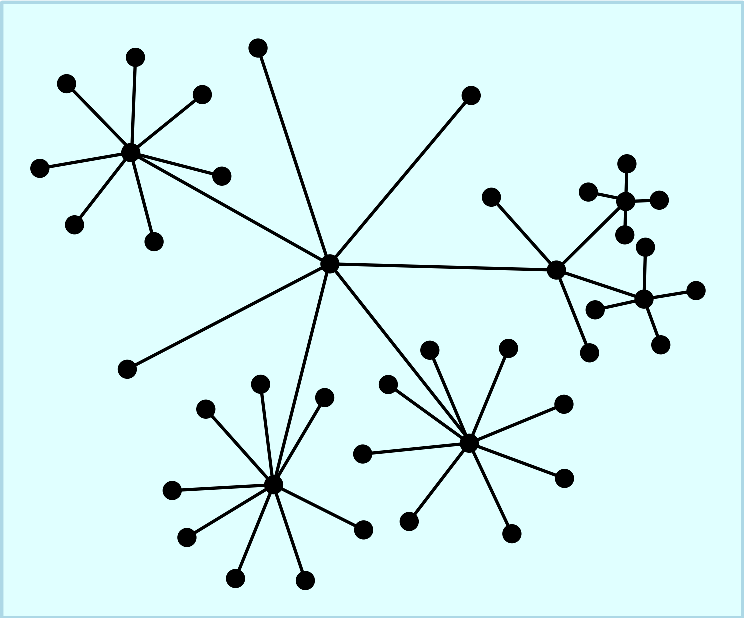
Karlsruhe Institute of Technology

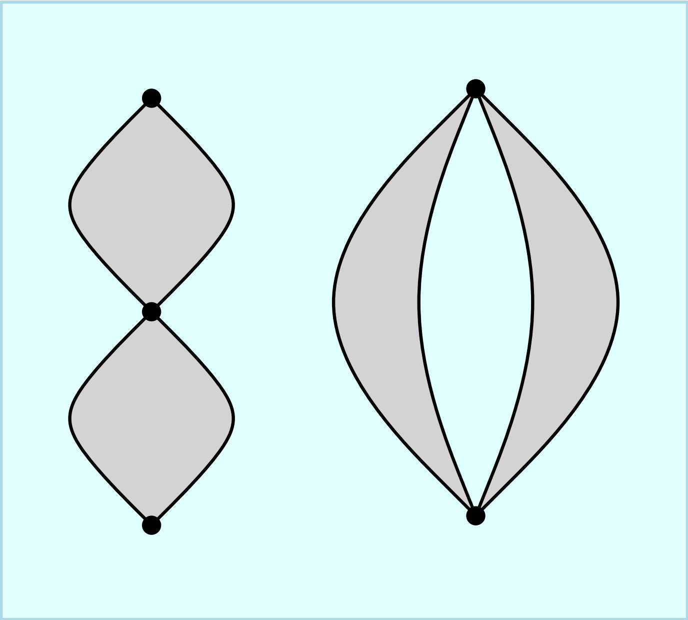
# Algorithms for graph visualization

## Divide and Conquer - Trees and Series-Parallel Graphs

WINTER SEMESTER 2014/2015

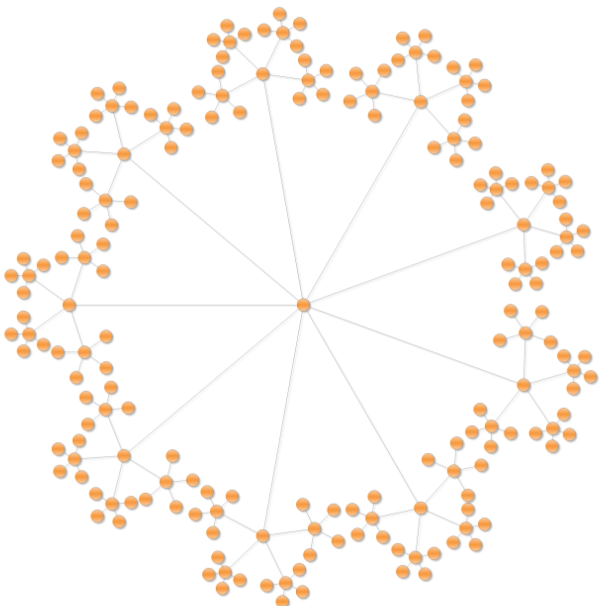
**Tamara Mchedlidze** – MARTIN NÖLLENBURG





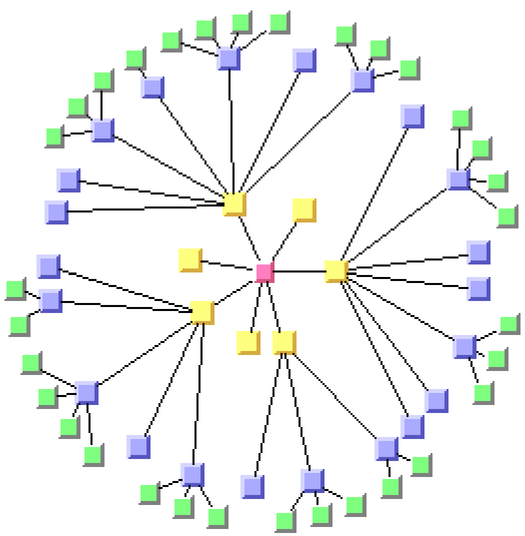
# Baloon Layout

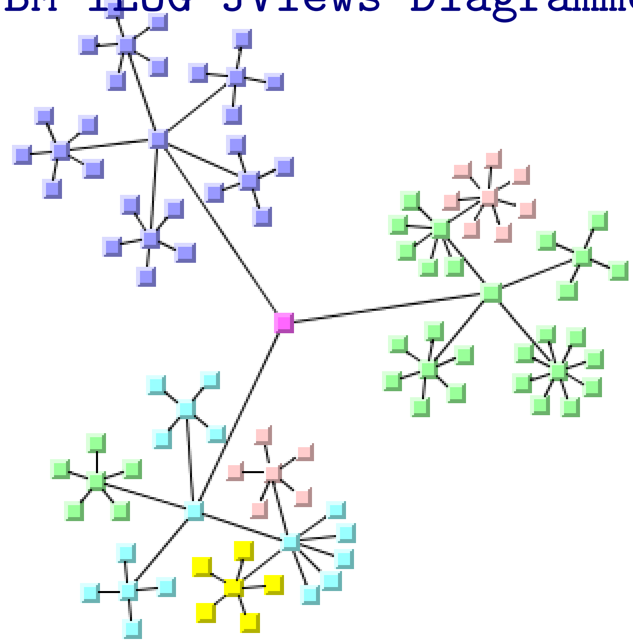
Karlsruhe Institute of Technology

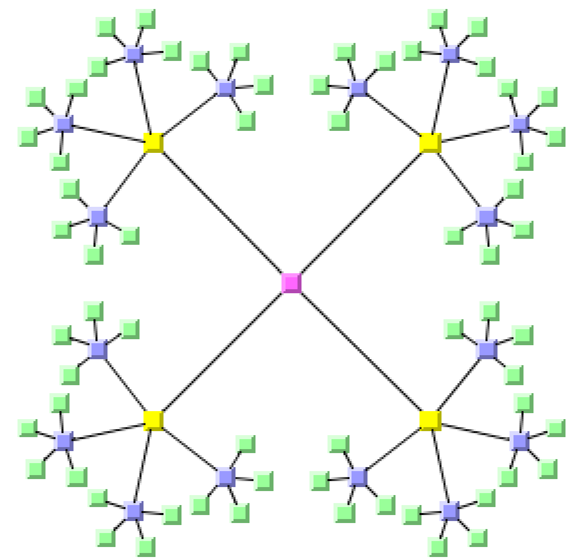


NEVRON-Visualize your success: www.nevron.com


IBM ILOG JViews Diagrammer: www.ibm.com





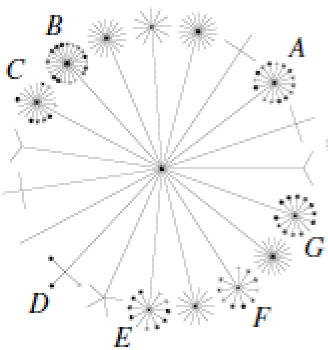


# Baloon Layout

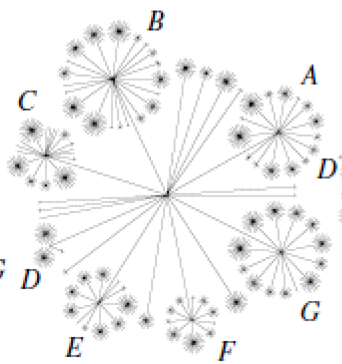
Karlsruhe Institute of Technology

## A baloon drawing has the following properties:

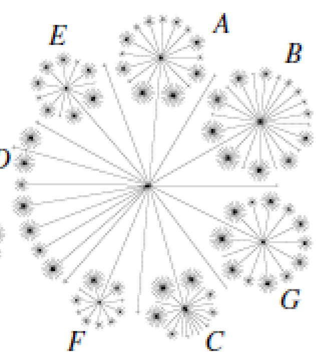
- All the children of the same parent lie on circle centered at their parent
- The drawing is planar
- The further an edge from the root is, the shorter it becomes




All subtrees at the same depth have the same size.




Subtrees may have different size, the tree is ordered.

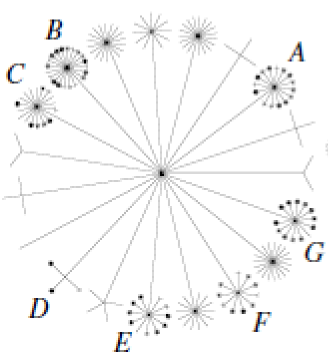



Subtrees may have different size, the tree is unordered. Drawn by Lin& Yen Algorithm.

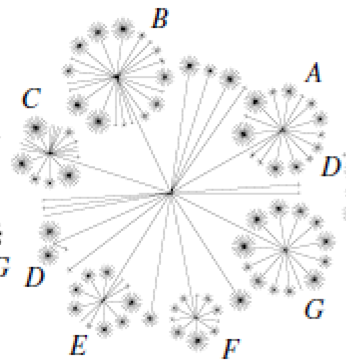
# Baloon Layout

## A baloon drawing has the following properties:

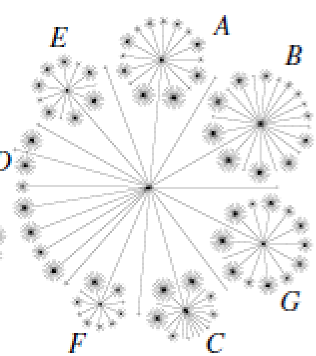
- All the children of the same parent lie on circle centered at their parent
- The drawing is planar
- The further an edge from the root is, the shorter it becomes




All subtrees at the same depth have the same size.



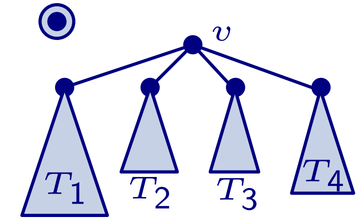

Subtrees may have different size, the tree is ordered.

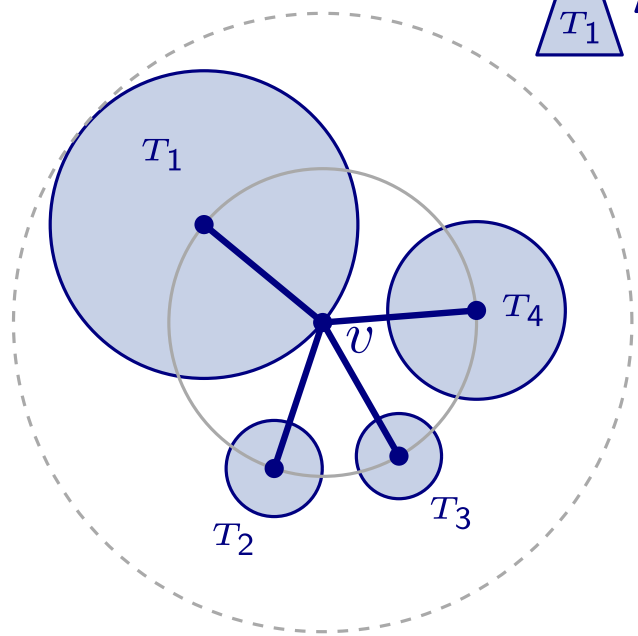



Subtrees may have different size, the tree is unordered. Drawn by Lin& Yen Algorithm.

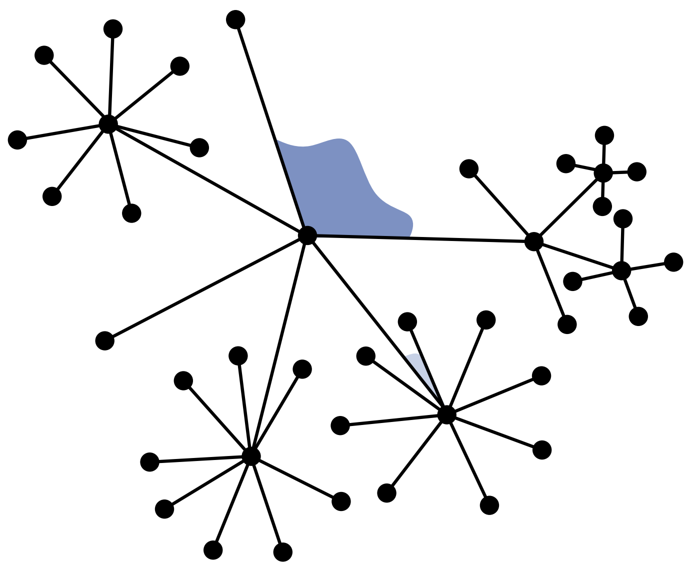
**Induction base:** 

**Induction step:**



# Baloon Layout

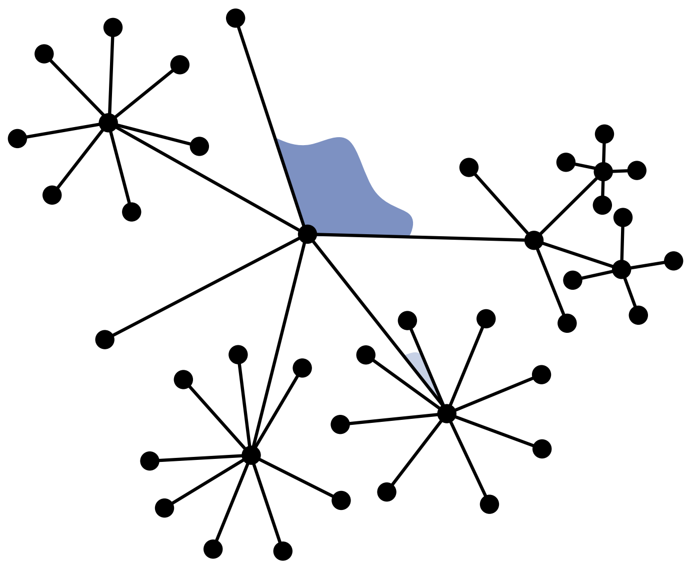


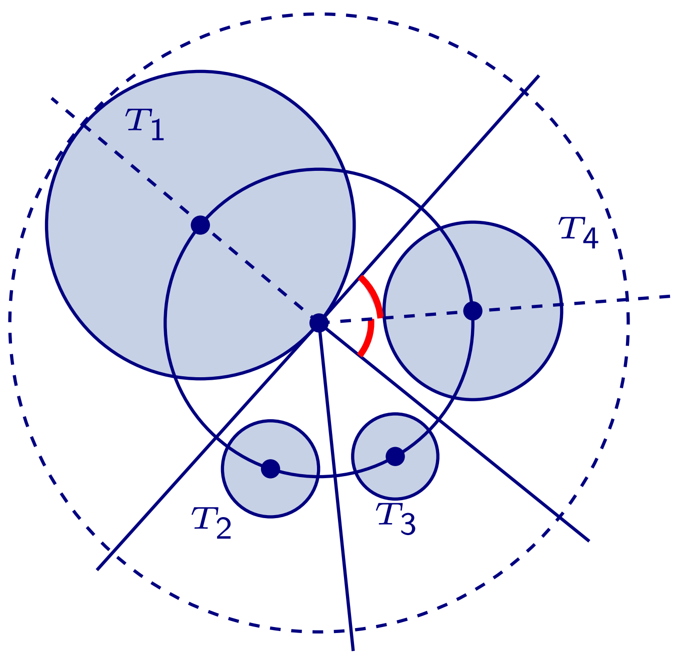
**Aesthetics:**

- Aspect ratio $= \frac{\text{largest angle}}{\text{smallest angle}}$
- Angular resolution $=$ $\min\{\text{angle between two adjacent edges}\}$

**Question:** Can we find a baloon drawing with max angular resolution and min aspect ratio in an unordered tree? (Algorithm by Lin & Yen)

# Baloon Layout





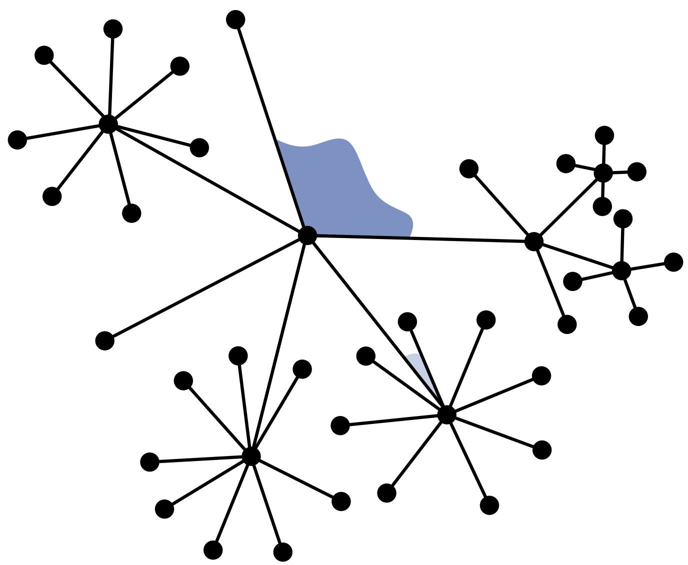

## Aesthetics:

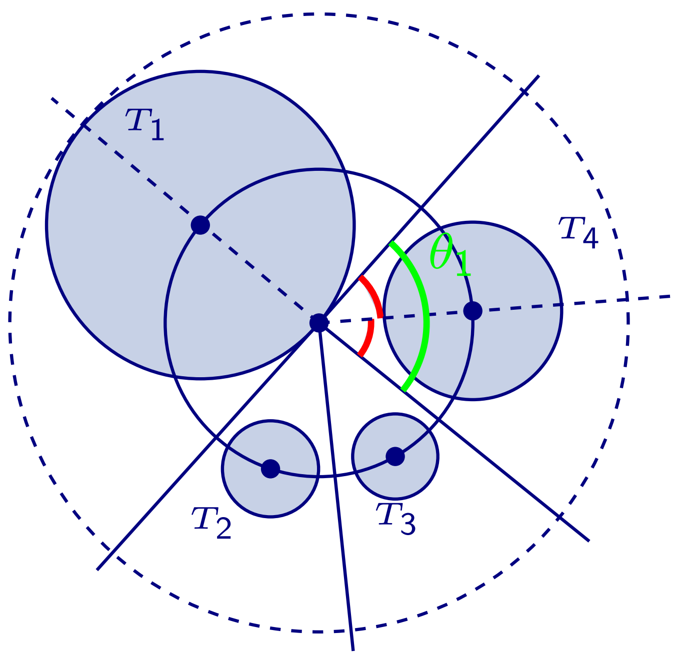
- $\text{Aspect ratio} = \dfrac{\text{largest angle}}{\text{smallest angle}}$
- $\text{Angular resolution} =$
  $\min\{\text{angle between two adjacent edges}\}$

**Question:** Can we find a baloon drawing with max angular resolution and min aspect ratio in an unordered tree? (Algorithm by Lin & Yen)

- We investigate drawing with **even angles** (drawing with uneven angles might have a better area)

# Baloon Layout





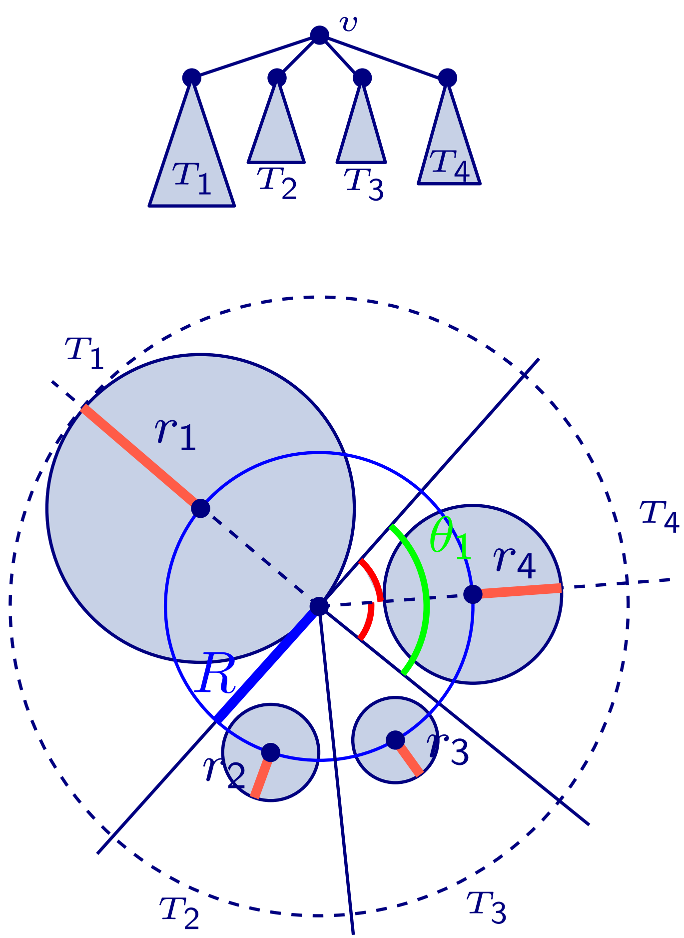

**Aesthetics:**

- Aspect ratio $= \frac{\text{largest angle}}{\text{smallest angle}}$
- Angular resolution $=$ $\min\{\text{angle between two adjacent edges}\}$

**Question:** Can we find a baloon drawing with max angular resolution and min aspect ratio in an unordered tree? (Algorithm by Lin & Yen)

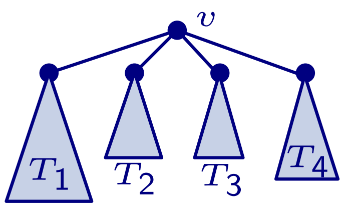
- We investigate drawing with **even angles** (drawing with uneven angles might have a better area)
- An arrangement of the subtree at a level can be describeed by a permutation $\sigma = \{1, 4, 2, 3\}$
- $\theta_i$ - angle of the wedge containing the subtree $T_i$

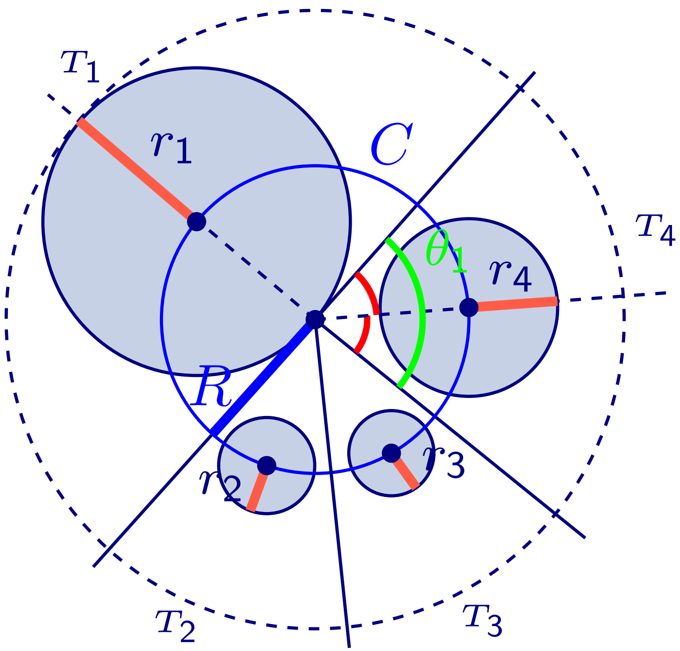
# Baloon Layout. Algorithm by Lin & Yen.




- Assume we are given the radii $r_i$ of the subtrees $T_i$. How to determine $\theta_i$ - angle of the wedge containing the circle $r_i$?

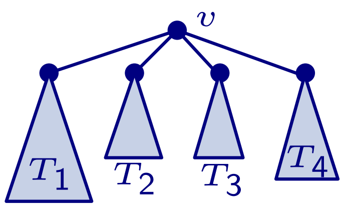
# Baloon Layout. Algorithm by Lin & Yen.

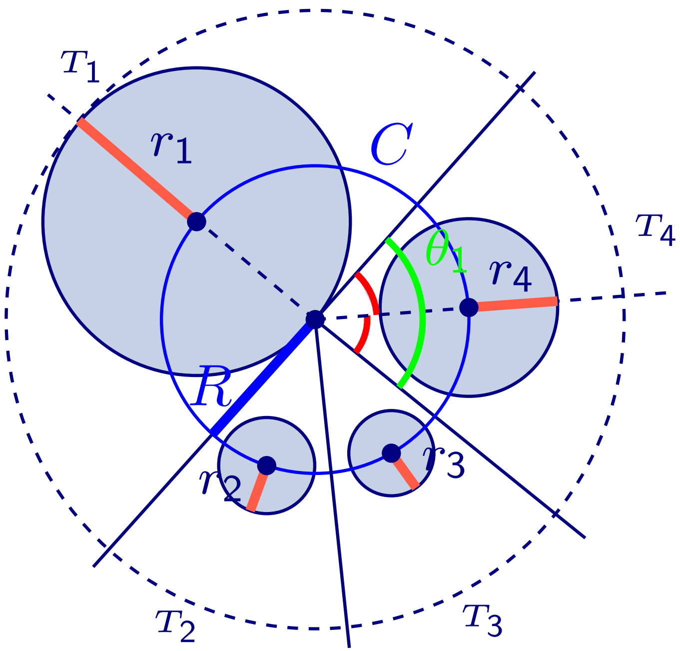



- Assume we are given the radii $r_i$ of the subtrees $T_i$. How to determine $\theta_i$ - angle of the wedge containing the circle $r_i$?
- Let $C \approx 2 \sum r_i$

# Baloon Layout. Algorithm by Lin & Yen.

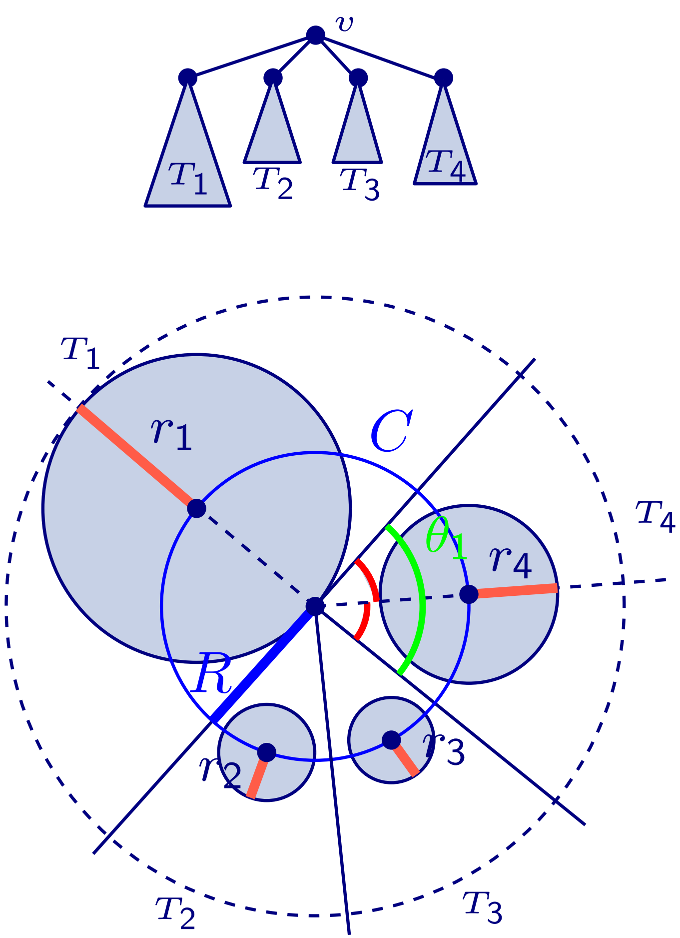



- Assume we are given the radii $r_i$ of the subtrees $T_i$. How to determine $\theta_i$ - angle of the wedge containing the circle $r_i$?

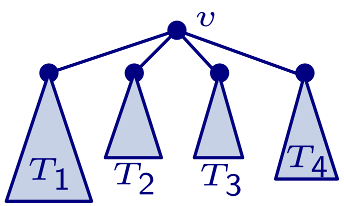
- Let $C \approx 2 \sum r_i$
- Let $r_{\text{max}} = \max\{r_i\}$

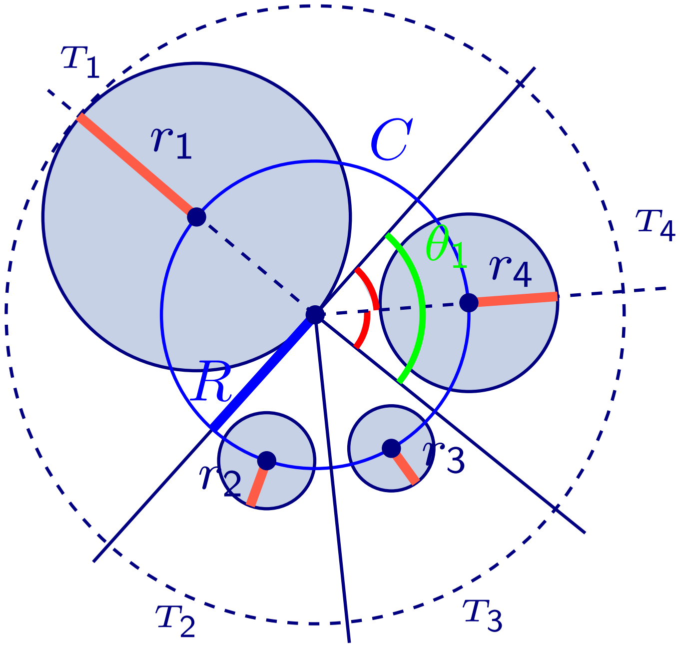
# Baloon Layout. Algorithm by Lin & Yen.




- Assume we are given the radii $r_i$ of the subtrees $T_i$. How to determine $\theta_i$ - angle of the wedge containing the circle $r_i$?
- Let $C \approx 2 \sum r_i$
- Let $r_{\max} = \max\{r_i\}$
- In order to avoid overlaps we compare $C$ and $2\pi r_{\max}$

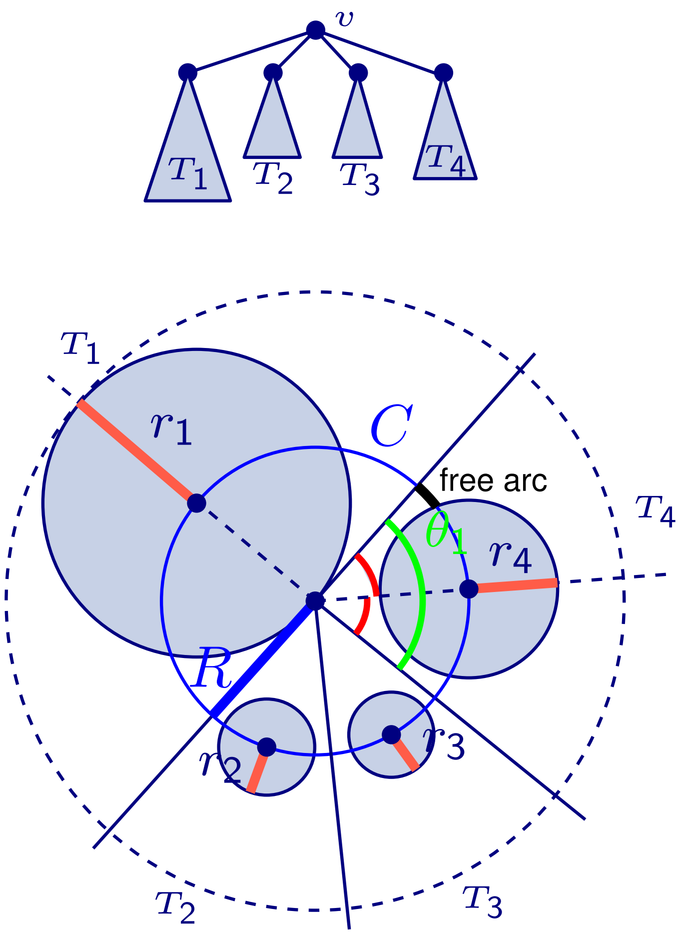
# Baloon Layout. Algorithm by Lin & Yen.



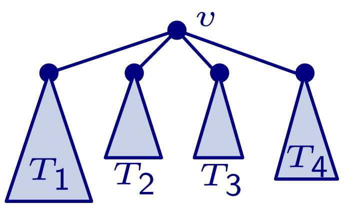

- Assume we are given the radii $r_i$ of the subtrees $T_i$. How to determine $\theta_i$ - angle of the wedge containing the circle $r_i$?
- Let $C \approx 2 \sum r_i$
- Let $r_{\max} = \max\{r_i\}$
- In order to avoid overlaps we compare $C$ and $2\pi r_{\max}$
- If $C \leq 2\pi r_{\max}$ we set $R = r_{\max}$, otherwise we set $R = \frac{C}{2\pi}$

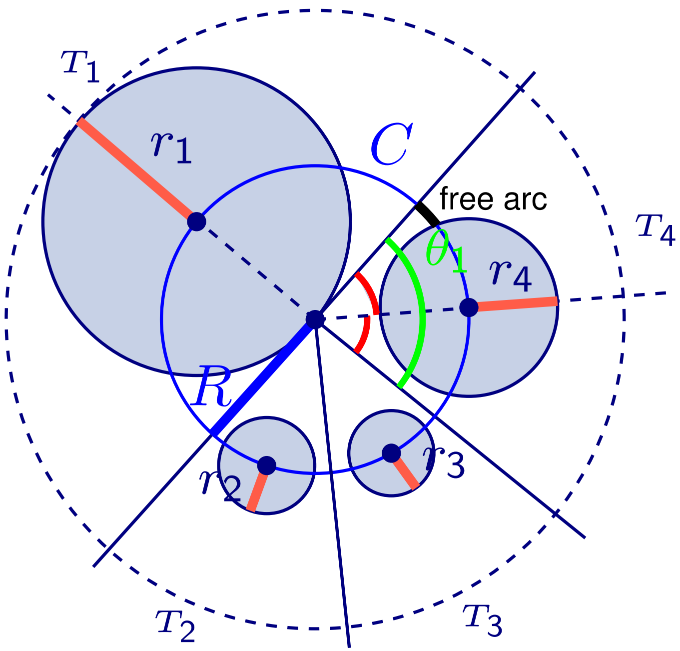
# Baloon Layout. Algorithm by Lin & Yen.




- Assume we are given the radii $r_i$ of the subtrees $T_i$. How to determine $\theta_i$ - angle of the wedge containing the circle $r_i$?
- Let $C \approx 2 \sum r_i$
- Let $r_{\text{max}} = \max\{r_i\}$
- In order to avoid overlaps we compare $C$ and $2\pi r_{\text{max}}$
- If $C \leq 2\pi r_{\text{max}}$ we set $R = r_{\text{max}}$, otherwise we set $R = \frac{C}{2\pi}$
- Set $\theta_i = 2(r_i + \text{free arc})/R$

# Baloon Layout. Algorithm by Lin & Yen.

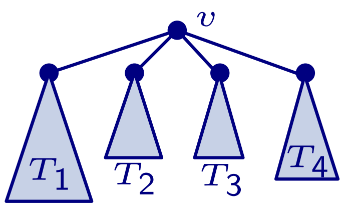


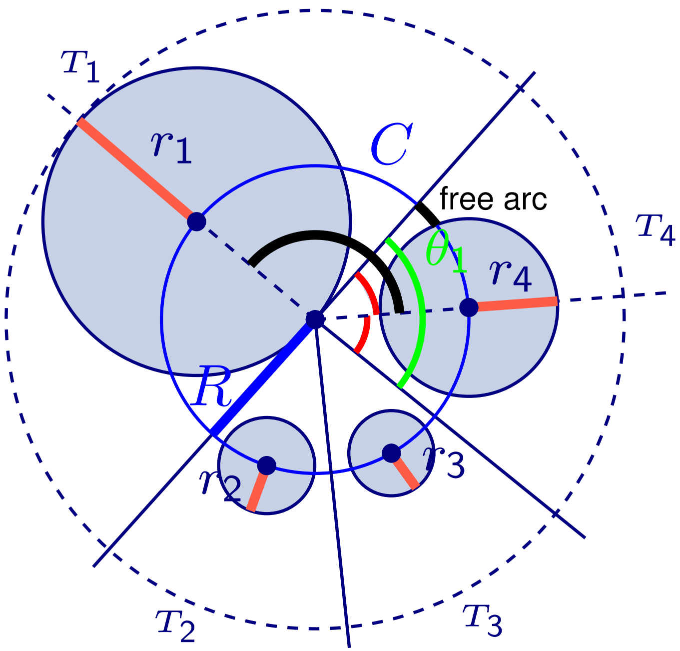




- Assume we are given the radii $r_i$ of the subtrees $T_i$. How to determine $\theta_i$ - angle of the wedge containing the circle $r_i$?

- Let $C \approx 2 \sum r_i$
- Let $r_{\max} = \max\{r_i\}$
- In order to avoid overlaps we compare $C$ and $2\pi r_{\max}$
- If $C \leq 2\pi r_{\max}$ we set $R = r_{\max}$, otherwise we set $R = \frac{C}{2\pi}$
- Set $\theta_i = 2(r_i + \text{free arc})/R$
- Let $\sigma$ be the permutation of the subtrees

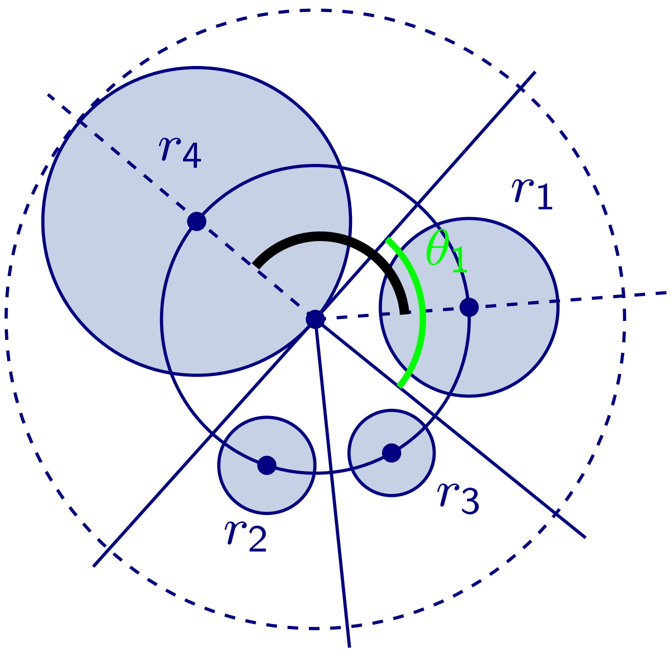
# Baloon Layout. Algorithm by Lin & Yen.



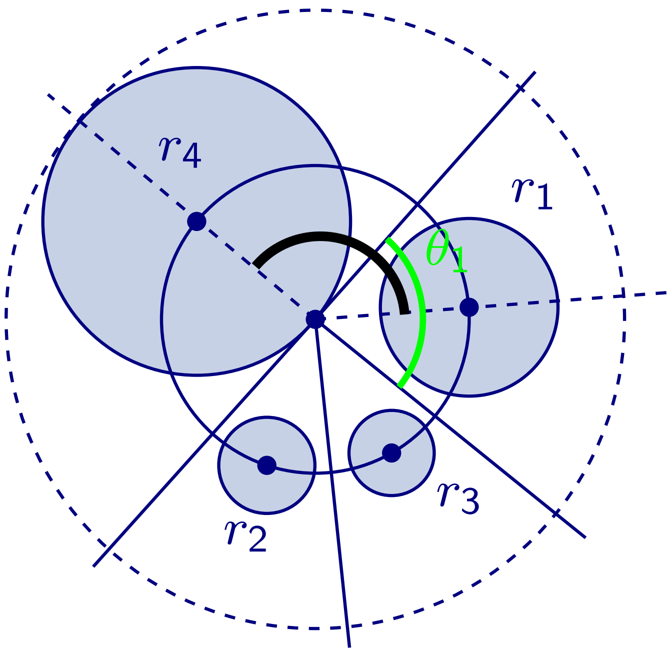

- Assume we are given the radii $r_i$ of the subtrees $T_i$. How to determine $\theta_i$ - angle of the wedge containing the circle $r_i$?
- Let $C \approx 2 \sum r_i$
- Let $r_{\max} = \max\{r_i\}$
- In order to avoid overlaps we compare $C$ and $2\pi r_{\max}$
- If $C \leq 2\pi r_{\max}$ we set $R = r_{\max}$, otherwise we set $R = \frac{C}{2\pi}$
- Set $\theta_i = 2(r_i + \text{free arc})/R$
- Let $\sigma$ be the permutation of the subtrees
- $\frac{\theta_{\sigma_i} + \theta_{\sigma_{i+1}}}{2}$ - angle between two consecutive edges

# Baloon Layout. Algorithm by Lin & Yen.



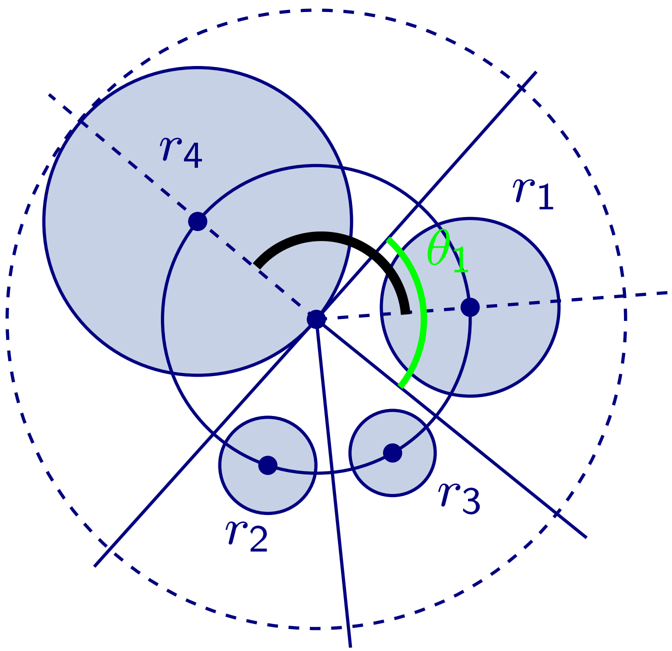

- $\frac{\theta_{\sigma_i} + \theta_{\sigma_{i+1}}}{2}$ - angle between two edges

# Baloon Layout. Algorithm by Lin & Yen.



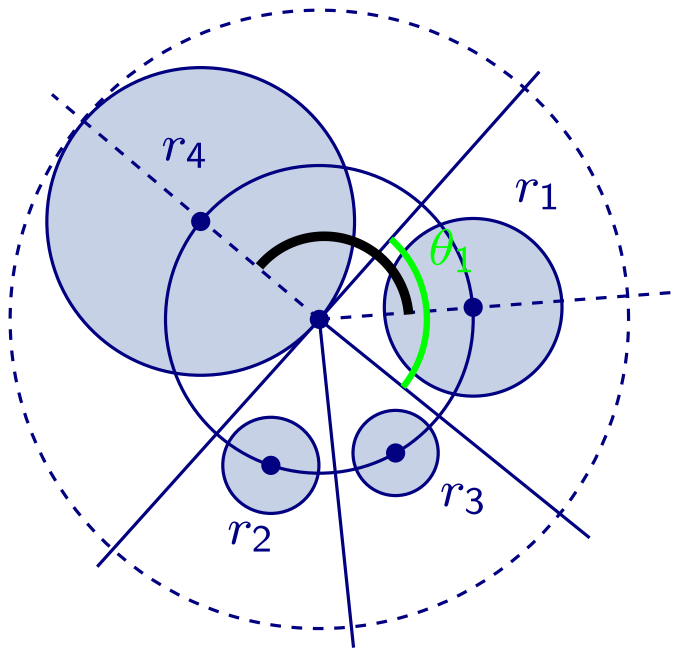

- $\frac{\theta_{\sigma_i}+\theta_{\sigma_{i+1}}}{2}$ - angle between two edges
- $AngResl_\sigma = \min_{1\leq i\leq n}\{\frac{\theta_{\sigma_i}+\theta_{\sigma_{i+1}}}{2}\}$

# Baloon Layout. Algorithm by Lin & Yen.



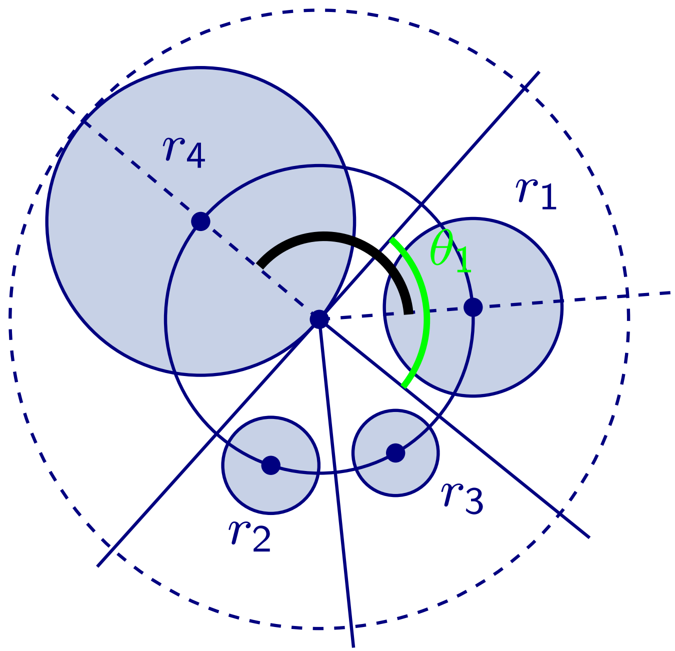

- $\frac{\theta_{\sigma_i}+\theta_{\sigma_{i+1}}}{2}$ - angle between two edges
- $AngResl_\sigma = \min_{1 \leq i \leq n}\{\frac{\theta_{\sigma_i}+\theta_{\sigma_{i+1}}}{2}\}$
- Permute the circles ($\sigma$) so that the $AngleResl_\sigma$ is maximized.

# Baloon Layout. Algorithm by Lin & Yen.



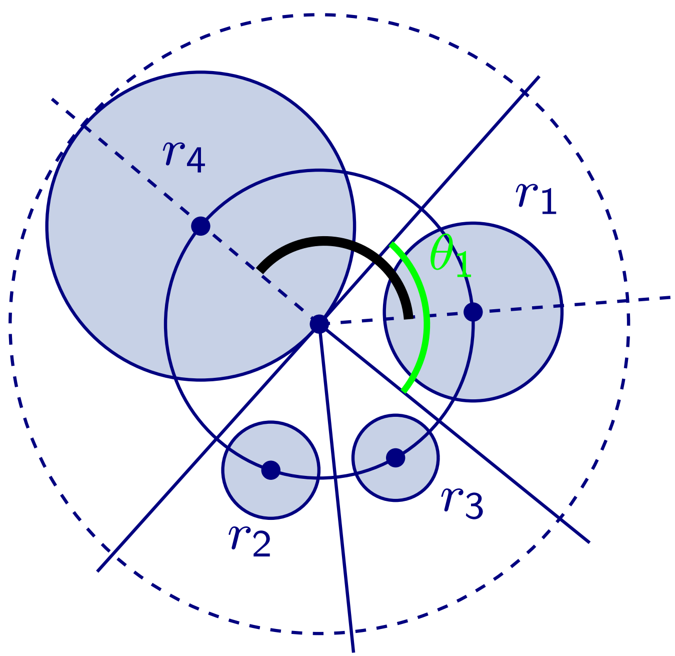

- $\frac{\theta_{\sigma_i} + \theta_{\sigma_{i+1}}}{2}$ - angle between two edges
- $AngResl_\sigma = \min_{1 \le i \le n}\{\frac{\theta_{\sigma_i} + \theta_{\sigma_{i+1}}}{2}\}$
- Permute the circles ($\sigma$) so that the $AngleResl_\sigma$ is maximized.
- Let $m_1, m_2, \ldots, m_k, mid, M_k, M_{k-1}, \ldots, M_2, M_1$ be the angles $\theta$ in the increasing ordering, i.e. $m_i$ ($M_i$) is $i$-th min (max), $mid$-unique medium, in case of odd number of circles and even $k$.

# Baloon Layout. Algorithm by Lin & Yen.



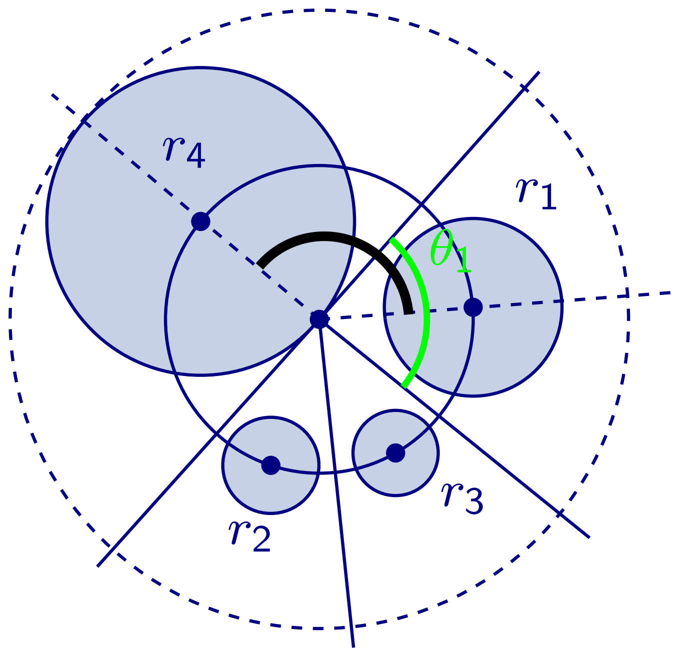

- $\frac{\theta_{\sigma_i}+\theta_{\sigma_{i+1}}}{2}$ - angle between two edges
- $AngResl_\sigma = \min_{1\leq i\leq n}\{\frac{\theta_{\sigma_i}+\theta_{\sigma_{i+1}}}{2}\}$
- Permute the circles ($\sigma$) so that the $AngleResl_\sigma$ is maximized.
- Let $m_1, m_2, \ldots, m_k, mid, M_k, M_{k-1}, \ldots, M_2, M_1$ be the angles $\theta$ in the increasing ordering, i.e. $m_i$ ($M_i$) is $i$-th min (max), $mid$-unique medium, in case of odd number of circles and even $k$.
- Let $\sigma = \{M_1, m_2, M_3, m_4, \ldots, M_{k-1}, m_k, mid, M_k, m_{k-1}, \ldots, M_4, m_3, M_2, m_1\}$. We show that $\sigma$ gives maximum angle resolution.

# Baloon Layout. Algorithm by Lin & Yen.



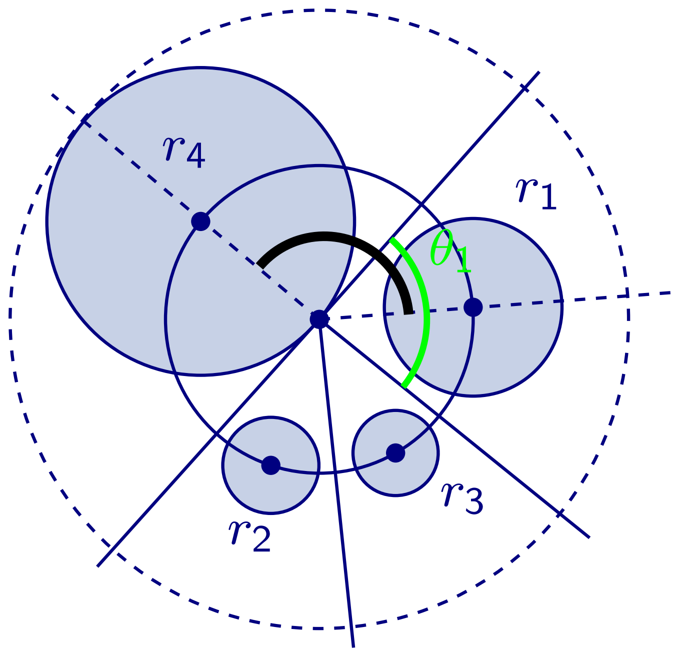

- $\frac{\theta_{\sigma_i}+\theta_{\sigma_{i+1}}}{2}$ - angle between two edges
- $AngResl_\sigma = \min_{1\leq i\leq n}\{\frac{\theta_{\sigma_i}+\theta_{\sigma_{i+1}}}{2}\}$
- Permute the circles ($\sigma$) so that the $AngleResl_\sigma$ is maximized.
- Let $m_1, m_2, \ldots, m_k, mid, M_k, M_{k-1}, \ldots, M_2, M_1$ be the angles $\theta$ in the increasing ordering, i.e. $m_i$ ($M_i$) is $i$-th min (max), $mid$-unique medium, in case of odd number of circles and even $k$.
- Let $\sigma = \{M_1, m_2, M_3, m_4, \ldots, M_{k-1}, m_k, mid, M_k, m_{k-1}, \ldots, M_4, m_3, M_2, m_1\}$. We show that $\sigma$ gives maximum angle resolution.
- Let $\alpha_{ij} = \frac{M_i+m_j}{2}$. Angles $\frac{mid+m_k}{2}, \alpha_{(i-1)i}, \frac{M_k+mid}{2}, \alpha_{i(i-1)}$ are in $\sigma$.

# Baloon Layout. Algorithm by Lin & Yen.




- $\frac{\theta_{\sigma_i}+\theta_{\sigma_{i+1}}}{2}$ - angle between two edges
- $AngResl_\sigma = \min_{1\le i\le n}\{\frac{\theta_{\sigma_i}+\theta_{\sigma_{i+1}}}{2}\}$
- Permute the circles ($\sigma$) so that the $AngleResl_\sigma$ is maximized.
- Let $m_1, m_2, \ldots, m_k, mid, M_k, M_{k-1}, \ldots, M_2, M_1$ be the angles $\theta$ in the increasing ordering, i.e. $m_i$ ($M_i$) is $i$-th min (max), $mid$-unique medium, in case of odd number of circles and even $k$.
- Let $\sigma = \{M_1, m_2, M_3, m_4, \ldots, M_{k-1}, m_k, mid, M_k, m_{k-1}, \ldots, M_4, m_3, M_2, m_1\}$. We show that $\sigma$ gives maximum angle resolution.
- Let $\alpha_{ij} = \frac{M_i+m_j}{2}$. Angles $\frac{mid+m_k}{2}, \alpha_{(i-1)i}, \frac{M_k+mid}{2}, \alpha_{i(i-1)}$ are in $\sigma$.
- Relations among $\alpha_{ij}$:

# Baloon Layout. Algorithm by Lin & Yen.




- $\frac{\theta_{\sigma_i}+\theta_{\sigma_{i+1}}}{2}$ - angle between two edges
- $AngResl_\sigma = \min_{1\leq i\leq n}\{\frac{\theta_{\sigma_i}+\theta_{\sigma_{i+1}}}{2}\}$
- Permute the circles ($\sigma$) so that the $AngleResl_\sigma$ is maximized.
- Let $m_1, m_2, \ldots, m_k, mid, M_k, M_{k-1}, \ldots, M_2, M_1$ be the angles $\theta$ in the increasing ordering, i.e. $m_i$ ($M_i$) is $i$-th min (max), $mid$-unique medium, in case of odd number of circles and even $k$.
- Let $\sigma = \{M_1, m_2, M_3, m_4, \ldots, M_{k-1}, m_k, mid, M_k, m_{k-1}, \ldots, M_4, m_3, M_2, m_1\}$. We show that $\sigma$ gives maximum angle resolution.
- Let $\alpha_{ij} = \frac{M_i+m_j}{2}$. Angles $\frac{mid+m_k}{2}, \alpha_{(i-1)i}, \frac{M_k+mid}{2}, \alpha_{i(i-1)}$ are in $\sigma$.
- Relations among $\alpha_{ij}$:

$$\alpha_{12} > \alpha_{32} < \alpha_{34} > \cdots > \alpha_{j(j-1)} < \cdots > \alpha_{k(k-1)} < \frac{M_k+mid}{2} > \frac{mid+m_k}{2} < \alpha_{(k-1)k} > \cdots > \alpha_{43} < \alpha_{23} > \alpha_{21} < \alpha_{12}.$$

# Baloon Layout. Algorithm by Lin & Yen.

- Recall $\alpha_{ij} = \frac{M_i + m_j}{2}$
- Relations among $\alpha_{ij}$:

$\alpha_{12} > \alpha_{32} < \alpha_{34} > \cdots > \alpha_{j(j-1)} < \cdots > \alpha_{k(k-1)} < \frac{M_k + mid}{2} > \frac{mid + m_k}{2} < \alpha_{(k-1)k} > \cdots > \alpha_{43} < \alpha_{23} > \alpha_{21} < \alpha_{12}$.

- Smallest angle in $\sigma$ is either: $\frac{mid + m_k}{2}$ or $\alpha_{j(j-1)}$, while the size of the biggest angle is either $\frac{M_k + mid}{2}$ or $\alpha_{(l-1)l}$, $j, l \in \{2, \ldots, k\}$.

# Baloon Layout. Algorithm by Lin & Yen.

- Recall $\alpha_{ij} = \frac{M_i + m_j}{2}$
- Relations among $\alpha_{ij}$:

$\alpha_{12} > \alpha_{32} < \alpha_{34} > \cdots > \alpha_{j(j-1)} < \cdots > \alpha_{k(k-1)} < \frac{M_k + mid}{2} > \frac{mid + m_k}{2} < \alpha_{(k-1)k} > \cdots > \alpha_{43} < \alpha_{23} > \alpha_{21} < \alpha_{12}$.

- Smallest angle in $\sigma$ is either: $\frac{mid + m_k}{2}$ or $\alpha_{j(j-1)}$, while the size of the biggest angle is either $\frac{M_k + mid}{2}$ or $\alpha_{(l-1)l}$, $j, l \in \{2, \ldots, k\}$.
- Assume the min angle of $\sigma$ is $\alpha_{i(i-1)} = \frac{M_i + m_{i-1}}{2}$, and assume $\sigma$ is not optimal.

# Baloon Layout. Algorithm by Lin & Yen.

- Recall $\alpha_{ij} = \frac{M_i + m_j}{2}$
- Relations among $\alpha_{ij}$:

$\alpha_{12} > \alpha_{32} < \alpha_{34} > \cdots > \alpha_{j(j-1)} < \cdots > \alpha_{k(k-1)} < \frac{M_k+mid}{2} > \frac{mid+m_k}{2} < \alpha_{(k-1)k} > \cdots > \alpha_{43} < \alpha_{23} > \alpha_{21} < \alpha_{12}$.

- Smallest angle in $\sigma$ is either: $\frac{mid+m_k}{2}$ or $\alpha_{j(j-1)}$, while the size of the biggest angle is either $\frac{M_k+mid}{2}$ or $\alpha_{(l-1)l}$, $j, l \in \{2, \dots, k\}$.
- Assume the min angle of $\sigma$ is $\alpha_{i(i-1)} = \frac{M_i+m_{i-1}}{2}$, and assume $\sigma$ is not optimal.
- Let $\delta$ be a permutation with maximum angle resolution

# Baloon Layout. Algorithm by Lin & Yen.

- Recall $\alpha_{ij} = \frac{M_i + m_j}{2}$
- Relations among $\alpha_{ij}$:

$\alpha_{12} > \alpha_{32} < \alpha_{34} > \cdots > \alpha_{j(j-1)} < \cdots > \alpha_{k(k-1)} < \frac{M_k + mid}{2} > \frac{mid + m_k}{2} < \alpha_{(k-1)k} > \cdots > \alpha_{43} < \alpha_{23} > \alpha_{21} < \alpha_{12}$.

- Smallest angle in $\sigma$ is either: $\frac{mid + m_k}{2}$ or $\alpha_{j(j-1)}$, while the size of the biggest angle is either $\frac{M_k + mid}{2}$ or $\alpha_{(l-1)l}$, $j, l \in \{2, \ldots, k\}$.
- Assume the min angle of $\sigma$ is $\alpha_{i(i-1)} = \frac{M_i + m_{i-1}}{2}$, and assume $\sigma$ is not optimal.
- Let $\delta$ be a permutation with maximum angle resolution
- If $M_i$ and $m_{i-1}$ neighbor in $\delta$ then $optAngResl = \alpha_{i,i-1}$ (???)

# Baloon Layout. Algorithm by Lin & Yen.

- Recall $\alpha_{ij} = \frac{M_i + m_j}{2}$
- Relations among $\alpha_{ij}$:

$\alpha_{12} > \alpha_{32} < \alpha_{34} > \cdots > \alpha_{j(j-1)} < \cdots > \alpha_{k(k-1)} < \frac{M_k+mid}{2} > \frac{mid+m_k}{2} < \alpha_{(k-1)k} > \cdots > \alpha_{43} < \alpha_{23} > \alpha_{21} < \alpha_{12}$.

- Smallest angle in $\sigma$ is either: $\frac{mid+m_k}{2}$ or $\alpha_{j(j-1)}$, while the size of the biggest angle is either $\frac{M_k+mid}{2}$ or $\alpha_{(l-1)l}$, $j, l \in \{2, \dots, k\}$.
- Assume the min angle of $\sigma$ is $\alpha_{i(i-1)} = \frac{M_i + m_{i-1}}{2}$, and assume $\sigma$ is not optimal.
- Let $\delta$ be a permutation with maximum angle resolution
- If $M_i$ and $m_{i-1}$ neighbor in $\delta$ then $optAngResl = \alpha_{i,i-1}$ (???)
- If they do not, let $x, y$ be the neighbors of $m_{i-1}$ in $\delta$, then:

$$\underbrace{m_1 < \cdots < m_{i-1}}_{i-1} < \dots < M_i < \underbrace{\dots < x < \dots < y < M_1}_{i-1}$$

# Baloon Layout. Algorithm by Lin & Yen.

- Thus, $M_i$ and $m_{i-1}$ neighbor in $\delta$ and therefore $AngRes_\sigma = AngRes_\delta$, i.e. $\sigma$ maximizes the size of the smallest angle.

# Baloon Layout. Algorithm by Lin & Yen.

- Thus, $M_i$ and $m_{i-1}$ neighbor in $\delta$ and therefore $AngRes_\sigma = AngRes_\delta$, i.e. $\sigma$ maximizes the size of the smallest angle.
- Similarly, we can show that $\sigma$ minimizes the lagest angle

# Baloon Layout. Algorithm by Lin & Yen.

- Thus, $M_i$ and $m_{i-1}$ neighbor in $\delta$ and therefore $AngRes_\sigma = AngRes_\delta$, i.e. $\sigma$ maximizes the size of the smallest angle.

- Similarly, we can show that $\sigma$ minimizes the lagest angle

- Recall that: $AspRatio_\sigma = \frac{\max_{1 \leq i \leq n}\{\frac{\theta_{\sigma_i} + \theta_{\sigma_{i+1}}}{2}\}}{\min_{1 \leq i \leq n}\{\frac{\theta_{\sigma_i} + \theta_{\sigma_{i+1}}}{2}\}}$

# Baloon Layout. Algorithm by Lin & Yen.

- Thus, $M_i$ and $m_{i-1}$ neighbor in $\delta$ and therefore $AngRes_\sigma = AngRes_\delta$, i.e. $\sigma$ maximizes the size of the smallest angle.

- Similarly, we can show that $\sigma$ minimizes the lagest angle

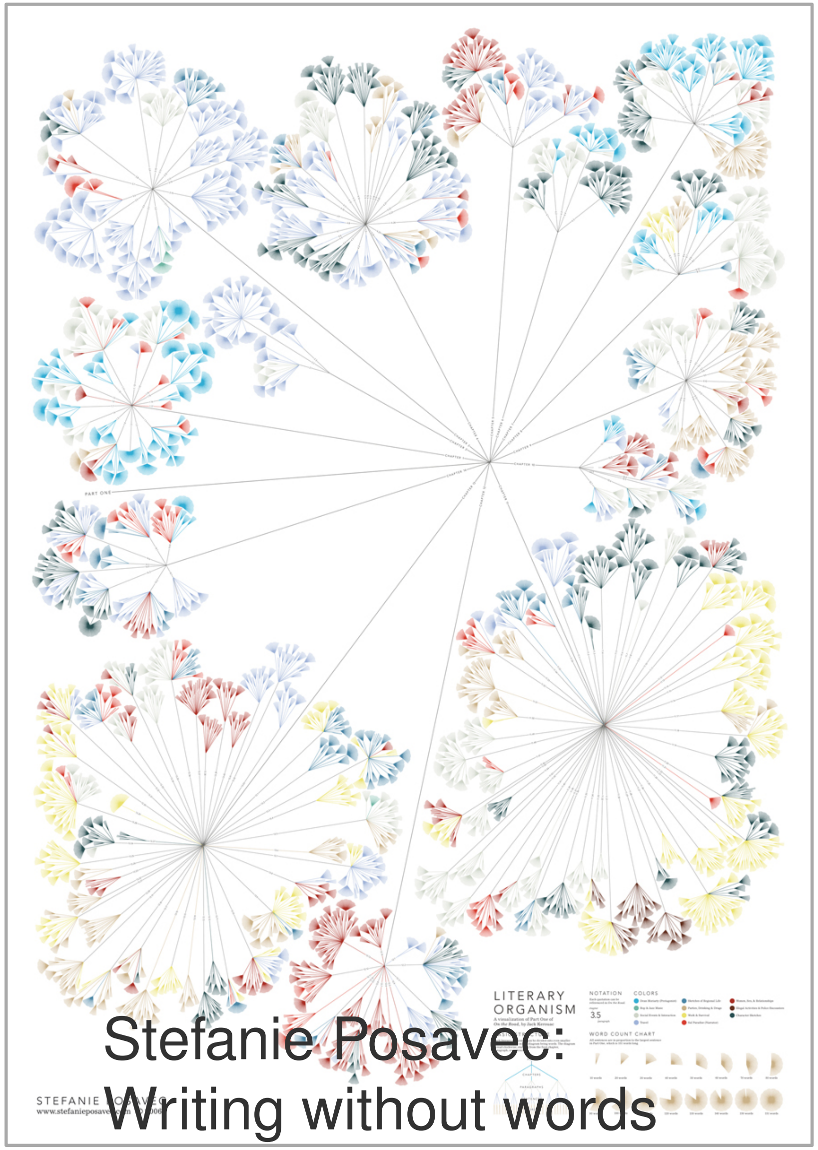
- Recall that: $AspRatio_\sigma = \frac{\max_{1 \leq i \leq n}\{\frac{\theta_{\sigma_i} + \theta_{\sigma_{i+1}}}{2}\}}{\min_{1 \leq i \leq n}\{\frac{\theta_{\sigma_i} + \theta_{\sigma_{i+1}}}{2}\}}$

- The radii and therefore the angles are independent on each level

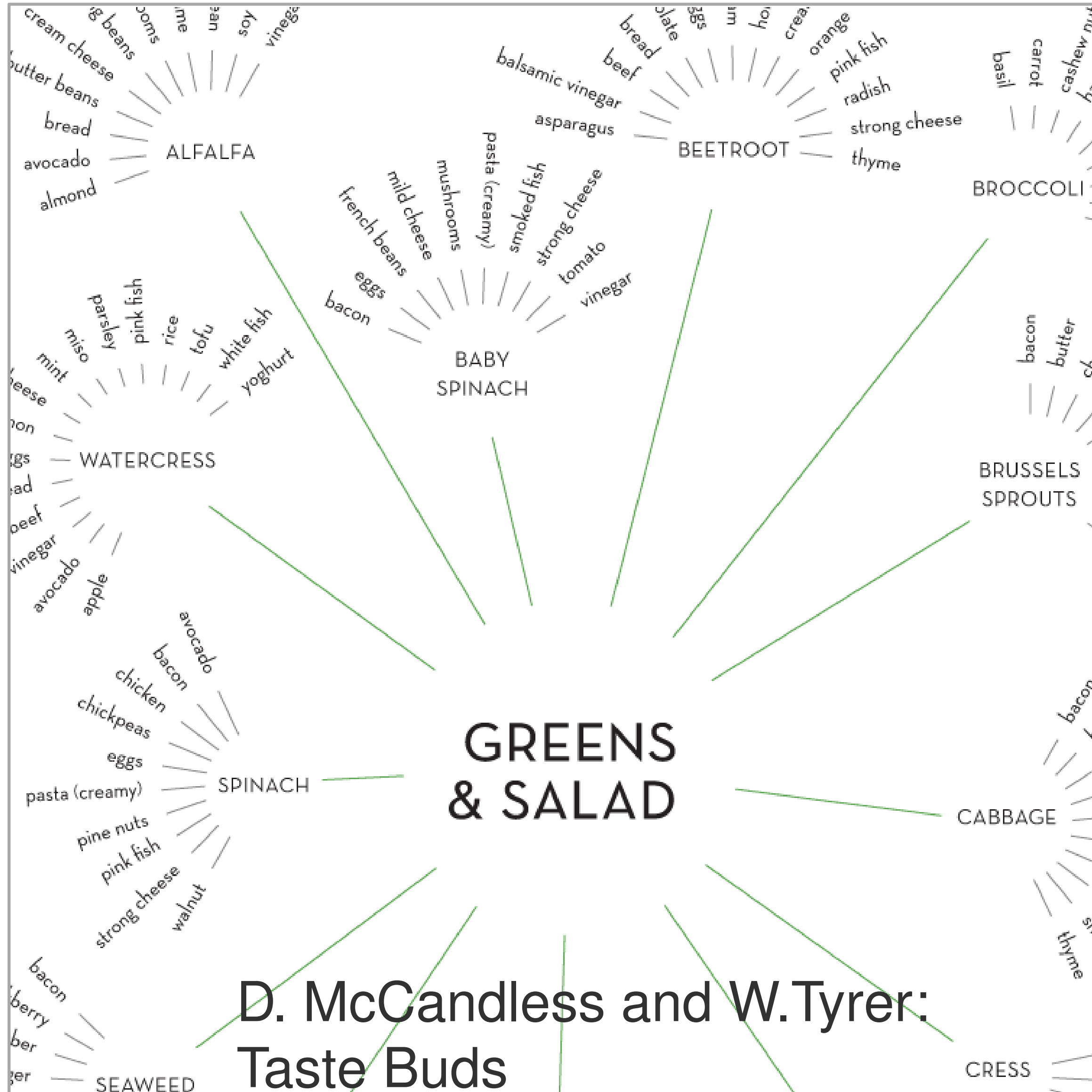
# Baloon Layout. Algorithm by Lin & Yen.

- Thus, $M_i$ and $m_{i-1}$ neighbor in $\delta$ and therefore $AngRes_\sigma = AngRes_\delta$, i.e. $\sigma$ maximizes the size of the smallest angle.
- Similarly, we can show that $\sigma$ minimizes the lagest angle
- Recall that: $AspRatio_\sigma = \frac{\max_{1\leq i\leq n}\{\frac{\theta_{\sigma_i}+\theta_{\sigma_{i+1}}}{2}\}}{\min_{1\leq i\leq n}\{\frac{\theta_{\sigma_i}+\theta_{\sigma_{i+1}}}{2}\}}$
- The radii and therefore the angles are independent on each level
- Therefore, if we apply $\sigma$ at each level, we obtain an optimal aspect ratio. $\square$

# Applications of (almost) Ballon Layout




Stefanie Posavec:
Writing without words

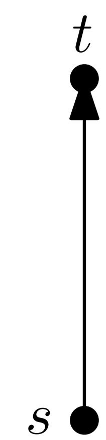



D. McCandless and W.Tyrer:
Taste Buds

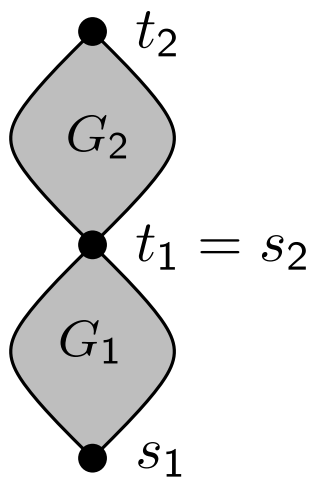
# Series-parallel Graphs

Graph $G$ is **series-parallel**, if

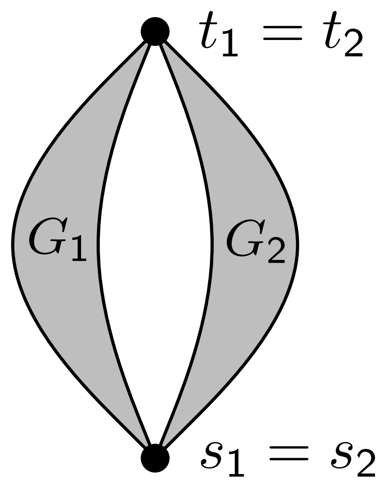
- It contains a single edge $(s, t)$ ($s$-source, $t$-sink)
- It consists of two series-parallel graphs $G_1, G_2$ with sources $s_1, s_2$ and sinks $t_1, t_2$ which are combined using one of the following rules:




**Series composition:**
Identify $t_1$ and $s_2$,
$s_1$ is the source of $G$, $t_2$ is the sink of $G$




**Parallel composition:**
Identify $s_1, s_2$ and set it to be source of $G$
Identify $t_1, t_2$ and set it to be sink of $G$



# Series-parallel Graphs. Decomposition Tree.



**Lemma**

Series-parallel graphs are acyclic and planar.

In order to proof this statement we can use a **decomposition tree** of $G$, which is a binary tree $T$ with nodes of three types: S,P and Q-type.

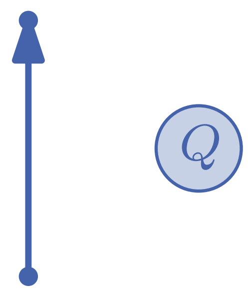
# Series-parallel Graphs. Decomposition Tree.

**Lemma**

Series-parallel graphs are acyclic and planar.

In order to proof this statement we can use a **decomposition tree** of $G$, which is a binary tree $T$ with nodes of three types: S,P and Q-type.

- If $G$ is a single edge, then the corresponding node is Q-node



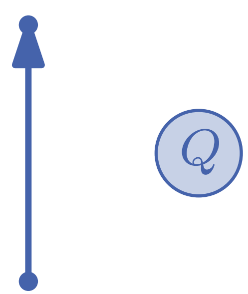
# Series-parallel Graphs. Decomposition Tree.

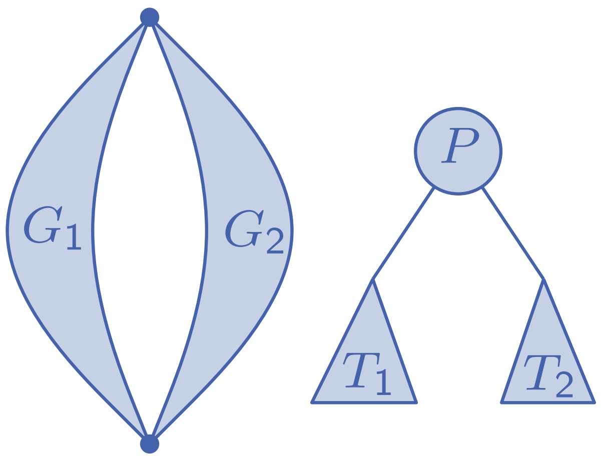
**Lemma**

Series-parallel graphs are acyclic and planar.

In order to proof this statement we can use a **decomposition tree** of $G$, which is a binary tree $T$ with nodes of three types: S,P and Q-type.

- If $G$ is a single edge, then the corresponding node is Q-node
- If $G$ is a parallel composition of $G_1$ (with tree $T_1$) and $G_2$ (with tree $T_2$), then the root of $T$ is P-node and $T_1$ is its left subtree, $T_2$ is its right subtree



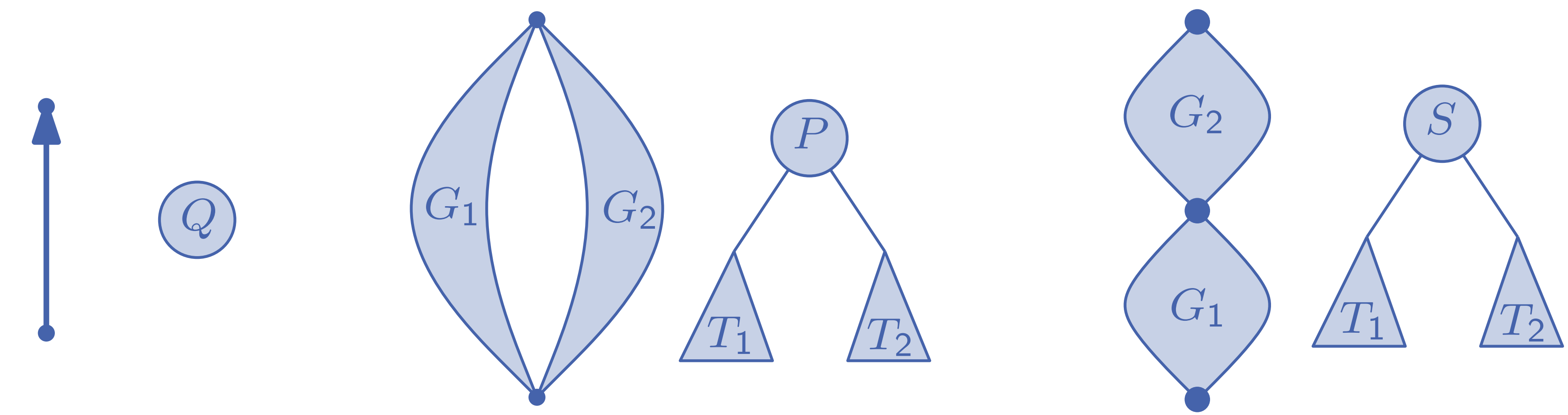
# Series-parallel Graphs. Decomposition Tree.

## Lemma

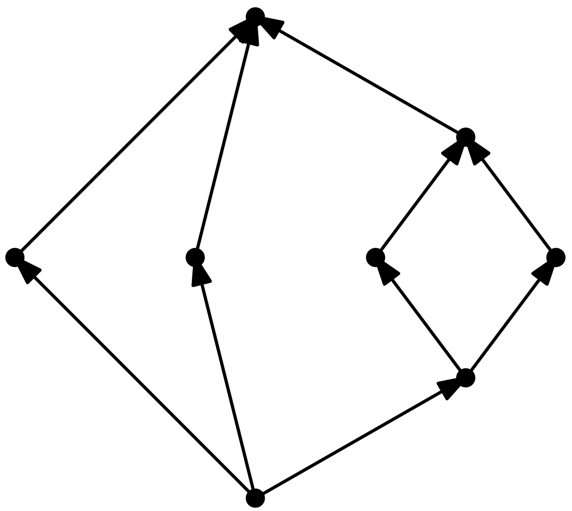
Series-parallel graphs are acyclic and planar.

In order to proof this statement we can use a **decomposition tree** of $G$, which is a binary tree $T$ with nodes of three types: S,P and Q-type.

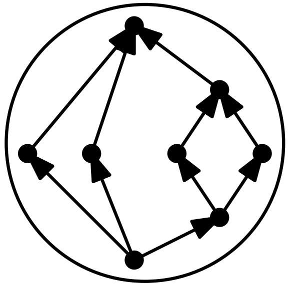
- If $G$ is a single edge, then the corresponding node is Q-node
- If $G$ is a parallel composition of $G_1$ (with tree $T_1$) and $G_2$ (with tree $T_2$), then the root of $T$ is P-node and $T_1$ is its left subtree, $T_2$ is its right subtree
- If $G$ is a series composition of $G_1$ (with tree $T_1$) and $G_2$ (with tree $T_2$), then the root of $T$ is S-node and $T_1$ is its left subtree, $T_2$ is its right subtree



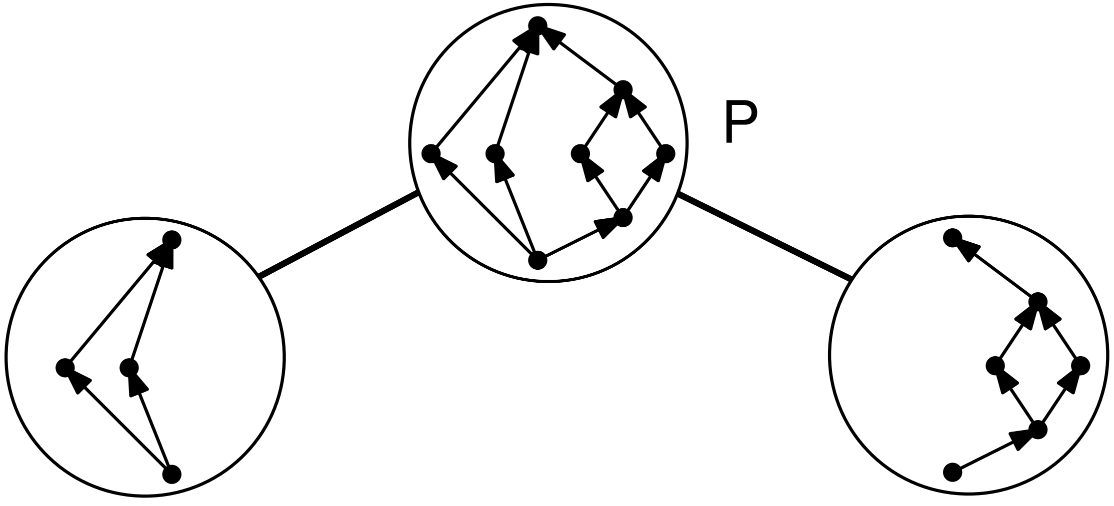
# Series-parallel Graphs. Decomposition Example.



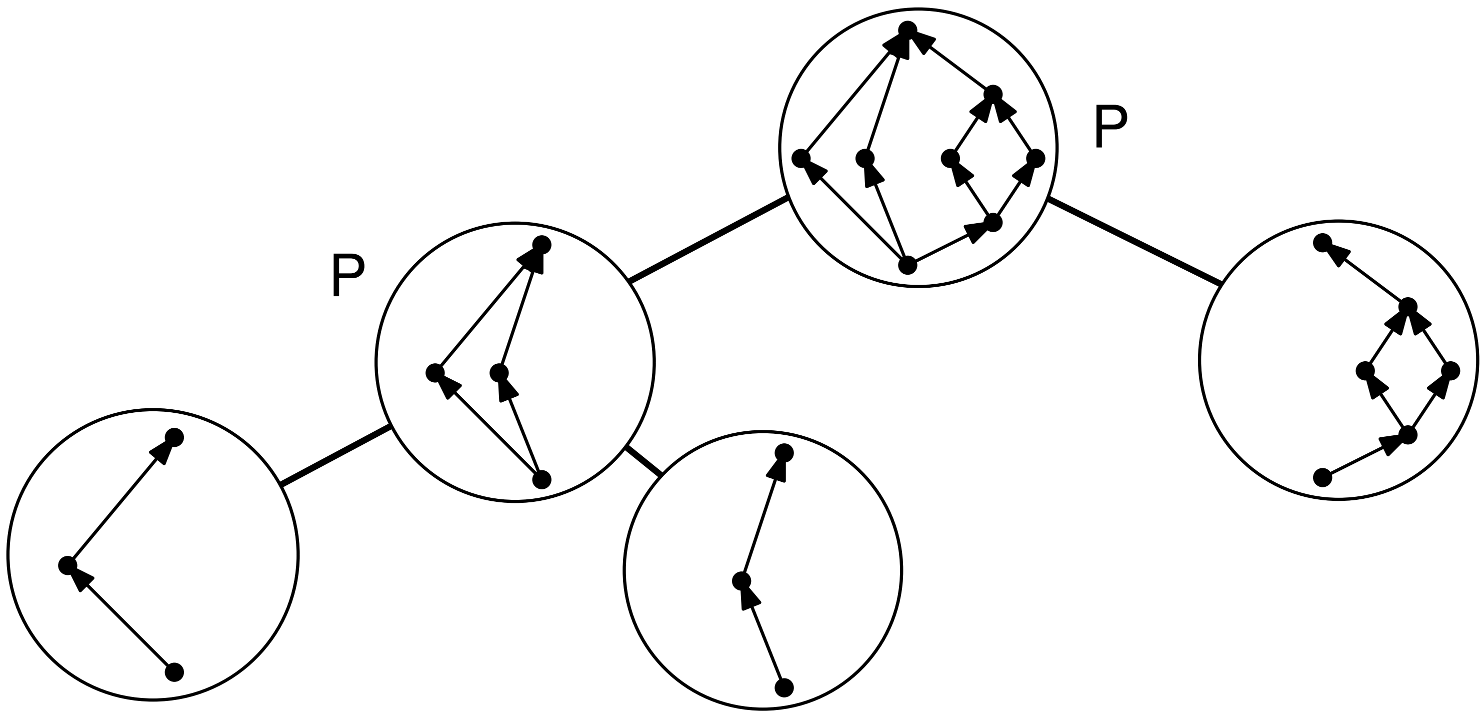
# Series-parallel Graphs. Decomposition Example.



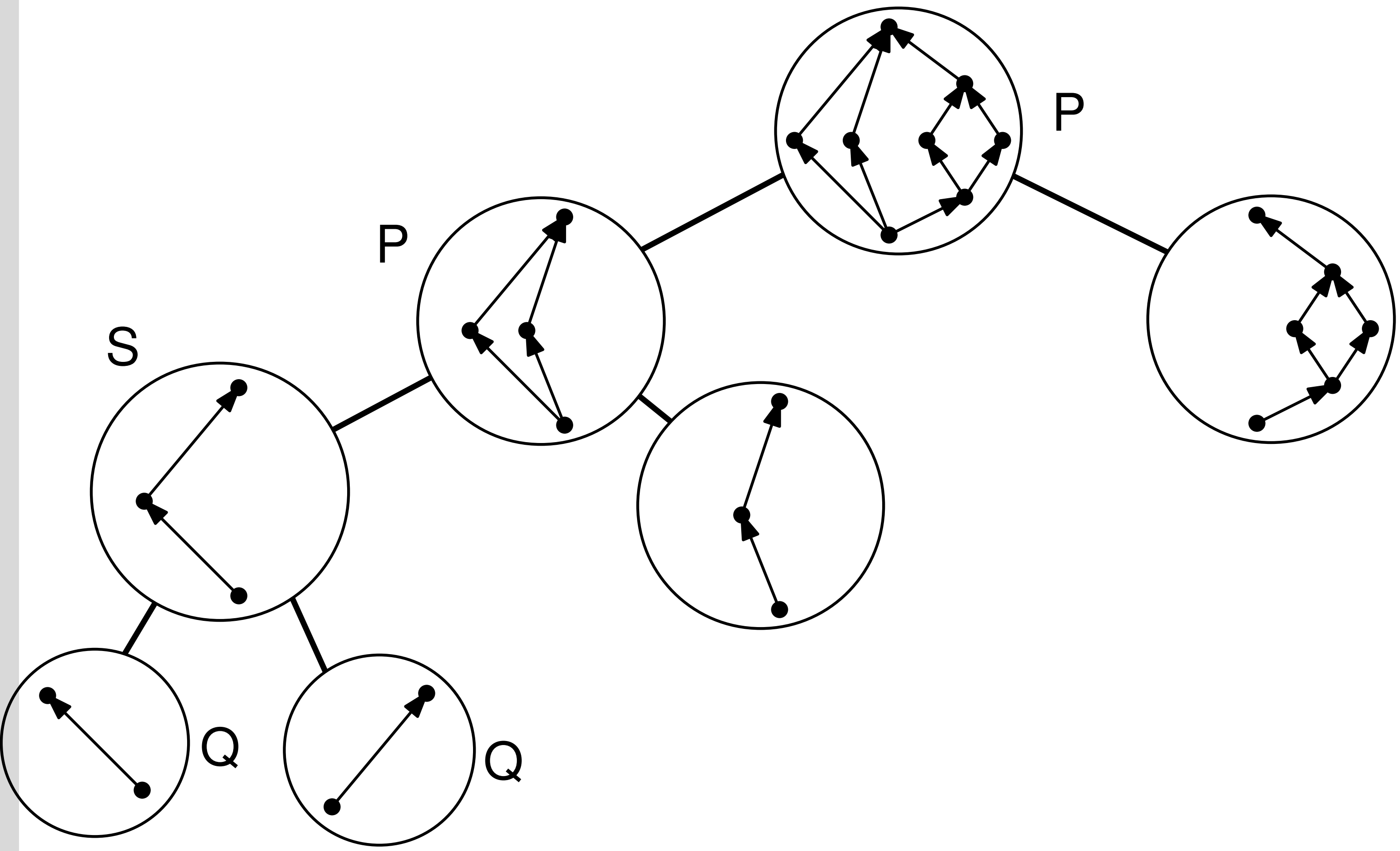
# Series-parallel Graphs. Decomposition Example.



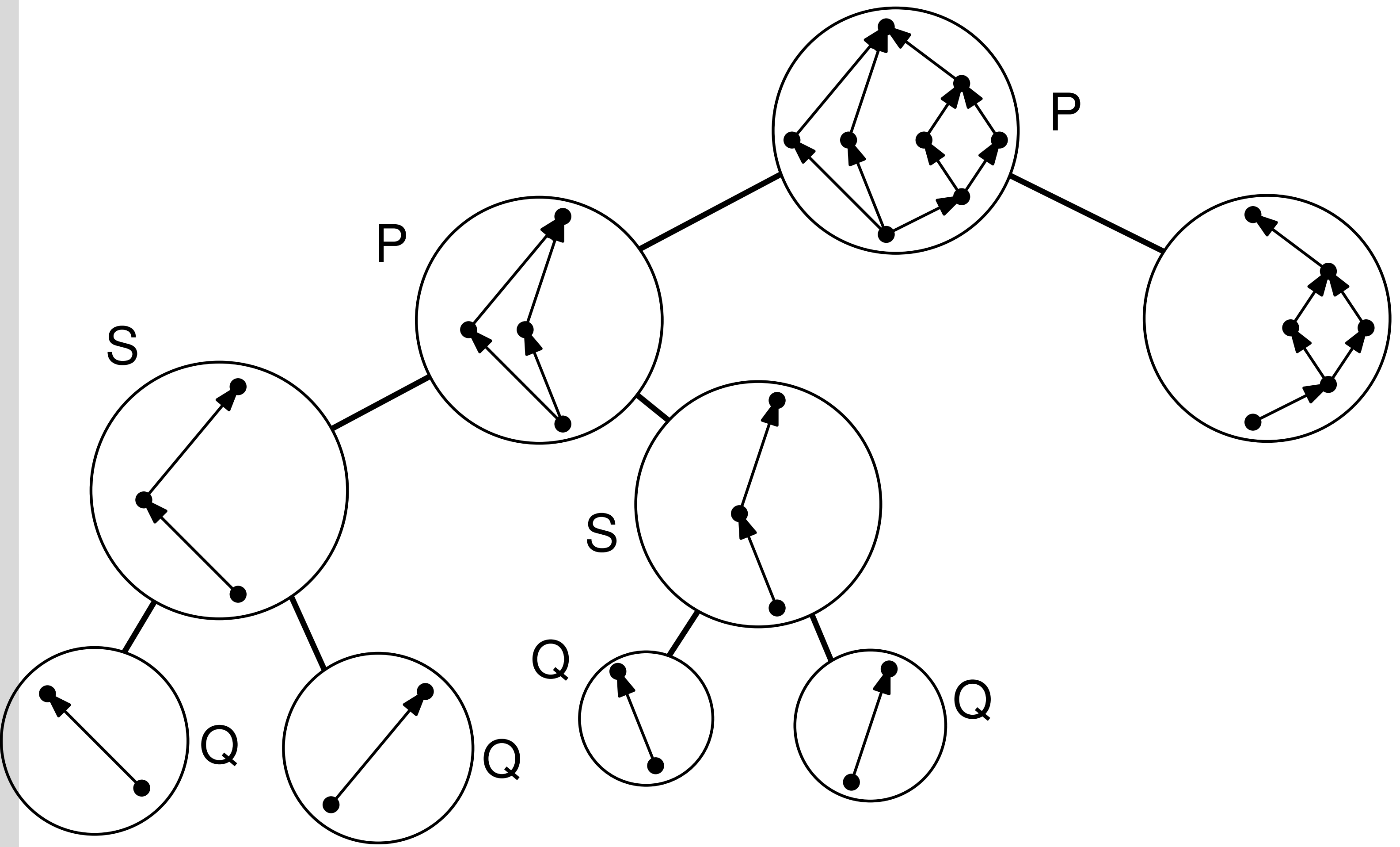
# Series-parallel Graphs. Decomposition Example.



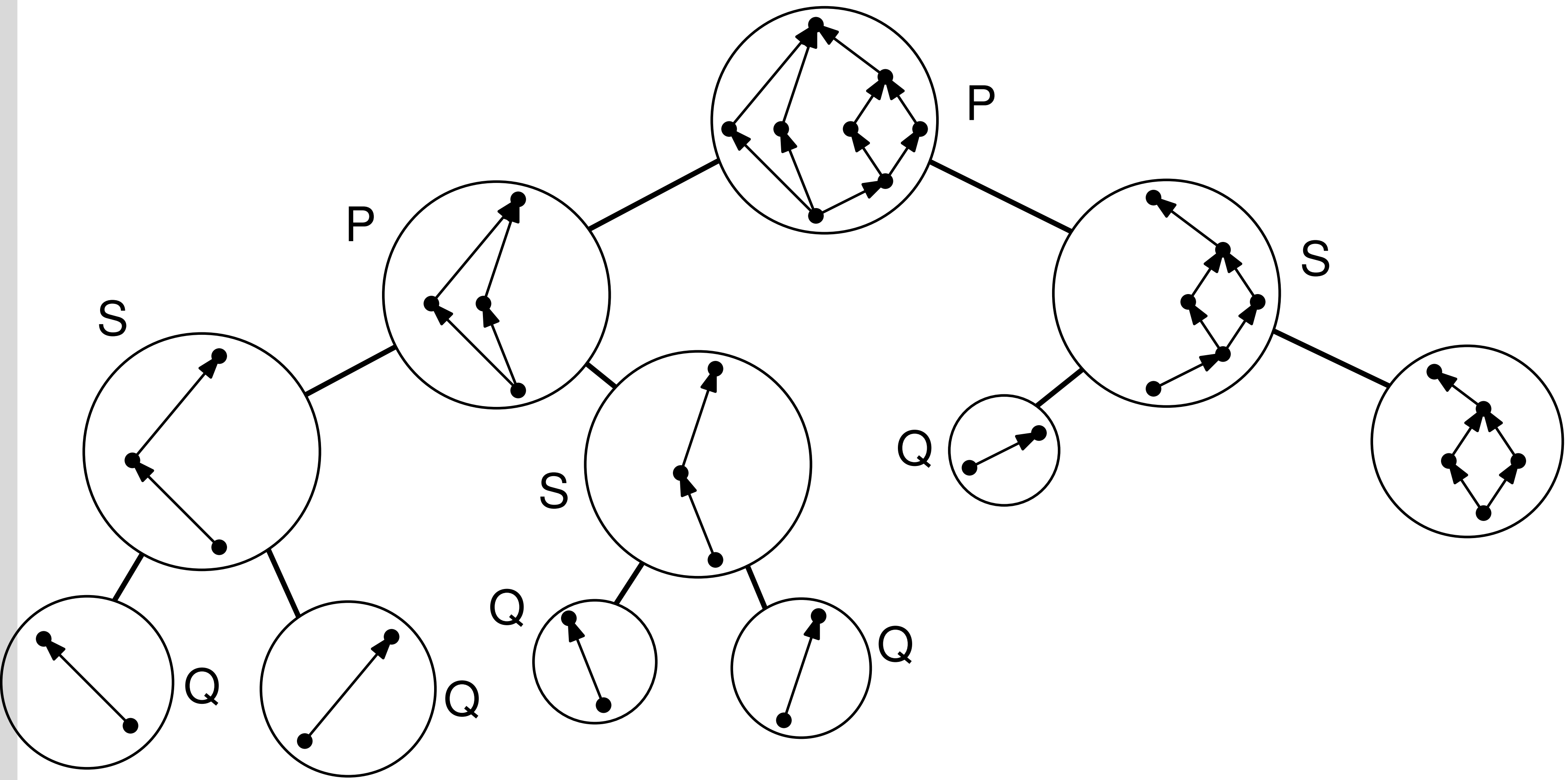
# Series-parallel Graphs. Decomposition Example.



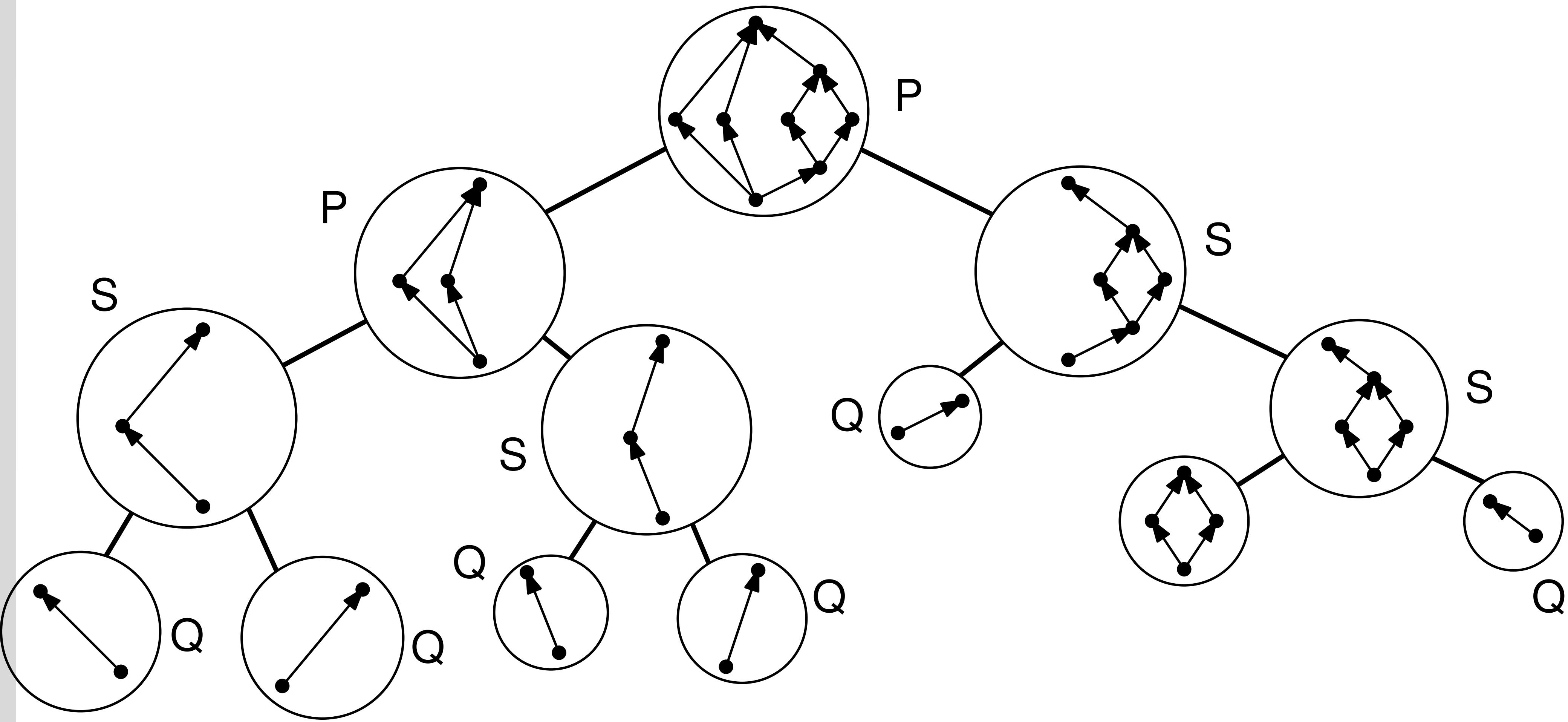
# Series-parallel Graphs. Decomposition Example



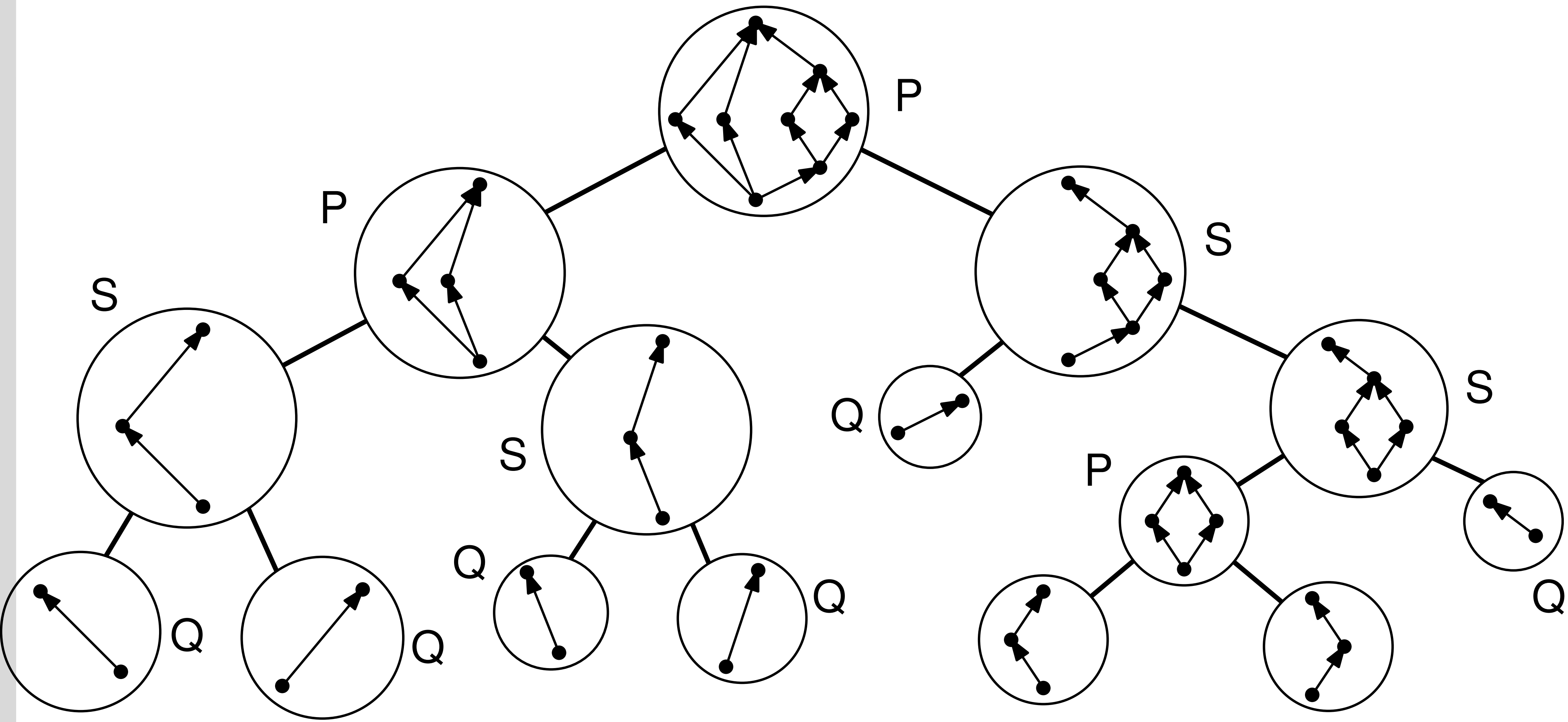
# Series-parallel Graphs. Decomposition Example.



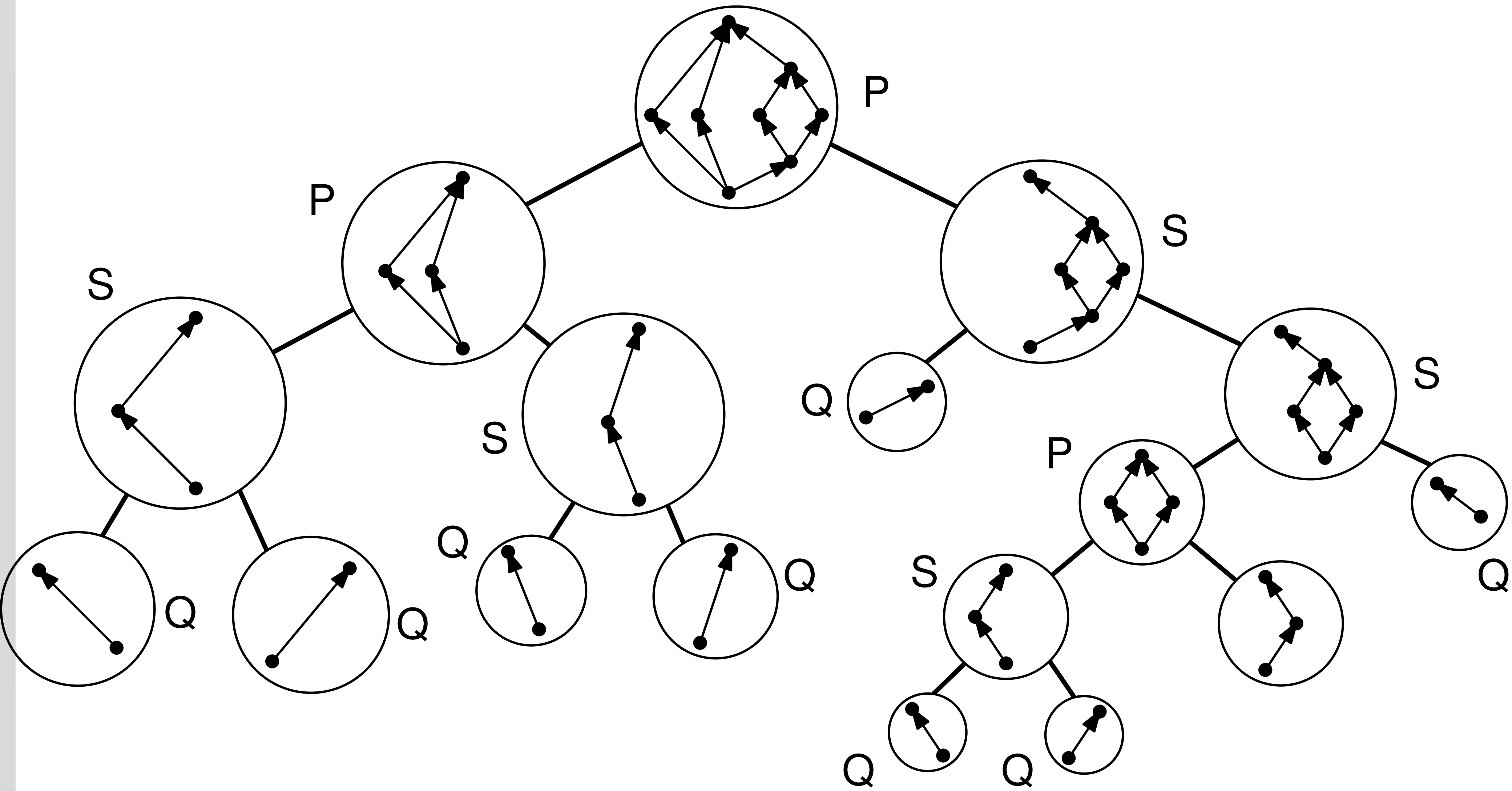
# Series-parallel Graphs. Decomposition Example.



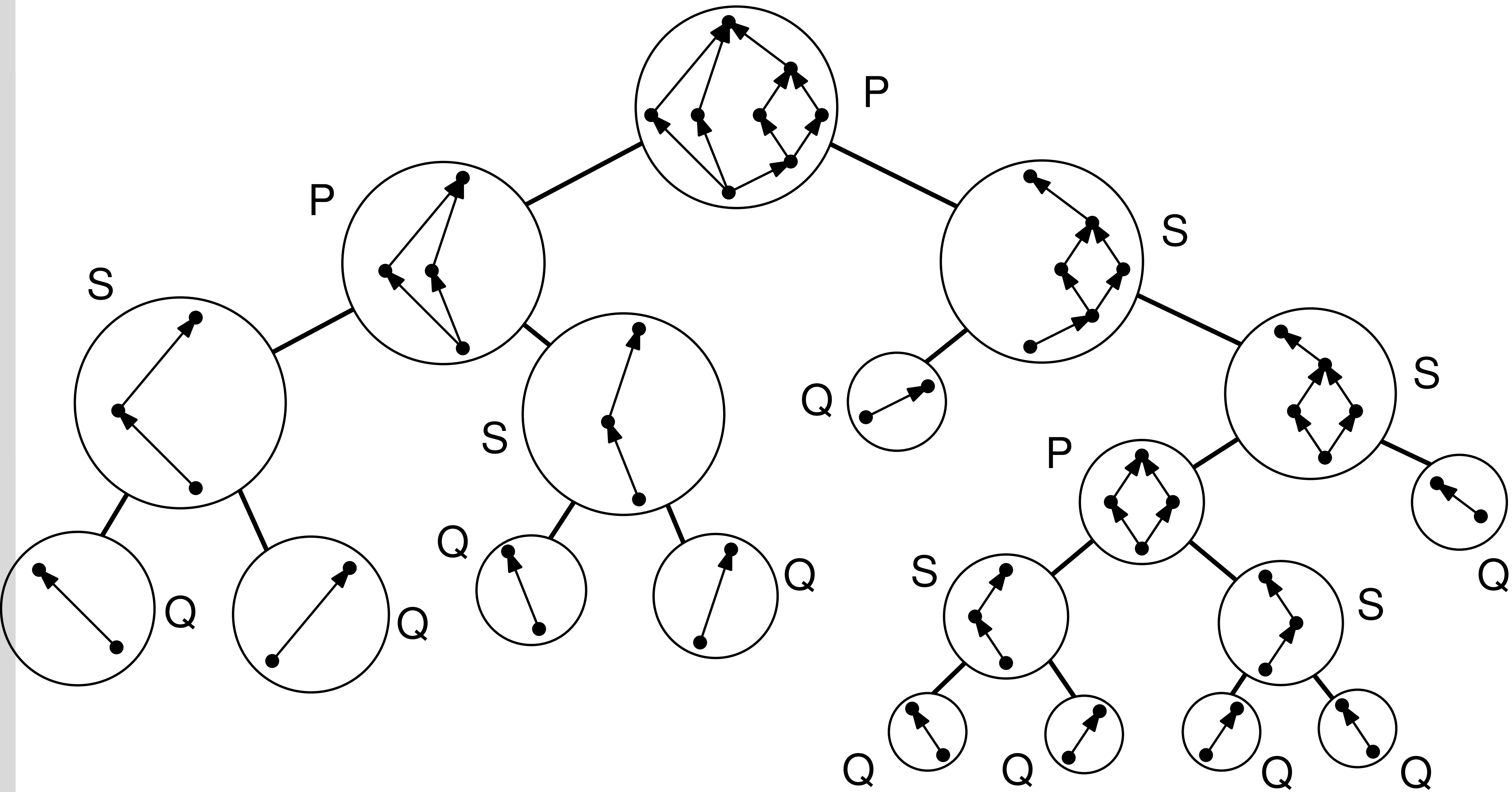
# Series-parallel Graphs. Decomposition Example.



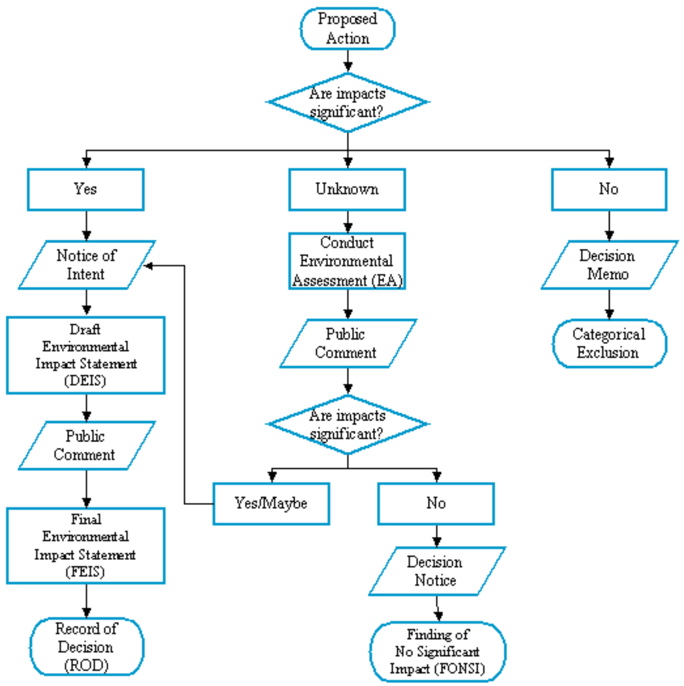
# Series-parallel Graphs. Decomposition Example.



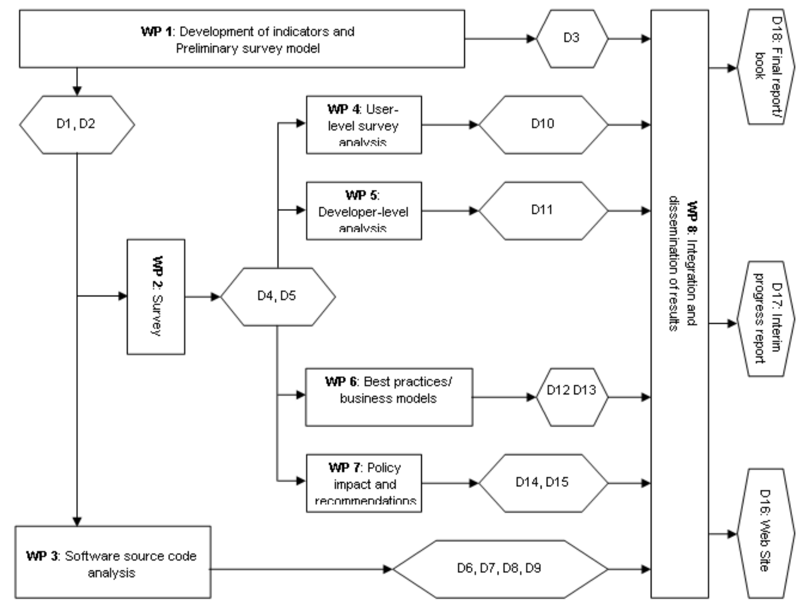
# Series-parallel Graphs. Decomposition Example.



# Series-parallel Graphs. Applications.



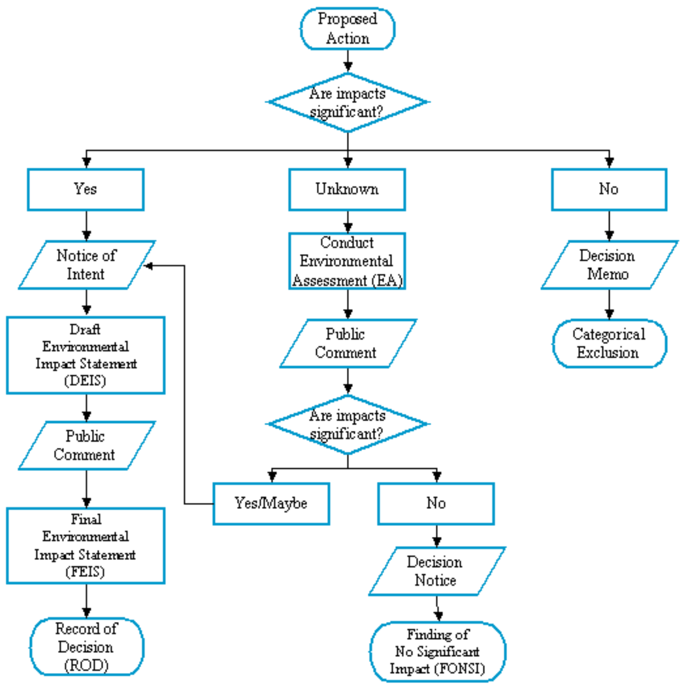

Flowcharts



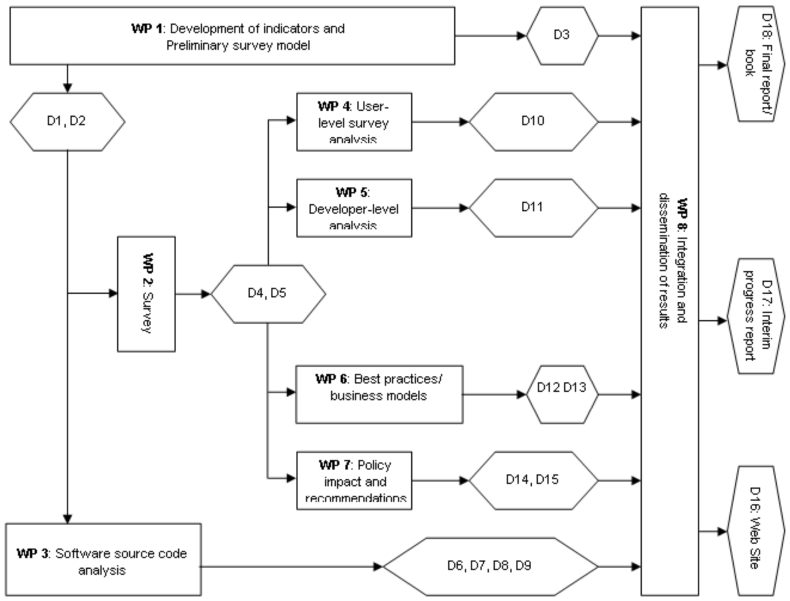

PERT-Diagrams

(Program Evaluation and Review Technique)

# Series-parallel Graphs. Applications.




Flowcharts



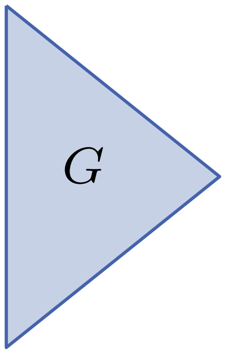

PERT-Diagrams

(Program Evaluation and Review Technique)

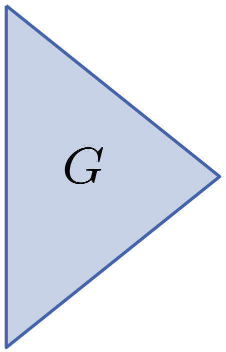
**Computational Complexity:** Linear time algorithms for $\mathcal{NP}$-hard problems (e.g. Maximum Matching, Maximum Independent Set, Hamiltonian Completion)

# Straight-line Drawing of SP-Graphs

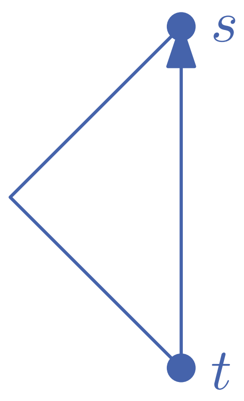
- Draw graph $G$ inside a right-angled isosceles bounding triangle $\Delta(G)$



# Straight-line Drawing of SP-Graphs

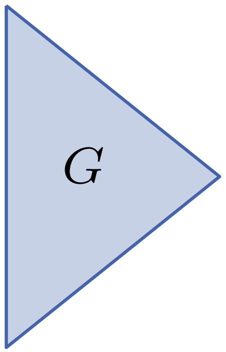


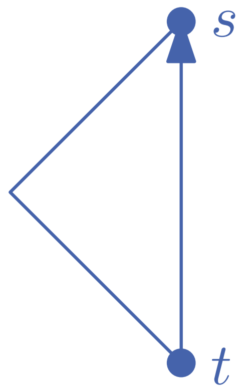

- Draw graph $G$ inside a right-angled isosceles bounding triangle $\Delta(G)$
- Q-Nodes (Induction base):

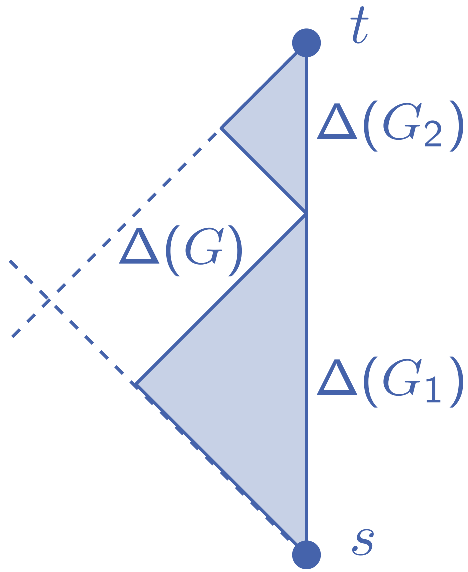


# Straight-line Drawing of SP-Graphs

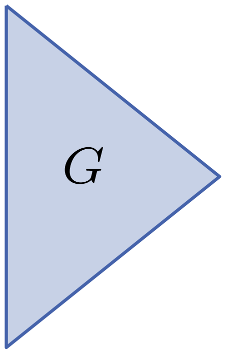
- Draw graph $G$ inside a right-angled isosceles bounding triangle $\Delta(G)$
- Q-Nodes (Induction base):
- S-Nodes (series composition)



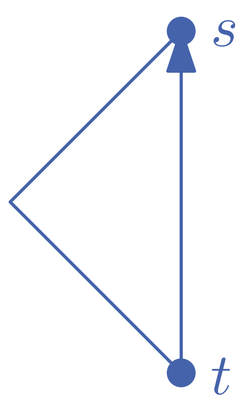



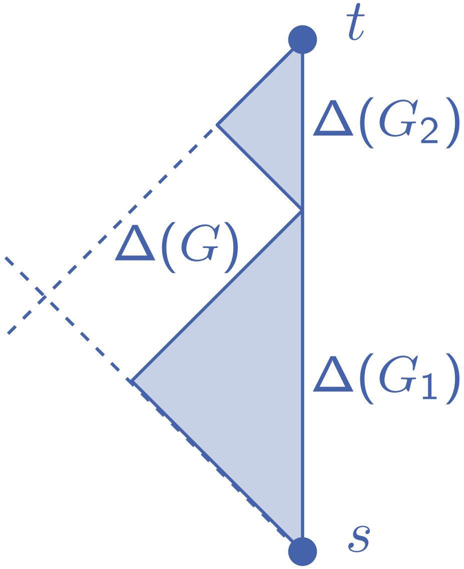



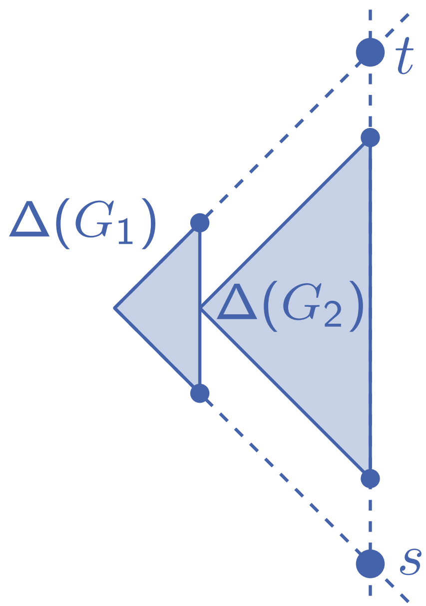
# Straight-line Drawing of SP-Graphs



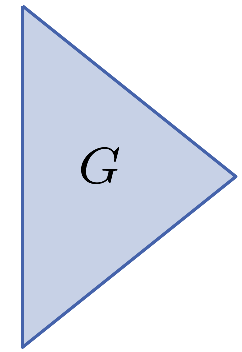

- Draw graph $G$ inside a right-angled isosceles bounding triangle $\Delta(G)$
- Q-Nodes (Induction base):
- S-Nodes (series composition)
- P-Nodes (parallel composition)



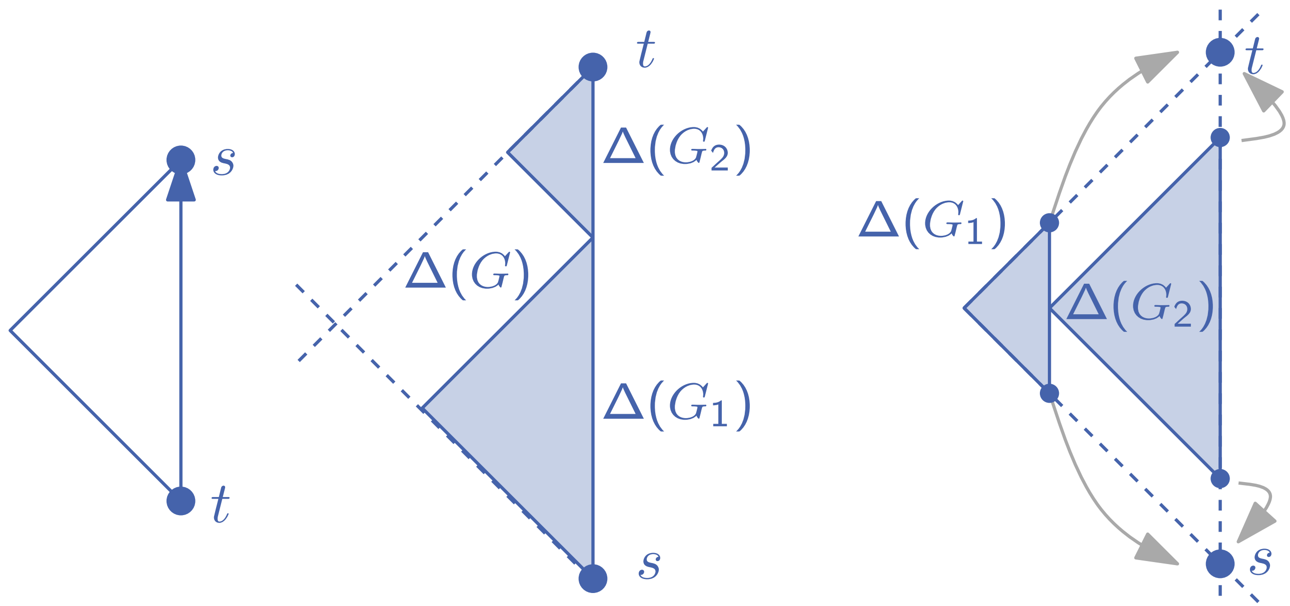



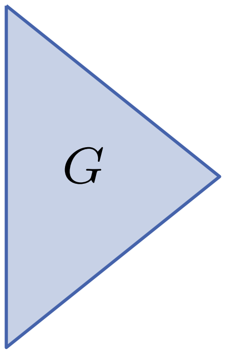
# Straight-line Drawing of SP-Graphs



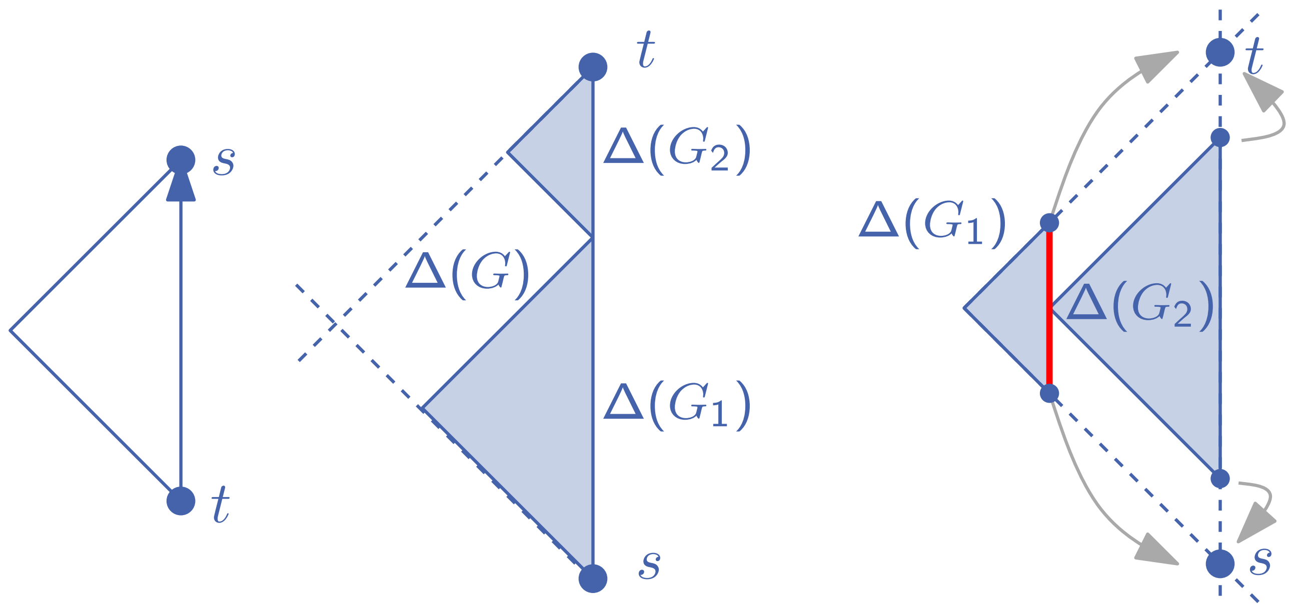

- Draw graph $G$ inside a right-angled isosceles bounding triangle $\Delta(G)$
- Q-Nodes (Induction base):
- S-Nodes (series composition)
- P-Nodes (parallel composition)



# Straight-line Drawing of SP-Graphs

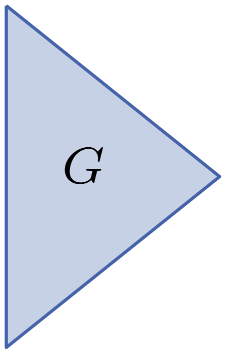


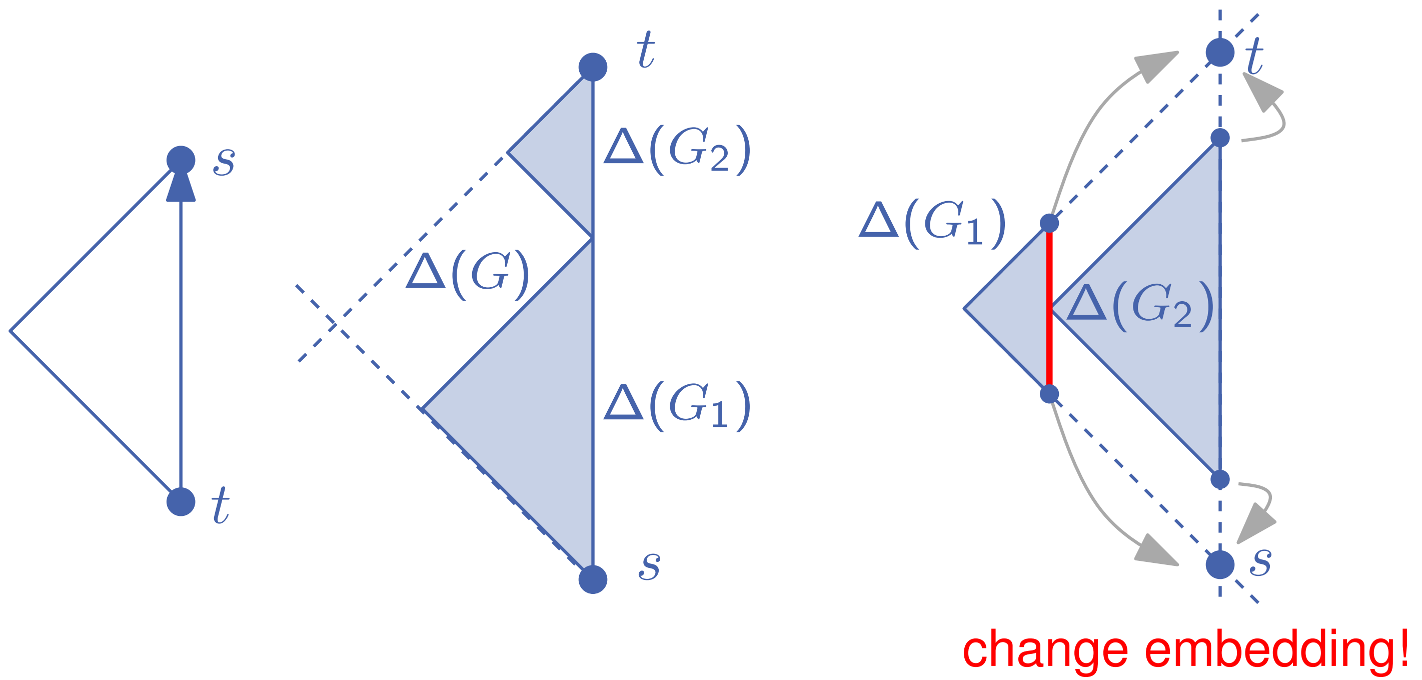

- Draw graph $G$ inside a right-angled isosceles bounding triangle $\Delta(G)$
- Q-Nodes (Induction base):
- S-Nodes (series composition)
- P-Nodes (parallel composition)



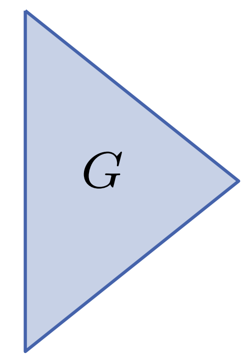
# Straight-line Drawing of SP-Graphs

- Draw graph $G$ inside a right-angled isosceles bounding triangle $\Delta(G)$
- Q-Nodes (Induction base):
- S-Nodes (series composition)
- P-Nodes (parallel composition)

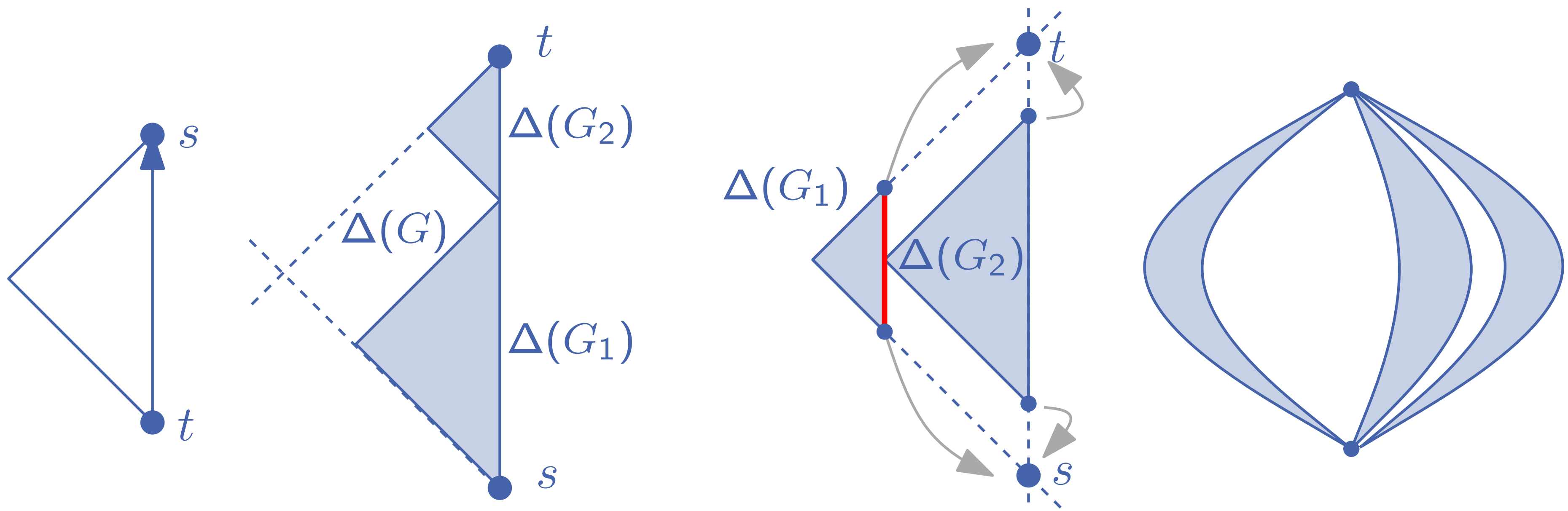


# Straight-line Drawing of SP-Graphs

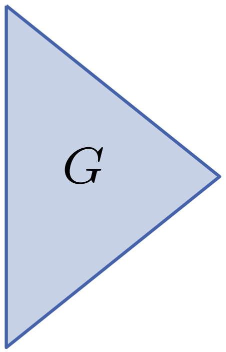



- Draw graph $G$ inside a right-angled isosceles bounding triangle $\Delta(G)$
- Q-Nodes (Induction base):
- S-Nodes (series composition)
- P-Nodes (parallel composition)

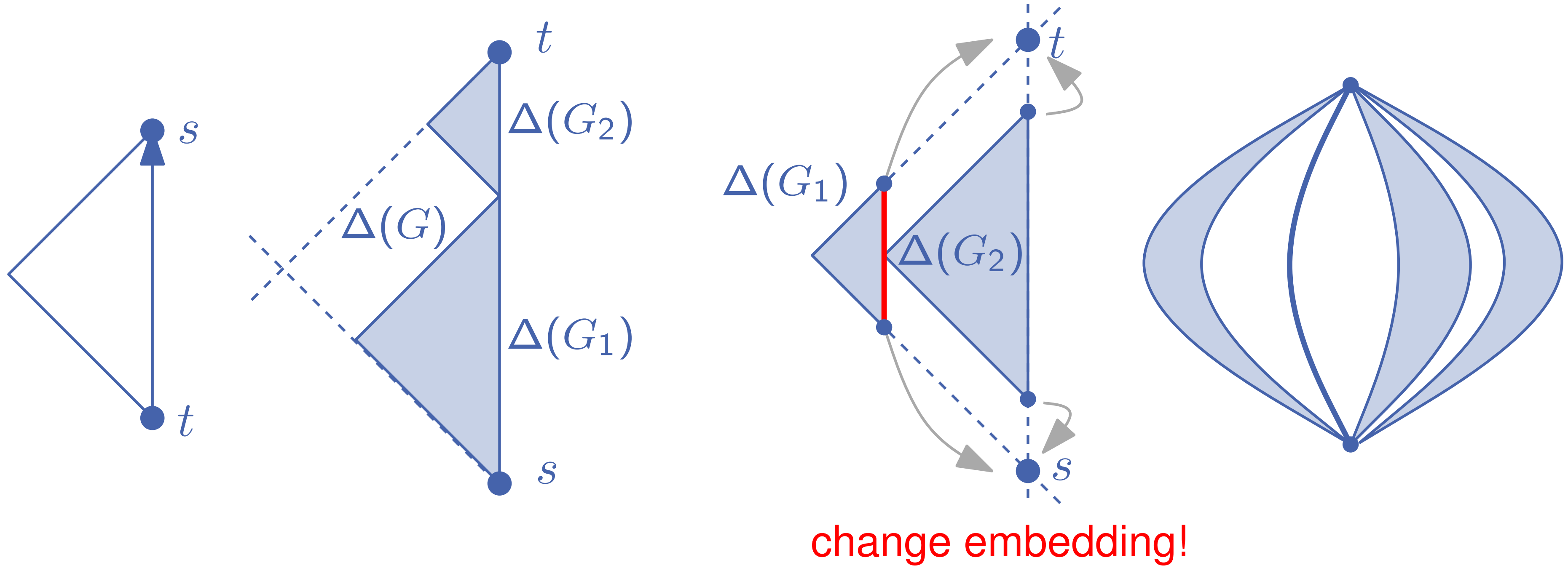



change embedding!

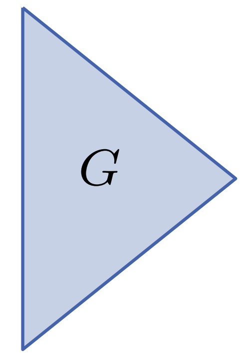
# Straight-line Drawing of SP-Graphs



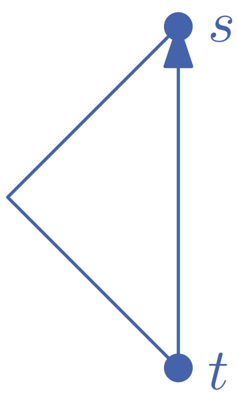

- Draw graph $G$ inside a right-angled isosceles bounding triangle $\Delta(G)$
- Q-Nodes (Induction base):
- S-Nodes (series composition)
- P-Nodes (parallel composition)

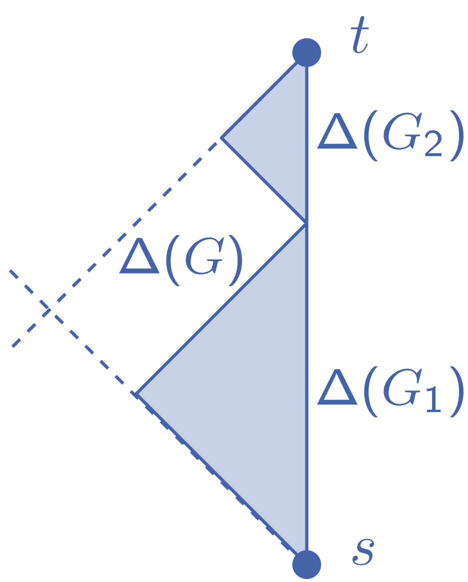


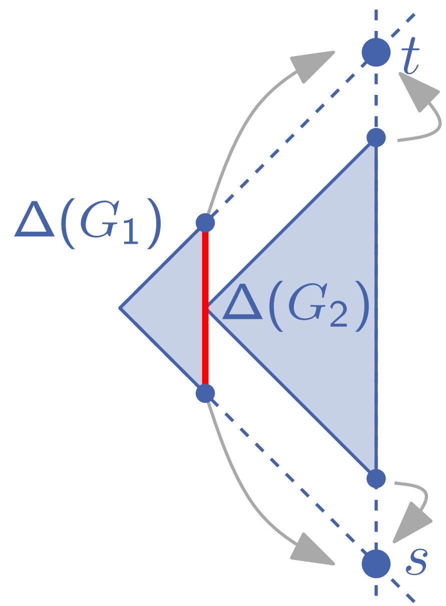
# Straight-line Drawing of SP-Graphs



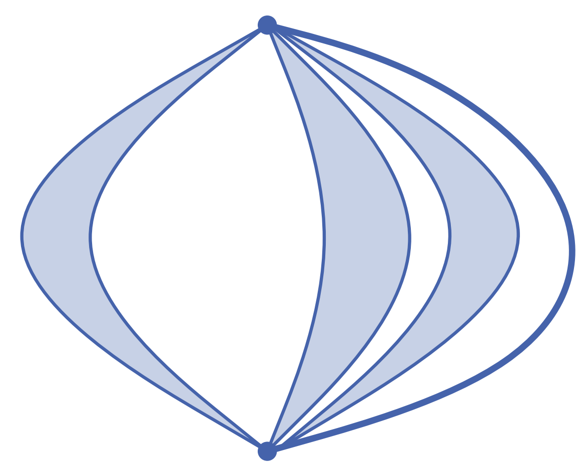

- Draw graph $G$ inside a right-angled isosceles bounding triangle $\Delta(G)$
- Q-Nodes (Induction base):
- S-Nodes (series composition)
- P-Nodes (parallel composition)



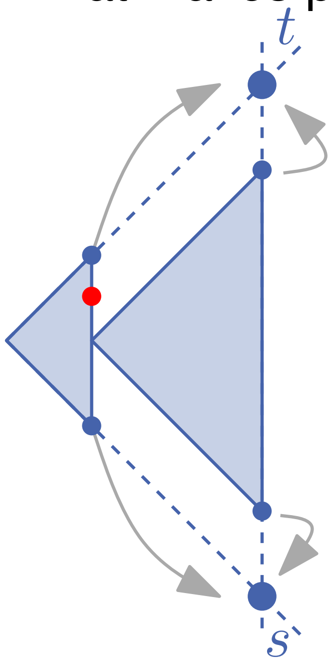




change embedding!



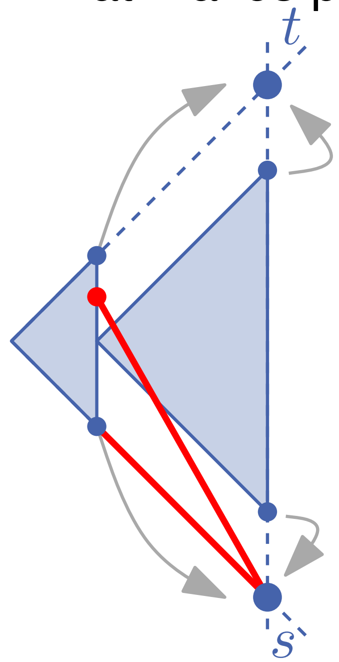
# Straight-line Drawing of SP-Graphs

- What makes parallel composition possible without creating crossings?



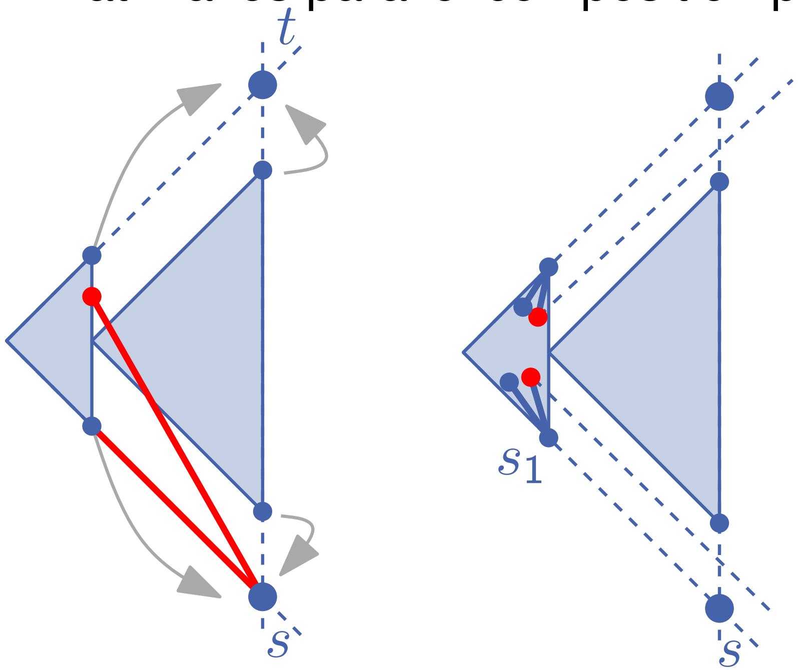
# Straight-line Drawing of SP-Graphs

- What makes parallel composition possible without creating crossings?



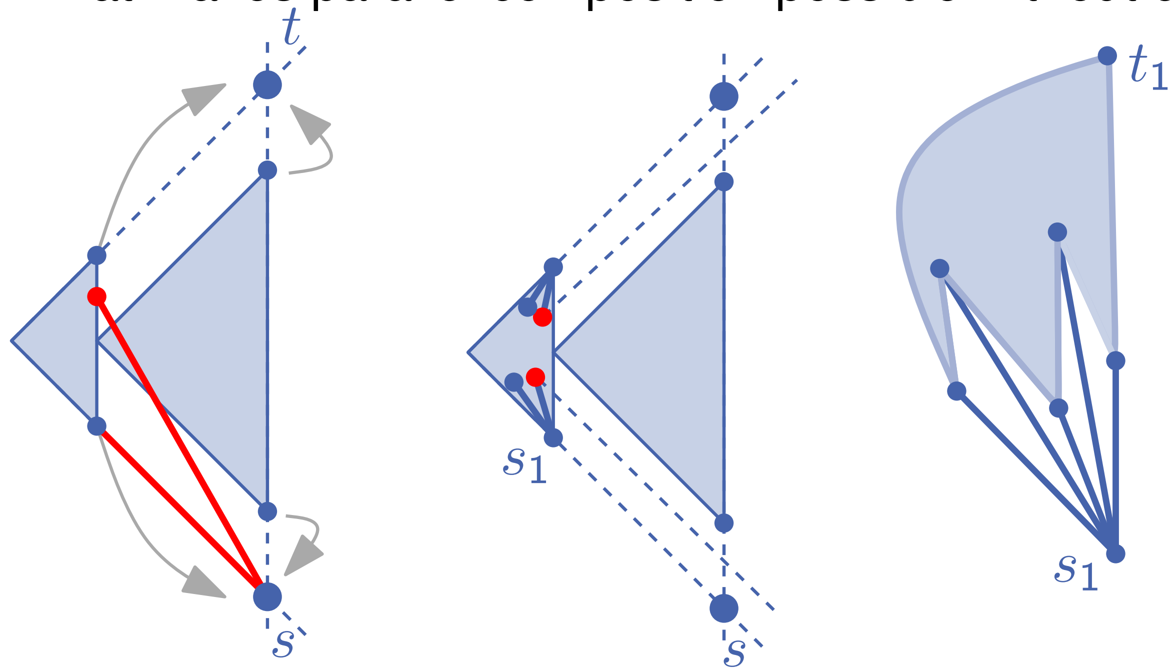
# Straight-line Drawing of SP-Graphs

- What makes parallel composition possible without creating crossings?



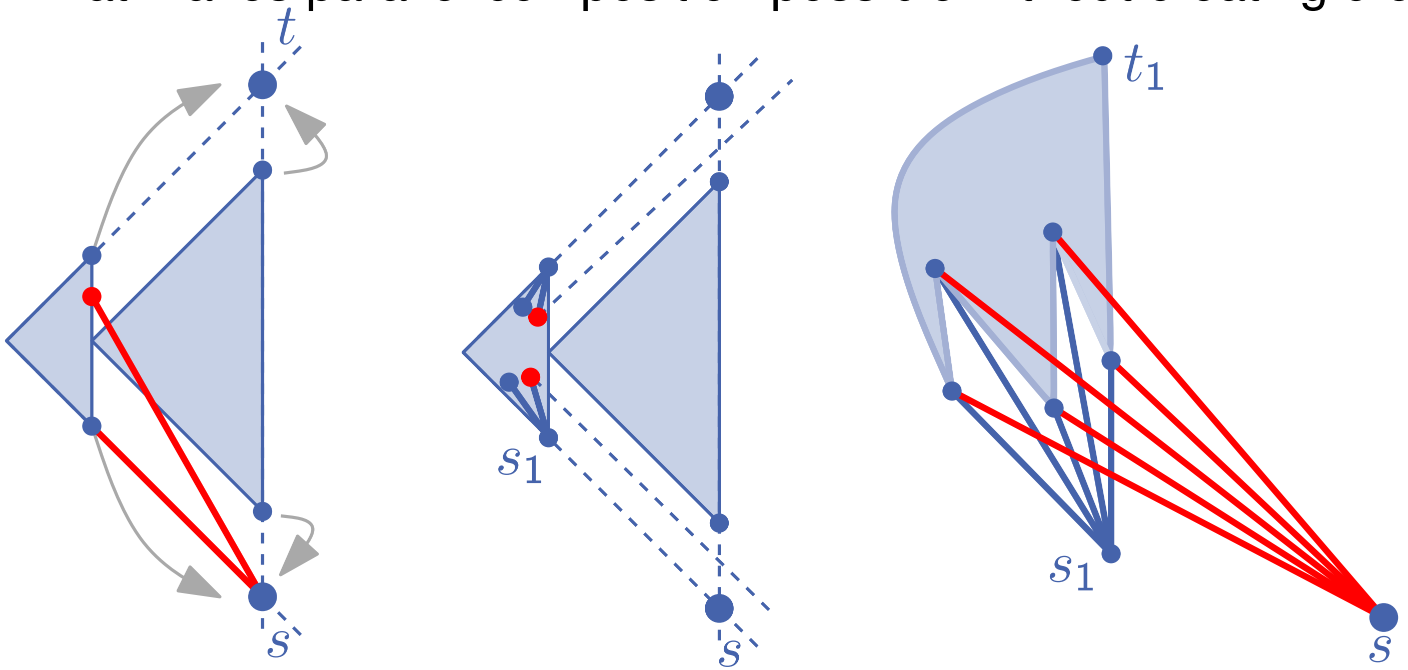
# Straight-line Drawing of SP-Graphs

- What makes parallel composition possible without creating crossings?



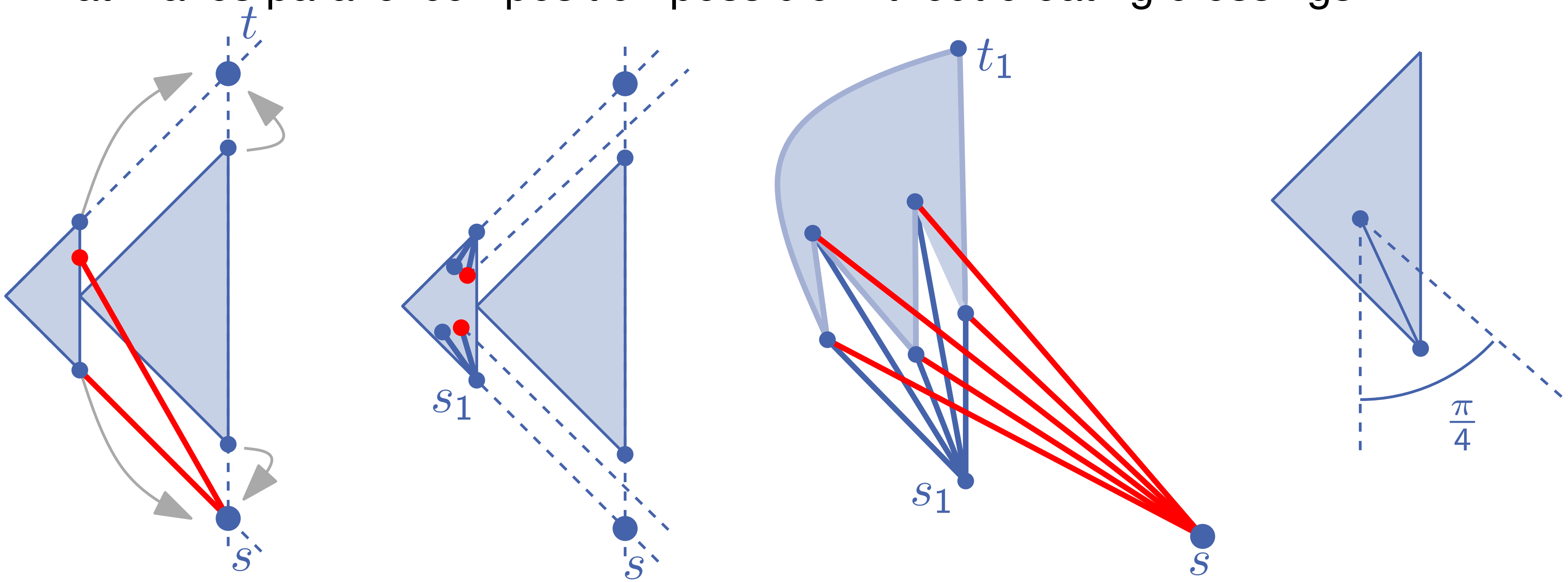
# Straight-line Drawing of SP-Graphs

- What makes parallel composition possible without creating crossings?



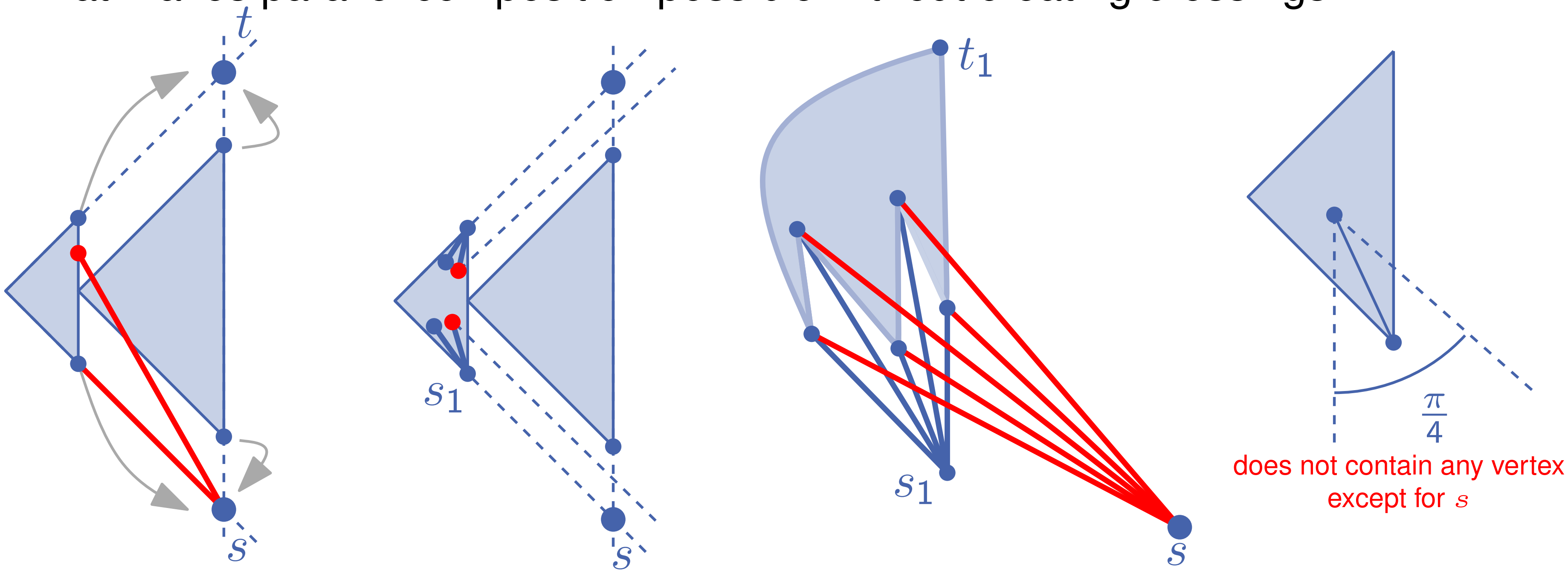
# Straight-line Drawing of SP-Graphs

- What makes parallel composition possible without creating crossings?



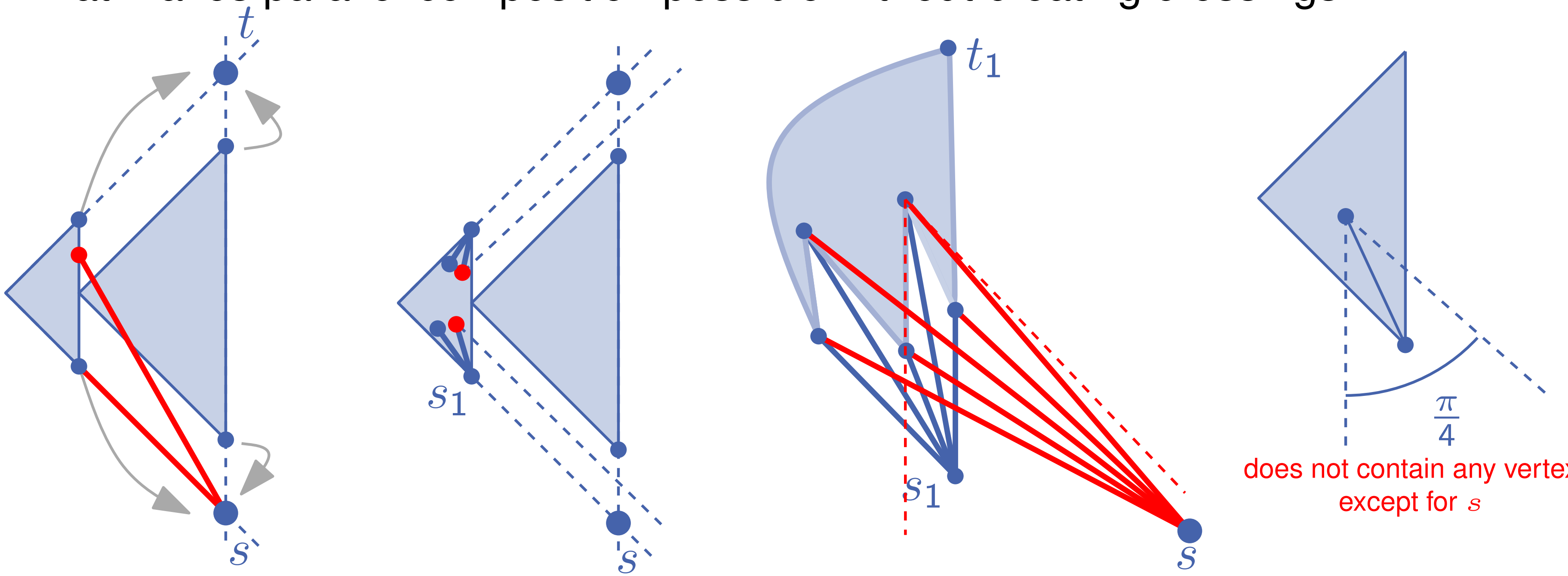
# Straight-line Drawing of SP-Graphs

- What makes parallel composition possible without creating crossings?



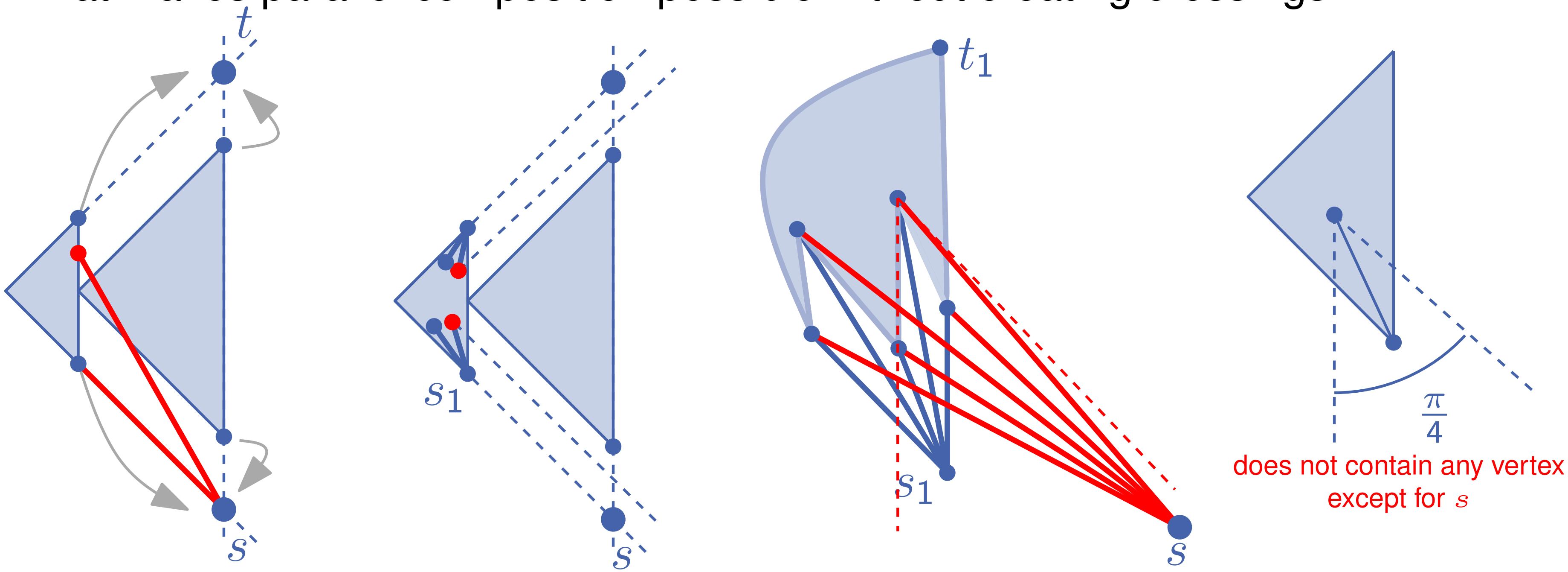
# Straight-line Drawing of SP-Graphs

- What makes parallel composition possible without creating crossings?



# Straight-line Drawing of SP-Graphs

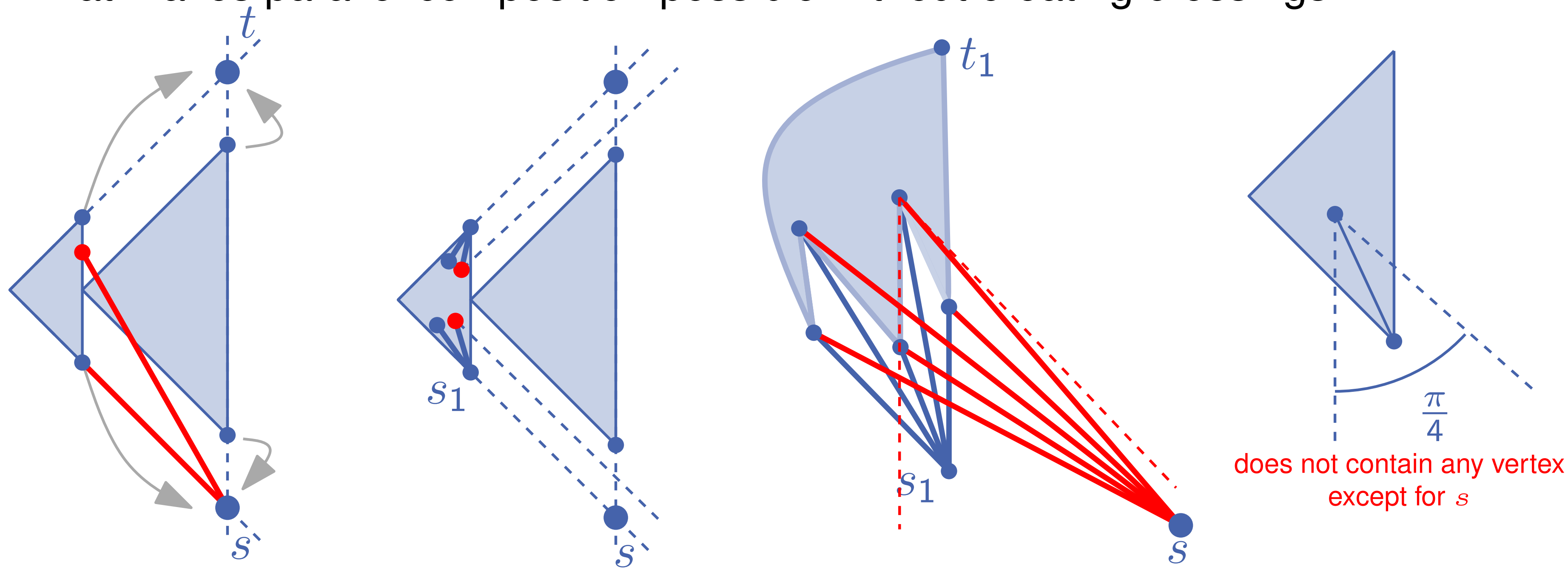
- What makes parallel composition possible without creating crossings?




**Lemma** If this condition holds then parallel composition results in a planar drawing.

# Straight-line Drawing of SP-Graphs

- What makes parallel composition possible without creating crossings?

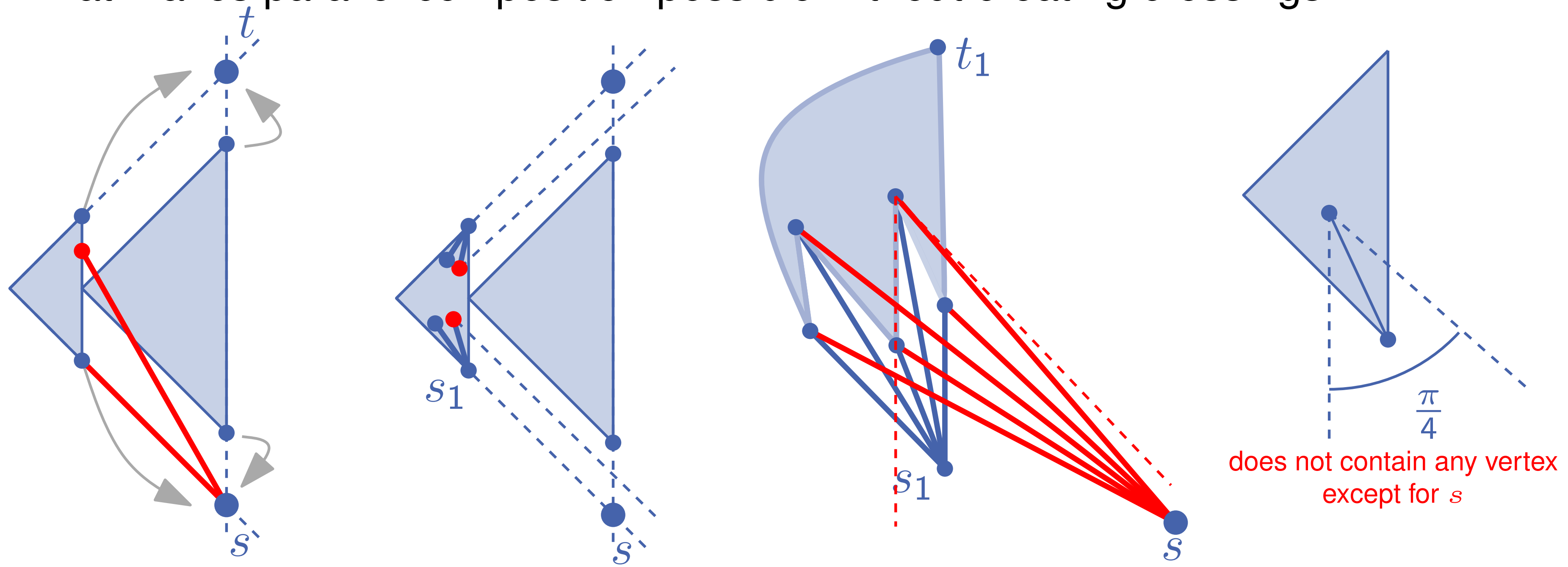



**Lemma** If this condition holds then parallel composition results in a planar drawing.

- The condition can be preserved during the induction step.

# Straight-line Drawing of SP-Graphs

- What makes parallel composition possible without creating crossings?

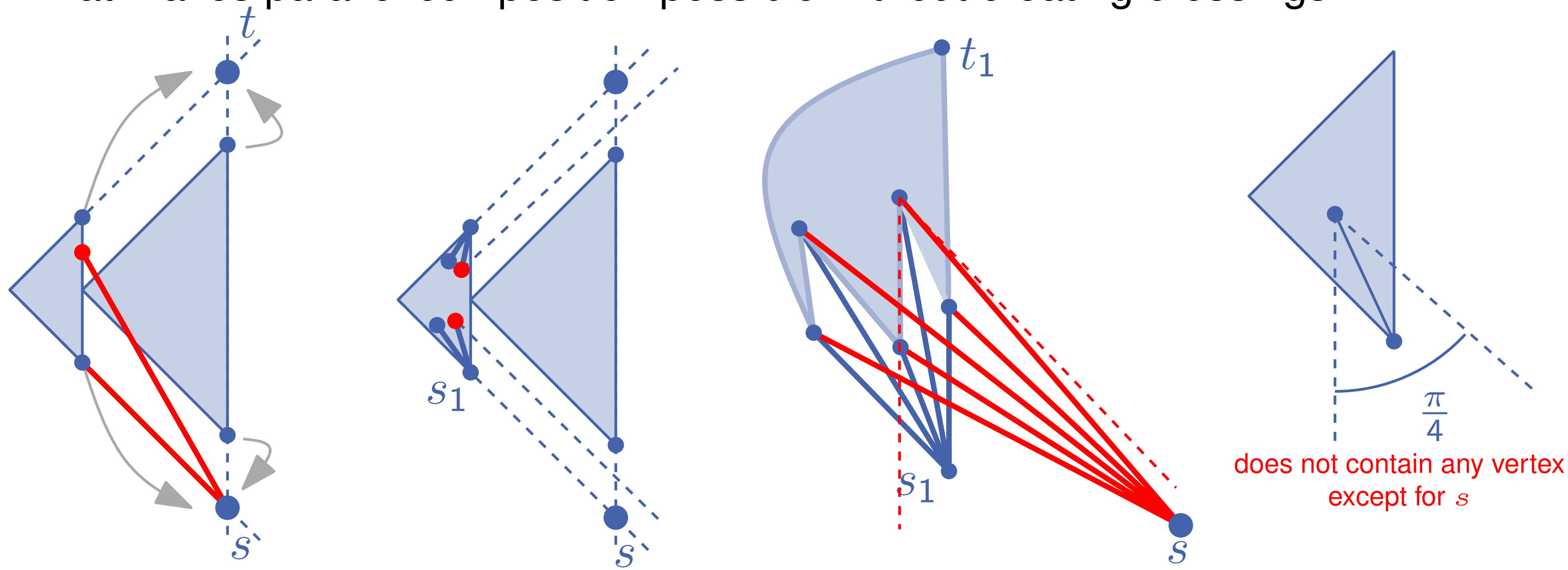



**Lemma** If this condition holds then parallel composition results in a planar drawing.

- The condition can be preserved during the induction step.
- The area of the drawing is?

# Straight-line Drawing of SP-Graphs

- What makes parallel composition possible without creating crossings?

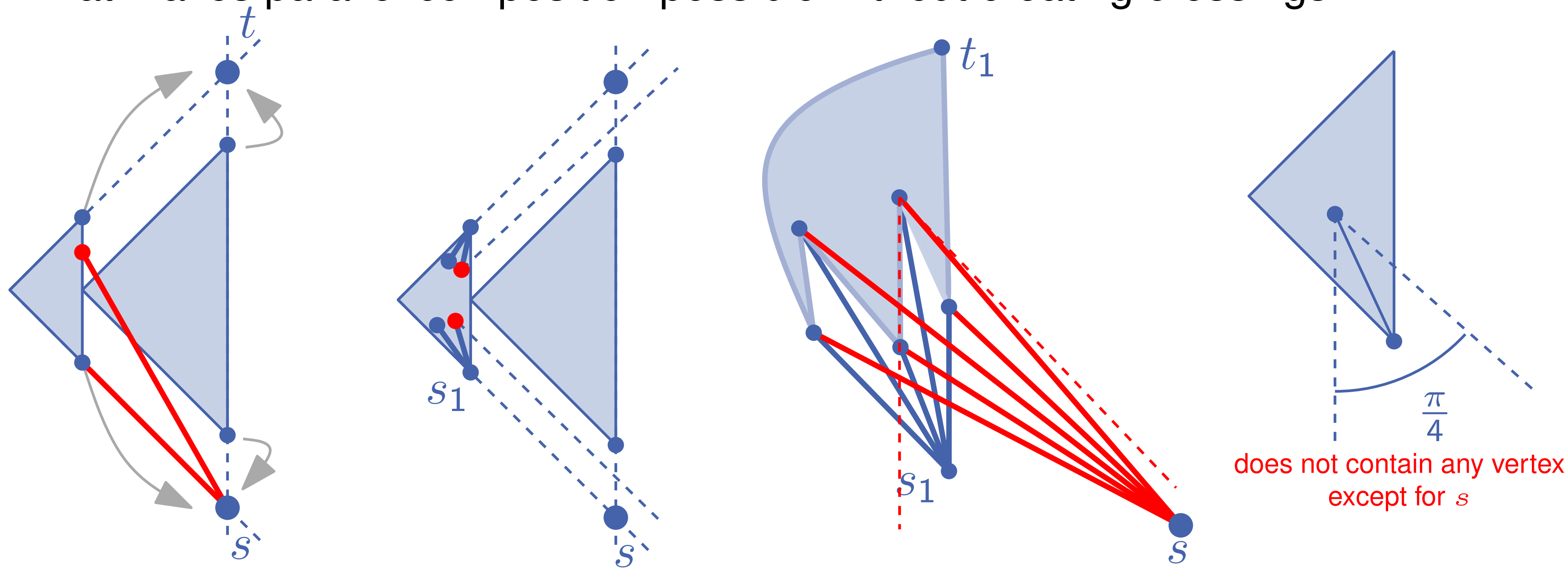



**Lemma** If this condition holds then parallel composition results in a planar drawing.

- The condition can be preserved during the induction step.
- The area of the drawing is? $O(m^2), m$ is the number of edges

# Straight-line Drawing of SP-Graphs

- What makes parallel composition possible without creating crossings?




## Theorem

A series-parallel graph $G$ (**with variable embedding**) admits an **upward planar** straigh-line drawing with $O(n^2)$ area.

# Lower Bound for the Area

**Theorem [Bertolazzi et al. 94]**

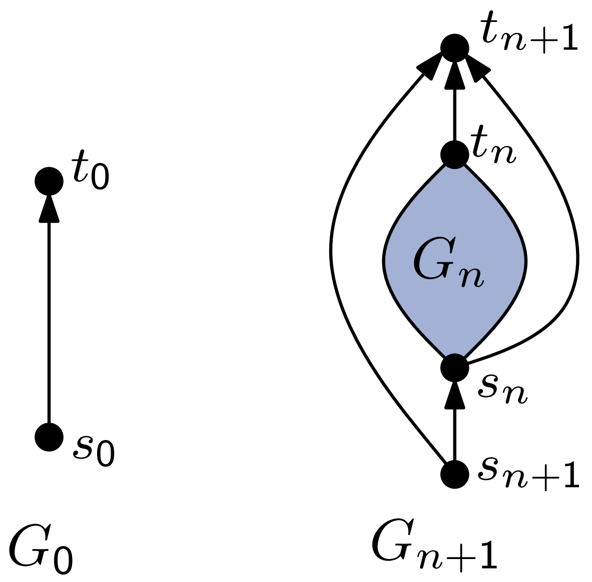
There exists a $2n$-vertex series-parallel graph $G_n$ such that any upward planar drawing of $G_n$ respecting embedding requires area $\Omega(4^n)$.

# Lower Bound for the Area

**Theorem [Bertolazzi et al. 94]**

There exists a $2n$-vertex series-parallel graph $G_n$ such that any upward planar drawing of $G_n$ respecting embedding requires area $\Omega(4^n)$.

Proof:

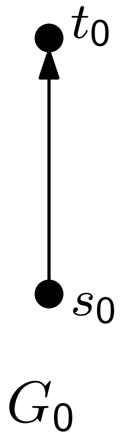


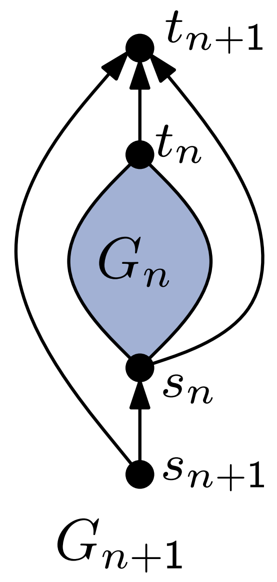
# Lower Bound for the Area

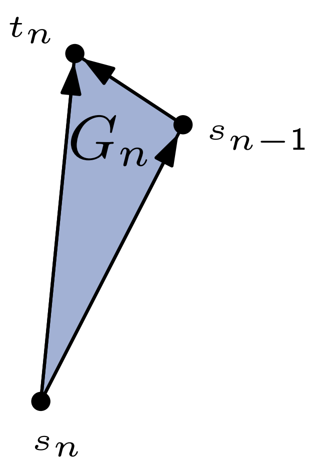
**Theorem [Bertolazzi et al. 94]**

There exists a $2n$-vertex series-parallel graph $G_n$ such that any upward planar drawing of $G_n$ respecting embedding requires area $\Omega(4^n)$.

Proof:



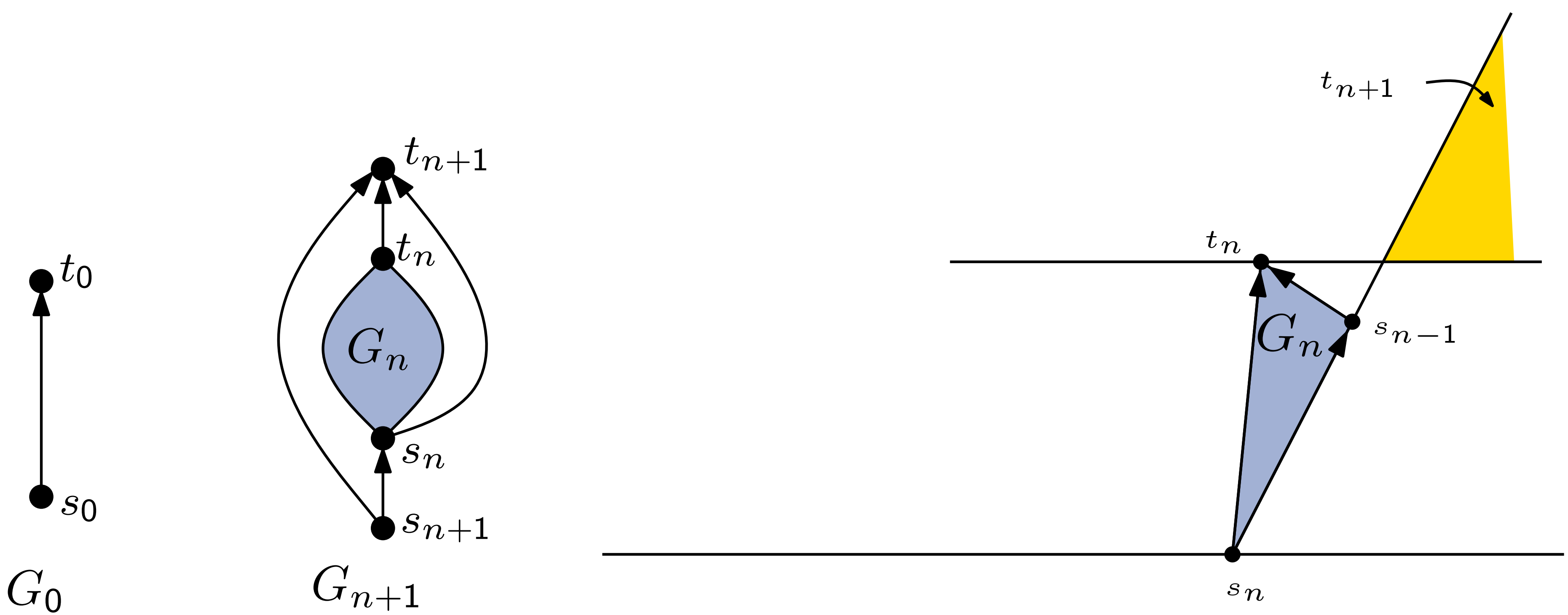
# Lower Bound for the Area



## Theorem [Bertolazzi et al. 94]

There exists a $2n$-vertex series-parallel graph $G_n$ such that any upward planar drawing of $G_n$ respecting embedding requires area $\Omega(4^n)$.

Proof:



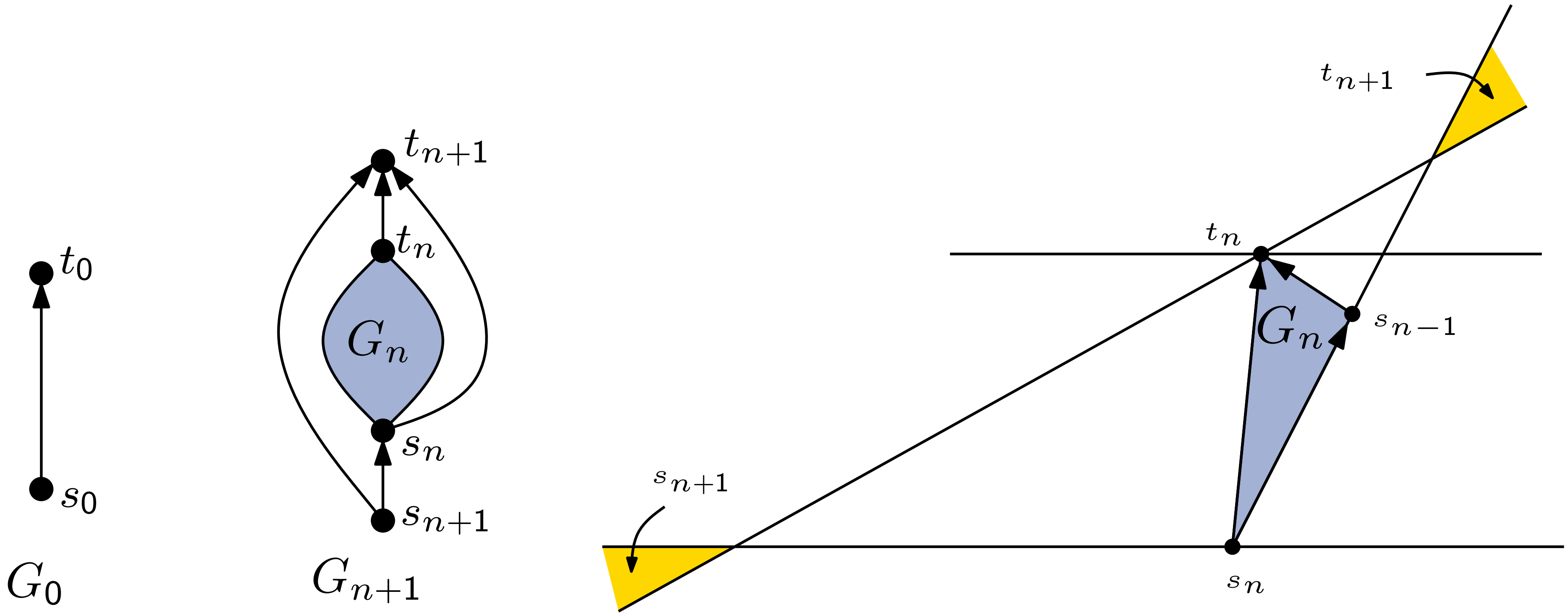
# Lower Bound for the Area

Karlsruhe Institute of Technology

**Theorem [Bertolazzi et al. 94]**

There exists a $2n$-vertex series-parallel graph $G_n$ such that any upward planar drawing of $G_n$ respecting embedding requires area $\Omega(4^n)$.

Proof:





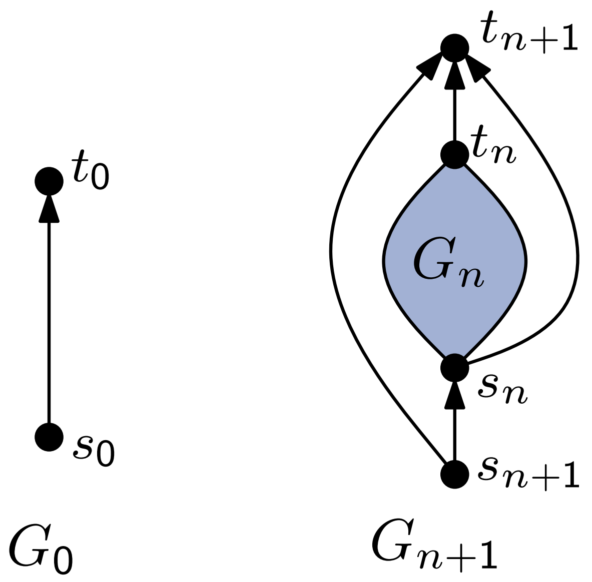

Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

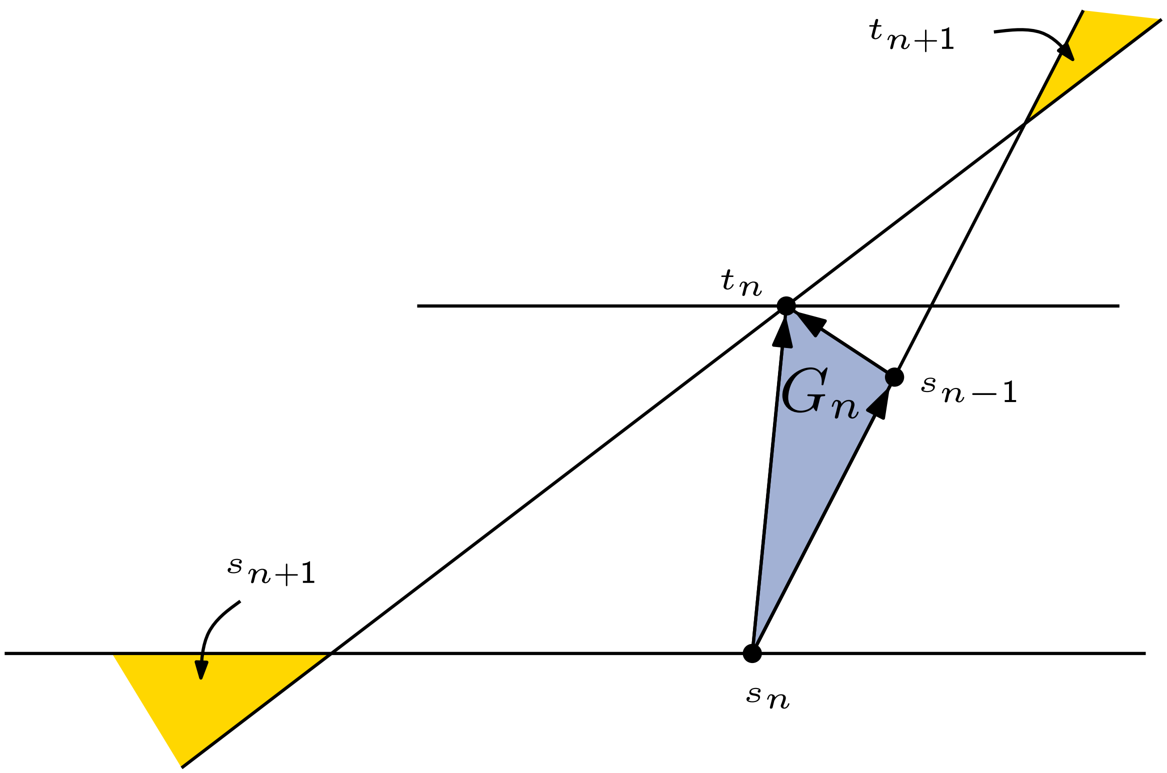
# Lower Bound for the Area

**Theorem [Bertolazzi et al. 94]**

There exists a $2n$-vertex series-parallel graph $G_n$ such that any upward planar drawing of $G_n$ respecting embedding requires area $\Omega(4^n)$.

Proof:



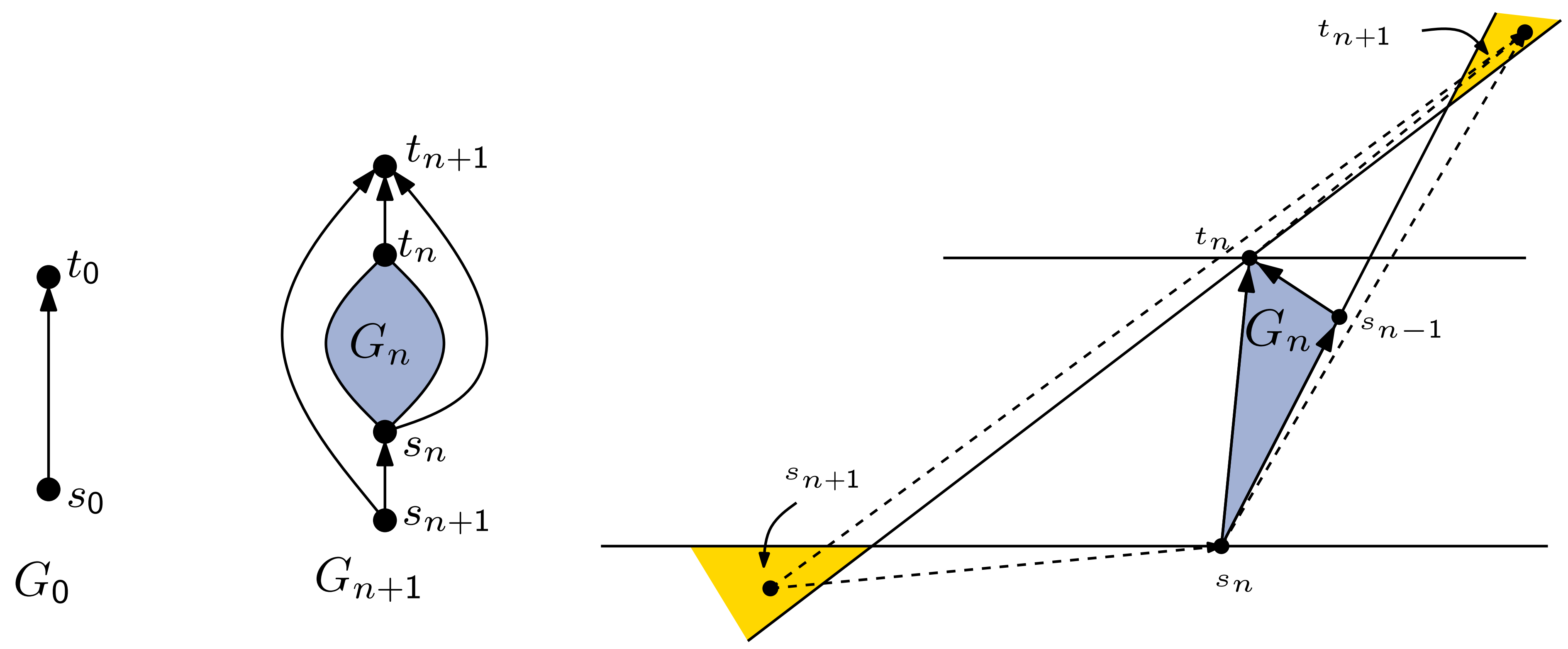



# Lower Bound for the Area

## Theorem [Bertolazzi et al. 94]

There exists a $2n$-vertex series-parallel graph $G_n$ such that any upward planar drawing of $G_n$ respecting embedding requires area $\Omega(4^n)$.

Proof:

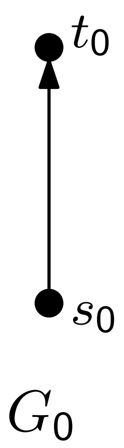


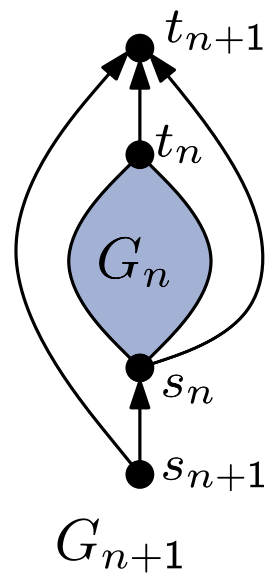
# Lower Bound for the Area

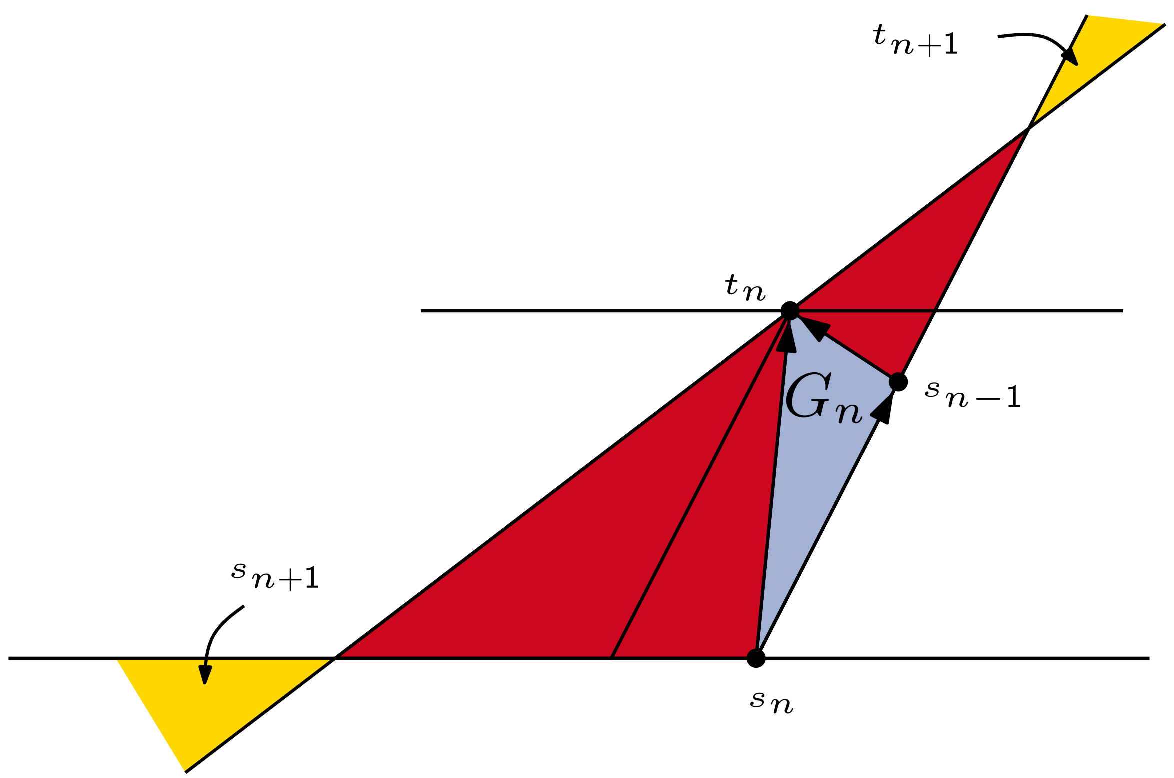
## Theorem [Bertolazzi et al. 94]

There exists a $2n$-vertex series-parallel graph $G_n$ such that any upward planar drawing of $G_n$ respecting embedding requires area $\Omega(4^n)$.

Proof:



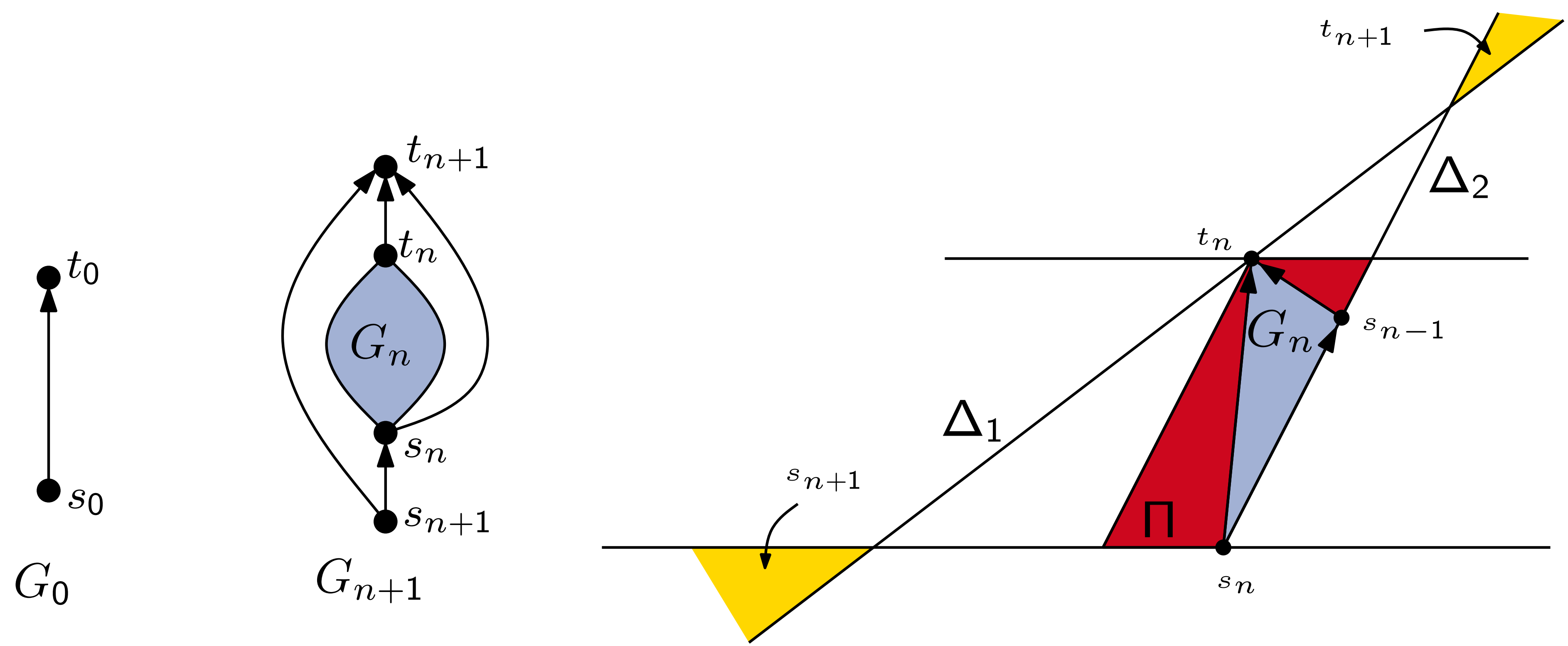



# Lower Bound for the Area

**Theorem [Bertolazzi et al. 94]**

There exists a $2n$-vertex series-parallel graph $G_n$ such that any upward planar drawing of $G_n$ respecting embedding requires area $\Omega(4^n)$.

Proof:



# Lower Bound for the Area

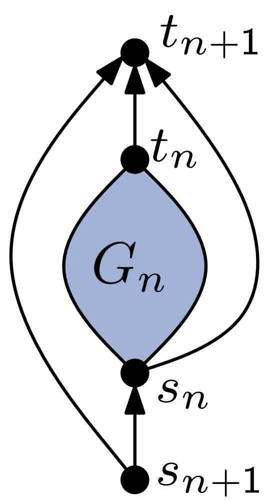
## Theorem [Bertolazzi et al. 94]

There exists a $2n$-vertex series-parallel graph $G_n$ such that any upward planar drawing of $G_n$ respecting embedding requires area $\Omega(4^n)$.

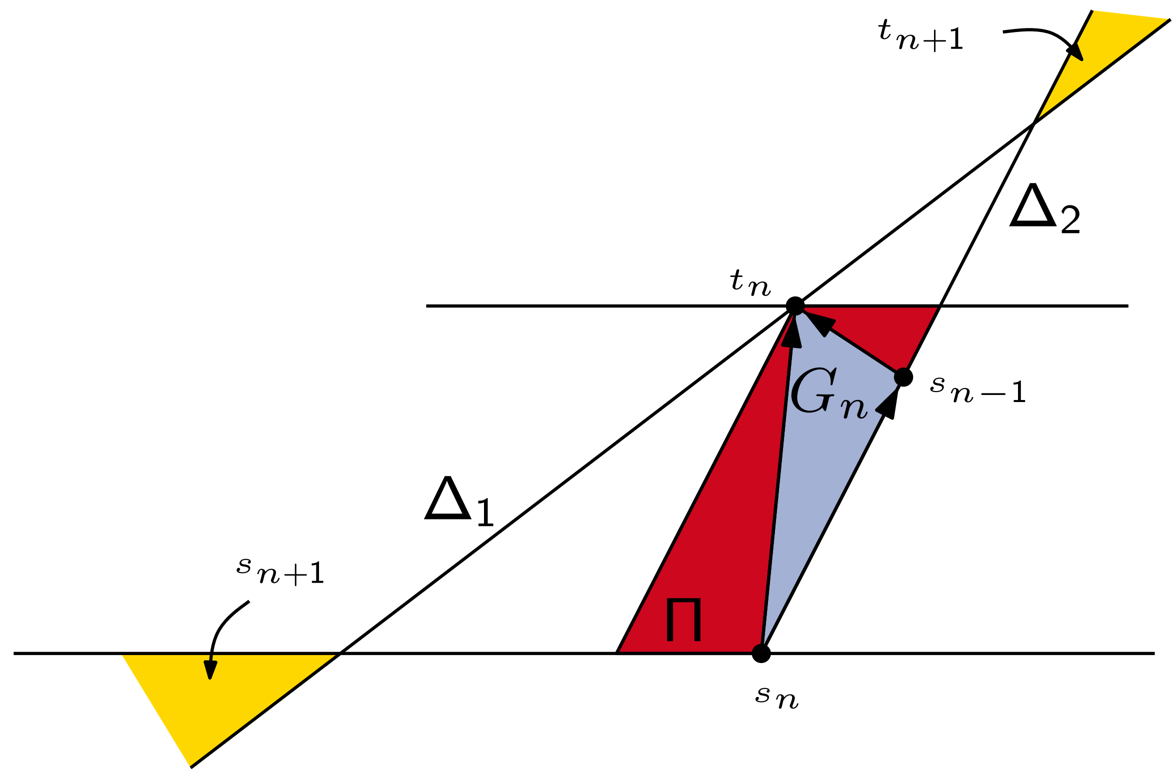
Proof:

- We have that: $Area(\Pi) > 2 \cdot Area(G_n)$


$G_0$




$G_{n+1}$



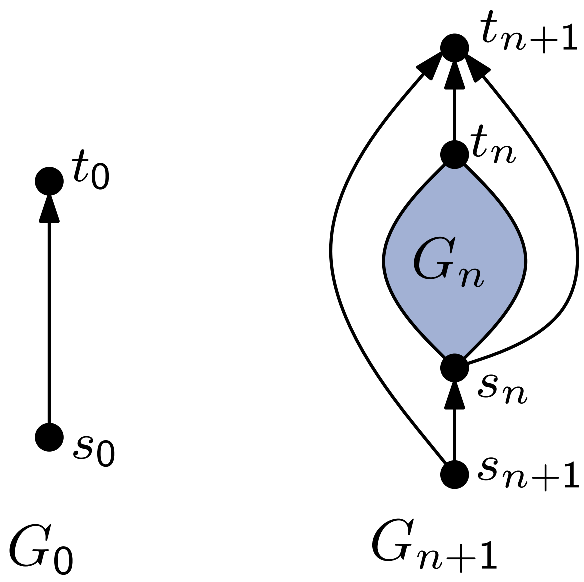
# Lower Bound for the Area

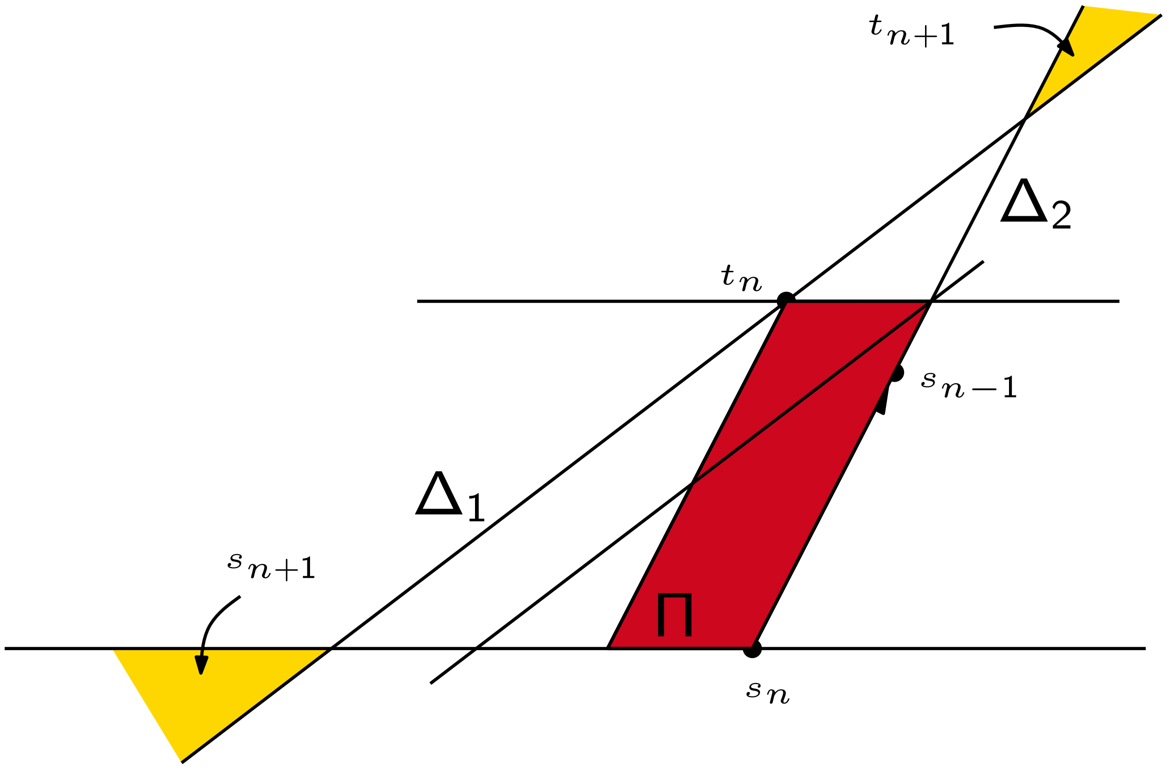
**Theorem [Bertolazzi et al. 94]**

There exists a $2n$-vertex series-parallel graph $G_n$ such that any upward planar drawing of $G_n$ respecting embedding requires area $\Omega(4^n)$.

Proof:

- We have that: $Area(\Pi) > 2 \cdot Area(G_n)$



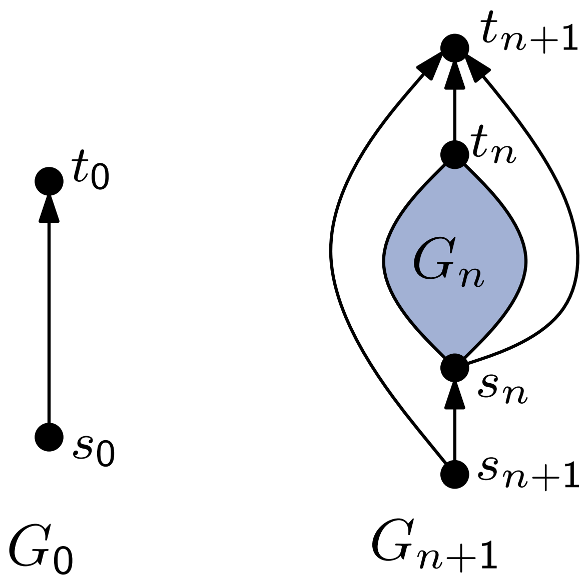
# Lower Bound for the Area

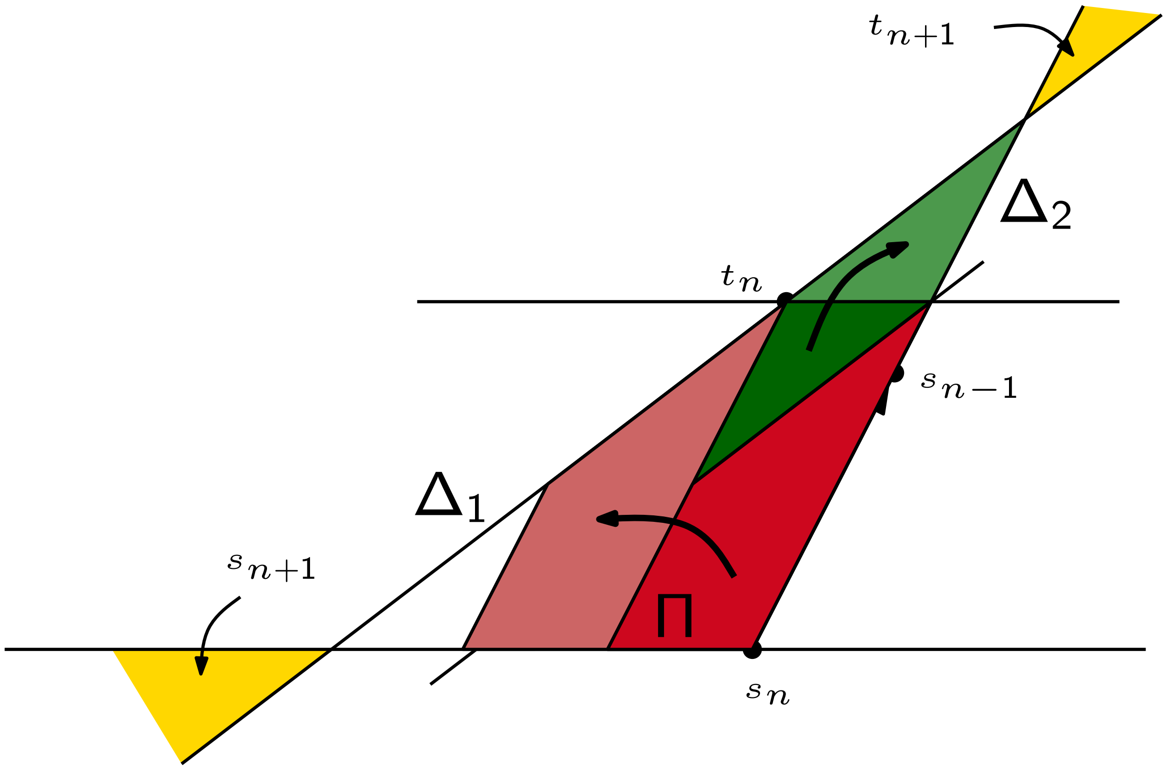
**Theorem [Bertolazzi et al. 94]**

There exists a $2n$-vertex series-parallel graph $G_n$ such that any upward planar drawing of $G_n$ respecting embedding requires area $\Omega(4^n)$.

Proof:

- We have that: $Area(\Pi) > 2 \cdot Area(G_n)$



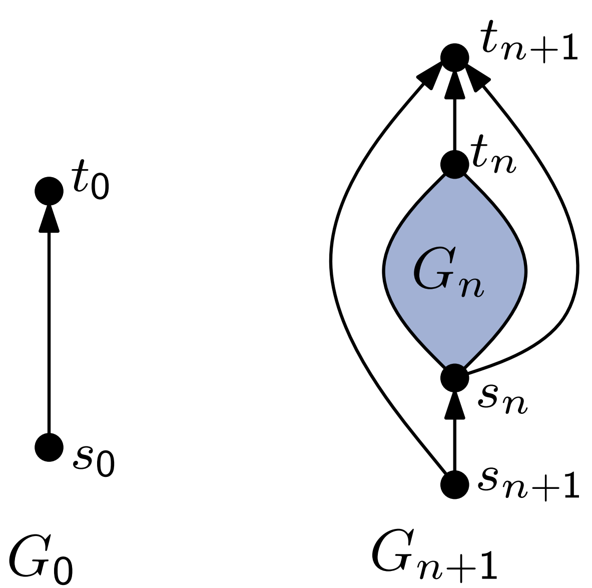
# Lower Bound for the Area

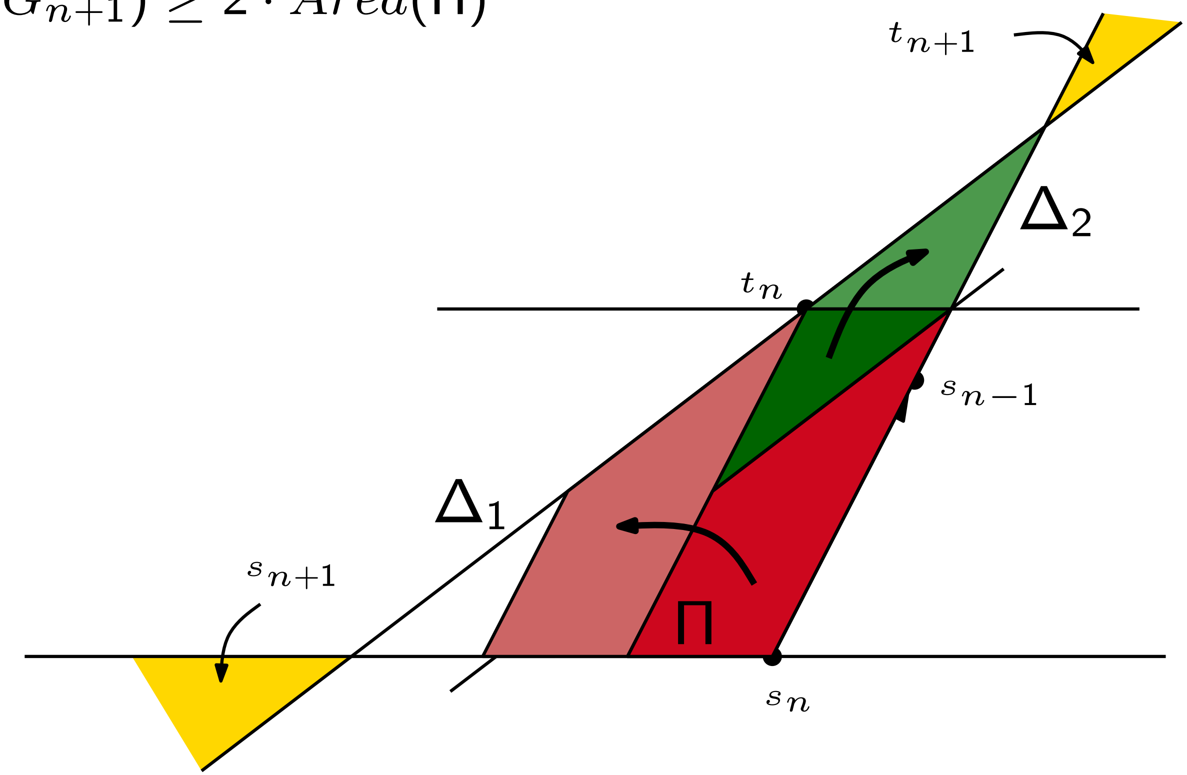
**Theorem [Bertolazzi et al. 94]**

There exists a $2n$-vertex series-parallel graph $G_n$ such that any upward planar drawing of $G_n$ respecting embedding requires area $\Omega(4^n)$.

Proof:

- We have that: $Area(\Pi) > 2 \cdot Area(G_n)$
- $Area(G_{n+1}) \geq 2 \cdot Area(\Pi)$



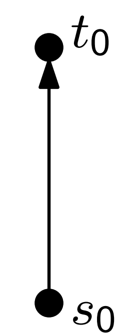
# Lower Bound for the Area

## Theorem [Bertolazzi et al. 94]

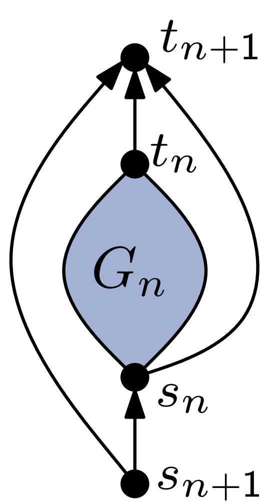
There exists a $2n$-vertex series-parallel graph $G_n$ such that any upward planar drawing of $G_n$ respecting embedding requires area $\Omega(4^n)$.

Proof:

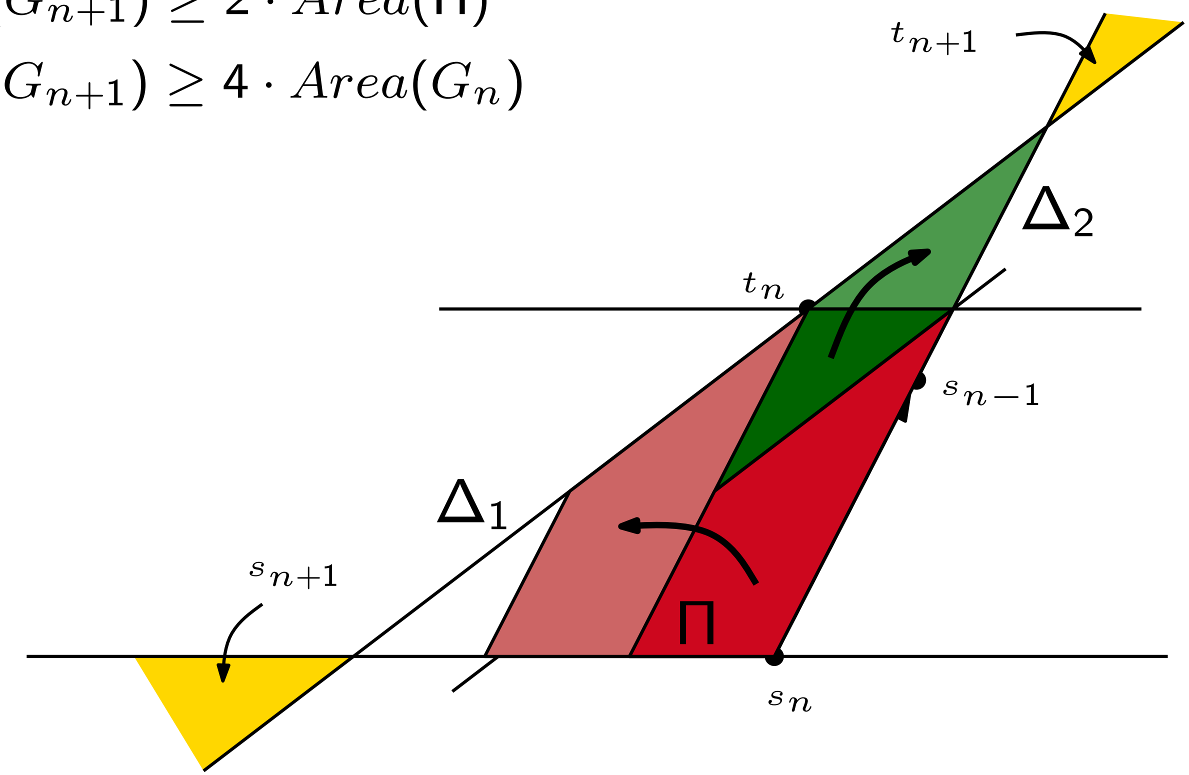
- We have that: $Area(\Pi) > 2 \cdot Area(G_n)$
- $Area(G_{n+1}) \geq 2 \cdot Area(\Pi)$
- $Area(G_{n+1}) \geq 4 \cdot Area(G_n)$



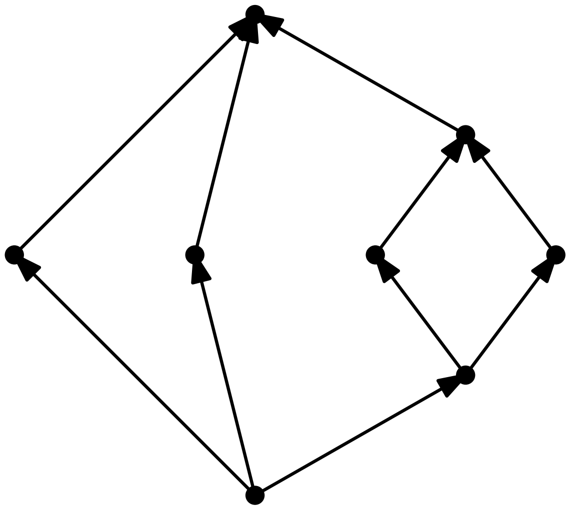

$G_0$

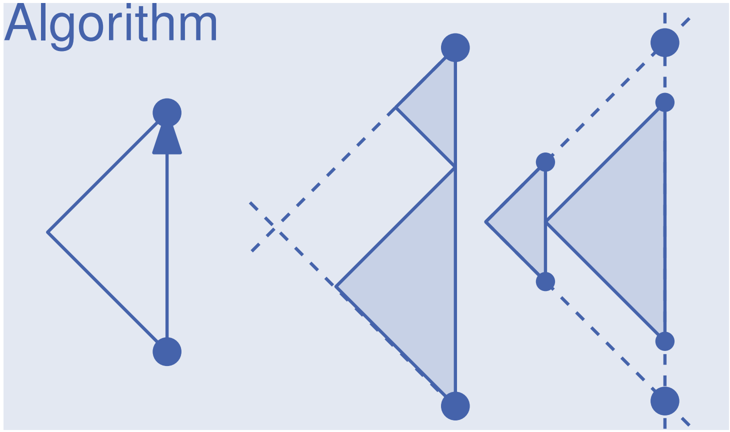


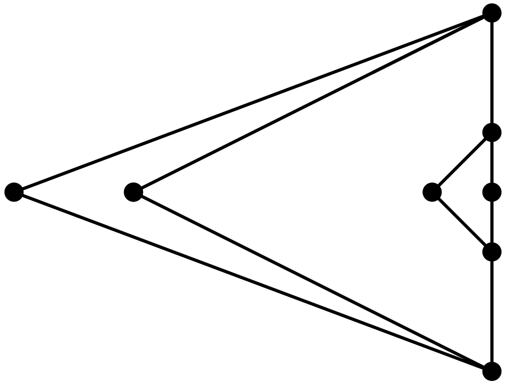

$G_{n+1}$



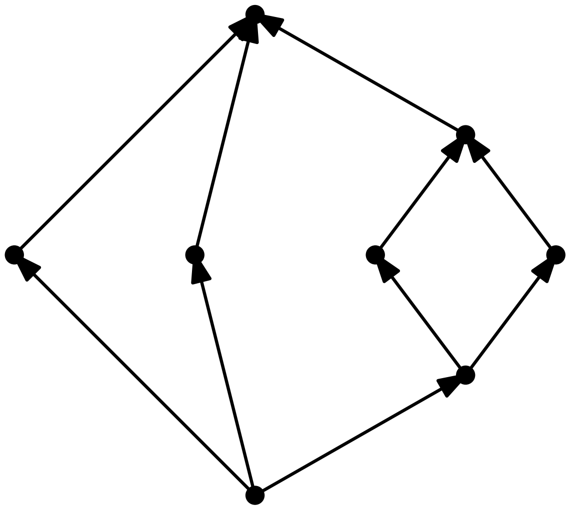
# Property of the Algorithm

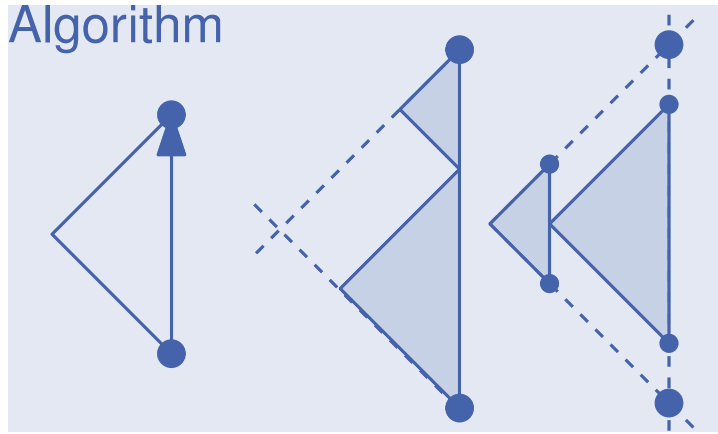


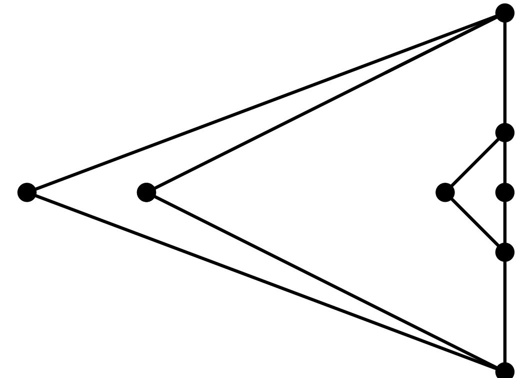


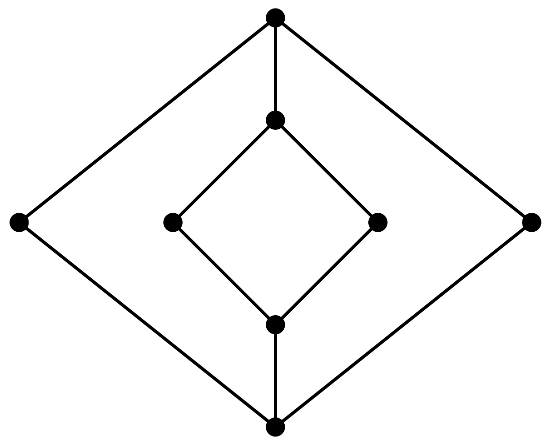



# Property of the Algorithm



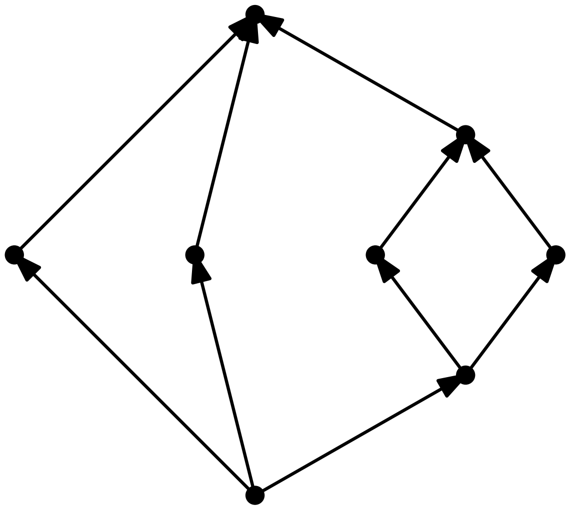


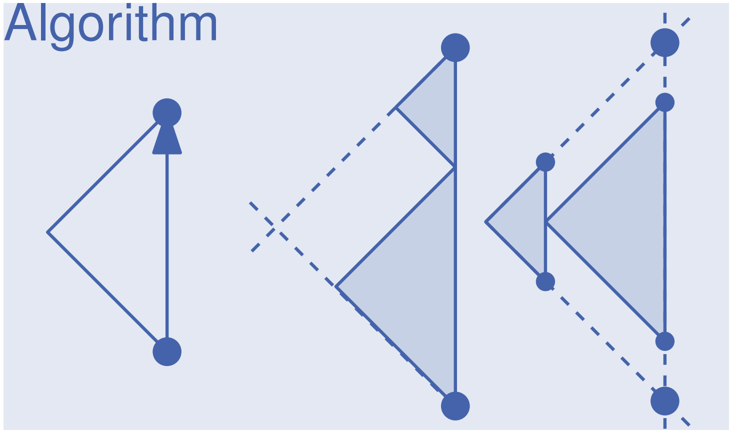



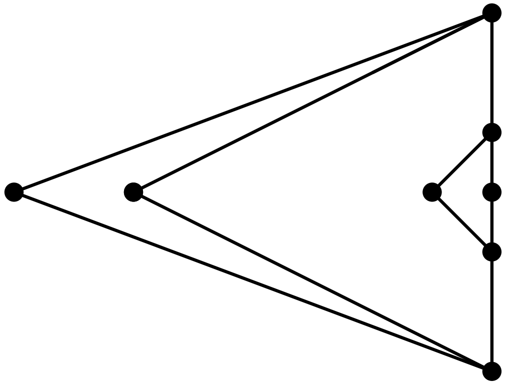


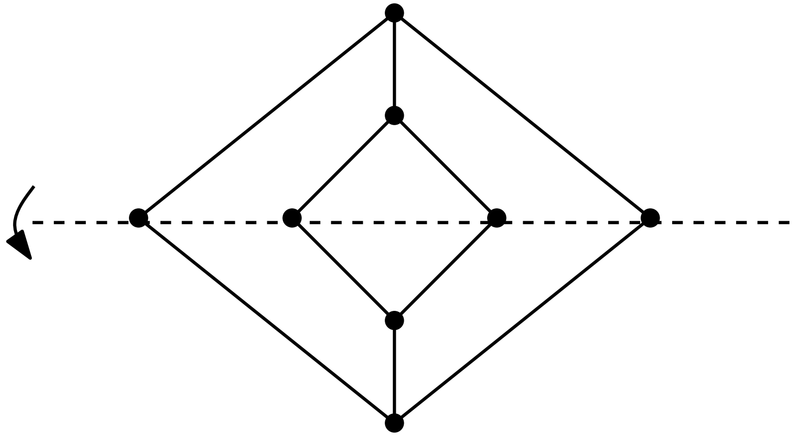
**nicer???**

# Property of the Algorithm



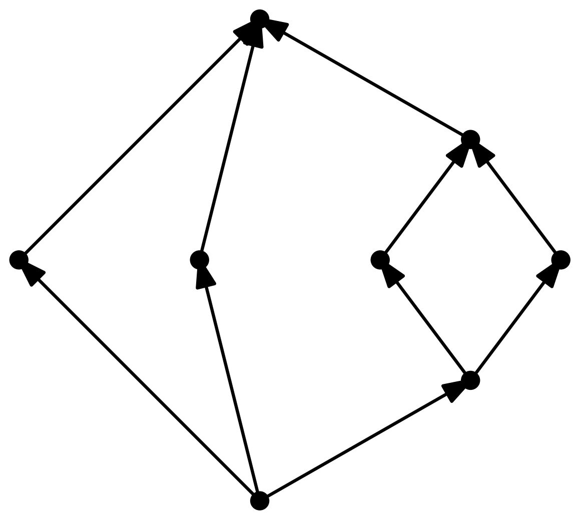


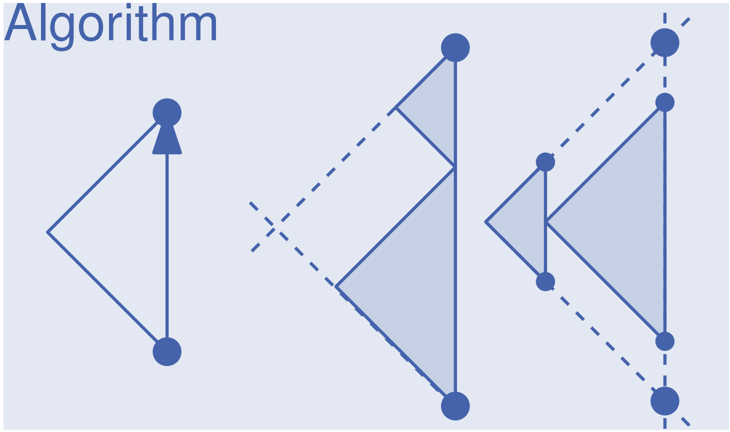



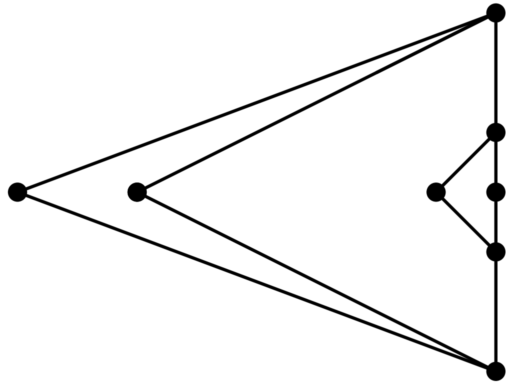


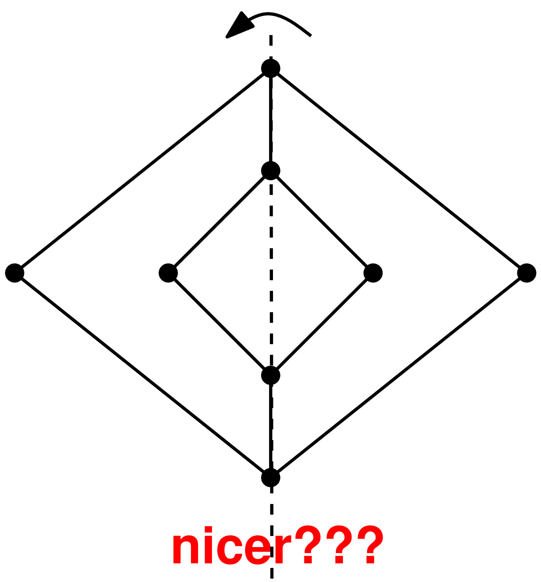
**nicer???**

# Property of the Algorithm

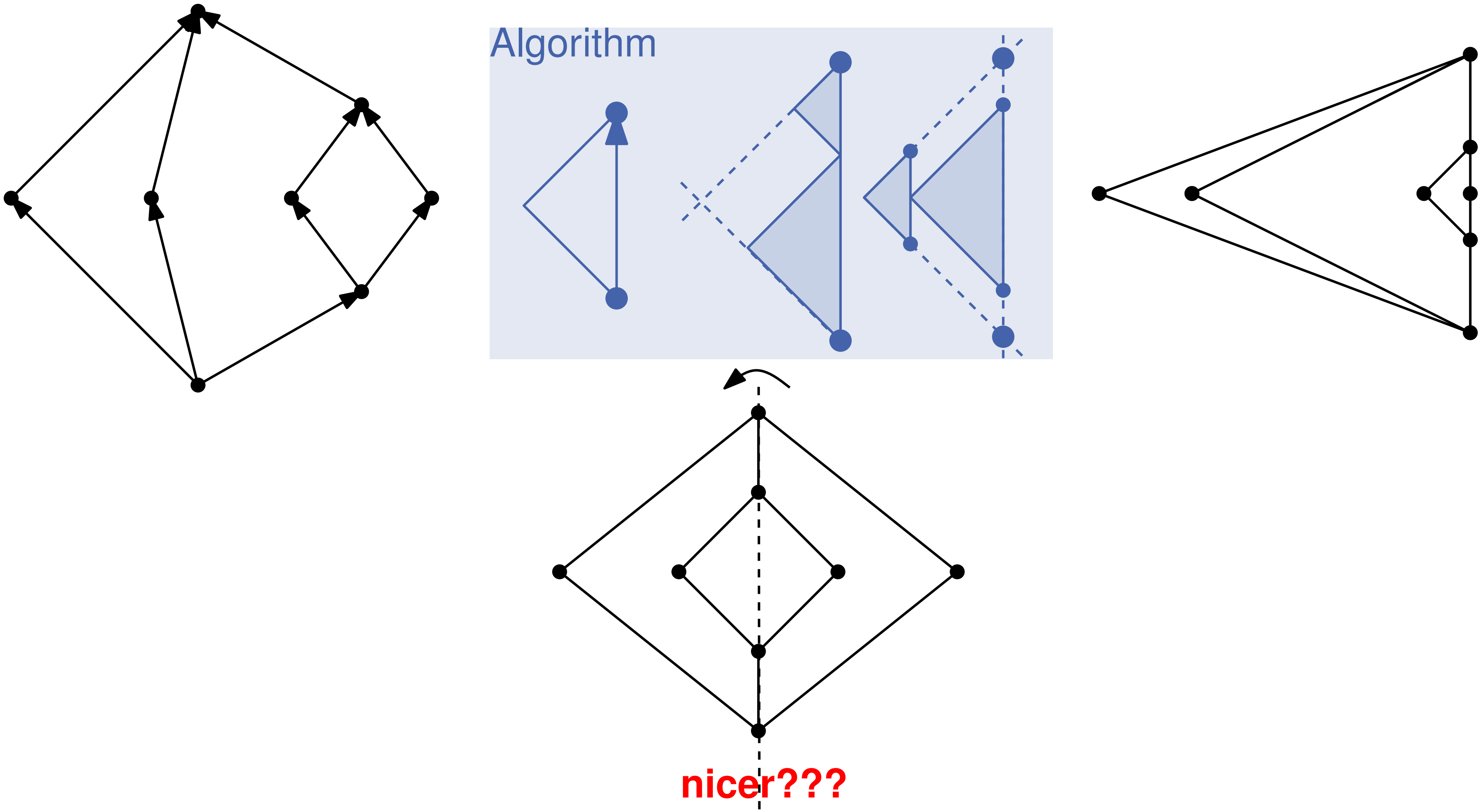






# Property of the Algorithm




- Algorithm by Hong, Eades and Lee (2000) creates symmetrical drawings of series-parallel graphs.