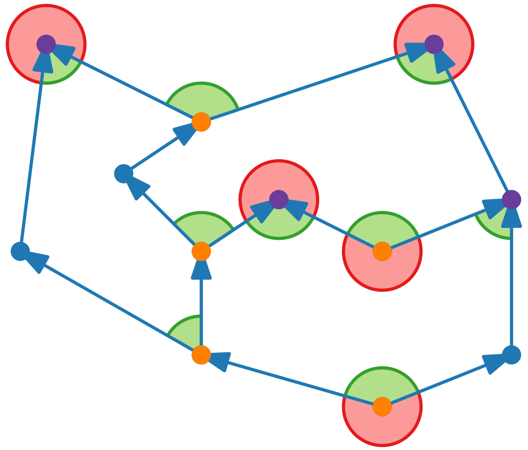
Visualisation of graphs

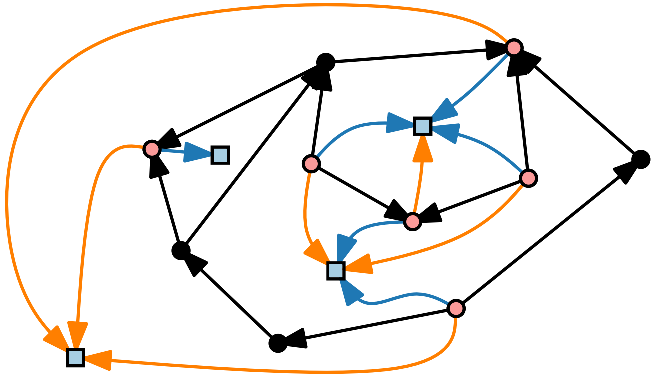
Upward planar drawings

Flow methods

Antonios Symvonis · Chrysanthi Raftopoulou

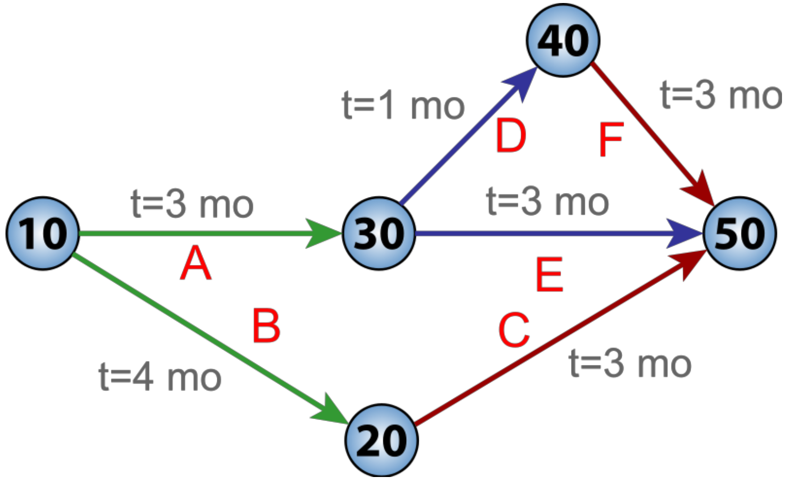
Fall semester 2022



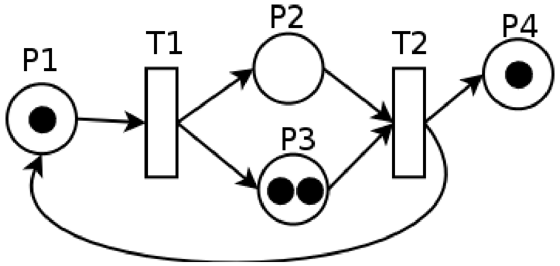


Upward planar drawings – motivation

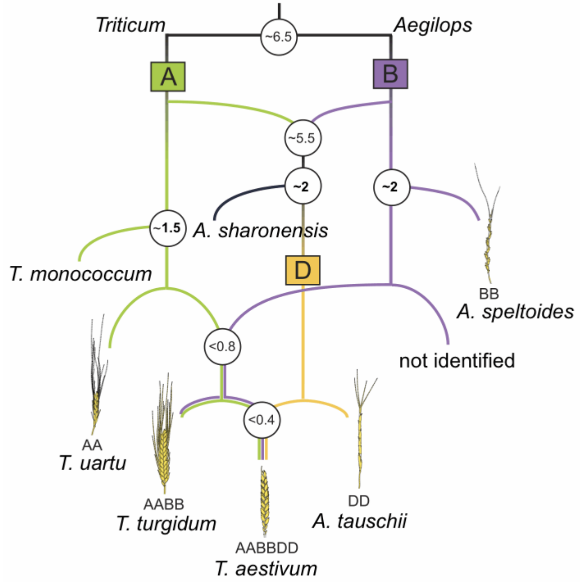
- What may the direction of edges in a digraph represent?
  - Time
  - Flow
  - Hierarchie
  - ...



PERT diagram



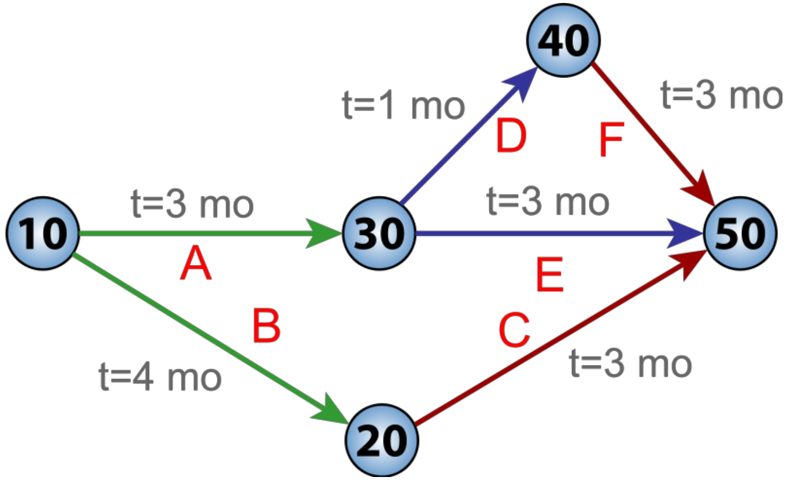
Petri net



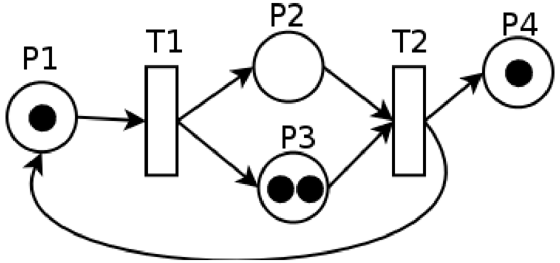
Phylogenetic network

Upward planar drawings – motivation

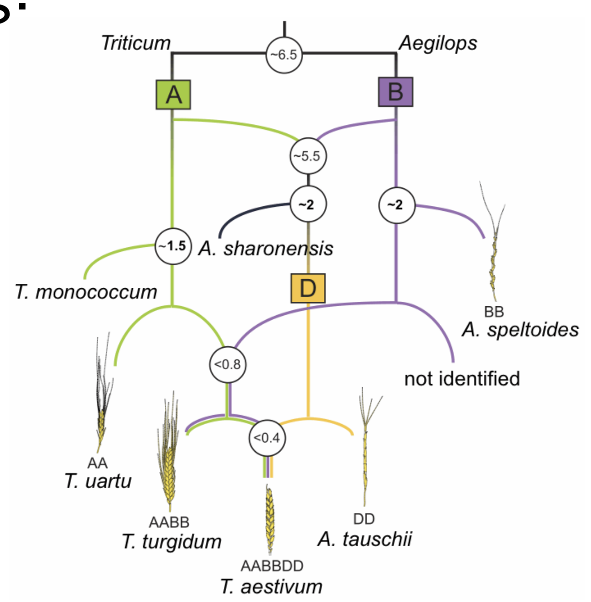
- What may the direction of edges in a digraph represent?
  - Time
  - Flow
  - Hierarchie
  - . . .
- Would be nice to have general direction preserved in drawing.



PERT diagram



Petri net

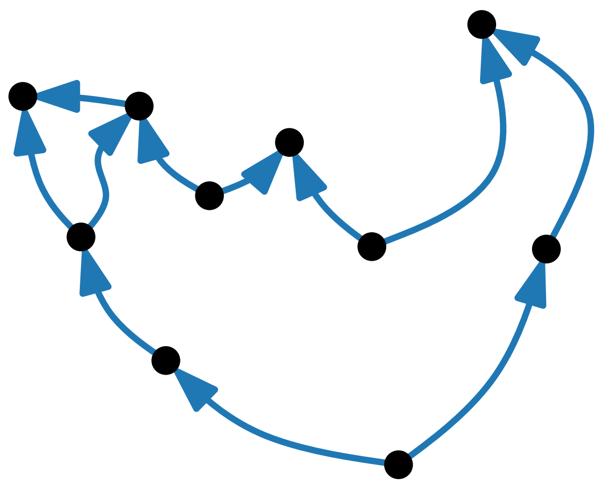


Phylogenetic network

Upward planar drawings – definition

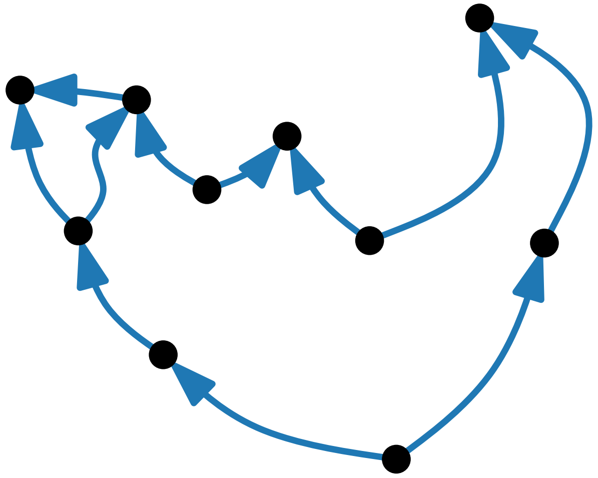
Definition.
A directed graph $G = (V, E)$ is **upward planar** when it admits a drawing Γ (vertices = points, edges = simple curves) that is

- planar and
- where each edge is drawn as an upward, y-monotone curve.



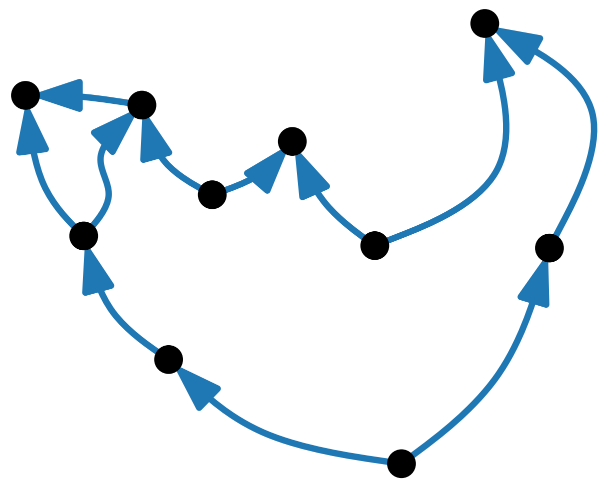
Upward planarity – necessary conditions

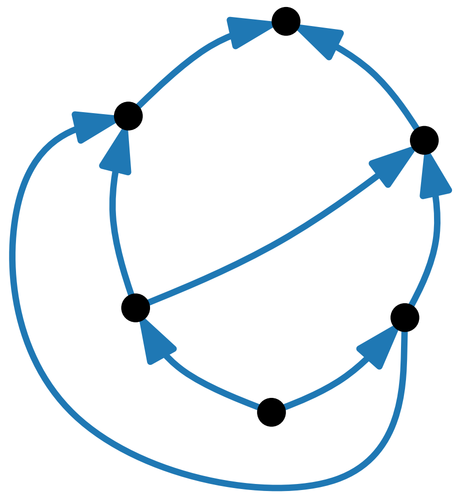
- For a digraph G to be upward planar, it has to be:
  - planar



Upward planarity – necessary conditions

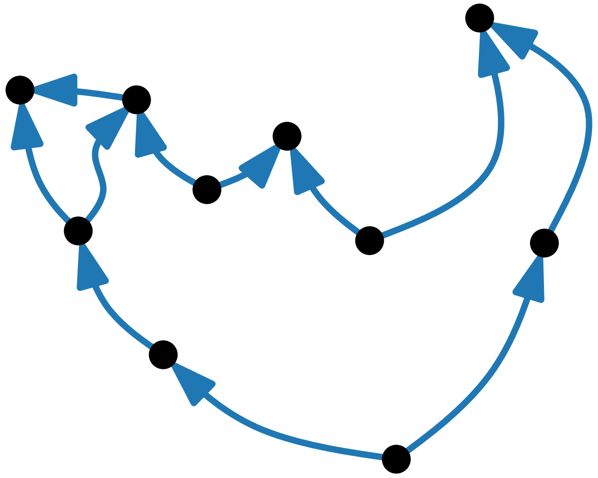
- For a digraph G to be upward planar, it has to be:
  - planar
  - acyclic

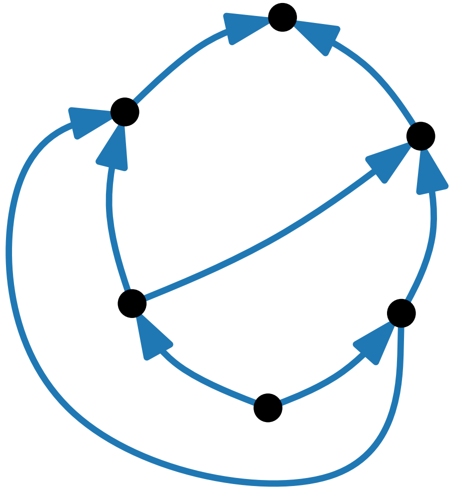


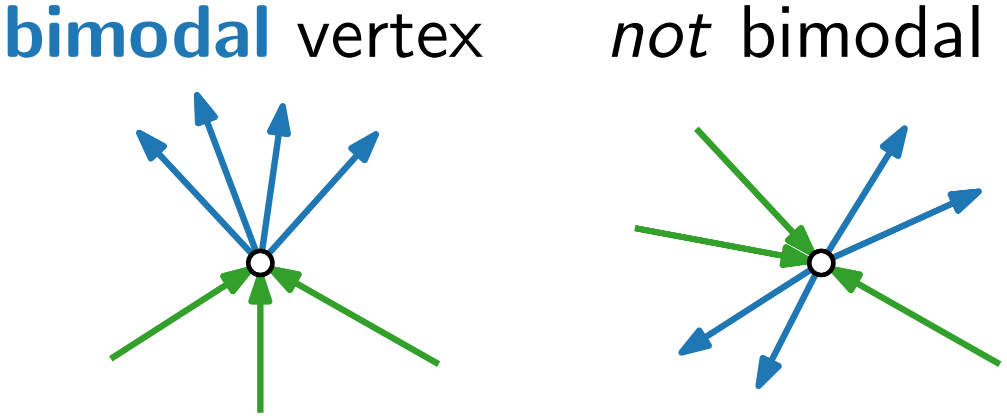


Upward planarity – necessary conditions

- For a digraph G to be upward planar, it has to be:
  - planar
  - acyclic

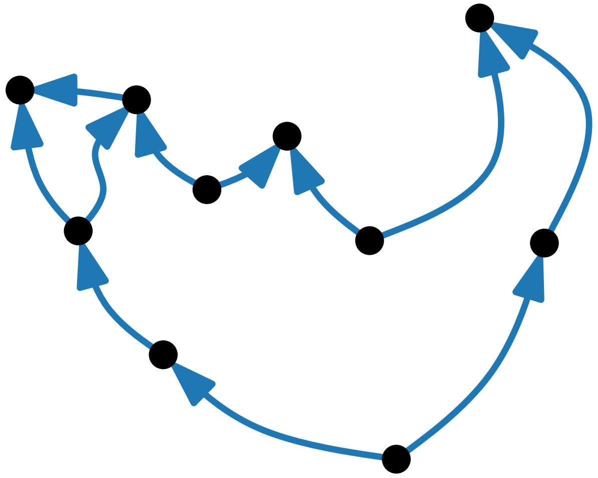


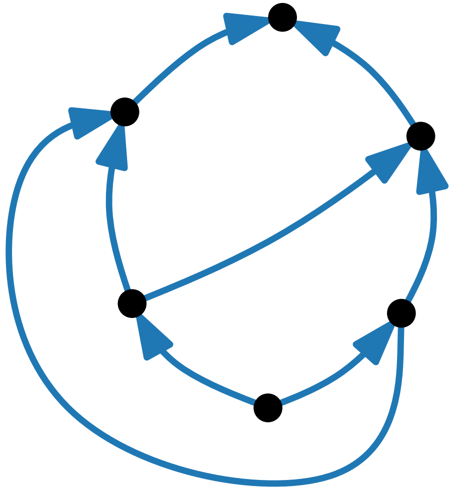




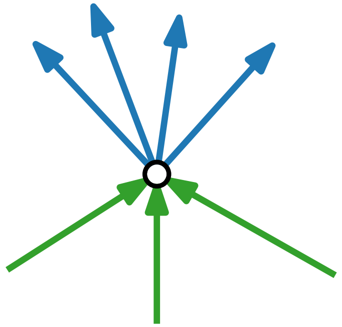
Upward planarity – necessary conditions

- For a digraph G to be upward planar, it has to be:
  - planar
  - acyclic
  - bimodal

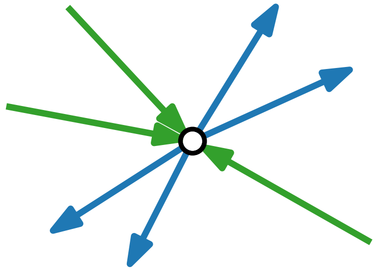




bimodal vertex

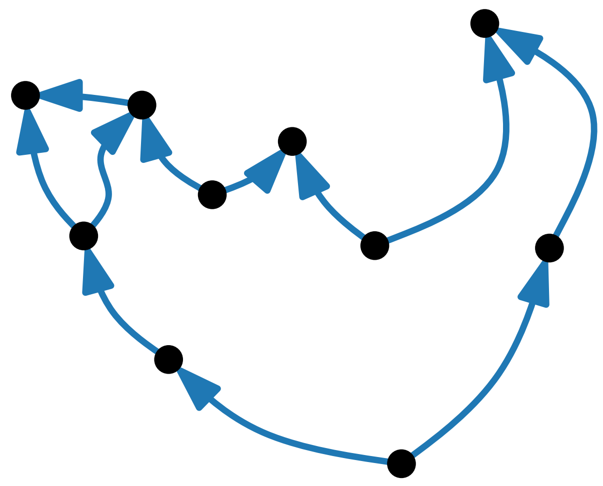


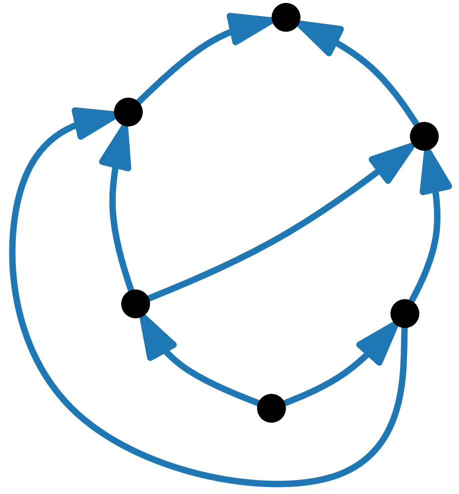
not bimodal



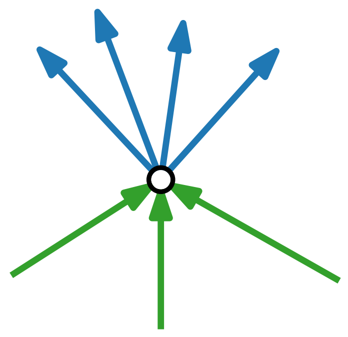
Upward planarity – necessary conditions

- For a digraph G to be upward planar, it has to be:
  - planar
  - acyclic
  - bimodal
- . . . but these conditions are *not sufficient*.

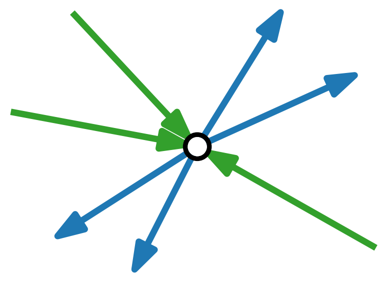




bimodal vertex



not bimodal



Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:

1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:

1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

$\underbrace{}$ no crossings

Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:

1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

no crossings

acyclic digraph with
a single source s and single sink t

Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:
1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Additionally:
Embedded such that s and t are on the outerface f_0.

no crossings

acyclic digraph with
a single source s and single sink t

Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:

1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Additionally:
Embedded such that s and t are on the outerface f_0.
or:
Edge (s, t) exists.

no crossings

acyclic digraph with
a single source s and single sink t

Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:
1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Proof.

(2) $\Rightarrow$ **(1)** By definition.

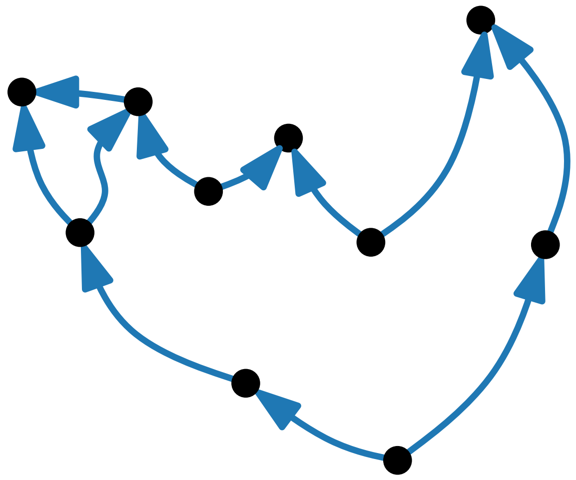
Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:

1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Proof.

(2) $\Rightarrow$ **(1)** By definition. **(1)** $\Leftrightarrow$ **(3)** Example:



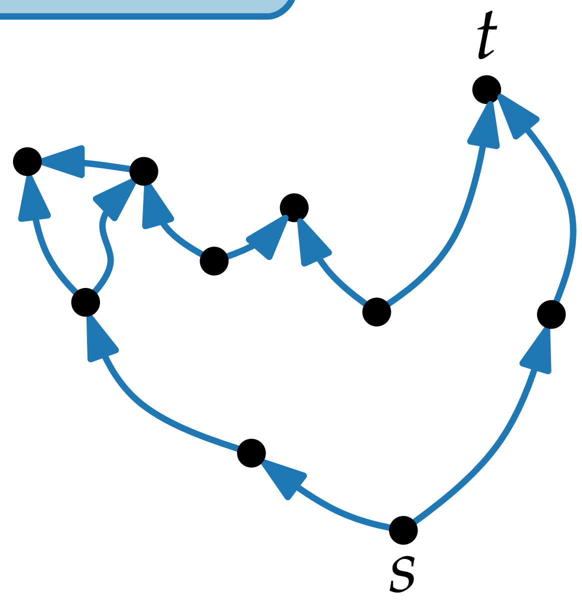
Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:

1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Proof.

(2) $\Rightarrow$ **(1)** By definition. **(1)** $\Leftrightarrow$ **(3)** Example:



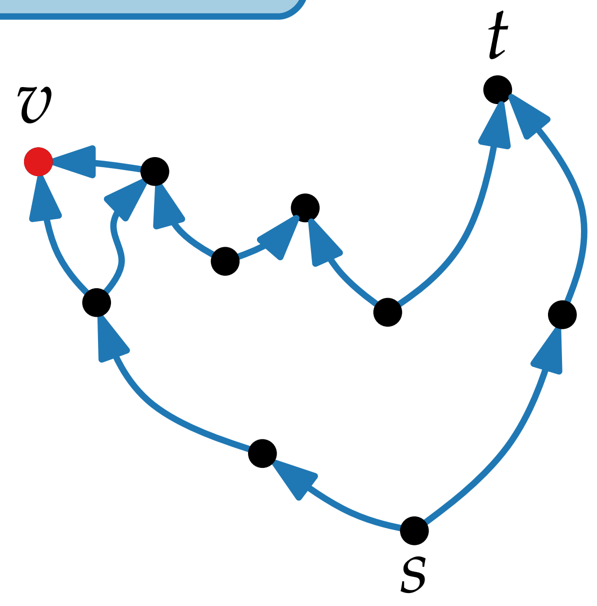
Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:

1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Proof.

(2) $\Rightarrow$ **(1)** By definition. **(1)** $\Leftrightarrow$ **(3)** Example:



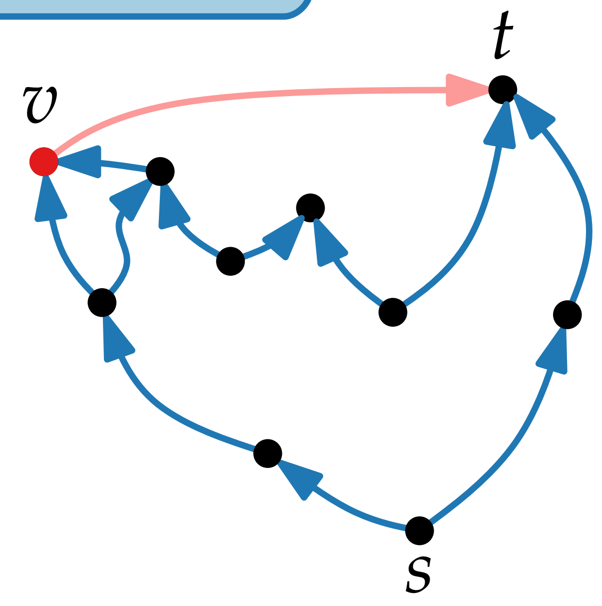
Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:

1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Proof.

(2) $\Rightarrow$ **(1)** By definition. **(1)** $\Leftrightarrow$ **(3)** Example:



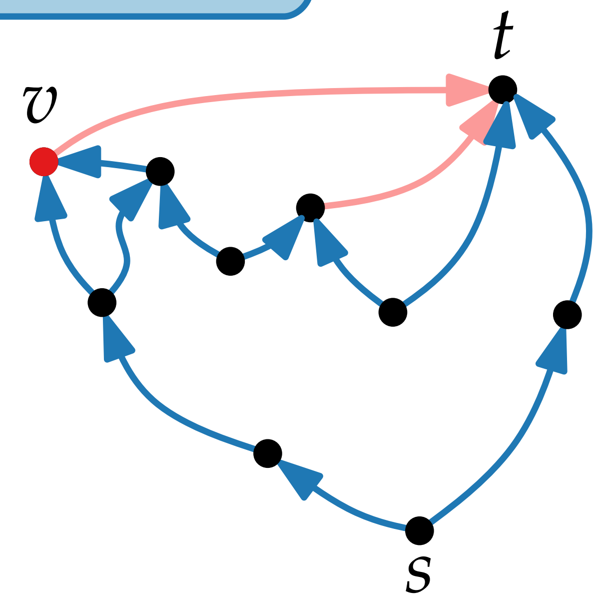
Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:

1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Proof.

(2) $\Rightarrow$ **(1)** By definition. **(1)** $\Leftrightarrow$ **(3)** Example:

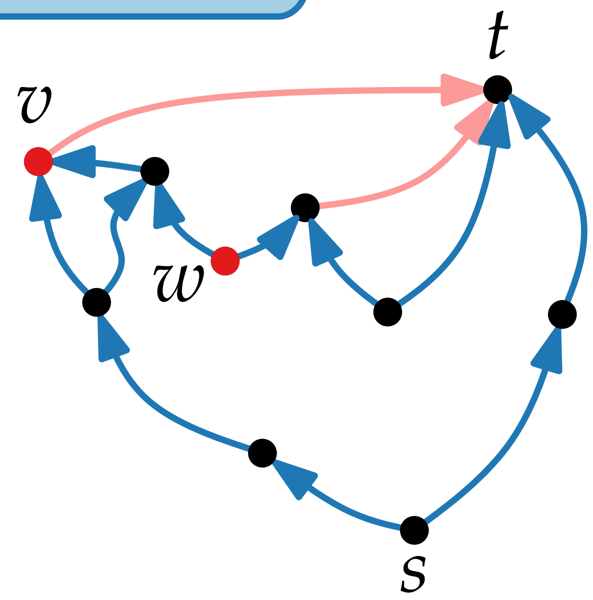


Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:
1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Proof.

(2) $\Rightarrow$ **(1)** By definition. **(1)** $\Leftrightarrow$ **(3)** Example:



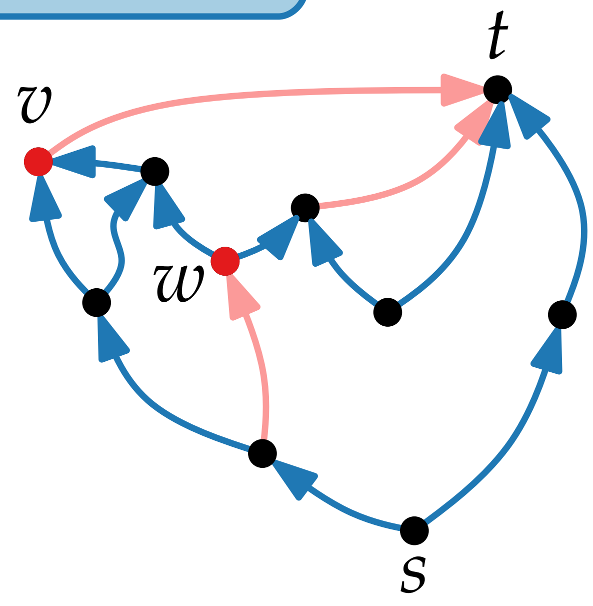
Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:

1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Proof.

(2) $\Rightarrow$ **(1)** By definition. **(1)** $\Leftrightarrow$ **(3)** Example:

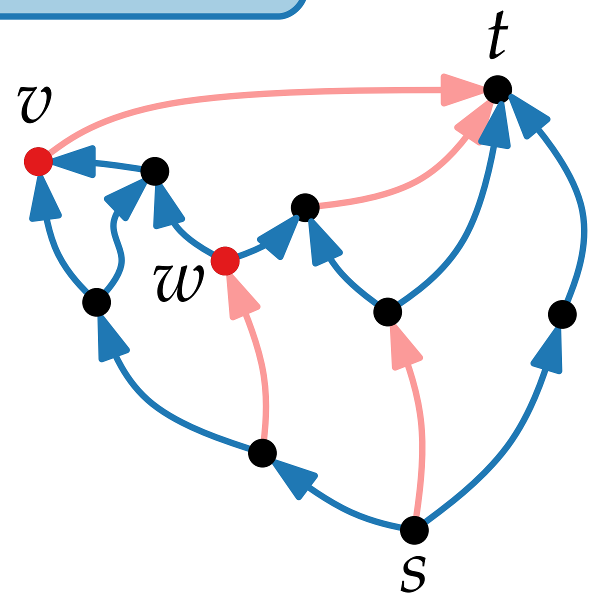


Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:
1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Proof.

(2) $\Rightarrow$ **(1)** By definition. **(1)** $\Leftrightarrow$ **(3)** Example:



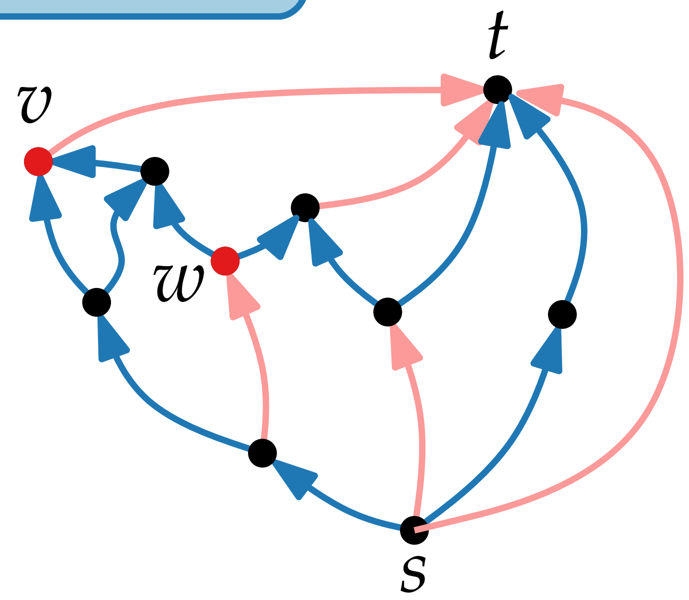
Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:

1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Proof.

(2) $\Rightarrow$ **(1)** By definition. **(1)** $\Leftrightarrow$ **(3)** Example:

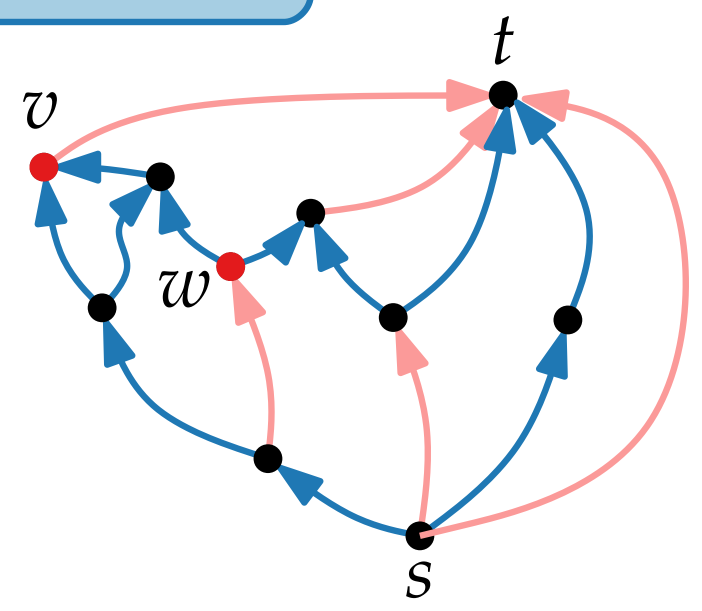


Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:
1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Proof.

(2) $\Rightarrow$ **(1)** By definition. **(1)** $\Leftrightarrow$ **(3)** Example:
(3) $\Rightarrow$ **(2)** Triangulate & construct drawing:



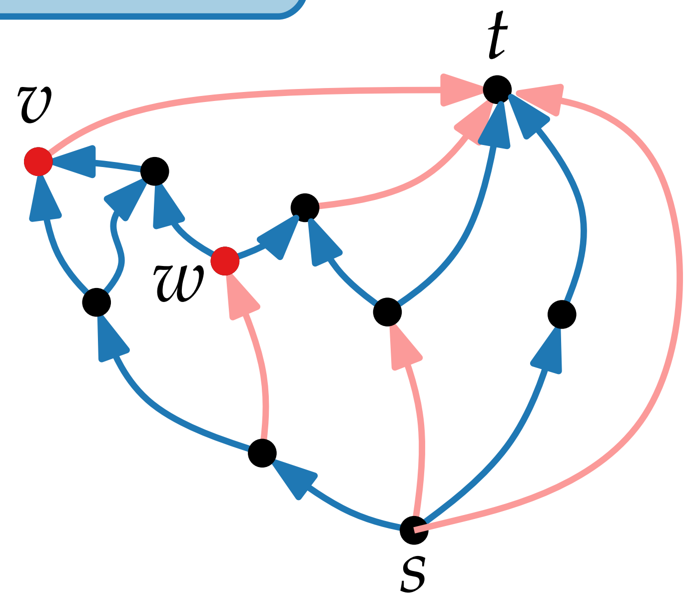
Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:
1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Proof.

(2) ⇒ (1) By definition. **(1) ⇔ (3)** Example:
(3) ⇒ (2) Triangulate & construct drawing:

Claim.
Can draw in prespecified triangle.



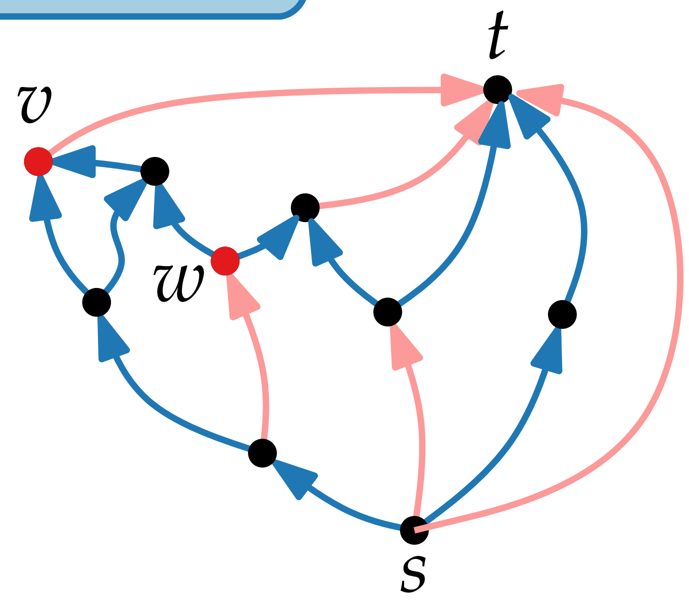
Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:

1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Proof.

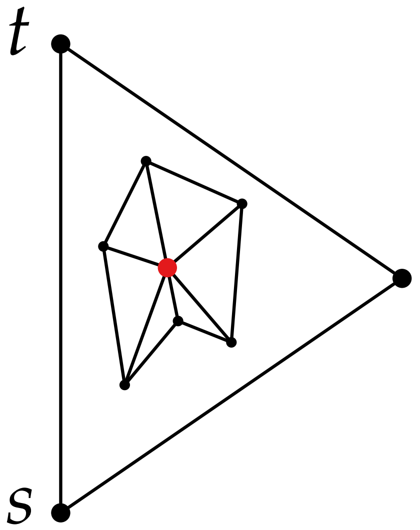
(2) $\Rightarrow$ **(1)** By definition. **(1)** $\Leftrightarrow$ **(3)** Example:



(3) $\Rightarrow$ **(2)** Triangulate & construct drawing:

Claim.
Can draw in prespecified triangle.

Case 1:

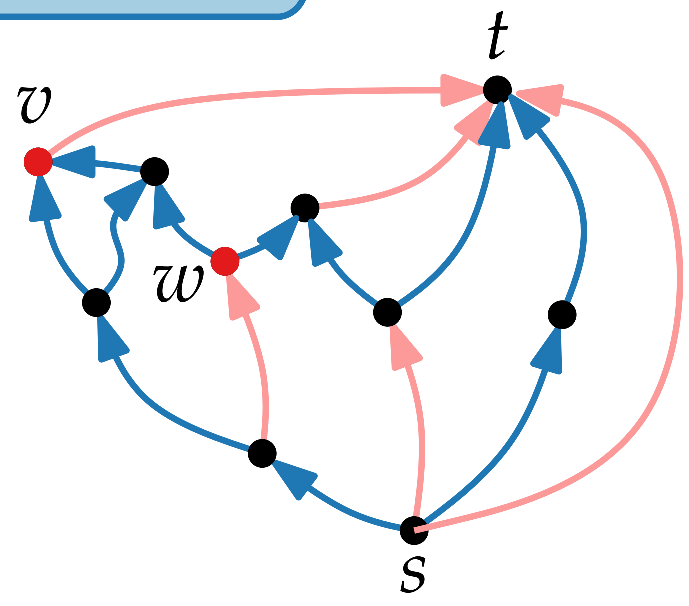


Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:
1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Proof.

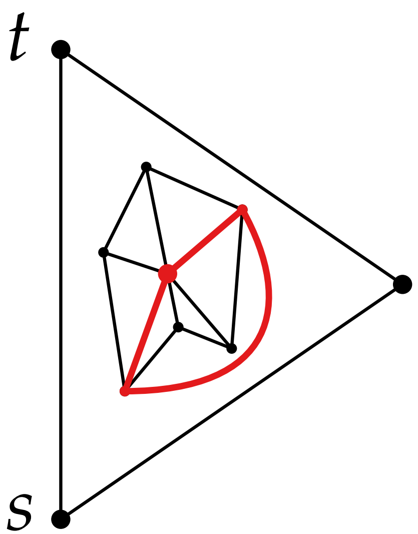
(2) ⇒ (1) By definition. **(1) ⇔ (3)** Example:



(3) ⇒ (2) Triangulate & construct drawing:

Claim.
Can draw in prespecified triangle.

Case 1:

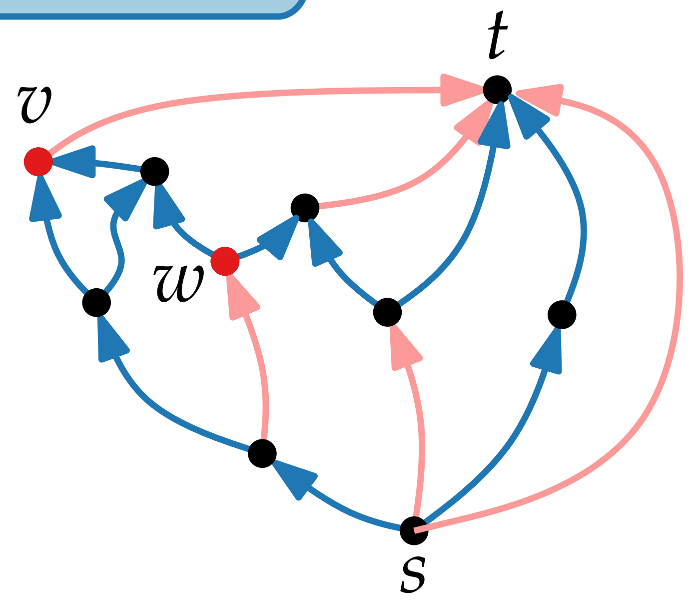


Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:
1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Proof.

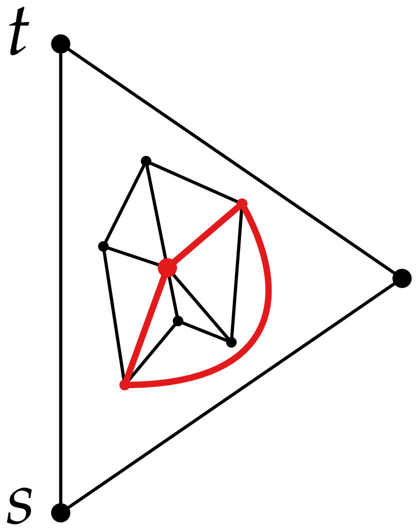
(2) $\Rightarrow$ **(1)** By definition. **(1)** $\Leftrightarrow$ **(3)** Example:



(3) $\Rightarrow$ **(2)** Triangulate & construct drawing:

Claim.
Can draw in prespecified triangle. Apply induction.

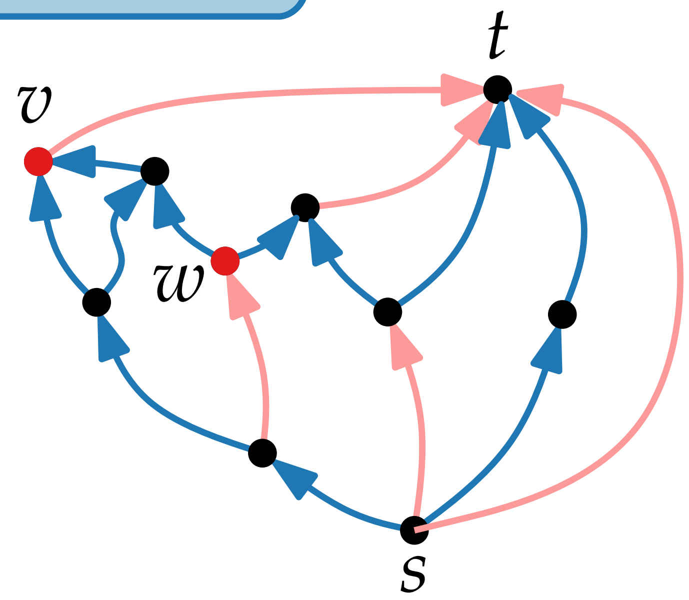
Case 1:



Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:
1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

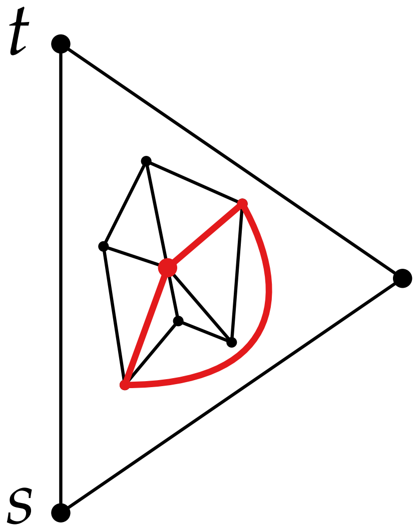
Proof.



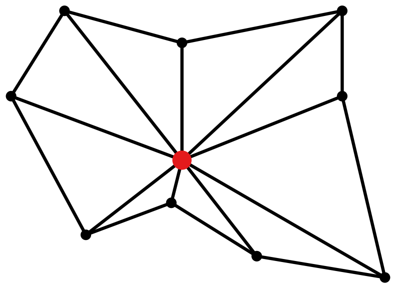
(2) ⇒ (1) By definition. **(1) ⇔ (3)** Example:
(3) ⇒ (2) Triangulate & construct drawing:

Claim.
Can draw in prespecified triangle. Apply induction.

Case 1:



Case 2:

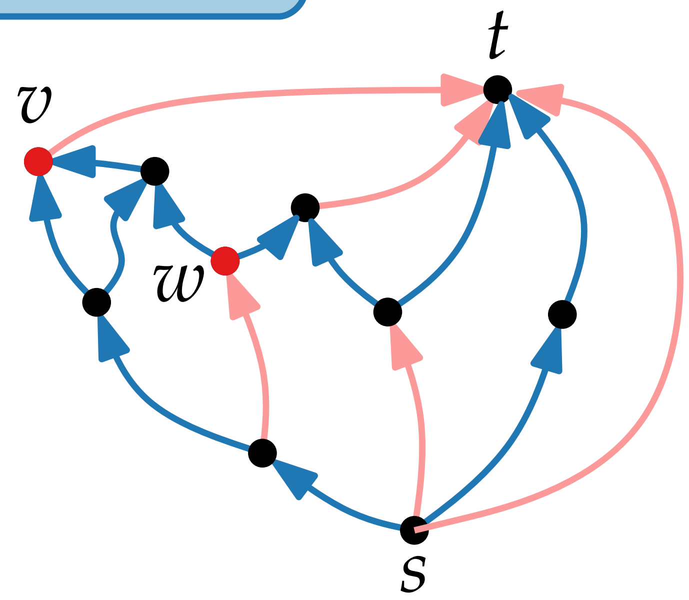


Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:

1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

Proof.

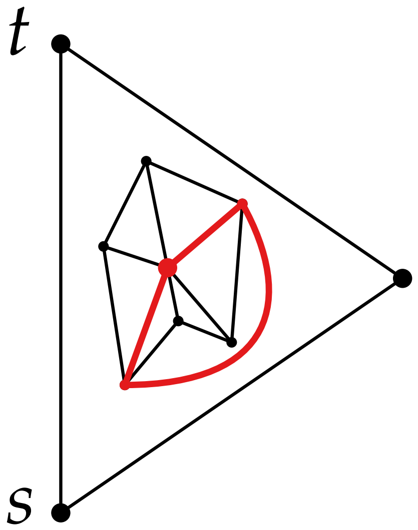


(2) ⇒ (1) By definition. **(1) ⇔ (3)** Example:

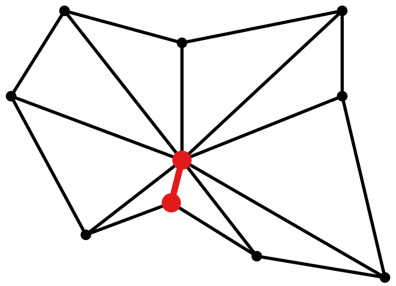
(3) ⇒ (2) Triangulate & construct drawing:

Claim.
Can draw in prespecified triangle. Apply induction.

Case 1:



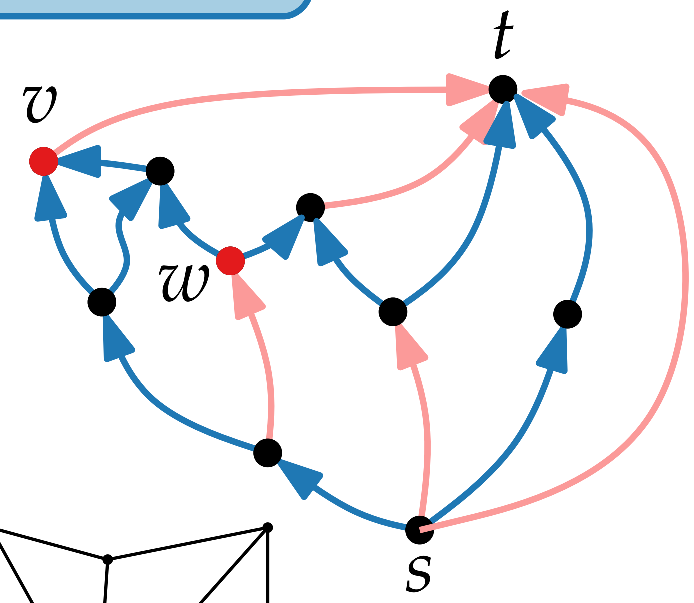
Case 2:



Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:
1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

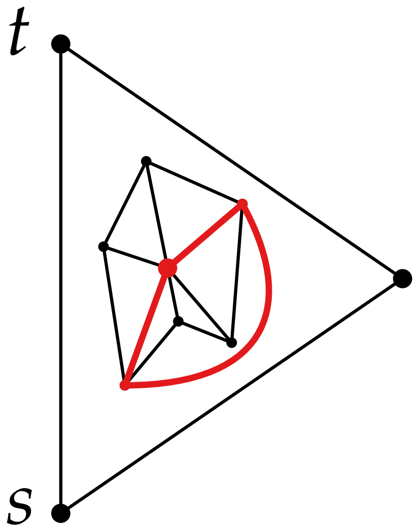
Proof.



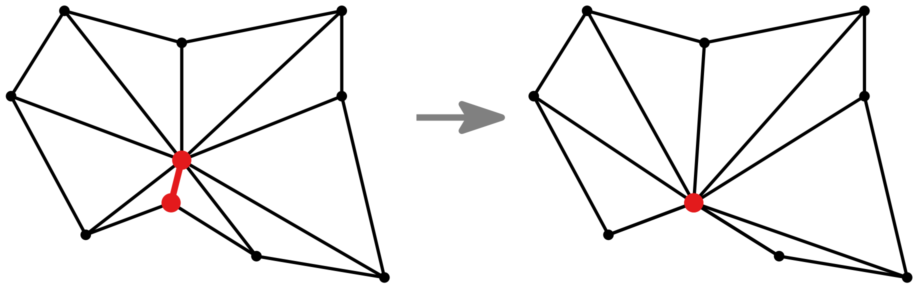
(2) ⇒ (1) By definition. **(1) ⇔ (3)** Example:
(3) ⇒ (2) Triangulate & construct drawing:

Claim.
Can draw in prespecified triangle. Apply induction.

Case 1:



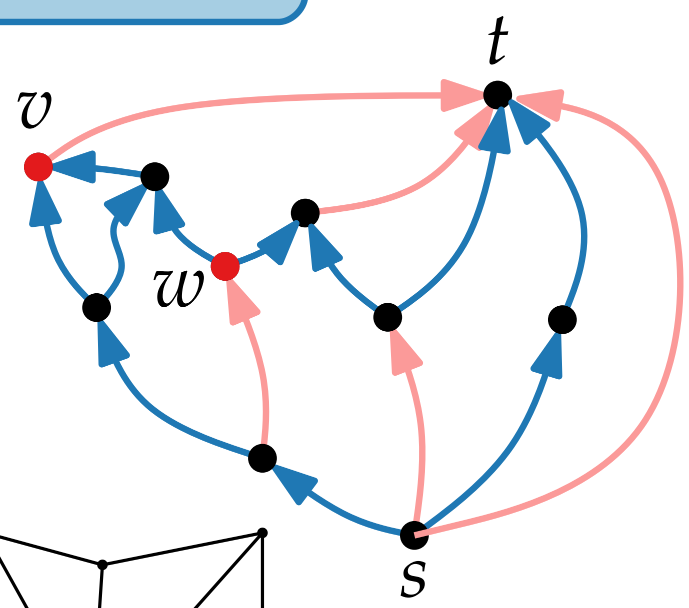
Case 2:



Upward planarity – characterisation

Theorem 1. [Kelly 1987, Di Battista, Tamassia, 1988, see GD Ch. 6]
For a digraph G the following statements are equivalent:
1. G is upward planar.
2. G admits an upward planar straight-line drawing.
3. G is the spanning subgraph of a planar st-digraph.

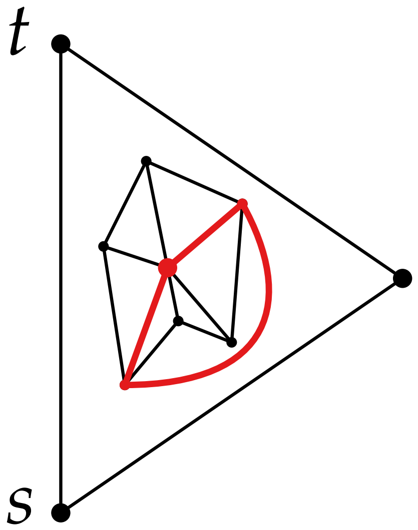
Proof.



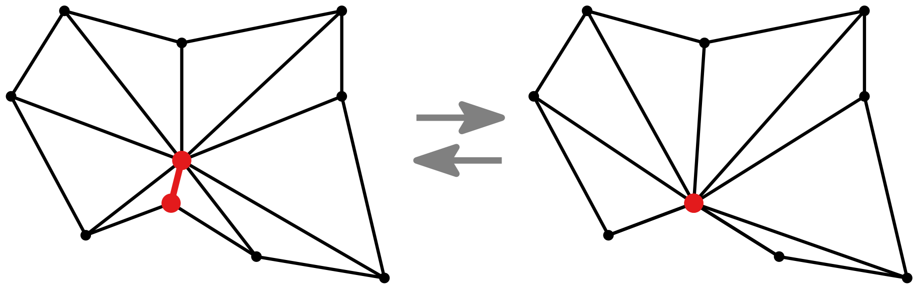
(2) ⇒ (1) By definition. **(1) ⇔ (3)** Example:
(3) ⇒ (2) Triangulate & construct drawing:

Claim.
Can draw in prespecified triangle.
Apply induction.

Case 1:



Case 2:



Upward planarity – complexity

Theorem. [Garg, Tamassia, 1995]
For a *planar acyclic* digraph it is in general NP-hard to decide whether it is upward planar.

Upward planarity – complexity

Theorem. [Garg, Tamassia, 1995]
For a *planar acyclic* digraph it is in general NP-hard to decide whether it is upward planar.

Theorem 2. [Bertolazzi et al., 1994]
For a *combinatorially embedded* planar digraph it can be tested in $\mathcal{O}(n^2)$ time whether it is upward planar.

Upward planarity – complexity

Theorem. [Garg, Tamassia, 1995]
For a *planar acyclic* digraph it is in general NP-hard to decide whether it is upward planar.

Theorem 2. [Bertolazzi et al., 1994]
For a *combinatorially embedded* planar digraph it can be tested in $\mathcal{O}(n^2)$ time whether it is upward planar.

Corollary.
For a *triconnected* planar digraph it can be tested in $\mathcal{O}(n^2)$ time whether it is upward planar.

Upward planarity – complexity

Theorem. [Garg, Tamassia, 1995]
For a *planar acyclic* digraph it is in general NP-hard to decide whether it is upward planar.

Theorem 2. [Bertolazzi et al., 1994]
For a *combinatorially embedded* planar digraph it can be tested in $\mathcal{O}(n^2)$ time whether it is upward planar.

Corollary.
For a *triconnected* planar digraph it can be tested in $\mathcal{O}(n^2)$ time whether it is upward planar.

Theorem. [Hutton, Libow, 1996]
For a *single-source* acyclic digraph it can be tested in $\mathcal{O}(n)$ time whether it is upward planar.

The problem

Fixed embedding upward planarity testing.
Let $G = (V, E)$ be a plane digraph with the embedding given by the set of faces F and the outer face f_0.
Test whether G is upward planar (wrt to F, f_0).

The problem

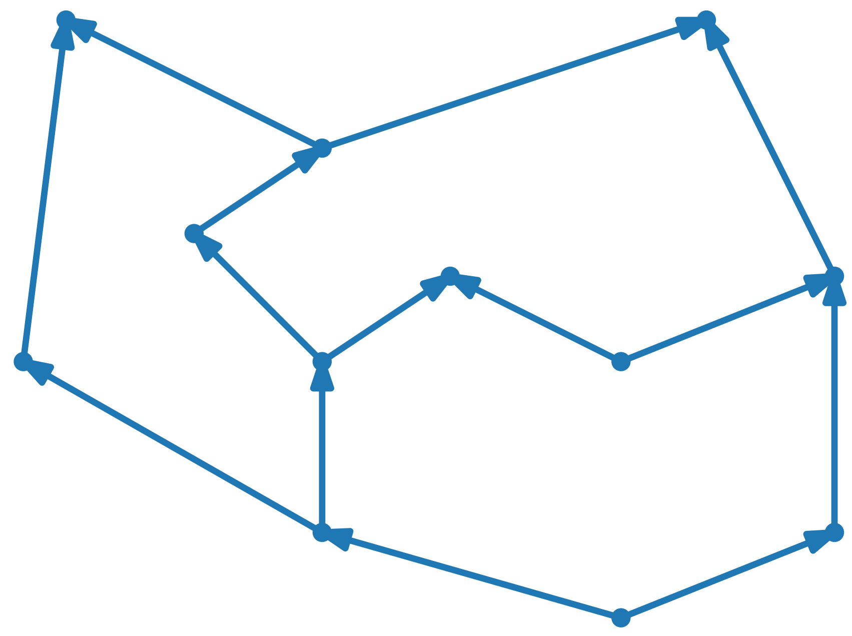
Fixed embedding upward planarity testing.
Let $G = (V, E)$ be a plane digraph with the embedding given by the set of faces F and the outer face f_0.
Test whether G is upward planar (wrt to F, f_0).

Idea.

- Find property that any upward planar drawing of G satisfies.
- Formalise property.
- Find algorithm to test property.

Angles, local sources & sinks

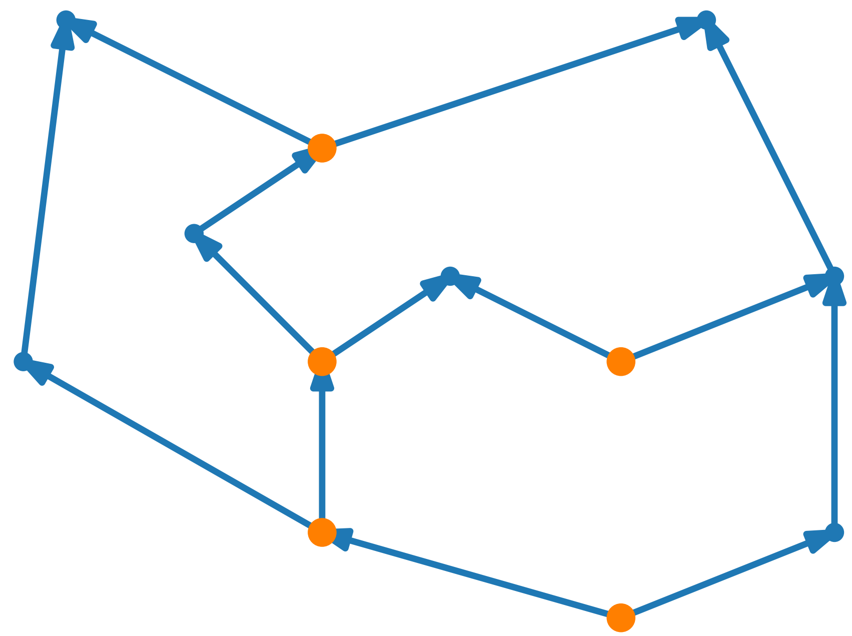
Definitions.



Angles, local sources & sinks

Definitions.

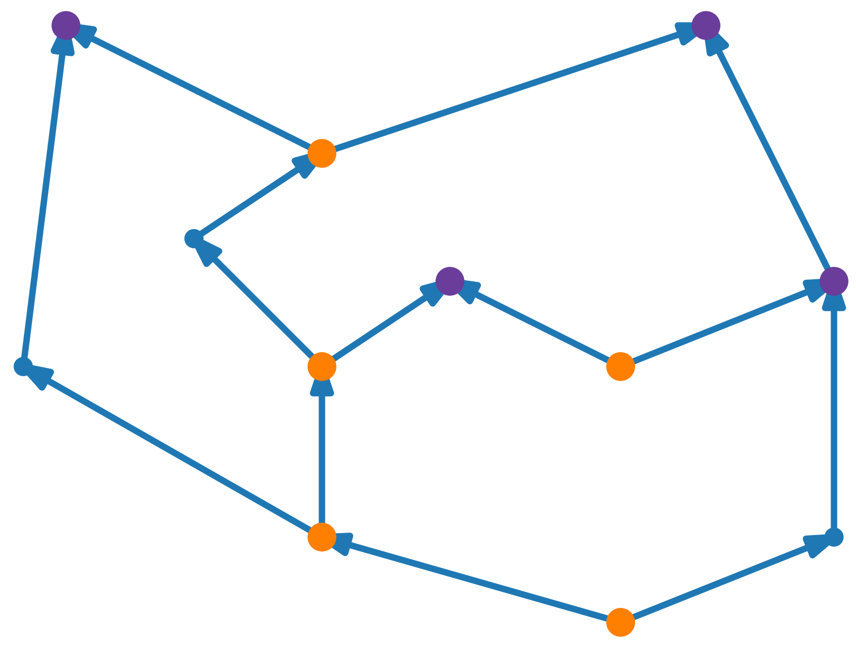
- A vertex v is a **local source** wrt to a face f if v has two outgoing edges on ∂f.



Angles, local sources & sinks

Definitions.

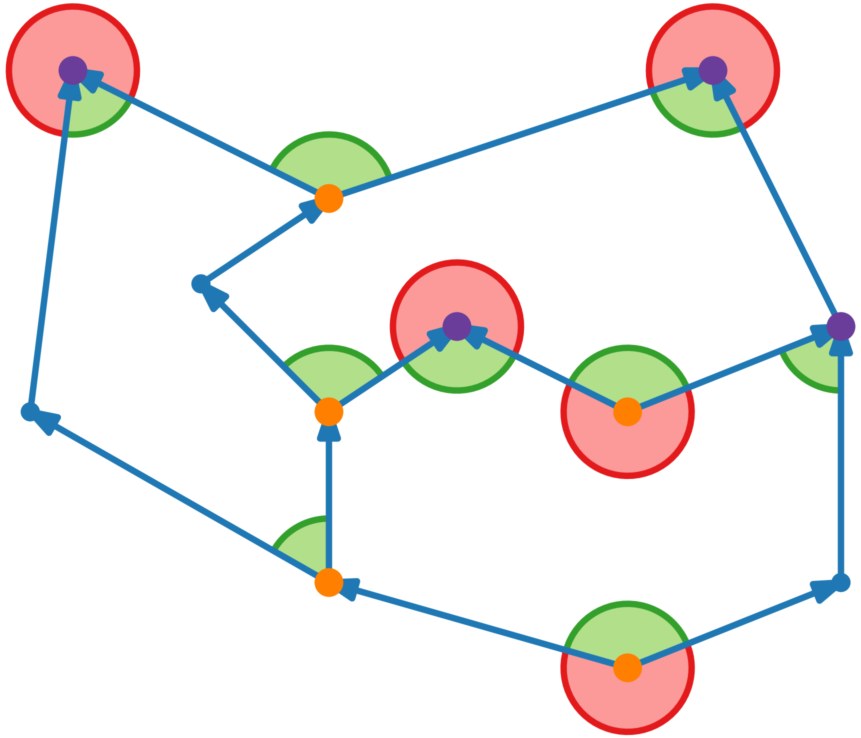
- A vertex v is a **local source** wrt to a face f if v has two outgoing edges on ∂f.
- A vertex v is a **local sink** wrt to a face f if v has two incoming edges on ∂f.



Angles, local sources & sinks

Definitions.

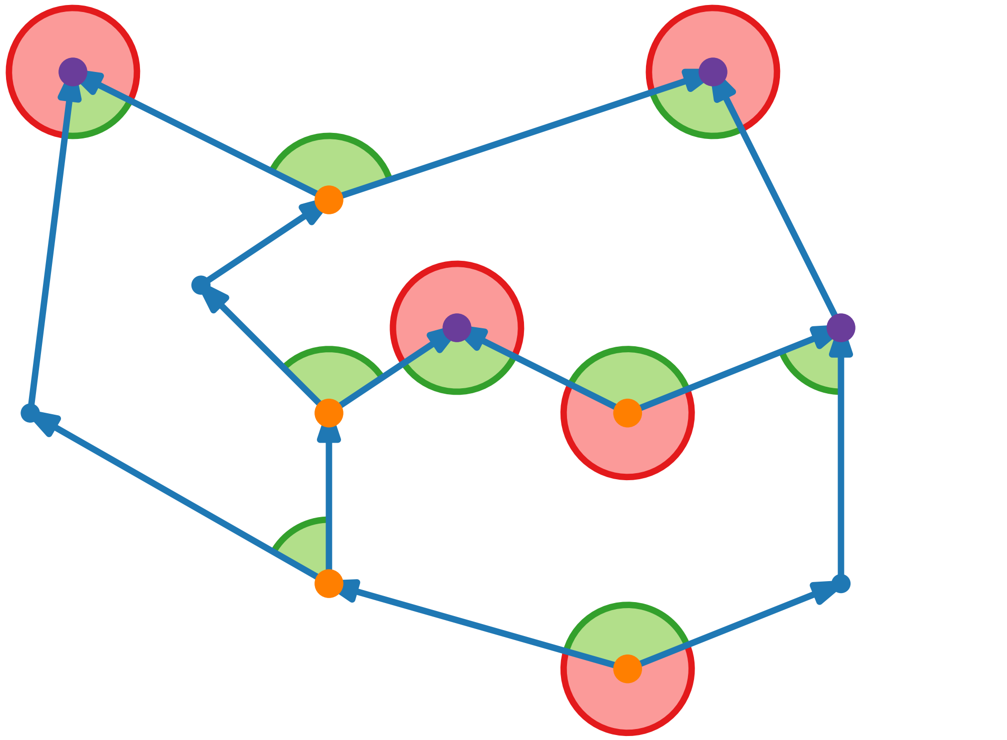
- A vertex v is a **local source** wrt to a face f if v has two outgoing edges on ∂f.
- A vertex v is a **local sink** wrt to a face f if v has two incoming edges on ∂f.
- An angle α is **large** when $\alpha > \pi$ and **small** otherwise.
- $L(v) = \#$ large angles at v
- $L(f) = \#$ large angles in f



Angles, local sources & sinks

Definitions.

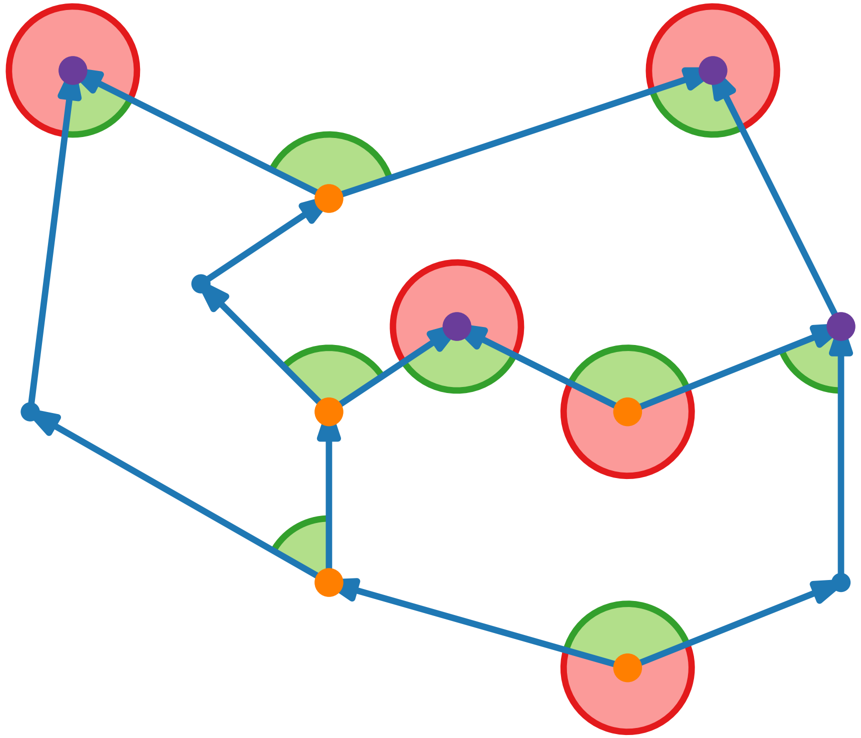
- A vertex v is a **local source** wrt to a face f if v has two outgoing edges on ∂f.
- A vertex v is a **local sink** wrt to a face f if v has two incoming edges on ∂f.
- An angle α is **large** when $\alpha > \pi$ and **small** otherwise.
- $L(v) = \#$ large angles at v
- $L(f) = \#$ large angles in f
- $S(v)$ & $S(f)$ for $\#$ small angles



Angles, local sources & sinks

Definitions.

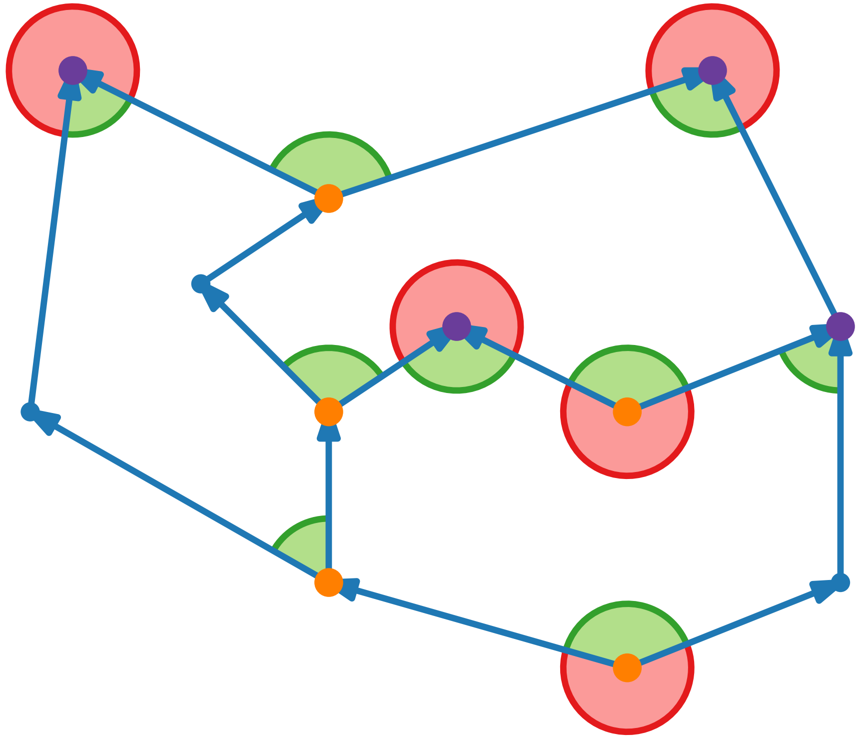
- A vertex v is a **local source** wrt to a face f if v has two outgoing edges on ∂f.
- A vertex v is a **local sink** wrt to a face f if v has two incoming edges on ∂f.
- An angle α is **large** when $\alpha > \pi$ and **small** otherwise.
- $L(v) = \#$ large angles at v
- $L(f) = \#$ large angles in f
- $S(v)$ & $S(f)$ for $\#$ small angles
- $A(f) = \#$ local sources wrt to f
  $= \#$ local sinks wrt to f



Angles, local sources & sinks

Definitions.

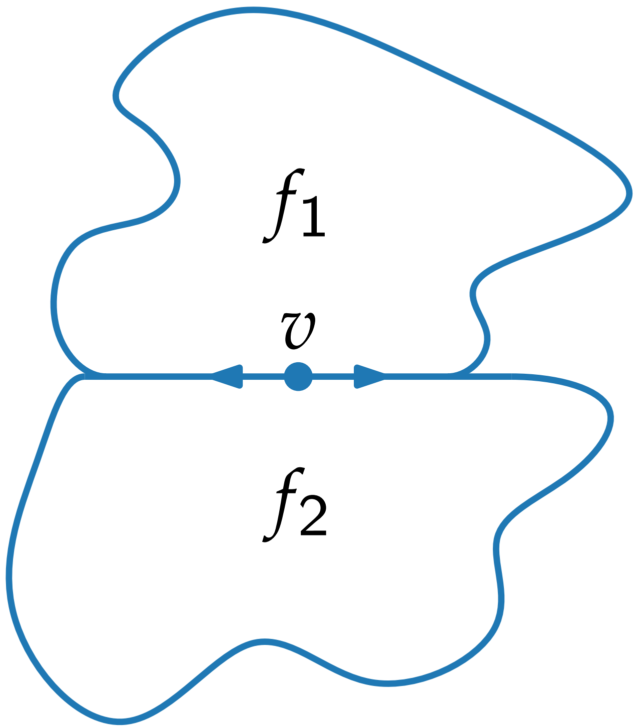
- A vertex v is a **local source** wrt to a face f if v has two outgoing edges on ∂f.
- A vertex v is a **local sink** wrt to a face f if v has two incoming edges on ∂f.
- An angle α is **large** when $\alpha > \pi$ and **small** otherwise.
- $L(v) = \#$ large angles at v
- $L(f) = \#$ large angles in f
- $S(v)$ & $S(f)$ for $\#$ small angles
- $A(f) = \#$ local sources wrt to f
  $= \#$ local sinks wrt to f



Lemma 1.
$L(f) + S(f) = 2A(f)$

Assignment problem

- Vertex v is a global source for f_1 and f_2.
- Has v a **large** angle in f_1 or f_2?



Angle relations

Lemma 2.

$$L(f) - S(f) = \begin{cases} -2, & f \neq f_0 \\ +2, & f = f_0 \end{cases}$$

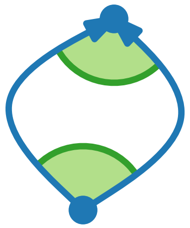
Angle relations

Lemma 2.

$$L(f) - S(f) = \begin{cases} -2, & f \neq f_0 \\ +2, & f = f_0 \end{cases}$$

Proof by induction.

- $L(f) = 0$

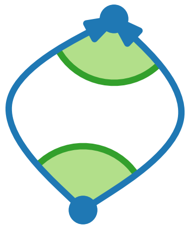


Angle relations

Lemma 2.

$$L(f) - S(f) = \begin{cases} -2, & f \neq f_0 \\ +2, & f = f_0 \end{cases}$$

Proof by induction.

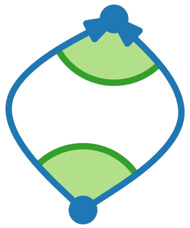
- $L(f) = 0$  $\Rightarrow S(f) = 2$

Angle relations

Lemma 2.

$$L(f) - S(f) = \begin{cases} -2, & f \neq f_0 \\ +2, & f = f_0 \end{cases}$$

Proof by induction.

- $L(f) = 0$  $\Rightarrow S(f) = 2$
- $L(f) \geq 1$

Split f with **edge** from a large angle at a “low” sink u to

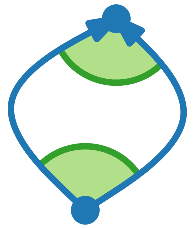
Angle relations

Lemma 2.

$$L(f) - S(f) = \begin{cases} -2, & f \neq f_0 \\ +2, & f = f_0 \end{cases}$$

Proof by induction.

- $L(f) = 0$

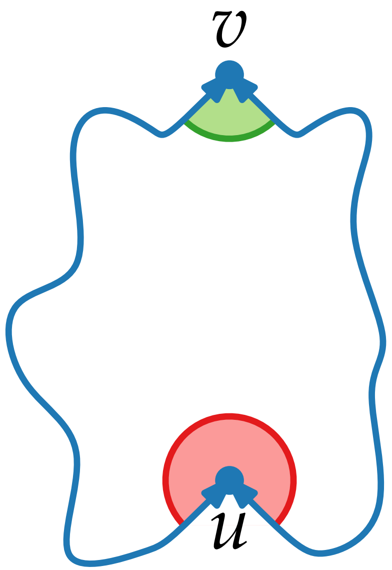


$\Rightarrow S(f) = 2$

- $L(f) \geq 1$

Split f with **edge** from a large angle at a “low” sink u to

- sink v with small angle:

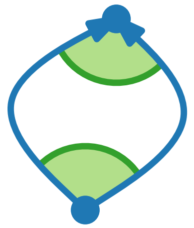


Angle relations

Lemma 2.

$$L(f) - S(f) = \begin{cases} -2, & f \neq f_0 \\ +2, & f = f_0 \end{cases}$$

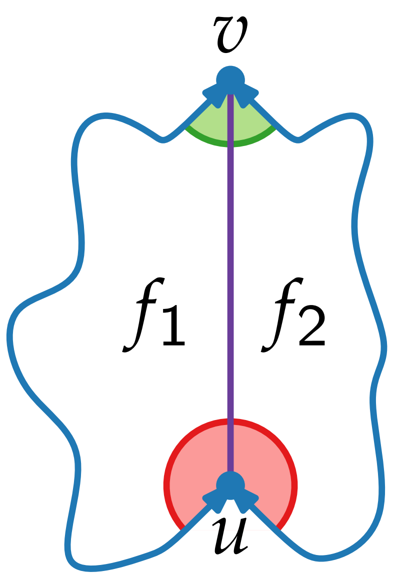
Proof by induction.

- $L(f) = 0$  $\Rightarrow S(f) = 2$

- $L(f) \geq 1$

Split f with **edge** from a large angle at a "low" sink u to

- sink v with small angle:

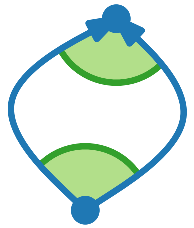


Angle relations

Lemma 2.

$$L(f) - S(f) = \begin{cases} -2, & f \neq f_0 \\ +2, & f = f_0 \end{cases}$$

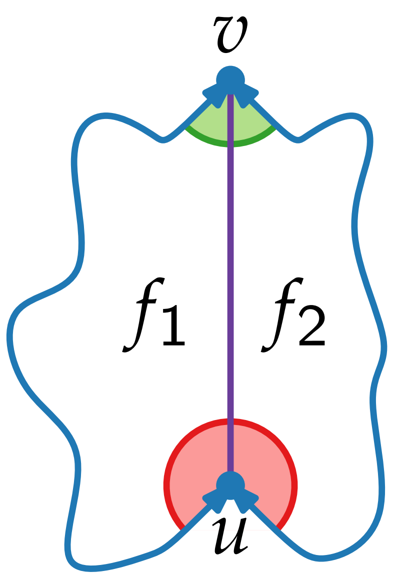
Proof by induction.

- $L(f) = 0$  $\Rightarrow S(f) = 2$

- $L(f) \geq 1$

Split f with **edge** from a large angle at a "low" sink u to

- sink v with small angle:



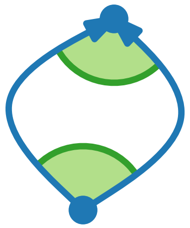
$$\begin{aligned} L(f) - S(f) &= L(f_1) + L(f_2) + 1 \\ &\quad - (S(f_1) + S(f_2) - 1) \\ &= -2 \end{aligned}$$

Angle relations

Lemma 2.

$$L(f) - S(f) = \begin{cases} -2, & f \neq f_0 \\ +2, & f = f_0 \end{cases}$$

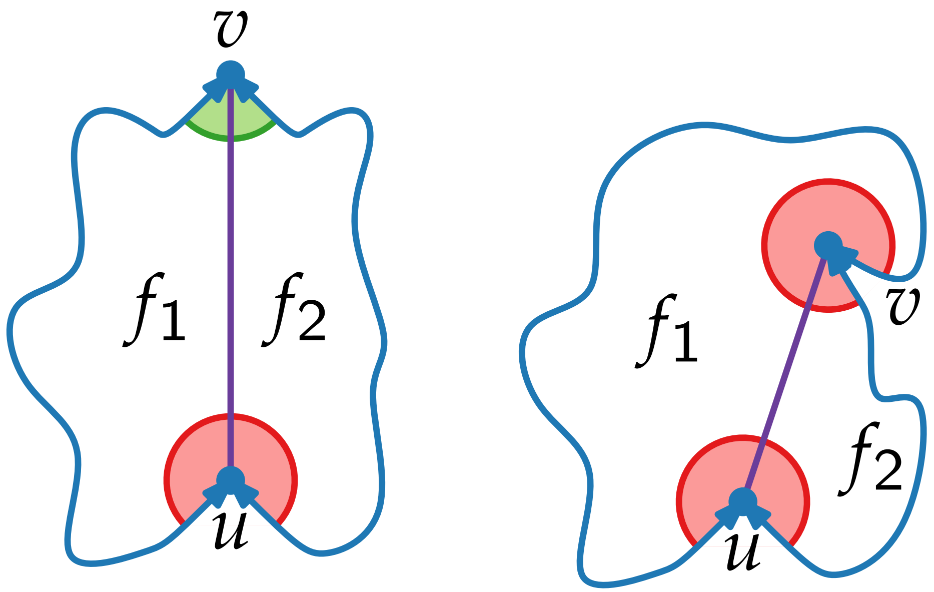
Proof by induction.

- $L(f) = 0$  $\Rightarrow S(f) = 2$

- $L(f) \geq 1$

Split f with **edge** from a large angle at a "low" sink u to

- sink v with small/large angle:



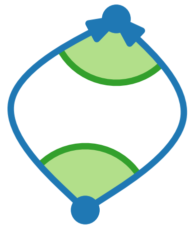
$$\begin{aligned} L(f) - S(f) &= L(f_1) + L(f_2) + 1 \\ &\quad - (S(f_1) + S(f_2) - 1) \\ &= -2 \end{aligned}$$

Angle relations

Lemma 2.

$$L(f) - S(f) = \begin{cases} -2, & f \neq f_0 \\ +2, & f = f_0 \end{cases}$$

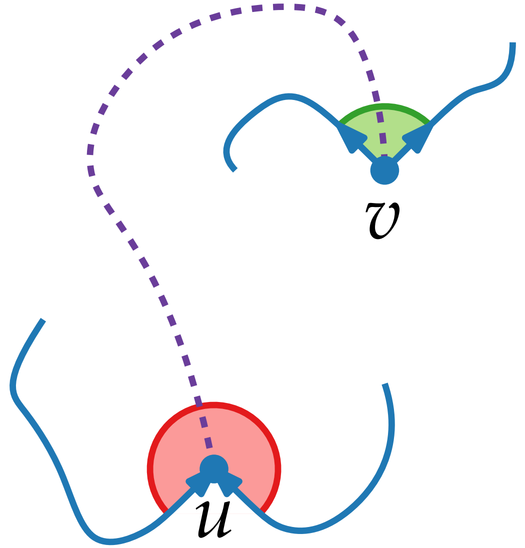
Proof by induction.

- $L(f) = 0$  $\Rightarrow S(f) = 2$

- $L(f) \geq 1$

Split f with **edge** from a large angle at a "low" sink u to

- source v with small angle:



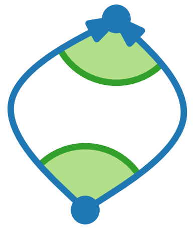
Angle relations

Lemma 2.

$$L(f) - S(f) = \begin{cases} -2, & f \neq f_0 \\ +2, & f = f_0 \end{cases}$$

Proof by induction.

- $L(f) = 0$

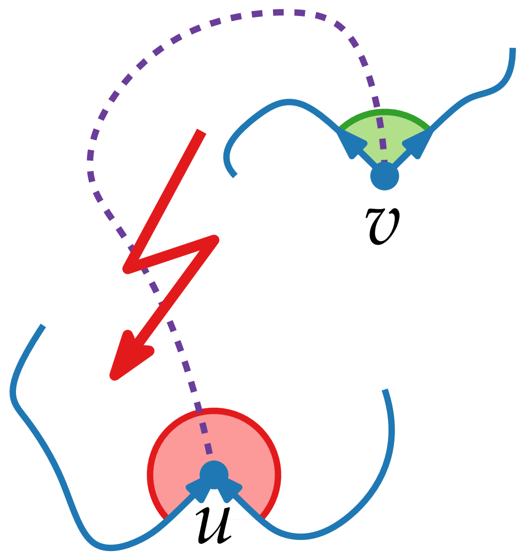


$\Rightarrow S(f) = 2$

- $L(f) \geq 1$

Split f with **edge** from a large angle at a "low" sink u to

- source v with ~~small~~ angle:

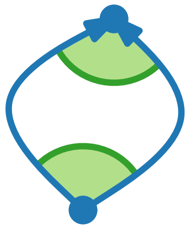


Angle relations

Lemma 2.

$$L(f) - S(f) = \begin{cases} -2, & f \neq f_0 \\ +2, & f = f_0 \end{cases}$$

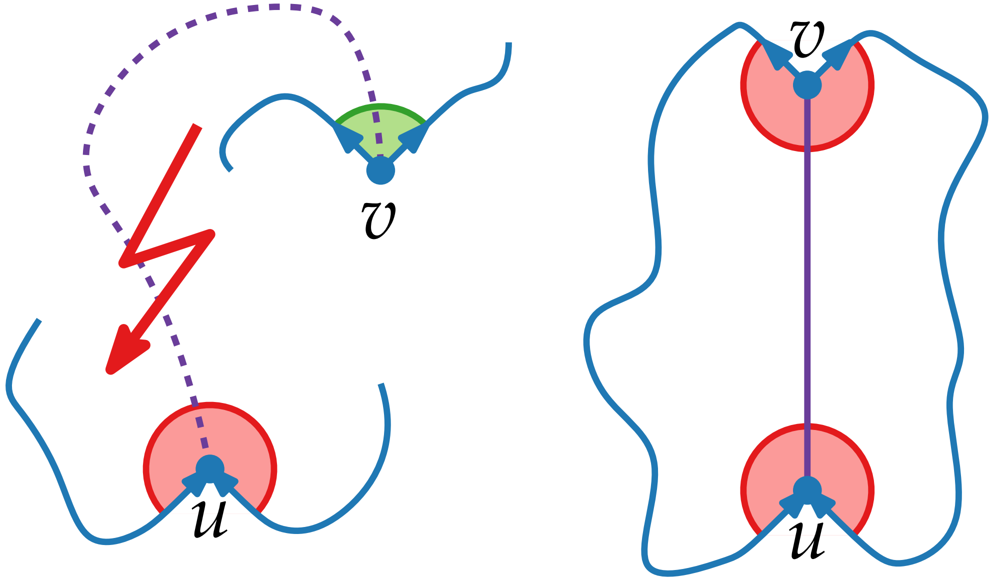
Proof by induction.

- $L(f) = 0$  $\Rightarrow S(f) = 2$

- $L(f) \geq 1$

Split f with **edge** from a large angle at a "low" sink u to

- source v with ~~small~~/large angle:



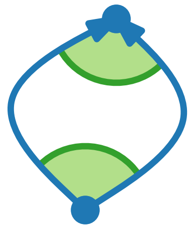
$$\begin{aligned} L(f) - S(f) &= L(f_1) + L(f_2) + 2 \\ &\quad - (S(f_1) + S(f_2)) \\ &= -2 \end{aligned}$$

Angle relations

Lemma 2.

$$L(f) - S(f) = \begin{cases} -2, & f \neq f_0 \\ +2, & f = f_0 \end{cases}$$

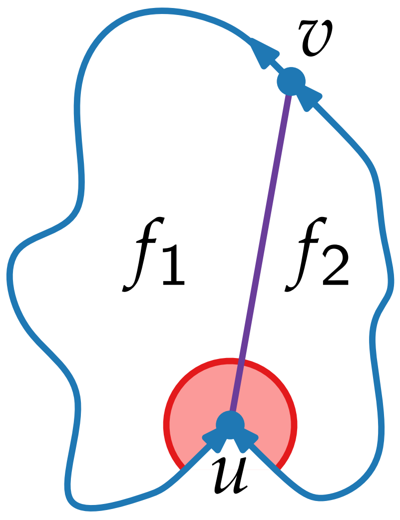
Proof by induction.

- $L(f) = 0$  $\Rightarrow S(f) = 2$

- $L(f) \geq 1$

Split f with **edge** from a large angle at a "low" sink u to

- vertex v that is neither source nor sink:



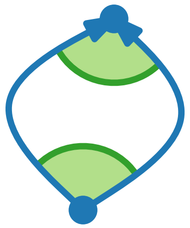
$$\begin{aligned} L(f) - S(f) &= L(f_1) + L(f_2) + 1 \\ &\quad - (S(f_1) + S(f_2) - 1) \\ &= -2 \end{aligned}$$

Angle relations

Lemma 2.

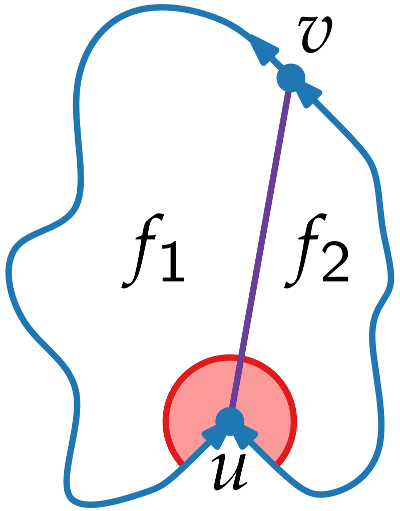
$$L(f) - S(f) = \begin{cases} -2, & f \neq f_0 \\ +2, & f = f_0 \end{cases}$$

Proof by induction.

- $L(f) = 0$  $\Rightarrow S(f) = 2$
- $L(f) \geq 1$

Split f with **edge** from a large angle at a "low" sink u to

- vertex v that is neither source nor sink:



$$\begin{aligned} L(f) - S(f) &= L(f_1) + L(f_2) + 1 \\ &\quad - (S(f_1) + S(f_2) - 1) \\ &= -2 \end{aligned}$$

- Otherwise "high" source u exists.

Number of large angles

Lemma 3.
In every upward planar drawing of G holds that

- for each vertex $v \in V$: $L(v) = \begin{cases} 0 & v \text{ inner vertex,} \\ 1 & v \text{ source/sink;} \end{cases}$
- for each face f: $L(f) = \begin{cases} A(f) - 1 & f \neq f_0, \\ A(f) + 1 & f = f_0. \end{cases}$

Proof.
Observation and from Lemma 1: $L(f) + S(f) = 2A(f)$
and from Lemma 2: $L(f) - S(f) = \pm 2.$

Assignment of large angles to faces

- Let S and T be the sets of sources and sinks, respectively.

Assignment of large angles to faces

- Let S and T be the sets of sources and sinks, respectively.

Definition.
A **consistent assignment** $\Phi\colon S \cup T \to F$ is a mapping where

$$\Phi\colon v \mapsto \text{ incident face, where } v \text{ forms large angle}$$

such that

$$|\Phi^{-1}(f)| =$$

Assignment of large angles to faces

- Let S and T be the sets of sources and sinks, respectively.

Definition.
A **consistent assignment** $\Phi\colon S \cup T \to F$ is a mapping where

$$\Phi\colon v \mapsto \text{ incident face, where } v \text{ forms large angle}$$

such that

$$|\Phi^{-1}(f)| = L(f) =$$

Assignment of large angles to faces

- Let S and T be the sets of sources and sinks, respectively.

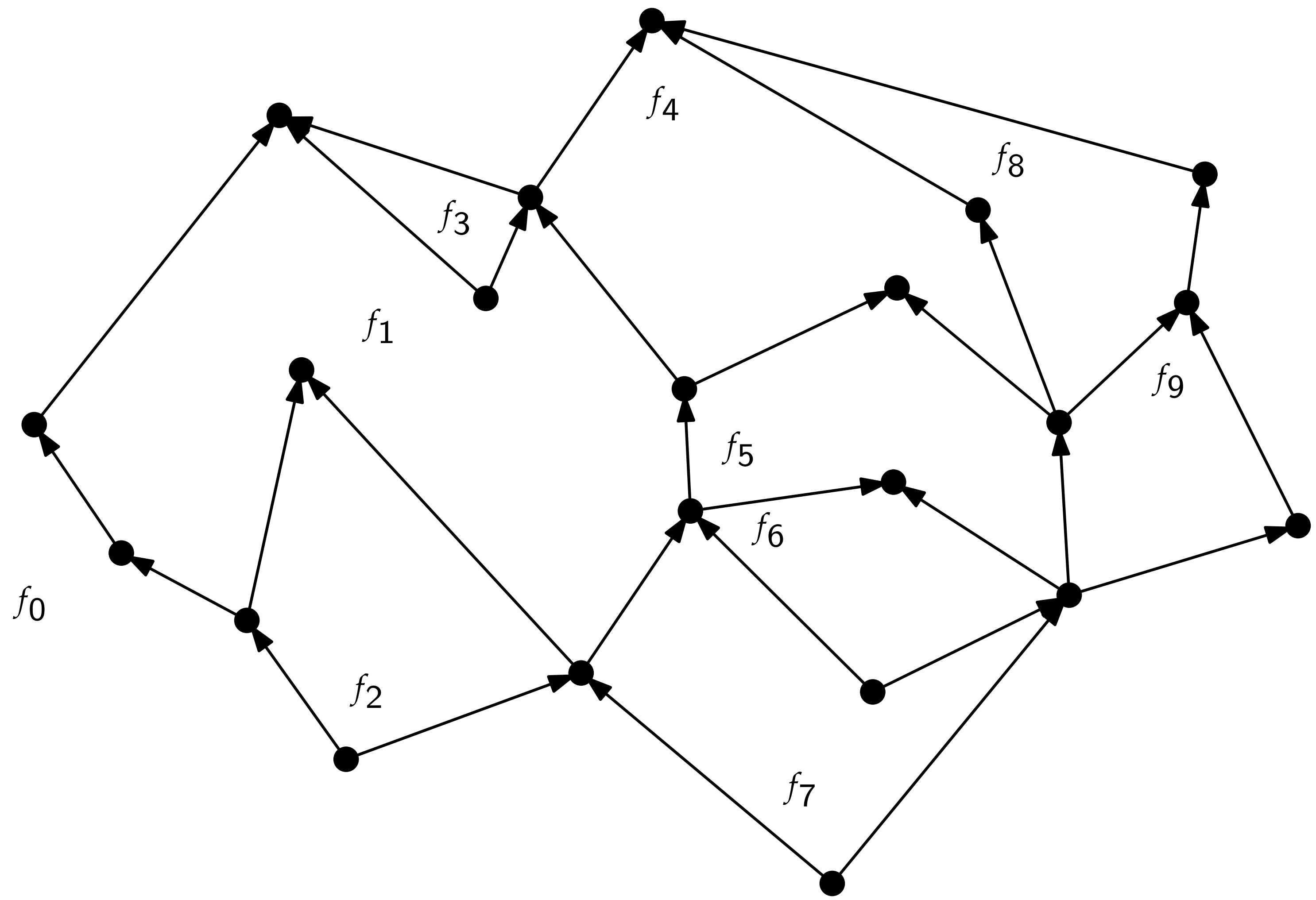
Definition.
A **consistent assignment** $\Phi\colon S \cup T \to F$ is a mapping where

$$\Phi\colon v \mapsto \text{ incident face, where } v \text{ forms large angle}$$

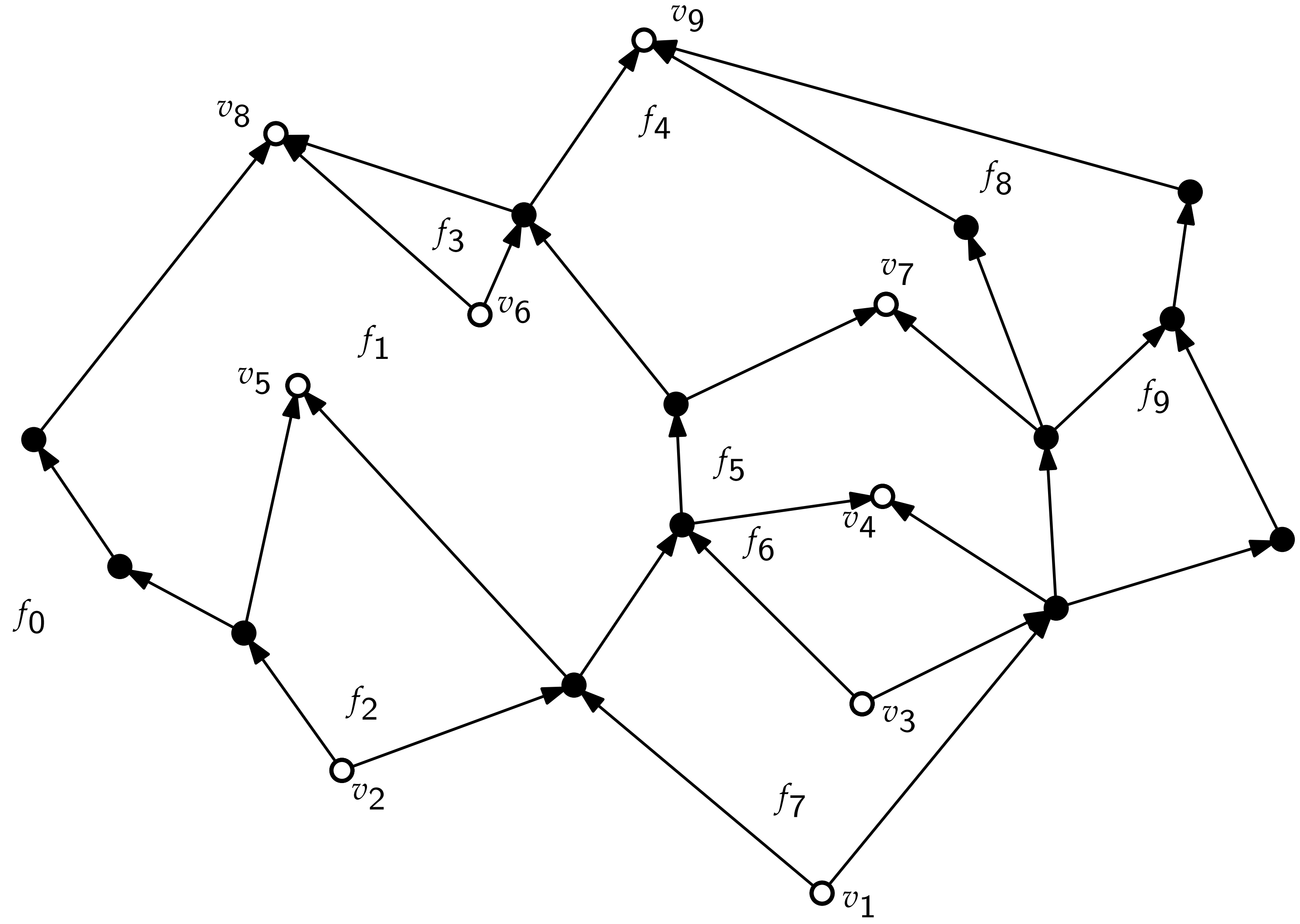
such that

$$|\Phi^{-1}(f)| = L(f) = \begin{cases} A(f) - 1 & \text{if } f \neq f_0, \\ A(f) + 1 & \text{if } f = f_0. \end{cases}$$

Example of angle to face assignment

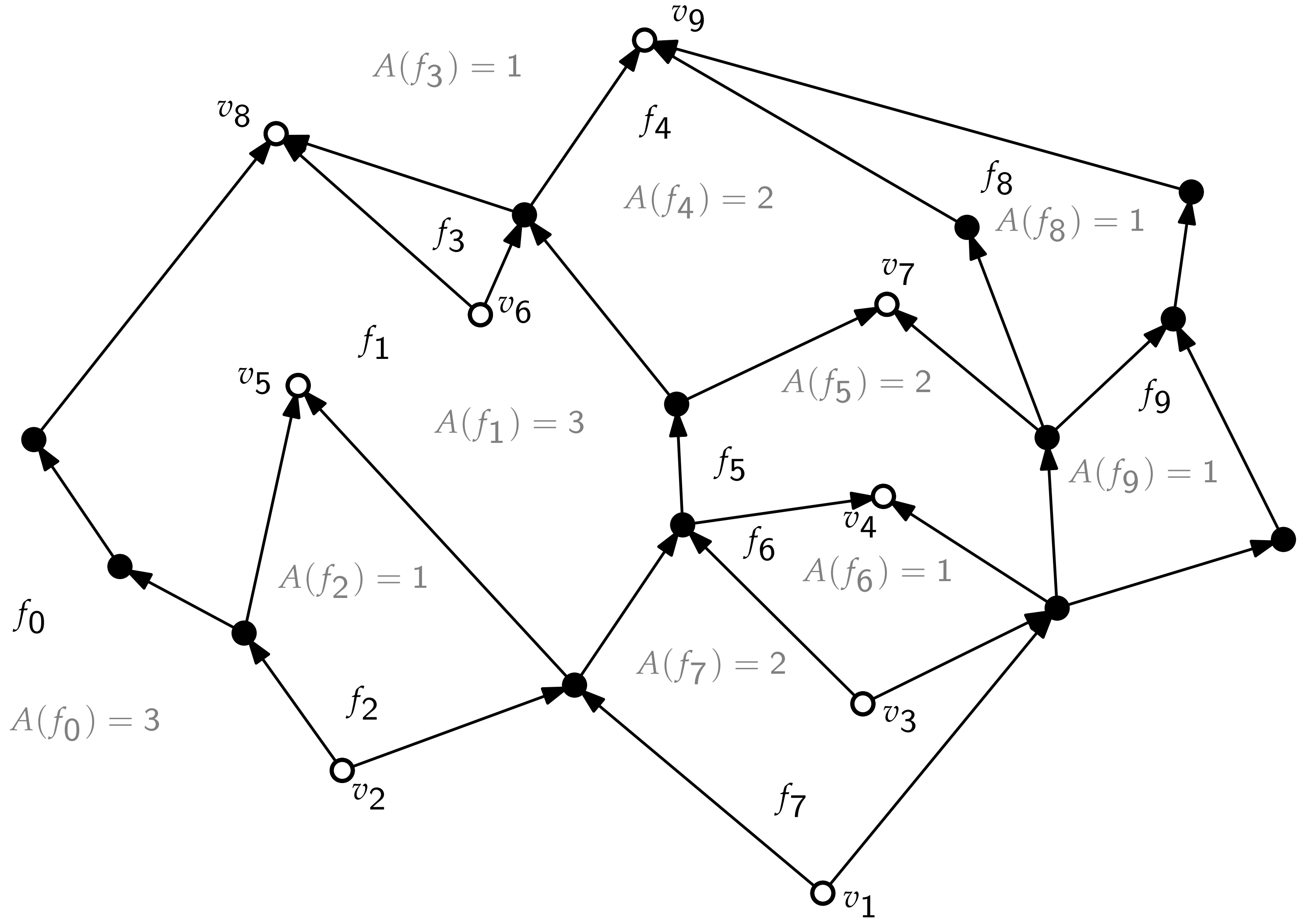


Example of angle to face assignment



○ global sources & sinks

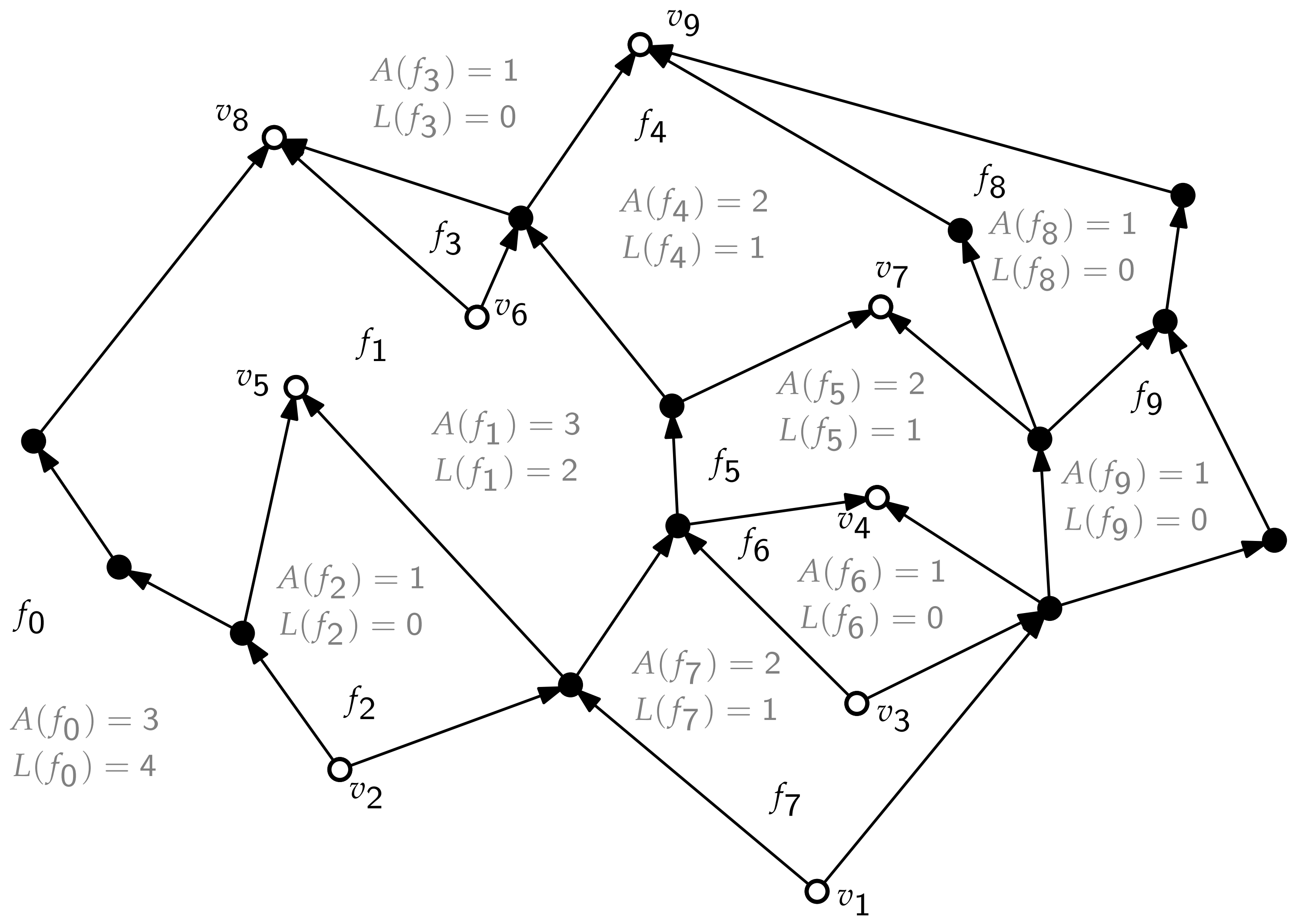
Example of angle to face assignment



- ○ global sources & sinks

$A(f)$ # sources/sinks of f

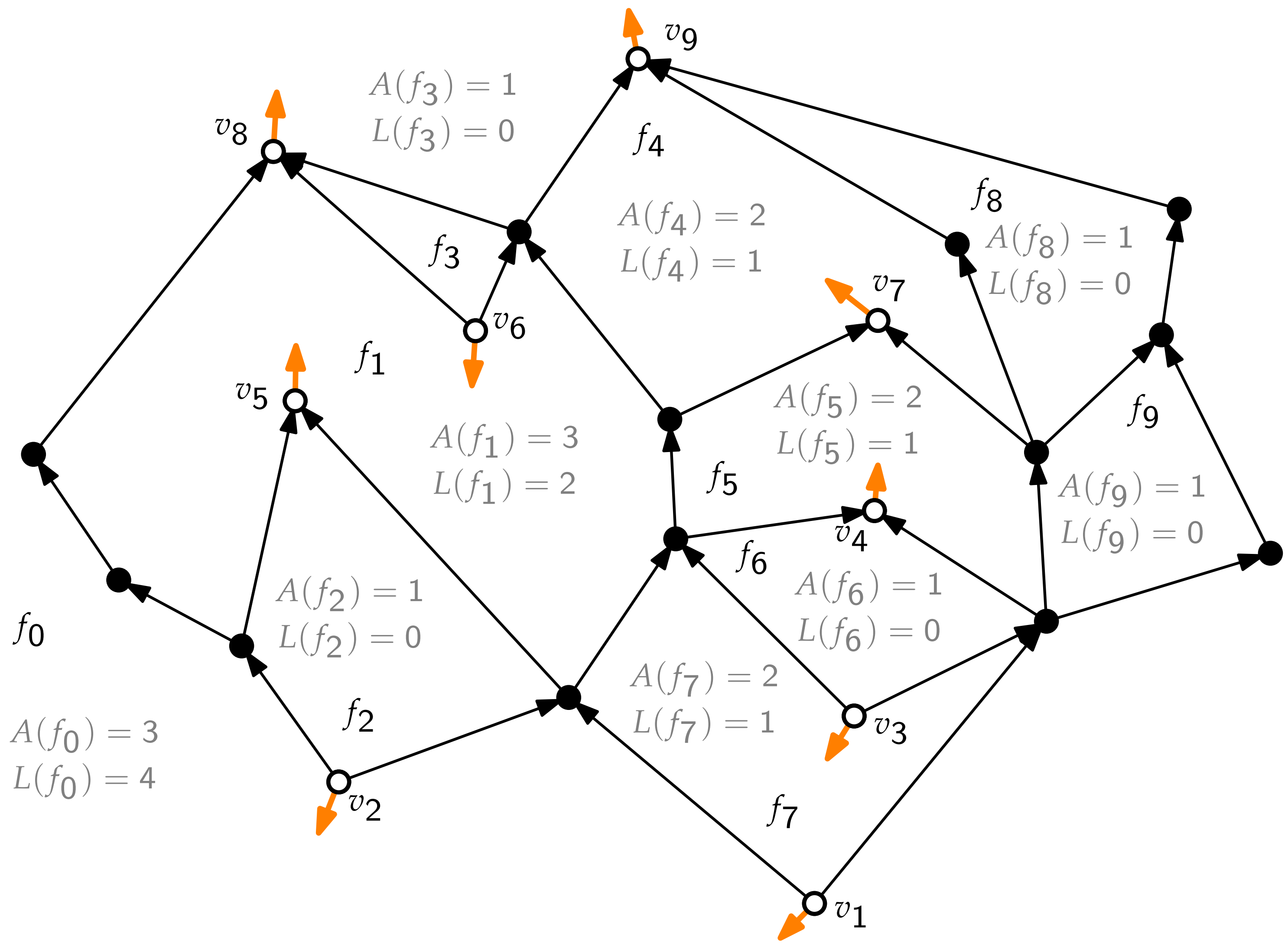
Example of angle to face assignment



∘ global sources & sinks

$A(f)$ # sources/sinks of f

Example of angle to face assignment



- ○ global sources & sinks
- $A(f)$ # sources/sinks of f
- assignment $\Phi : S \cup T \to F$

Result characterisation

Theorem 3.
Let $G = (V, E)$ be an acyclic plane digraph with embedding given by F, f_0.
Then G is upward planar (respecting F, f_0) if and only if G is bimodal and there exists consistent assignment Φ.

Result characterisation

Theorem 3.
Let $G = (V, E)$ be an acyclic plane digraph with embedding given by F, f_0.
Then G is upward planar (respecting F, f_0) if and only if G is bimodal and there exists consistent assignment Φ.

Proof.
$\Rightarrow$: As constructed before.

Result characterisation

Theorem 3.
Let $G = (V, E)$ be an acyclic plane digraph with embedding given by F, f_0.
Then G is upward planar (respecting F, f_0) if and only if G is bimodal and there exists consistent assignment Φ.

Proof.

$\Rightarrow$: As constructed before.

$\Leftarrow$: Idea:

- Construct planar st-digraph that is supergraph of G.
- Apply equivalence from Theorem 1.

Refinement algorithm – $\Phi, F, f_0 \rightarrow$ st-digraph

Refinement algorithm – $\Phi, F, f_0 \to$ st-digraph

Let f be a face. Consider the clockwise angle sequence σ_f of L/S on local sources and sinks of f.

Refinement algorithm – $\Phi, F, f_0 \rightarrow$ st-digraph

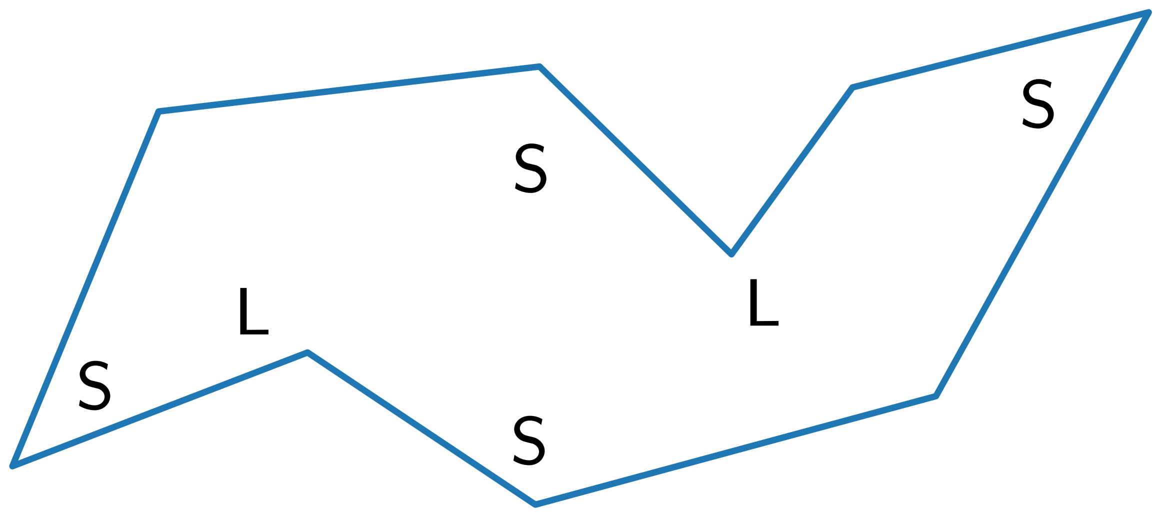
Let f be a face. Consider the clockwise angle sequence σ_f of L/S on local sources and sinks of f.

- Goal: Add edges to break large angles (sources and sinks).

Refinement algorithm – $\Phi, F, f_0 \to$ st-digraph

Let f be a face. Consider the clockwise angle sequence σ_f of L/S on local sources and sinks of f.

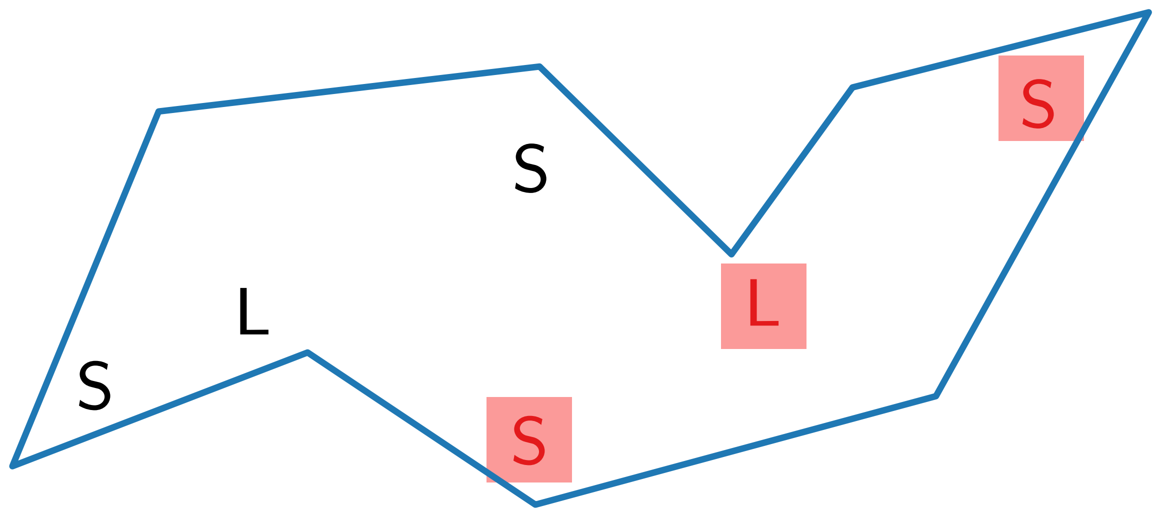
- Goal: Add edges to break large angles (sources and sinks).
- For $f \neq f_0$ with $|\sigma_f| \geq 2$ containing $\langle$L, S, S$\rangle$ at vertices x, y, z:



Refinement algorithm – $\Phi, F, f_0 \to$ st-digraph

Let f be a face. Consider the clockwise angle sequence σ_f of L/S on local sources and sinks of f.

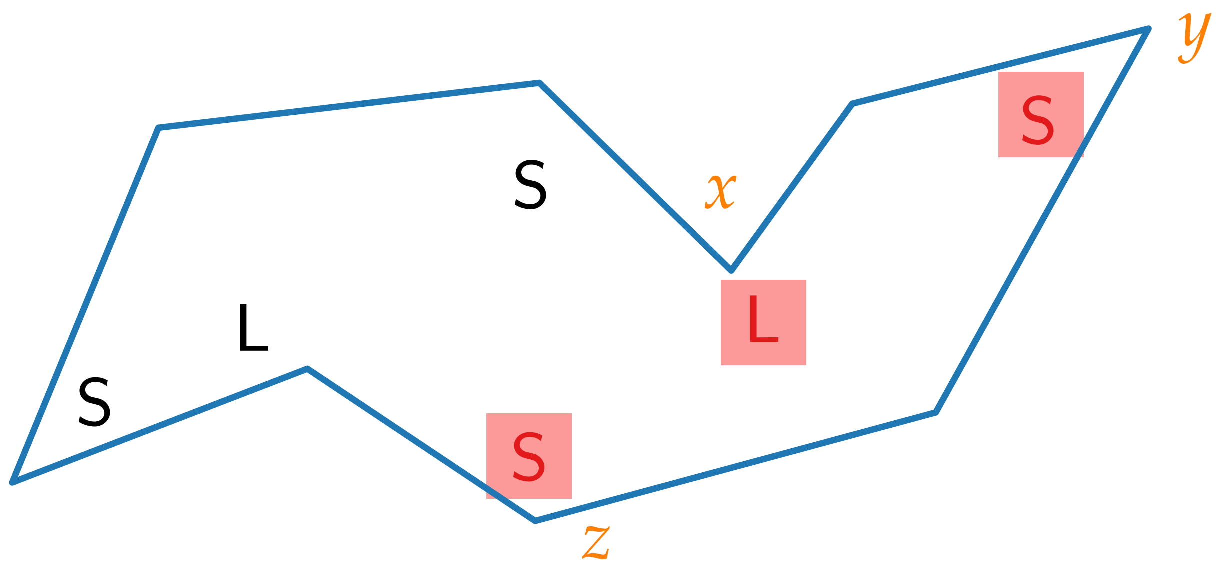
- Goal: Add edges to break large angles (sources and sinks).
- For $f \neq f_0$ with $|\sigma_f| \geq 2$ containing $\langle$L, S, S$\rangle$ at vertices x, y, z:



Refinement algorithm – $\Phi, F, f_0 \to$ st-digraph

Let f be a face. Consider the clockwise angle sequence σ_f of L/S on local sources and sinks of f.

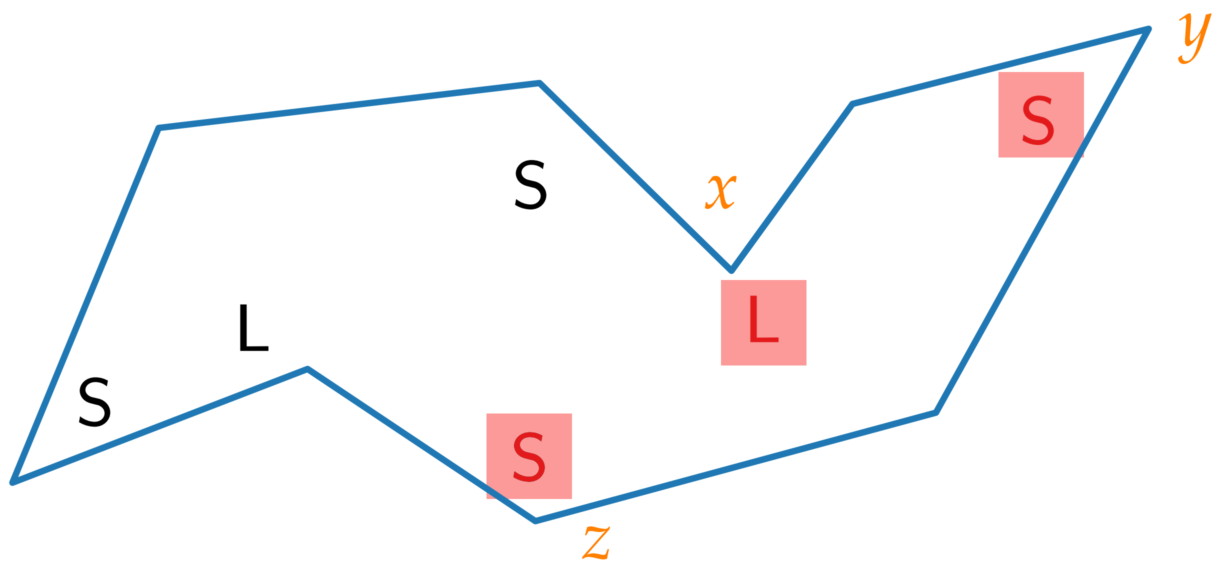
- Goal: Add edges to break large angles (sources and sinks).
- For $f \neq f_0$ with $|\sigma_f| \geq 2$ containing $\langle$L, S, S$\rangle$ at vertices x, y, z:



Refinement algorithm – $\Phi, F, f_0 \rightarrow$ st-digraph

Let f be a face. Consider the clockwise angle sequence σ_f of L/S on local sources and sinks of f.

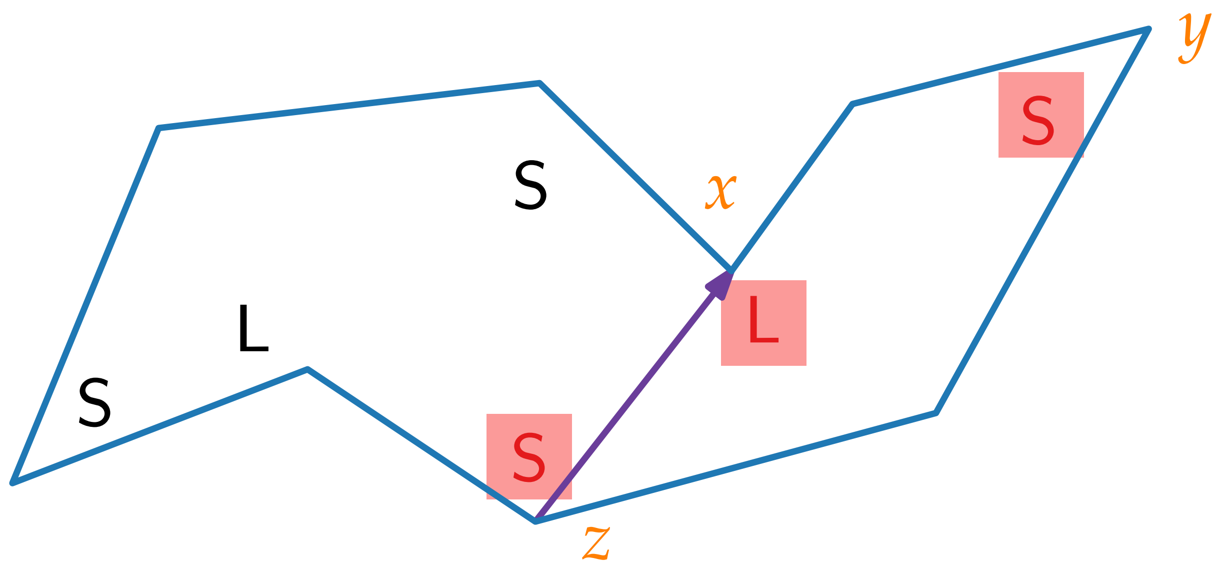
- Goal: Add edges to break large angles (sources and sinks).
- For $f \neq f_0$ with $|\sigma_f| \geq 2$ containing $\langle$L, S, S$\rangle$ at vertices x, y, z:
  - x source $\Rightarrow$ insert edge (z, x)



Refinement algorithm – $\Phi, F, f_0 \to$ st-digraph

Let f be a face. Consider the clockwise angle sequence σ_f of L/S on local sources and sinks of f.

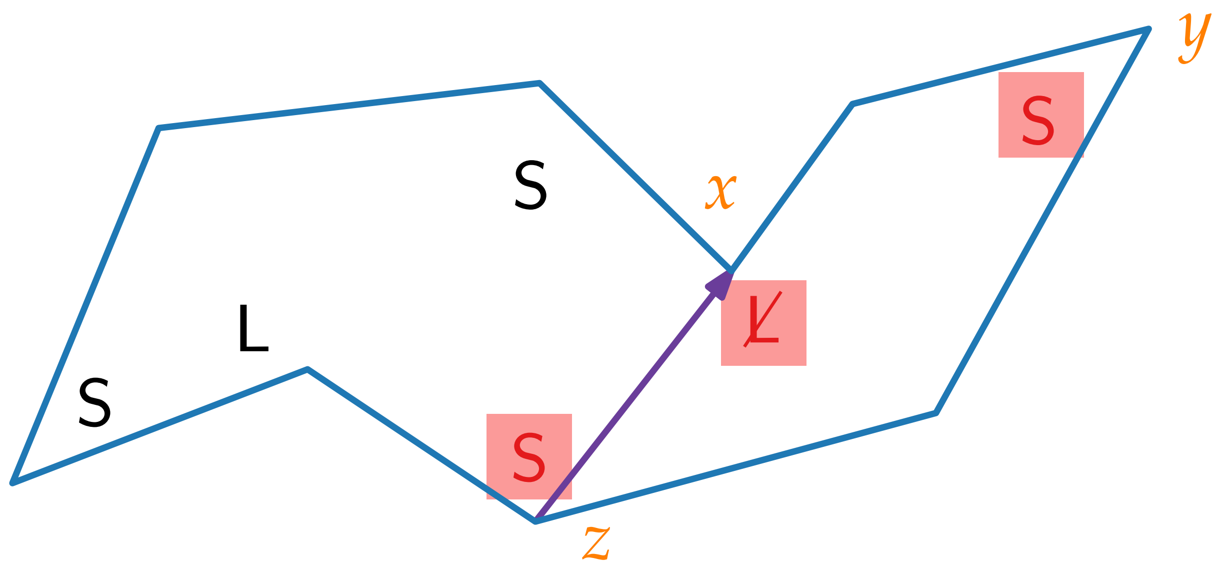
- Goal: Add edges to break large angles (sources and sinks).
- For $f \neq f_0$ with $|\sigma_f| \geq 2$ containing $\langle$L, S, S$\rangle$ at vertices x, y, z:
  - x source $\Rightarrow$ insert edge (z, x)



Refinement algorithm – $\Phi, F, f_0 \rightarrow$ st-digraph

Let f be a face. Consider the clockwise angle sequence σ_f of L/S on local sources and sinks of f.

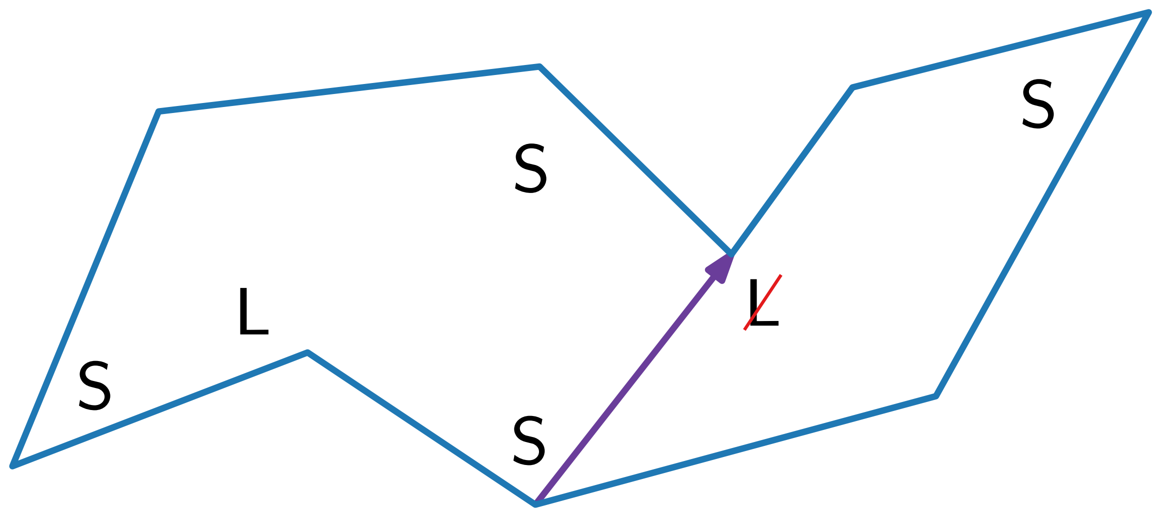
- Goal: Add edges to break large angles (sources and sinks).
- For $f \neq f_0$ with $|\sigma_f| \geq 2$ containing $\langle$L, S, S$\rangle$ at vertices x, y, z:
  - x source $\Rightarrow$ insert edge (z, x)



Refinement algorithm – $\Phi, F, f_0 \to$ st-digraph

Let f be a face. Consider the clockwise angle sequence σ_f of L/S on local sources and sinks of f.

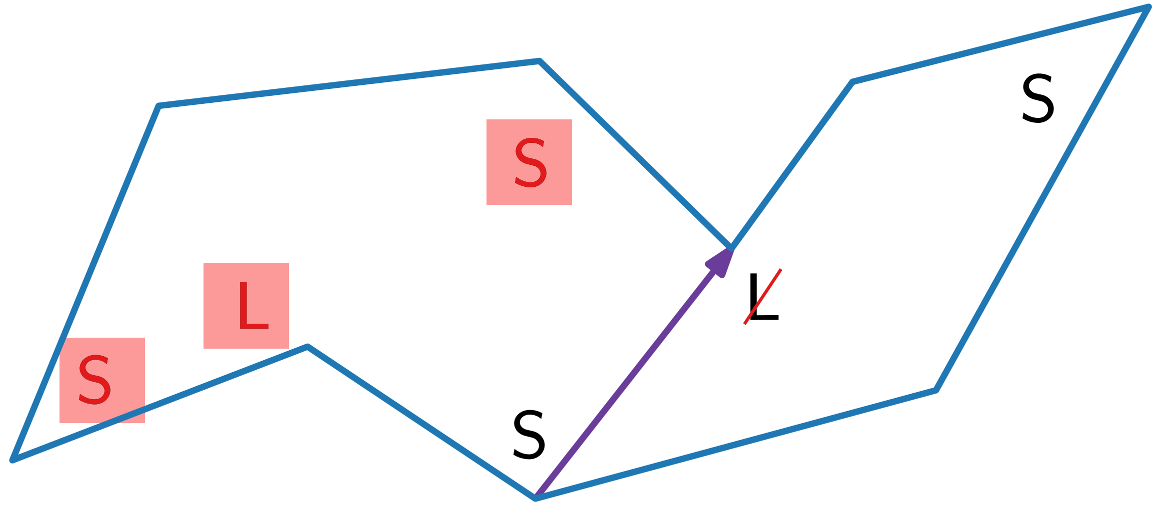
- Goal: Add edges to break large angles (sources and sinks).
- For $f \neq f_0$ with $|\sigma_f| \geq 2$ containing $\langle$L, S, S$\rangle$ at vertices x, y, z:
  - x source $\Rightarrow$ insert edge (z, x)



Refinement algorithm – $\Phi, F, f_0 \to$ st-digraph

Let f be a face. Consider the clockwise angle sequence σ_f of L/S on local sources and sinks of f.

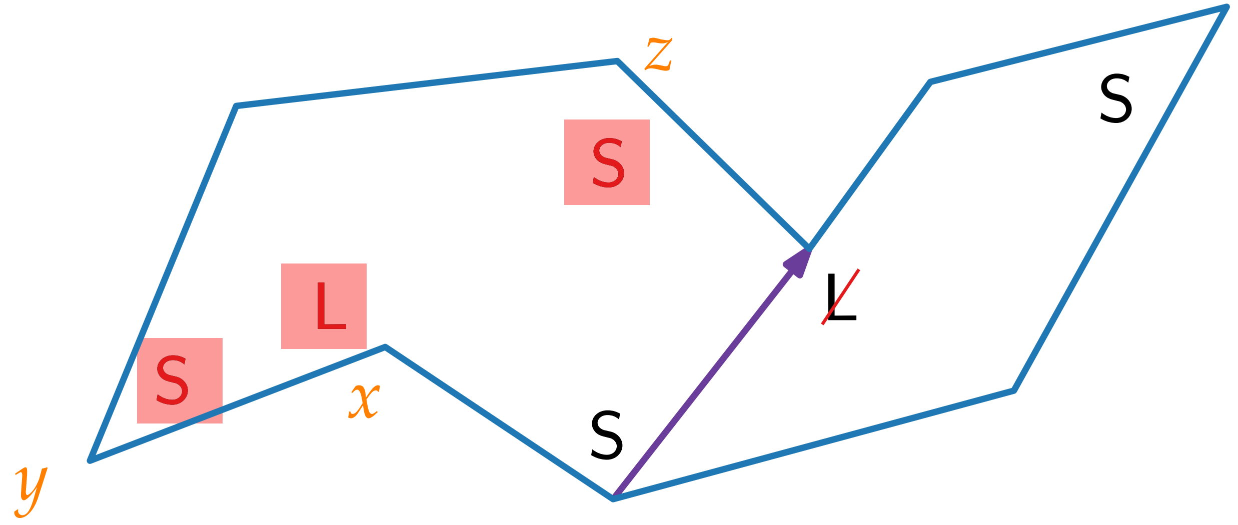
- Goal: Add edges to break large angles (sources and sinks).
- For $f \neq f_0$ with $|\sigma_f| \geq 2$ containing $\langle$L, S, S$\rangle$ at vertices x, y, z:
  - x source $\Rightarrow$ insert edge (z, x)



Refinement algorithm – $\Phi, F, f_0 \to$ st-digraph

Let f be a face. Consider the clockwise angle sequence σ_f of L/S on local sources and sinks of f.

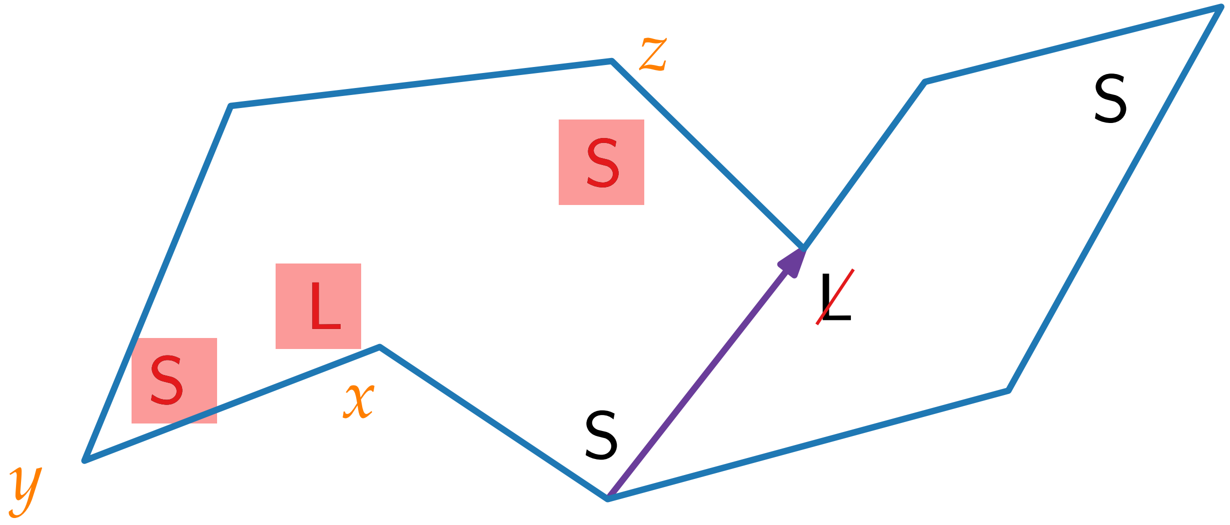
- Goal: Add edges to break large angles (sources and sinks).
- For $f \neq f_0$ with $|\sigma_f| \geq 2$ containing $\langle$L, S, S$\rangle$ at vertices x, y, z:
  - x source $\Rightarrow$ insert edge (z, x)



Refinement algorithm – $\Phi, F, f_0 \rightarrow$ st-digraph

Let f be a face. Consider the clockwise angle sequence σ_f of L/S on local sources and sinks of f.

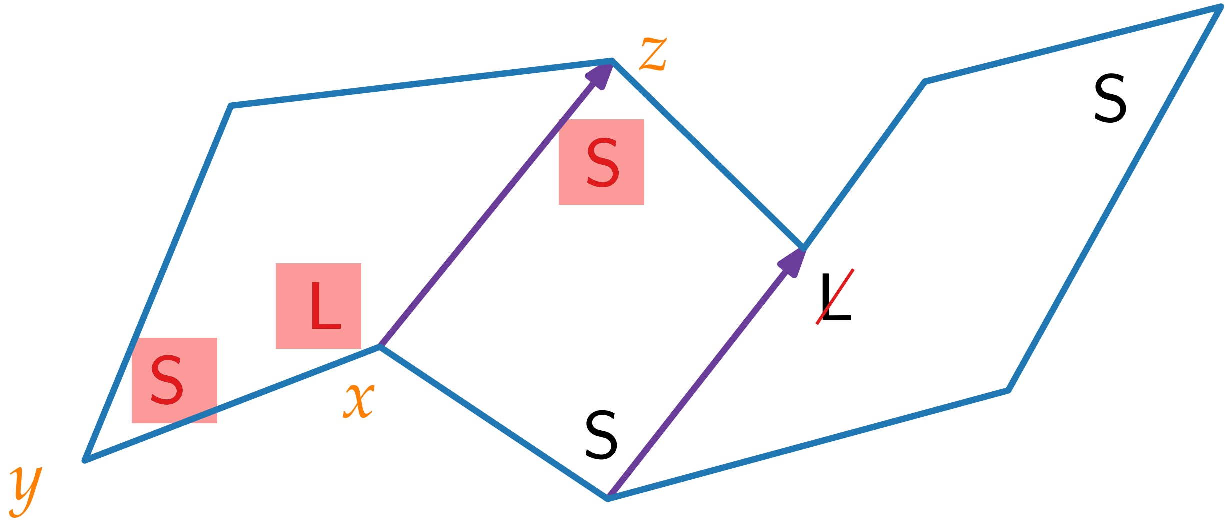
- Goal: Add edges to break large angles (sources and sinks).
- For $f \neq f_0$ with $|\sigma_f| \geq 2$ containing $\langle$L, S, S$\rangle$ at vertices x, y, z:
  - x source $\Rightarrow$ insert edge (z, x)
  - x sink $\Rightarrow$ insert edge (x, z).



Refinement algorithm – $\Phi, F, f_0 \to$ st-digraph

Let f be a face. Consider the clockwise angle sequence σ_f of L/S on local sources and sinks of f.

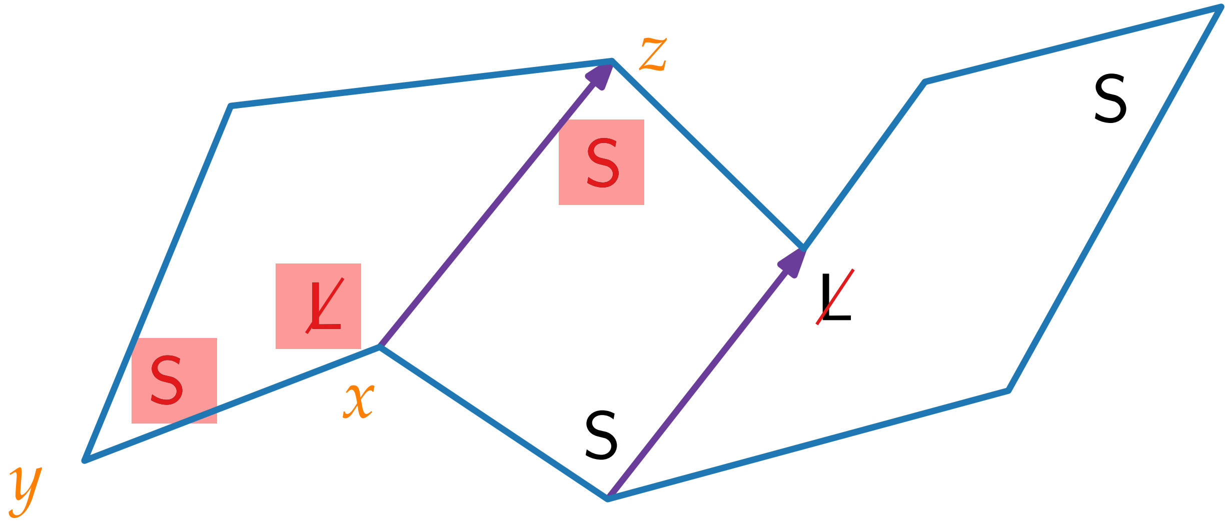
- Goal: Add edges to break large angles (sources and sinks).
- For $f \neq f_0$ with $|\sigma_f| \geq 2$ containing $\langle$L, S, S$\rangle$ at vertices x, y, z:
  - x source $\Rightarrow$ insert edge (z, x)
  - x sink $\Rightarrow$ insert edge (x, z).



Refinement algorithm – $\Phi, F, f_0 \to$ st-digraph

Let f be a face. Consider the clockwise angle sequence σ_f of L/S on local sources and sinks of f.

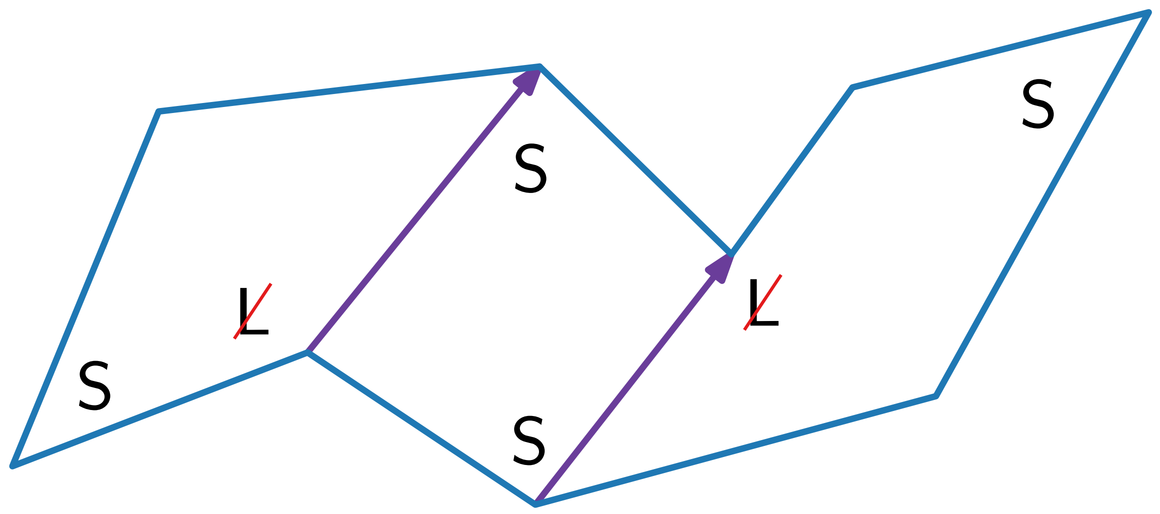
- Goal: Add edges to break large angles (sources and sinks).
- For $f \neq f_0$ with $|\sigma_f| \geq 2$ containing $\langle$L, S, S$\rangle$ at vertices x, y, z:
  - x source $\Rightarrow$ insert edge (z, x)
  - x sink $\Rightarrow$ insert edge (x, z).



Refinement algorithm – $\Phi, F, f_0 \rightarrow$ st-digraph

Let f be a face. Consider the clockwise angle sequence σ_f of L/S on local sources and sinks of f.

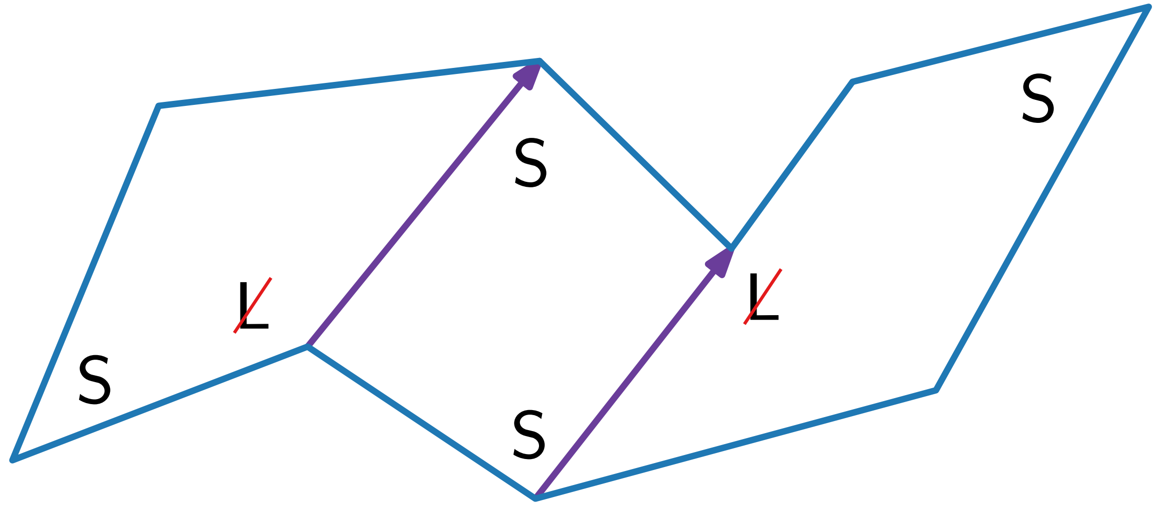
- Goal: Add edges to break large angles (sources and sinks).
- For $f \neq f_0$ with $|\sigma_f| \geq 2$ containing $\langle$L, S, S$\rangle$ at vertices x, y, z:
  - x source $\Rightarrow$ insert edge (z, x)
  - x sink $\Rightarrow$ insert edge (x, z).



Refinement algorithm – $\Phi, F, f_0 \rightarrow$ st-digraph

Let f be a face. Consider the clockwise angle sequence σ_f of L/S on local sources and sinks of f.

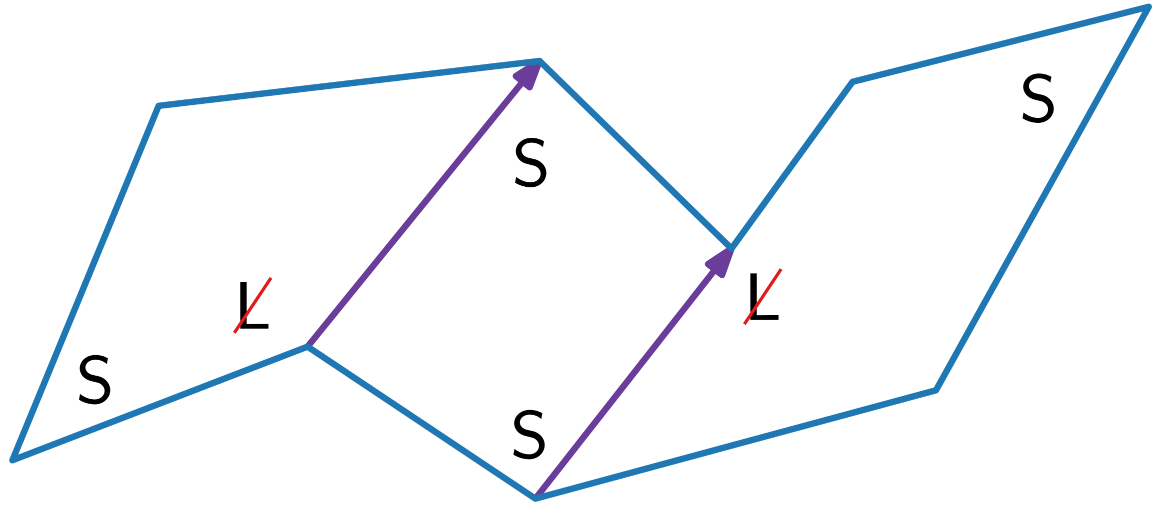
- Goal: Add edges to break large angles (sources and sinks).
- For $f \neq f_0$ with $|\sigma_f| \geq 2$ containing $\langle$L, S, S$\rangle$ at vertices x, y, z:
  - x source $\Rightarrow$ insert edge (z, x)
  - x sink $\Rightarrow$ insert edge (x, z).
- Refine outer face f_0.



Refinement algorithm – $\Phi, F, f_0 \rightarrow$ st-digraph

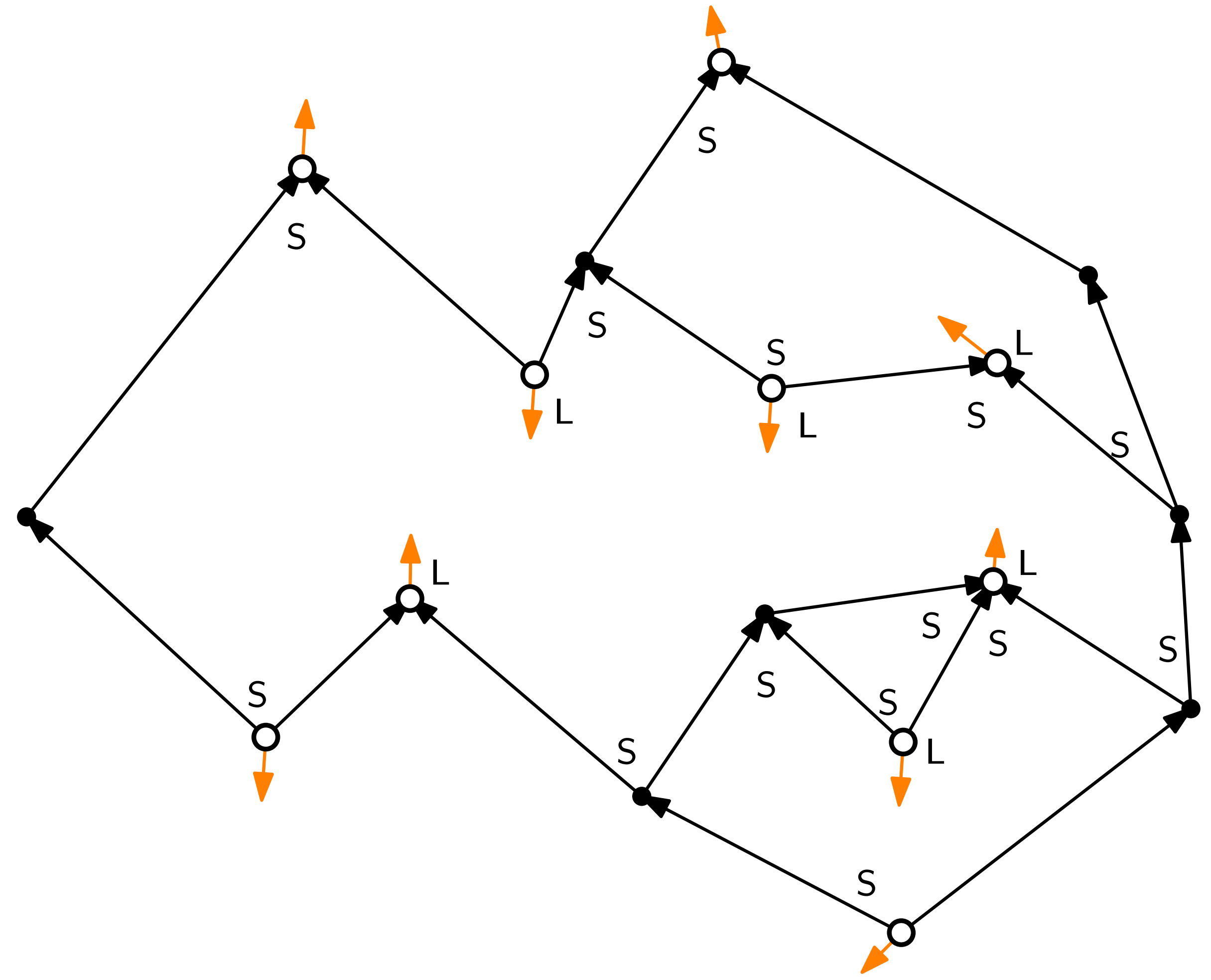
Let f be a face. Consider the clockwise angle sequence σ_f of L/S on local sources and sinks of f.

- Goal: Add edges to break large angles (sources and sinks).
- For $f \neq f_0$ with $|\sigma_f| \geq 2$ containing $\langle$L, S, S$\rangle$ at vertices x, y, z:
  - x source $\Rightarrow$ insert edge (z, x)
  - x sink $\Rightarrow$ insert edge (x, z).
- Refine outer face f_0.

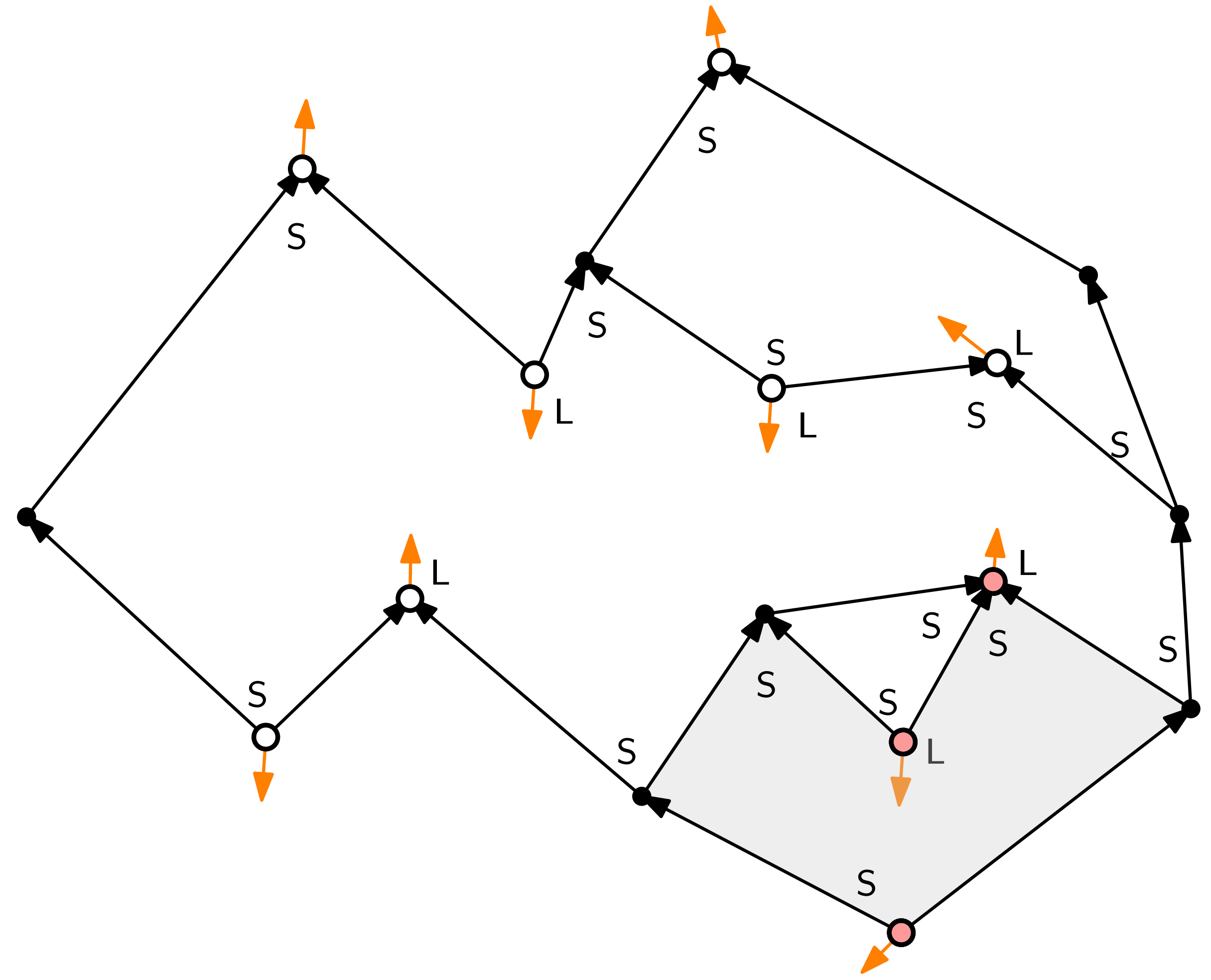


- Refine all faces. $\Rightarrow$ G is contained in a planar st-digraph.
- Planarity, acyclicity, bimodality are invariants under construction.

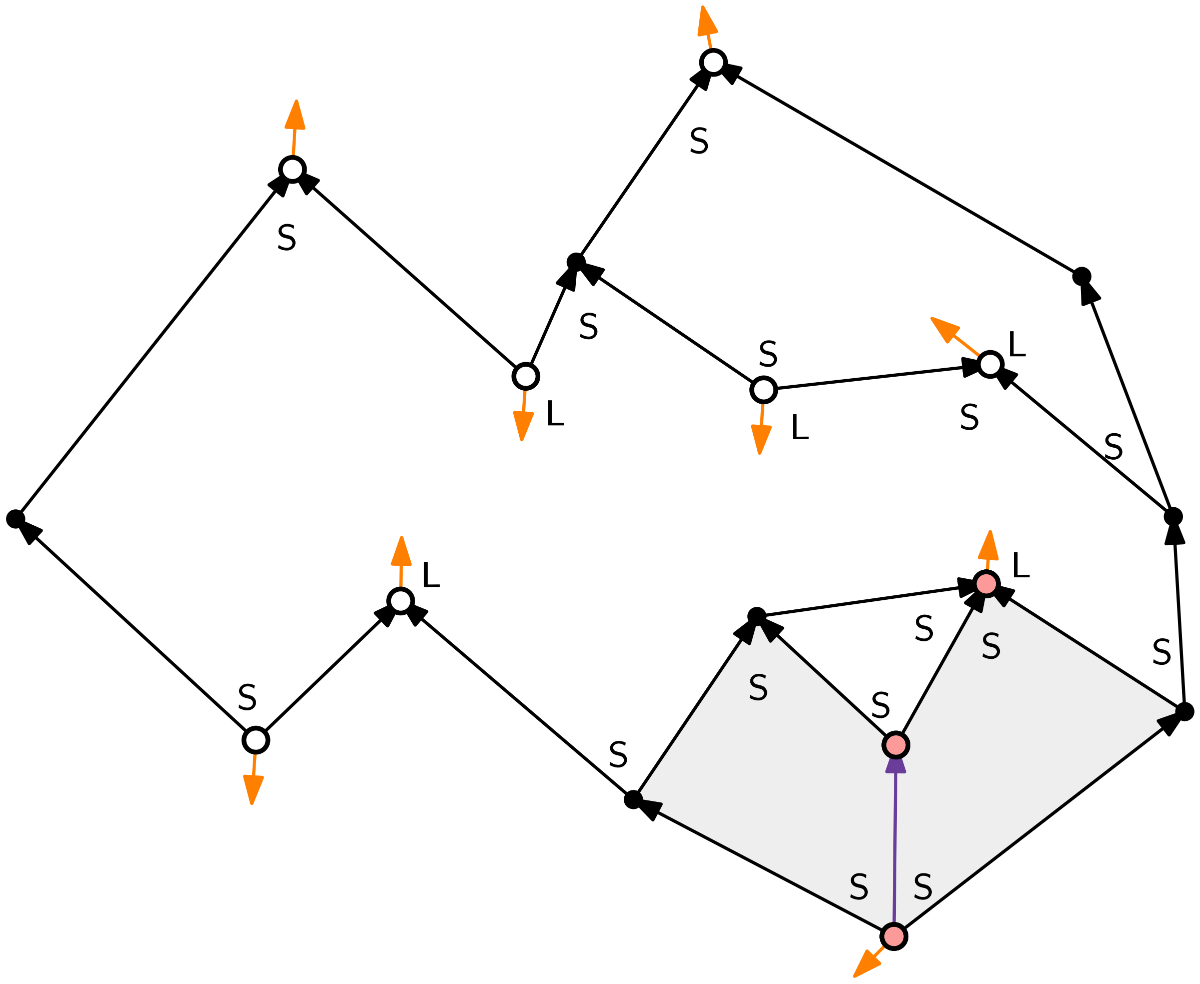
Refinement example



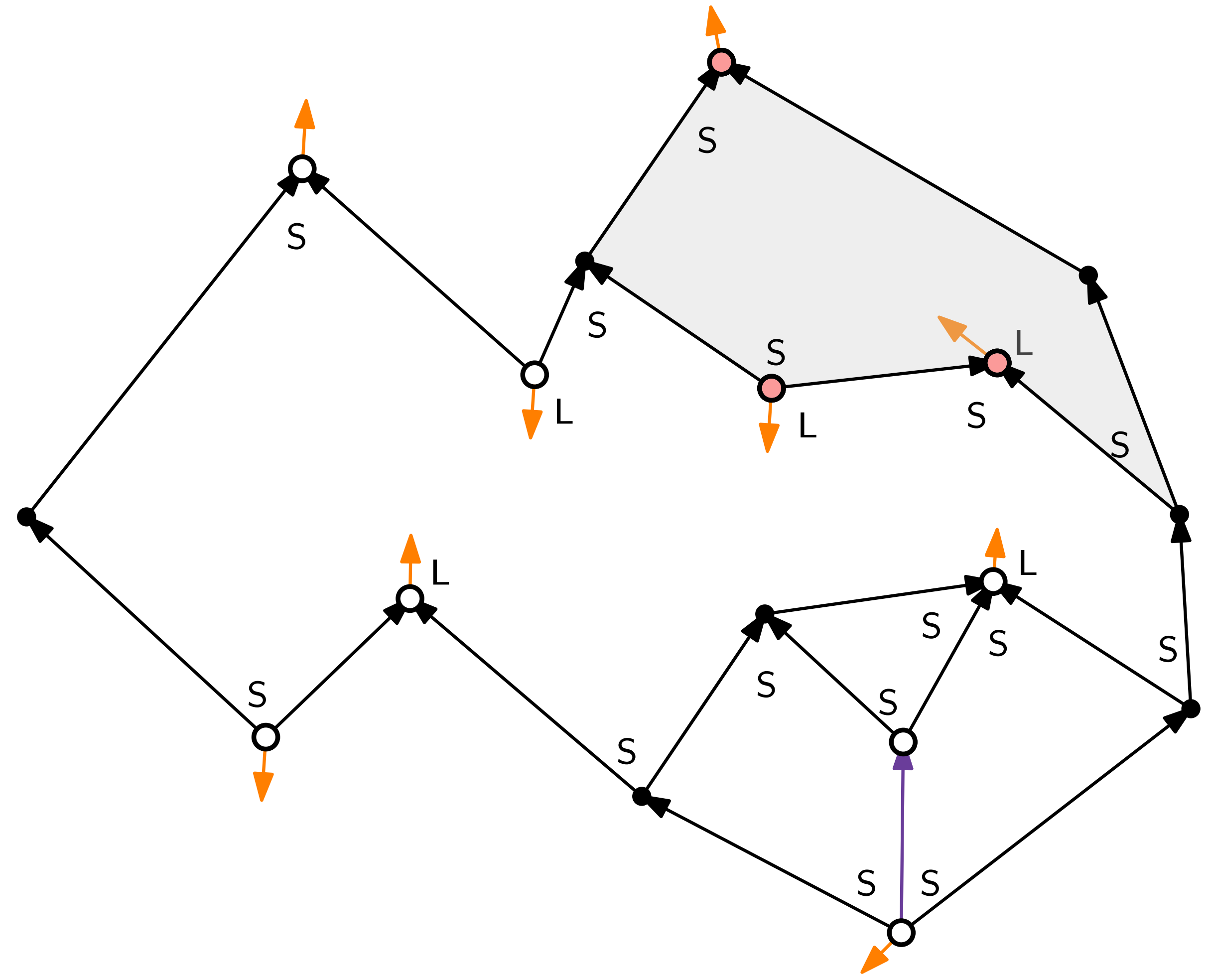
Refinement example



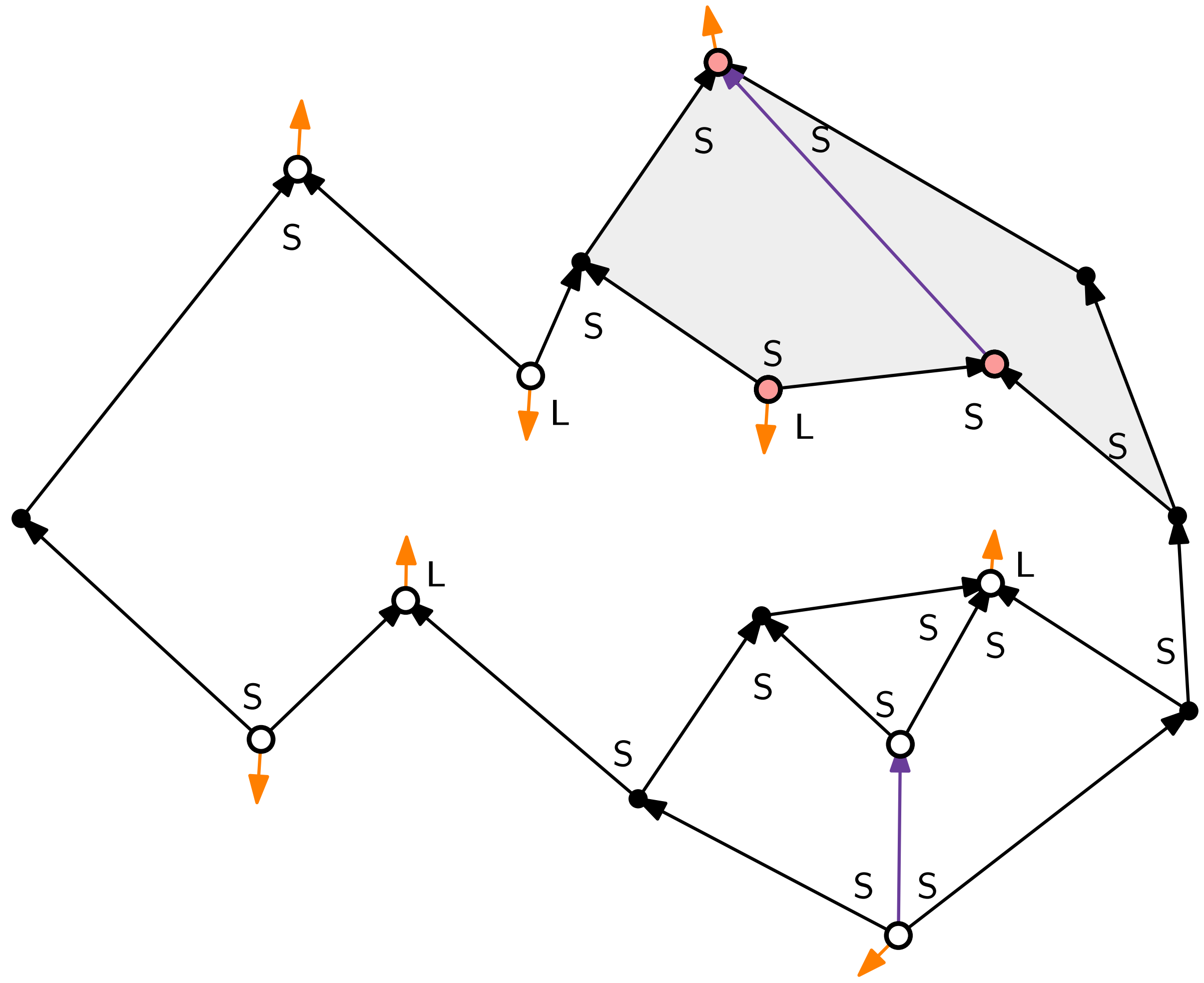
Refinement example



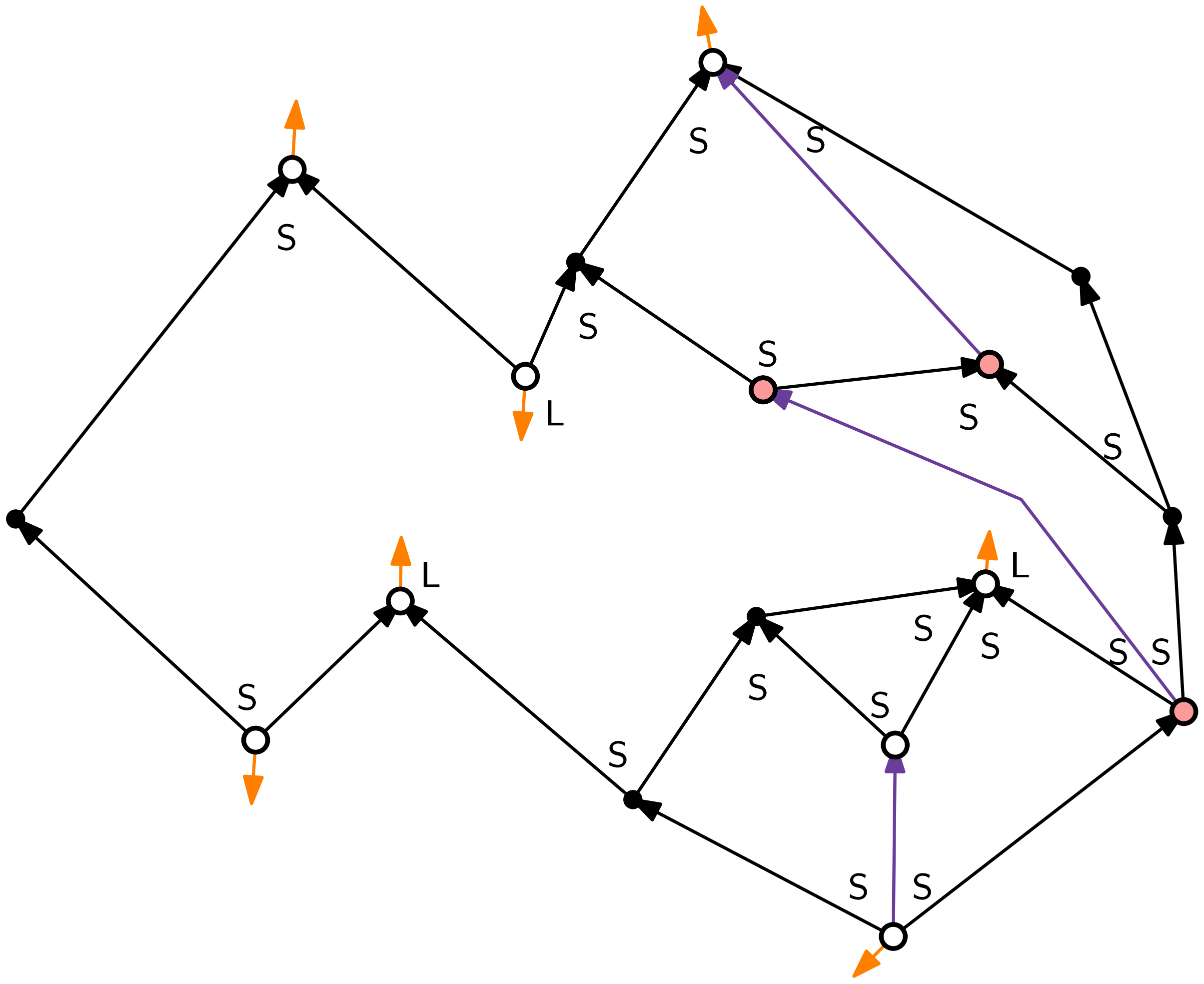
Refinement example



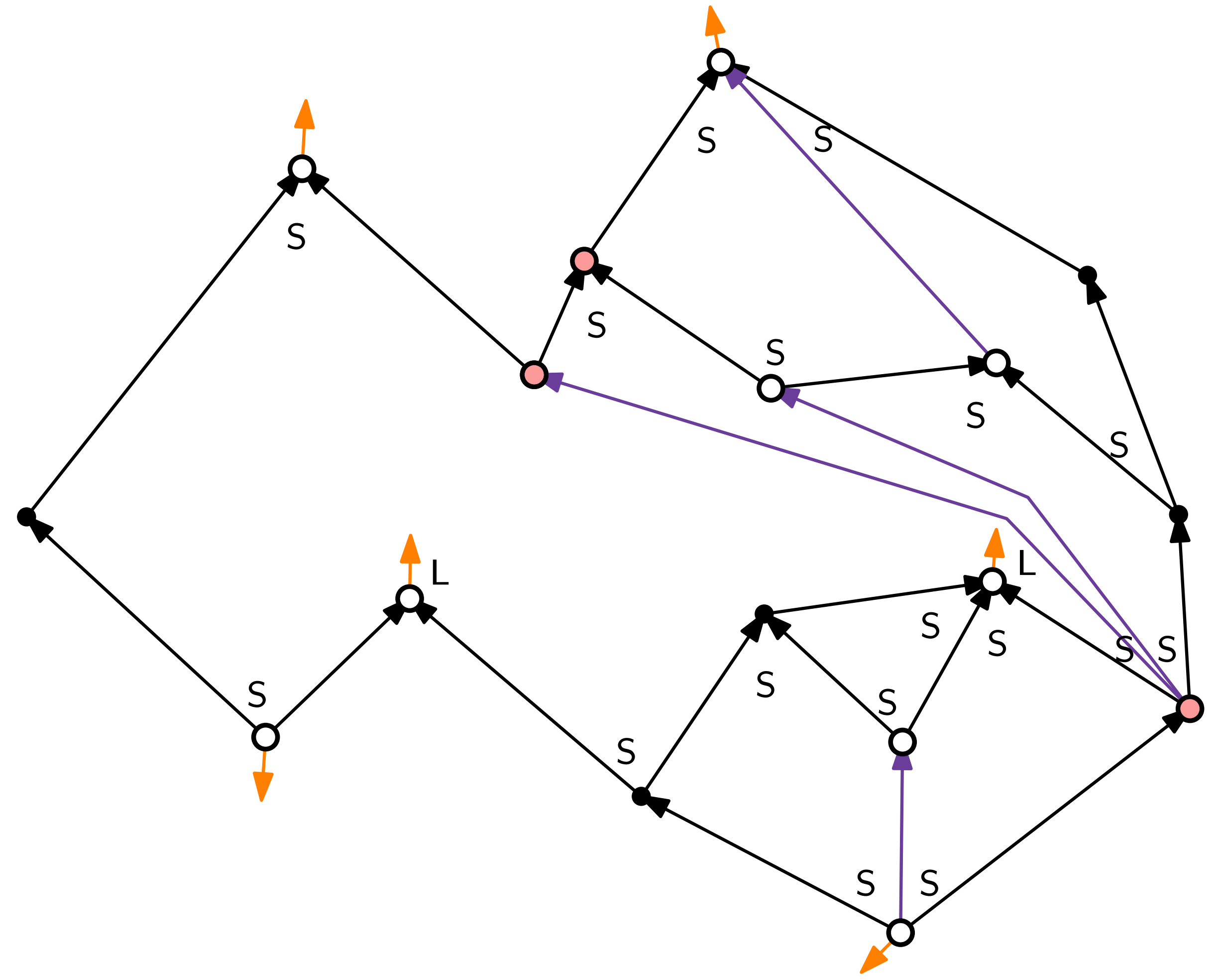
Refinement example



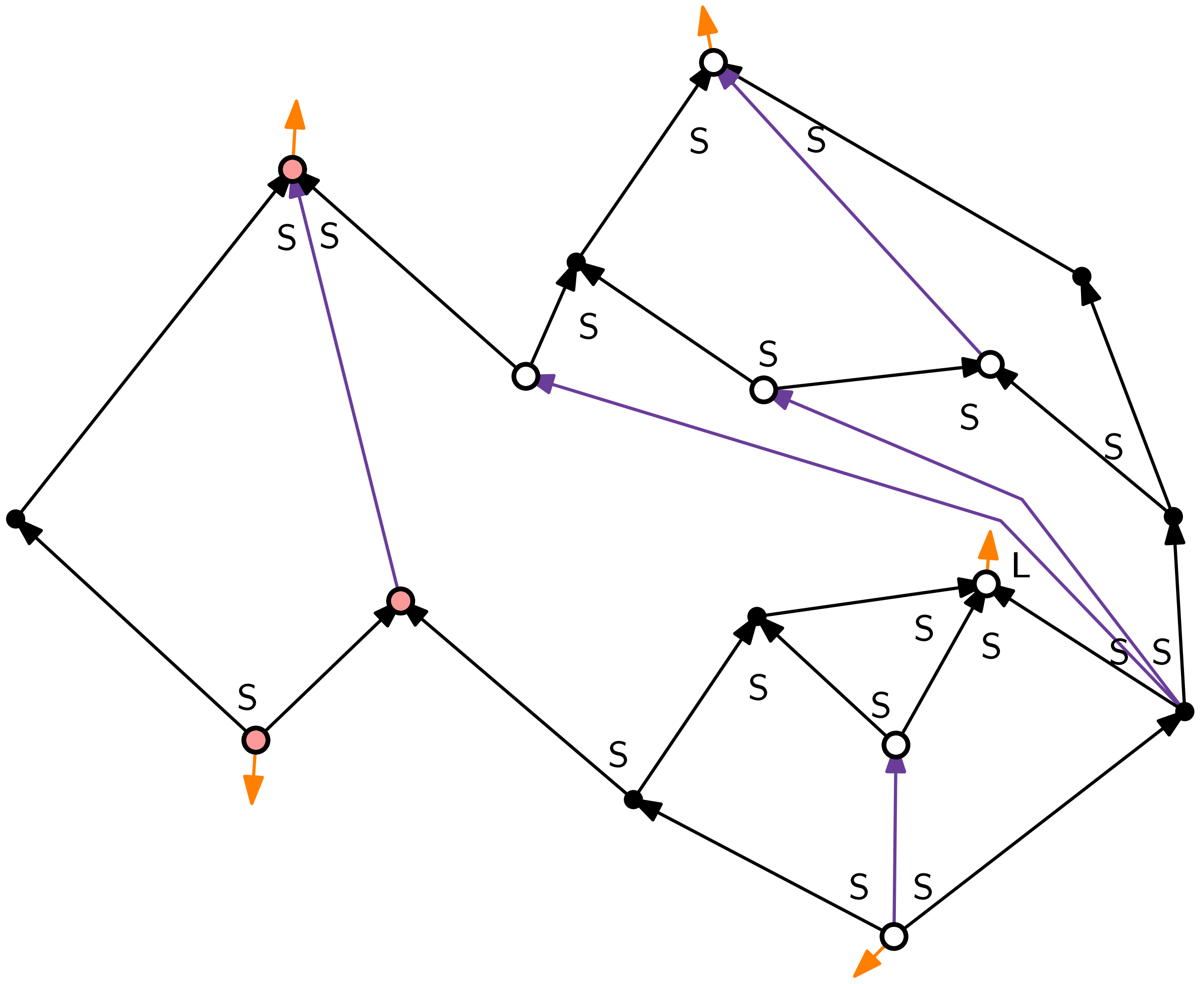
Refinement example



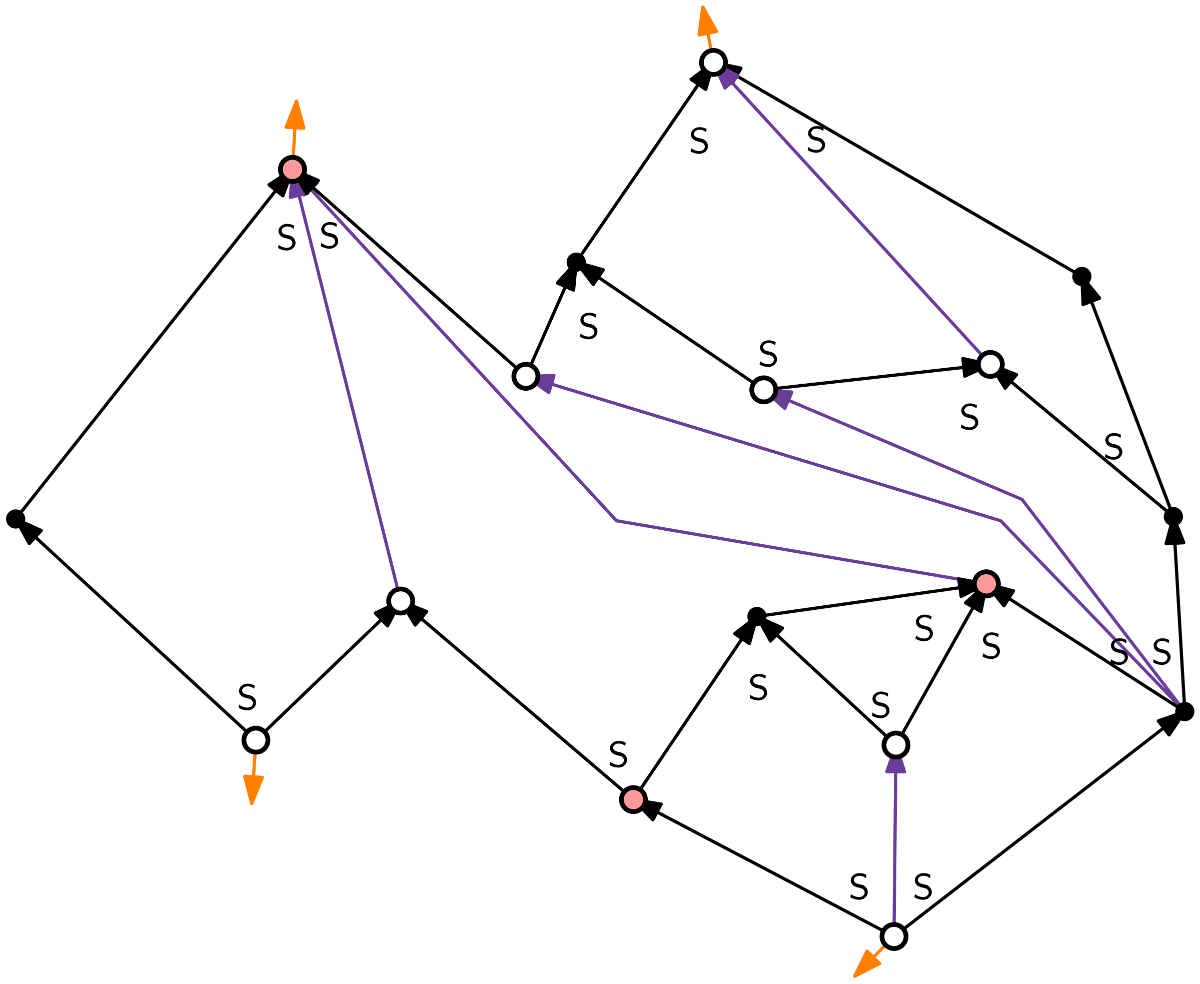
Refinement example



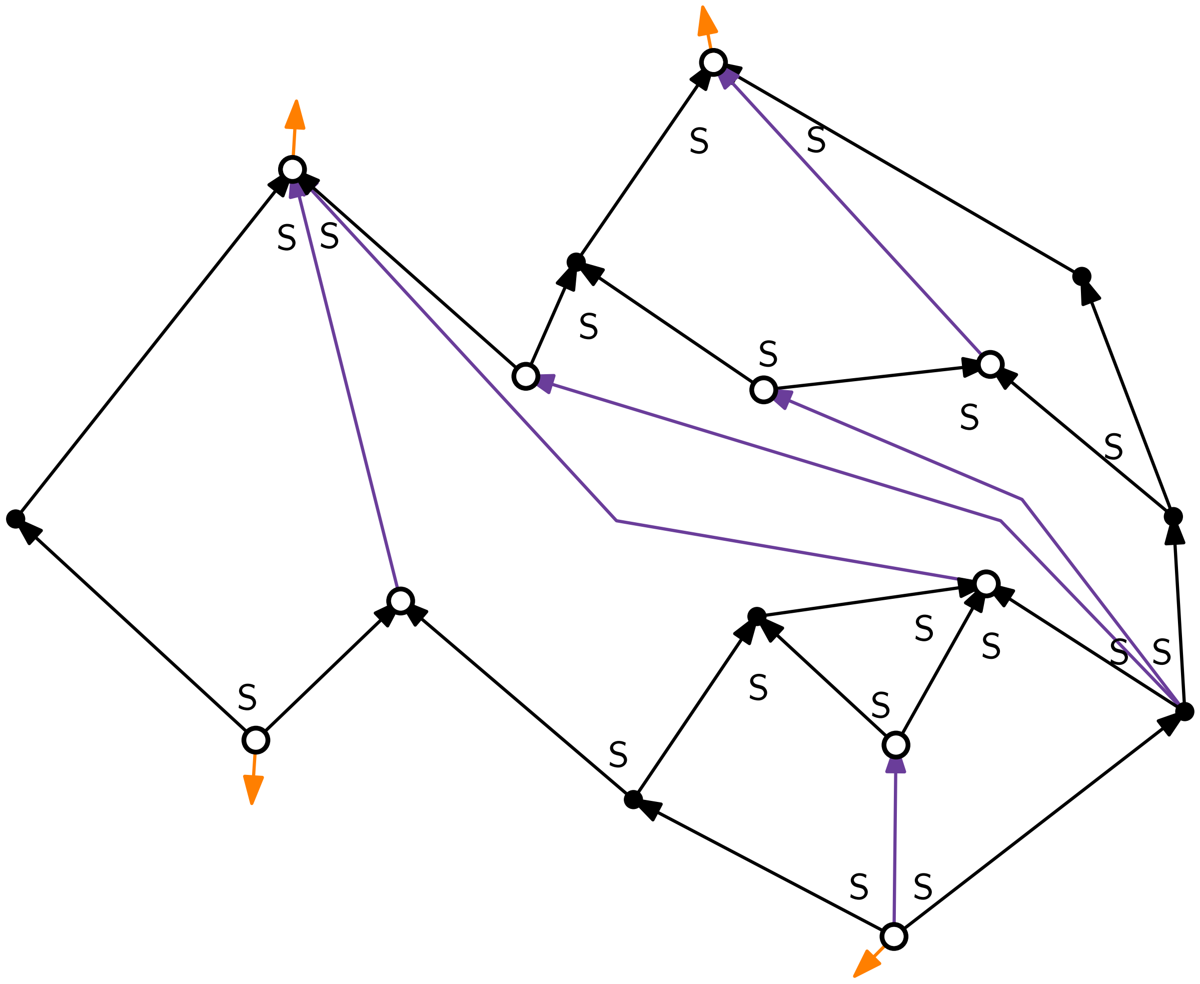
Refinement example



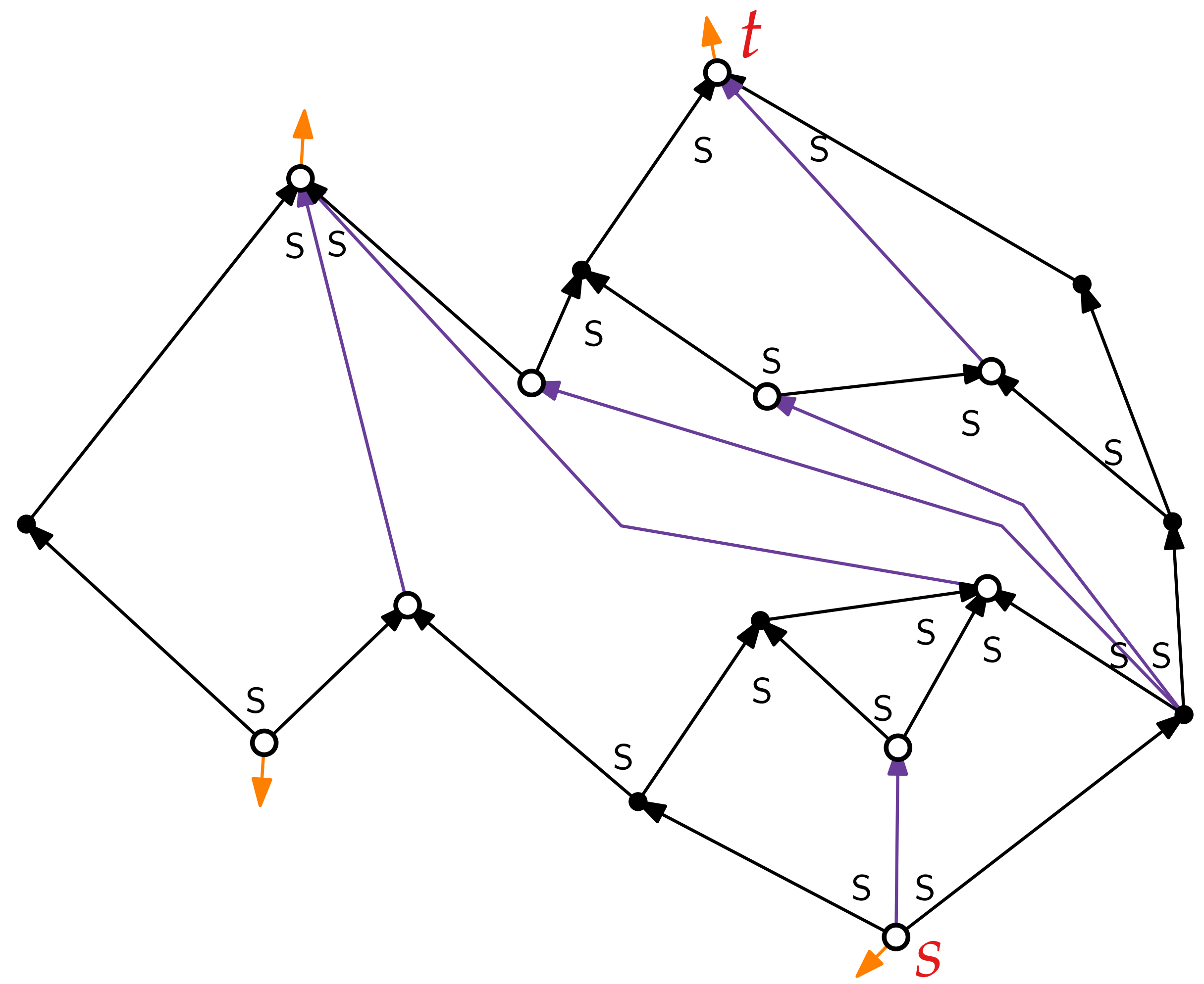
Refinement example



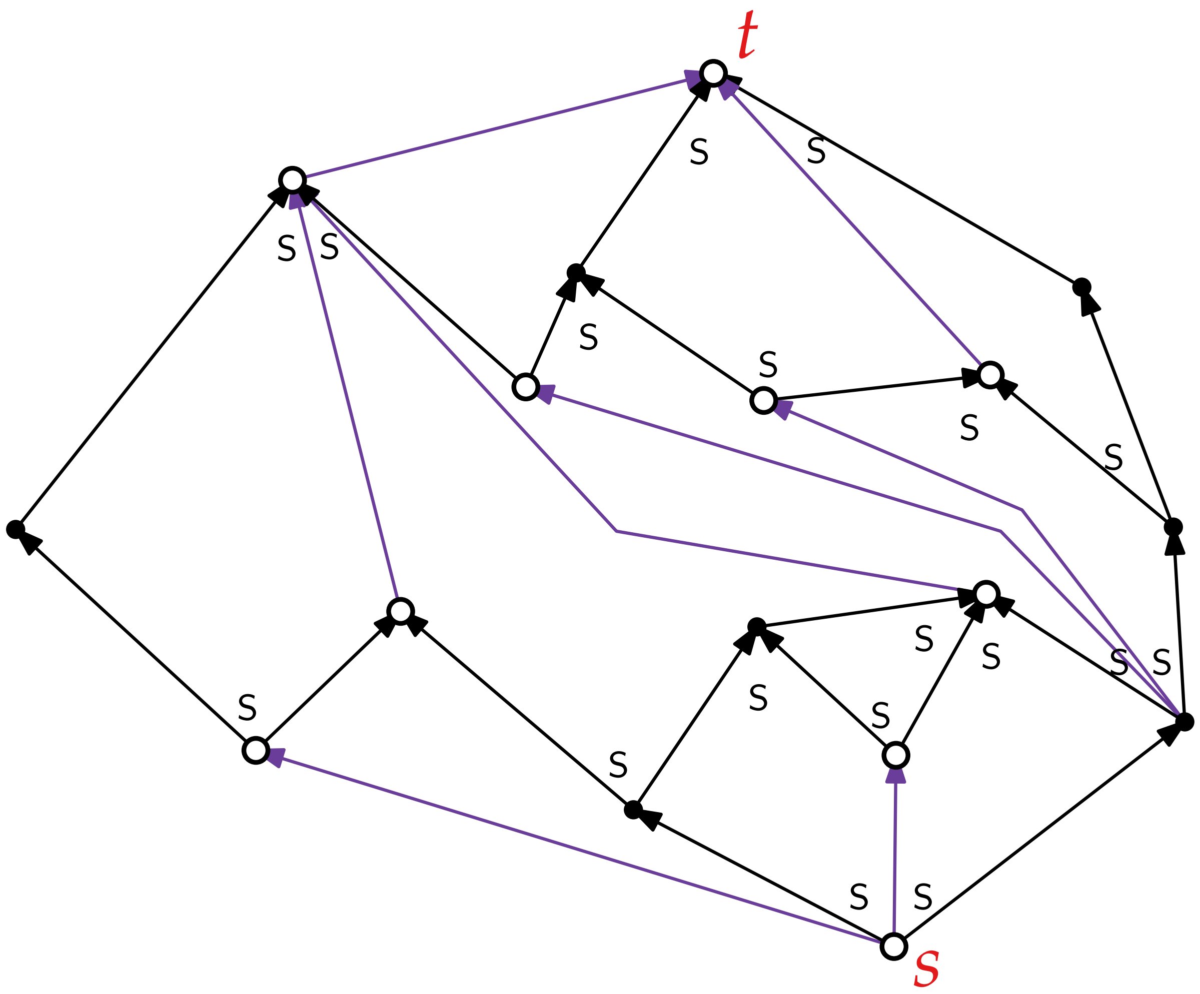
Refinement example



Refinement example



Refinement example



Result upward planarity test

Theorem 2. [Bertolazzi et al., 1994]
For a *combinatorially embedded* planar digraph G it can be tested in $\mathcal{O}(n^2)$ time whether it is upward planar.

Result upward planarity test

Theorem 2. [Bertolazzi et al., 1994]
For a *combinatorially embedded* planar digraph G it can be tested in $\mathcal{O}(n^2)$ time whether it is upward planar.

Proof.

- Test for bimodality.
- Test for a consistent assignment Φ (via flow network).

Result upward planarity test

Theorem 2. [Bertolazzi et al., 1994]
For a *combinatorially embedded* planar digraph G it can be tested in $\mathcal{O}(n^2)$ time whether it is upward planar.

Proof.

- Test for bimodality.
- Test for a consistent assignment Φ (via flow network).
- If G bimodal and Φ exists, refine G to plane st-digraph H.
- Draw H upward planar.
- Deleted edges added in refinement step.

Finding a consistent assignment

Idea.
Flow $(v, f) = 1$ from global source/sink v to the incident face f its large angle gets assigned to.

Finding a consistent assignment

Idea.
Flow $(v, f) = 1$ from global source/sink v to the incident face f its large angle gets assigned to.

Flow network.
$N_{F,f_0}(G) = ((W, E'); \ell; u; d)$

- $W =$
- $E' =$
- $\ell(e) =$
- $u(e) =$
- $d(p) =$

Finding a consistent assignment

Idea.

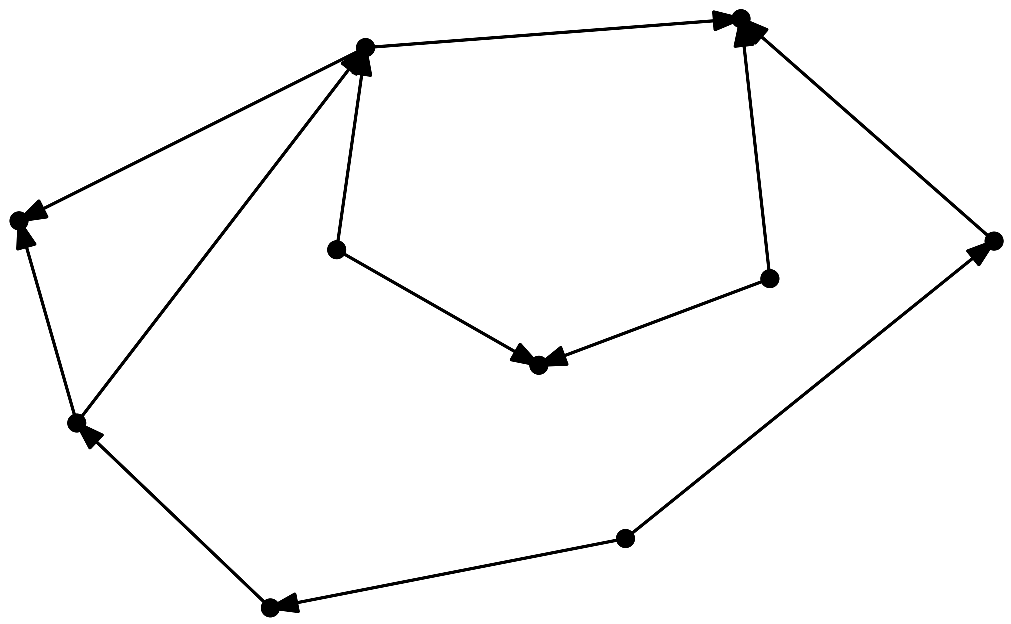
Flow $(v, f) = 1$ from global source/sink v to the incident face f its large angle gets assigned to.

Flow network.

$N_{F,f_0}(G) = ((W, E'); \ell; u; d)$

- $W =$
- $E' =$
- $\ell(e) =$
- $u(e) =$
- $d(p) =$

Example.



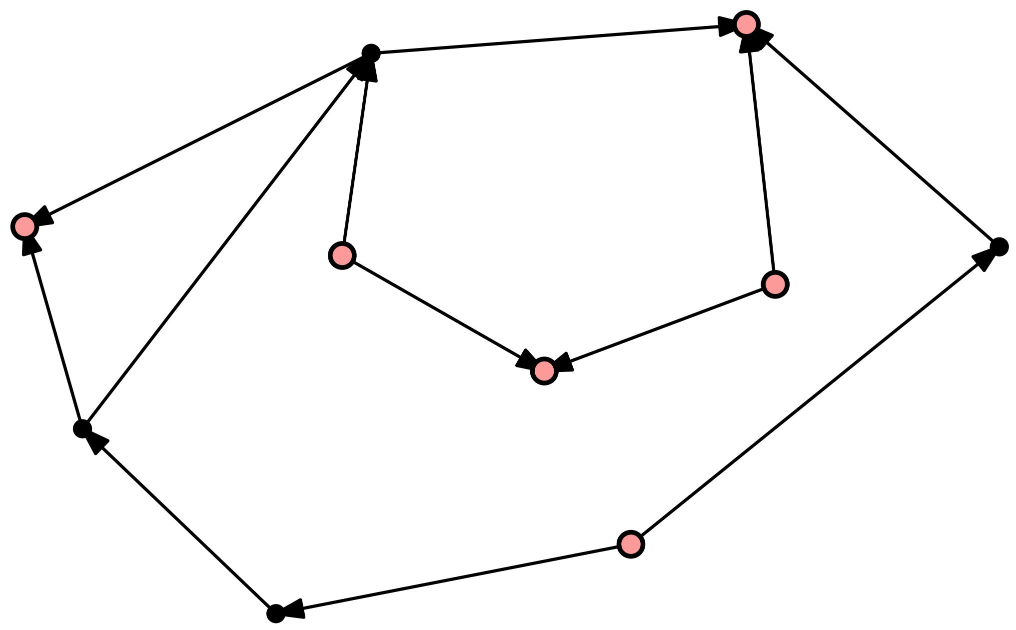
Finding a consistent assignment

Idea.
Flow $(v, f) = 1$ from global source/sink v to the incident face f its large angle gets assigned to.

Flow network.
$N_{F,f_0}(G) = ((W, E'); \ell; u; d)$

- $W = \{v \in V \mid v$ source or sink$\} \cup$
- $E' =$
- $\ell(e) =$
- $u(e) =$
- $d(p) =$

Example.



Finding a consistent assignment

Idea.

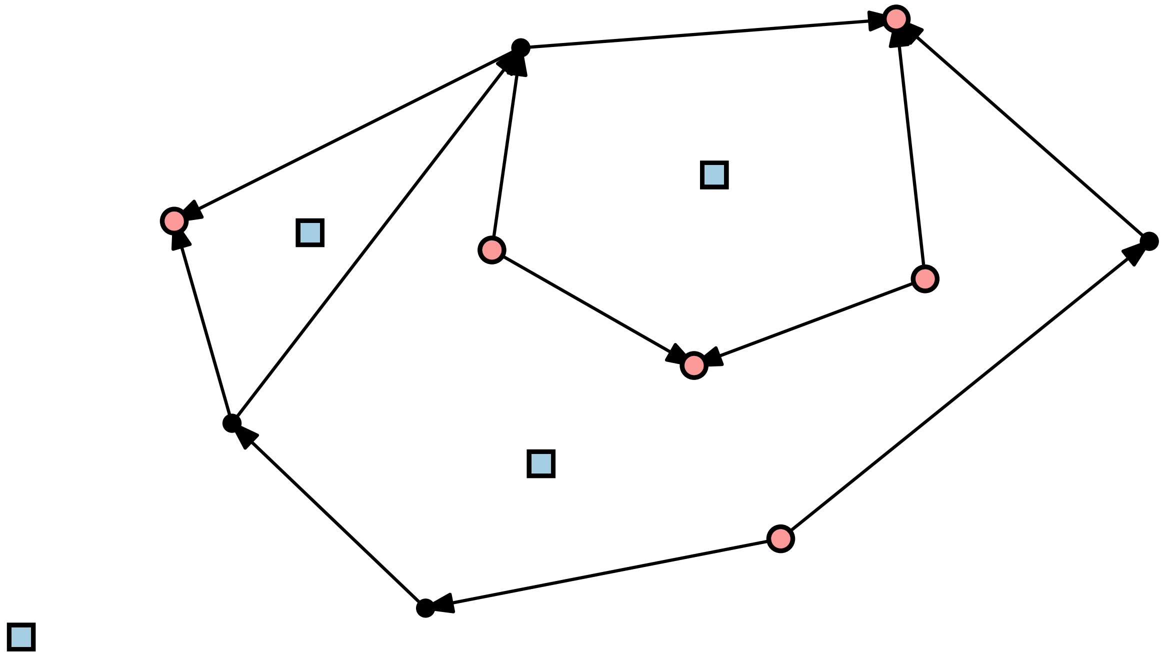
Flow $(v, f) = 1$ from global source/sink v to the incident face f its large angle gets assigned to.

Flow network.

$N_{F,f_0}(G) = ((W, E'); \ell; u; d)$

- $W = \{v \in V \mid v \text{ source or sink}\} \cup F$
- $E' =$
- $\ell(e) =$
- $u(e) =$
- $d(p) =$

Example.



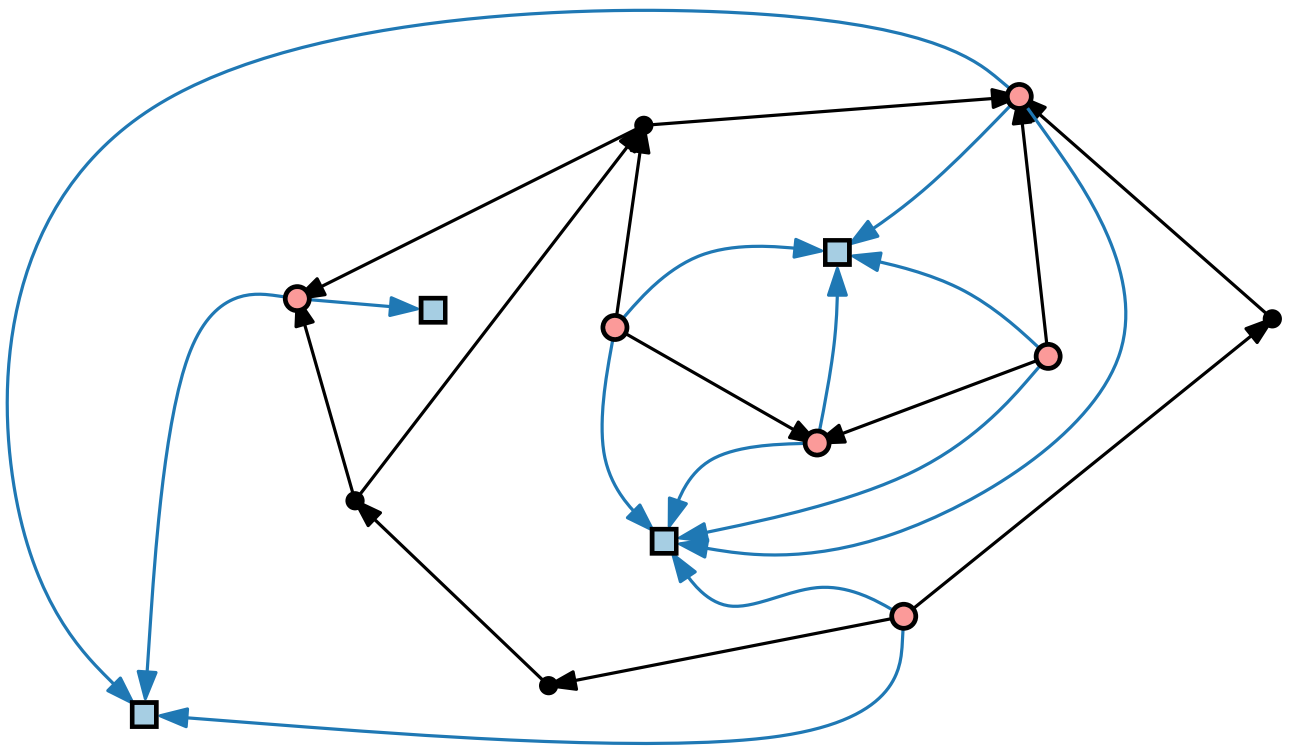
Finding a consistent assignment

Idea.
Flow $(v, f) = 1$ from global source/sink v to the incident face f its large angle gets assigned to.

Flow network.
$N_{F,f_0}(G) = ((W, E'); \ell; u; d)$

- $W = \{v \in V \mid v$ source or sink$\} \cup F$
- $E' = \{(v, f) \mid v$ incident to $f\}$
- $\ell(e) =$
- $u(e) =$
- $d(p) =$

Example.



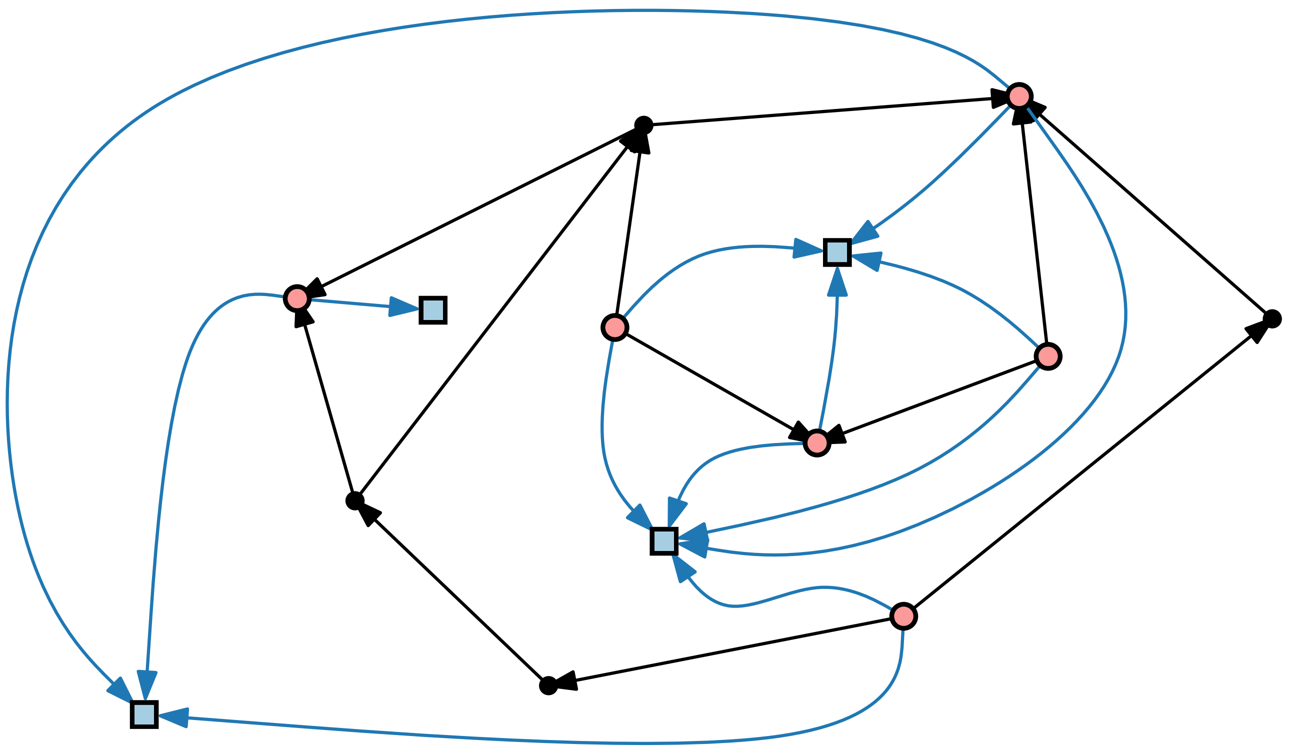
Finding a consistent assignment

Idea.
Flow $(v, f) = 1$ from global source/sink v to the incident face f its large angle gets assigned to.

Flow network.
$N_{F,f_0}(G) = ((W, E'); \ell; u; d)$

- $W = \{v \in V \mid v \text{ source or sink}\} \cup F$
- $E' = \{(v, f) \mid v \text{ incident to } f\}$
- $\ell(e) = 0 \ \forall e \in E'$
- $u(e) = 1 \ \forall e \in E'$
- $d(p) =$

Example.



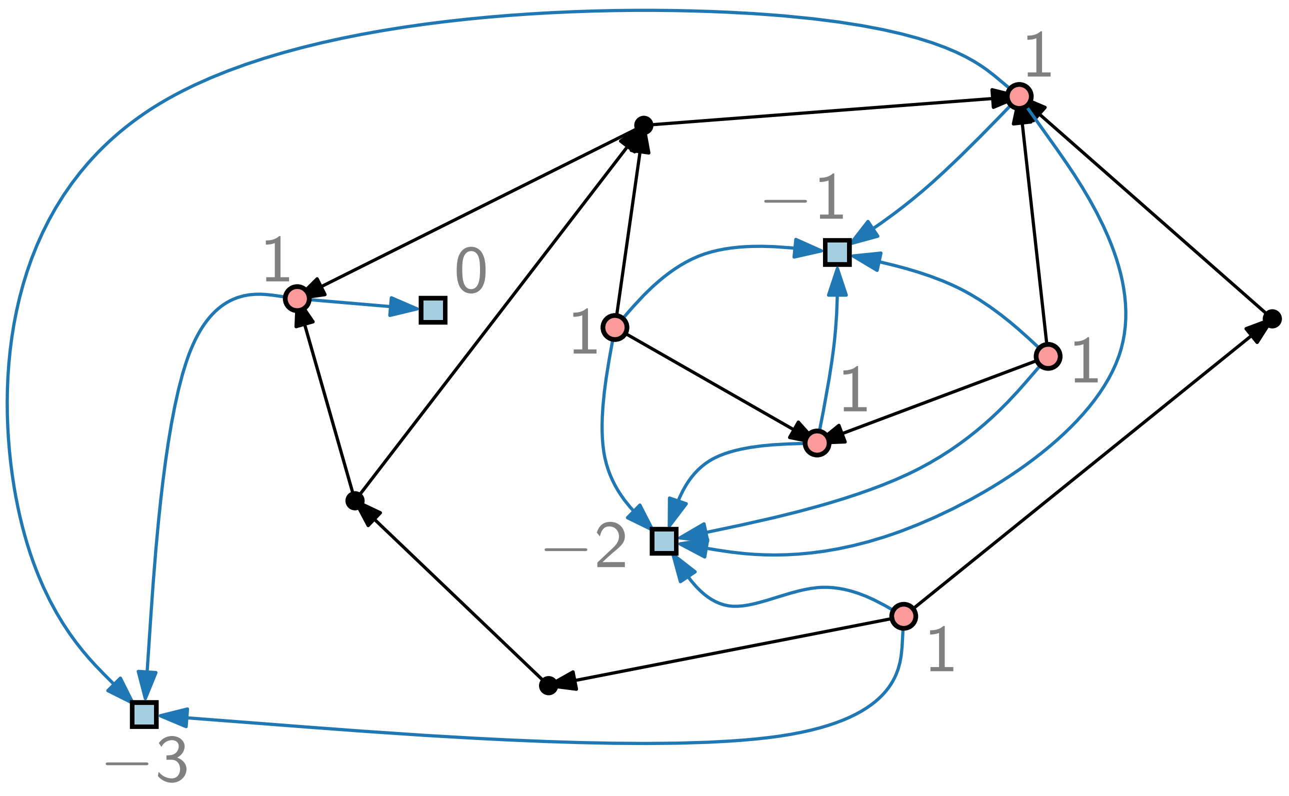
Finding a consistent assignment

Idea.
Flow $(v, f) = 1$ from global source/sink v to the incident face f its large angle gets assigned to.

Flow network.
$N_{F,f_0}(G) = ((W, E'); \ell; u; d)$

- $W = \{v \in V \mid v \text{ source or sink}\} \cup F$
- $E' = \{(v, f) \mid v \text{ incident to } f\}$
- $\ell(e) = 0 \ \forall e \in E'$
- $u(e) = 1 \ \forall e \in E'$
- $d(p) = \begin{cases} 1 & \forall p \in W \cap V \\ -(A(p) - 1) & \forall p \in F \setminus \{f_0\} \\ -(A(p) + 1) & p = f_0 \end{cases}$

Example.



Finding a consistent assignment

Idea.

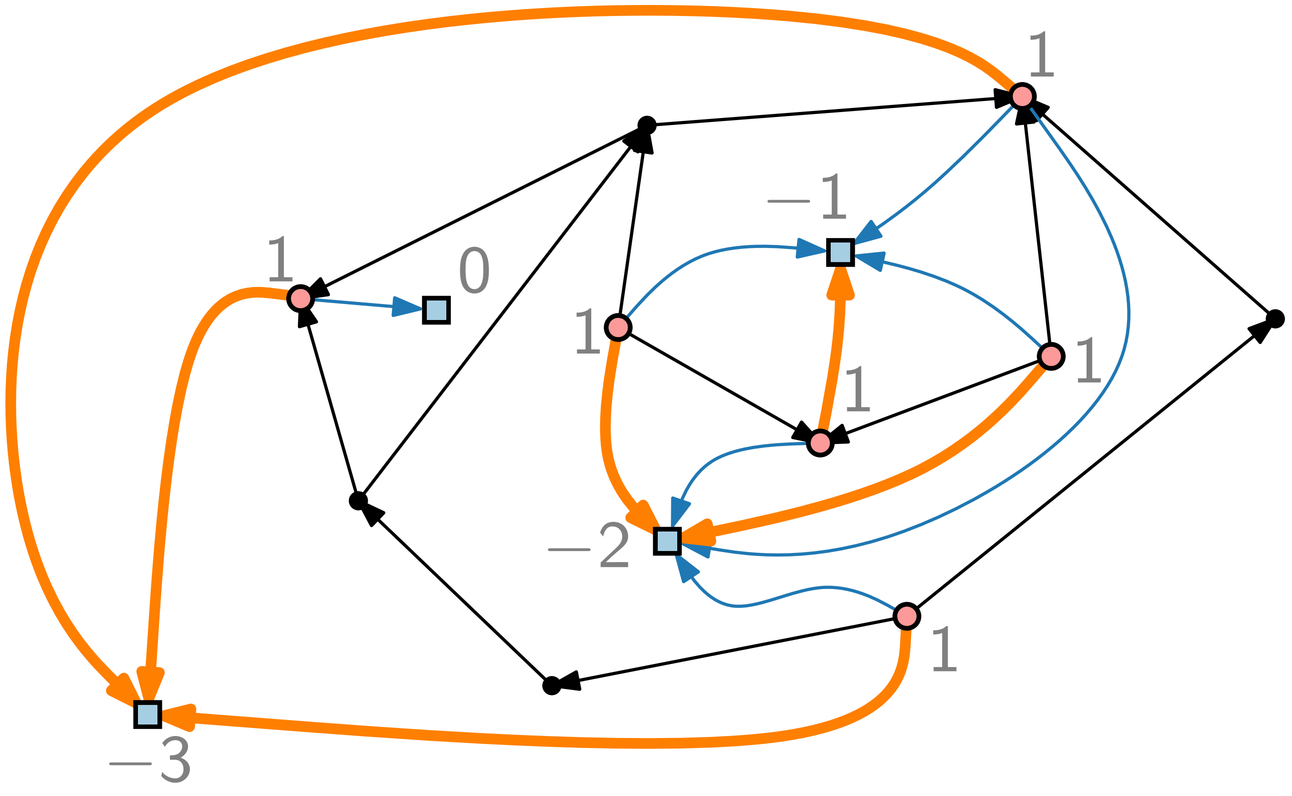
Flow $(v, f) = 1$ from global source/sink v to the incident face f its large angle gets assigned to.

Flow network.

$N_{F,f_0}(G) = ((W, E'); \ell; u; d)$

- $W = \{v \in V \mid v \text{ source or sink}\} \cup F$
- $E' = \{(v, f) \mid v \text{ incident to } f\}$
- $\ell(e) = 0 \ \forall e \in E'$
- $u(e) = 1 \ \forall e \in E'$
- $d(p) = \begin{cases} 1 & \forall p \in W \cap V \\ -(A(p) - 1) & \forall p \in F \setminus \{f_0\} \\ -(A(p) + 1) & p = f_0 \end{cases}$

Example.



Discussion

- There exist fixed-parameter tractable algorithms to test upward planarity of general digraphs with the parameter being the number of triconnected components.
  [Healy, Lynch 2005, Didimo et al. 2009]

Discussion

- There exist fixed-parameter tractable algorithms to test upward planarity of general digraphs with the parameter being the number of triconnected components.
  [Healy, Lynch 2005, Didimo et al. 2009]
- Finding assignment in Theorem 2 can be sped up to $\mathcal{O}(n + r^{1.5})$ where $r = \#$ sources/sinks.
  [Abbasi, Healy, Rextin 2010]

Discussion

- There exist fixed-parameter tractable algorithms to test upward planarity of general digraphs with the parameter being the number of triconnected components.
  [Healy, Lynch 2005, Didimo et al. 2009]
- Finding assignment in Theorem 2 can be sped up to $\mathcal{O}(n + r^{1.5})$ where $r = \#$ sources/sinks.
  [Abbasi, Healy, Rextin 2010]
- Many related concepts have been studied: quasi-planarity, upward drawings of mixed graphs, upward planarity on cyclinder/torus, ...

Literature

- [GD Ch. 6] for detailed explanation

Orginal papers referenced:

- [Kelly '87] Fundamentals of Planar Ordered Sets
- [Di Battista, Tamassia '88] Algorithms for Plane Representations of Acyclic Digraphs
- [Garg, Tamassia '95] On the Computational Complexity of Upward and Rectilinear Planarity Testing
- [Hutton, Lubiw '96] Upward Planar Drawing of Single-Source Acyclic Digraphs
- [Bertolazzi, Di Battista, Mannino, Tamassia '94] Upward Drawings of Triconnected Digraphs
- [Healy, Lynch '05] Building Blocks of Upward Planar Digraphs
- [Didimo, Giardano, Liotta '09] Upward Spirality and Upward Planarity Testing
- [Abbasi, Healy, Rextin '10] Improving the running time of embedded upward planarity testing