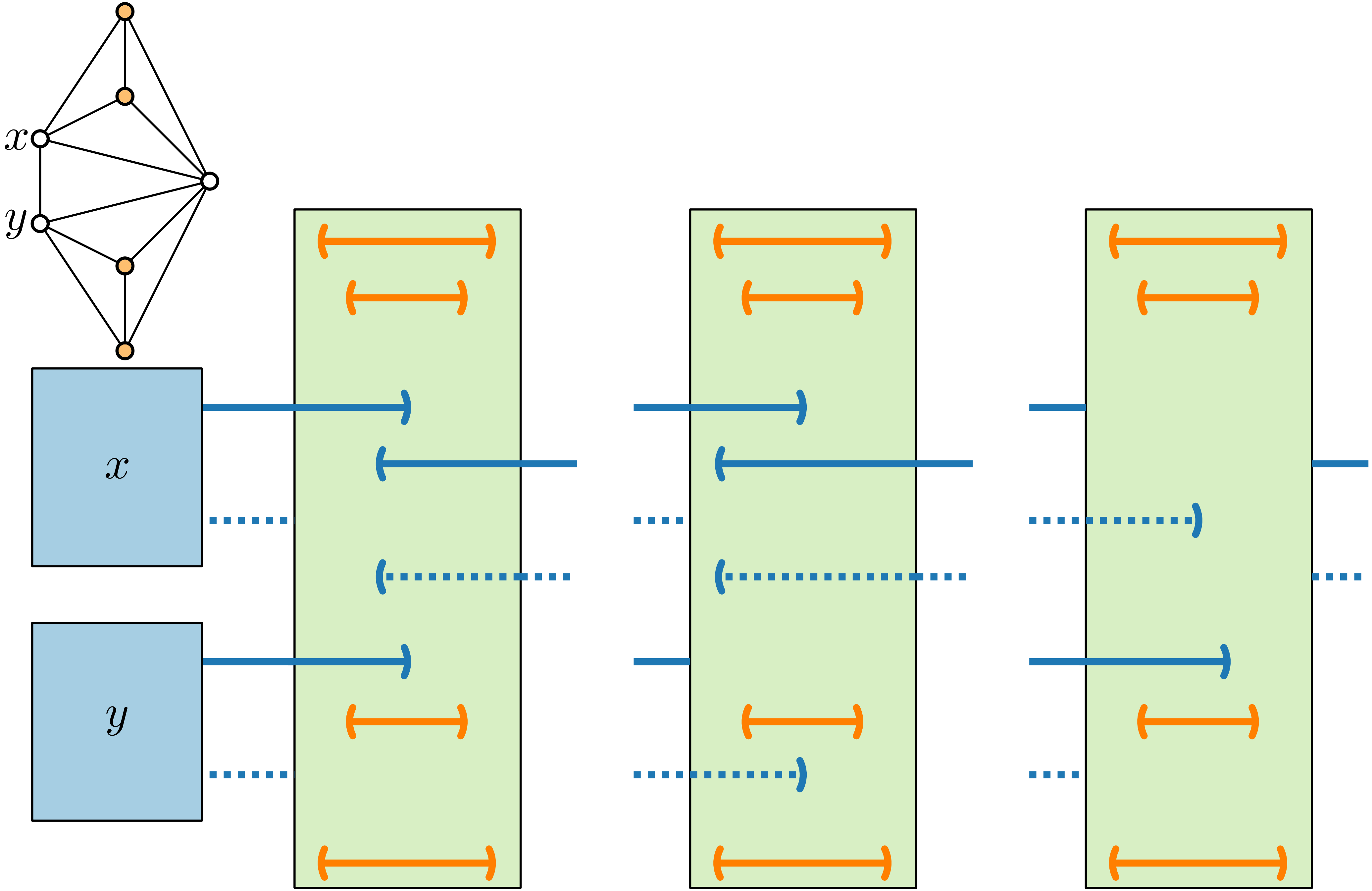
OR' Gadget



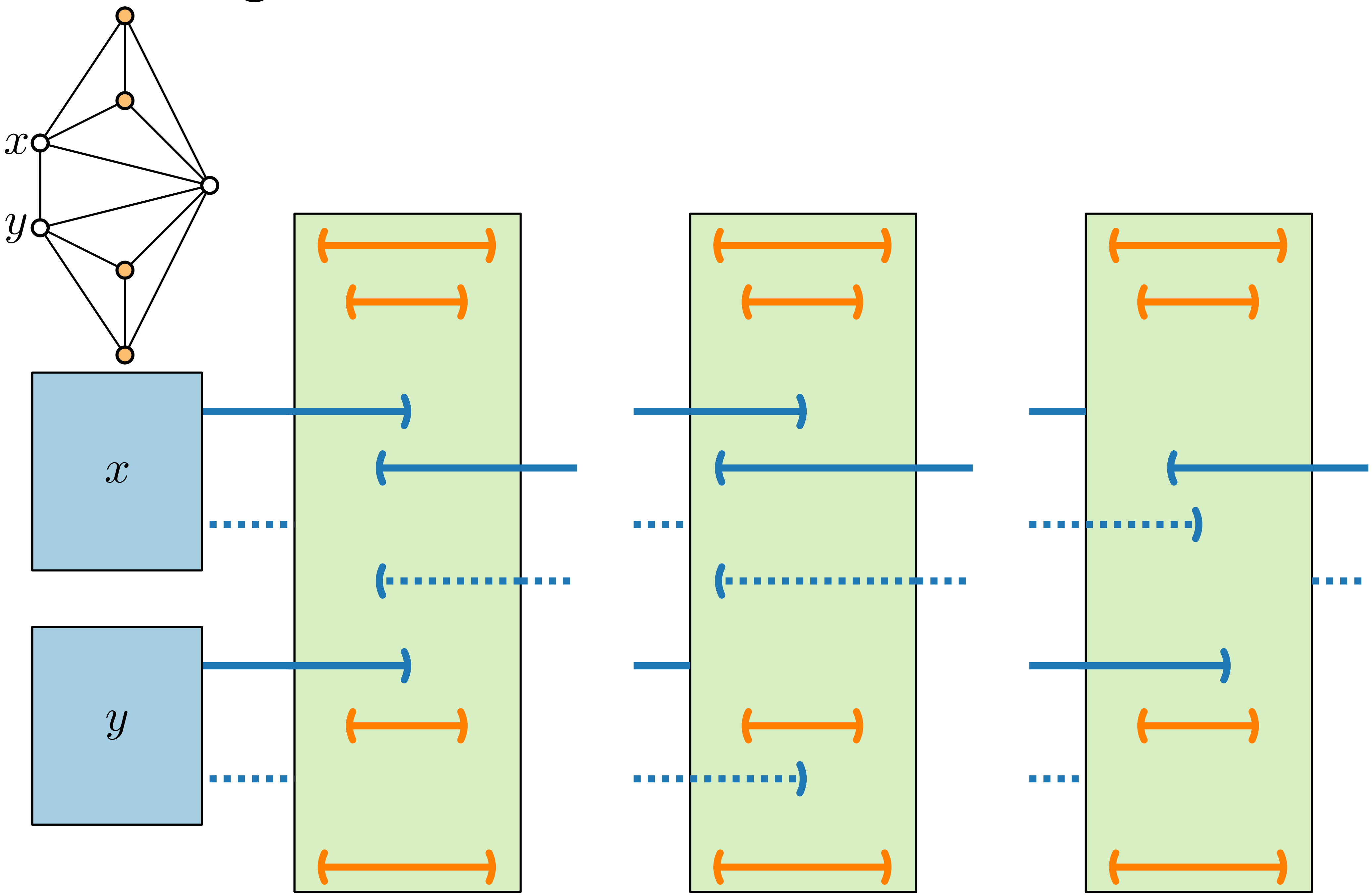
OR' Gadget



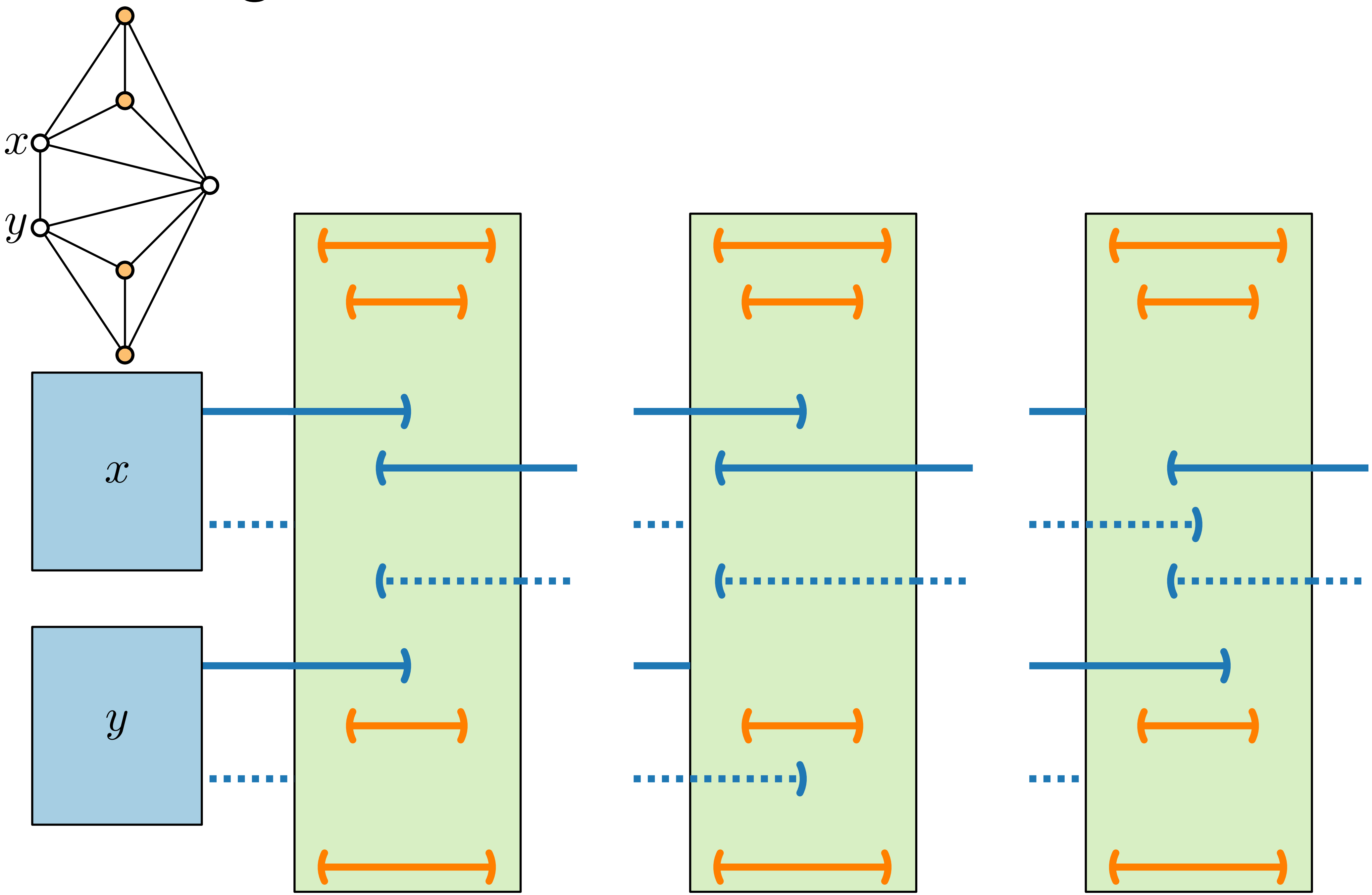
OR' Gadget



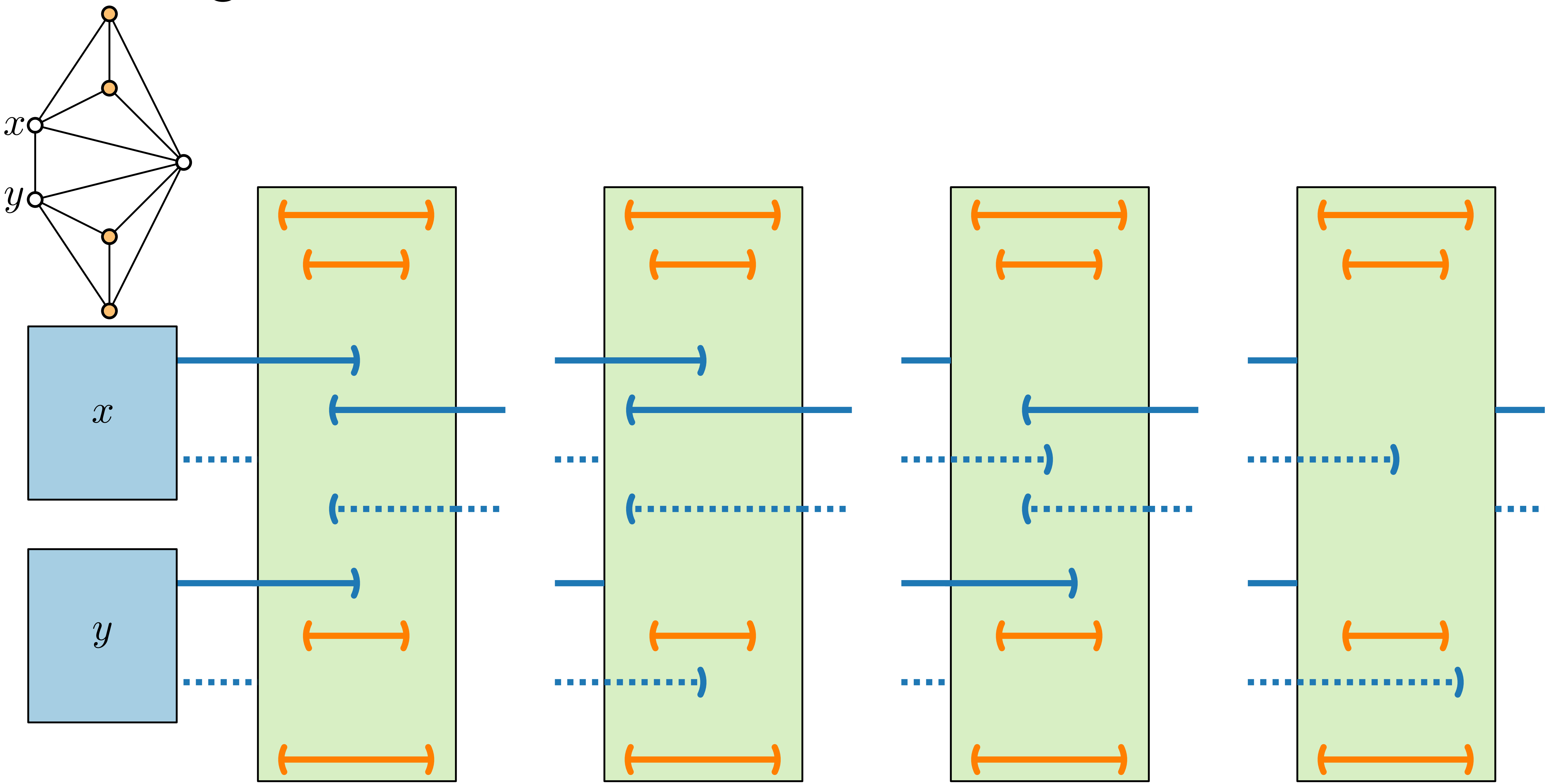
OR' Gadget



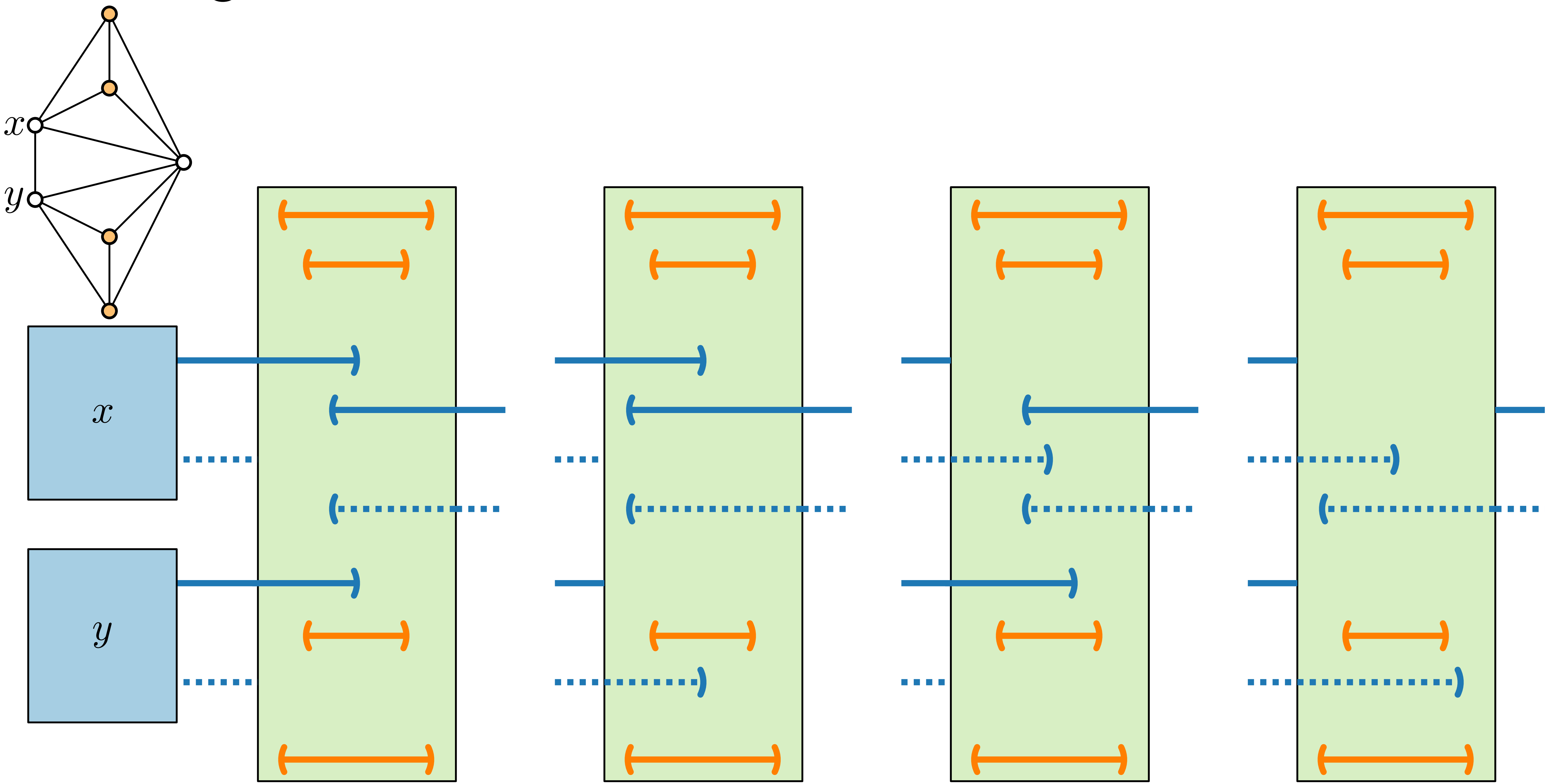
OR' Gadget



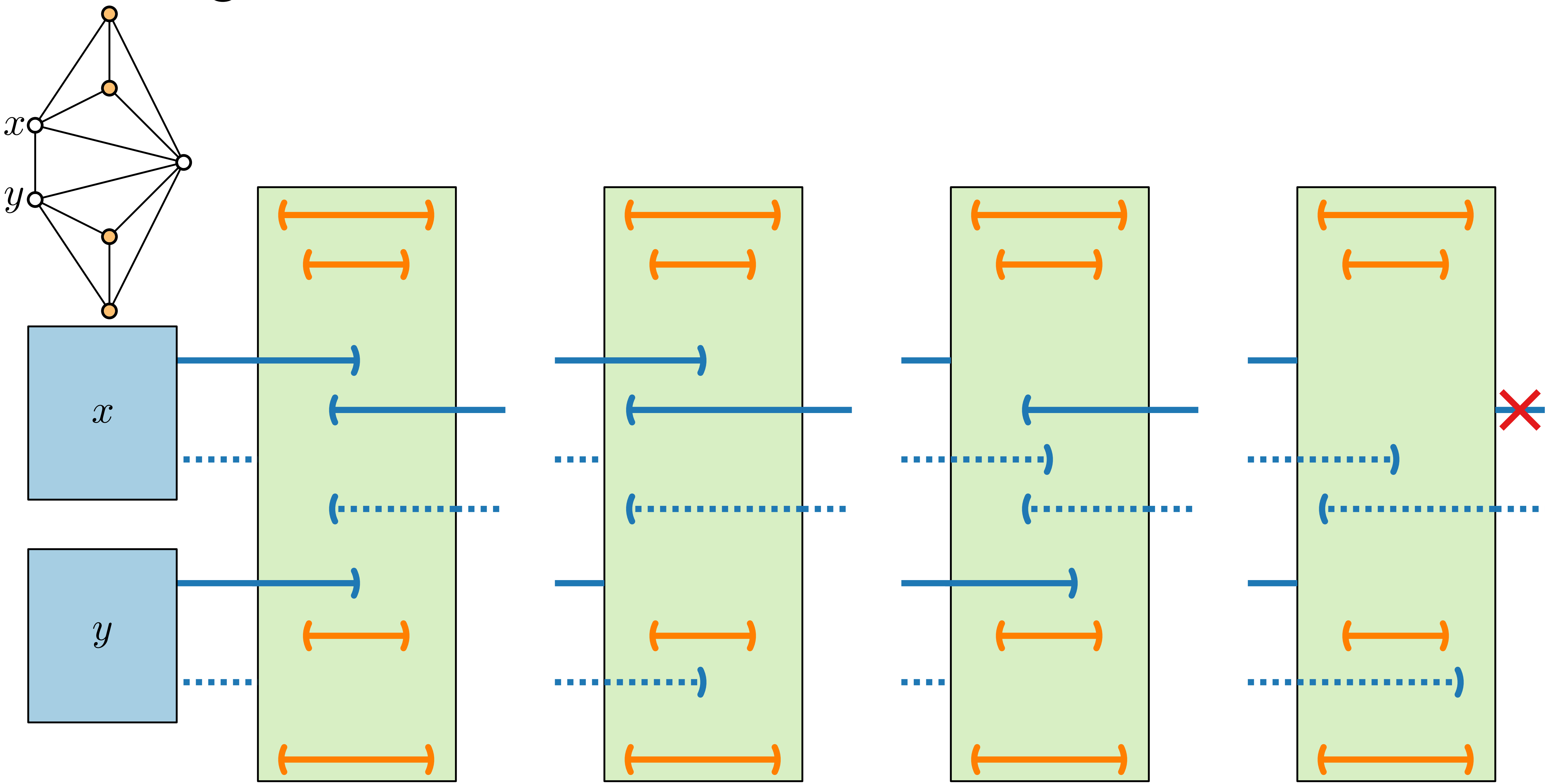
OR' Gadget



OR' Gadget



OR' Gadget



Discussion

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

Discussion

- *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.

Discussion

- *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.
- 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).

Discussion

- *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.
- 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 be solved in polynomial time for st-graphs? s?

Discussion

- *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.
- 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 be solved in polynomial time for st-graphs? For DAGs?

Discussion

- *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.
- 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 be solved in polynomial time for st-graphs? For DAGs?
- Can **Strong** Bar Visibility Recognition / Representation Extension can be solved in polynomial time for st-graphs?

Literature

Main source:

- [Chaplick, Guśpiel, Gutowski, Krawczyk, Liotta '18]
  The Partial Visibility Representation Extension Problem

Referenced papers:

- [Gutwenger, Mutzel '01] A Linear Time Implementation of SPQR-Trees
- [Wismath '85] Characterizing bar line-of-sight graphs
- [Tamassia, Tollis '86] Algorithms for visibility representations of planar graphs
- [Andreae '92] Some results on visibility graphs
- [Chaplick, Dorbec, Kratchovíl, Montassier, Stacho '14]
  Contact representations of planar graphs: Extending a partial representation is hard