Chapter 5
Graph Algorithms

Algorithm Theory
WS 2014/15

Fabian Kuhn

Network Flow: Definition

Flow: function $\boldsymbol{f: E \to \mathbb{R}_{\geq 0}}$

- $f(e)$ is the amount of flow carried by edge e

Capacity Constraints:

- For each edge $e \in E$, $f(e) \leq c_e$

Flow Conservation:

- For each node $v \in V \setminus \{s, t\}$,

$$\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$$

Flow Value:

$$|f| := \sum_{e \text{ out of } s} f((s,u)) = \sum_{e \text{ into } t} f((v,t))$$

Notation

goal: find flow f of max value

We define:

$$f^{\text{in}}(v) := \sum_{e \text{ into } v} f(e), \qquad f^{\text{out}}(v) := \sum_{e \text{ out of } v} f(e)$$

For a set $S \subseteq V$:

$$f^{\text{in}}(S) := \sum_{e \text{ into } S} f(e), \qquad f^{\text{out}}(S) := \sum_{e \text{ out of } S} f(e)$$

Flow conservation: $\forall v \in V \setminus \{s,t\}$: $f^{\text{in}}(v) = f^{\text{out}}(v)$

Flow value: $|f| = f^{\text{out}}(s) = f^{\text{in}}(t)$

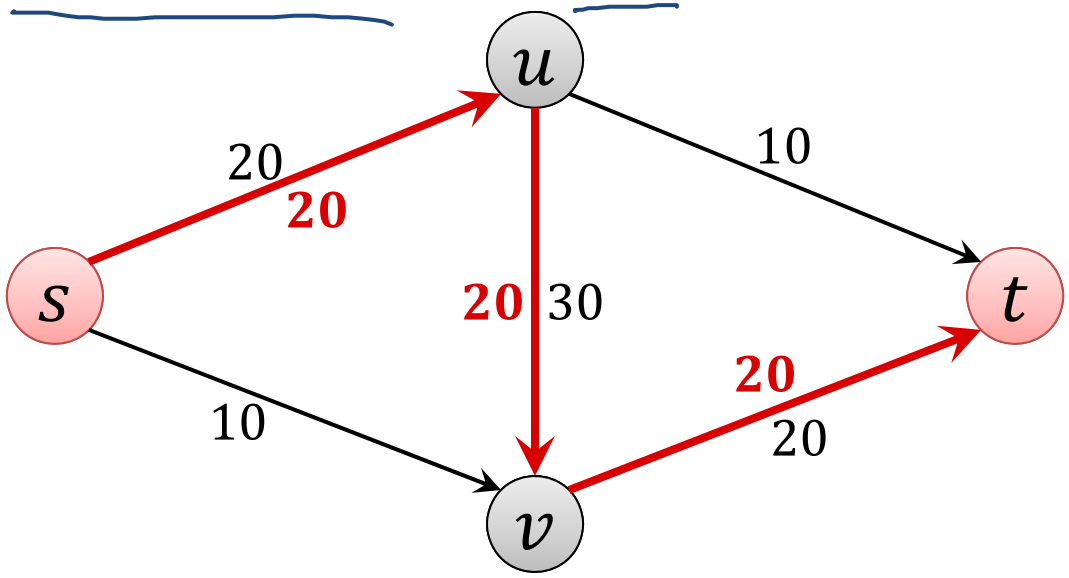
For simplicity: Assume that all capacities are positive integers

Residual Graph

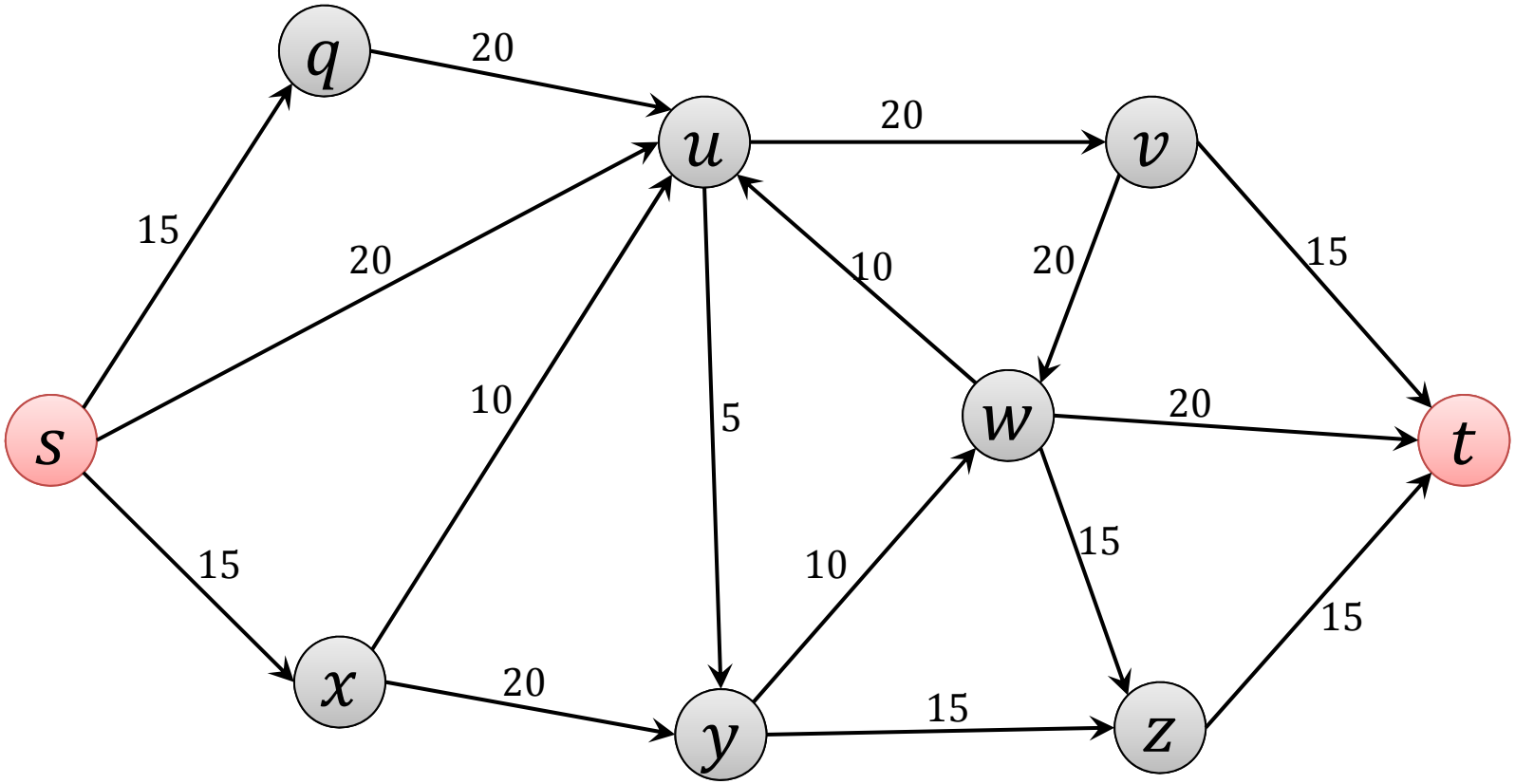
Given a flow network $G = (V, E)$ with capacities c_e (for $e \in E$)

For a flow f on G, define directed graph $G_f = (V_f, E_f)$ as follows:

- Node set $V_f = V$
- For each edge $e = (u, v)$ in E, there are two edges in E_f:
  - forward edge $e = (u, v)$ with residual capacity $c_e - f(e)$
  - backward edge $e' = (v, u)$ with residual capacity $f(e)$

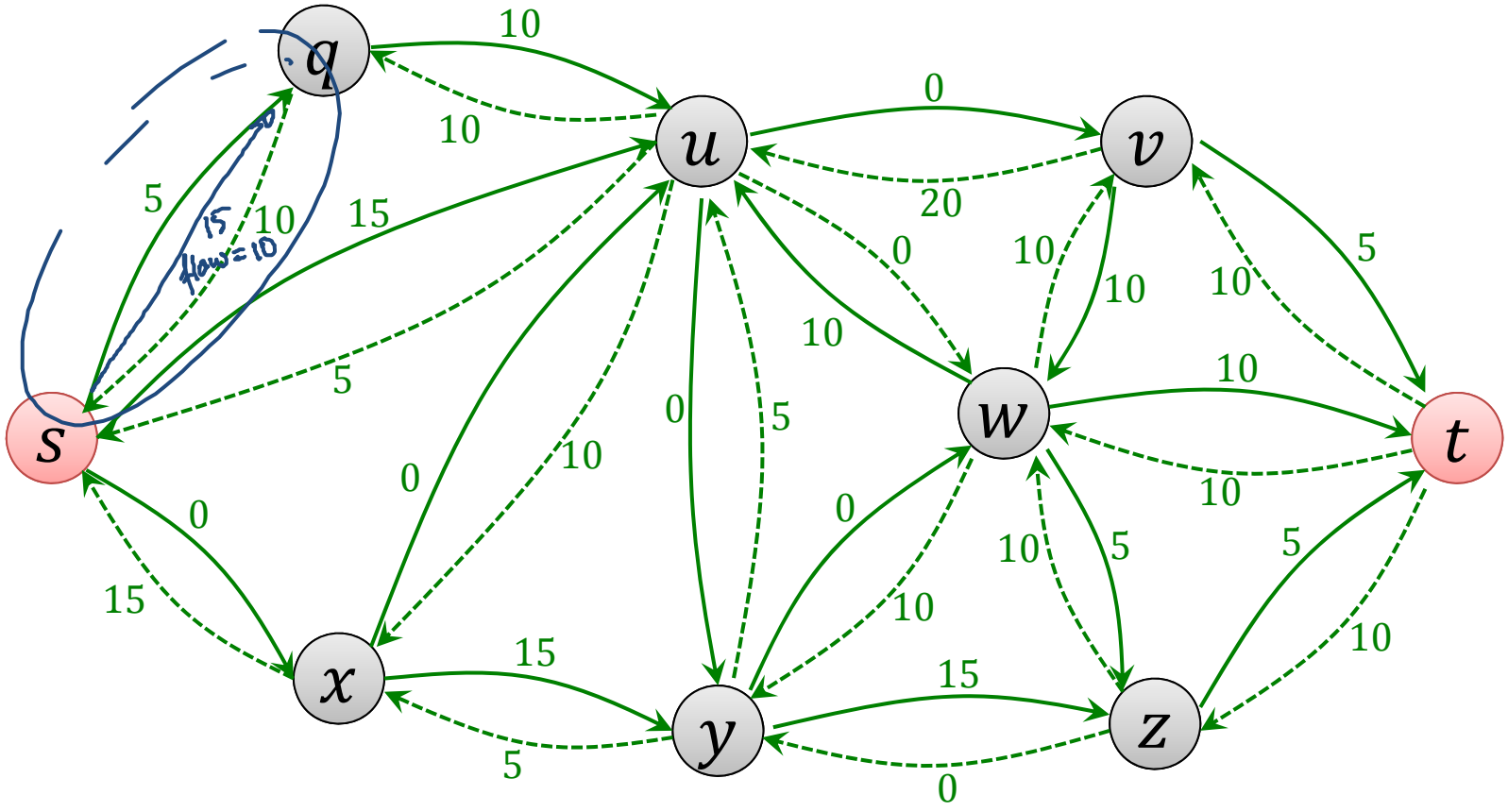


Residual Graph: Example



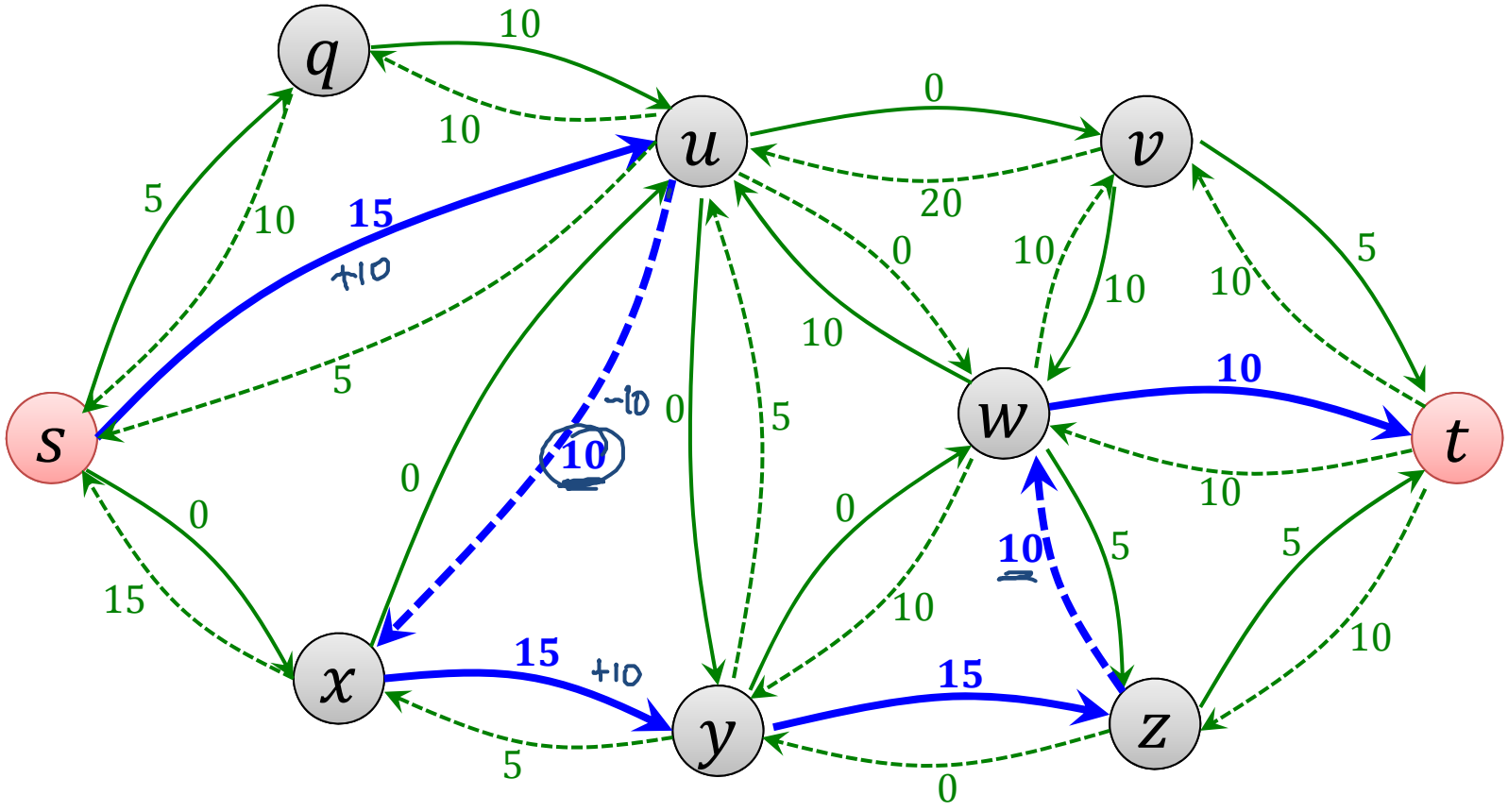
Residual Graph: Example

Residual Graph G_f



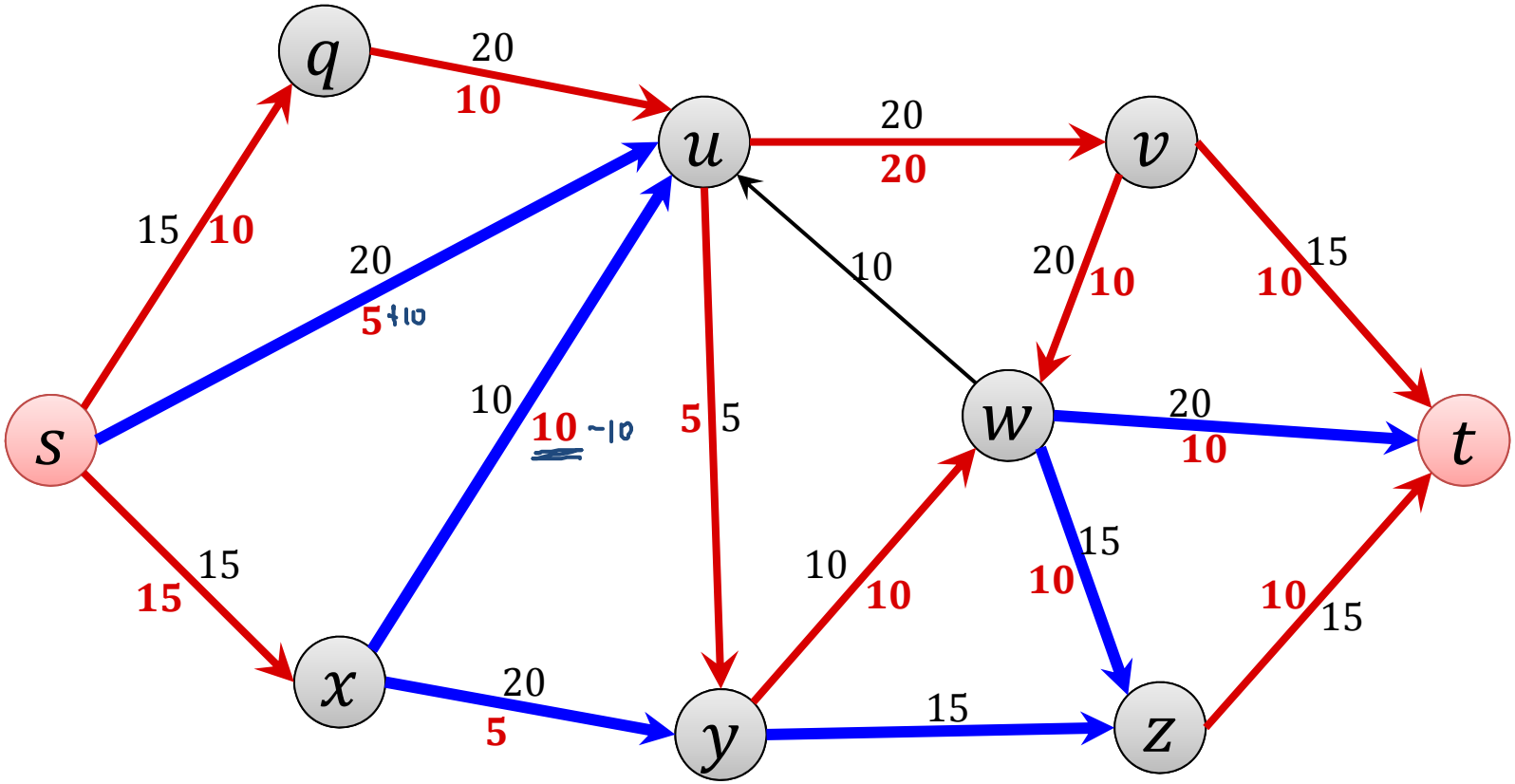
Augmenting Path

Residual Graph G_f



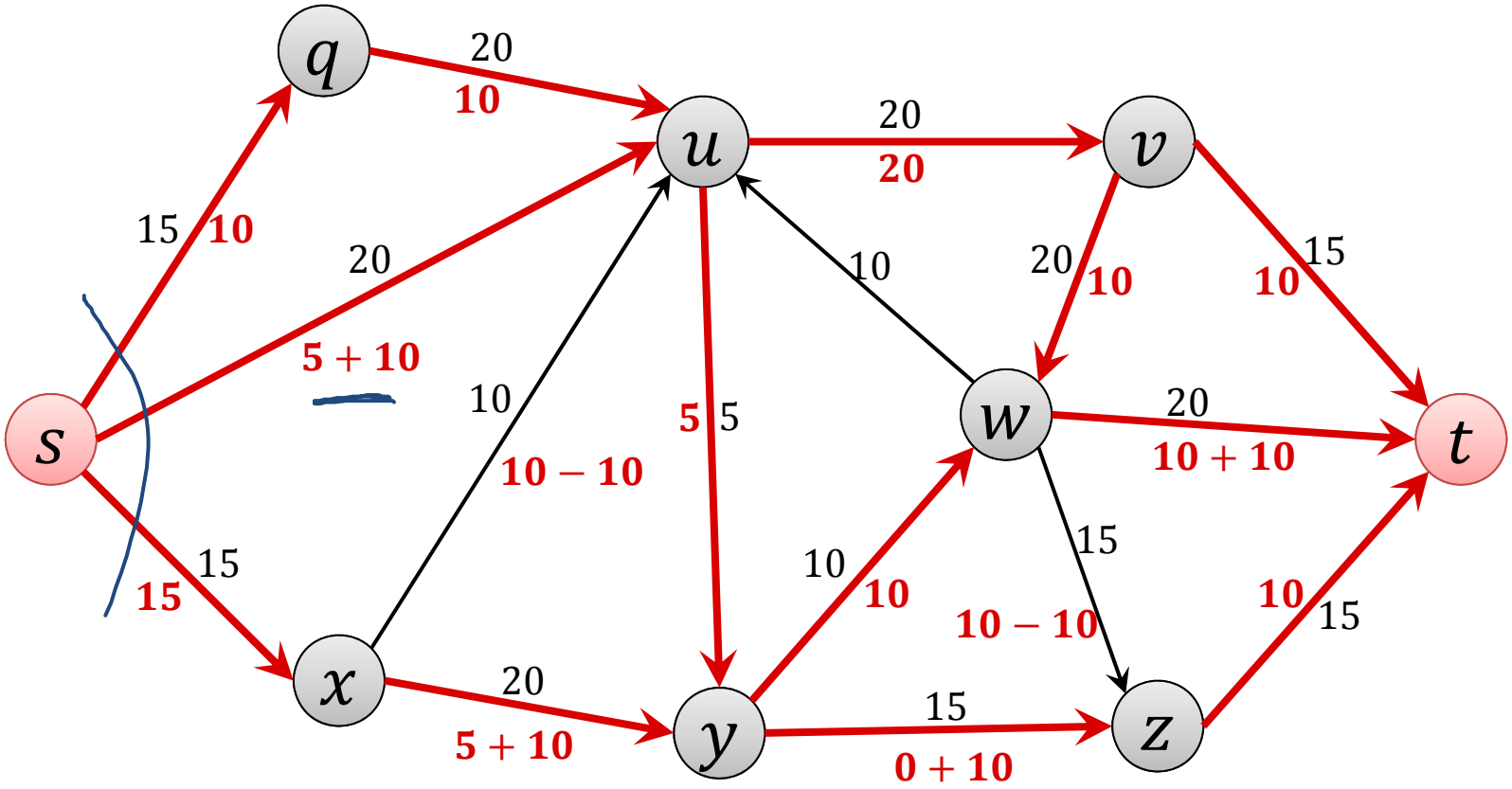
Augmenting Path

Augmenting Path



Augmenting Path

New Flow



Augmenting Path

Definition:
An augmenting path P is a (simple) s-t-path on the residual graph G_f on which each edge has residual capacity > 0.

$\text{bottleneck}(P, f)$: minimum residual capacity on any edge of the augmenting path P

Augment flow f to get flow f':

- For every forward edge (u, v) on P:

$$f'((u,v)) := f((u,v)) + \text{bottleneck}(P, f)$$

- For every backward edge (u, v) on P:

$$f'((v,u)) := f((v,u)) - \text{bottleneck}(P, f)$$

Augmented Flow

Lemma: Given a flow f and an augmenting path P, the resulting augmented flow f' is legal and its value is

$$|f'| = |f| + \mathbf{bottleneck}(P, f).$$

Proof:

Ford-Fulkerson Algorithm

- Improve flow using an augmenting path as long as possible:

1. Initially, $f(e) = 0$ for all edges $e \in E$, $G_f = G$
2. **while** there is an augmenting s-t-path P in G_f **do**
3. Let P be an augmenting s-t-path in G_f;
4. $f' := \text{augment}(f, P)$;
5. update f to be f';
6. update the residual graph G_f
7. **end**;

Ford-Fulkerson Running Time

Theorem: If all edge capacities are integers, the Ford-Fulkerson algorithm terminates after at most C iterations, where

$$C = \sum_{e \text{ out of } s} c_e .$$

Proof:

At all times, for all $e \in E$: $f(e)$ is an integer

Initially: $f(e) = 0$ ✓

In one iteration: augm. path P : residual cap. of all edges are integers

$\text{bottleneck}(f, P) > 0$ (is an integer)

$\Rightarrow \geq 1$

→ new flow integer

→ new flow value larger by ≥ 1

every flow value $\leq C$

Ford-Fulkerson Running Time

Theorem: If all edge capacities are integers, the Ford-Fulkerson algorithm can be implemented to run in $O(mC)$ time.

Proof:

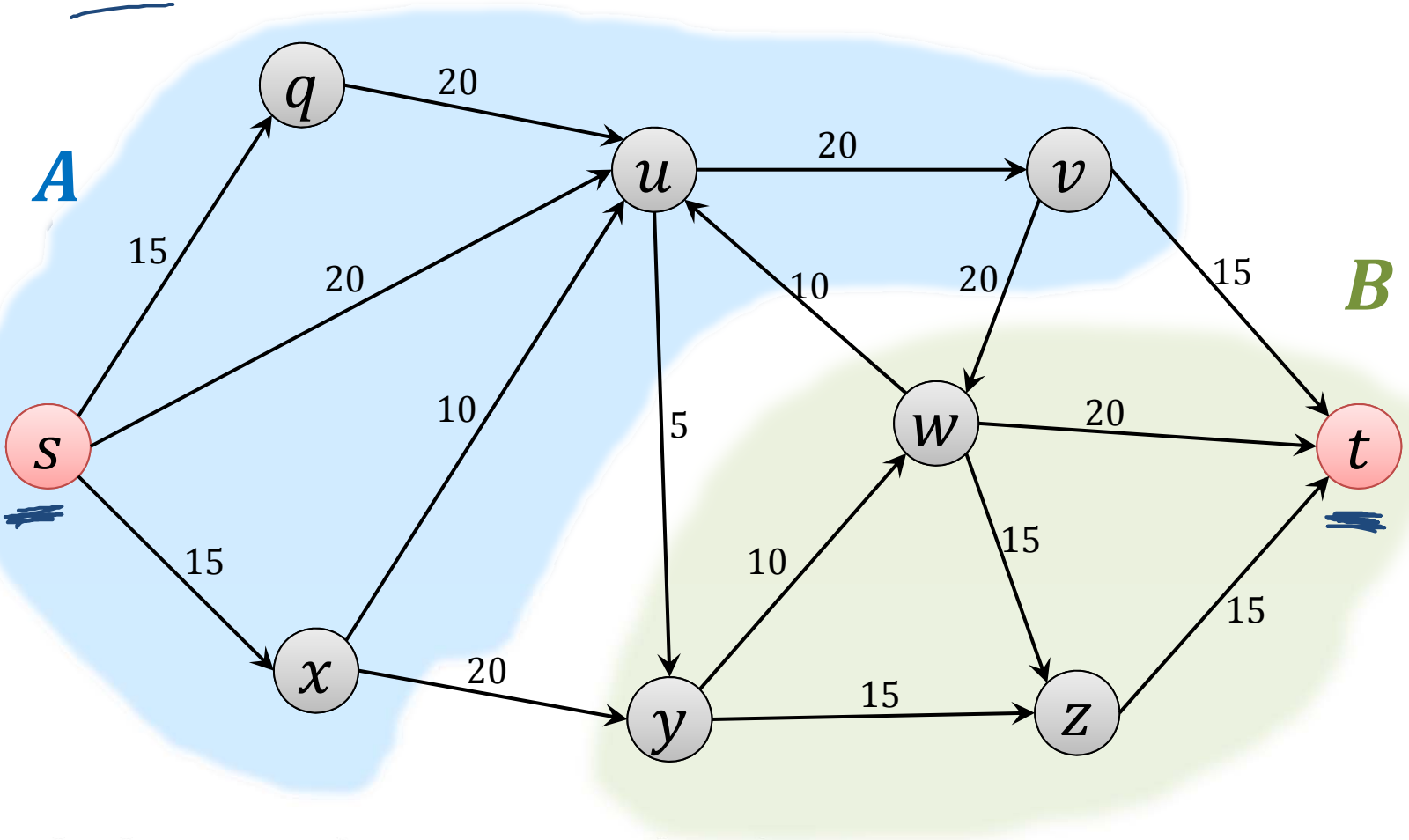
Claim: one iteration can be computed in $O(m)$ time

1. compute / update residual graph G_f → first iter.: $O(m)$; later iter.: $O(n)$
2. find augmenting path / conclude there is no aug. p.
   s-t path in G_f with res. cap. > 0
   → find augmenting path: DFS search / BFS search — $O(m)$ time
3. update flow values : $O(n)$ time

s-t Cuts

Definition:

An s-t cut is a partition (A, B) of the vertex set such that $s \in A$ and $t \in B$

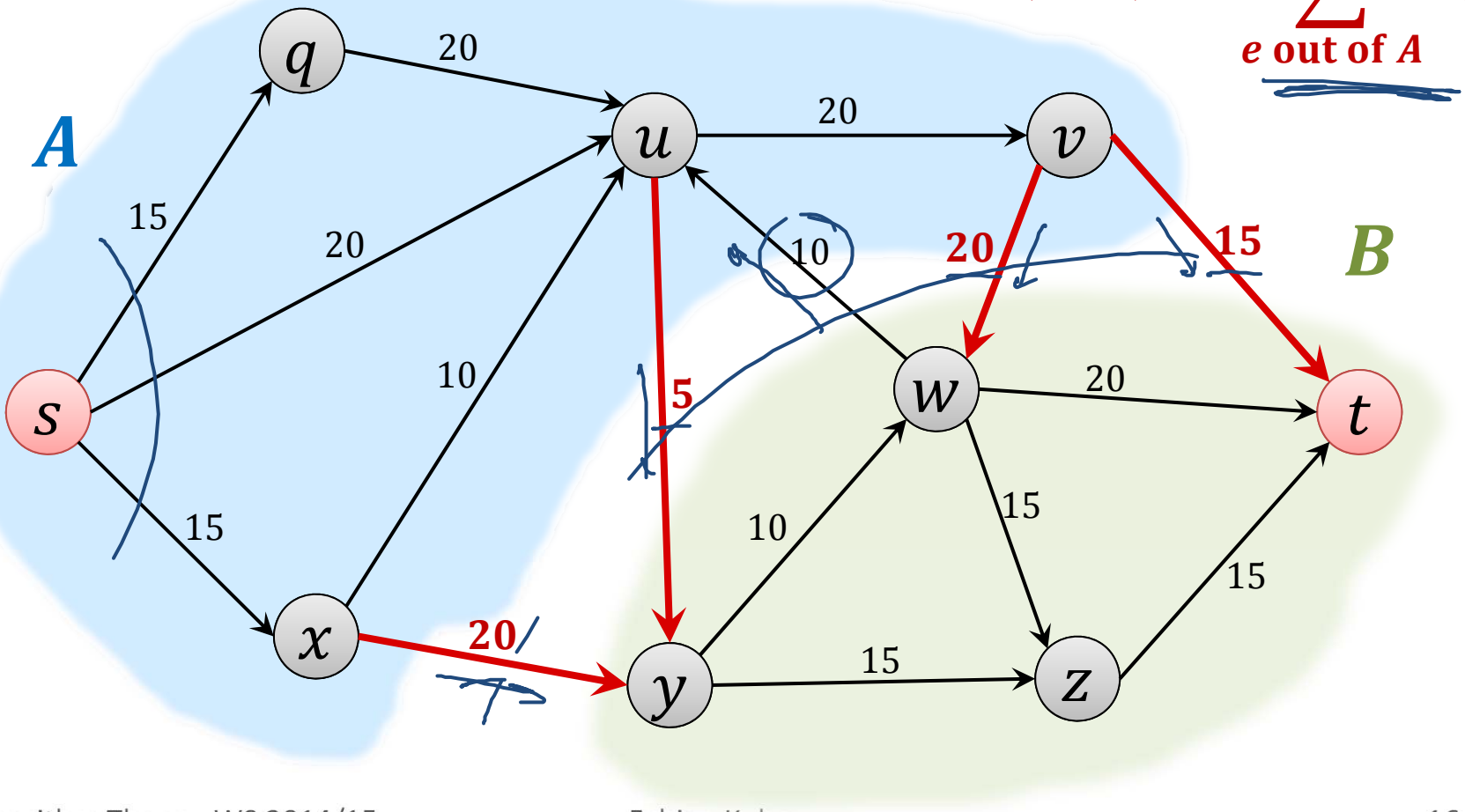


Cut Capacity

Definition:

The capacity $c(A,B)$ of an s-t-cut (A,B) is defined as

$$c(A,B) := \sum_{e \text{ out of } A} c_e .$$



Cuts and Flow Value

Lemma: Let f be any s-t flow, and (A, B) any s-t cut. Then,

$$|f| = f^{\text{out}}(A) - f^{\text{in}}(A).$$

Proof:

$$|f| = f^{out}(s) = f^{in}(t)$$

$$|f| = f^{out}(s) - \underbrace{f^{in}(s)}_{=0}$$

flow conservation

$$= \sum_{v \in A} \left(f^{out}_{(v)} - f^{in}_{(v)}\right) \qquad \left(\forall v \in A \setminus \{s\} : f^{out}_{(v)} = f^{in}_{(v)}\right)$$

$=0$ except for $v = s$

$$= f^{out}(A) - f^{in}(A)$$

Cuts and Flow Value

Lemma: Let f be any s-t flow, and (A, B) any s-t cut. Then,

$$|f| = f^{\text{out}}(A) - f^{\text{in}}(A).$$

Lemma: Let f be any s-t flow, and (A, B) any s-t cut. Then,

$$|f| = f^{\text{in}}(B) - f^{\text{out}}(B).$$

Proof:

symmetric

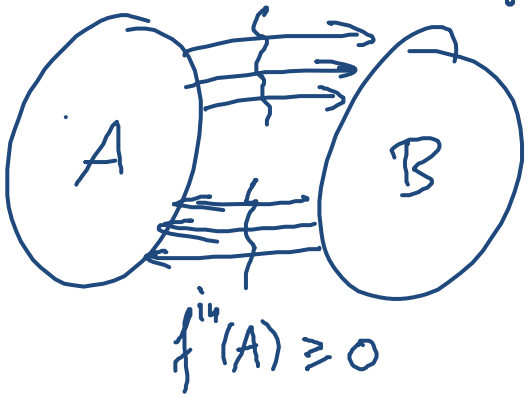
Upper Bound on Flow Value

Lemma:

Let f be any s-t flow and (A, B) and s-t cut. Then $\boldsymbol{|f| \leq c(A,B)}$.

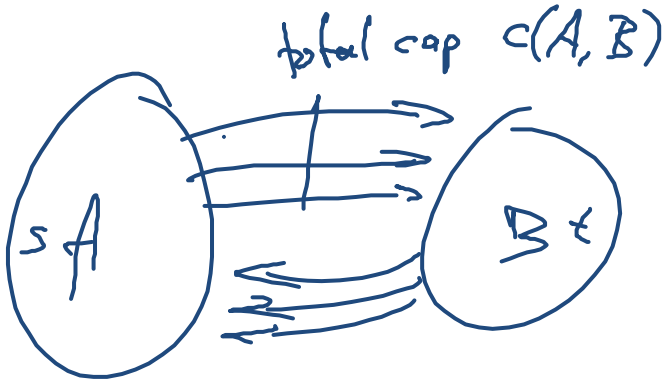
Proof:

$$f^{out}(A) \leq \sum_{\text{edges leaving } A} c_e = c(A,B)$$



$$|f| = f^{out}(A) - f^{in}(A)$$

$$\leq c(A,B) - 0$$



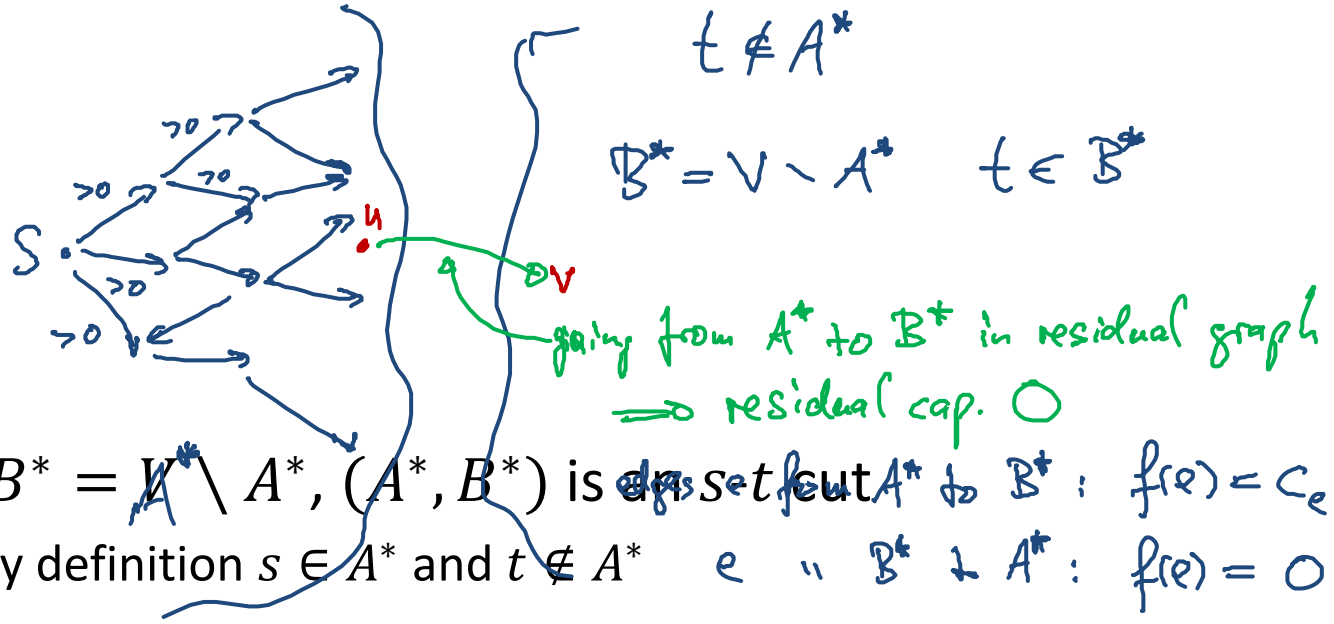
Ford-Fulkerson Gives Optimal Solution

Lemma: If f is an s-t flow such that there is no augmenting path in G_f, then there is an s-t cut (A^*, B^*) in G for which

$$|f| = c(A^*, B^*).$$

Proof:

- Define A^*: set of nodes that can be reached from s on a path with positive residual capacities in G_f:



- For $B^* = V \setminus A^*$, (A^*, B^*) is an s-t cut
  - By definition $s \in A^*$ and $t \notin A^*$