

Algorithmen II

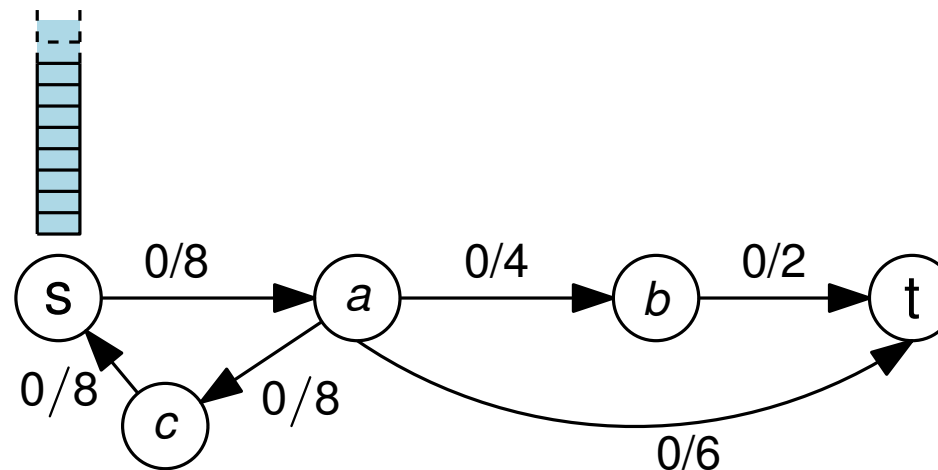
Vorlesung am 31.10.2013

INSTITUT FÜR THEORETISCHE INFORMATIK · PROF. DR. DOROTHEA WAGNER



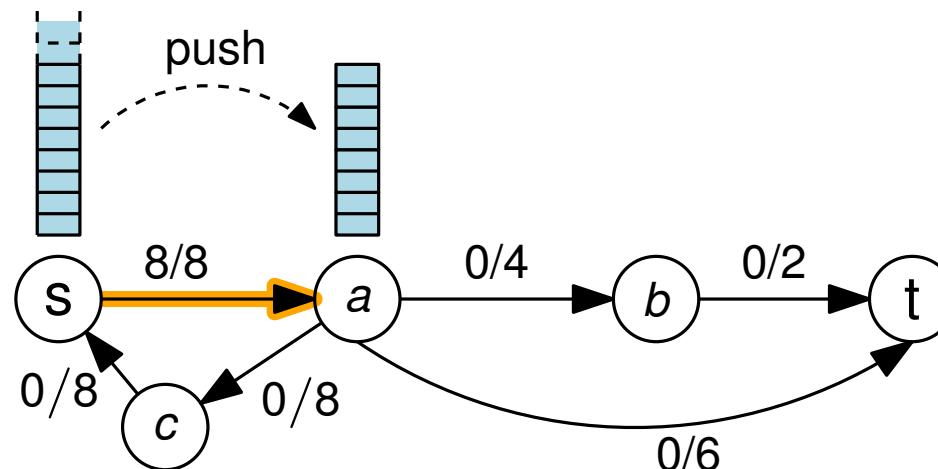
Flussalgorithmus von Goldberg und Tarjan (1988)

- Basiert nicht auf erhöhenden Wegen sondern auf zwei Operationen: **PUSH** und **RELABEL**.
- Fluss in Zwischenschritten nicht unbedingt gültig (Flusserhaltung nicht garantiert).



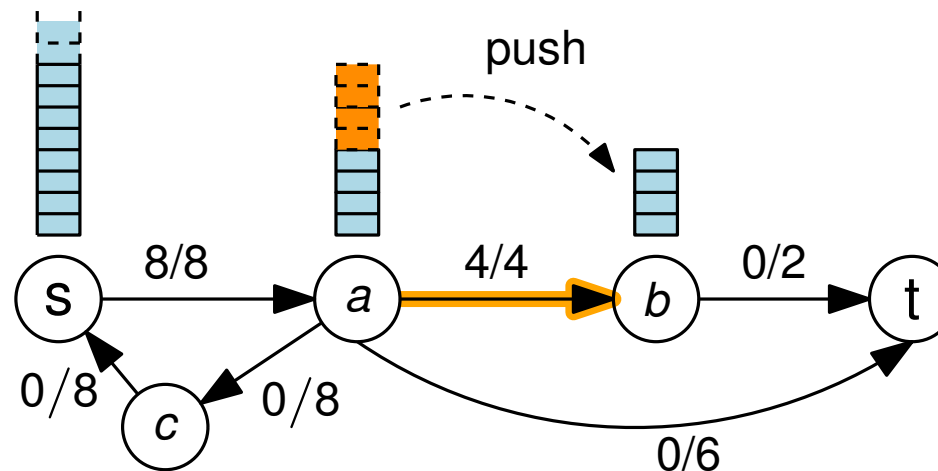
- Im Beispiel nur PUSH-Operation veranschaulicht: Drückt den Fluss zum nächsten Knoten und erzeugt dort gegebenenfalls Überschuss, der zurück geführt werden muss.
- RELABEL-Operation garantiert, dass Fluss in die richtige Richtung gedrückt wird.
 - im Beispiel darf der Fluss nicht von a über c nach s zurück gedrückt werden.

- Basiert nicht auf erhöhenden Wegen sondern auf zwei Operationen: PUSH und RELABEL.
- Fluss in Zwischenschritten nicht unbedingt gültig (Flusserhaltung nicht garantiert).



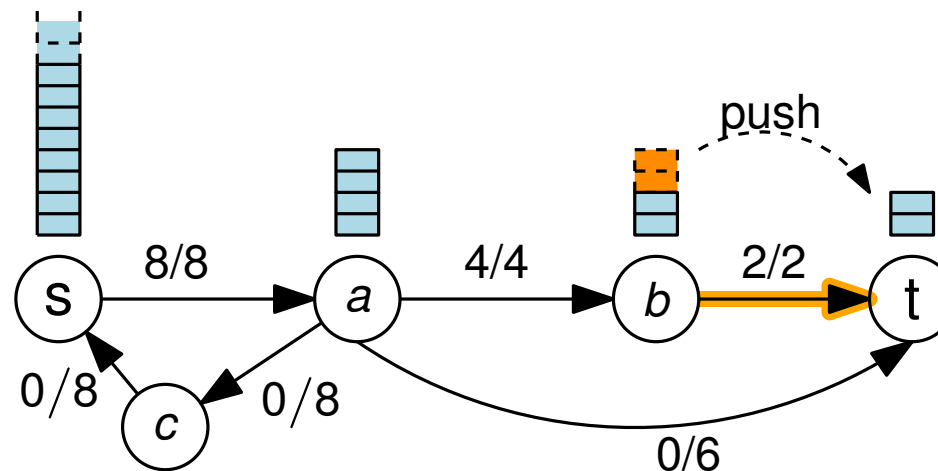
- Im Beispiel nur PUSH-Operation veranschaulicht: Drückt den Fluss zum nächsten Knoten und erzeugt dort gegebenenfalls Überschuss, der zurück geführt werden muss.
- RELABEL-Operation garantiert, dass Fluss in die richtige Richtung gedrückt wird.
 - im Beispiel darf der Fluss nicht von a über c nach s zurück gedrückt werden.

- Basiert nicht auf erhöhenden Wegen sondern auf zwei Operationen: PUSH und RELABEL.
- Fluss in Zwischenschritten nicht unbedingt gültig (Flusserhaltung nicht garantiert).



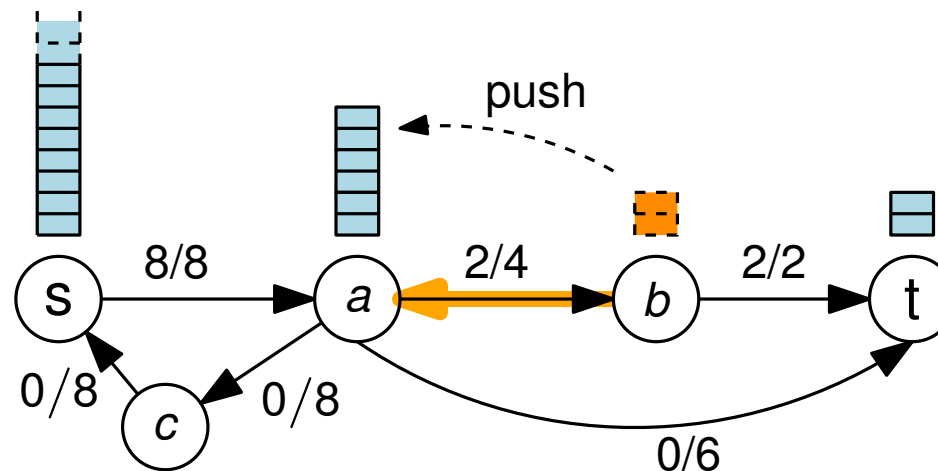
- Im Beispiel nur PUSH-Operation veranschaulicht: Drückt den Fluss zum nächsten Knoten und erzeugt dort gegebenenfalls Überschuss, der zurück geführt werden muss.
- RELABEL-Operation garantiert, dass Fluss in die richtige Richtung gedrückt wird.
 - im Beispiel darf der Fluss nicht von a über c nach s zurück gedrückt werden.

- Basiert nicht auf erhöhenden Wegen sondern auf zwei Operationen: PUSH und RELABEL.
- Fluss in Zwischenschritten nicht unbedingt gültig (Flusserhaltung nicht garantiert).



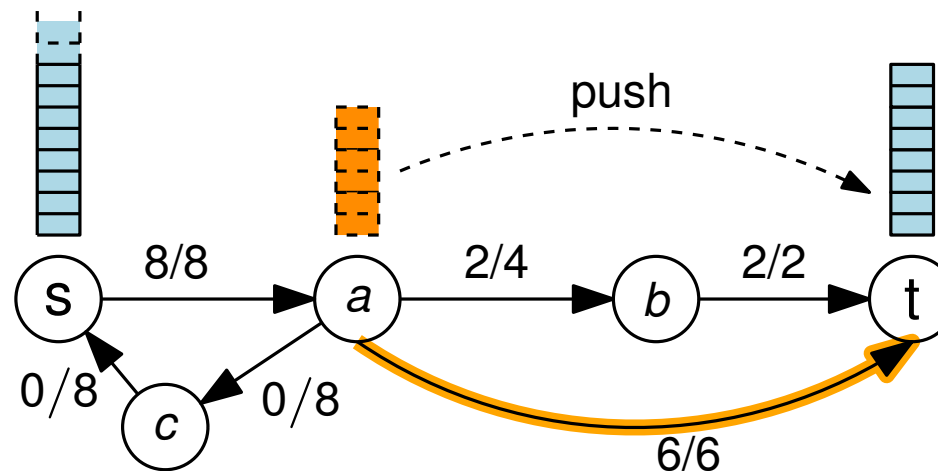
- Im Beispiel nur PUSH-Operation veranschaulicht: Drückt den Fluss zum nächsten Knoten und erzeugt dort gegebenenfalls Überschuss, der zurück geführt werden muss.
- RELABEL-Operation garantiert, dass Fluss in die richtige Richtung gedrückt wird.
 - im Beispiel darf der Fluss nicht von a über c nach s zurück gedrückt werden.

- Basiert nicht auf erhöhenden Wegen sondern auf zwei Operationen: PUSH und RELABEL.
- Fluss in Zwischenschritten nicht unbedingt gültig (Flusserhaltung nicht garantiert).



- Im Beispiel nur PUSH-Operation veranschaulicht: Drückt den Fluss zum nächsten Knoten und erzeugt dort gegebenenfalls Überschuss, der zurück geführt werden muss.
- RELABEL-Operation garantiert, dass Fluss in die richtige Richtung gedrückt wird.
 - im Beispiel darf der Fluss nicht von a über c nach s zurück gedrückt werden.

- Basiert nicht auf erhöhenden Wegen sondern auf zwei Operationen: PUSH und RELABEL.
- Fluss in Zwischenschritten nicht unbedingt gültig (Flusserhaltung nicht garantiert).



- Im Beispiel nur PUSH-Operation veranschaulicht: Drückt den Fluss zum nächsten Knoten und erzeugt dort gegebenenfalls Überschuss, der zurück geführt werden muss.
- RELABEL-Operation garantiert, dass Fluss in die richtige Richtung gedrückt wird.
 - im Beispiel darf der Fluss nicht von a über c nach s zurück gedrückt werden.

Anpassung

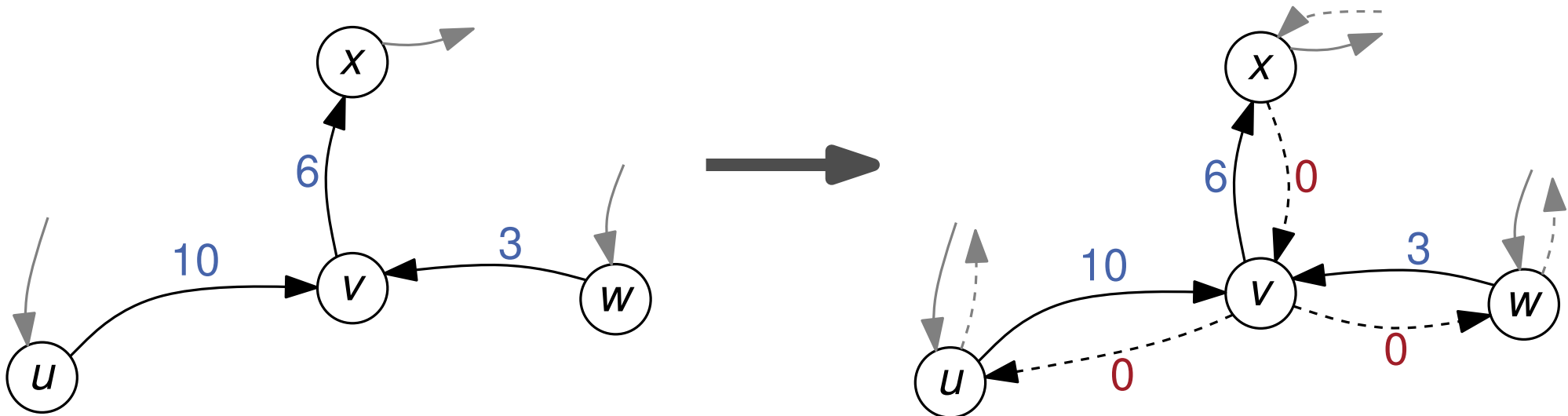
Gegeben: Netzwerk (D, s, t, c)

Erweitere c von $c: E \rightarrow \mathbb{R}_0^+$ auf $c: V \times V \rightarrow \mathbb{R}_0^+$, indem

$$c(v, w) := \begin{cases} \text{bisheriger Wert} & (v, w) \in E \\ 0 & (v, w) \notin E \end{cases}$$

Konstruiere aus $D = (V, E)$ neuen Graphen $D' = (V, E')$:

$$E' := E \cup \{(v, w) \in V \times V \mid (w, v) \in E \text{ und } (v, w) \notin E\}$$



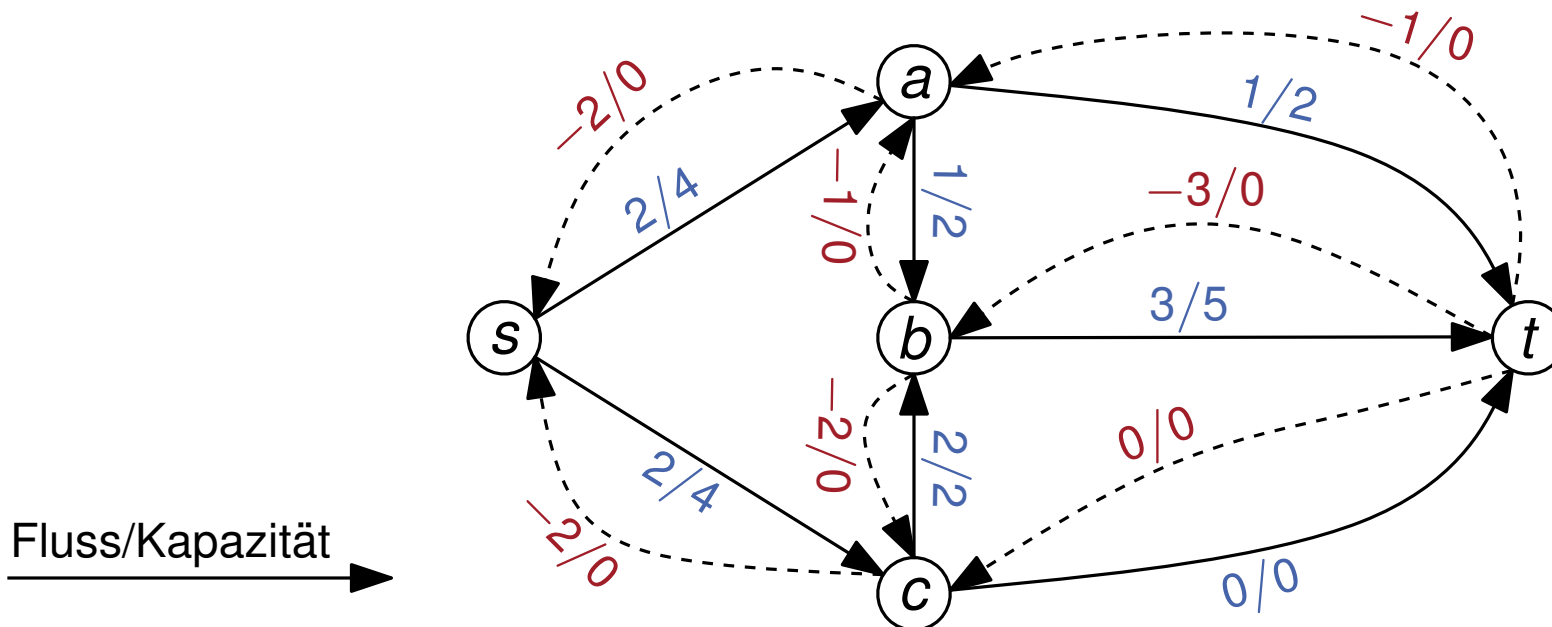
$(u, v) \in V \times V$ mit $(u, v) \notin E$ und $(v, u) \notin E$ werden nicht dargestellt.

Erweiterte Flussdefinition

Definition: Gegeben ein Netzwerk (D, s, t, c) mit angepasster Gewichtsfunktion, dann ist ein *Fluss* eine Abbildung $f: V \times V \rightarrow \mathbb{R}$ mit

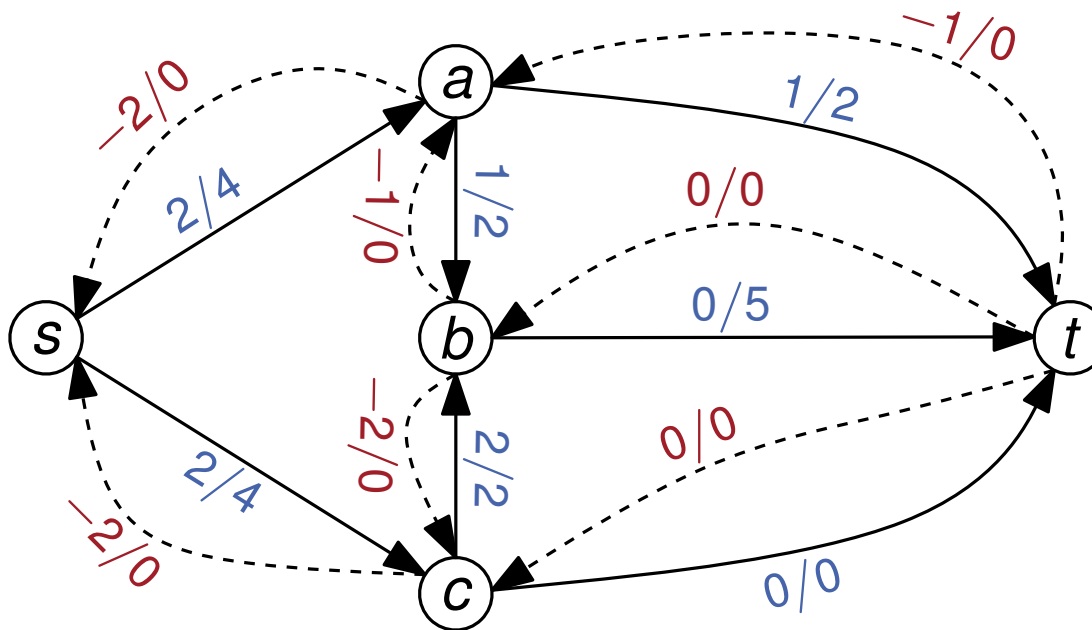
1. *Kapazitätsbedingung:* für alle $(v, w) \in V \times V$ gilt $f(v, w) \leq c(v, w)$
2. *Antisymmetrie:* für alle $(v, w) \in V \times V$ gilt $f(v, w) = -f(w, v)$
3. *Flusserhaltung:* für alle $v \in V \setminus \{s, t\}$ gilt $\sum_{u \in V} f(u, v) = 0$

Wert eines Flusses: $w(f) = \sum_{v \in V} f(s, v) = \sum_{v \in V} f(v, t)$



Definition: Ein *Präfluss* ist eine Abbildung $f: V \times V \rightarrow \mathbb{R}$, welche die Kapazitätsbedingung und die Antisymmetriebedingung erfüllt sowie

$$\text{für alle } v \in V \setminus \{s\} \quad \sum_{u \in V} f(u, v) \geq 0$$



In Knoten b fließt mehr hinein als hinaus.

Flussüberschuss und Restkapazität

Definition: Sei f ein Präfluss.

Flussüberschuss: $e(v) := \sum_{u \in V} f(u, v)$ mit $v \in V \setminus \{t\}$

Restkapazität: Abbildung $r_f: E' \rightarrow \mathbb{R}$ sodass $\forall (u, v) \in E'$ gilt $r_f(u, v) := c(u, v) - f(u, v)$

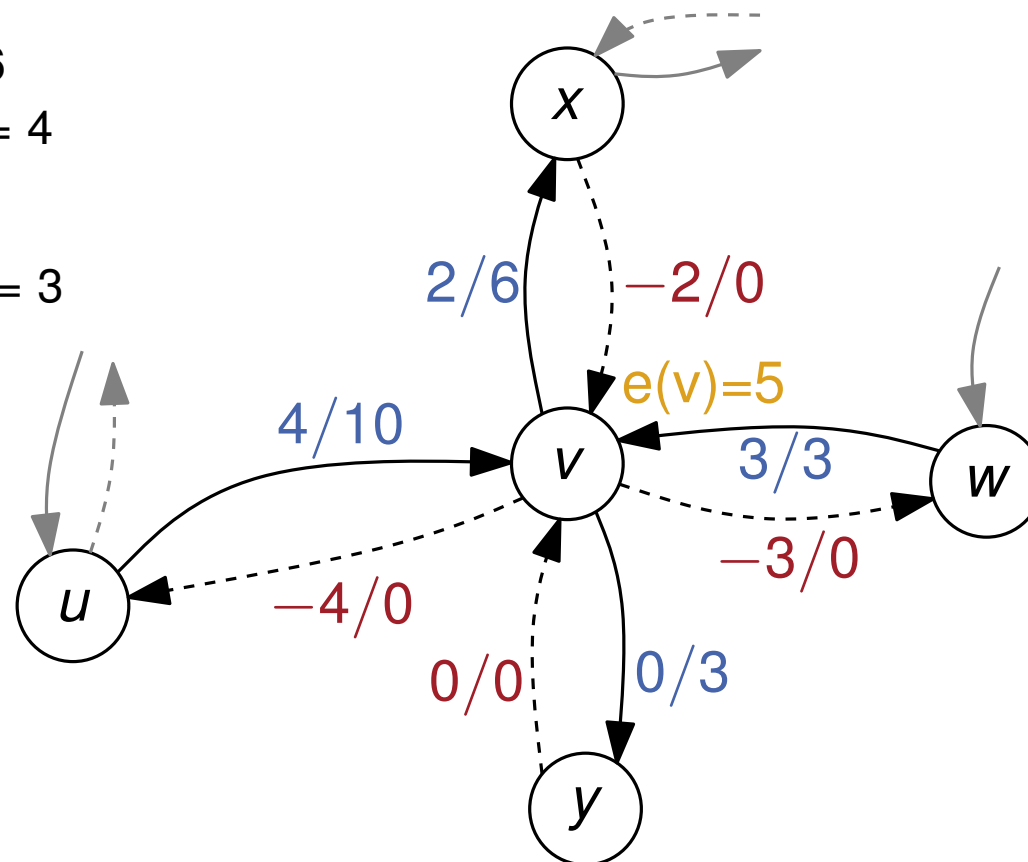
Beispiel:

$$r_f(u, v) = 10 - 4 = 6$$

$$r_f(v, u) = 0 - (-4) = 4$$

$$r_f(w, v) = 3 - 3 = 0$$

$$r_f(v, w) = 0 - (-3) = 3$$

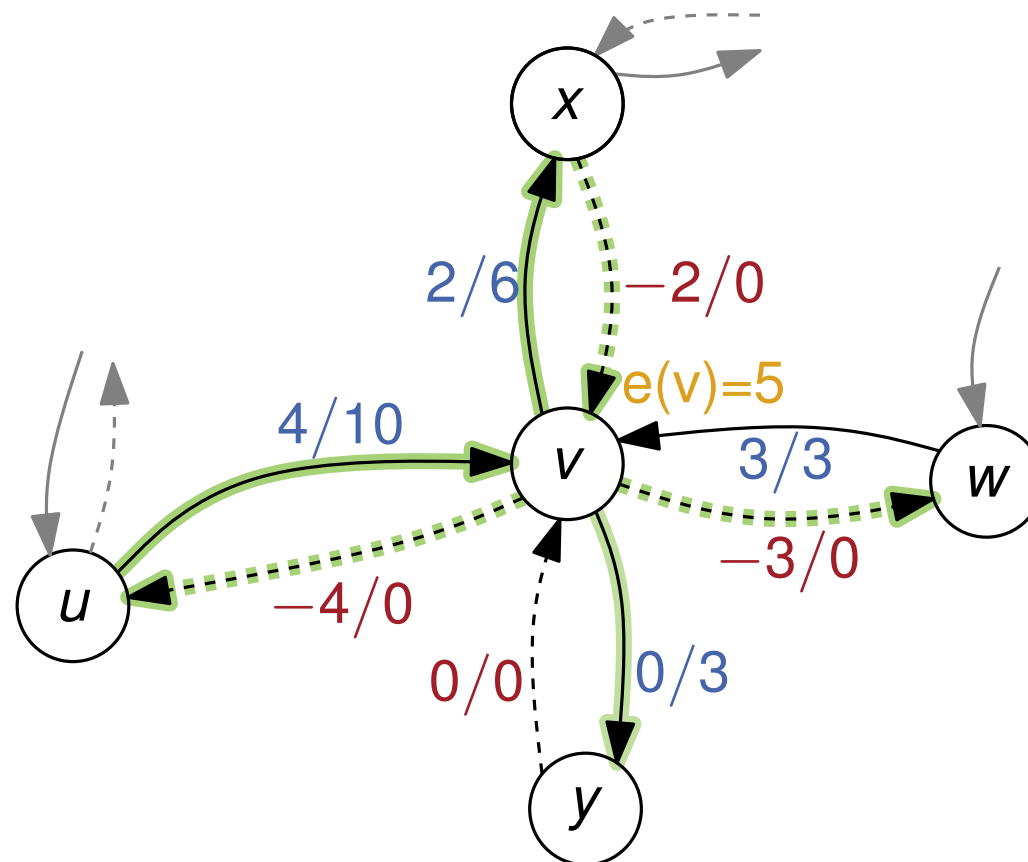


$$e(v) = 4 + 3 - 2 = 5$$

Flussüberschuss und Restkapazität

Definition: Eine Kante $(u, v) \in E'$ heißt **Residualkante** bezüglich eines Präflusses f , falls $r_f(u, v) > 0$.

Der *Residualgraph* zu f ist gegeben durch $D_f(V, E_f)$ mit $E_f := \{(u, v) \in E' \mid r_f(u, v) > 0\}$



Definition: Eine Kante $(u, v) \in E$ heißt

- *nicht saturiert*, falls $0 \leq f(u, v) < c(u, v)$, und
- *nicht leer*, falls $0 < f(u, v) \leq c(u, v)$

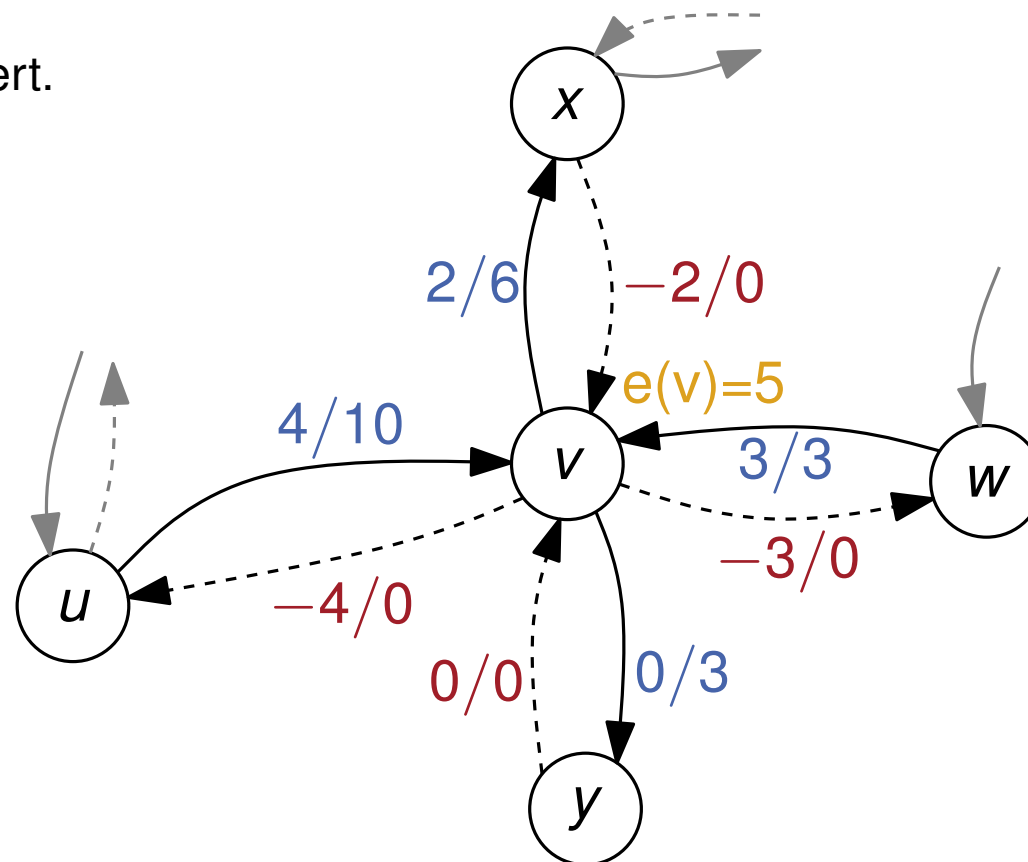
Beispiel:

(u, v) ist nicht saturiert.

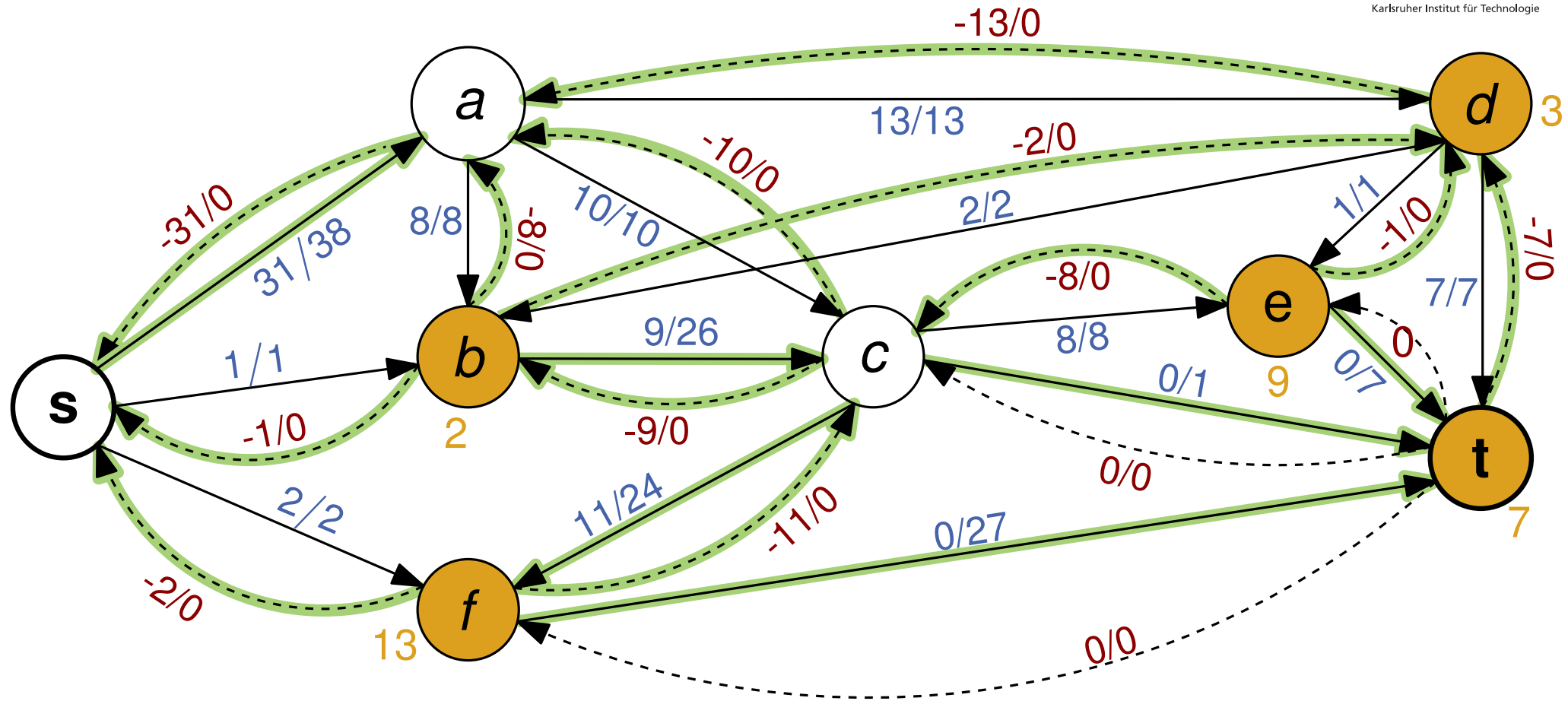
(w, v) ist saturiert.

(u, v) ist nicht leer.

(v, y) ist leer.



Beispiel



— Ursprüngliche Kanten

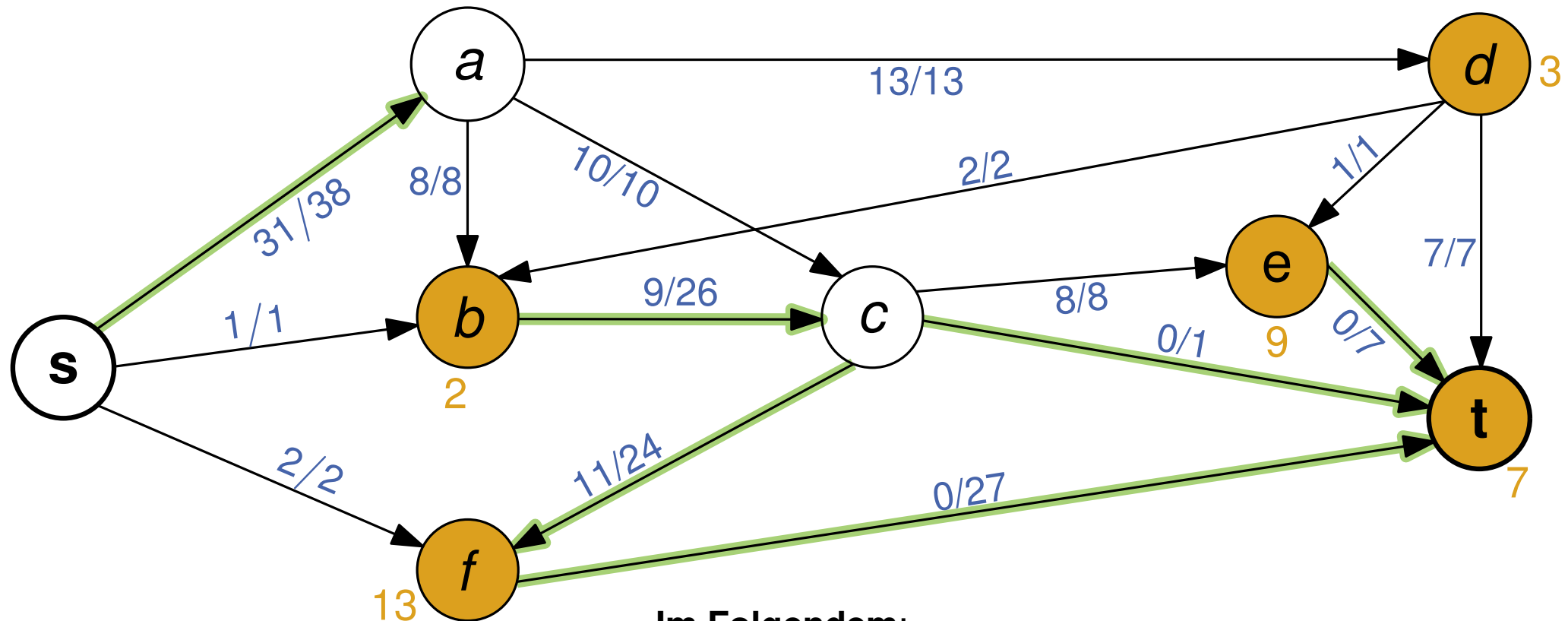
- - Hinzugefügte Kanten

— Residualkanten

Fluss/Kapazität

Überschuss

Beispiel



Im Folgendem:

Für bessere Übersicht werden häufig nur ursprüngliche Kanten gezeichnet.

— Ursprüngliche Kanten

- - - Hinzugefügte Kanten

— Residualkanten

Fluss/Kapazität →

■ Überschuss

Definition: Eine Abbildung $dist: V \rightarrow \mathbb{N}_0 \cup \{\infty\}$ heißt *zulässige Markierung* bzgl. eines Präflusses f , falls:

- $dist(s) = |V|$ und $dist(t) = 0$ und
- für alle $v \in V \setminus \{t\}$ und alle $(v, w) \in E_f$ gilt $dist(v) \leq dist(w) + 1$

Ein Knoten $v \in V \setminus \{t\}$ heißt *aktiv* im Laufe des Algorithmus, wenn $e(v) > 0$ und $dist(v) < \infty$.

Erinnerung: $e(v)$ ist der Flussüberschuss von v .

- Zu Beginn wird $dist(s) := |V|$ und $dist(v) := 0$ für alle $v \in V \setminus \{s\}$ gesetzt.
- $dist(v)$ wird geändert, aber stets zulässig gehalten.
- Es gilt stets:
 - $dist(s) = |V|$
 - Falls $dist(v) < |V|$ für $v \in V$, so ist $dist(v)$ eine untere Schranke für den Abstand von v zu t im Residualgraph D_f .
 - Falls $dist(v) > |V|$, so ist t von v in D_f nicht erreichbar und $dist(v) - |V|$ ist untere Schranke für Abstand von v zu s in D_f .

PUSH-Operation

Operation **PUSH**(D, f, v, w)

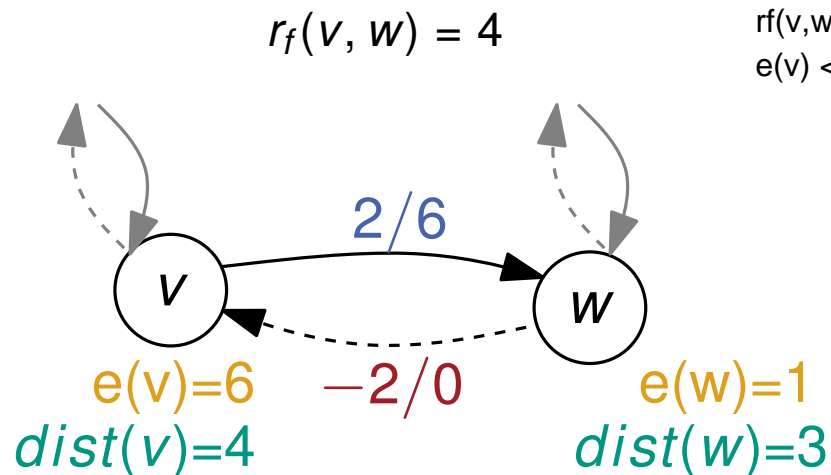
Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $dist(v) = dist(w) + 1$.

Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

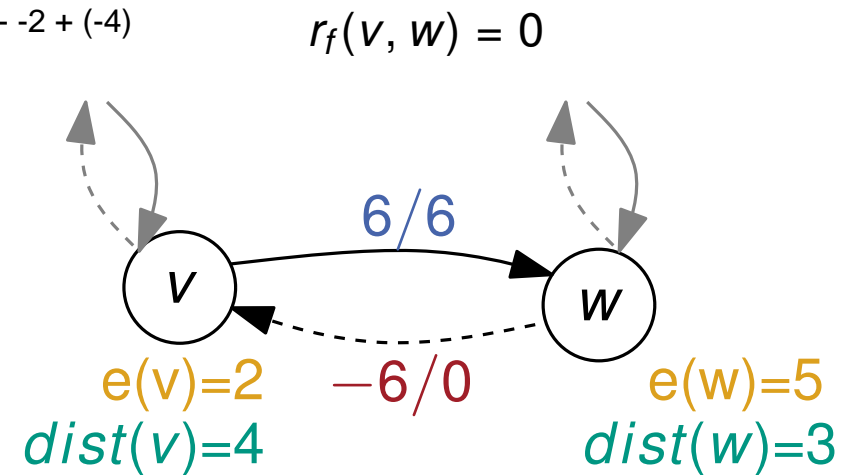
- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta, f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta, r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta, e(w) \leftarrow e(w) + \Delta$

Beispiel:

```
delta <- min{6, 4}
f(v,w) <- 2+4 , f(w,v) <- -2 -4
rf(v,w) <- rf(v,w)-4 , rf(w,v) <- -2 + (-4)
e(v) <- 6 - 4 , e(w) <- 1 + 4
```



PUSH(v, w)



RELABEL-Operation

Operation $\text{RELABEL}(D, f, v, \text{dist})$

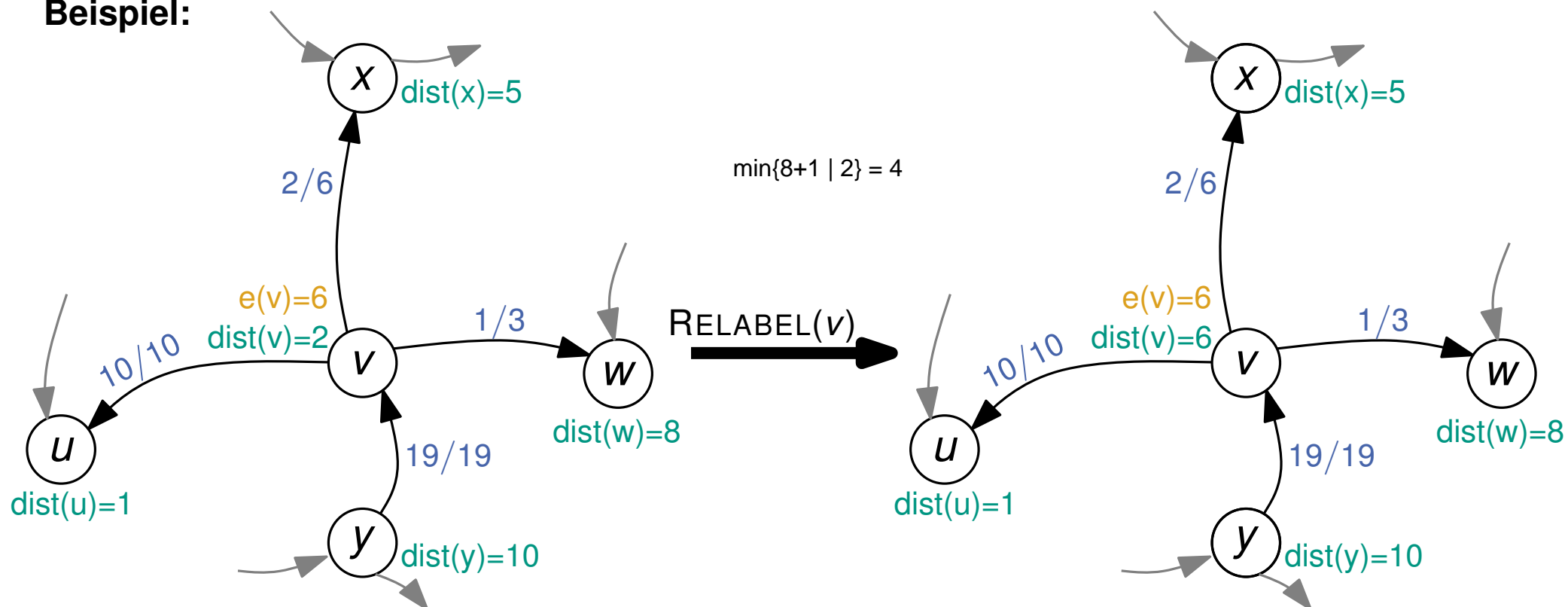
Vorbedingung: v ist aktiv und für alle w mit $r_f(v, w) > 0$ gilt $\text{dist}(v) \leq \text{dist}(w)$

Effekt: $\text{dist}(v)$ wird erhöht.

$$\text{dist}(v) = \begin{cases} \infty, & \text{falls } \{w \mid r_f(v, w) > 0\} = \emptyset, \\ \min\{\text{dist}(w) + 1 \mid r_f(v, w) > 0\} & \text{sonst.} \end{cases}$$

ist Teiler von

Beispiel:



Algorithmus von Goldberg und Tarjan

Für alle $(v, w) \in V \times V$ mit $(v, w) \notin E$ setze:

- $c(v, w) \leftarrow 0$

Für alle $(v, w) \in V \times V$ setze:

- $f(v, w) \leftarrow 0$

- $r_f(v, w) \leftarrow c(v, w)$

Setze $dist(s) \leftarrow |V|$

Für alle $v \in V \setminus \{s\}$ setze:

- $f(s, v) \leftarrow c(s, v), r_f(s, v) \leftarrow 0$

- $f(v, s) \leftarrow -c(s, v), r_f(v, s) \leftarrow c(v, s) - f(v, s)$

- $dist(v) \leftarrow 0$

- $e(v) \leftarrow c(s, v)$

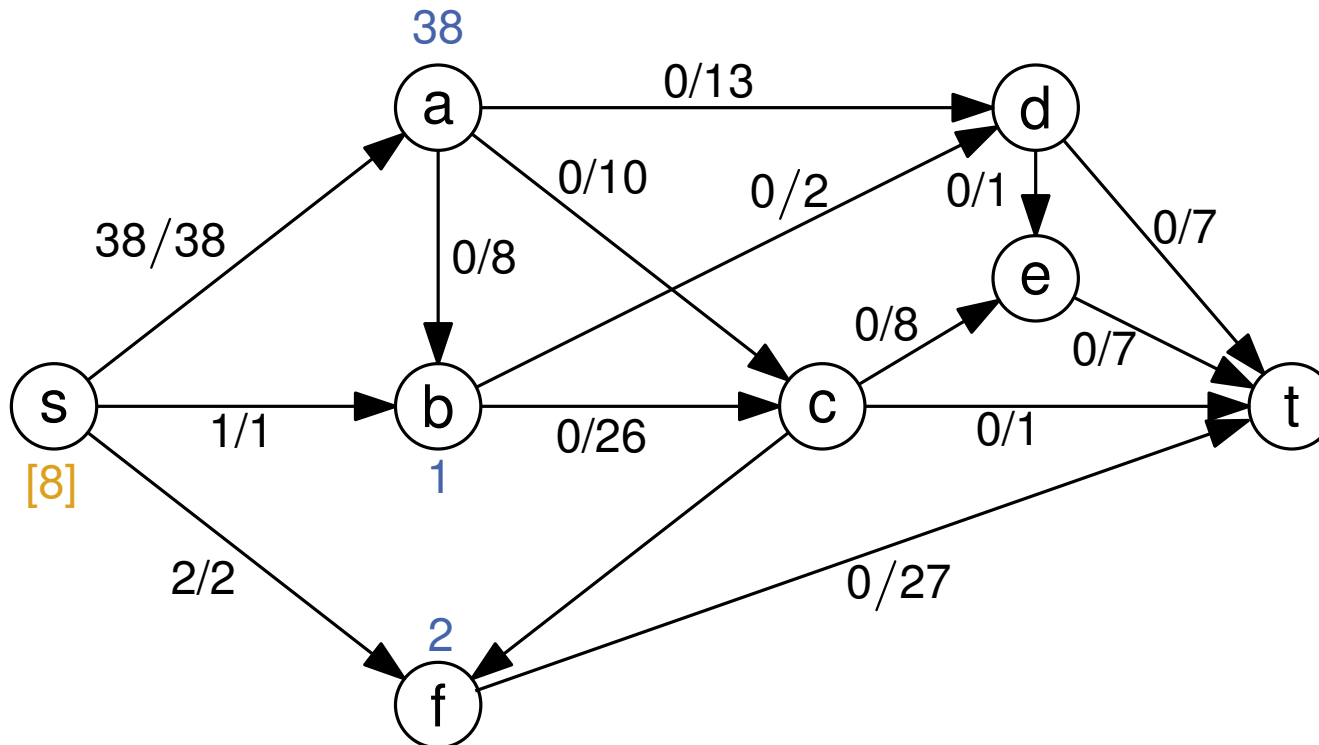
Eingabe: Netzwerk (D, s, t, c) mit $D = (V, E)$ und $c : E \rightarrow \mathbb{R}_0^+$

Ausgabe: Maximaler Fluss f .

Solange es aktiven Knoten gibt:

- Wähle beliebigen aktiven Knoten v .
- Führe für v eine zulässige Operation PUSH oder RELABEL aus.

Beispiel



Ausgeführte Operation:
Initialisierung

Anstehende Operation:
RELABEL(a)

■ $dist(v)$

■ Überschuss $e(v)$

Operation PUSH(D, f, v, w)

Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $dist(v) = dist(w) + 1$.

Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta, f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta, r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta, e(w) \leftarrow e(w) + \Delta$

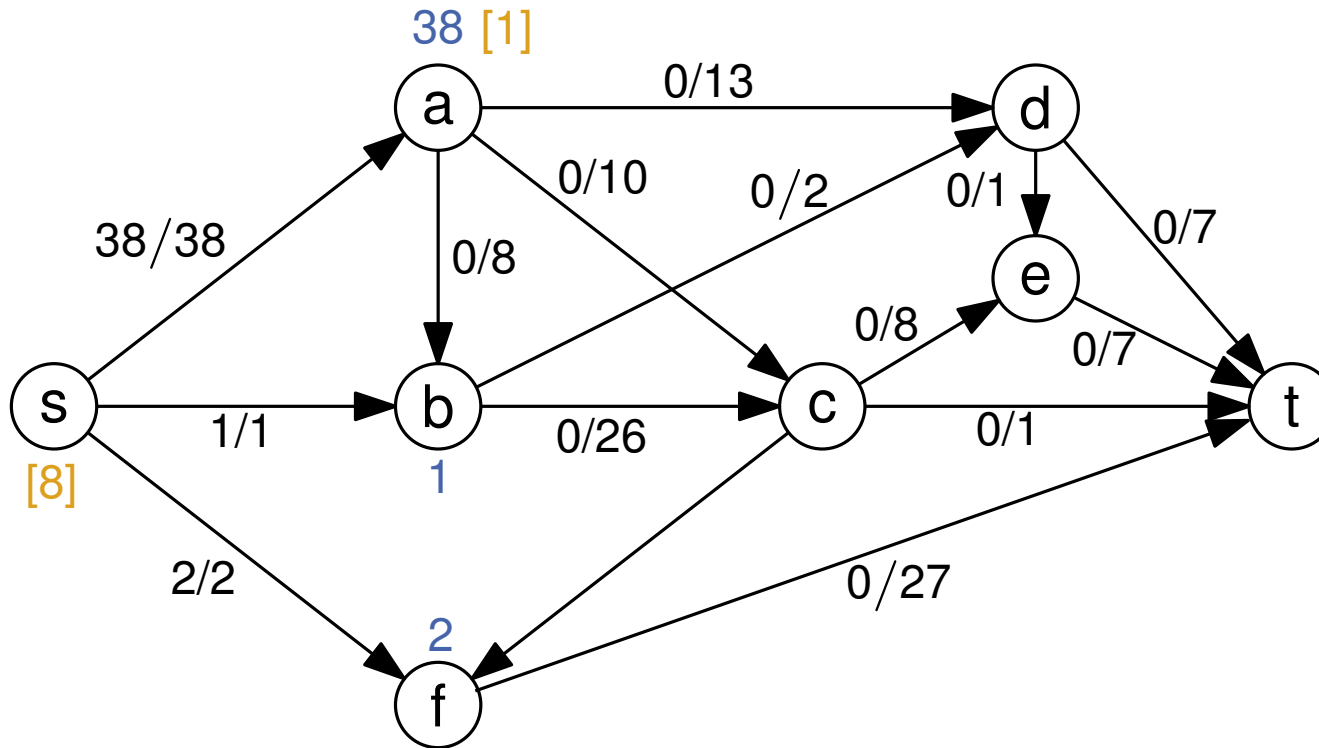
Operation RELABEL($D, f, v, dist$)

Vorbedingung: v ist aktiv und für alle w mit $r_f(v, w) > 0$ gilt $dist(v) \leq dist(w)$

Effekt: $dist(v)$ wird erhöht.

$$dist(v) = \begin{cases} \infty, & \text{falls } \{w \mid r_f(v, w) > 0\} = \emptyset, \\ \min\{dist(w) + 1 \mid r_f(v, w) > 0\} & \text{sonst.} \end{cases}$$

Beispiel



■ $dist(v)$

■ Überschuss $e(v)$

Operation $PUSH(D, f, v, w)$

Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $dist(v) = dist(w) + 1$.

Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta, f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta, r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta, e(w) \leftarrow e(w) + \Delta$

Operation $RELABEL(D, f, v, dist)$

Vorbedingung: v ist aktiv und für alle w mit $r_f(v, w) > 0$ gilt $dist(v) \leq dist(w)$

Effekt: $dist(v)$ wird erhöht.

$$dist(v) = \begin{cases} \infty, & \text{falls } \{w \mid r_f(v, w) > 0\} = \emptyset, \\ \min\{dist(w) + 1 \mid r_f(v, w) > 0\} & \text{sonst.} \end{cases}$$

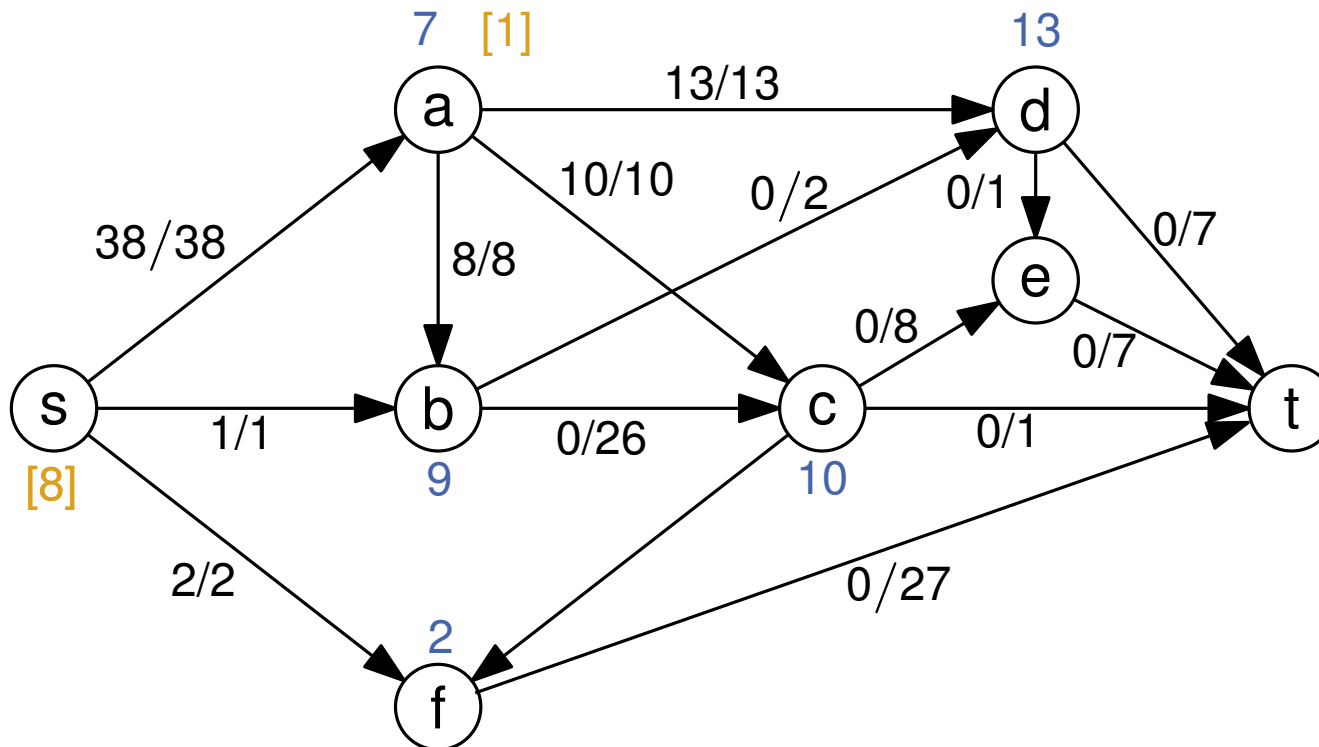
Ausgeführte Operationen:

Initialisierung
 $RELABEL(a)$

Anstehende Operationen:

$PUSH(a, b)$ mit $\Delta = 8$
 $PUSH(a, c)$ mit $\Delta = 10$
 $PUSH(a, d)$ mit $\Delta = 13$

Beispiel



Ausgeführte Operationen:

Initialisierung

RELABEL(a)

PUSH(a, b) mit $\Delta = 8$

PUSH(a, c) mit $\Delta = 10$

PUSH(a, d) mit $\Delta = 13$

Anstehende Operation:

RELABEL(a)

■ $dist(v)$

■ Überschuss $e(v)$

Operation PUSH(D, f, v, w)

Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $dist(v) = dist(w) + 1$.

Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta, f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta, r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta, e(w) \leftarrow e(w) + \Delta$

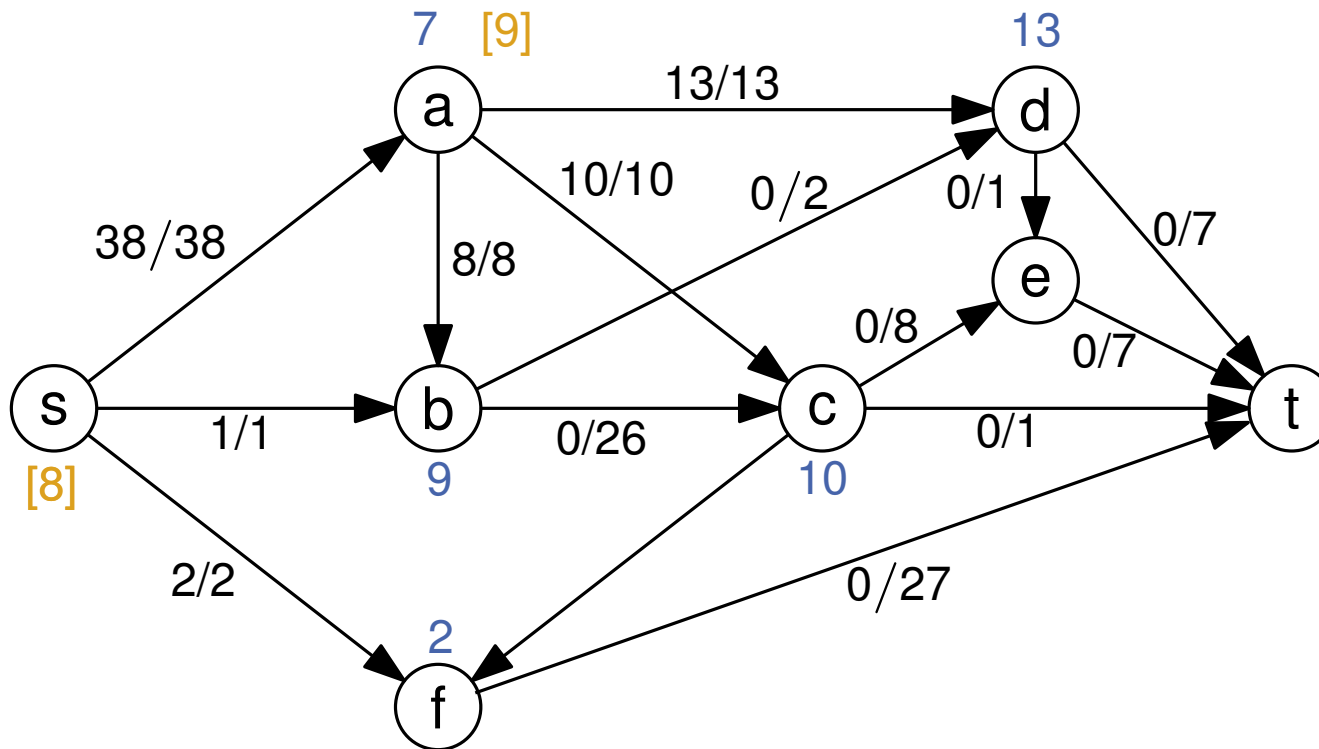
Operation RELABEL($D, f, v, dist$)

Vorbedingung: v ist aktiv und für alle w mit $r_f(v, w) > 0$ gilt $dist(v) \leq dist(w)$

Effekt: $dist(v)$ wird erhöht.

$$dist(v) = \begin{cases} \infty, & \text{falls } \{w \mid r_f(v, w) > 0\} = \emptyset, \\ \min\{dist(w) + 1 \mid r_f(v, w) > 0\} & \text{sonst.} \end{cases}$$

Beispiel



Ausgeführte Operationen:

Initialisierung

RELABEL(a)

PUSH(a, b) mit $\Delta = 8$

PUSH(a, c) mit $\Delta = 10$

PUSH(a, d) mit $\Delta = 13$

RELABEL(a)

Anstehende Operation:

PUSH(a, s) mit $\Delta = 7$

■ $dist(v)$

■ Überschuss $e(v)$

Operation PUSH(D, f, v, w)

Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $dist(v) = dist(w) + 1$.

Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta, f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta, r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta, e(w) \leftarrow e(w) + \Delta$

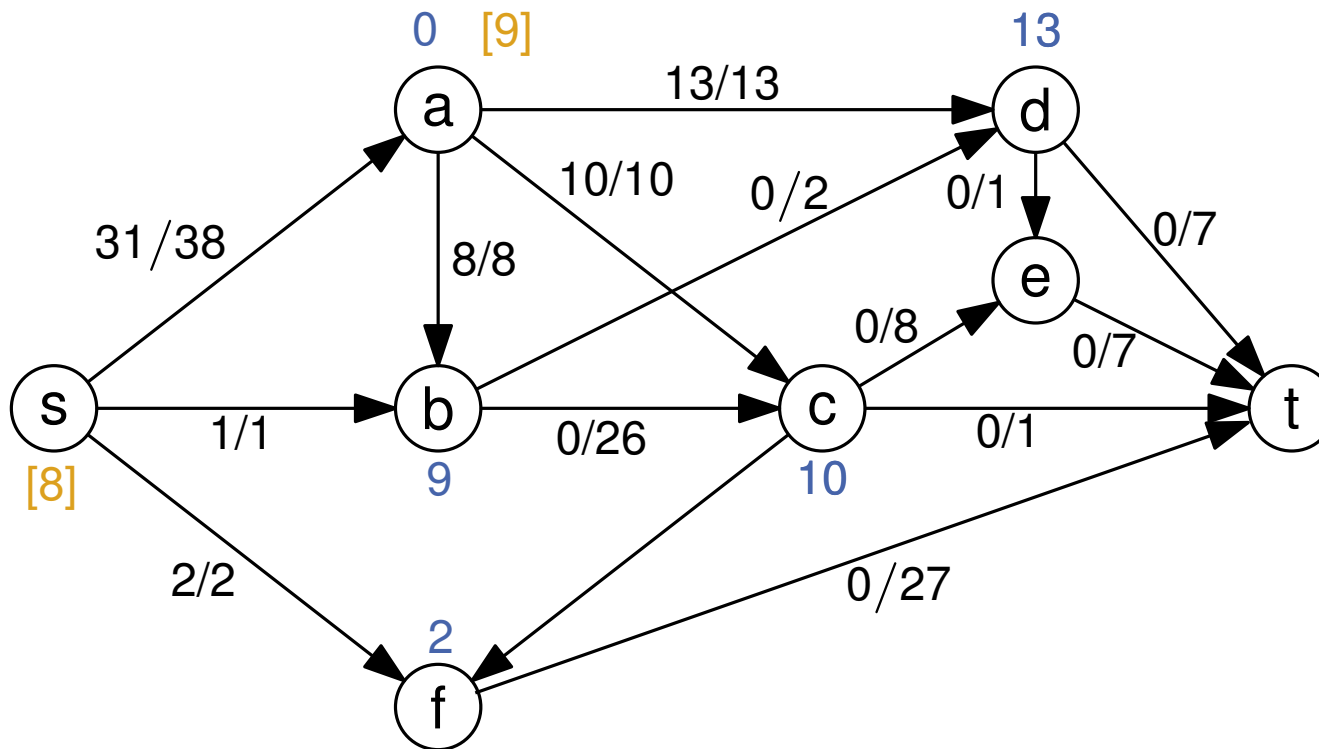
Operation RELABEL($D, f, v, dist$)

Vorbedingung: v ist aktiv und für alle w mit $r_f(v, w) > 0$ gilt $dist(v) \leq dist(w)$

Effekt: $dist(v)$ wird erhöht.

$$dist(v) = \begin{cases} \infty, & \text{falls } \{w \mid r_f(v, w) > 0\} = \emptyset, \\ \min\{dist(w) + 1 \mid r_f(v, w) > 0\} & \text{sonst.} \end{cases}$$

Beispiel



Ausgeführte Operationen:

Initialisierung

RELABEL(a)

PUSH(a, b) mit $\Delta = 8$

PUSH(a, c) mit $\Delta = 10$

PUSH(a, d) mit $\Delta = 13$

RELABEL(a)

PUSH(a, s) mit $\Delta = 7$

■ $dist(v)$

■ Überschuss $e(v)$

Operation PUSH(D, f, v, w)

Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $dist(v) = dist(w) + 1$.

Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta, f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta, r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta, e(w) \leftarrow e(w) + \Delta$

Operation RELABEL($D, f, v, dist$)

Vorbedingung: v ist aktiv und für alle w mit $r_f(v, w) > 0$ gilt $dist(v) \leq dist(w)$

Effekt: $dist(v)$ wird erhöht.

$$dist(v) = \begin{cases} \infty, & \text{falls } \{w \mid r_f(v, w) > 0\} = \emptyset, \\ \min\{dist(w) + 1 \mid r_f(v, w) > 0\} & \text{sonst.} \end{cases}$$

Klausurrelevant?!

Korrektheitsbeweis des Algorithmus von Goldberg und Tarjan

1. Schritt: Wenn Algorithmus terminiert und die Markierungen endlich bleiben, dann ist das Ergebnis ein Maximalfluss.

- a) Solange aktiver Knoten vorhanden, kann Operation PUSH oder Operation RELABEL angewendet werden.
- b) Es gilt stets: f ist Präfluss und $dist$ ist bezüglich f zulässige Markierung.
- c) t ist im Residualgraph D_f des Präflusses f von s aus nicht erreichbar.

2. Schritt: Algorithmus terminiert und Markierungen bleiben endlich:

- a) Finde obere Schranke für $dist$.
- b) Finde obere Schranke für Anzahl Aufrufe von RELABEL.
- c) Finde obere Schranke für Anzahl Aufrufe von PUSH.

Bezeichne f die Abbildung, die schrittweise konstruiert wird.

Lemma 4.20: Sei f ein Präfluss auf D , die Funktion $dist$ eine bezüglich f zulässige Markierung auf V und $v \in V$ ein aktiver Knoten. Dann ist entweder eine PUSH-Operation von v oder eine RELABEL-Operation von v zulässig.

Beweis:

Operation PUSH(D, f, v, w)

Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $dist(v) = dist(w) + 1$.

Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta, f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta, r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta, e(w) \leftarrow e(w) + \Delta$

Operation RELABEL($D, f, v, dist$)

Vorbedingung: v ist aktiv und für alle w mit $r_f(v, w) > 0$ gilt $dist(v) \leq dist(w)$

Effekt: $dist(v)$ wird erhöht.

$$dist(v) = \begin{cases} \infty, & \text{falls } \{w \mid r_f(v, w) > 0\} = \emptyset, \\ \min\{dist(w) + 1 \mid r_f(v, w) > 0\} & \text{sonst.} \end{cases}$$

Lemma 4.20: Sei f ein Präfluss auf D , die Funktion $dist$ eine bezüglich f zulässige Markierung auf V und $v \in V$ ein aktiver Knoten. Dann ist entweder eine PUSH-Operation von v oder eine RELABEL-Operation von v zulässig.

Beweis: Erinnerung: Knoten v ist aktiv wenn $e(v) > 0$ und $dist(v) < \infty$.

$dist(v)$ nach Annahme zulässig, d.h. insbesondere:

$$dist(v) \leq dist(w) + 1 \text{ für alle } w \text{ mit } r_f(v, w) > 0.$$

Operation PUSH(D, f, v, w)

Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $dist(v) = dist(w) + 1$.

Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta, f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta, r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta, e(w) \leftarrow e(w) + \Delta$

Operation RELABEL($D, f, v, dist$)

Vorbedingung: v ist aktiv und für alle w mit $r_f(v, w) > 0$ gilt $dist(v) \leq dist(w)$

Effekt: $dist(v)$ wird erhöht.

$$dist(v) = \begin{cases} \infty, & \text{falls } \{w \mid r_f(v, w) > 0\} = \emptyset, \\ \min\{dist(w) + 1 \mid r_f(v, w) > 0\} & \text{sonst.} \end{cases}$$

Lemma 4.20: Sei f ein Präfluss auf D , die Funktion $dist$ eine bezüglich f zulässige Markierung auf V und $v \in V$ ein aktiver Knoten. Dann ist entweder eine PUSH-Operation von v oder eine RELABEL-Operation von v zulässig.

Beweis: Erinnerung: Knoten v ist aktiv wenn $e(v) > 0$ und $dist(v) < \infty$.

$dist(v)$ nach Annahme zulässig, d.h. insbesondere:

$$dist(v) \leq dist(w) + 1 \text{ für alle } w \text{ mit } r_f(v, w) > 0.$$

Falls $PUSH(v, w)$ für kein w mit $r_f(v, w) > 0$ zulässig ist, dann gilt:

$$dist(v) \leq dist(w) \quad \text{für alle } w \text{ mit } r_f(v, w) > 0.$$

→ RELABEL(v) ist zulässig.

Operation $PUSH(D, f, v, w)$

Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $dist(v) = dist(w) + 1$.

Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta, f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta, r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta, e(w) \leftarrow e(w) + \Delta$

Operation $RELABEL(D, f, v, dist)$

Vorbedingung: v ist aktiv und für alle w mit $r_f(v, w) > 0$ gilt $dist(v) \leq dist(w)$

Effekt: $dist(v)$ wird erhöht.

$$dist(v) = \begin{cases} \infty, & \text{falls } \{w \mid r_f(v, w) > 0\} = \emptyset, \\ \min\{dist(w) + 1 \mid r_f(v, w) > 0\} & \text{sonst.} \end{cases}$$

Lemma 4.21: Während des Ablaufs des Algorithmus von Goldberg und Tarjan ist f stets ein Präfluss und $dist$ stets eine bezüglich f zulässige Markierung.

Lemma 4.21: Während des Ablaufs des Algorithmus von Goldberg und Tarjan ist f stets ein Präfluss und $dist$ stets eine bezüglich f zulässige Markierung.

Beweis durch Induktion über Anzahl k zulässiger Operationen.

IA: Behauptung ist aufgrund der Initialisierung richtig.

→ Annahme: *Behauptung gilt nach k -ter Operation.*

Für alle $(v, w) \in V \times V$ mit $(v, w) \notin E$ setze:

- $c(v, w) \leftarrow 0$

Für alle $(v, w) \in V \times V$ setze:

- $f(v, w) \leftarrow 0$

- $r_f(v, w) \leftarrow c(v, w)$

Setze $dist(s) \leftarrow |V|$

Für alle $v \in V \setminus \{s\}$ setze:

- $f(s, v) \leftarrow c(s, v), r_f(s, v) \leftarrow 0$

- $f(v, s) \leftarrow -c(s, v), r_f(v, s) \leftarrow c(v, s) - f(v, s)$

- $dist(v) \leftarrow 0$

- $e(v) \leftarrow c(s, v)$

$dist$ heißt *zulässig* wenn

- $dist(s) = |V|$ und $dist(t) = 0$ und

- $\forall v \in V \setminus \{t\}$ und $\forall (v, w) \in E_f: dist(v) \leq dist(w) + 1$

Lemma 4.21: Während des Ablaufs des Algorithmus von Goldberg und Tarjan ist f stets ein Präfluss und $dist$ stets eine bezüglich f zulässige Markierung.

Beweis durch Induktion über Anzahl k zulässiger Operationen.

IA: Behauptung ist aufgrund der Initialisierung richtig.

→ Annahme: *Behauptung gilt nach k -ter Operation.*

IS: $(k+1)$ -te Operation ist $PUSH(v, w)$: $dist$ bleibt unverändert, f ändert sich

$dist$ heißt *zulässig* wenn

- $dist(s) = |V|$ und $dist(t) = 0$ und
- $\forall v \in V \setminus \{t\}$ und $\forall (v, w) \in E_f: dist(v) \leq dist(w) + 1$

Operation $PUSH(D, f, v, w)$

Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $dist(v) = dist(w) + 1$.

Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta, f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta, r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta, e(w) \leftarrow e(w) + \Delta$

Lemma 4.21: Während des Ablaufs des Algorithmus von Goldberg und Tarjan ist f stets ein Präfluss und $dist$ stets eine bezüglich f zulässige Markierung.

Beweis durch Induktion über Anzahl k zulässiger Operationen.

IA: Behauptung ist aufgrund der Initialisierung richtig.

→ Annahme: *Behauptung gilt nach k -ter Operation.*

IS: $(k+1)$ -te Operation ist $PUSH(v, w)$: $dist$ bleibt unverändert, f ändert sich

1. Präfluss bleibt offensichtlich erhalten.

2. Betrachte Zustand nach Ausführung von $PUSH(v, w)$:

1. Fall, $r_f(v, w) = 0$: $dist$ bleibt trivialerweise zulässig.

2. Fall, $r_f(w, v) > 0$: $dist$ bleibt zulässig, denn

für Ausführung von $PUSH$ muss gelten: $dist(w) = dist(v) - 1 \leq dist(v) + 1$

$dist$ heißt *zulässig* wenn

- $dist(s) = |V|$ und $dist(t) = 0$ und
- $\forall v \in V \setminus \{t\}$ und $\forall (v, w) \in E_f: dist(v) \leq dist(w) + 1$

Operation $PUSH(D, f, v, w)$

Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $dist(v) = dist(w) + 1$.

Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta, f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta, r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta, e(w) \leftarrow e(w) + \Delta$

Lemma 4.21: Während des Ablaufs des Algorithmus von Goldberg und Tarjan ist f stets ein Präfluss und $dist$ stets eine bezüglich f zulässige Markierung.

Beweis durch Induktion über Anzahl k zulässiger Operationen.

IA: Behauptung ist aufgrund der Initialisierung richtig.

→ Annahme: *Behauptung gilt nach k -ter Operation.*

IS: $(k+1)$ -te Operation ist RELABEL(v): $dist$ wird verändert, f bleibt unverändert

$dist$ heißt *zulässig* wenn

- $dist(s) = |V|$ und $dist(t) = 0$ und
- $\forall v \in V \setminus \{t\}$ und $\forall (v, w) \in E_f: dist(v) \leq dist(w) + 1$

Operation RELABEL($D, f, v, dist$)

Vorbedingung: v ist aktiv und für alle w mit $r_f(v, w) > 0$ gilt $dist(v) \leq dist(w)$

Effekt: $dist(v)$ wird erhöht.

$$dist(v) = \begin{cases} \infty, & \text{falls } \{w \mid r_f(v, w) > 0\} = \emptyset, \\ \min\{dist(w) + 1 \mid r_f(v, w) > 0\} & \text{sonst.} \end{cases}$$

Lemma 4.21: Während des Ablaufs des Algorithmus von Goldberg und Tarjan ist f stets ein Präfluss und $dist$ stets eine bezüglich f zulässige Markierung.

Beweis durch Induktion über Anzahl k zulässiger Operationen.

IA: Behauptung ist aufgrund der Initialisierung richtig.

→ Annahme: *Behauptung gilt nach k -ter Operation.*

IS: $(k+1)$ -te Operation ist RELABEL(v): $dist$ wird verändert, f bleibt unverändert

Vor RELABEL(v) gilt:

$$dist(v) \leq dist(w) \text{ für alle } w \text{ mit } r_f(v, w) > 0.$$

RELABEL(v) setzt:

$$dist(v) = \min\{dist(w) + 1 \mid r_f(v, w) > 0\}$$

Folglich: $dist$ wieder zulässig.

Operation RELABEL($D, f, v, dist$)

Vorbedingung: v ist aktiv und für alle w mit $r_f(v, w) > 0$ gilt $dist(v) \leq dist(w)$

Effekt: $dist(v)$ wird erhöht.

$$dist(v) = \begin{cases} \infty, & \text{falls } \{w \mid r_f(v, w) > 0\} = \emptyset, \\ \min\{dist(w) + 1 \mid r_f(v, w) > 0\} & \text{sonst.} \end{cases}$$

$dist$ heißt *zulässig* wenn

- $dist(s) = |V|$ und $dist(t) = 0$ und
- $\forall v \in V \setminus \{t\}$ und $\forall (v, w) \in E_f: dist(v) \leq dist(w) + 1$

Lemma 4.22: Sei f ein Präfluss und $dist$ bezüglich f zulässig. Dann ist t im Residualgraph D_f von s aus nicht erreichbar (es gibt also keinen gerichteten s - t -Weg in D_f).

Lemma 4.22: Sei f ein Präfluss und $dist$ bezüglich f zulässig. Dann ist t im Residualgraph D_f von s aus nicht erreichbar (es gibt also keinen gerichteten s - t -Weg in D_f).

Beweis: Annahme es gibt einen solchen Weg $s = v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_\ell = t$ in D_f

Es gilt also:

$$r_f(v_i, v_{i+1}) > 0 \text{ für } 0 \leq i \leq \ell - 1$$

Da $dist$ zulässig ist gilt deshalb:

$$dist(v_i) \leq dist(v_{i+1}) + 1 \text{ für } 0 \leq i \leq \ell - 1$$

$$\left. \begin{array}{l} r_f(v_i, v_{i+1}) > 0 \text{ für } 0 \leq i \leq \ell - 1 \\ dist(v_i) \leq dist(v_{i+1}) + 1 \text{ für } 0 \leq i \leq \ell - 1 \end{array} \right\} \rightarrow dist(s) \leq dist(t) + \ell$$

$dist$ heißt zulässig wenn

- $dist(s) = |V|$ und $dist(t) = 0$ und
- $\forall v \in V \setminus \{t\}$ und $\forall (v, w) \in E_f: dist(v) \leq dist(w) + 1$

Erreichbarkeit der Senke im Residualgraph

Lemma 4.22: Sei f ein Präfluss und $dist$ bezüglich f zulässig. Dann ist t im Residualgraph D_f von s aus nicht erreichbar (es gibt also keinen gerichteten s - t -Weg in D_f).

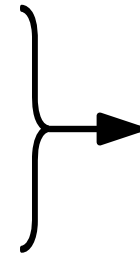
Beweis: Annahme es gibt einen solchen Weg $s = v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_\ell = t$ in D_f

Es gilt also:

$$r_f(v_i, v_{i+1}) > 0 \text{ für } 0 \leq i \leq \ell - 1$$

Da $dist$ zulässig ist gilt deshalb:

$$dist(v_i) \leq dist(v_{i+1}) + 1 \text{ für } 0 \leq i \leq \ell - 1$$



$$dist(s) \leq dist(t) + \ell$$

Aus $dist(s) \leq dist(t) + \ell$, $dist(t) = 0$ und $\ell \leq |V| - 1$ folgt:

$$dist(s) < |V|$$



$$dist(s) = |V|$$

$dist$ heißt zulässig wenn

- $dist(s) = |V|$ und $dist(t) = 0$ und
- $\forall v \in V \setminus \{t\}$ und $\forall (v, w) \in E_f: dist(v) \leq dist(w) + 1$

Satz 4.23: Falls der Algorithmus von Goldberg und Tarjan terminiert und am Ende alle Markierungen endlich sind, dann ist der konstruierte Präfluss ein Maximalfluss im Netzwerk $(D; s, t; c)$.

Satz 4.23: Falls der Algorithmus von Goldberg und Tarjan terminiert und am Ende alle Markierungen endlich sind, dann ist der konstruierte Präfluss ein Maximalfluss im Netzwerk $(D; s, t; c)$.

Beweis: Sei f Ergebnis des Algorithmus

1. Nach **Lemma 4.21** ist f Präfluss.
2. Nach **Lemma 4.20** bricht Algorithmus ab, wenn kein aktiver Knoten existiert.
3. Nach Voraussetzung und **Lemma 4.20** gilt

$$e(v) = 0 \text{ für alle } v \in V \setminus \{s, t\}$$

→ **f ist ein Fluss.**

Satz 4.23: Falls der Algorithmus von Goldberg und Tarjan terminiert und am Ende alle Markierungen endlich sind, dann ist der konstruierte Präfluss ein Maximalfluss im Netzwerk $(D; s, t; c)$.

Beweis: Sei f Ergebnis des Algorithmus

1. Nach **Lemma 4.21** ist f Präfluss.
2. Nach **Lemma 4.20** bricht Algorithmus ab, wenn kein aktiver Knoten existiert.
3. Nach Voraussetzung und **Lemma 4.20** gilt

$$e(v) = 0 \text{ für alle } v \in V \setminus \{s, t\}$$


→ **f ist ein Fluss.**

Nach **Lemma 4.22** gibt es keinen Weg von s nach t .

→ Es gibt keinen bezüglich f erhöhenden Weg von s nach t in D .

f ist Maximalfluss im Netzwerk $(D; s; t; c)$

1. Schritt: Wenn Algorithmus terminiert und die Markierungen endlich bleiben, dann ist das Ergebnis ein Maximalfluss.

- a) Solange aktiver Knoten vorhanden, kann Operation PUSH oder Operation RELABEL angewendet werden.
 - b) Es gilt stets: f ist Präfluss und $dist$ ist bezüglich f zulässige Markierung.
 - c) t ist im Residualgraph D_f des Präflusses f von s aus nicht erreichbar.
- 

2. Schritt: Algorithmus terminiert und Markierungen bleiben endlich:

- a) Finde obere Schranke für $dist$.
- b) Finde obere Schranke für Anzahl Aufrufe von RELABEL.
- c) Finde obere Schranke für Anzahl Aufrufe von PUSH.

Bezeichne f die Abbildung, die schrittweise konstruiert wird.

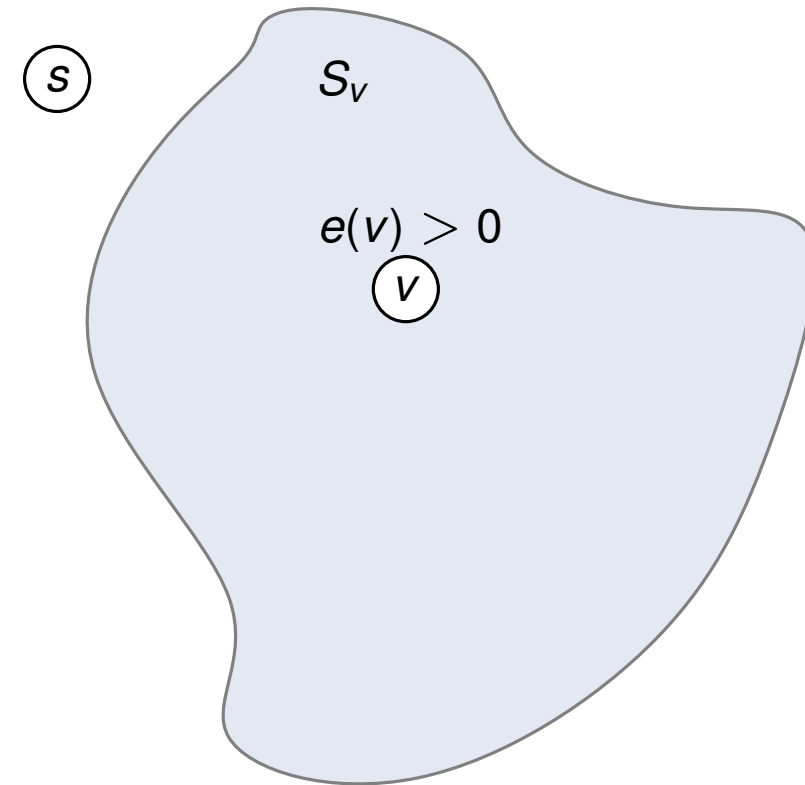
Lemma 4.24: Sei f ein Präfluss auf D . Wenn für v gilt, dass $e(v) > 0$, so ist s in D_f von v aus erreichbar.

Erreichbarkeit der Quelle im Residualgraph

Lemma 4.24: Sei f ein Präfluss auf D . Wenn für v gilt, dass $e(v) > 0$, so ist s in D_f von v aus erreichbar.

Beweis: Sei S_v die Menge der Knoten, die in D_f von v aus erreichbar sind.

Annahme: s ist nicht in S_v enthalten.



Erreichbarkeit der Quelle im Residualgraph

Lemma 4.24: Sei f ein Präfluss auf D . Wenn für v gilt, dass $e(v) > 0$, so ist s in D_f von v aus erreichbar.

Beweis: Sei S_v die Menge der Knoten, die in D_f von v aus erreichbar sind.

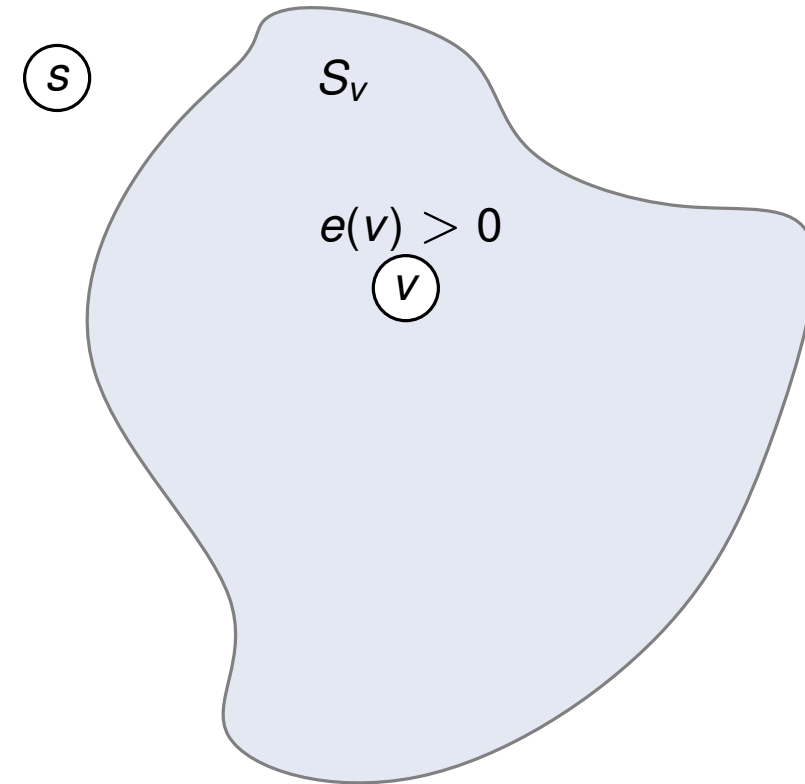
Annahme: s ist nicht in S_v enthalten.

Da f Präfluss ist und $s \notin S_v$, gilt

$$\sum_{w \in S_v} e(w) \geq 0$$

Da $v \in S_v$, gilt

$$\sum_{w \in S_v} e(w) > 0$$



Erreichbarkeit der Quelle im Residualgraph

Lemma 4.24: Sei f ein Präfluss auf D . Wenn für v gilt, dass $e(v) > 0$, so ist s in D_f von v aus erreichbar.

Beweis: Sei S_v die Menge der Knoten, die in D_f von v aus erreichbar sind.

Annahme: s ist nicht in S_v enthalten.

Da f Präfluss ist und $s \notin S_v$, gilt

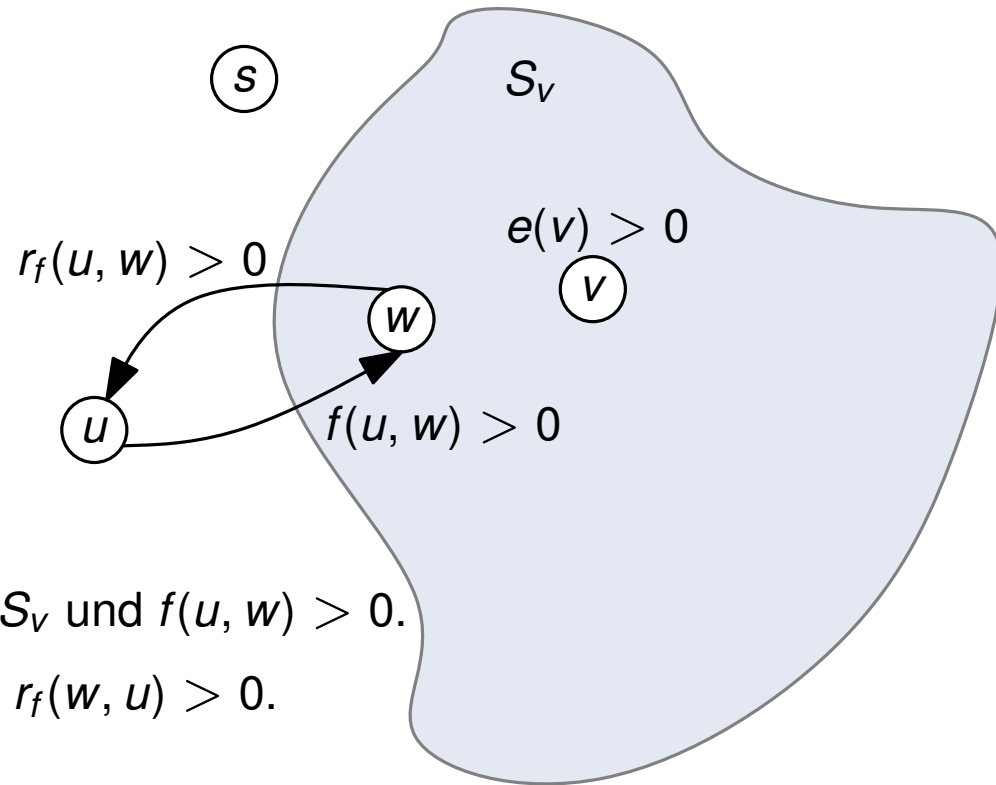
$$\sum_{w \in S_v} e(w) \geq 0$$

Da $v \in S_v$, gilt

$$\sum_{w \in S_v} e(w) > 0$$

Damit gibt es eine Kante (u, w) mit $u \notin S_v$, $w \in S_v$ und $f(u, w) > 0$.

Die Gegenkante (w, u) besitzt also Restkapazität $r_f(w, u) > 0$.



Widerspruch zu: u ist nicht in D_f von v aus erreichbar.

Lemma 4.25: Während des gesamten Algorithmus gilt

$$\forall v \in V \text{ } dist(v) \leq 2|V| - 1 .$$

Lemma 4.25: Während des gesamten Algorithmus gilt

$$\forall v \in V \text{ dist}(v) \leq 2|V| - 1 .$$

Beweis:

Initialisierung: Behauptung gilt offensichtlich.

Beliebiger Zeitpunkt wenn $\text{dist}(v)$ von v geändert wird:

dist heißt *zulässig* wenn

- $\text{dist}(s) = |V|$ und $\text{dist}(t) = 0$ und
- $\forall v \in V \setminus \{t\}$ und $\forall (v, w) \in E_f: \text{dist}(v) \leq \text{dist}(w) + 1$

Operation RELABEL(D, f, v, dist)

Vorbedingung: v ist aktiv und für alle w mit $r_f(v, w) > 0$ gilt $\text{dist}(v) \leq \text{dist}(w)$

Effekt: $\text{dist}(v)$ wird erhöht.

$$\text{dist}(v) = \begin{cases} \infty , & \text{falls } \{w \mid r_f(v, w) > 0\} = \emptyset , \\ \min\{\text{dist}(w) + 1 \mid r_f(v, w) > 0\} & \text{sonst.} \end{cases}$$

Lemma 4.25: Während des gesamten Algorithmus gilt

$$\forall v \in V \text{ dist}(v) \leq 2|V| - 1 .$$

Beweis:

Initialisierung: Behauptung gilt offensichtlich.

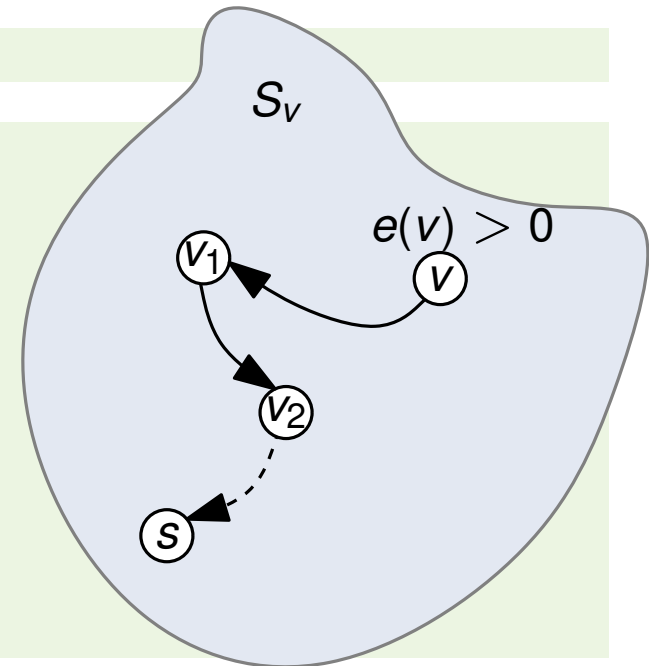
Beliebiger Zeitpunkt wenn $\text{dist}(v)$ von v geändert wird:

Knoten v ist aktiv, d.h. $e(v) > 0$.

Nach **Lemma 4.24:** s ist von v in D_f erreichbar:

$$v = v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_\ell = s$$

mit $\text{dist}(v_i) \leq \text{dist}(v_{i+1}) + 1$ für $1 \leq i \leq \ell - 1$



Operation RELABEL(D, f, v, dist)

Vorbedingung: v ist aktiv und für alle w mit $r_f(v, w) > 0$ gilt $\text{dist}(v) \leq \text{dist}(w)$

Effekt: $\text{dist}(v)$ wird erhöht.

$$\text{dist}(v) = \begin{cases} \infty, & \text{falls } \{w \mid r_f(v, w) > 0\} = \emptyset, \\ \min\{\text{dist}(w) + 1 \mid r_f(v, w) > 0\} & \text{sonst.} \end{cases}$$

dist heißt zulässig wenn

- $\text{dist}(s) = |V|$ und $\text{dist}(t) = 0$ und
- $\forall v \in V \setminus \{t\}$ und $\forall (v, w) \in E_f: \text{dist}(v) \leq \text{dist}(w) + 1$

Obere Schranke für Distanz

Lemma 4.25: Während des gesamten Algorithmus gilt

$$\forall v \in V \text{ dist}(v) \leq 2|V| - 1.$$

Beweis:

Initialisierung: Behauptung gilt offensichtlich.

Beliebiger Zeitpunkt wenn $\text{dist}(v)$ von v geändert wird:

Knoten v ist aktiv, d.h. $e(v) > 0$.

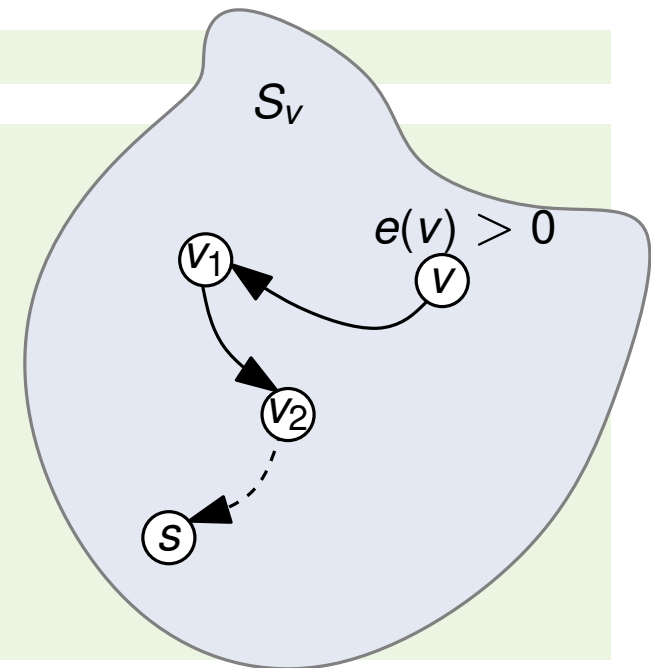
Nach **Lemma 4.24:** s ist von v in D_f erreichbar:

$$v = v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_\ell = s$$

mit $\text{dist}(v_i) \leq \text{dist}(v_{i+1}) + 1$ für $1 \leq i \leq \ell - 1$

Wegen $\ell \leq |V| - 1$ folgt:

$$\text{dist}(v) \leq \text{dist}(s) + \ell \leq 2|V| - 1$$



Operation RELABEL(D, f, v, dist)

Vorbedingung: v ist aktiv und für alle w mit $r_f(v, w) > 0$ gilt $\text{dist}(v) \leq \text{dist}(w)$

Effekt: $\text{dist}(v)$ wird erhöht.

$$\text{dist}(v) = \begin{cases} \infty, & \text{falls } \{w \mid r_f(v, w) > 0\} = \emptyset, \\ \min\{\text{dist}(w) + 1 \mid r_f(v, w) > 0\} & \text{sonst.} \end{cases}$$

dist heißt zulässig wenn

- $\text{dist}(s) = |V|$ und $\text{dist}(t) = 0$ und
- $\forall v \in V \setminus \{t\}$ und $\forall (v, w) \in E_f: \text{dist}(v) \leq \text{dist}(w) + 1$

Obere Schranke für Distanz

Lemma 4.25: Während des gesamten Algorithmus gilt

$$\forall v \in V \text{ dist}(v) \leq 2|V| - 1.$$

Beweis:

Initialisierung: Behauptung gilt offensichtlich.

Beliebiger Zeitpunkt wenn $\text{dist}(v)$ von v geändert wird:

Knoten v ist aktiv, d.h. $e(v) > 0$.

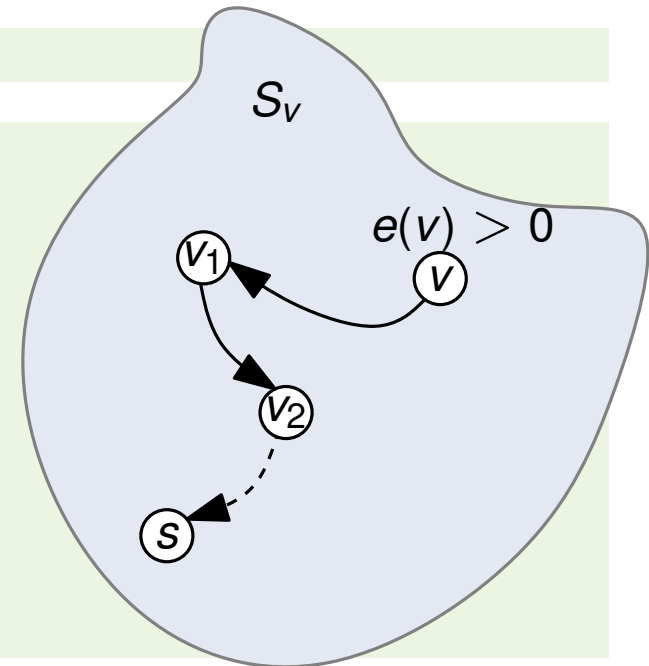
Nach **Lemma 4.24**: s ist von v in D_f erreichbar:

$$v = v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_\ell = s$$

mit $\text{dist}(v_i) \leq \text{dist}(v_{i+1}) + 1$ für $1 \leq i \leq \ell - 1$

Wegen $\ell \leq |V| - 1$ folgt:


$$\text{dist}(v) \leq \text{dist}(s) + \ell \leq 2|V| - 1$$





Es folgt:

Lemma 4.26: Im Laufe der Ausführung werden pro Knoten höchstens $2|V| - 1$ RELABEL-Operationen ausgeführt. Die Gesamtzahl der RELABEL-Operationen ist also höchstens $2|V|^2$.

1. Schritt: Wenn Algorithmus terminiert und die Markierungen endlich bleiben, dann ist das Ergebnis ein Maximalfluss.

- a) Solange aktiver Knoten vorhanden, kann Operation PUSH oder Operation RELABEL angewendet werden.
 - b) Es gilt stets: f ist Präfluss und $dist$ ist bezüglich f zulässige Markierung.
 - c) t ist im Residualgraph D_f des Präflusses f von s aus nicht erreichbar.
- 

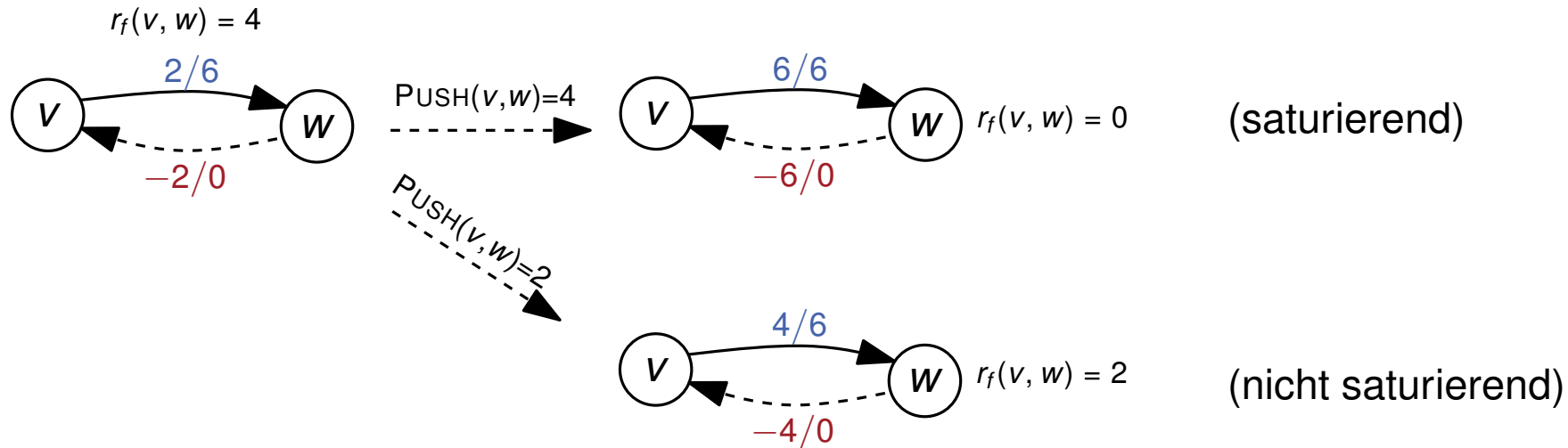
2. Schritt: Algorithmus terminiert und Markierungen bleiben endlich:

- a) Finde obere Schranke für $dist$. 
- b) Finde obere Schranke für Anzahl Aufrufe von RELABEL. 
- c) Finde obere Schranke für Anzahl Aufrufe von PUSH.

Bezeichne f die Abbildung, die schrittweise konstruiert wird.

Abschätzung PUSH-Operationen

Definition 4.27: Eine Operation $\text{PUSH}(v, w)$ heißt *saturierend*, wenn danach $r_f(v, w) = 0$ gilt. Ansonsten heißt $\text{PUSH}(v, w)$ *nicht saturierend*.

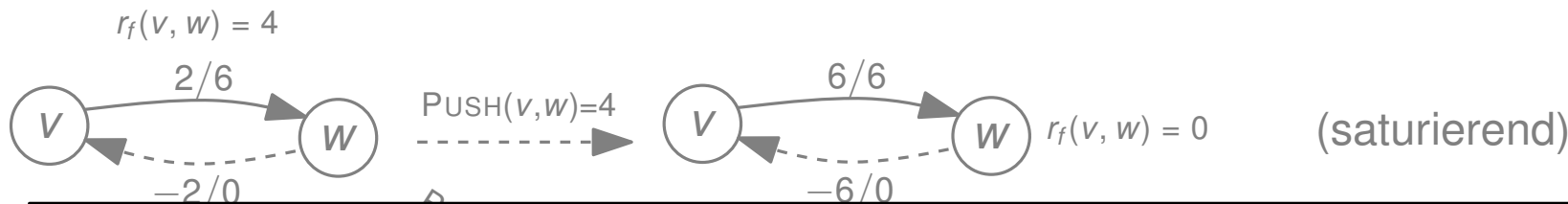


Zeige:

Lemma 4.28: Während des Algorithmus werden höchstens $2|V||E|$ saturierende PUSH ausgeführt.

Lemma 4.29: Während des Algorithmus werden höchstens $4|V|^2|E|$ nicht saturierende PUSH ausgeführt.

Definition 4.27: Eine Operation $\text{PUSH}(v, w)$ heißt *saturierend*, wenn danach $r_f(v, w) = 0$ gilt. Ansonsten heißt $\text{PUSH}(v, w)$ *nicht saturierend*.



Damit ergibt sich:

Satz 4.30: Der Algorithmus von Goldberg und Tarjan terminiert nach $\mathcal{O}(|V|^2|E|)$ Ausführungen zulässiger PUSH- oder RELABEL-Operationen.

Lemma 4.28: Während des Algorithmus werden höchstens $2|V||E|$ saturierende PUSH ausgeführt.

Lemma 4.29: Während des Algorithmus werden höchstens $4|V|^2|E|$ nicht saturierende PUSH ausgeführt.

Lemma 4.28: Während des Algorithmus werden höchstens $2|V||E|$ saturierende PUSH ausgeführt.

Lemma 4.28: Während des Algorithmus werden höchstens $2|V||E|$ saturierende PUSH ausgeführt.

Beweis: Betrachte saturierendes $\text{PUSH}(v, w)$:

Es gilt: $r_f(v, w) > 0$ und $\text{dist}(v) = \text{dist}(w) + 1$

Operation $\text{PUSH}(D, f, v, w)$

Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $\text{dist}(v) = \text{dist}(w) + 1$.

Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta, f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta, r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta, e(w) \leftarrow e(w) + \Delta$

Lemma 4.28: Während des Algorithmus werden höchstens $2|V||E|$ saturierende PUSH ausgeführt.

Beweis: Betrachte saturierendes $\text{PUSH}(v, w)$:

Es gilt: $r_f(v, w) > 0$ und $\text{dist}(v) = \text{dist}(w) + 1$

—► Erneutes $\text{PUSH}(v, w)$ nur möglich, wenn in der Zwischenzeit $\text{Push}(w, v)$.

Für $\text{PUSH}(w, v)$ musste aber $\text{dist}(w) = \text{dist}(v) + 1$ gelten.

—► Nach Ausführung vom zweiten $\text{PUSH}(v, w)$:

$\text{dist}(v)$ und $\text{dist}(w)$ wurden um mindestens zwei erhöht.

$\text{PUSH}_1(v, w)$
...
 $\text{PUSH}_1(w, v)$
...
 $\text{PUSH}_2(v, w)$

Operation $\text{PUSH}(D, f, v, w)$

Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $\text{dist}(v) = \text{dist}(w) + 1$.

Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta, f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta, r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta, e(w) \leftarrow e(w) + \Delta$

Saturierende PUSH-Operationen

Lemma 4.28: Während des Algorithmus werden höchstens $2|V||E|$ saturierende PUSH ausgeführt.

Beweis: Betrachte saturierendes $\text{PUSH}(v, w)$:

Es gilt: $r_f(v, w) > 0$ und $\text{dist}(v) = \text{dist}(w) + 1$

—→ Erneutes $\text{PUSH}(v, w)$ nur möglich, wenn in der Zwischenzeit $\text{Push}(w, v)$.

Für $\text{PUSH}(w, v)$ musste aber $\text{dist}(w) = \text{dist}(v) + 1$ gelten.

—→ Nach Ausführung vom zweiten $\text{PUSH}(v, w)$:

$\text{dist}(v)$ und $\text{dist}(w)$ wurden um mindestens zwei erhöht.

Nach Lemma 4.25:

$\text{dist}(v) \leq 2|V| - 1$ und $\text{dist}(w) \leq 2|V| - 1$.

Für Kante (v, w) maximal $2|V| - 1$ viele saturierende PUSH-Operationen.

Insgesamt:

Maximal $2|V||E|$ viele saturierende PUSH-Operationen.

$\text{PUSH}_1(v, w)$
...
 $\text{PUSH}_1(w, v)$
...
 $\text{PUSH}_2(v, w)$

Operation $\text{PUSH}(D, f, v, w)$

Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $\text{dist}(v) = \text{dist}(w) + 1$.

Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta, f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta, r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta, e(w) \leftarrow e(w) + \Delta$

Nicht saturierende PUSH-Operationen

Lemma 4.29: Während des Algorithmus werden höchstens $4|V|^2|E|$ nicht saturierende PUSH ausgeführt.

Nicht saturierende PUSH-Operationen

Lemma 4.29: Während des Algorithmus werden höchstens $4|V|^2|E|$ nicht saturierende PUSH ausgeführt.

Beweis: Betrachte Veränderung von

$$\mathcal{D} = \sum_{\substack{v \in V \setminus \{s, t\}, \\ v \text{ aktiv}}} \text{dist}(v)$$

Initialisierung: $\mathcal{D} = 0$ (Es gilt immer $\mathcal{D} \geq 0$)

Auswirkungen der Operationen auf \mathcal{D}

Operation PUSH(D, f, v, w)

Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $\text{dist}(v) = \text{dist}(w) + 1$.

Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta, f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta, r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta, e(w) \leftarrow e(w) + \Delta$

Nicht saturierende PUSH-Operationen

Lemma 4.29: Während des Algorithmus werden höchstens $4|V|^2|E|$ nicht saturierende PUSH ausgeführt.

Beweis: Betrachte Veränderung von

$$\mathcal{D} = \sum_{\substack{v \in V \setminus \{s, t\}, \\ v \text{ aktiv}}} dist(v)$$

Initialisierung: $\mathcal{D} = 0$ (Es gilt immer $\mathcal{D} \geq 0$)

Auswirkungen der Operationen auf \mathcal{D}

Nicht saturierendes PUSH setzt \mathcal{D} um mind. 1 herab, denn:

v ist danach nicht aktiv und $dist(w) = dist(v) - 1$

Operation PUSH(D, f, v, w)

Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $dist(v) = dist(w) + 1$.

Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta, f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta, r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta, e(w) \leftarrow e(w) + \Delta$

Nicht saturierende PUSH-Operationen

Lemma 4.29: Während des Algorithmus werden höchstens $4|V|^2|E|$ nicht saturierende PUSH ausgeführt.

Beweis: Betrachte Veränderung von

$$\mathcal{D} = \sum_{\substack{v \in V \setminus \{s, t\}, \\ v \text{ aktiv}}} \text{dist}(v)$$

Initialisierung: $\mathcal{D} = 0$ (Es gilt immer $\mathcal{D} \geq 0$)

Auswirkungen der Operationen auf \mathcal{D}

Nicht saturierendes PUSH setzt \mathcal{D} um mind. 1 herab, denn:

v ist danach nicht aktiv und $\text{dist}(w) = \text{dist}(v) - 1$

Saturierendes PUSH erhöht \mathcal{D} um max. $2|V| - 1$ (Lemma 4.25).

→ saturierende PUSH-Operationen erhöhen \mathcal{D} um max. $\underbrace{(2|V| - 1) \cdot 2|V||E|}_{=A}$ (Lemma 4.28)

RELABEL-Operationen erhöhen \mathcal{D} um max. $\underbrace{(2|V| - 1) \cdot |V|}_{=B}$ (Lemma 4.26).

Operation PUSH(D, f, v, w)

Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $\text{dist}(v) = \text{dist}(w) + 1$.

Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta, f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta, r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta, e(w) \leftarrow e(w) + \Delta$

Nicht saturierende PUSH-Operationen

Lemma 4.29: Während des Algorithmus werden höchstens $4|V|^2|E|$ nicht saturierende PUSH ausgeführt.

Beweis: Betrachte Veränderung von

$$\mathcal{D} = \sum_{\substack{v \in V \setminus \{s, t\}, \\ v \text{ aktiv}}} \text{dist}(v)$$

Initialisierung: $\mathcal{D} = 0$ (Es gilt immer $\mathcal{D} \geq 0$)

Auswirkungen der Operationen auf \mathcal{D}

Nicht saturierendes PUSH setzt \mathcal{D} um mind. 1 herab, denn:

v ist danach nicht aktiv und $\text{dist}(w) = \text{dist}(v) - 1$

Saturierendes PUSH erhöht \mathcal{D} um max. $2|V| - 1$ (Lemma 4.25).

→ saturierende PUSH-Operationen erhöhen \mathcal{D} um max. $\underbrace{(2|V| - 1) \cdot 2|V||E|}_{=A}$ (Lemma 4.28)

RELABEL-Operationen erhöhen \mathcal{D} um max. $\underbrace{(2|V| - 1) \cdot |V|}_{=B}$ (Lemma 4.26).

Anzahl nicht saturierender PUSH:

$$\text{maximal } B + A \leq 4|V|^2|E|$$

Operation PUSH(D, f, v, w)

Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $\text{dist}(v) = \text{dist}(w) + 1$.

Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta, f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta, r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta, e(w) \leftarrow e(w) + \Delta$

Satz 4.30: Der Algorithmus von Goldberg und Tarjan terminiert nach $\mathcal{O}(|V|^2|E|)$ Ausführungen zulässiger PUSH- oder RELABEL-Operationen.

Laufzeit hängt stark von Wahl der aktiven Knoten ab:

- **FIFO-Implementierung:** Wähle aktive Knoten entsprechend *first-in-first-out*-Prinzip: $\mathcal{O}(|V|^3)$ Laufzeit.

Mit dynamischen Bäumen: $\mathcal{O}(|V||E| \log \frac{|V|^2}{|E|})$

- **Highest-Label:** Für PUSH wähle aktiven Knoten mit höchstem *dist*-Wert: $\mathcal{O}(|V|^2 |E|^{\frac{1}{2}})$

- **Excess-Scaling:** Für PUSH(v, w) wird die Kante (v, w) so gewählt, dass v aktiv, $e(v)$ *geeignet groß* und $e(w)$ *geeignet klein* ist: $\mathcal{O}(|E| + |V|^2 \log C)$,
mit $C = \max_{(u,v) \in E} c(u, v)$