

Algorithmen zur Visualisierung von Graphen

Flussmethoden und Einbettungsprobleme

Vorlesung im Wintersemester 2011/2012

Ignaz Rutter

16.11.2011

Winkelauflösung in geradlinigen Layouts

Konstruktion des Flussnetzwerks

■ $W := V \cup \mathcal{F}$

Konstruktion des Flussnetzwerks

- $W := V \cup \mathcal{F}$
- $A := \{(v, f) \in V \times \mathcal{F} : v \text{ inzident zu } f\}$

Konstruktion des Flussnetzwerks

- $W := V \cup \mathcal{F}$
- $A := \{(v, f) \in V \times \mathcal{F} : v \text{ inzident zu } f\}$
- $\ell(a) = 0 \quad \forall a \in A$

Konstruktion des Flussnetzwerks

- $W := V \cup \mathcal{F}$
- $A := \{(v, f) \in V \times \mathcal{F} : v \text{ inzident zu } f\}$
- $\ell(a) = 0 \quad \forall a \in A$
- $u(a) = 2\pi \quad \forall a \in A$

Konstruktion des Flussnetzwerks

- $W := V \cup \mathcal{F}$
- $A := \{(v, f) \in V \times \mathcal{F} : v \text{ inzident zu } f\}$
- $l(a) = 0 \quad \forall a \in A$
- $u(a) = 2\pi \quad \forall a \in A$
- $b(v) = 2\pi \quad \forall v \in V$

Konstruktion des Flussnetzwerks

- $W := V \cup \mathcal{F}$
- $A := \{(v, f) \in V \times \mathcal{F} : v \text{ inzident zu } f\}$
- $\ell(a) = 0 \quad \forall a \in A$
- $u(a) = 2\pi \quad \forall a \in A$
- $b(v) = 2\pi \quad \forall v \in V$
- $b(f) = \begin{cases} -(\deg(f) - 2)\pi & \text{falls } f \neq f_0 \\ -(\deg(f) + 2)\pi & \text{sonst} \end{cases}$

- $W := V \cup \mathcal{F}$
- $A := \{(v, f) \in V \times \mathcal{F} : v \text{ inzident zu } f\}$
- $\ell(a) = 0 \quad \forall a \in A$
- $u(a) = 2\pi \quad \forall a \in A$
- $b(v) = 2\pi \quad \forall v \in V$
- $b(f) = \begin{cases} -(\deg(f) - 2)\pi & \text{falls } f \neq f_0 \\ -(\deg(f) + 2)\pi & \text{sonst} \end{cases}$

Zuweisung von Winkelwerten liefert:

1. Knotenbed.: $\forall v \in V : \sum_{f \sim v} x(v, f) = 2\pi$
2. Facettenbed.: $\forall f \in \mathcal{F} : \sum_{v \sim f} x(v, f) = (\deg(f) \mp 2)\pi$

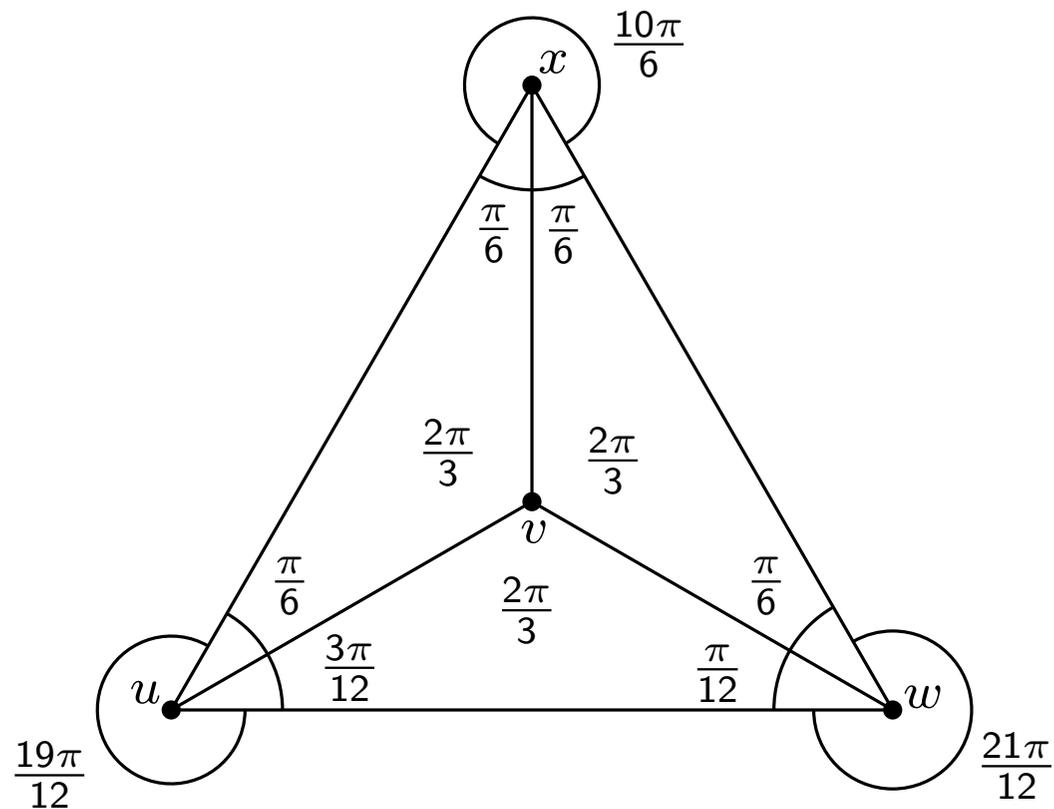
- $W := V \cup \mathcal{F}$
- $A := \{(v, f) \in V \times \mathcal{F} : v \text{ inzident zu } f\}$
- $\ell(a) = 0 \quad \forall a \in A$
- $u(a) = 2\pi \quad \forall a \in A$
- $b(v) = 2\pi \quad \forall v \in V$
- $b(f) = \begin{cases} -(\deg(f) - 2)\pi & \text{falls } f \neq f_0 \\ -(\deg(f) + 2)\pi & \text{sonst} \end{cases}$

Zuweisung von Winkelwerten liefert:

1. Knotenbed.: $\forall v \in V : \sum_{f \sim v} x(v, f) = 2\pi$
2. Facettenbed.: $\forall f \in \mathcal{F} : \sum_{v \sim f} x(v, f) = (\deg(f) \mp 2)\pi$

1. und 2. erfüllt: Zuweisung *lokal konsistent*

Gegenbeispiel Lokalkonsistenz



Satz (Di Battista & Vismara '93)

Gegeben planarer Dreiecksgraph mit kombinatorischer Einbettung und Winkelzuweisung, dann gilt:

Es existiert eine geradlienige Realisierung Einbettung mit f_0 konvex



1. \sum Knotenwinkel $= 2\pi$
2. \sum Facettenwinkel $= \pi$
3. $\forall v \approx f_0$: im Rad R_d^v gilt: $\prod_{i=1}^d \frac{\sin \alpha_i}{\sin \beta_i} = 1$
4. $\sum_{v \sim f_0} x(v, f_0) \leq \pi$

Satz (Di Battista & Vismara '93)

Gegeben planarer Dreiecksgraph mit kombinatorischer Einbettung und Winkelzuweisung, dann gilt:

Es existiert eine geradlienige Realisierung Einbettung mit f_0 konvex



1. $\sum \text{Knotenwinkel} = 2\pi$

2. $\sum \text{Facettenwinkel} = \pi$

3. $\forall v \approx f_0$: im Rad R_d^v gilt: $\prod_{i=1}^d \frac{\sin \alpha_i}{\sin \beta_i} = 1$

4. $\sum_{v \sim f_0} x(v, f_0) \leq \pi$

Konstruktion
in $O(n)$

Satz (Di Battista & Vismara '93)

Gegeben planarer Dreiecksgraph mit kombinatorischer Einbettung und Winkelzuweisung, dann gilt:

Es existiert eine geradlienige Realisierung Einbettung mit f_0 konvex



1. $\sum \text{Knotenwinkel} = 2\pi$

2. $\sum \text{Facettenwinkel} = \pi$

3. $\forall v \approx f_0$: im Rad R_d^v gilt: $\prod_{i=1}^d \frac{\sin \alpha_i}{\sin \beta_i} = 1$

4. $\sum_{v \sim f_0} x(v, f_0) \leq \pi$

} Konstruktion
in $O(n)$
3. + 4. nicht
durch Fluss
erfüllt

Satz (Malitz & Papkostas '92)

In einem triangulierten, planar eingebetteten Graph $G = (V, E)$, gibt es im zugehörigen Flussnetzwerk $N(G)$ einen Fluss x , dessen minimaler Kantenwert $x_{\min} \geq \frac{\pi}{3 \cdot (\deg_{\max}(G) - 1)}$ ist, wobei \deg_{\max} der maximale Grad eines Knoten in G ist.

Ergibt untere Schranke für obere Schranke der Winkelauflösung

Satz (Tamassia '87)

G Maxgrad-4-Graph mit fester planarer Einbettung

Orthogonale Zeichnung von G mit minimaler Anzahl an Knicken kann effizient berechnet werden.

Satz (Tamassia '87)

G Maxgrad-4-Graph mit fester planarer Einbettung

Orthogonale Zeichnung von G mit minimaler Anzahl an Knicken kann effizient berechnet werden.

Satz (Di Battista, Tamassia '88)

Gegeben ein gerichteter azyklischer Graph $D = (V, A)$ mit Einbettung \mathcal{F}, f_0 . Es kann effizient getestet werden, ob D mit ein Einbettung \mathcal{F} und äußerer Facette f_0 aufwärtsplanar ist. Ein entsprechendes Layout kann ggf. effizient konstruiert werden.

Satz (Garg, Tamassia '01)

G Maxgrad-4-Graph mit variabler Einbettung

Entscheiden, ob G orthogonale Zeichnung ohne Knicke besitzt ist NP-schwer.

Minimierung schwer auf $O(n^{1-\varepsilon})$ zu approximieren.

Satz (Garg, Tamassia '01)

$D = (V, A)$ planarer, gerichteter, azyklischer Graph mit variabler Einbettung.

Entscheiden, ob D eine aufwärtsplanare Zeichnung besitzt ist NP-schwer.

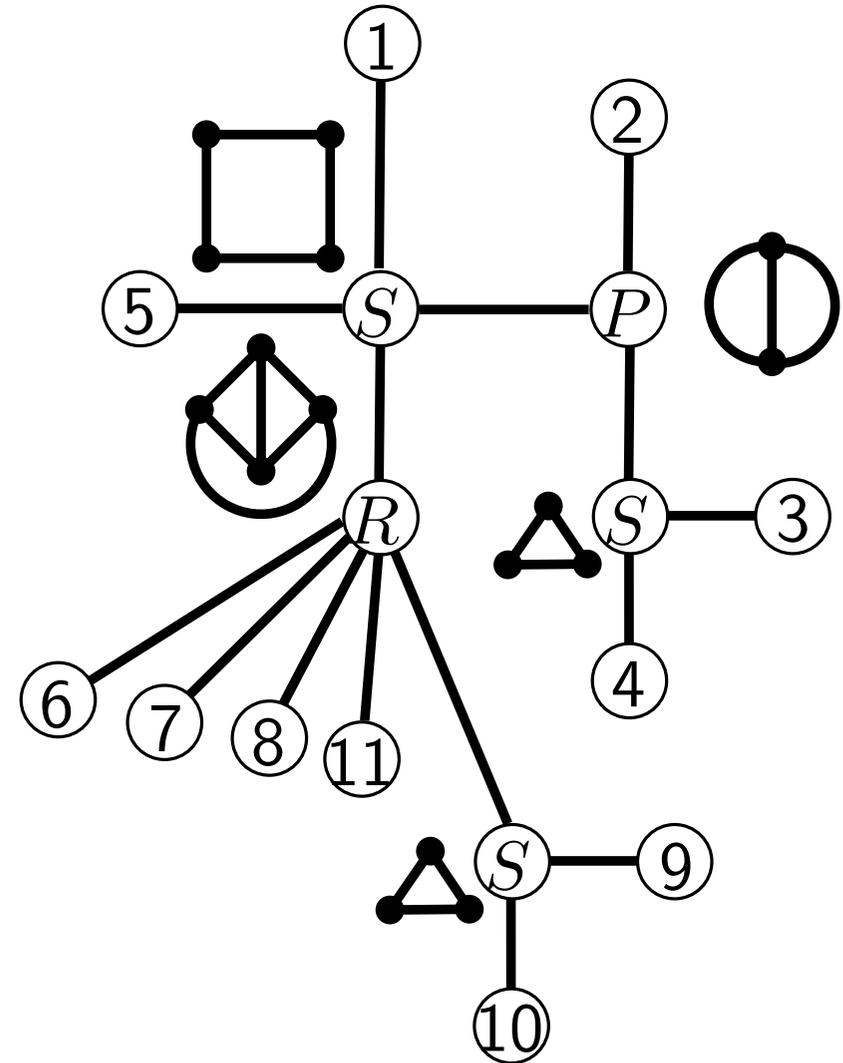
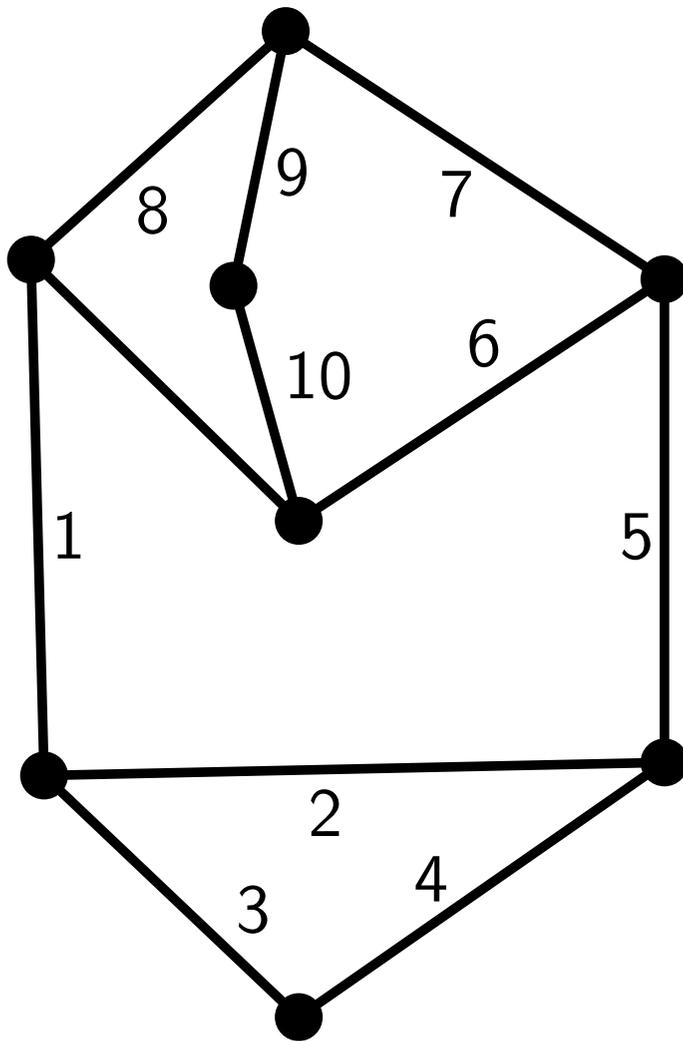
Satz (Biedl, Kant '94)

Jeder planare maxgrad-4 Graph außer dem Oktaedergraph besitzt orthogonale, planare Zeichnung mit höchstens zwei Knicke pro Kante.

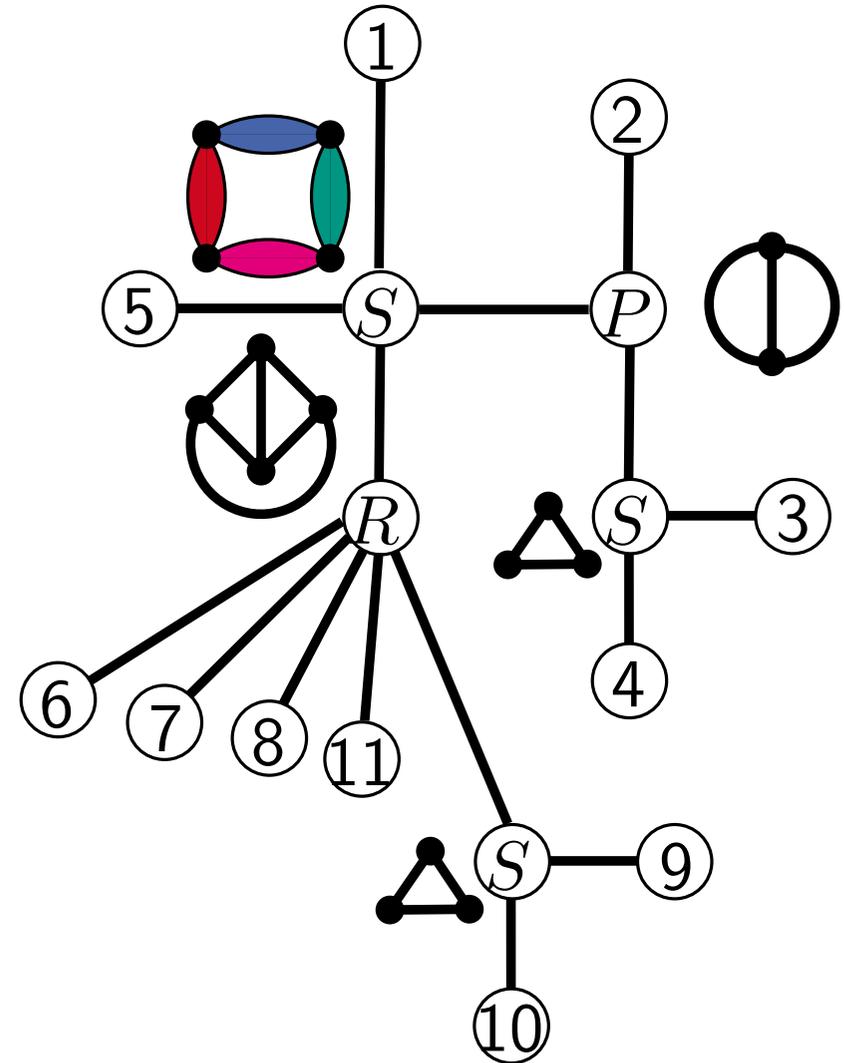
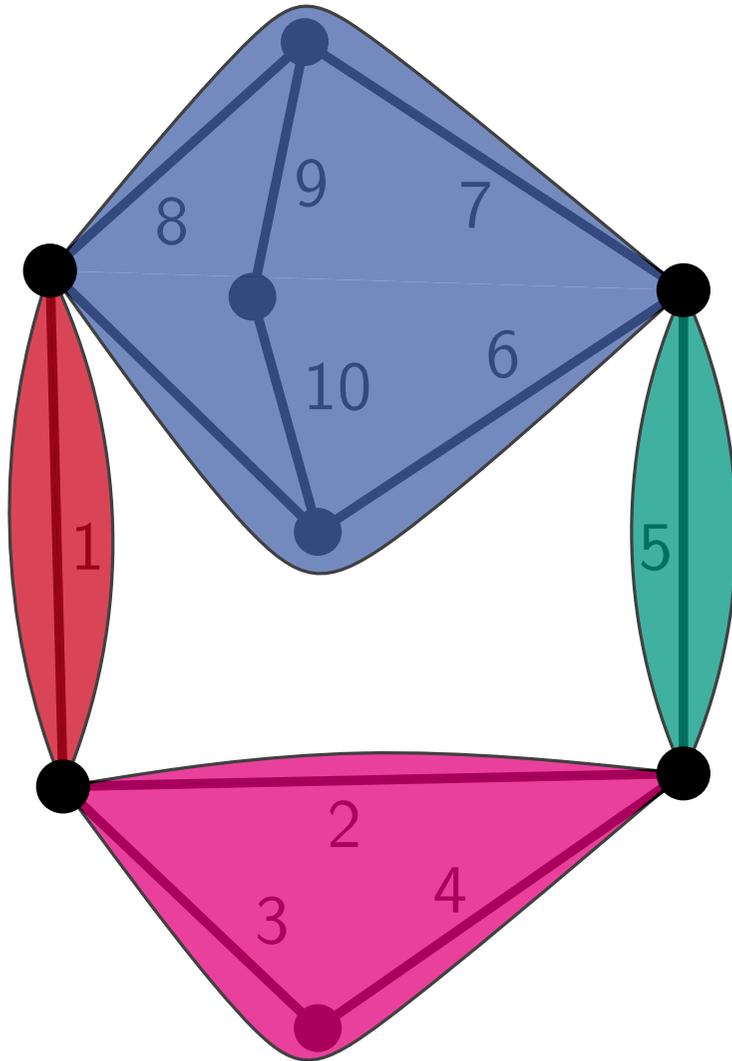
Variable Einbettungen: SPQR-Baum

(Eingeführt von Di Battista, Tamassia '96)

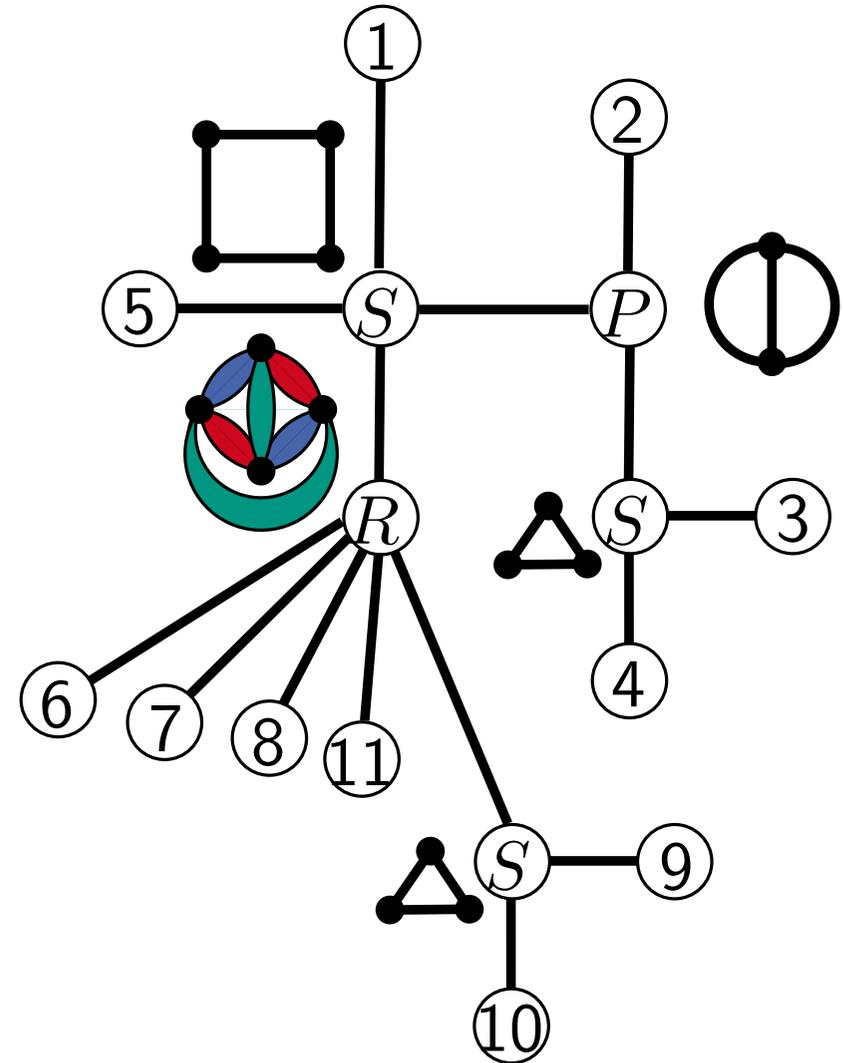
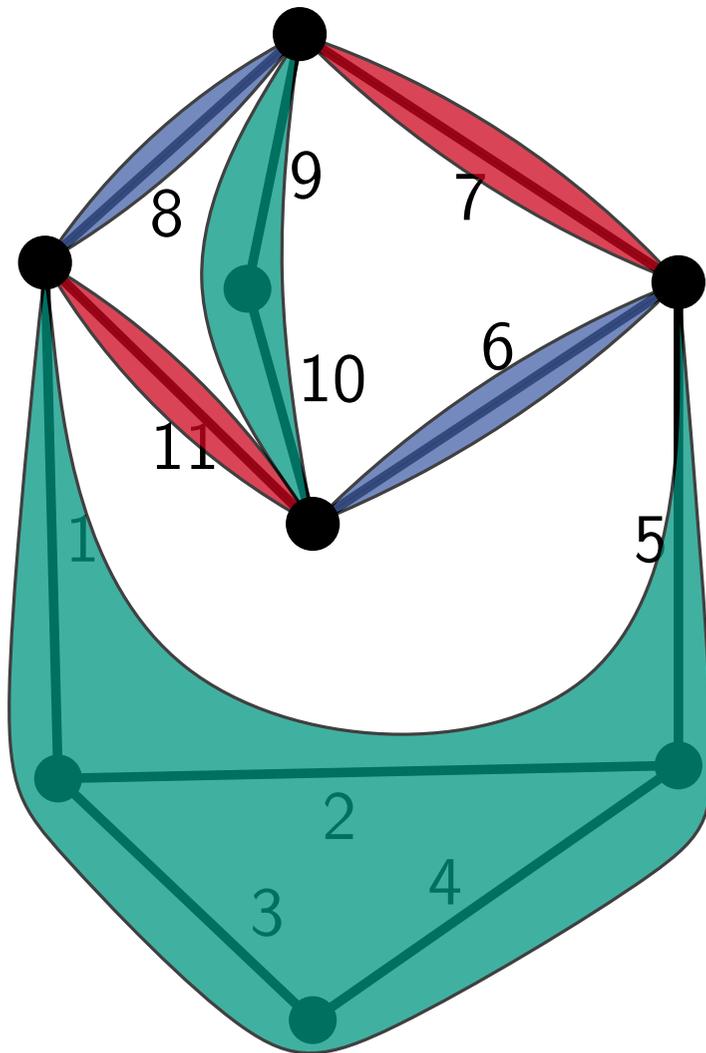
SPQR-Baum (Beispiel)



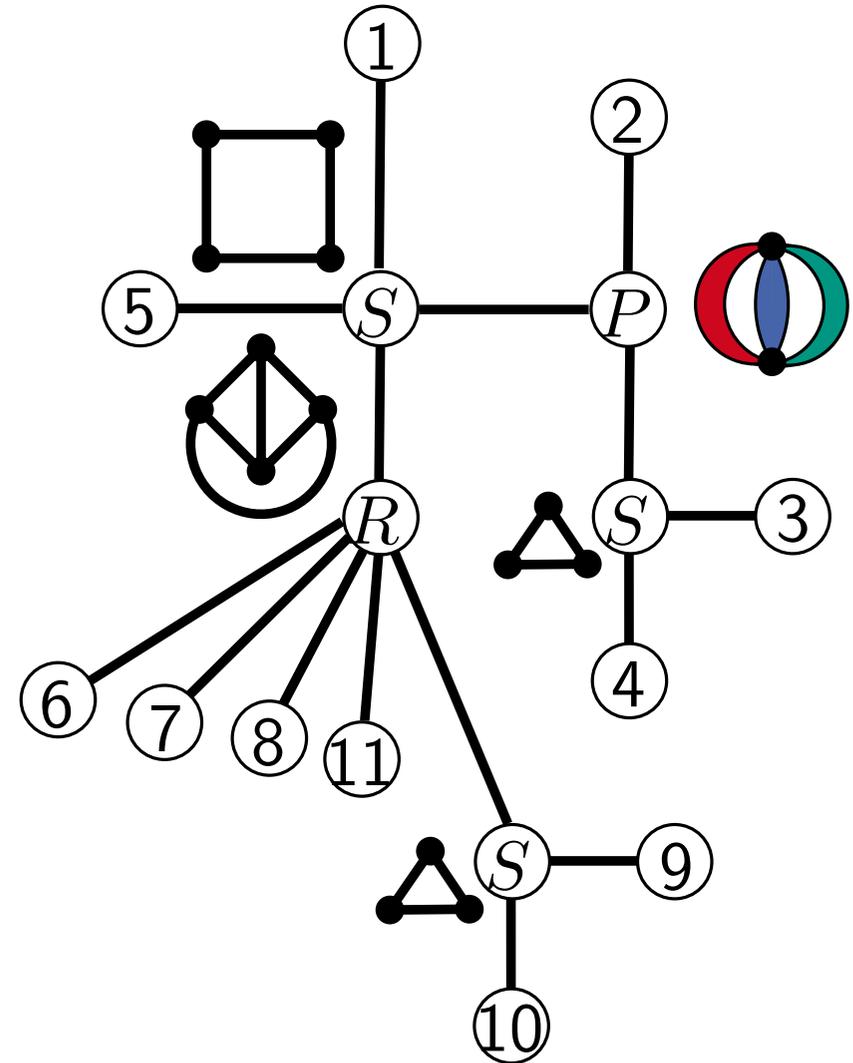
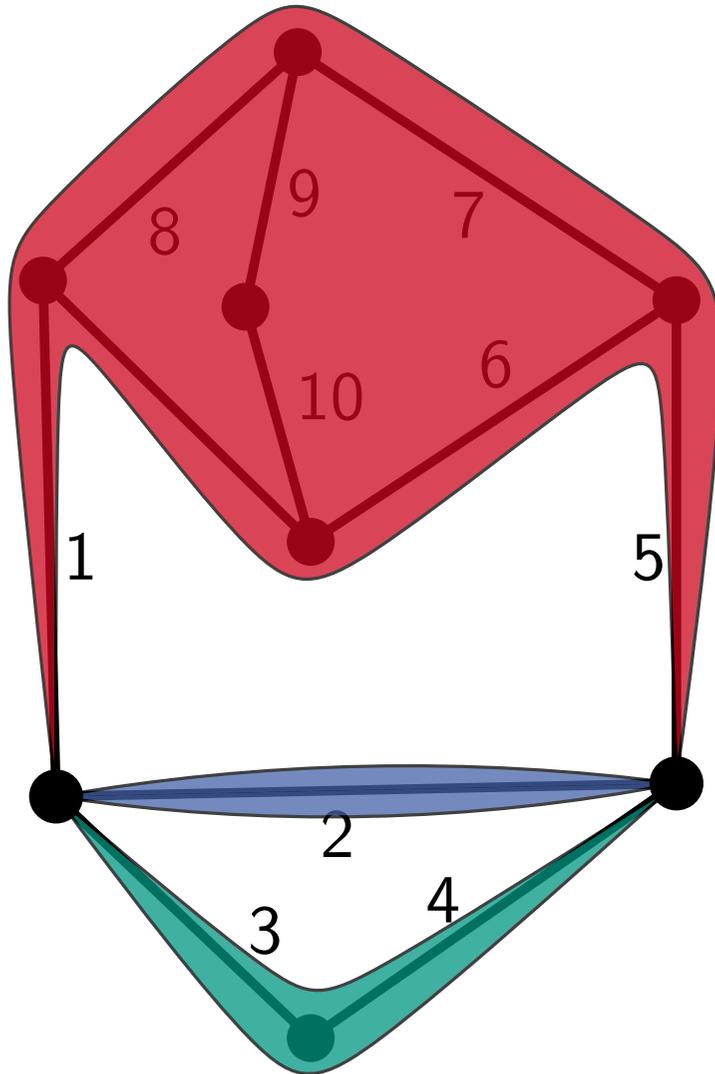
SPQR-Baum (Beispiel)



SPQR-Baum (Beispiel)



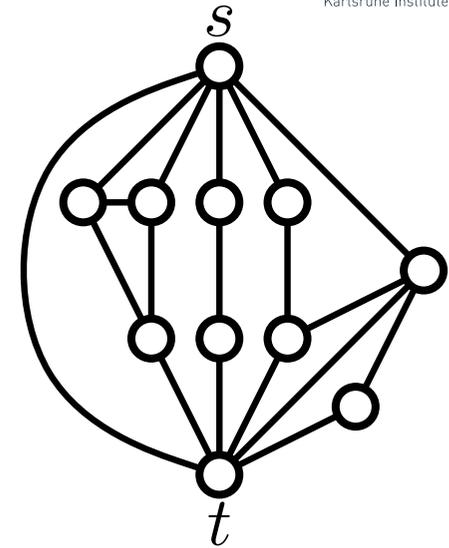
SPQR-Baum (Beispiel)



Split Pairs

Split pair: $\{s, t\}$ mit:

- st Kante oder
- $G - \{s, t\}$ nicht zusammenhängend



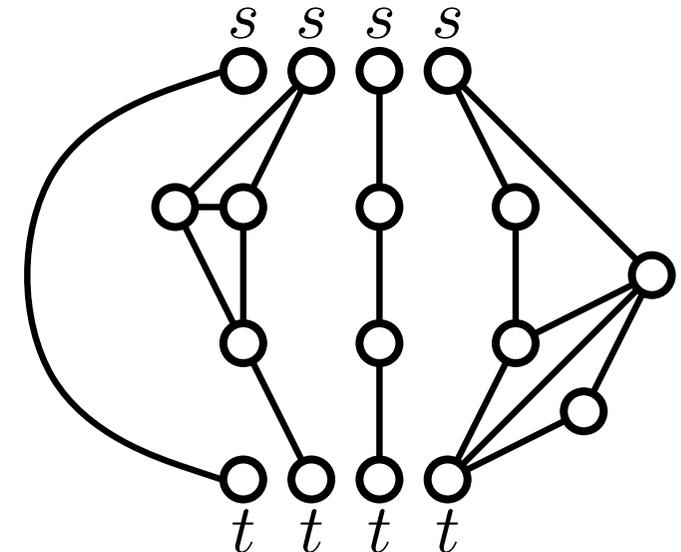
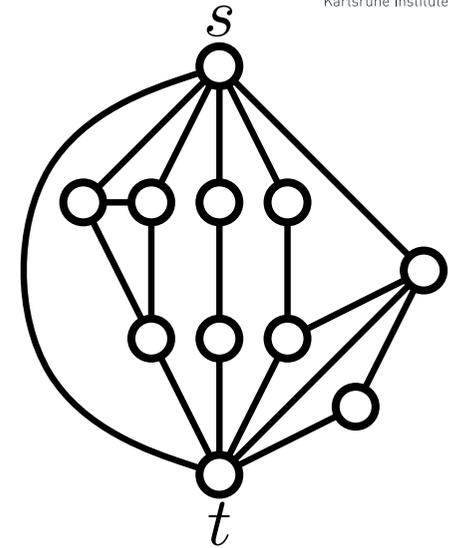
Split Pairs

Split pair: $\{s, t\}$ mit:

- st Kante oder
- $G - \{s, t\}$ nicht zusammenhängend

Maximale Split-Komponente eines Split Pairs $\{s, t\}$:

- Kante st
- Maximaler Teilgraph von G , für den $\{s, t\}$ kein split pair ist.



Split Pairs

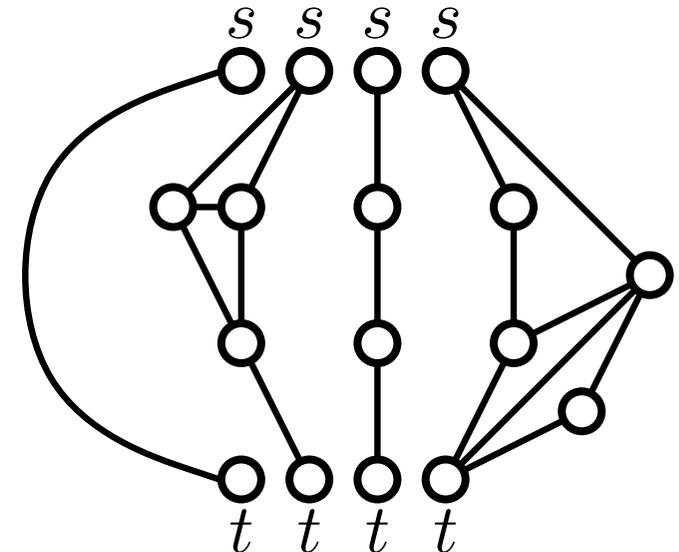
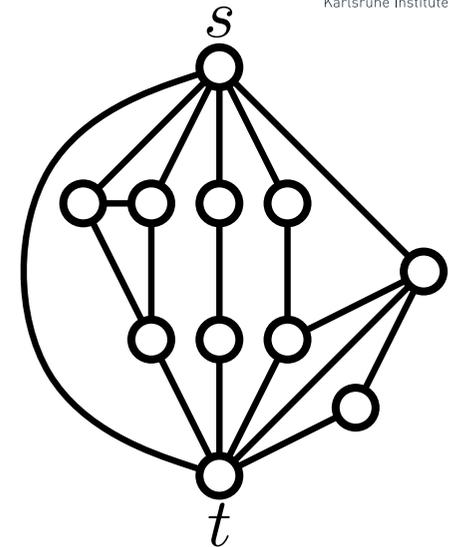
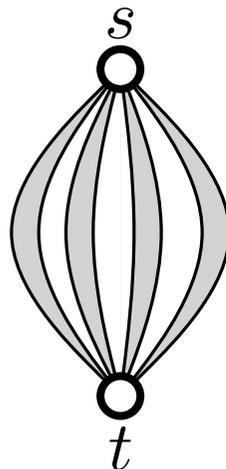
Split pair: $\{s, t\}$ mit:

- st Kante oder
- $G - \{s, t\}$ nicht zusammenhängend

Maximale Split-Komponente eines Split Pairs $\{s, t\}$:

- Kante st
- Maximaler Teilgraph von G , für den $\{s, t\}$ kein split pair ist.

Skizze von G bezüglich Split pair $\{s, t\}$:



SPQR-Baum, Fakten:

SPQR-Baum für 2-fach zusammenhängenden Graph G :

- Baum \mathcal{T}
- Blätter entsprechen bijektiv den Kanten von G
- Jeder Knoten μ von \mathcal{T} ist mit Multigraph $\text{skel}(\mu)$ assoziiert
Skelett von μ
- Blätter von \mathcal{T} heißen Q-Knoten

SPQR-Baum, Fakten:

SPQR-Baum für 2-fach zusammenhängenden Graph G :

- Baum \mathcal{T}
- Blätter entsprechen bijektiv den Kanten von G
- Jeder Knoten μ von \mathcal{T} ist mit Multigraph $\text{skel}(\mu)$ assoziiert
Skelett von μ
- Blätter von \mathcal{T} heißen Q-Knoten

Skelette von Knoten:

- Kreis, Menge paralleler Kanten oder 3-fach zshg. Graph
- Knoten von $\text{skel}(\mu)$ sind Knoten von G
- Kante uv in $\text{skel}(\mu)$ besagt: $\{u, v\}$ ist Split Pair in G
- Kanten sind entweder **virtuelle Kanten** oder **echte Kanten**
(Kanten aus G)
- Kanten in $\text{skel}(\mu)$ entsprechen Nachbarn von μ in \mathcal{T}

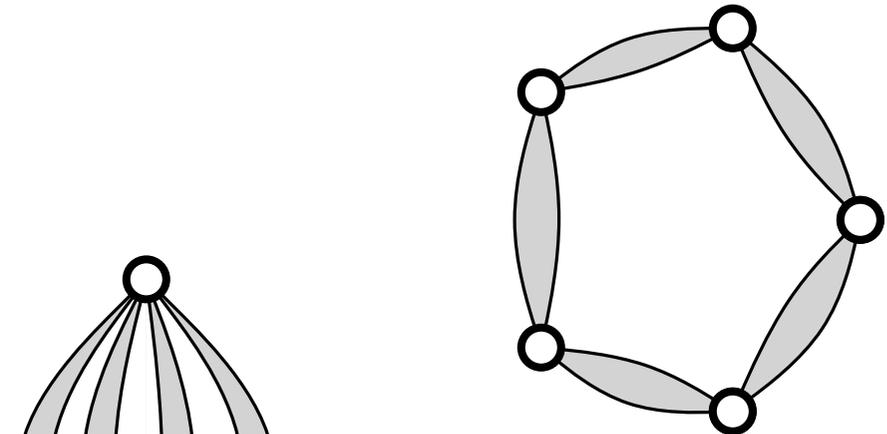
SPQR-Baum, Knoten und Skelette

Knoten-Art

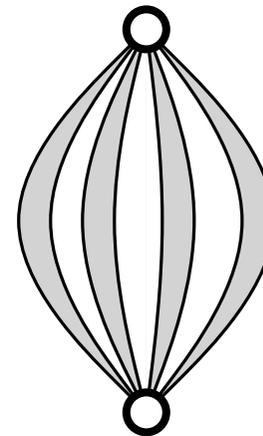
Q-Knoten (Blätter)



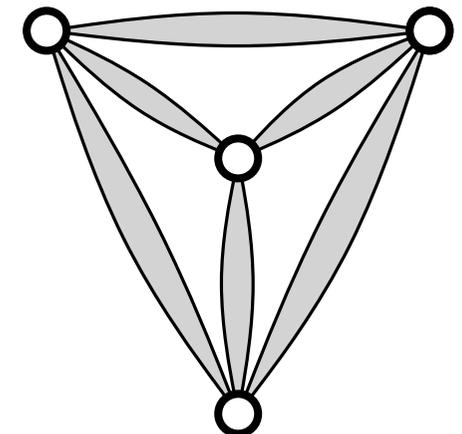
S-Knoten (Serie)



P-Knoten (Parallel)



R-Knoten (Rigid)



Konstruktion des SPQR-Baums

Wähle **Referenzkante** $e = u'v'$ von G

Zu jedem Zeitpunkt:

Geg.: Split-Komponente G^* , Knotenpaar $\{u, v\}$ und Knoten ν

Ges.: Füge Knoten μ , der G^* entspricht zu \mathcal{T} als Kind von ν hinzu

Eingangs:

$G^* := G - e$, $\{u, v\} := \{u', v'\}$, ν ist Q-Knoten, der e repräsentiert.

Konstruktion des SPQR-Baums

Wähle **Referenzkante** $e = u'v'$ von G

Zu jedem Zeitpunkt:

Geg.: Split-Komponente G^* , Knotenpaar $\{u, v\}$ und Knoten ν

Ges.: Füge Knoten μ , der G^* entspricht zu \mathcal{T} als Kind von ν hinzu

Eingangs:

$G^* := G - e$, $\{u, v\} := \{u', v'\}$, ν ist Q-Knoten, der e repräsentiert.

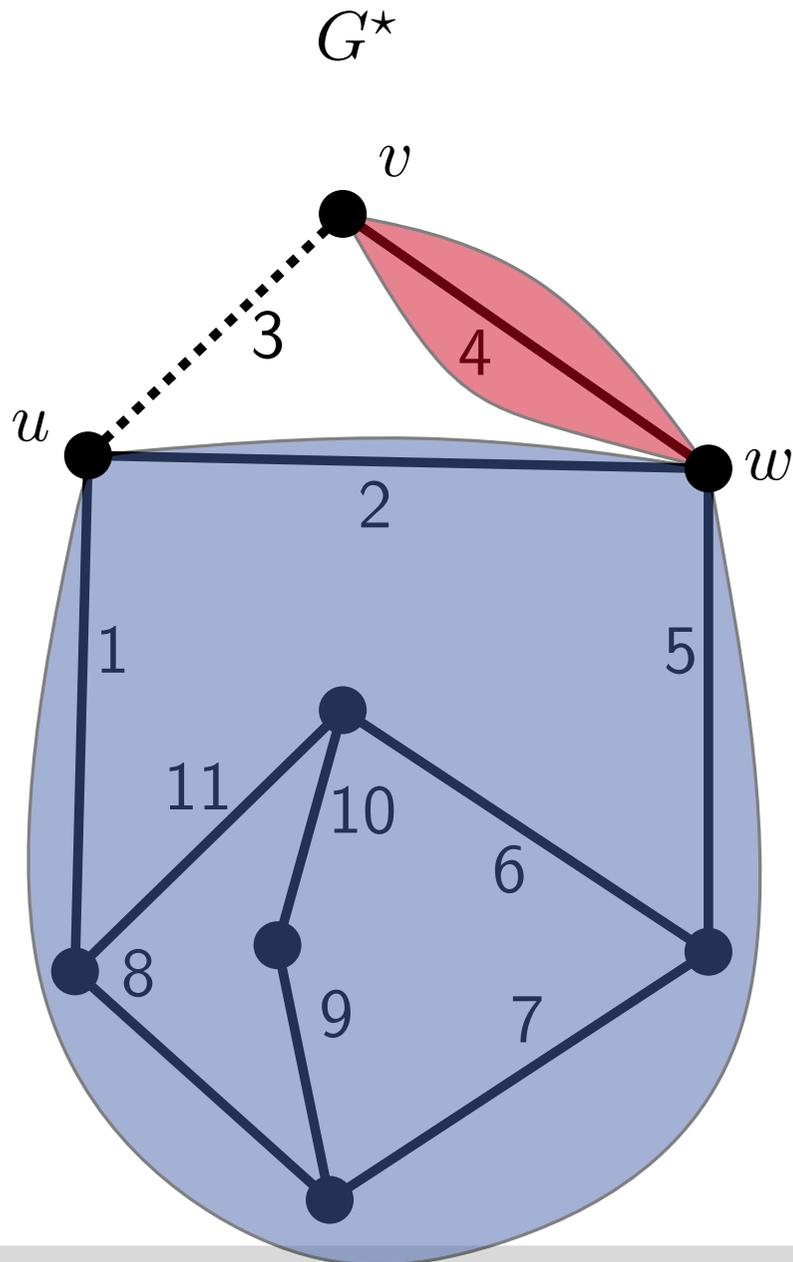
Fallunterscheidung gemäß Struktur von G^*

- Basisfall: G^* einzelne Kante $\Rightarrow \mathcal{T}$ ist einzelner Q-Knoten
- Serie: G^* nicht 2-fach zusammenhängend
- Parallel: $\{u, v\}$ Split Pair von G^*
- Rigid: ...

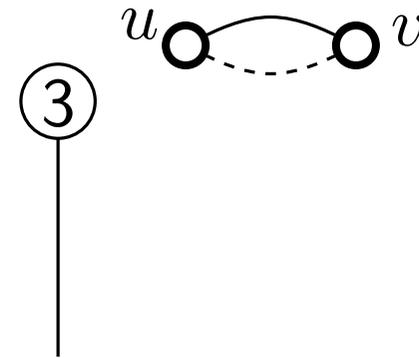
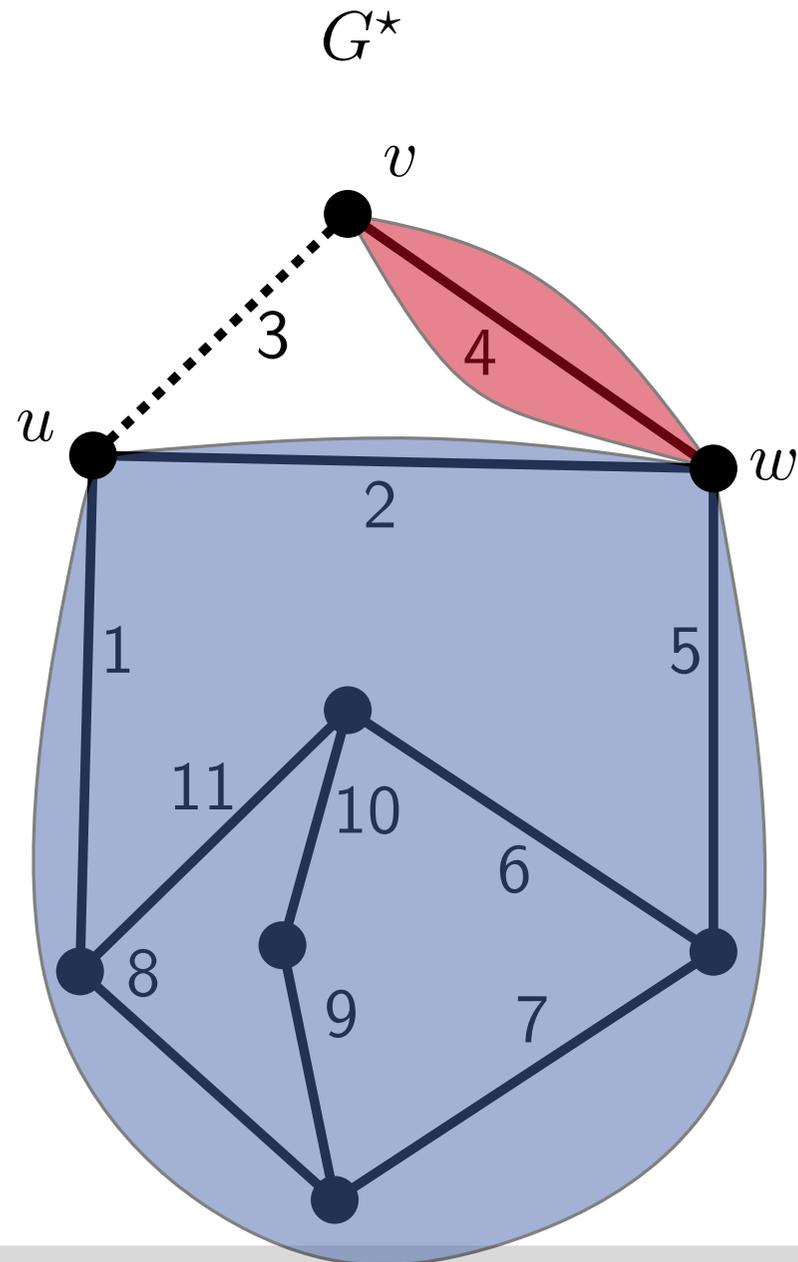
Definiere jeweils entsprechendes Skelett

Füge Kante uv zu jedem Skelett hinzu (*Rest des Graphen*)

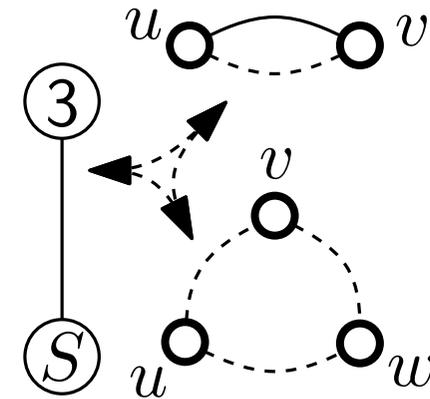
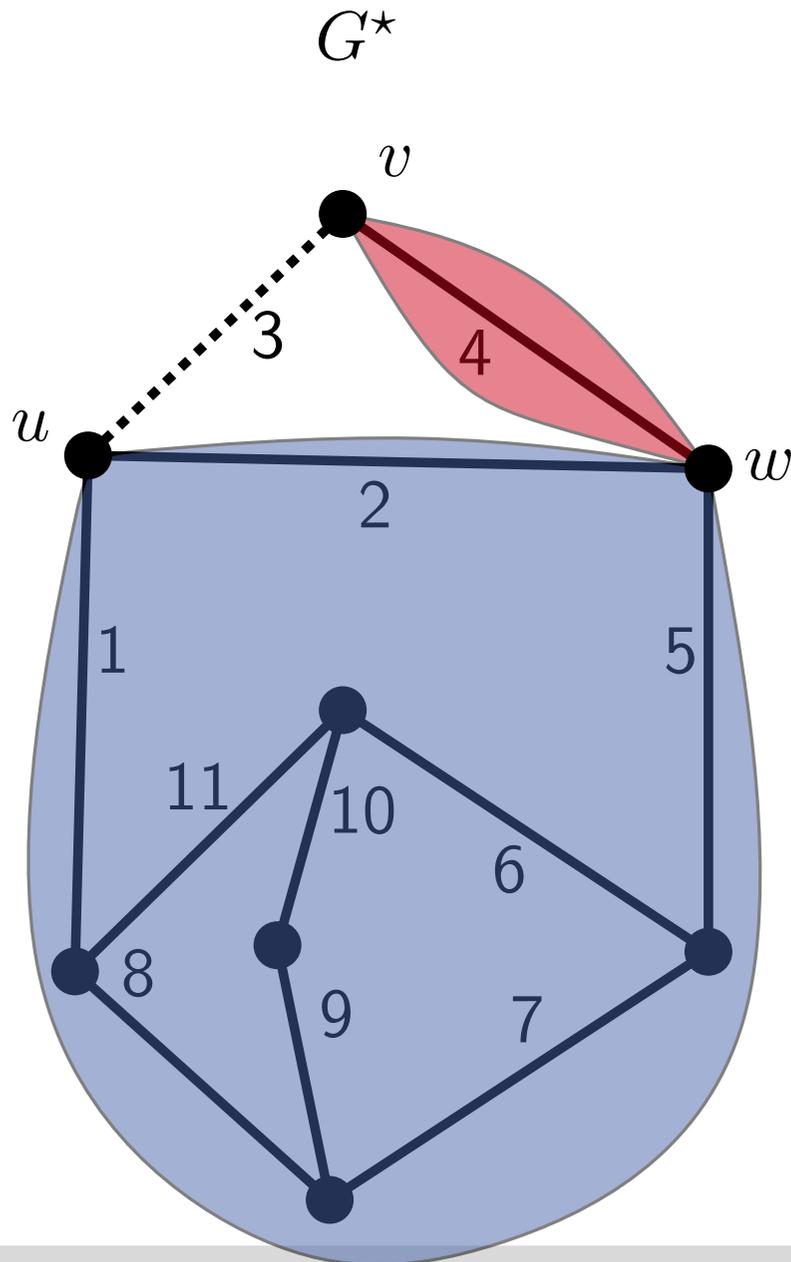
Konstruktion des SPQR-Baums



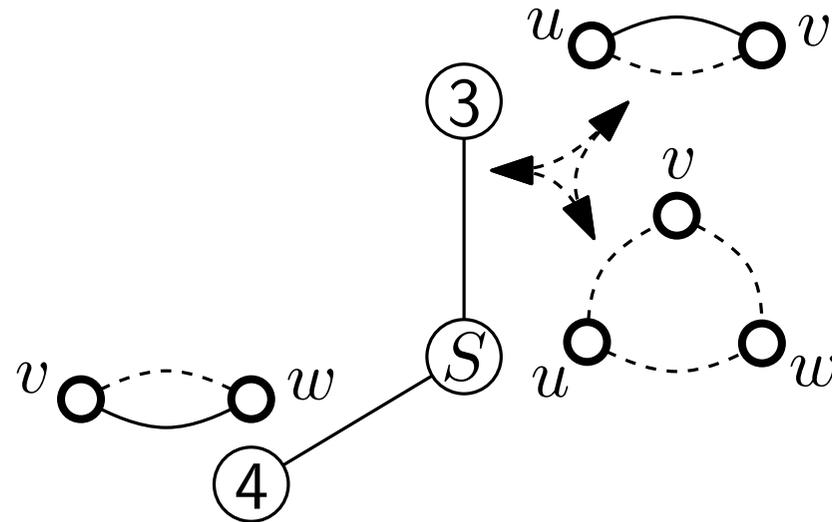
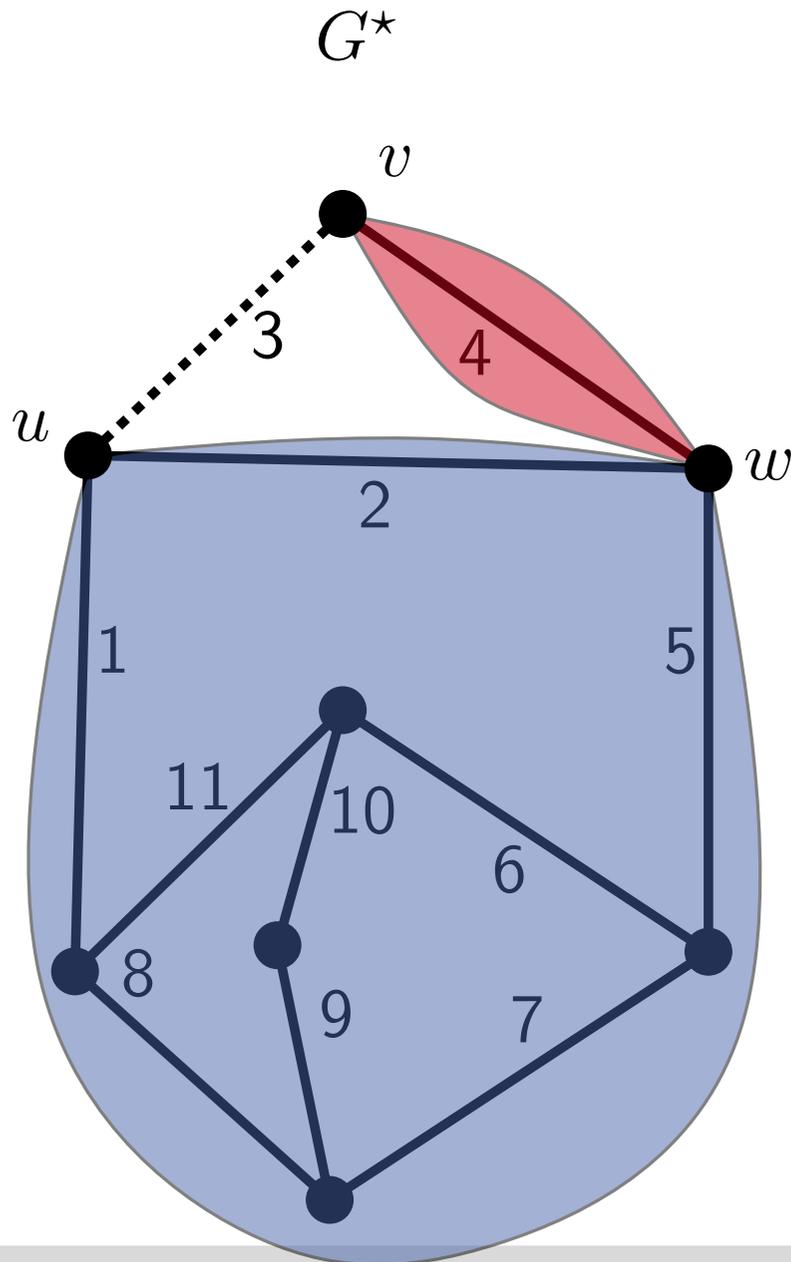
Konstruktion des SPQR-Baums



Konstruktion des SPQR-Baums

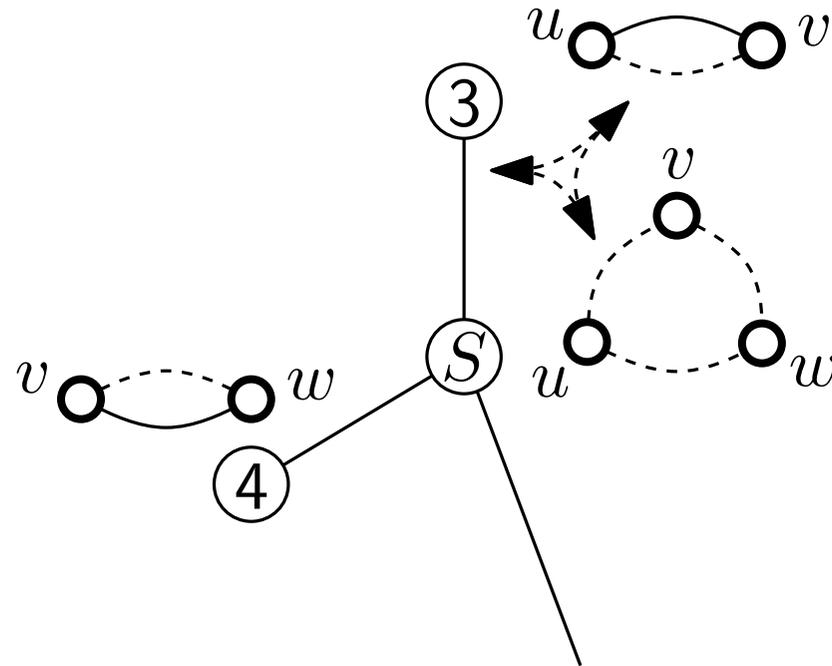
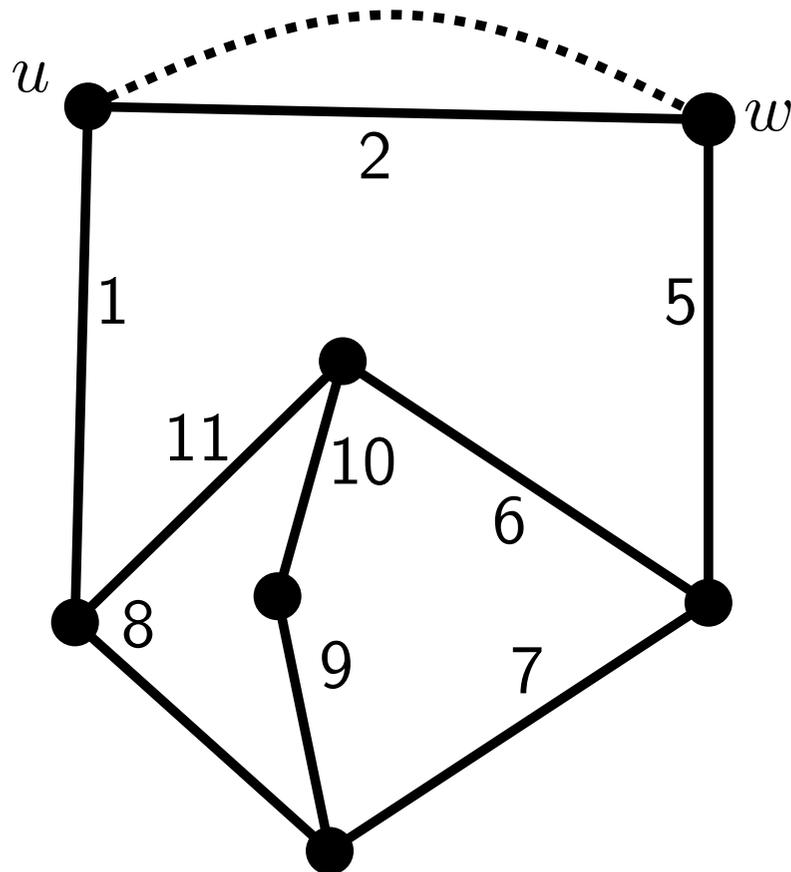


Konstruktion des SPQR-Baums



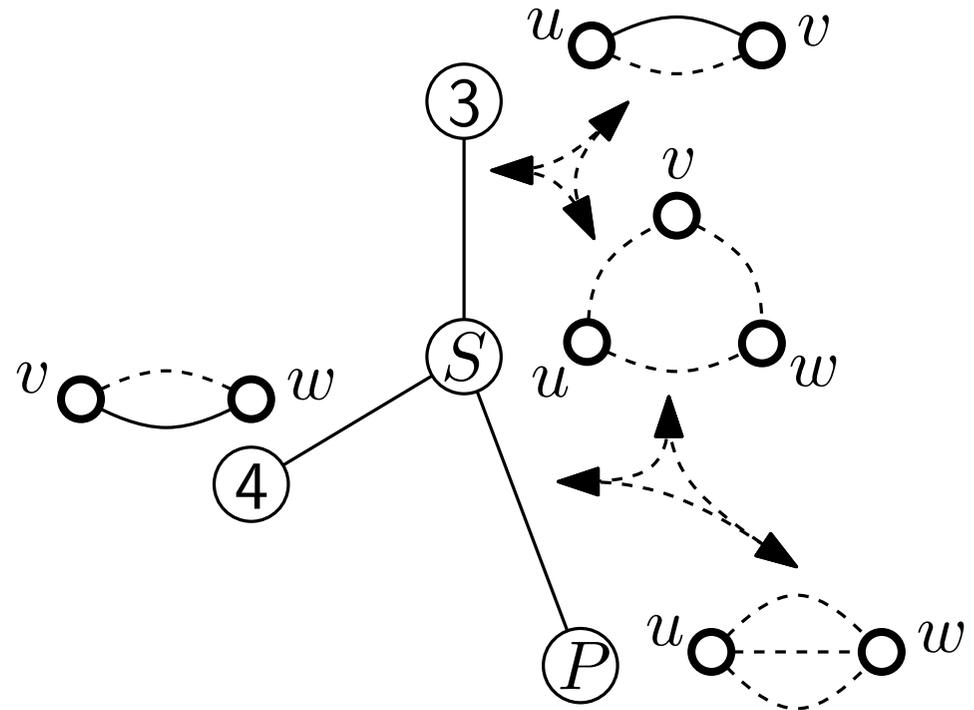
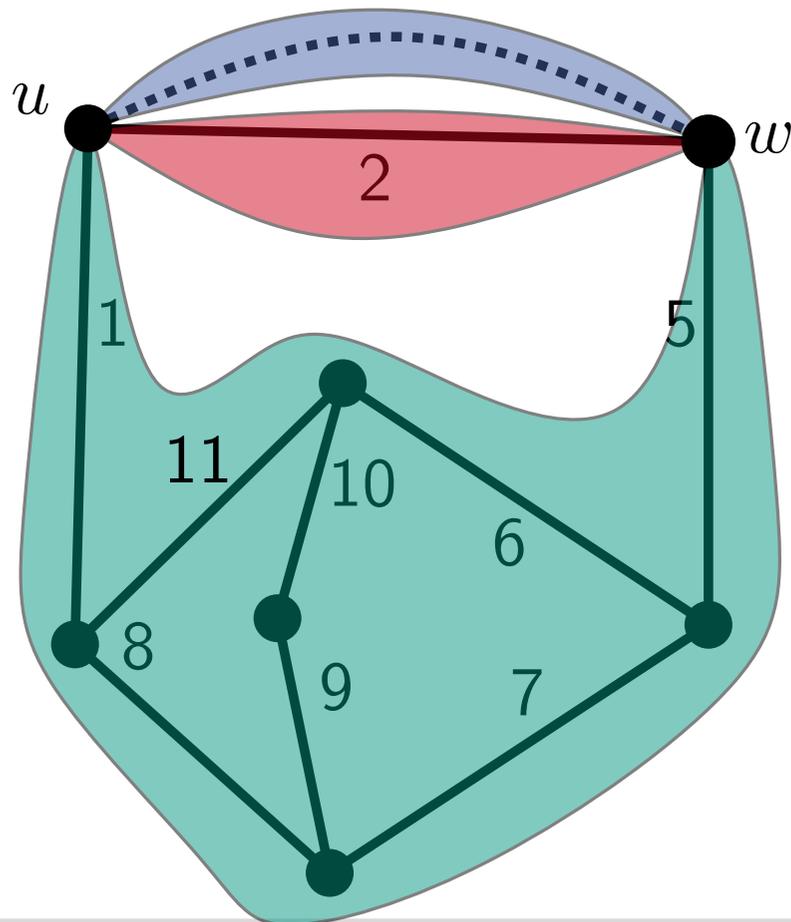
Konstruktion des SPQR-Baums

G^*



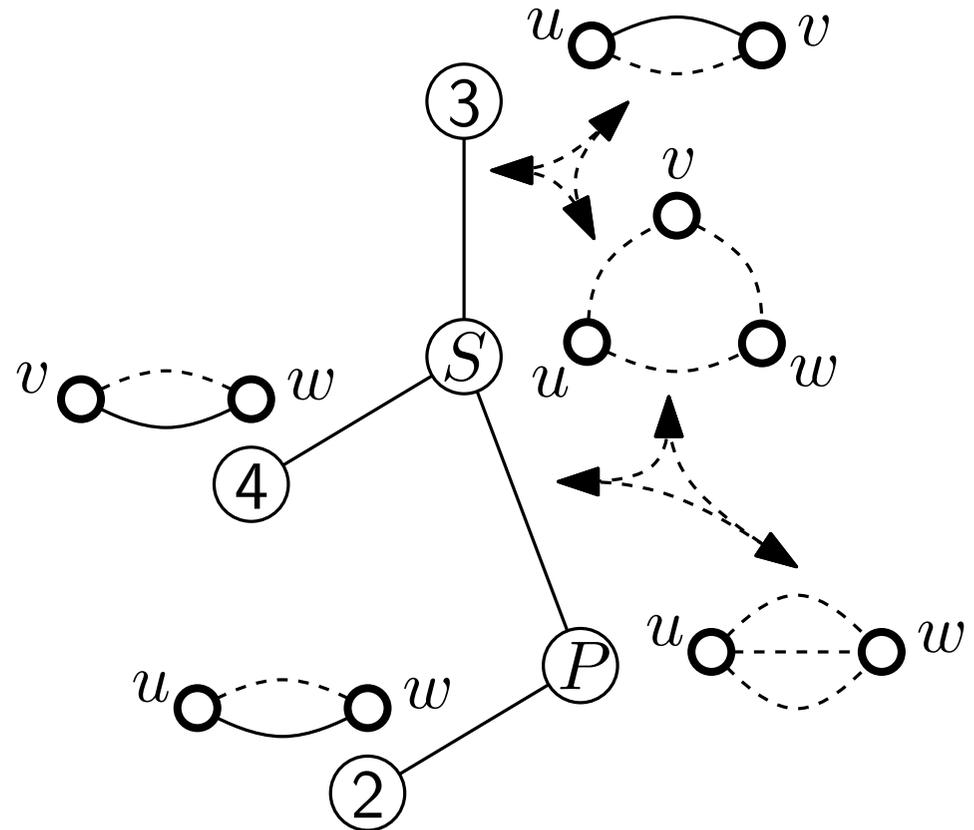
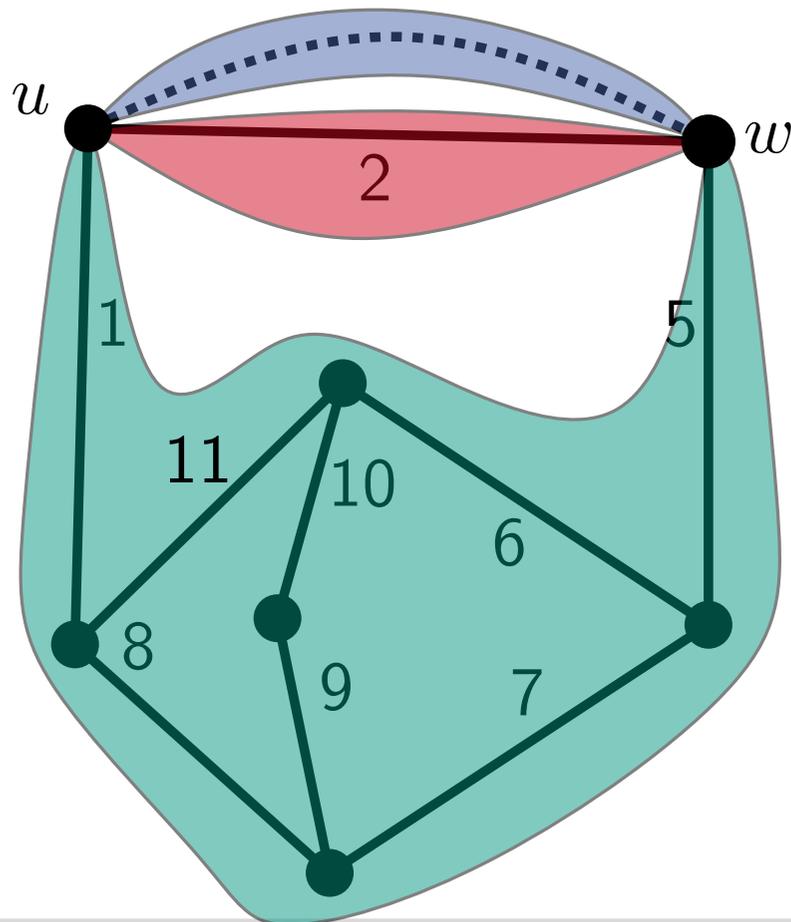
Konstruktion des SPQR-Baums

G^*



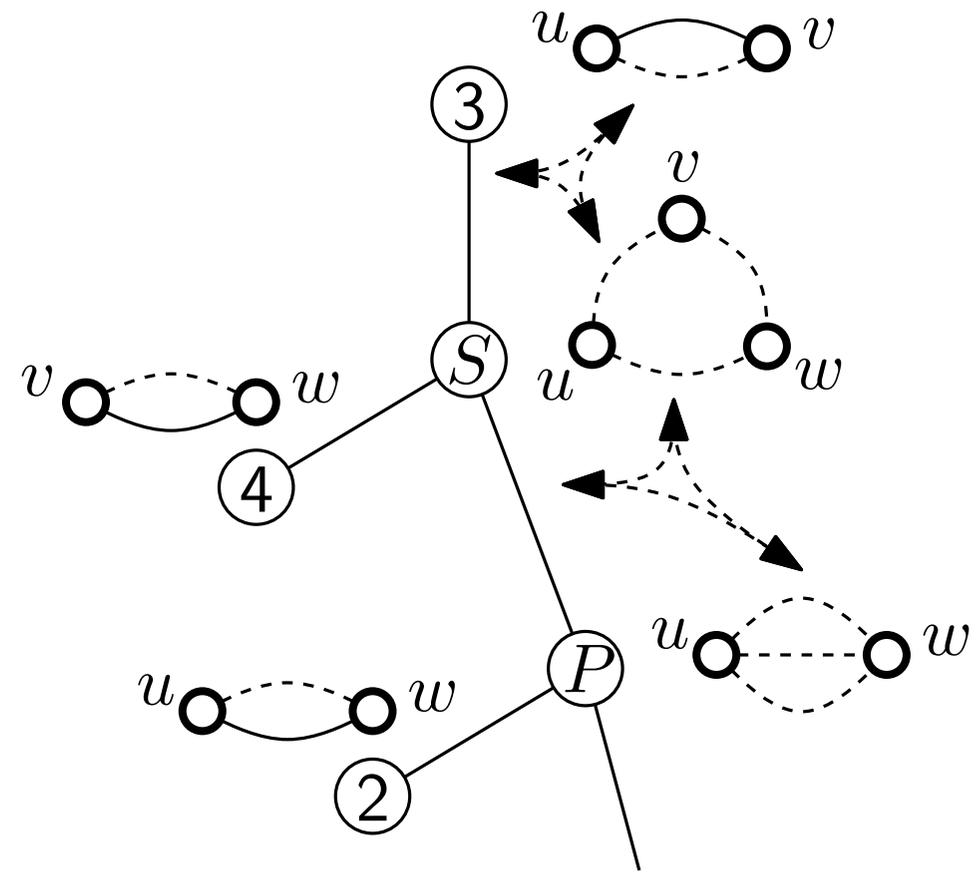
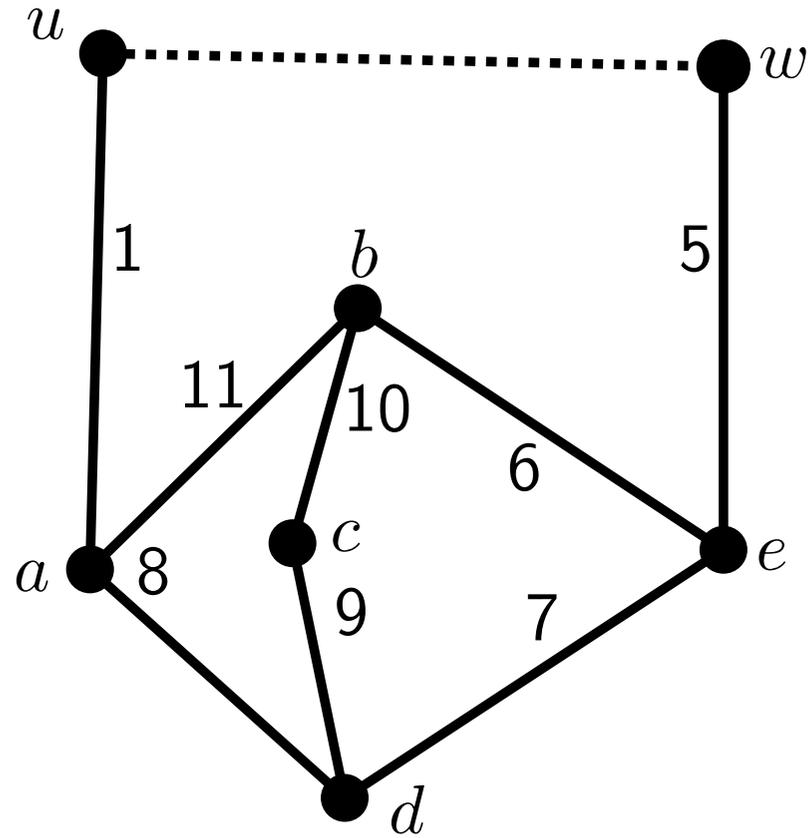
Konstruktion des SPQR-Baums

G^*



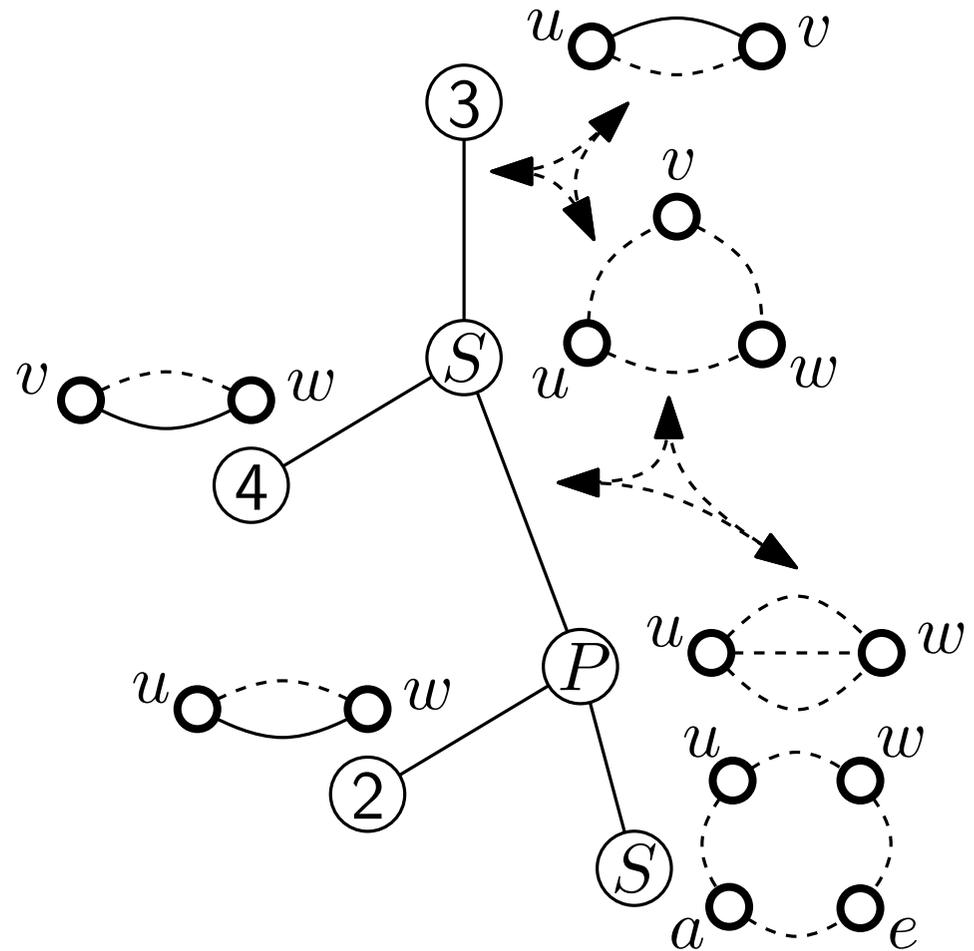
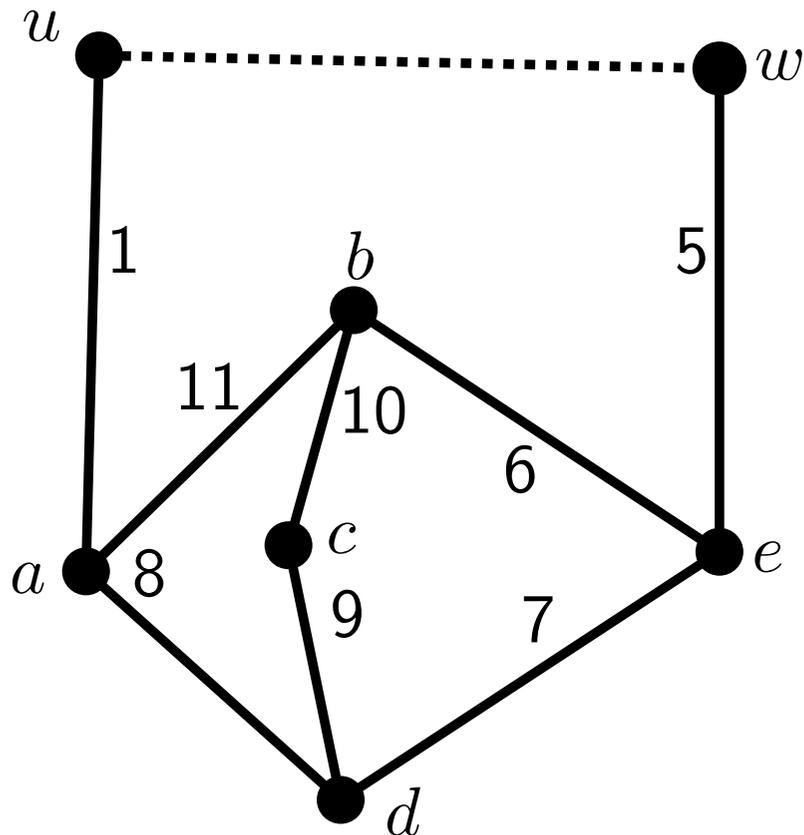
Konstruktion des SPQR-Baums

G^*



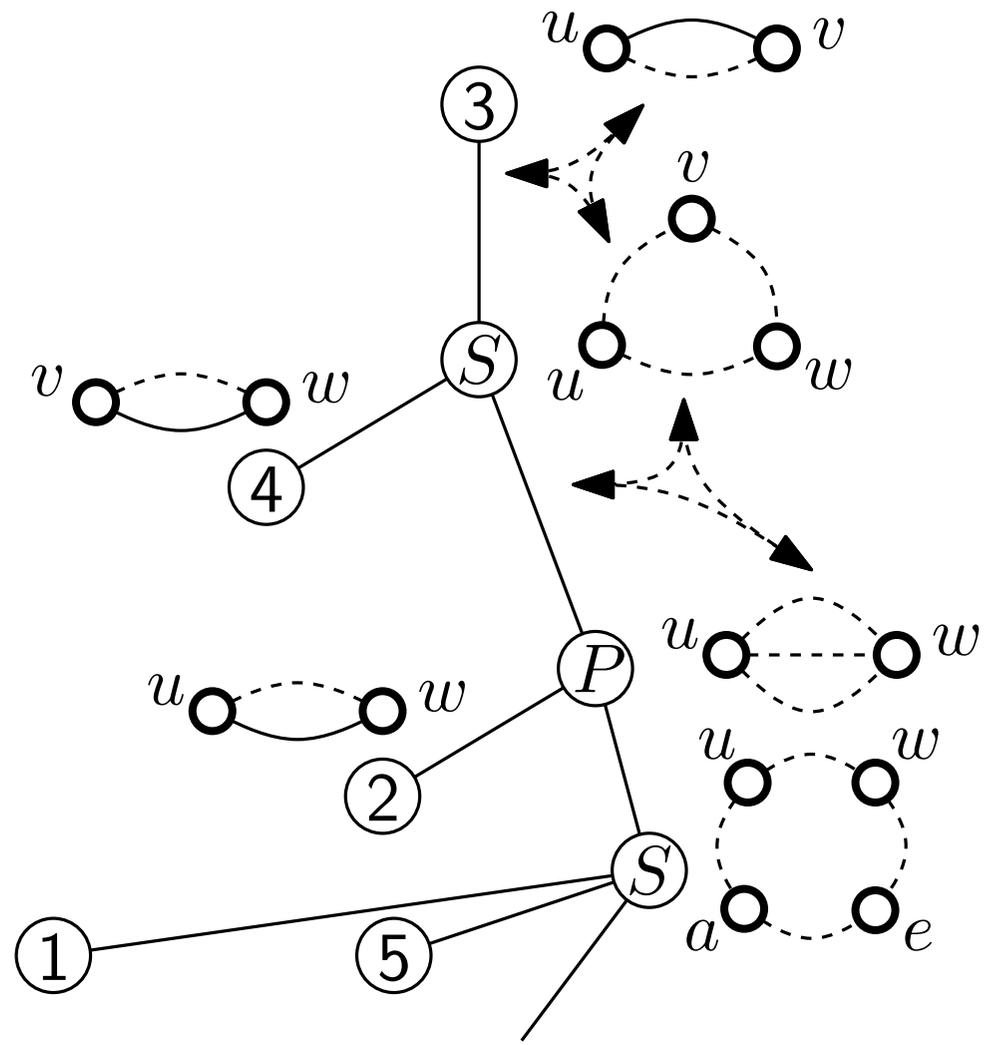
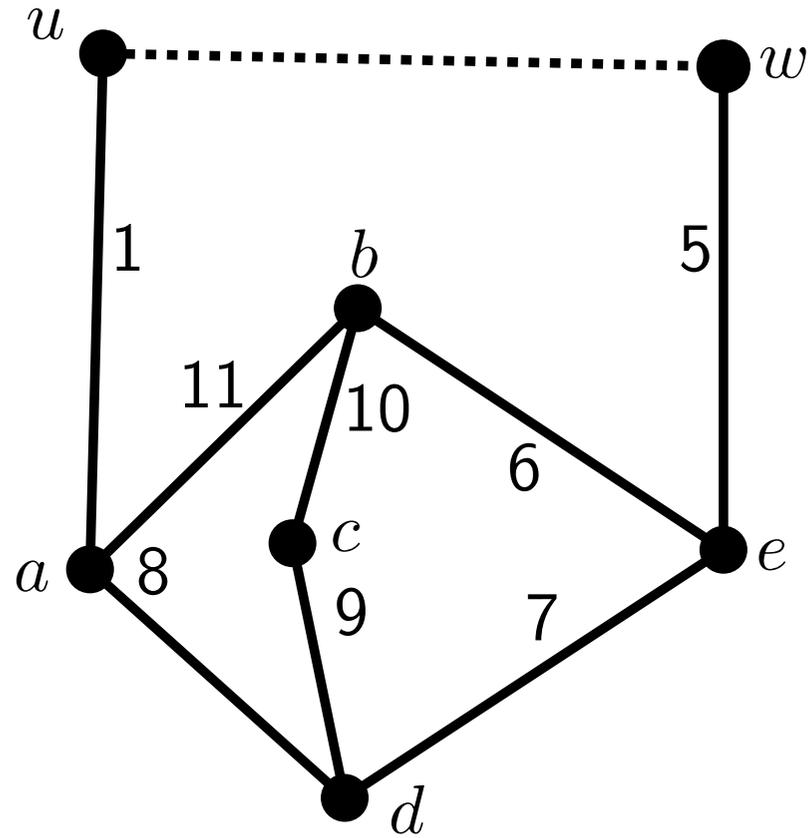
Konstruktion des SPQR-Baums

G^*



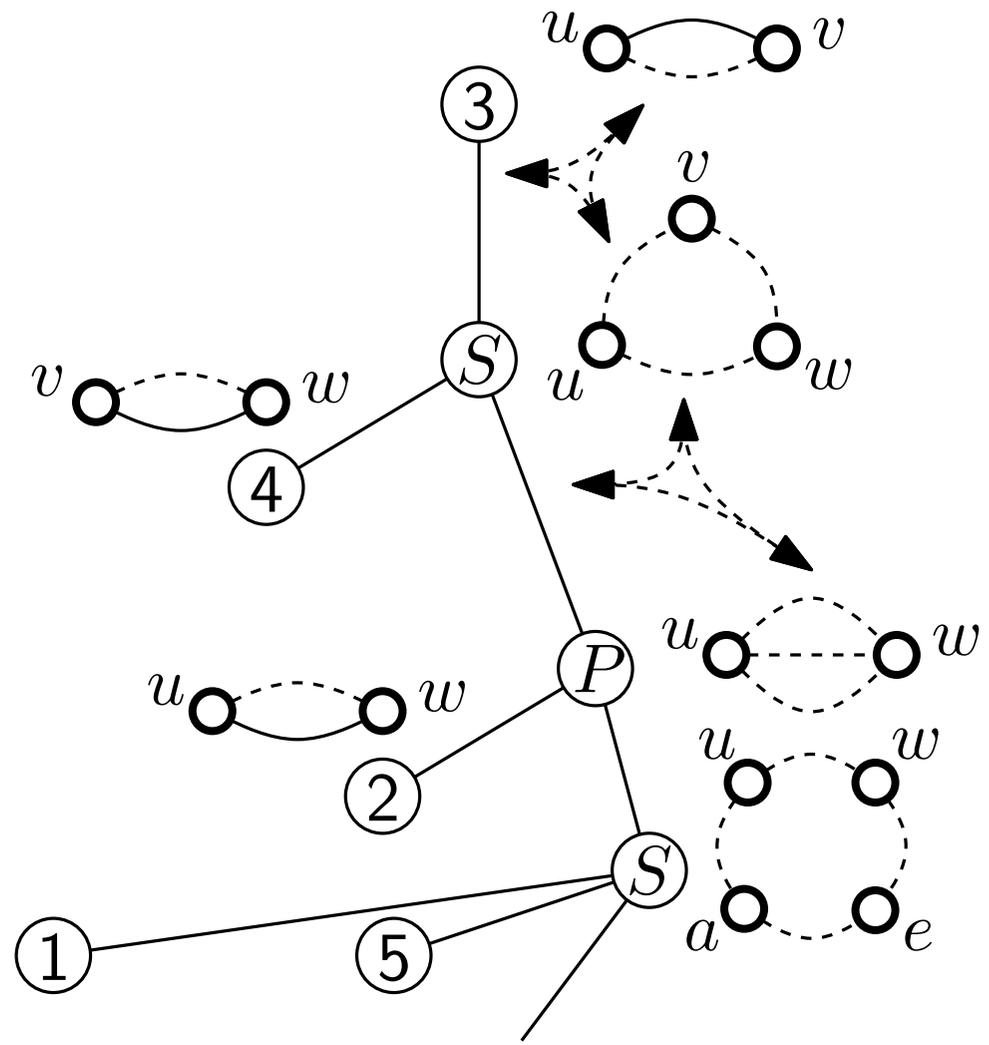
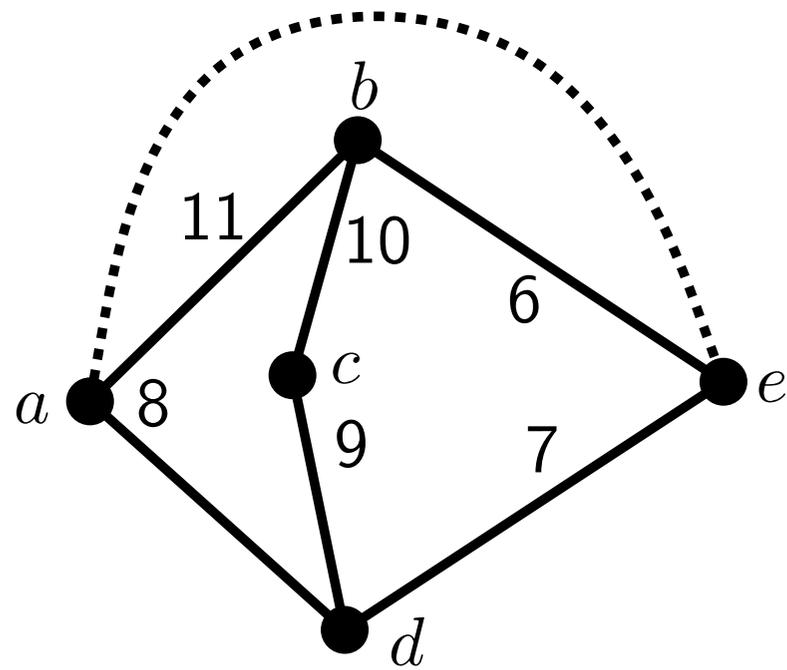
Konstruktion des SPQR-Baums

G^*



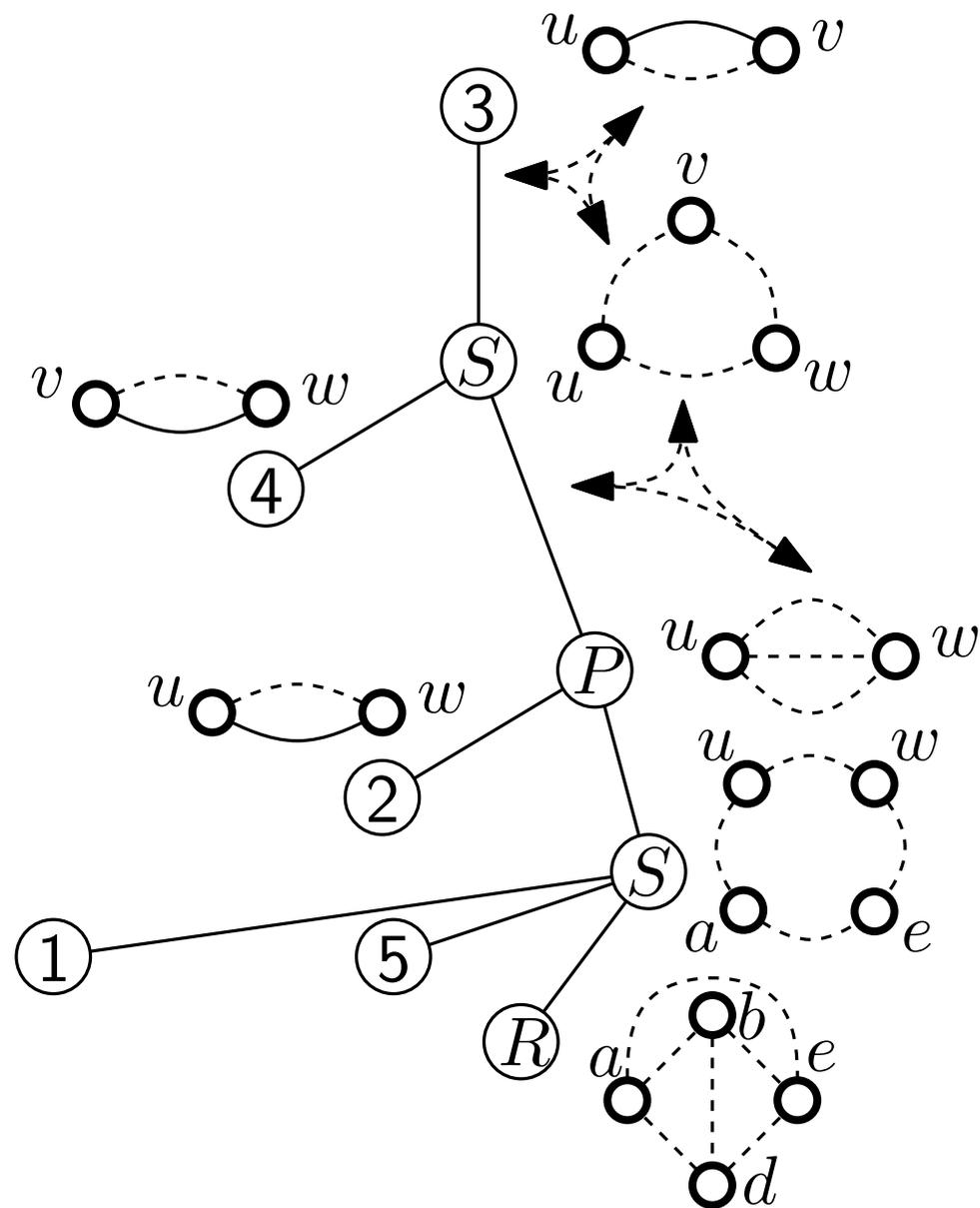
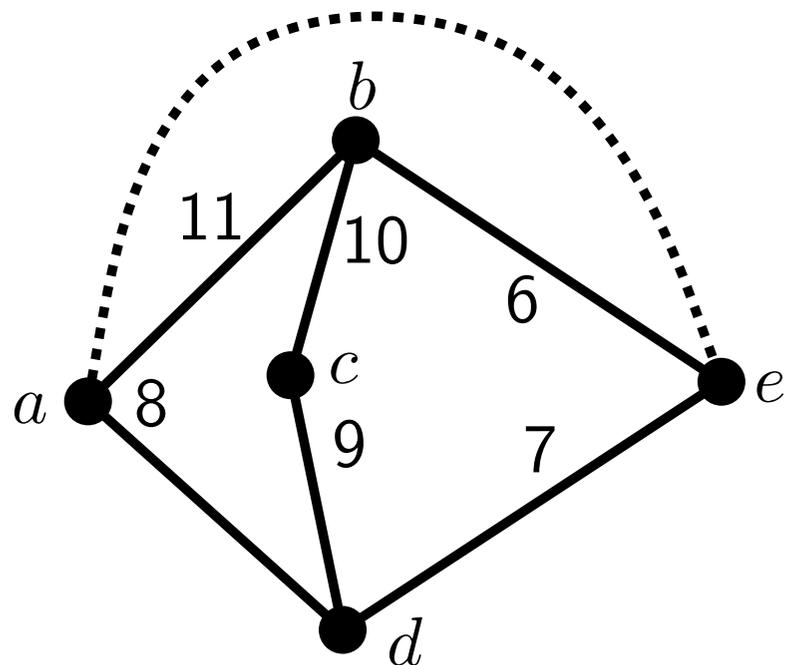
Konstruktion des SPQR-Baums

G^*



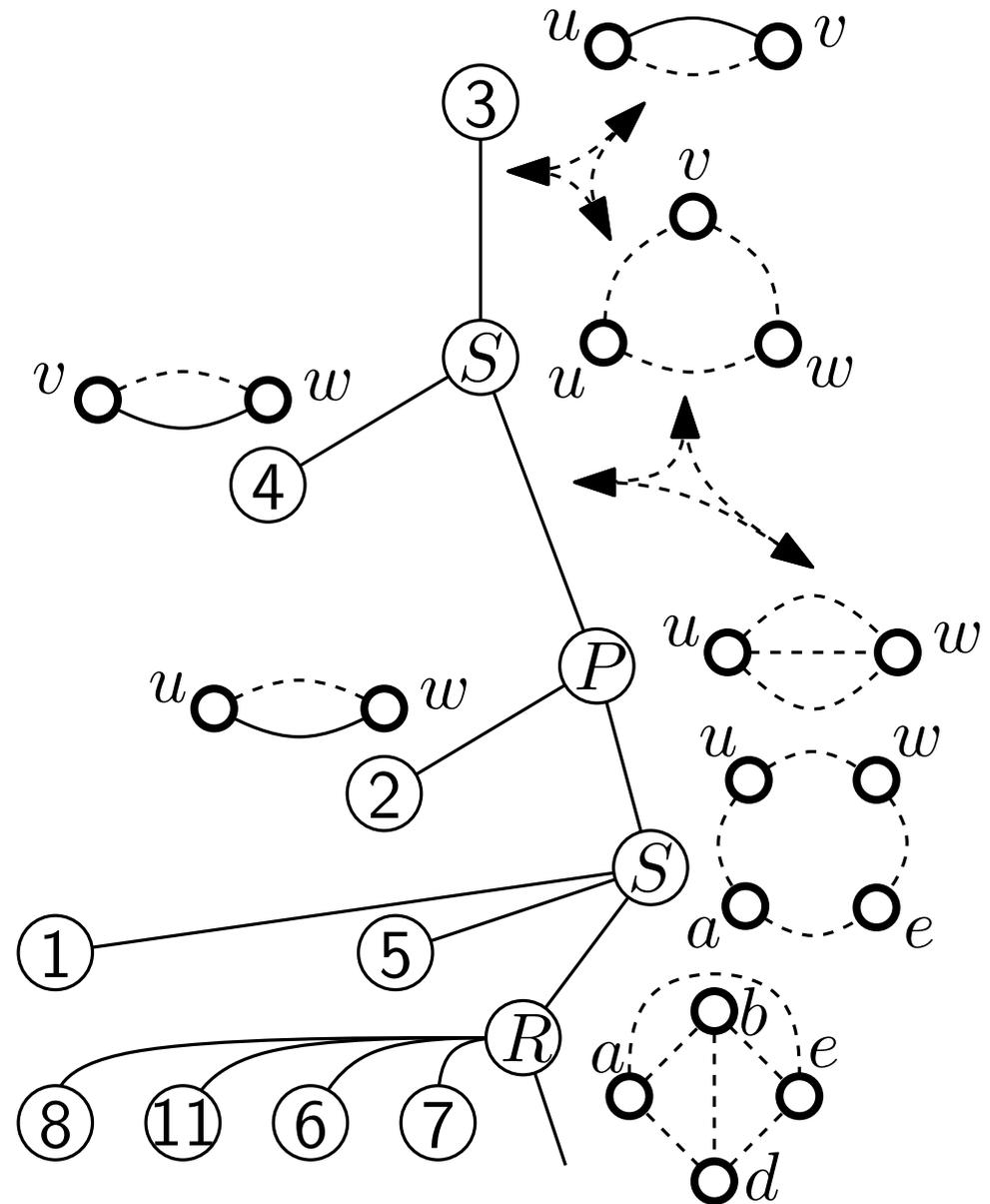
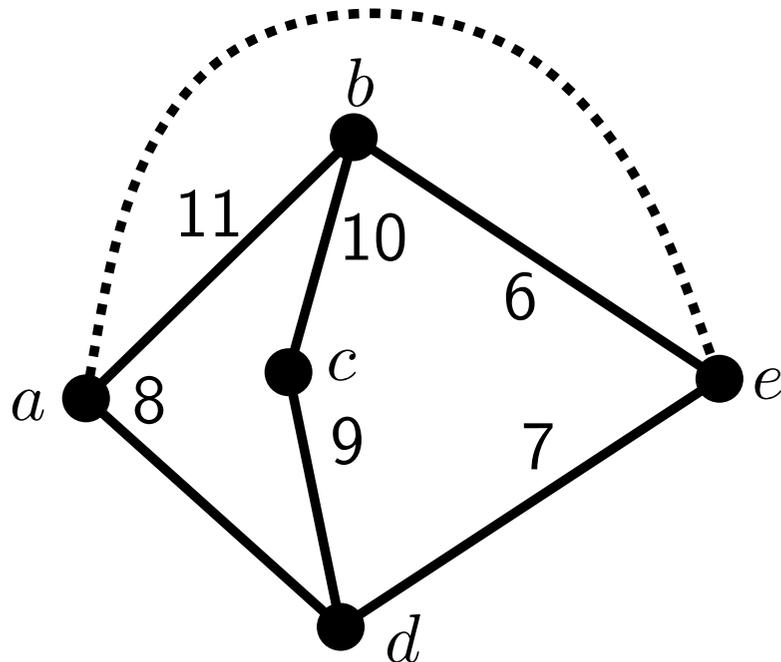
Konstruktion des SPQR-Baums

G^*

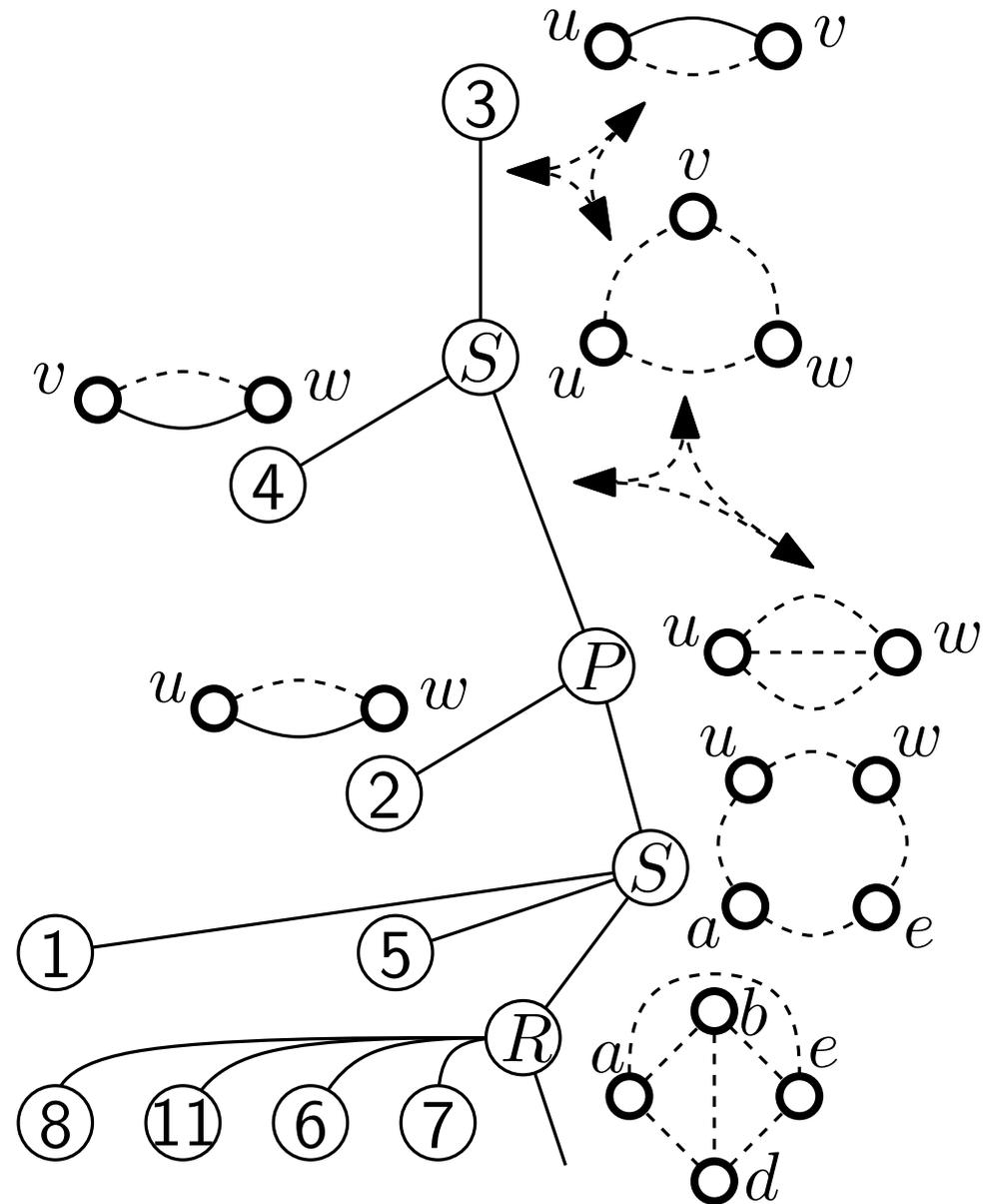
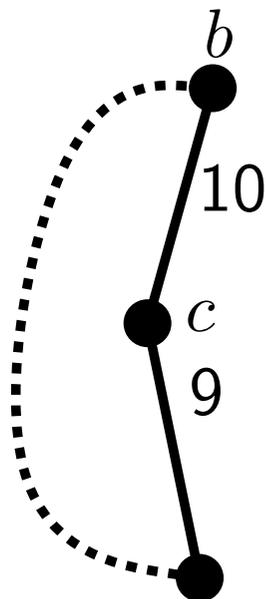


Konstruktion des SPQR-Baums

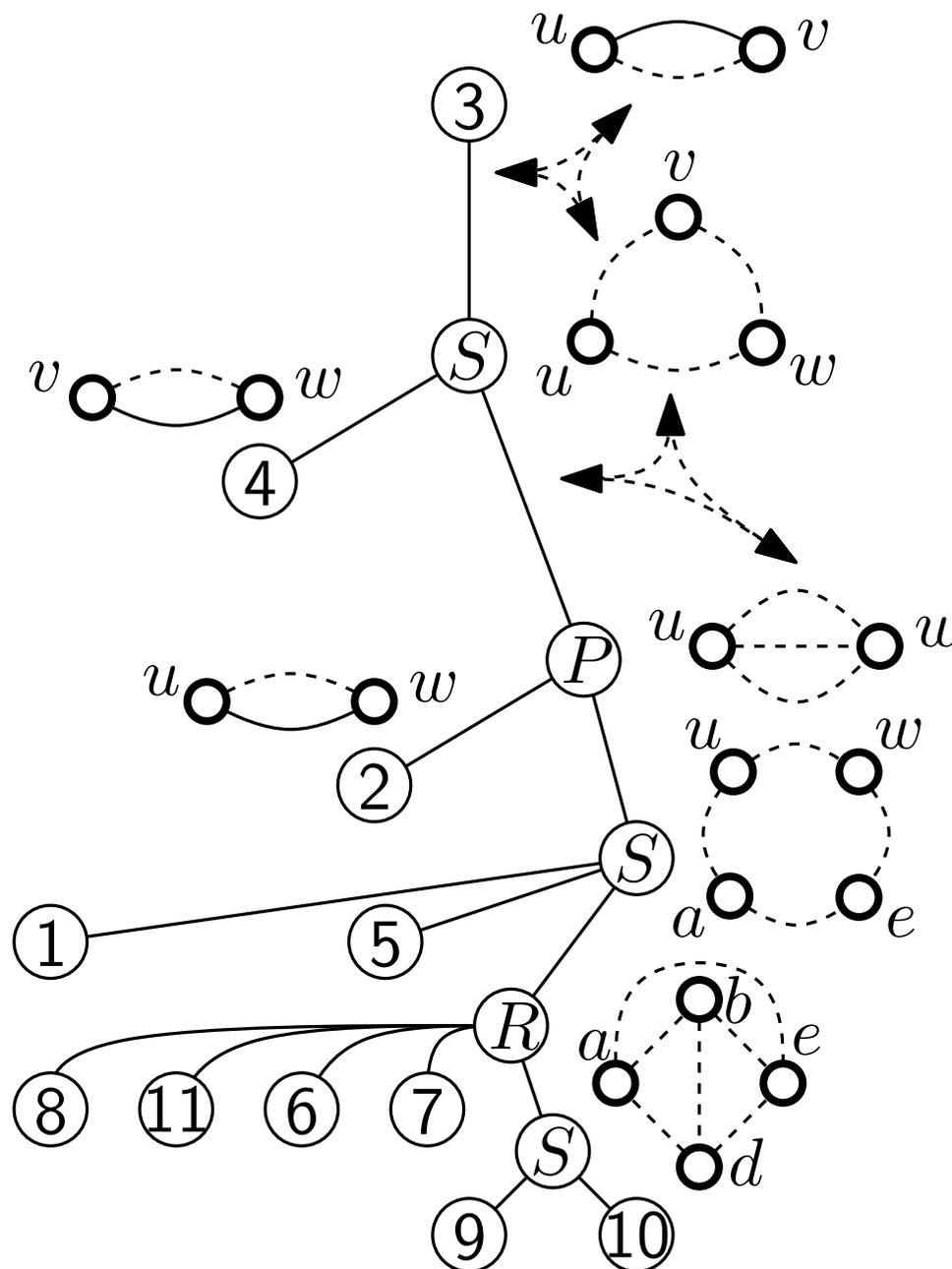
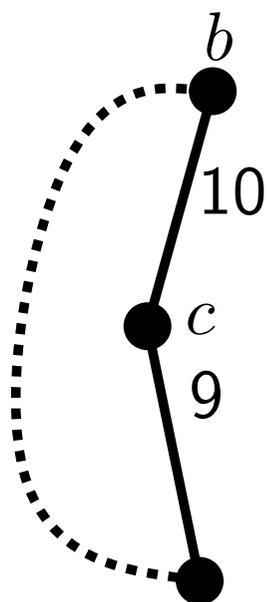
G^*



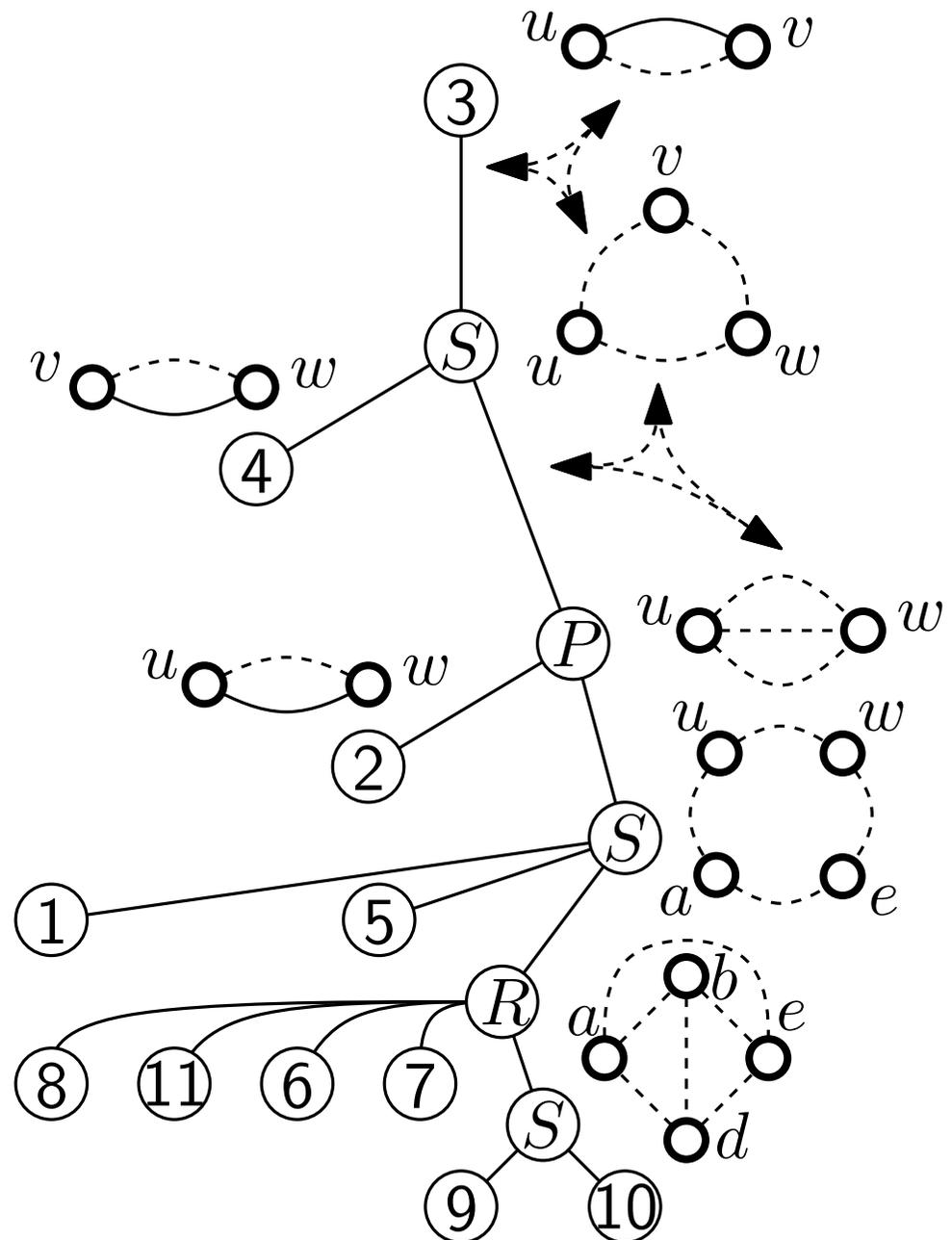
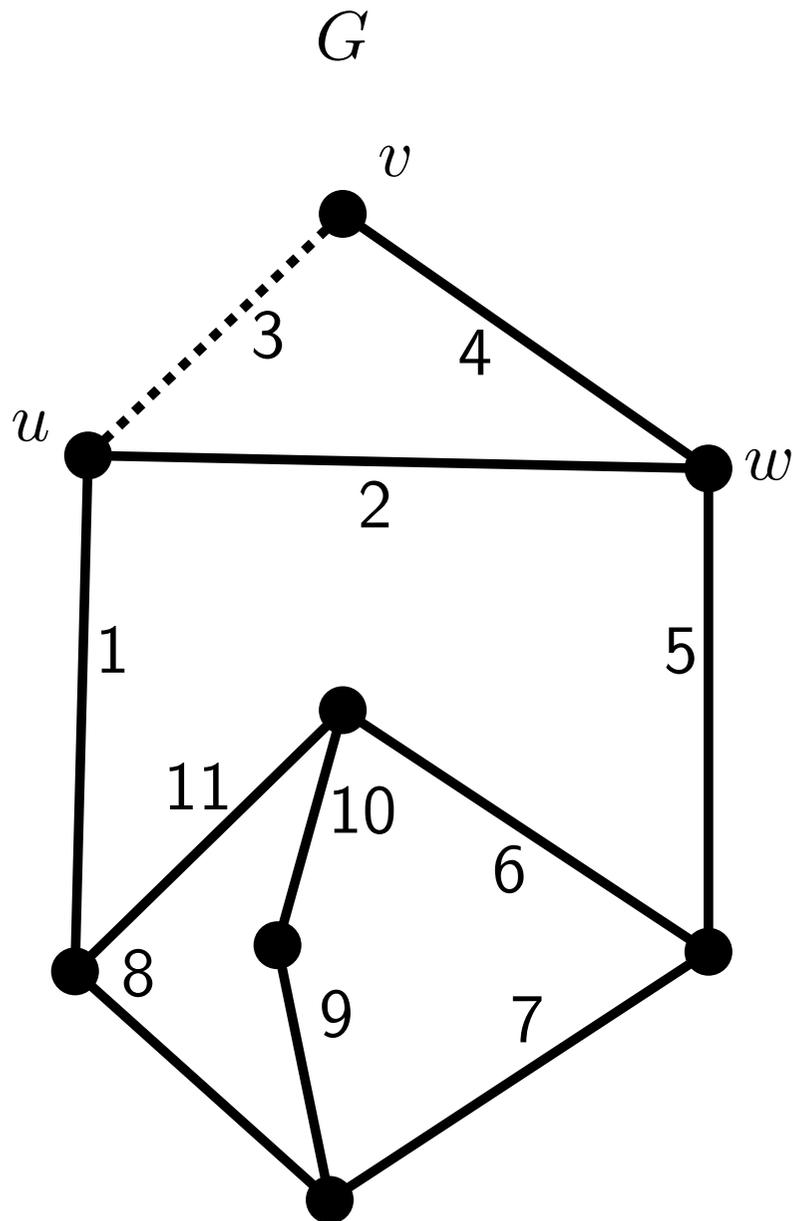
Konstruktion des SPQR-Baums



Konstruktion des SPQR-Baums

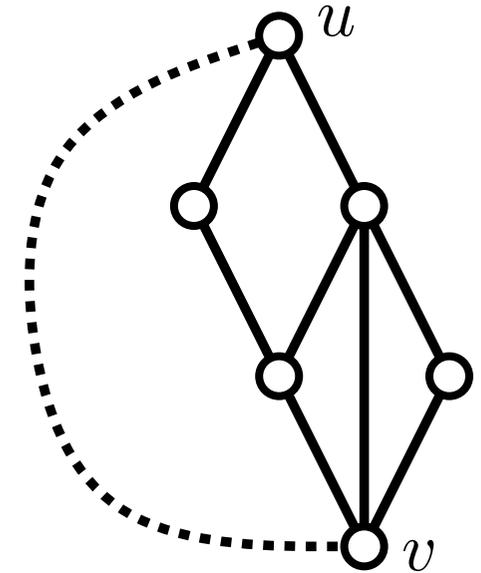


Konstruktion des SPQR-Baums



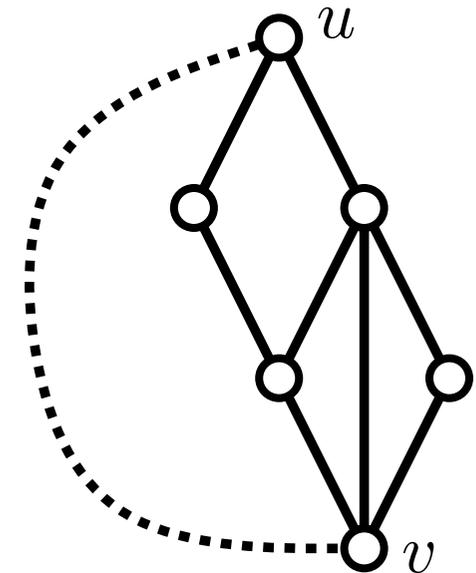
Konstruktion von R-Knoten

- $\{s, t\}$ Split Pair von G^* , G' max. Split-Komp.
- w **gehört echt** zu G' , wenn $w \neq s, t$
- G' ist **intern**, wenn weder u noch v echt zu G' gehören
sonst: G' **extern**



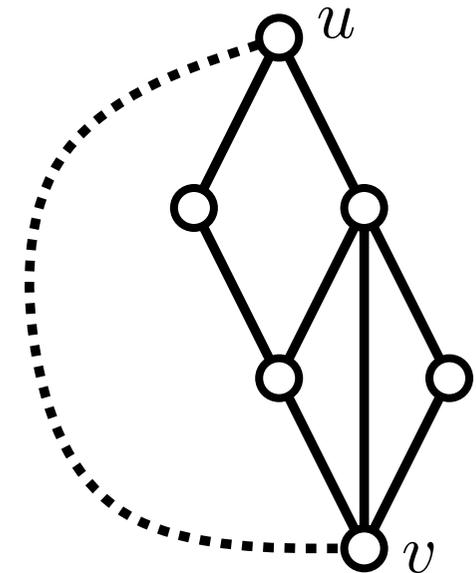
Konstruktion von R-Knoten

- $\{s, t\}$ Split Pair von G^* , G' max. Split-Komp.
- w **gehört echt** zu G' , wenn $w \neq s, t$
- G' ist **intern**, wenn weder u noch v echt zu G' gehören
sonst: G' **extern**
- $\{s, t\}$ ist **maximal**, wenn
 1. besitzt interne Split-Komponente
 2. keine andere maximal interne Splitkomponente bzgl. $\{s', t'\}$ die Knoten s und t enthält



Konstruktion von R-Knoten

- $\{s, t\}$ Split Pair von G^* , G' max. Split-Komp.
- w **gehört echt** zu G' , wenn $w \neq s, t$
- G' ist **intern**, wenn weder u noch v echt zu G' gehören
sonst: G' **extern**
- $\{s, t\}$ ist **maximal**, wenn
 1. besitzt interne Split-Komponente
 2. keine andere maximal interne Splitkomponente bzgl. $\{s', t'\}$ die Knoten s und t enthält



$\{u_1, v_1\}, \dots, \{u_k, v_k\}$ maximale Split Pairs, G_i Vereinigung aller inneren maximalen Split-Komponenten von $\{u_i, v_i\}$

Konstruktion von $\text{skel}(\mu)$: ersetze G_i durch Kante e_i

Verschmelzen von Skeletten

Skelette benachbarter Knoten teilen sich

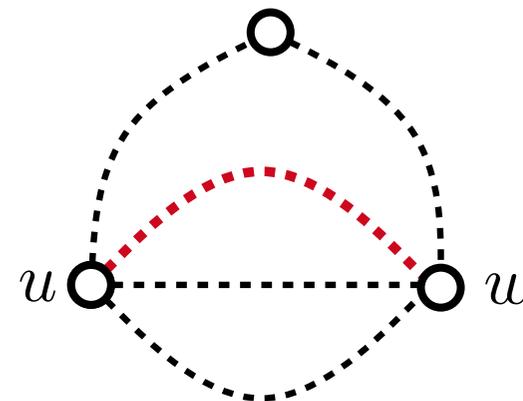
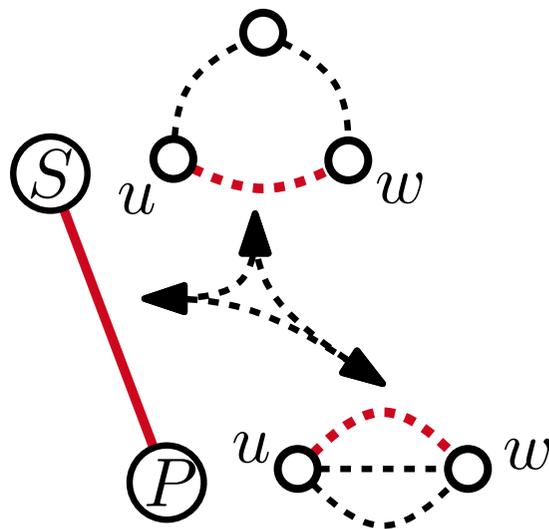
- genau zwei Knoten
- genau eine virtuelle Kante

Wurzelung liefert spezielle virtuelle Kante für jeden Knoten μ

- Endknoten heißen **Pole von μ**

Verschmelze zwei Skelette:

- Identifiziere gemeinsame Knoten und virtuelle Kante
- Entferne gemeinsame virtuelle Kante



Verschmelzen von Skeletten

Skelette benachbarter Knoten teilen sich

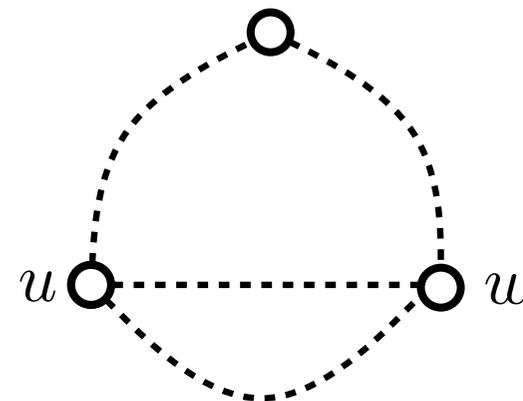
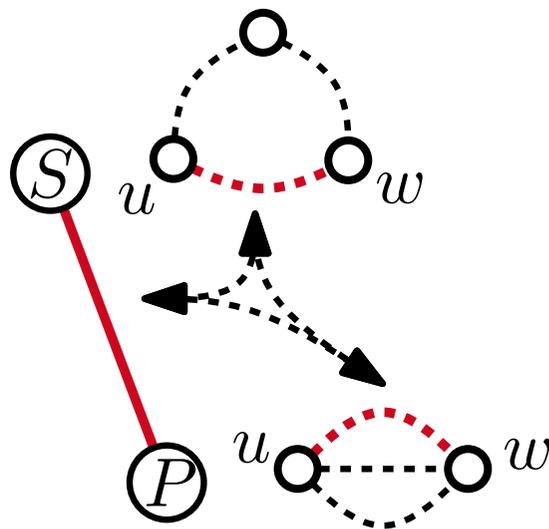
- genau zwei Knoten
- genau eine virtuelle Kante

Wurzelung liefert spezielle virtuelle Kante für jeden Knoten μ

- Endknoten heißen **Pole von μ**

Verschmelze zwei Skelette:

- Identifiziere gemeinsame Knoten und virtuelle Kante
- Entferne gemeinsame virtuelle Kante



Pertinent Graph $\text{pert}(\mu)$ entsteht durch sukzessives Verschmelzen von μ mit allen Skeletten im Teilbaum von μ

- Pertinent Graph des Wurzelknotens ist G
- Nützlich für dynamische Programmierung

Allokationsknoten eines Knoten v aus G sind alle Knoten μ , deren Skelett v enthält.

- Bilden Teilbaum von \mathcal{T}

SPQR-Baum und Planarität

Bis jetzt: Alles auch für nicht-planare Graphen

Jedes Skelett ist **Minor** von G

SPQR-Baum und Planarität

Bis jetzt: Alles auch für nicht-planare Graphen

Jedes Skelett ist **Minor** von G
 G planar \Rightarrow alle Skelette planar

SPQR-Baum und Planarität

Bis jetzt: Alles auch für nicht-planare Graphen

Jedes Skelett ist **Minor** von G
 G planar \Rightarrow alle Skelette planar

Umgekehrt:

Planare Einbettungen von Skeletten lassen sich verschmelzen
Alle Skelette planar $\Rightarrow G$ planar

SPQR-Baum und Planarität

Bis jetzt: Alles auch für nicht-planare Graphen

Jedes Skelett ist **Minor** von G
 G planar \Rightarrow alle Skelette planar

Umgekehrt:

Planare Einbettungen von Skeletten lassen sich verschmelzen
Alle Skelette planar $\Rightarrow G$ planar

Welche Rolle spielt die Wurzel?

- Referenzkante liegt auf äußerer Facette
- Bette Skelette so ein, dass Referenzkante außen liegt

Ohne Referenzkante: Einbettungen auf der Kugel

SPQR-Baum und Planarität

Bis jetzt: Alles auch für nicht-planare Graphen

Jedes Skelett ist **Minor** von G
 G planar \Rightarrow alle Skelette planar

Umgekehrt:

Planare Einbettungen von Skeletten lassen sich verschmelzen
Alle Skelette planar $\Rightarrow G$ planar

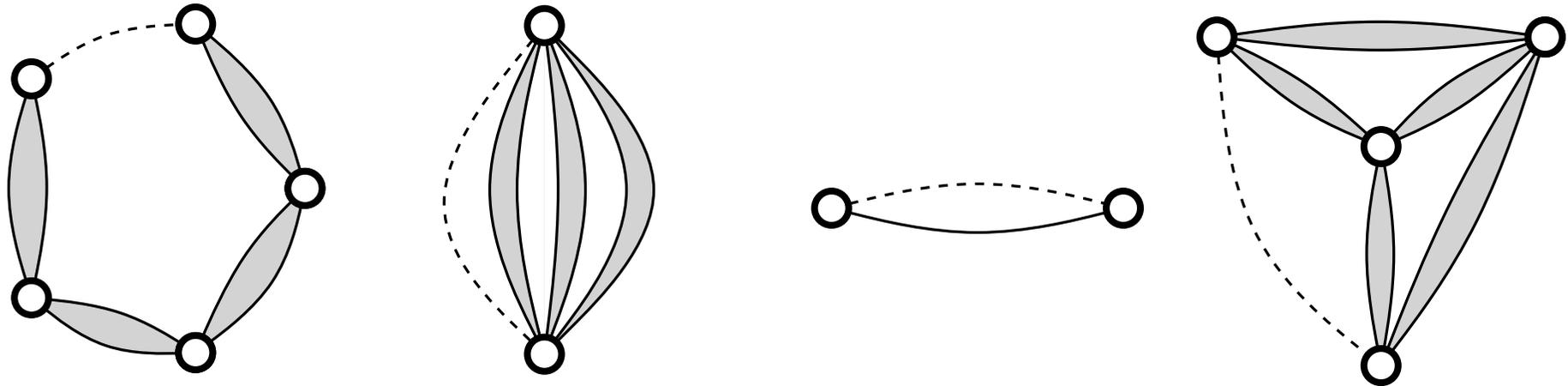
Welche Rolle spielt die Wurzel?

- Referenzkante liegt auf äußerer Facette
- Bette Skelette so ein, dass Referenzkante außen liegt

Ohne Referenzkante: Einbettungen auf der Kugel

Zentrale Aussage:
Einbettung von $G \equiv$ Einbettungen aller Skelette

Bemerkungen (nur für planare Graphen)



- SPQR-Baum repräsentiert alle planaren Einbettungen von 2-fach zusammenhängenden planaren Graphen
- SPQR-Baum hat Größe $O(n)$
- Berechnung in Linearzeit

[Gutwenger, Mutzel '00]

Ermöglicht Optimierung über alle Einbettungen

Problem (FLEXDRAW)

Geg.: Graph $G = (V, E)$, $\text{flex} : E \rightarrow \mathbb{N}_0$

Ges.: Orthogonale, planare Zeichnung von G mit
 $\text{bends}(e) \leq \text{flex}(e)$ für alle $e \in E$

Komplexität?

Problem (FLEXDRAW)

Geg.: Graph $G = (V, E)$, $\text{flex} : E \rightarrow \mathbb{N}_0$

Ges.: Orthogonale, planare Zeichnung von G mit
 $\text{bends}(e) \leq \text{flex}(e)$ für alle $e \in E$

Komplexität?

NP-schwer (setze $\text{flex} \equiv 0$)

Problem (FLEXDRAW)

Geg.: Graph $G = (V, E)$, $\text{flex} : E \rightarrow \mathbb{N}_0$

Ges.: Orthogonale, planare Zeichnung von G mit
 $\text{bends}(e) \leq \text{flex}(e)$ für alle $e \in E$

Komplexität?

NP-schwer (setze $\text{flex} \equiv 0$)

FLEXDRAW mit **positiver Flexibilität**: $\text{flex}(e) \geq 1$ für $e \in E$

Problem (FLEXDRAW)

Geg.: Graph $G = (V, E)$, $\text{flex} : E \rightarrow \mathbb{N}_0$

Ges.: Orthogonale, planare Zeichnung von G mit
 $\text{bends}(e) \leq \text{flex}(e)$ für alle $e \in E$

Komplexität?

NP-schwer (setze $\text{flex} \equiv 0$)

FLEXDRAW mit positiver Flexibilität: $\text{flex}(e) \geq 1$ für $e \in E$

Satz (Bläsius, Krug, Rutter, Wagner '11)

FLEXDRAW mit positiver Flexibilität lässt sich in $O(n^2)$ Zeit lösen.

Skizze für $O(n^{5/2})$