Algorithmen II
Vorlesung am 31.10.2013

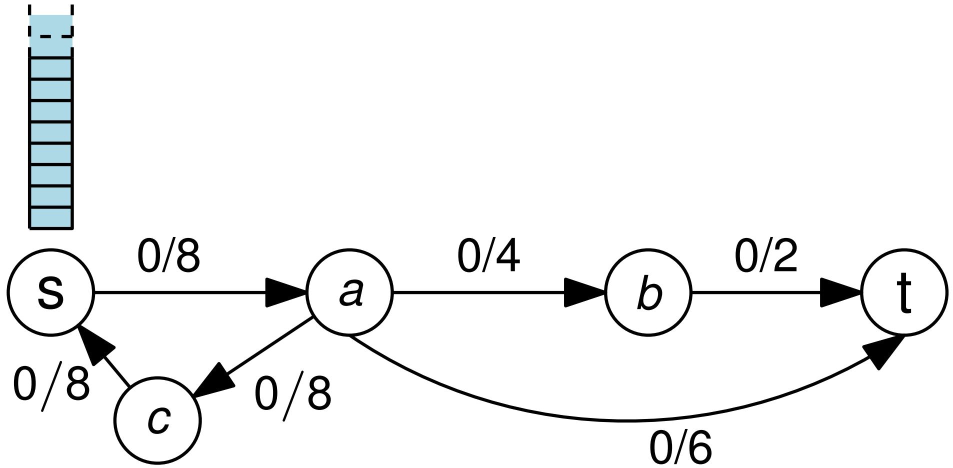
INSTITUT FÜR THEORETISCHE INFORMATIK · PROF. DR. DOROTHEA WAGNER

Flussalgorithmus von Goldberg und Tarjan (1988)

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Grundprinizip

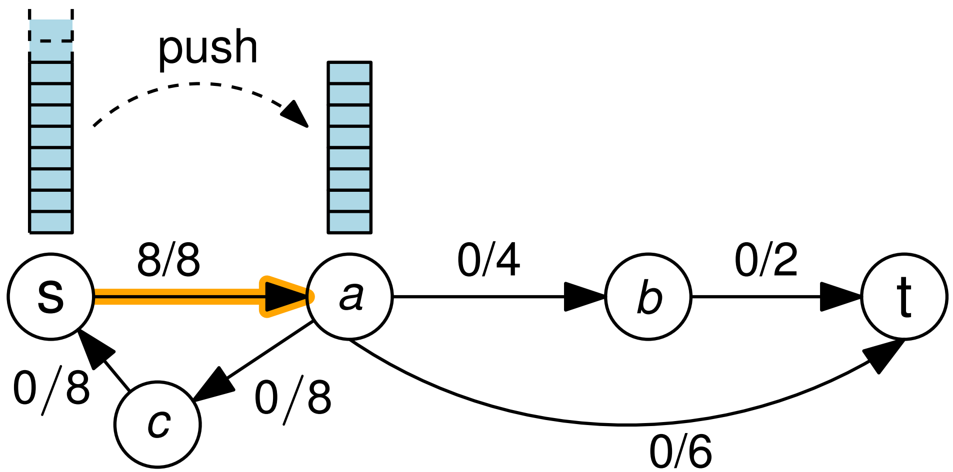
- Basiert nicht auf erhöhenden Wegen sondern auf zwei Operationen: PUSH und RELABEL.
- Fluss in Zwischenschritten nicht unbedingt gültig (Flusserhaltung nicht garantiert).



- Im Beispiel nur PUSH-Operation veranschaulicht: Drückt den Fluss zum nächsten Knoten und erzeugt dort gegebenfalls 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.

Grundprinizip

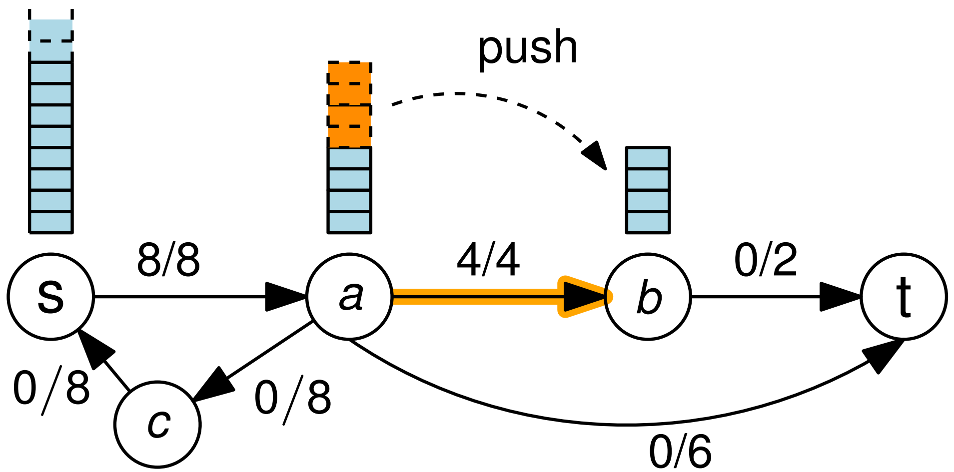
- Basiert nicht auf erhöhenden Wegen sondern auf zwei Operationen: PUSH und RELABEL.
- Fluss in Zwischenschritten nicht unbedingt gültig (Flusserhaltung nicht garantiert).



- Im Beispiel nur PUSH-Operation veranschaulicht: Drückt den Fluss zum nächsten Knoten und erzeugt dort gegebenfalls 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.

Grundprinizip

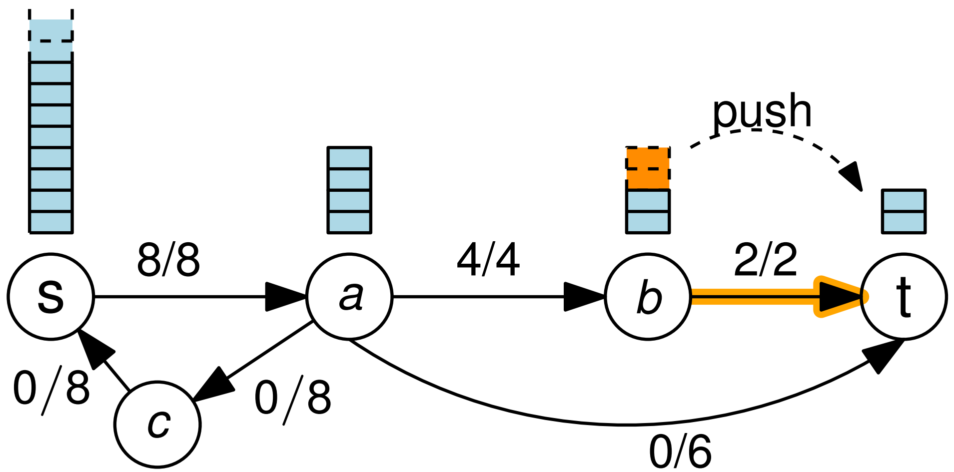
- Basiert nicht auf erhöhenden Wegen sondern auf zwei Operationen: PUSH und RELABEL.
- Fluss in Zwischenschritten nicht unbedingt gültig (Flusserhaltung nicht garantiert).



- Im Beispiel nur PUSH-Operation veranschaulicht: Drückt den Fluss zum nächsten Knoten und erzeugt dort gegebenfalls 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.

Grundprinizip

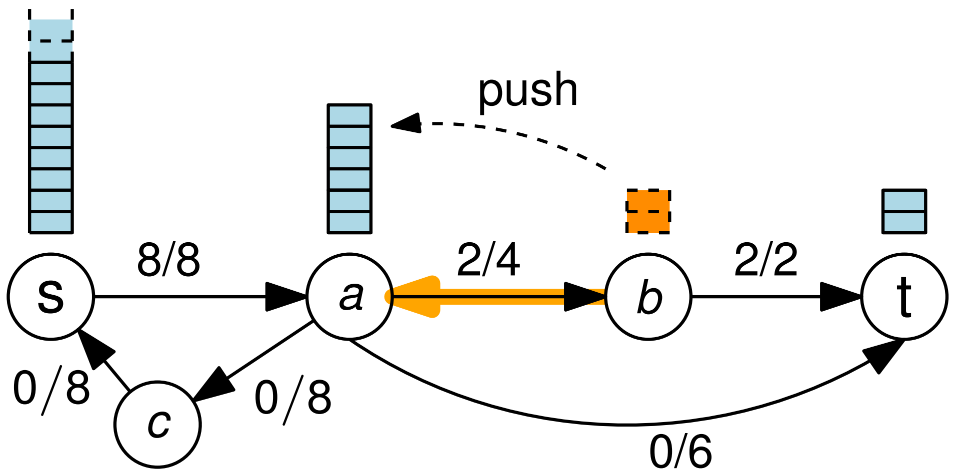
- Basiert nicht auf erhöhenden Wegen sondern auf zwei Operationen: PUSH und RELABEL.
- Fluss in Zwischenschritten nicht unbedingt gültig (Flusserhaltung nicht garantiert).



- Im Beispiel nur PUSH-Operation veranschaulicht: Drückt den Fluss zum nächsten Knoten und erzeugt dort gegebenfalls 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.

Grundprinizip

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



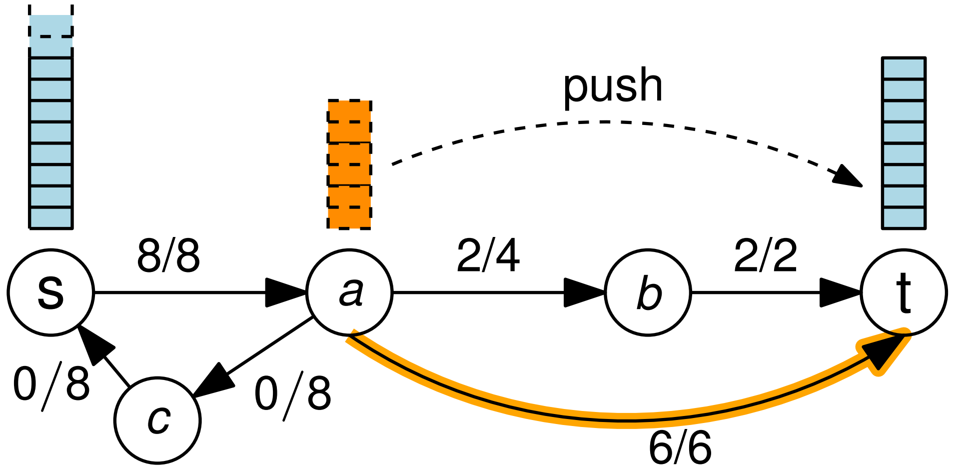
- Im Beispiel nur PUSH-Operation veranschaulicht: Drückt den Fluss zum nächsten Knoten und erzeugt dort gegebenfalls Überschuss, der zurück geführt werden muss.
- RELABEL-Operation garantiert, dass Fluss in die richtige Richtung gedrückt wird.
  - im Beispiel darf der Fluss nicht von *a* über *c* nach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

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Grundprinizip

Karlsruher Institut für Technologie

- 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 gegebenfalls 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.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Anpassung

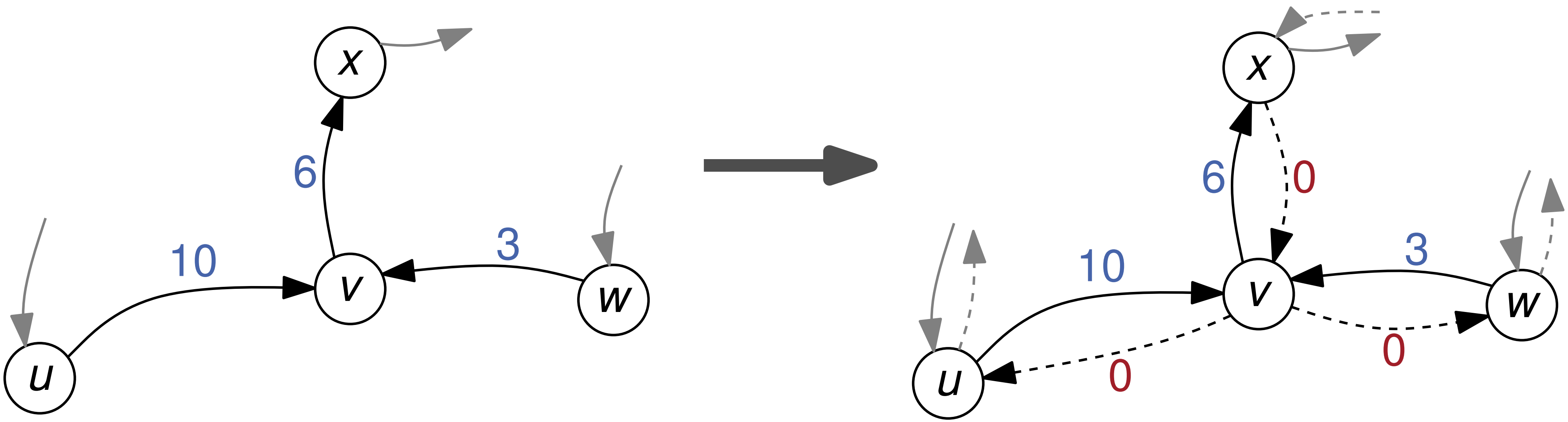
Gegeben: Netzwerk (D, s, t, c)

Erweitere c von $c\colon E \to \mathbb{R}_0^+$ auf $c\colon V \times V \to \mathbb{R}_0^+$, 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.

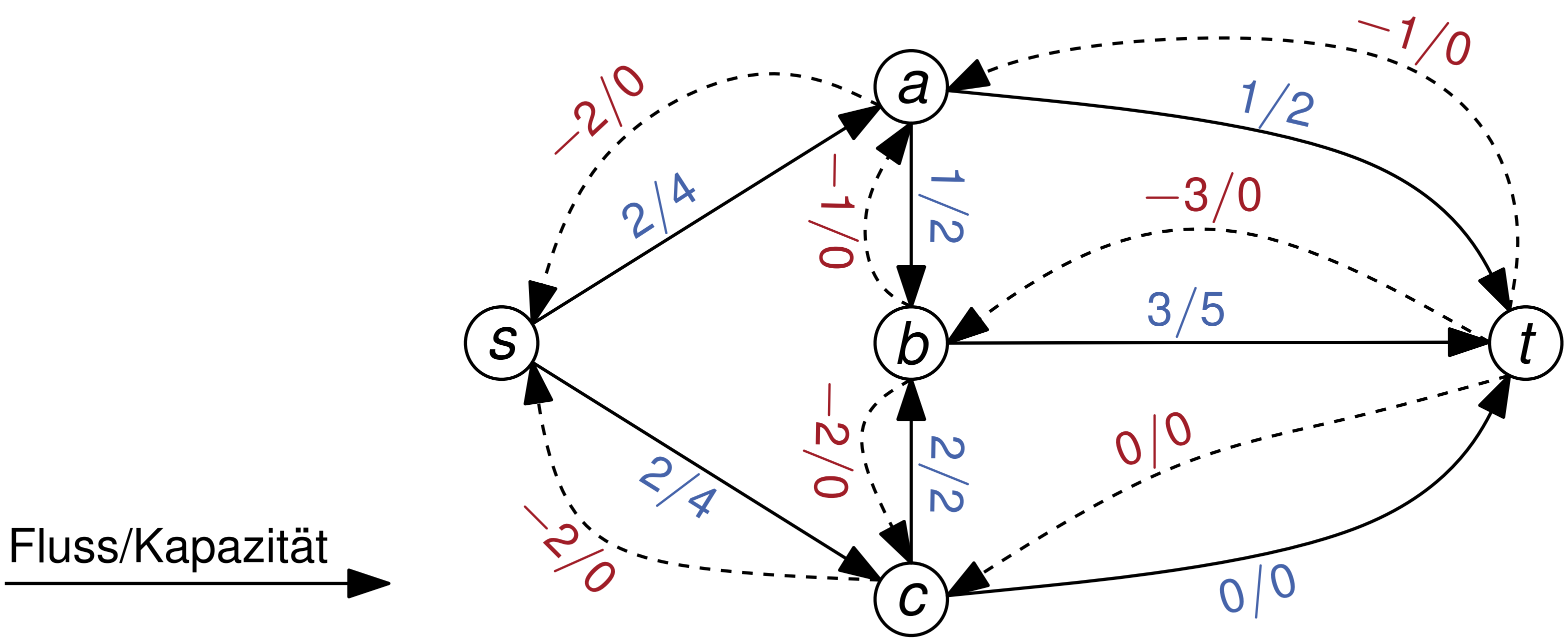
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Erweiterte Flussdefinition

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

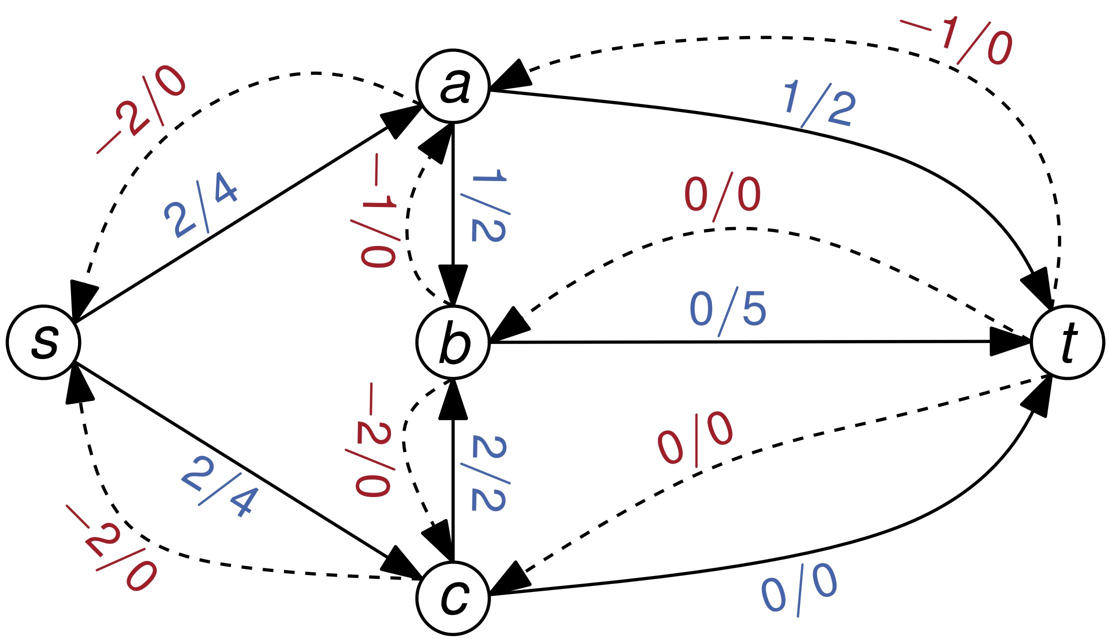


Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Präfluss

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

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



In Knoten b fließt mehr hinein als hinaus.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Flussüberschuss und Restkapazität

Definition: Sei f ein Präfluss.

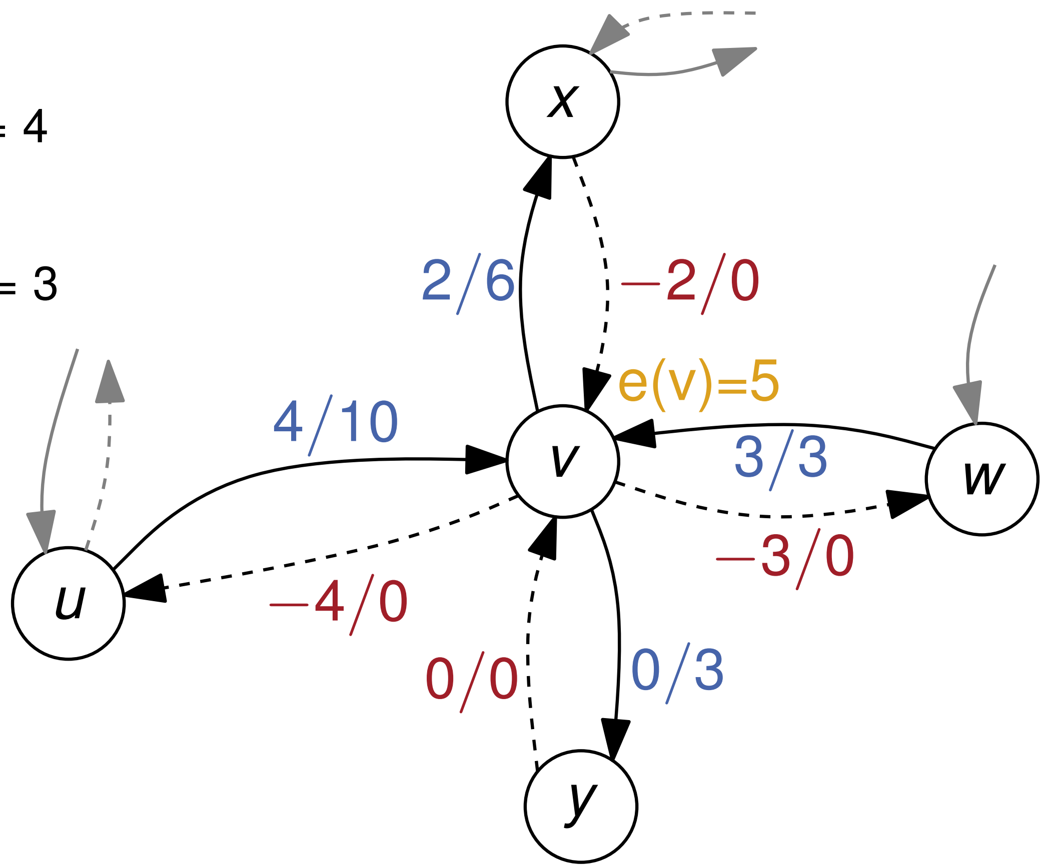
Flussüberschuss: $e(v) := \sum_{u \in V} f(u,v)$ mit $v \in V \backslash \{t\}$

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

Beispiel:

$r_f(u,v) = 10 - 4 = 6$
$r_f(v,u) = 0 - (-4) = 4$

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

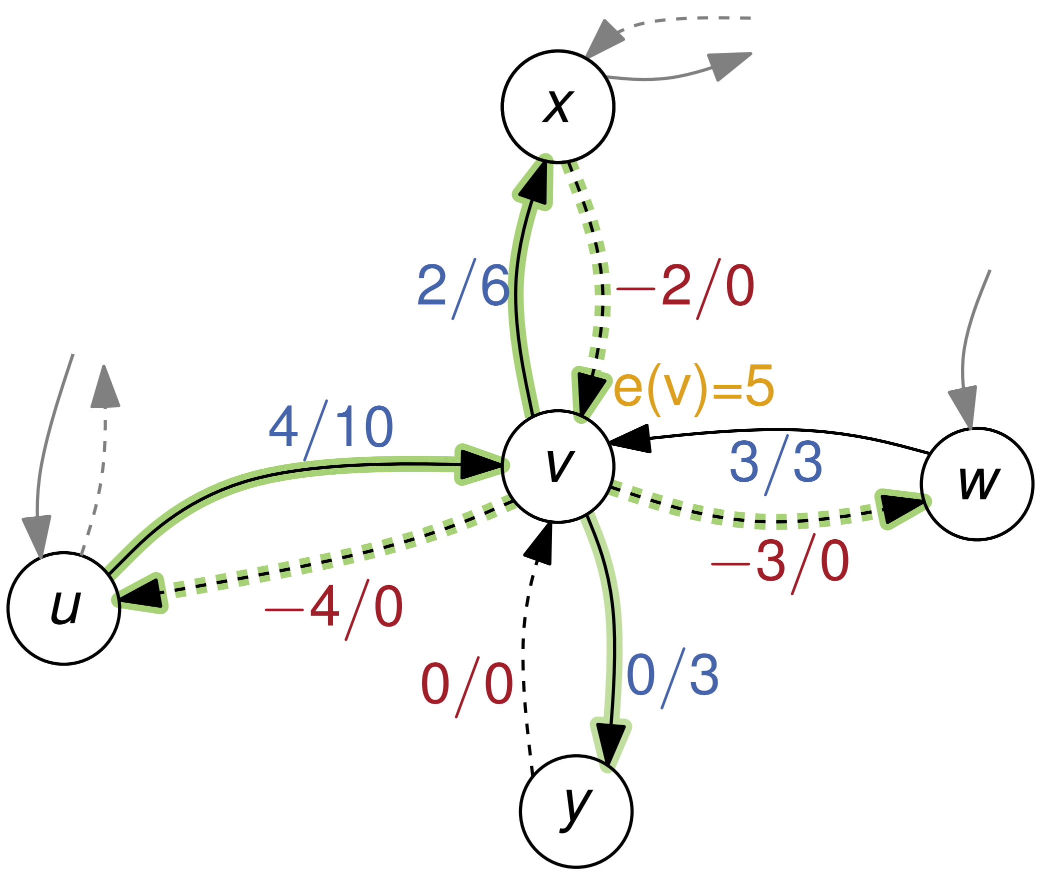


Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

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 $E_f := \{(u, v) \in E' \mid r_f(u, v) > 0\}$



Flussüberschuss und Restkapazität

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

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

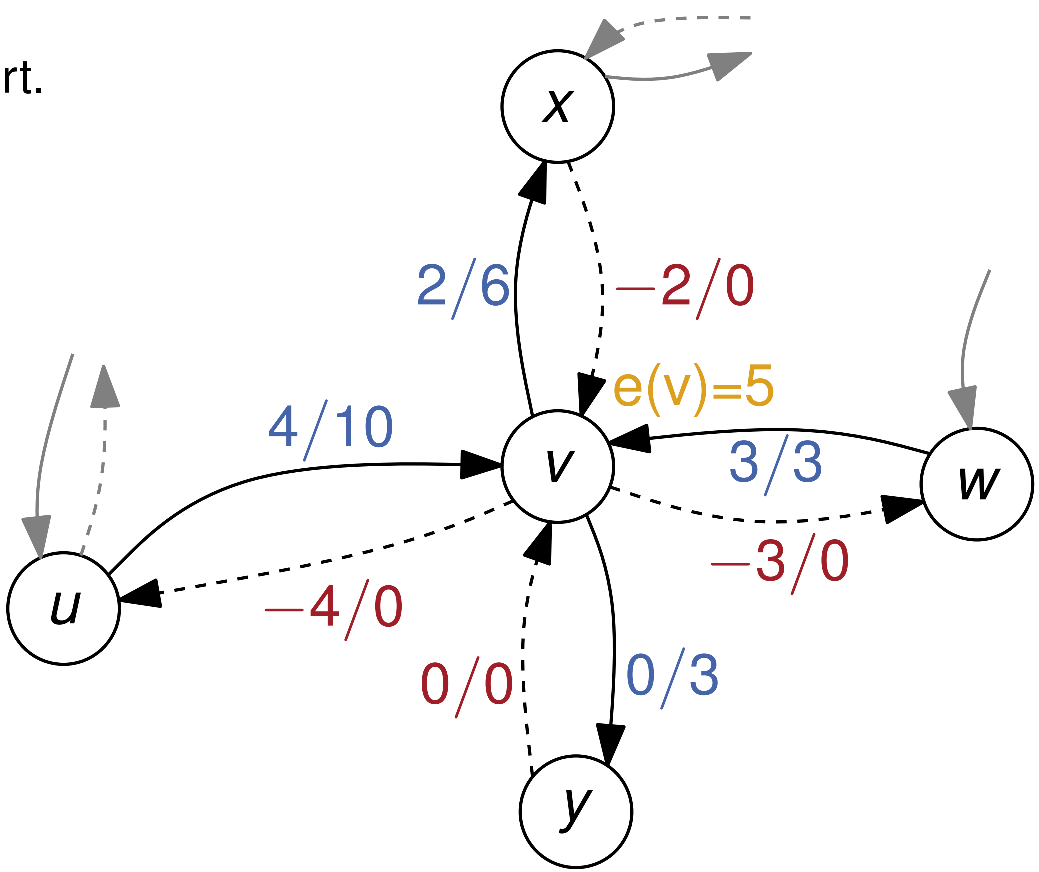
Beispiel:

(u, v) ist nicht saturiert.

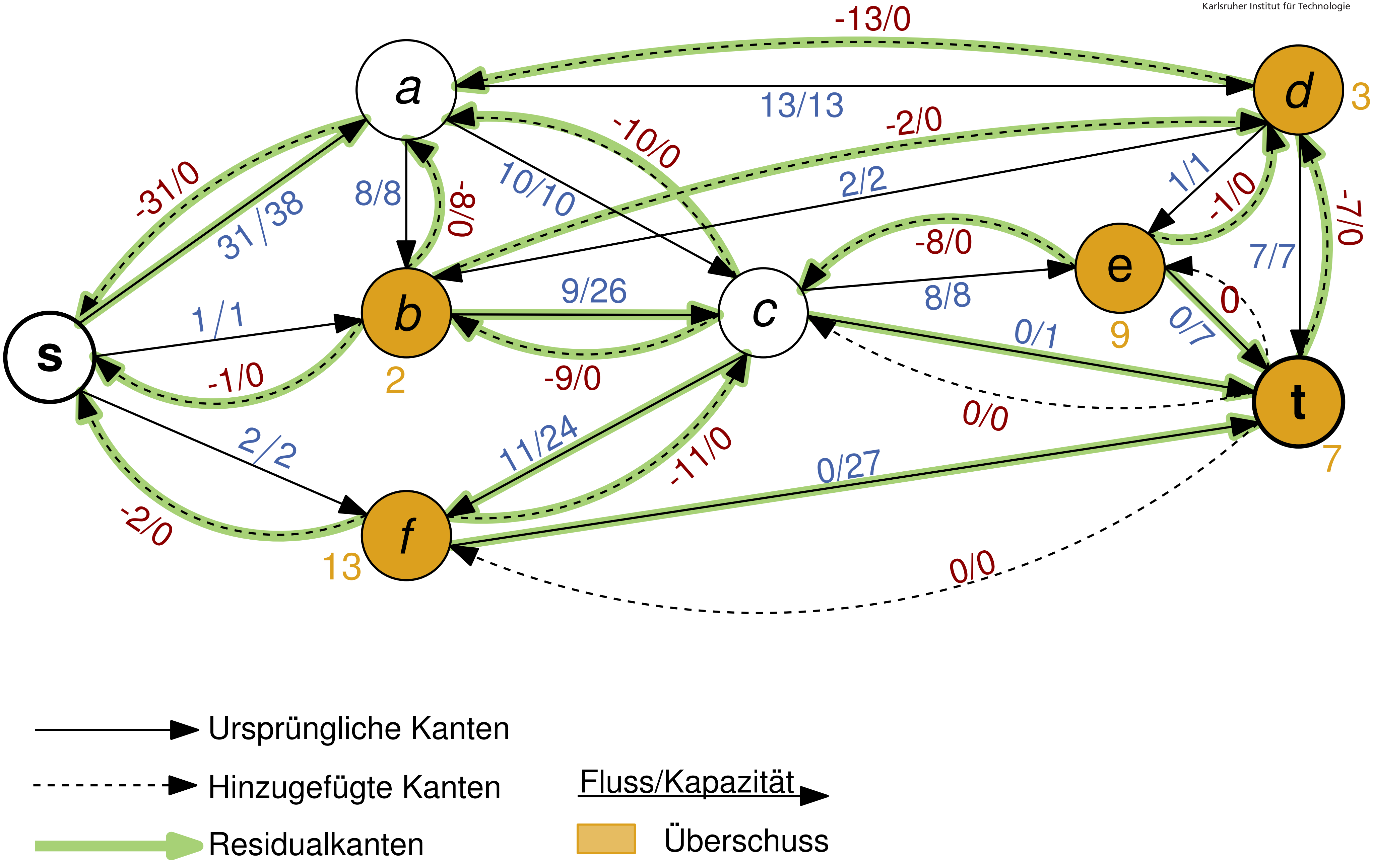
(w, v) ist saturiert.

(u, v) ist nicht leer.

(v, y) ist leer.



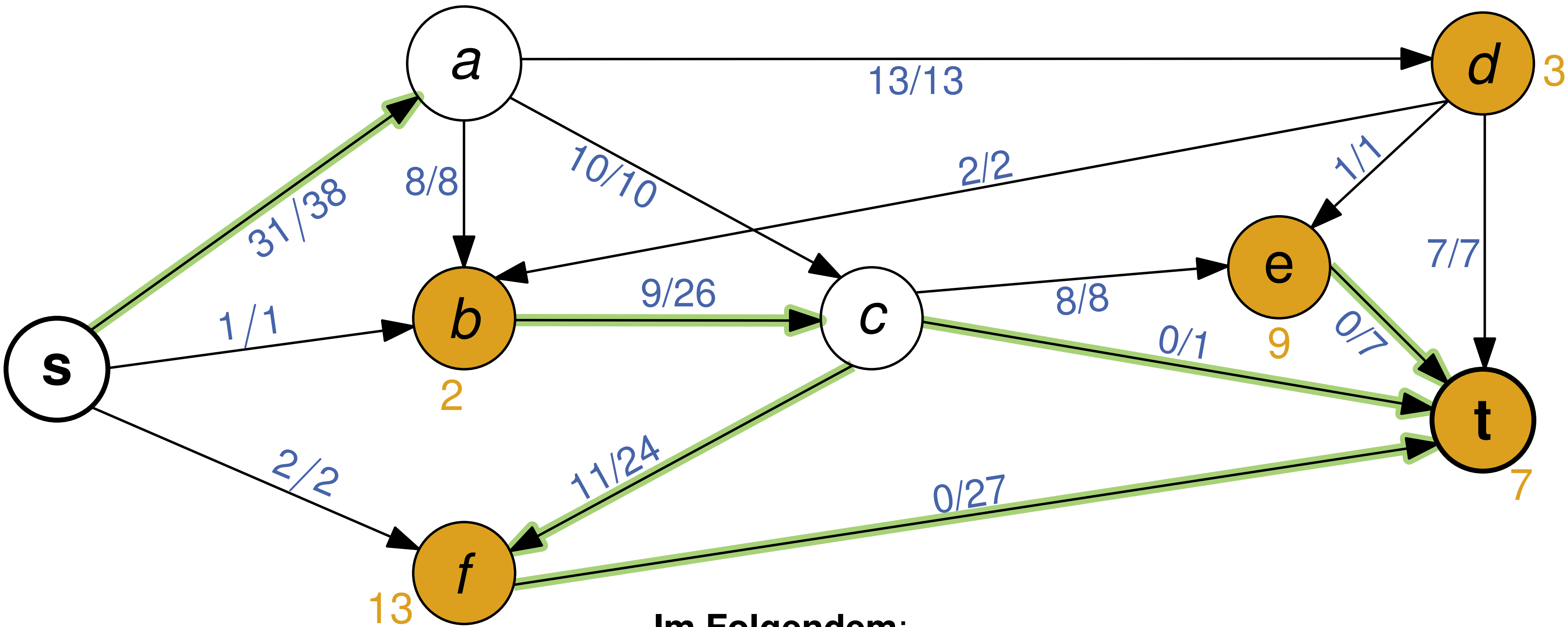
Beispiel



-13/0
13/13
-2/0
2/2
-10/0
10/10
8/8
-8/0
-31/0
31/38
1/1
-1/0
2/2
-2/0
9/26
-9/0
11/24
-11/0
0/27
0/0
-8/0
8/8
0/1
0/0
1/1
-1/0
7/7
-7/0
0/7
0
s
a
b
c
d
e
f
t
3
2
9
7
13

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

Zulässige Markierung

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

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

Ein Knoten $v \in V \backslash \{t\}$ 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 $v \in V \backslash \{s\}$ 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.

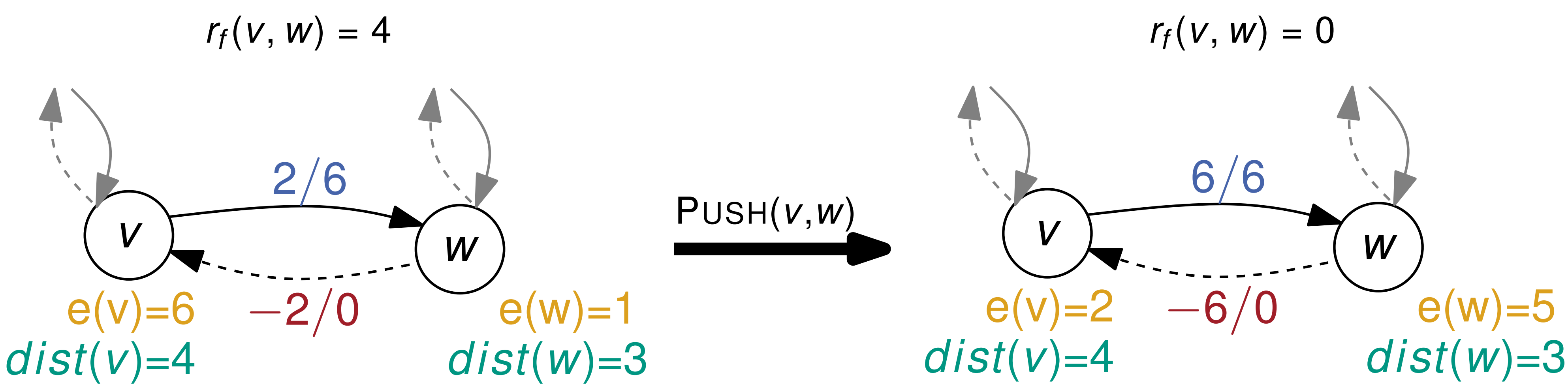
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

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:



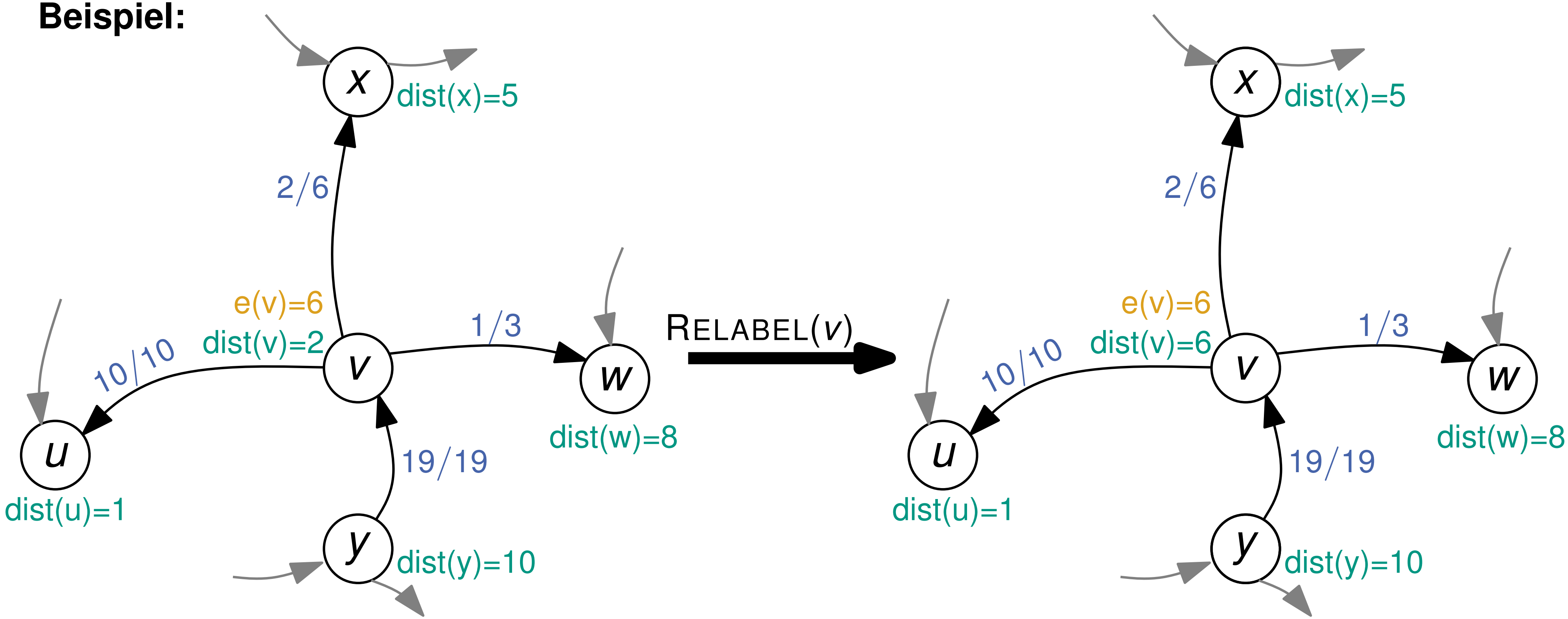
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

RELABEL-Operation

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:



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

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 $v \in V \backslash \{s\}$ setze:

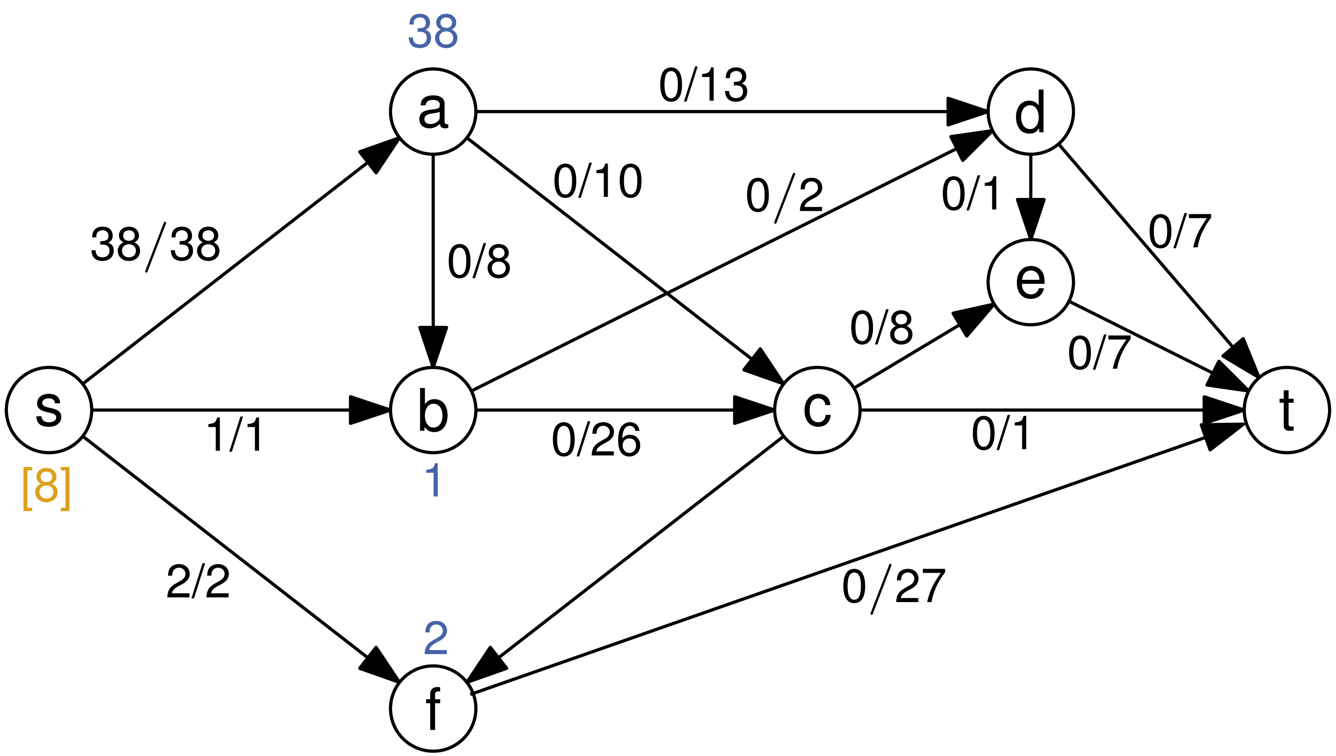
- $f(s, v) \leftarrow c(s, v)$, $r_f(s, v) \leftarrow 0$
- $f(v, s) \leftarrow -c(s, v)$, $r_f(v, s) \leftarrow c(v, s) - f(v, s)$
- $dist(v) \leftarrow 0$
- $e(v) \leftarrow c(s, v)$

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

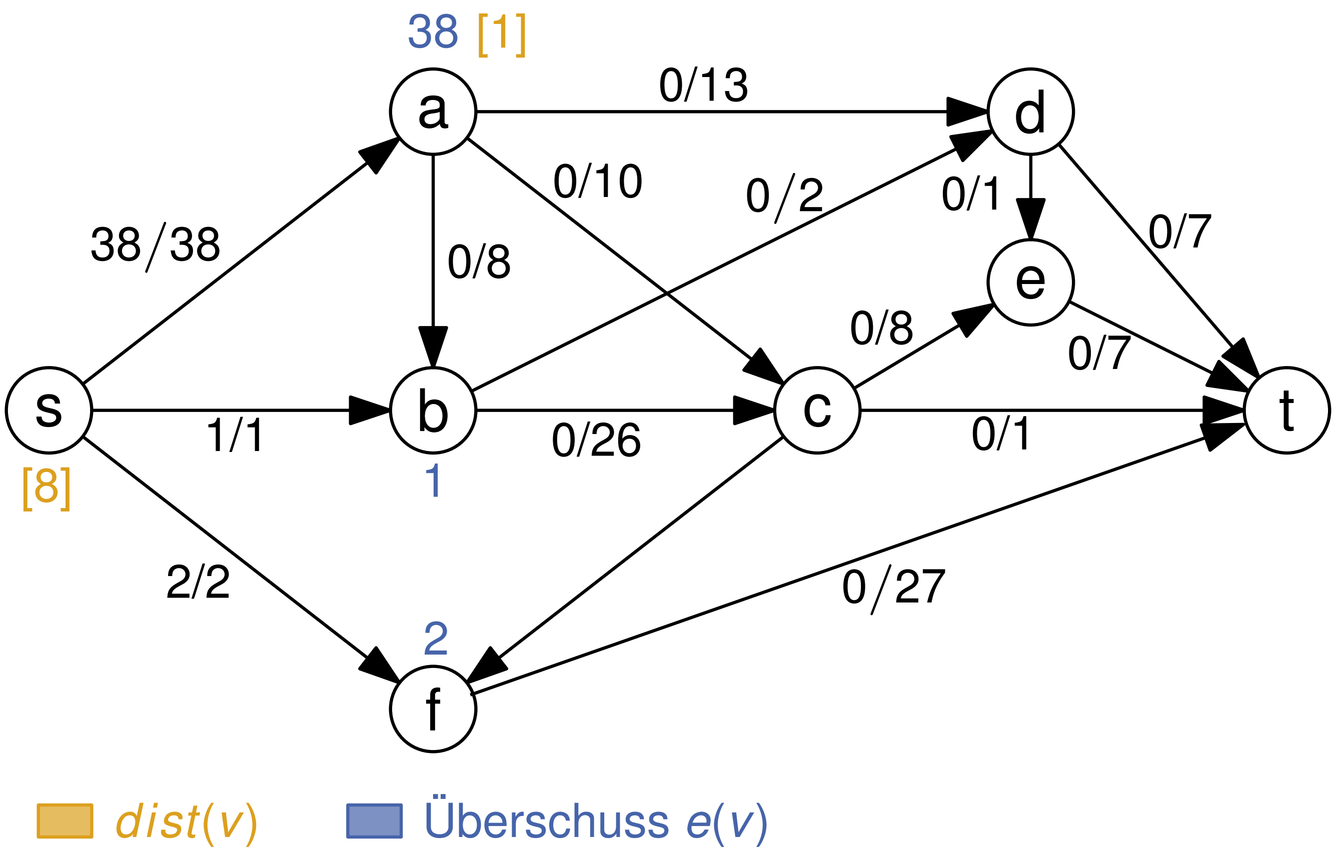
- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta$, $f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta$, $r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta$, $e(w) \leftarrow e(w) + \Delta$

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.

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

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Beispiel



Ausgeführte Operationen:

Initialisierung

RELABEL(a)

Anstehende Operationen:

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

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

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

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.

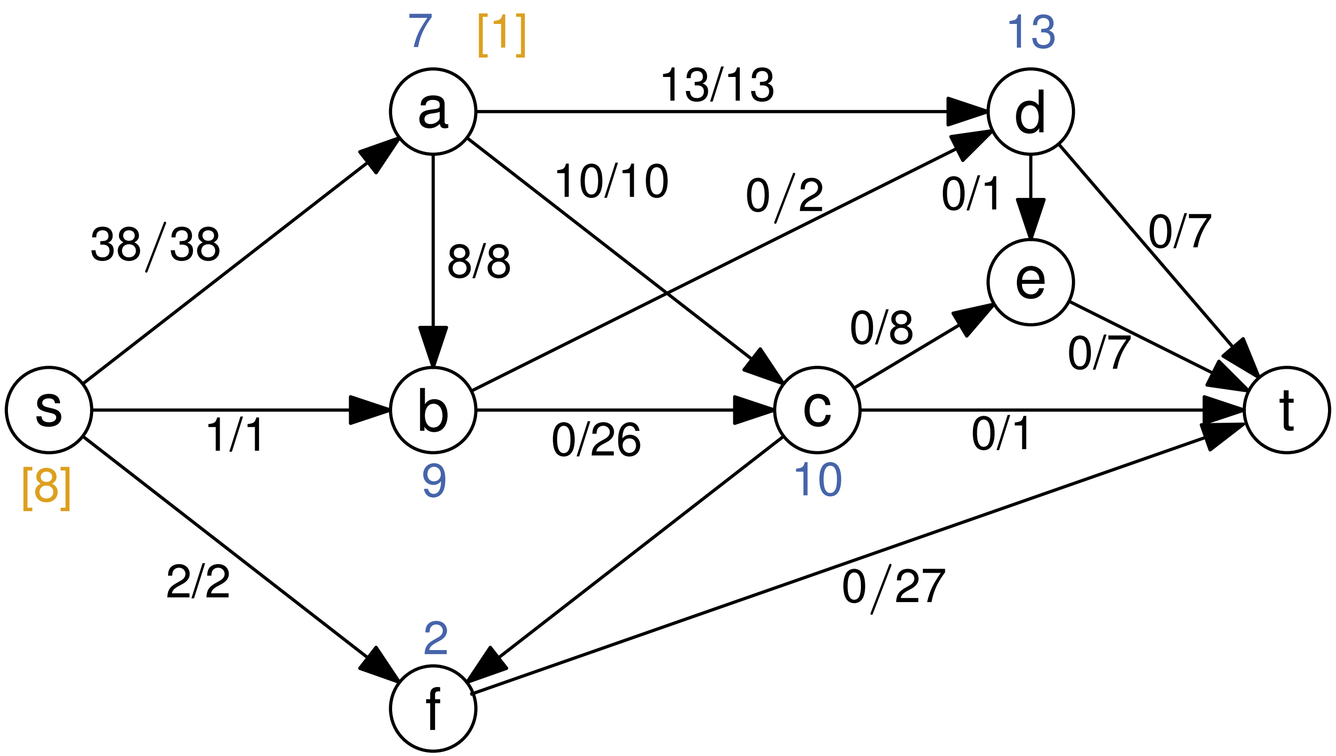
- $\Delta \leftarrow \min\{e(v), r_f(v,w)\}$
- $f(v,w) \leftarrow f(v,w) + \Delta$, $f(w,v) \leftarrow f(w,v) - \Delta$
- $r_f(v,w) \leftarrow r_f(v,w) - \Delta$, $r_f(w,v) \leftarrow r_f(w,v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta$, $e(w) \leftarrow e(w) + \Delta$

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.

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

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

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.

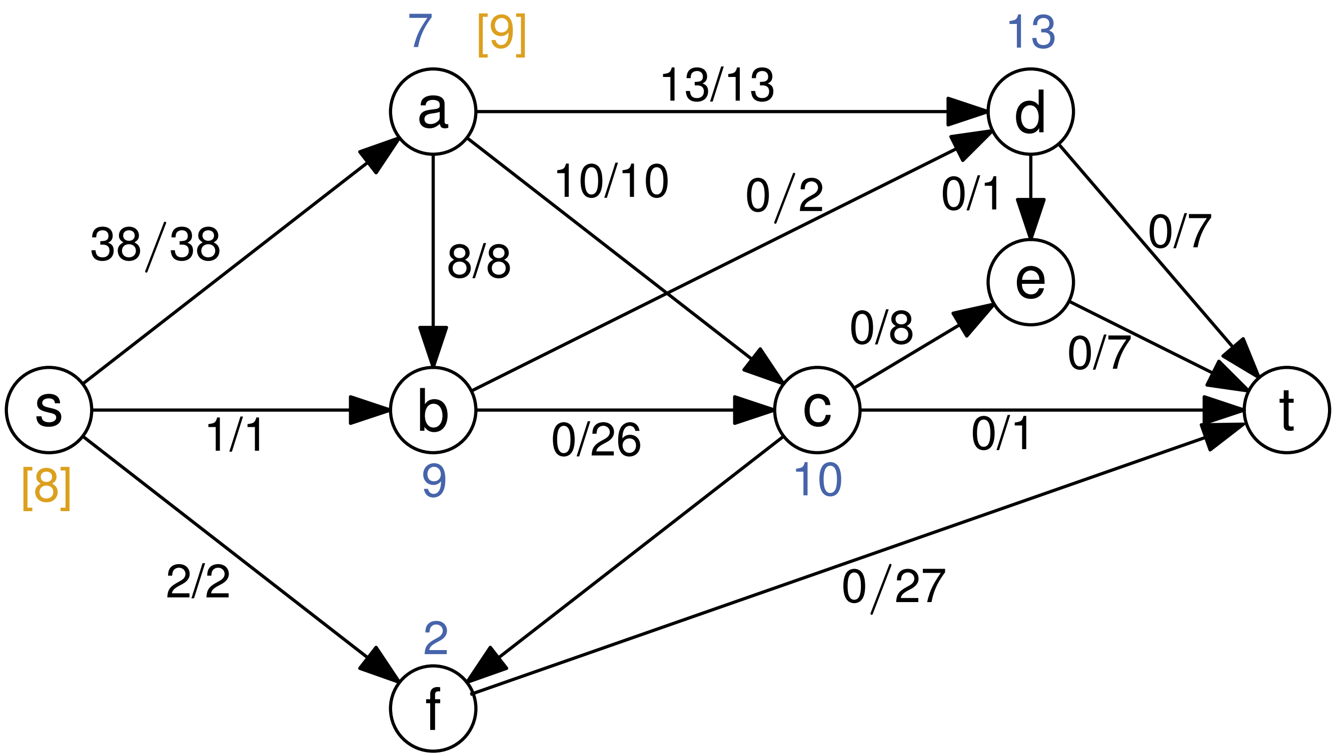
- $\Delta \leftarrow \min\{e(v), r_f(v, w)\}$
- $f(v, w) \leftarrow f(v, w) + \Delta$, $f(w, v) \leftarrow f(w, v) - \Delta$
- $r_f(v, w) \leftarrow r_f(v, w) - \Delta$, $r_f(w, v) \leftarrow r_f(w, v) + \Delta$
- $e(v) \leftarrow e(v) - \Delta$, $e(w) \leftarrow e(w) + \Delta$

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.

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

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

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 $e(v)$

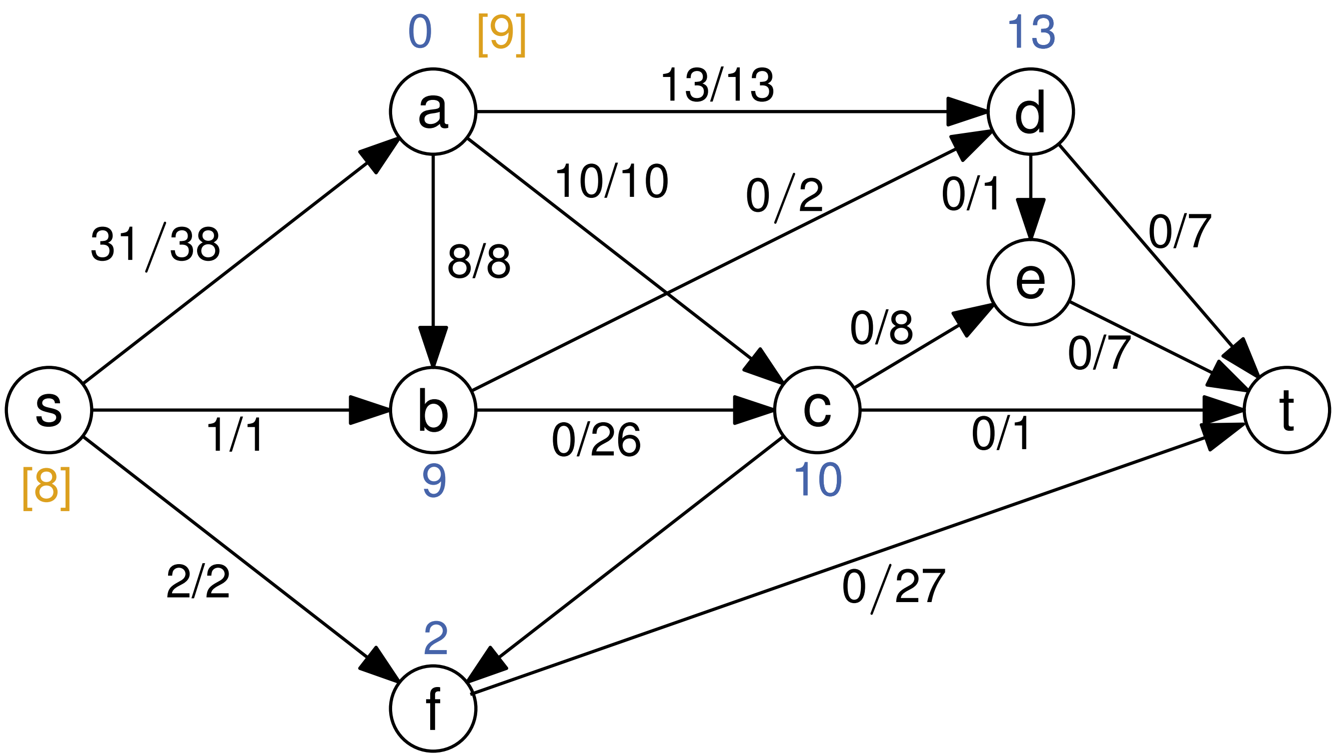
Operation PUSH(D, f, v, w)
Vorbedingung: v ist aktiv, $r_f(v, w) > 0$ und $dist(v) = dist(w) + 1$.
Effekt: Flussüberschuss wird von v nach w über Kante (v, w) geschoben.

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

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.

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

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Korrektheitsbeweis des Algorithmus von Goldberg und Tarjan

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Beweisstruktur

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.

Zulässigkeit der Operationen

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.

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

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Zulässigkeit der Operationen

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.

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

Zulässigkeit der Operationen

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:

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

$\longrightarrow$ 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.

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

Präfluss und 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.

Präfluss und 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.
$\longrightarrow$ 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 $v \in V \backslash \{s\}$ 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
- $\forall v \in V \backslash \{t\}$ und $\forall (v, w) \in E_f$: $dist(v) \leq dist(w) + 1$

Präfluss und zulässige Markierung

Karlsruher Institut für Technologie

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.
$\rightarrow$ Annahme: *Behauptung gilt nach k-ter Operation.*

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

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.

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

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

Präfluss und 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.
$\rightarrow$ 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: $dist(w) = dist(v) - 1 \leq dist(v) + 1$

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.

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

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

Präfluss und 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.
$\rightarrow$ Annahme: *Behauptung gilt nach k-ter Operation.*

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

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 $\forall (v, w) \in E_f:\ dist(v) \leq dist(w) + 1$

Präfluss und 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.
$\rightarrow$ 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:

$dist(v) \leq dist(w)$ für alle w 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 $\forall (v,w) \in E_f$: $dist(v) \leq dist(w) + 1$

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).

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 $s = v_0 \to v_1 \to \cdots \to v_\ell = t$ 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:

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

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

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

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

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 $s = v_0 \rightarrow v_1 \rightarrow \cdots \rightarrow v_\ell = t$ in D_f

Es gilt also:

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

Da $dist$ zulässig ist gilt deshalb:

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

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

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 $\forall (v, w) \in E_f$: $dist(v) \leq dist(w) + 1$

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Partielle Korrektheit des Algorithmus

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 $(D; s, t; c)$.

Partielle Korrektheit des Algorithmus

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 exisiert.
3. Nach Voraussetzung und **Lemma 4.20** gilt

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

$\longrightarrow$ **f ist ein Fluss.**

Partielle Korrektheit des Algorithmus

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 exisiert.
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 $(D; s; t; c)$

Beweisstruktur

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.

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.

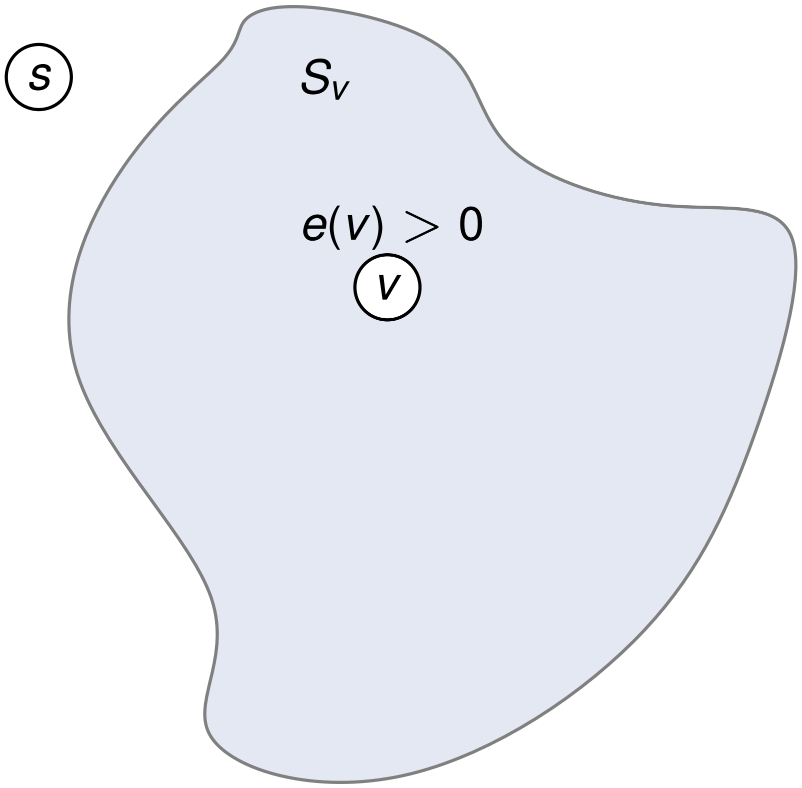
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

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.

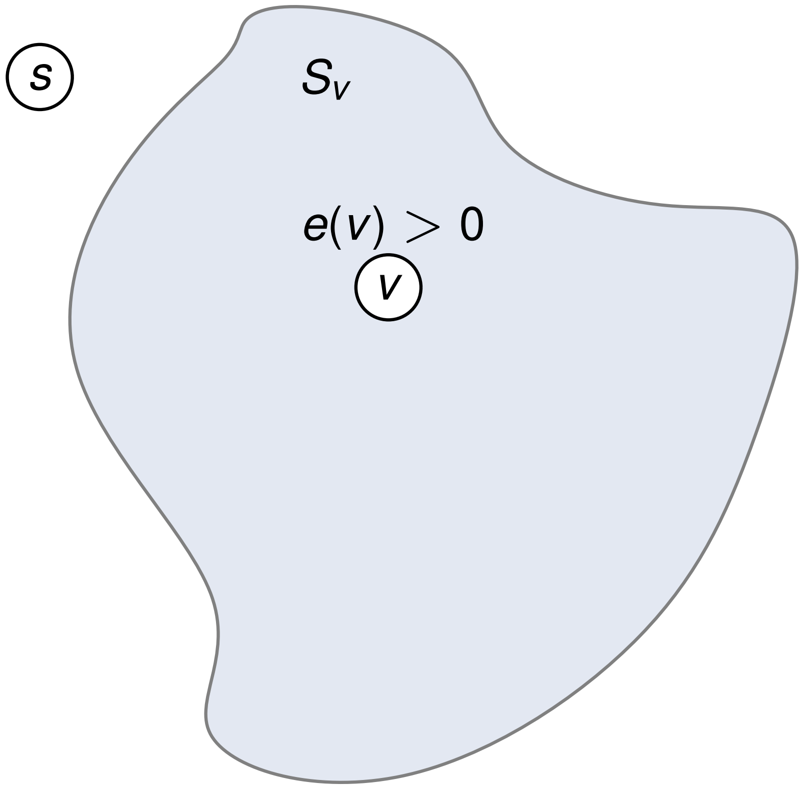
Annahme: s ist nicht in S_v enthalten.

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

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

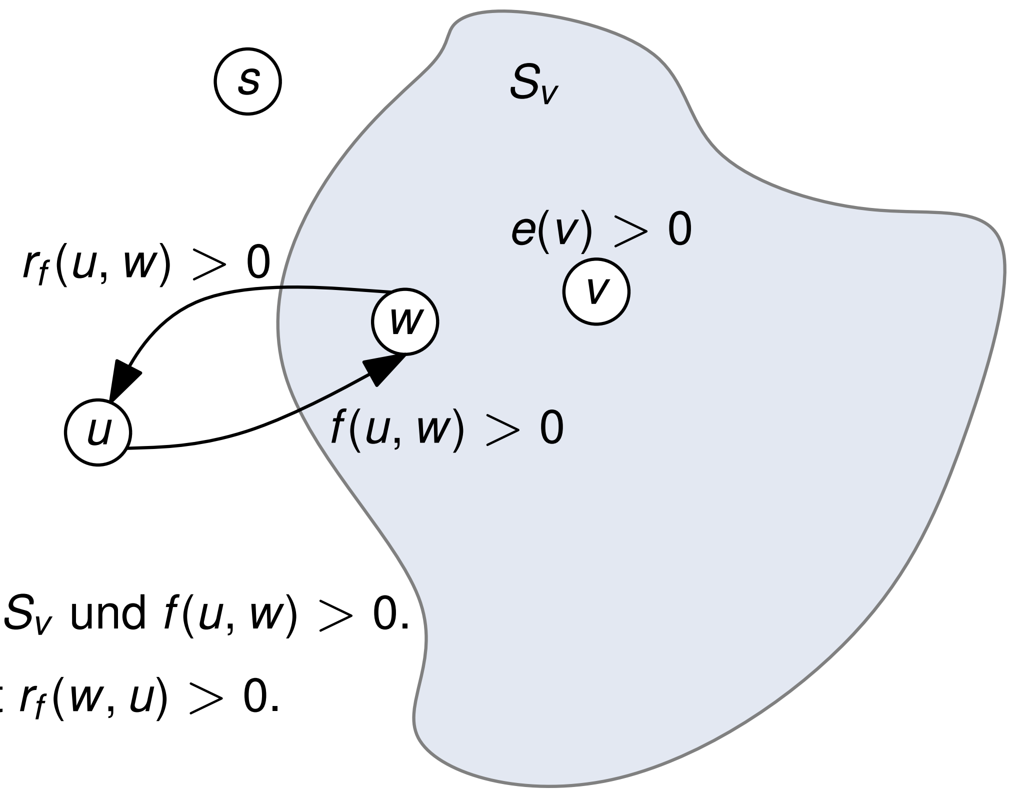


Erreichbarkeit der Quelle im Residualgraph

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.

Obere Schranke für Distanz

Karlsruher Institut für Technologie

Lemma 4.25: Während des gesamten Algorithmus gilt

$$\forall v \in V \;\; dist(v) \leq 2|V| - 1 \, .$$

Obere Schranke für Distanz

Lemma 4.25: Während des gesamten Algorithmus gilt

$$\forall v \in V \;\; dist(v) \leq 2|V| - 1 \; .$$

Beweis:

Initialisierung: Behauptung gilt offensichtlich.

Beliebiger Zeitpunkt wenn $dist(v)$ von v geändert wird:

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 $\forall (v, w) \in E_f : \; dist(v) \leq dist(w) + 1$

Obere Schranke für Distanz

Lemma 4.25: Während des gesamten Algorithmus gilt

$$\forall v \in V \;\; dist(v) \leq 2|V| - 1\ .$$

Beweis:

Initialisierung: Behauptung gilt offensichtlich.

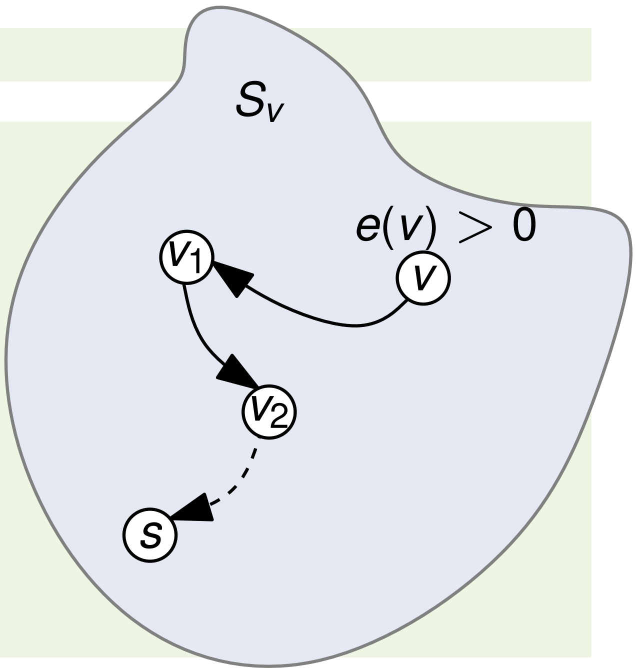
Beliebiger Zeitpunkt wenn $dist(v)$ von v geändert wird:

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

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

$$v = v_0 \longrightarrow v_1 \longrightarrow \cdots \longrightarrow v_\ell = s$$

mit $dist(v_i) \leq dist(v_{i+1}) + 1$ 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 $dist(v) \leq dist(w)$
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
- $\forall v \in V \backslash \{t\}$ und $\forall (v, w) \in E_f : \; dist(v) \leq dist(w) + 1$

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Obere Schranke für Distanz

Lemma 4.25: Während des gesamten Algorithmus gilt

$$\forall v \in V \;\; dist(v) \leq 2|V| - 1 \, .$$

Beweis:

Initialisierung: Behauptung gilt offensichtlich.

Beliebiger Zeitpunkt wenn $dist(v)$ von v geändert wird:

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

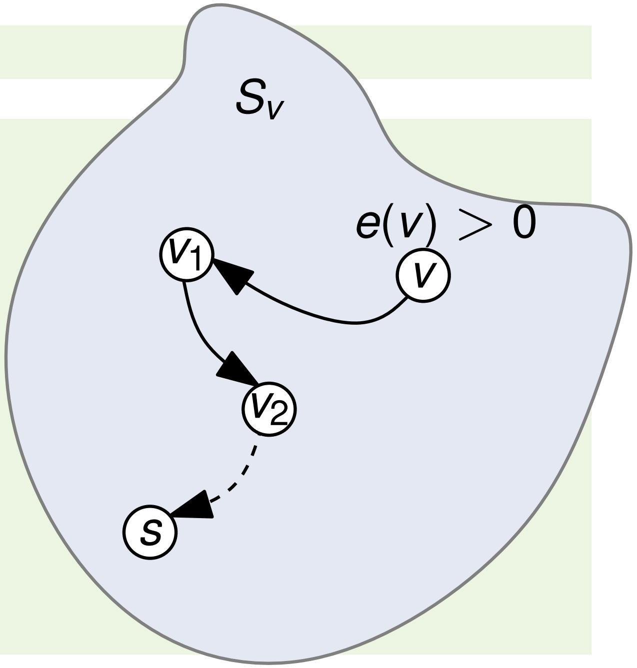
Nach **Lemma 4.24**: s ist von v in D_f erreichbar:

$$v = v_0 \longrightarrow v_1 \longrightarrow \cdots \longrightarrow v_\ell = s$$

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

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

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



Operation $\textbf{R}\textsc{elabel}$**(**$D, f, v, dist$**)**
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 $\forall (v, w) \in E_f : \; dist(v) \leq dist(w) + 1$

Obere Schranke für Distanz

Lemma 4.25: Während des gesamten Algorithmus gilt

$$\forall v \in V \;\; dist(v) \leq 2|V| - 1\ .$$

Beweis:

Initialisierung: Behauptung gilt offensichtlich.

Beliebiger Zeitpunkt wenn $dist(v)$ von v geändert wird:

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

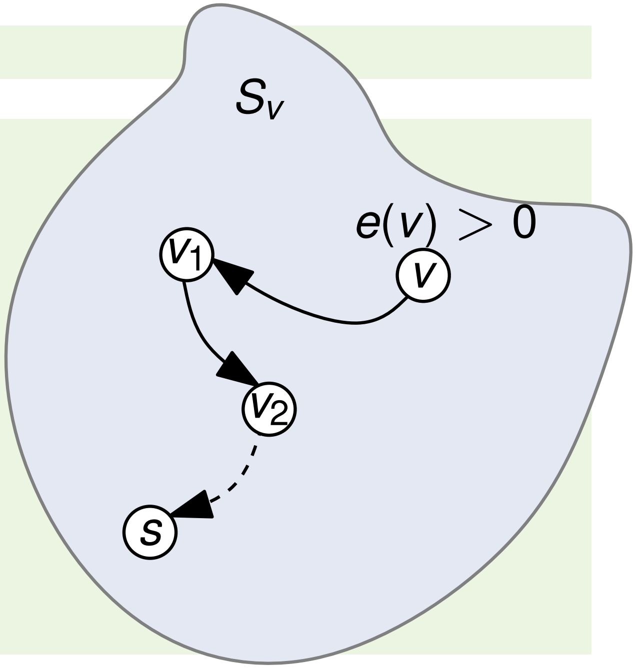
Nach **Lemma 4.24**: s ist von v in D_f erreichbar:

$$v = v_0 \longrightarrow v_1 \longrightarrow \cdots \longrightarrow v_\ell = s$$

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

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

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



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 $2|V|^2$.

Beweisstruktur

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.

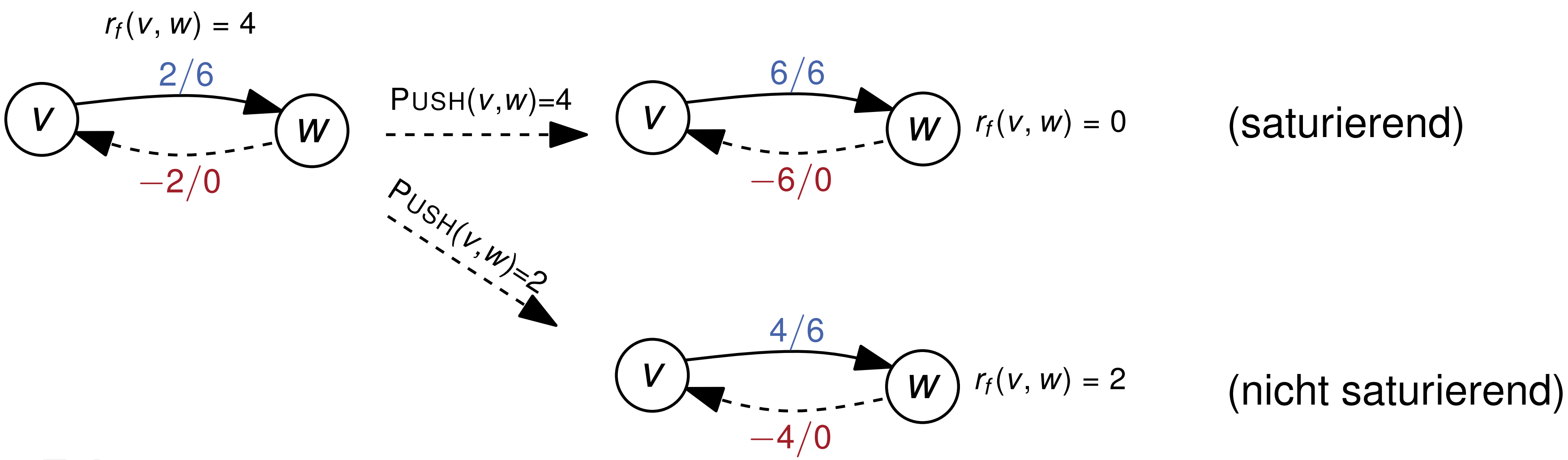
c) Finde obere Schranke für Anzahl Aufrufe von PUSH.

Bezeichne f die Abbildung, die schrittweise konstruiert wird.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Abschätzung PUSH-Operationen

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



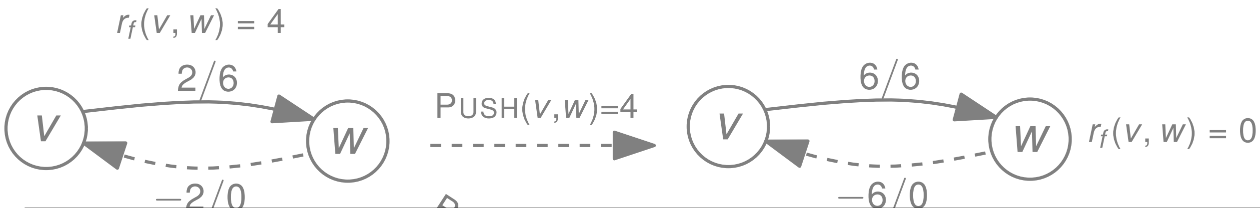
Zeige:

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.

Abschätzung PUSH-Operationen

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



(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.

ausgeführt.

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

Saturierende PUSH-Operationen

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

Saturierende PUSH-Operationen

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

Beweis: Betrachte saturierendes PUSH(v, w):

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

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.

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

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Saturierende PUSH-Operationen

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

Beweis: Betrachte saturierendes PUSH(v, w):

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

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

⟶ Erneutes PUSH(v, w) nur möglich, wenn in der Zwischenzeit *Push*(w, v).

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

⟶ Nach Ausführung vom zweiten PUSH(v, w):

$dist(v)$ und $dist(w)$ wurden um mindestens zwei erhöht.

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.

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

Saturierende PUSH-Operationen

Karlsruher Institut für Technologie

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

Beweis: Betrachte saturierendes PUSH(v, w):

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

→ Erneutes PUSH(v, w) nur möglich, wenn in der Zwischenzeit *Push*(w, v).

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

→ Nach Ausführung vom zweiten PUSH(v, w):

$dist(v)$ und $dist(w)$ wurden um mindestens zwei erhöht.

PUSH$_1(v, w)$
...
PUSH$_1(w, v)$
...
PUSH$_2(v, w)$

Nach Lemma 4.25:

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

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

Insgesamt:

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

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

Karlsruher Institut für Technologie

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

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

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 $\mathcal{D}$

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

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

Auswirkungen der Operationen auf $\mathcal{D}$

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

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

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

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

Auswirkungen der Operationen auf $\mathcal{D}$

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

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

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

$\longrightarrow$ saturiende 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 $dist(v) = dist(w) + 1$.
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 $\mathcal{D} = \sum_{\substack{v \in V \setminus \{s,t\}, \\ v \text{ aktiv}}} dist(v)$

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

Auswirkungen der Operationen auf $\mathcal{D}$

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

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

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

⟶ saturiende 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:

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

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.

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

Terminierung des Algorithmus

Karlsruher Institut für Technologie

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

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Implementation

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 $C = \max_{(u,v)\in E} c(u, v)$