Network Flows

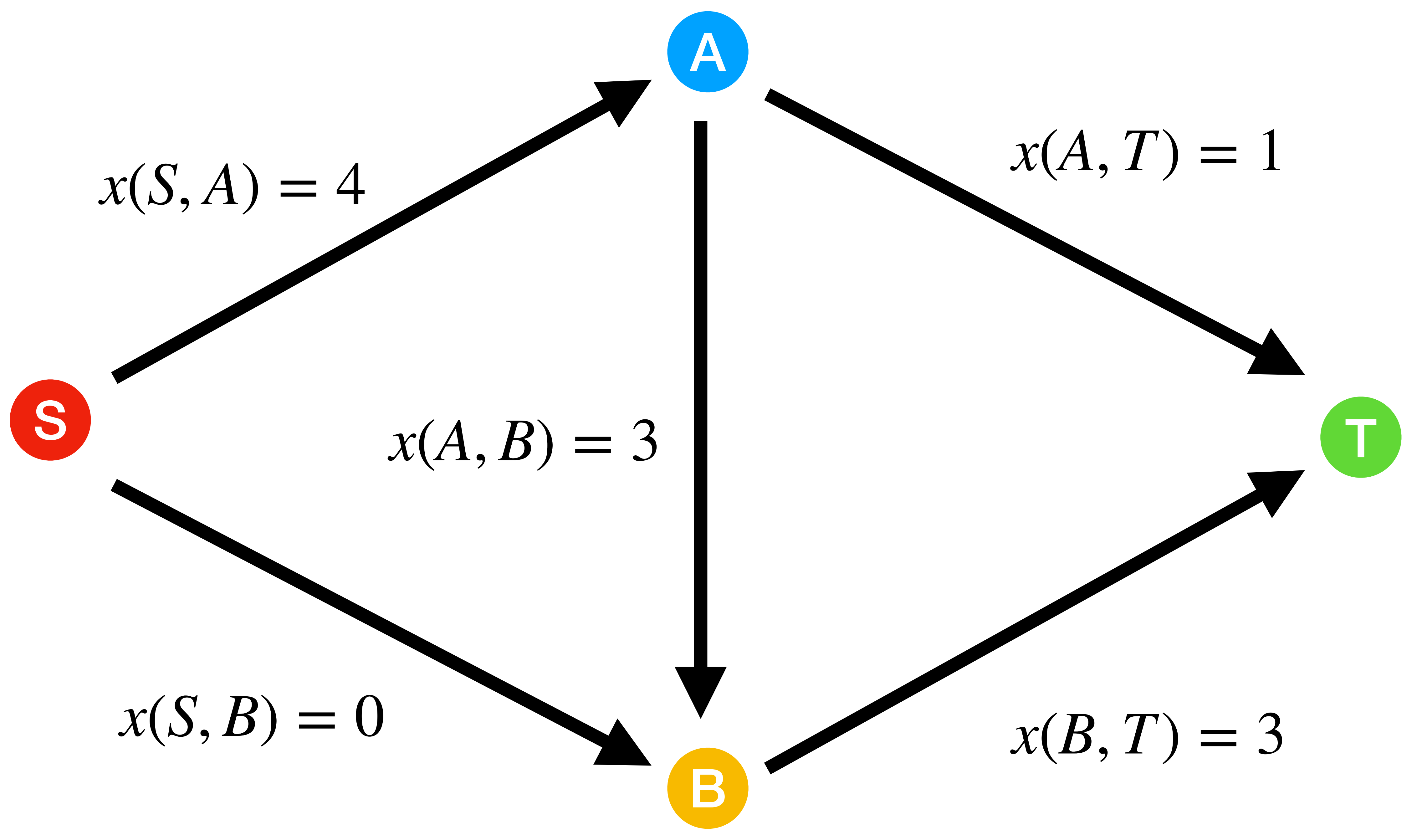
RampUp Discrete Optimization

Network flows

Given a digraph $G(V, A)$ with two distinct vertices $s, t \in V$. A function

$$x\colon A \to \mathbb{R}_{\geq 0},$$

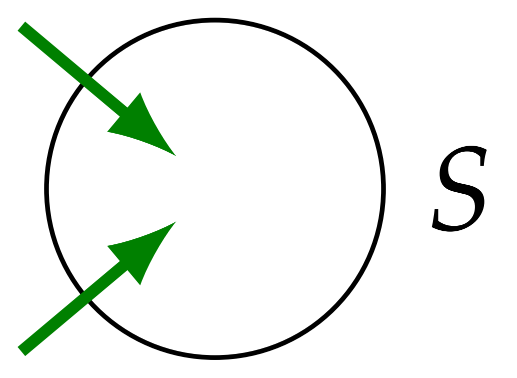
is called an *s-t-flow,*
if it fulfills *flow conservation*
for every vertex except s and t , i.e.,
the incoming flow equals the outgoing flow.


A
S
T
B
$x(S, A) = 4$
$x(A, T) = 1$
$x(A, B) = 3$
$x(S, B) = 0$
$x(B, T) = 3$

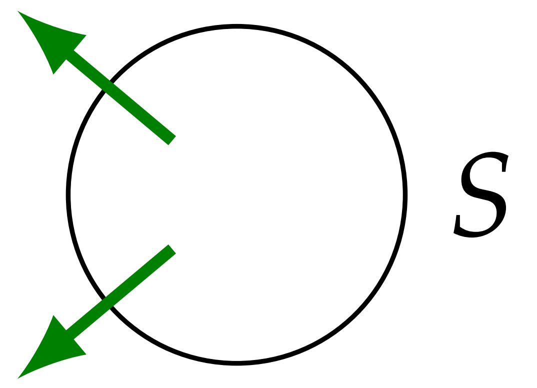
A Cut and its Boundary

A subset $S \subseteq V$ of the vertices is called a *cut* and its boundary consists of

$$\delta^+(S) = \{(v,w) \mid v \notin S,\ w \in S\}$$



$$\delta^-(S) = \{(v,w) \mid v \in S,\ w \notin S\}.$$



A cut S that contains the sink t but not the source s is called an *s-t-cut*.

MaxFlow Problem

Definition:
Excess of a vertex set

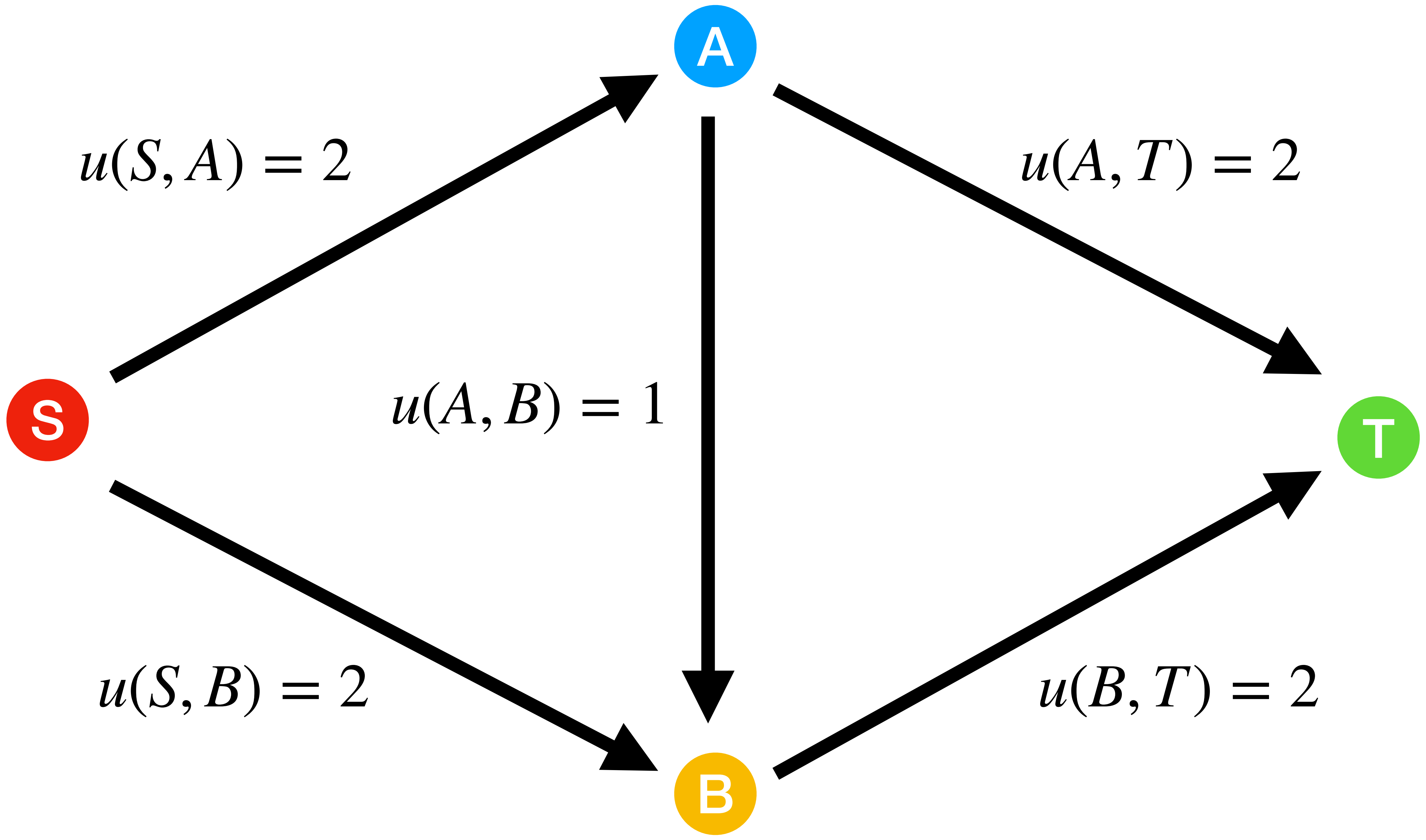
$$\mathrm{ex}_x(S) = \sum_{a \in \delta^+(S)} x_a - \sum_{a \in \delta^-(S)} x_a.$$

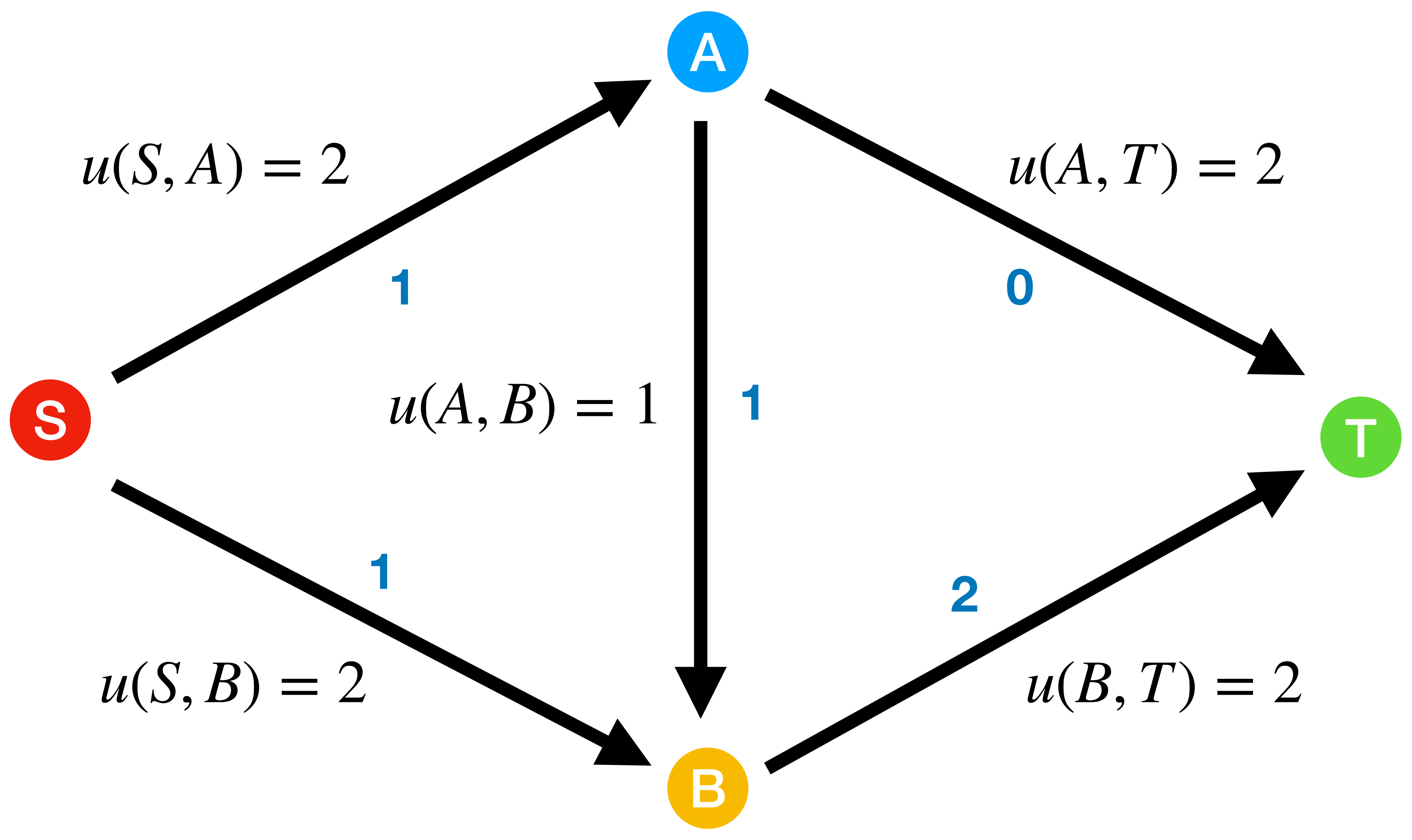
Given a non-negative function u on the arc set A, we define the **MaxFlow Problem:**

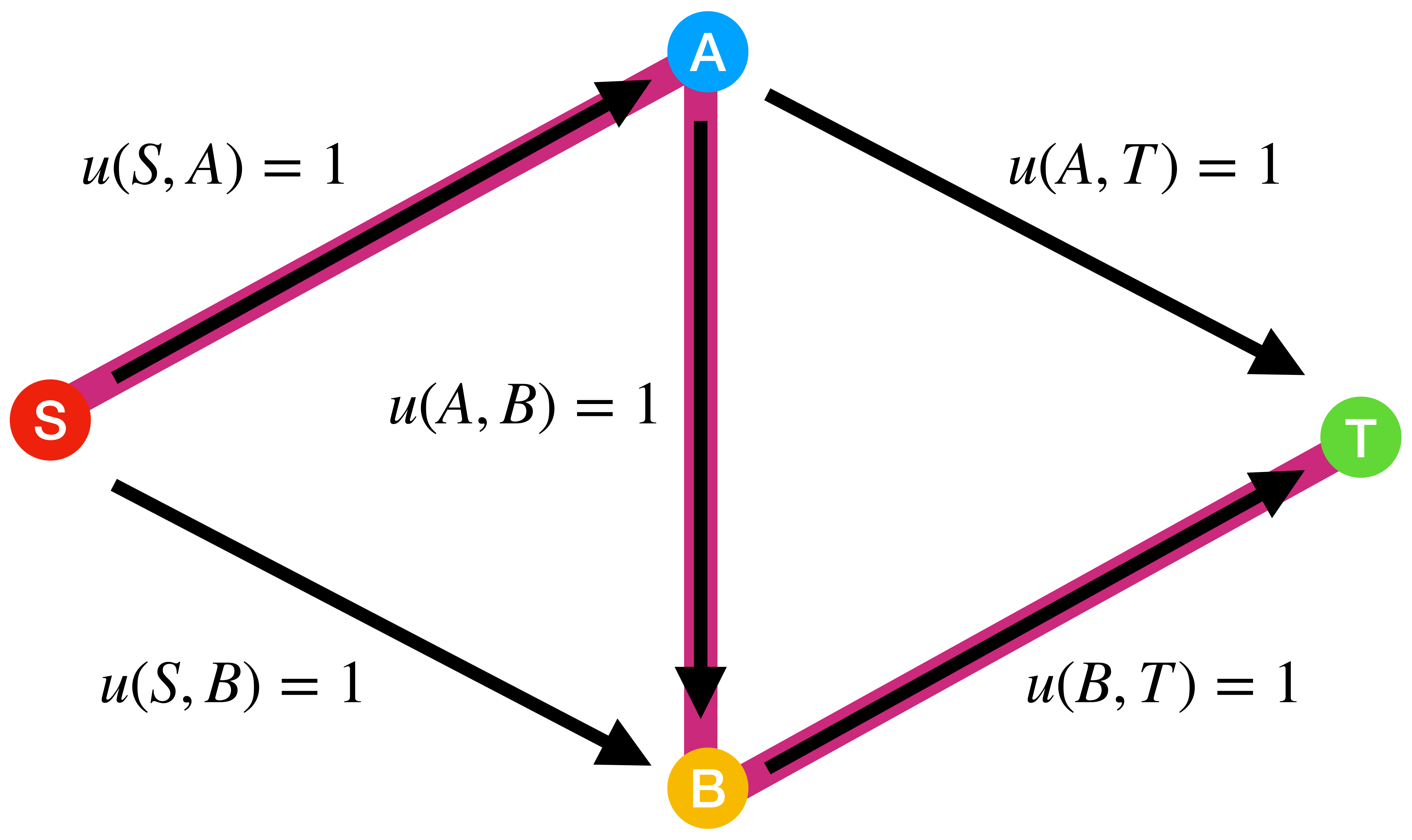
$$\begin{array}{llll} \max_x & \mathrm{ex}_x(t) & & \\ \mathrm{s.\,d.} & \mathrm{ex}_x(v) & = 0 & \forall v \in V \setminus \{s,t\} \\ & x_a & \leq u_a & \forall a \in A \\ & x_a & \geq 0 & \forall a \in A \end{array}$$

Objective function:
Flow Value
Flow conservation
Capacity
Non-negativity

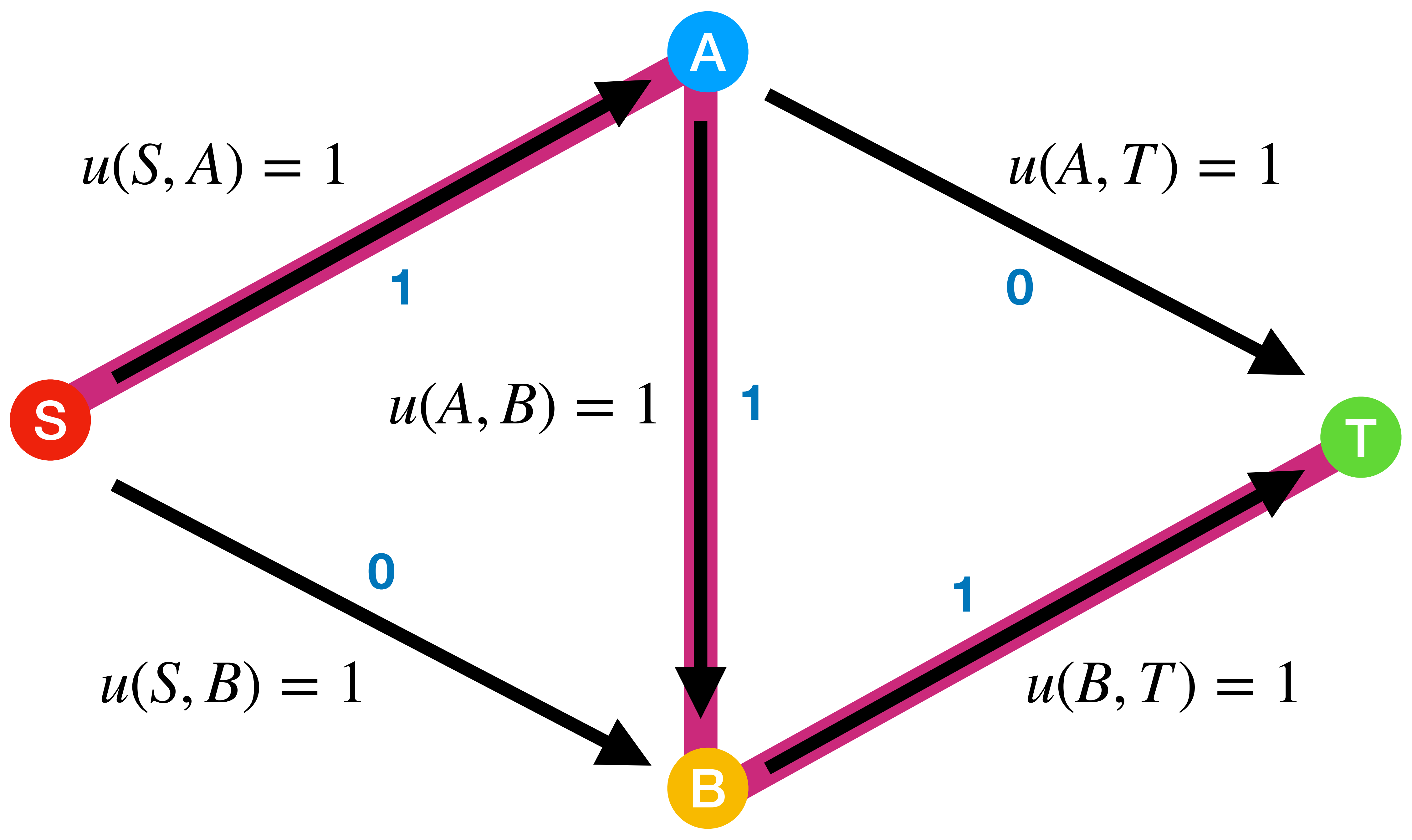
The capacity of a cut S is defined as: $u(\delta^+(S)) = \sum_{a \in \delta^+(S)} u(a).$


A
S
T
B
$u(S, A) = 2$
$u(A, T) = 2$
$u(A, B) = 1$
$u(S, B) = 2$
$u(B, T) = 2$


A
S
T
B
$u(S, A) = 2$
$u(A, T) = 2$
$u(A, B) = 1$
$u(S, B) = 2$
$u(B, T) = 2$
1
0
1
1
2

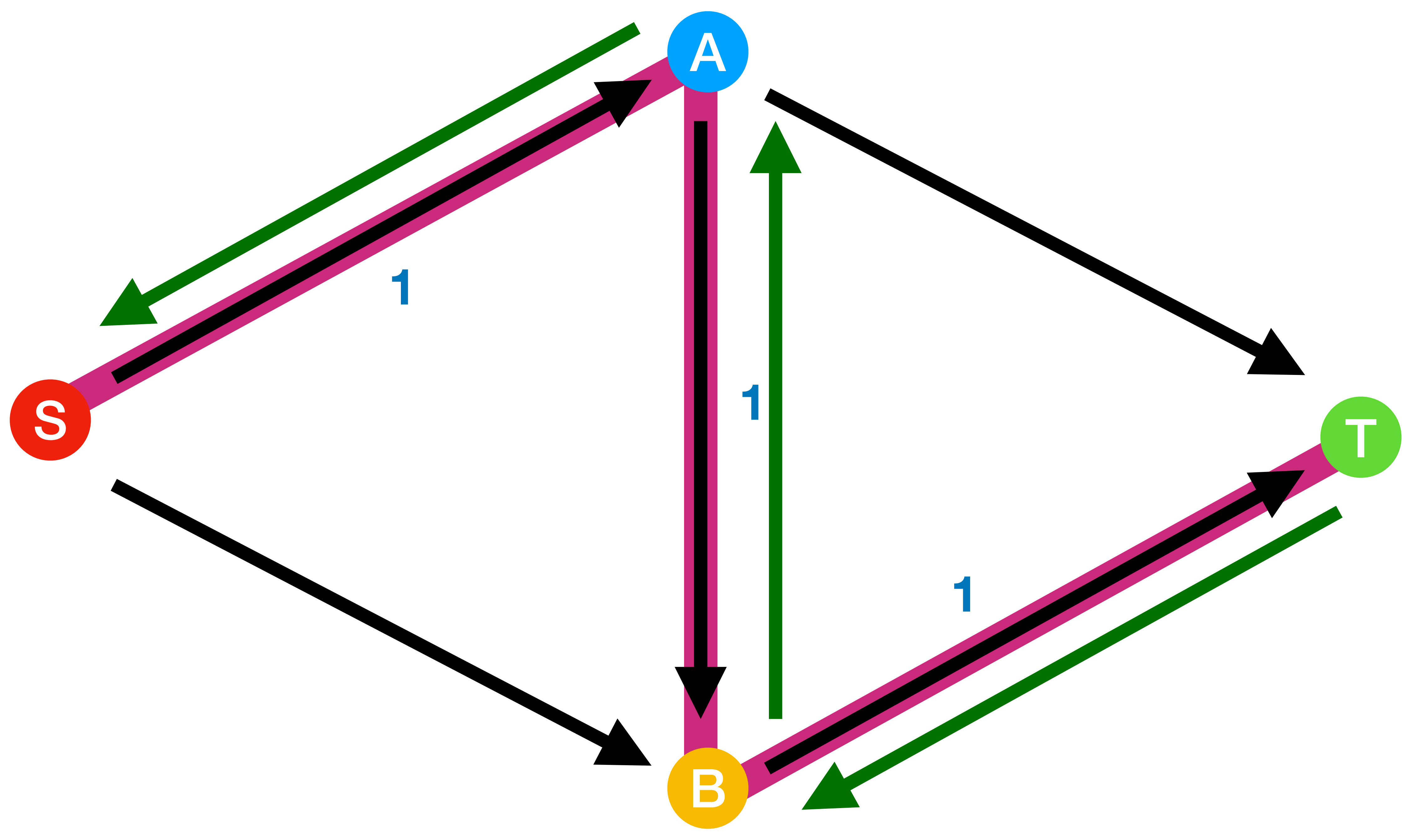


S
A
B
T
$u(S, A) = 1$
$u(A, T) = 1$
$u(A, B) = 1$
$u(S, B) = 1$
$u(B, T) = 1$

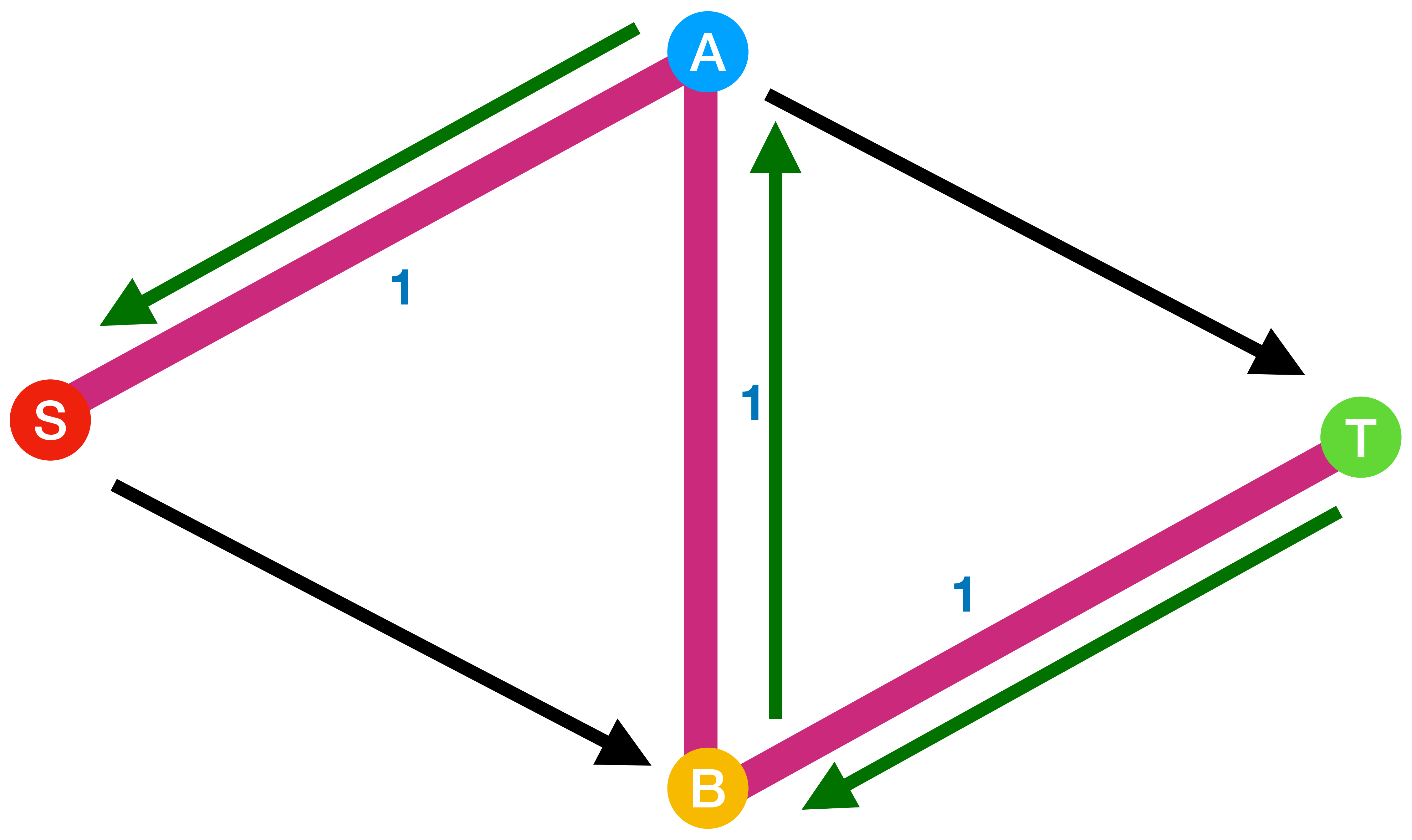


A
S
T
B
$u(S, A) = 1$
$u(A, T) = 1$
$u(A, B) = 1$
$u(S, B) = 1$
$u(B, T) = 1$
1
0
1
0
1

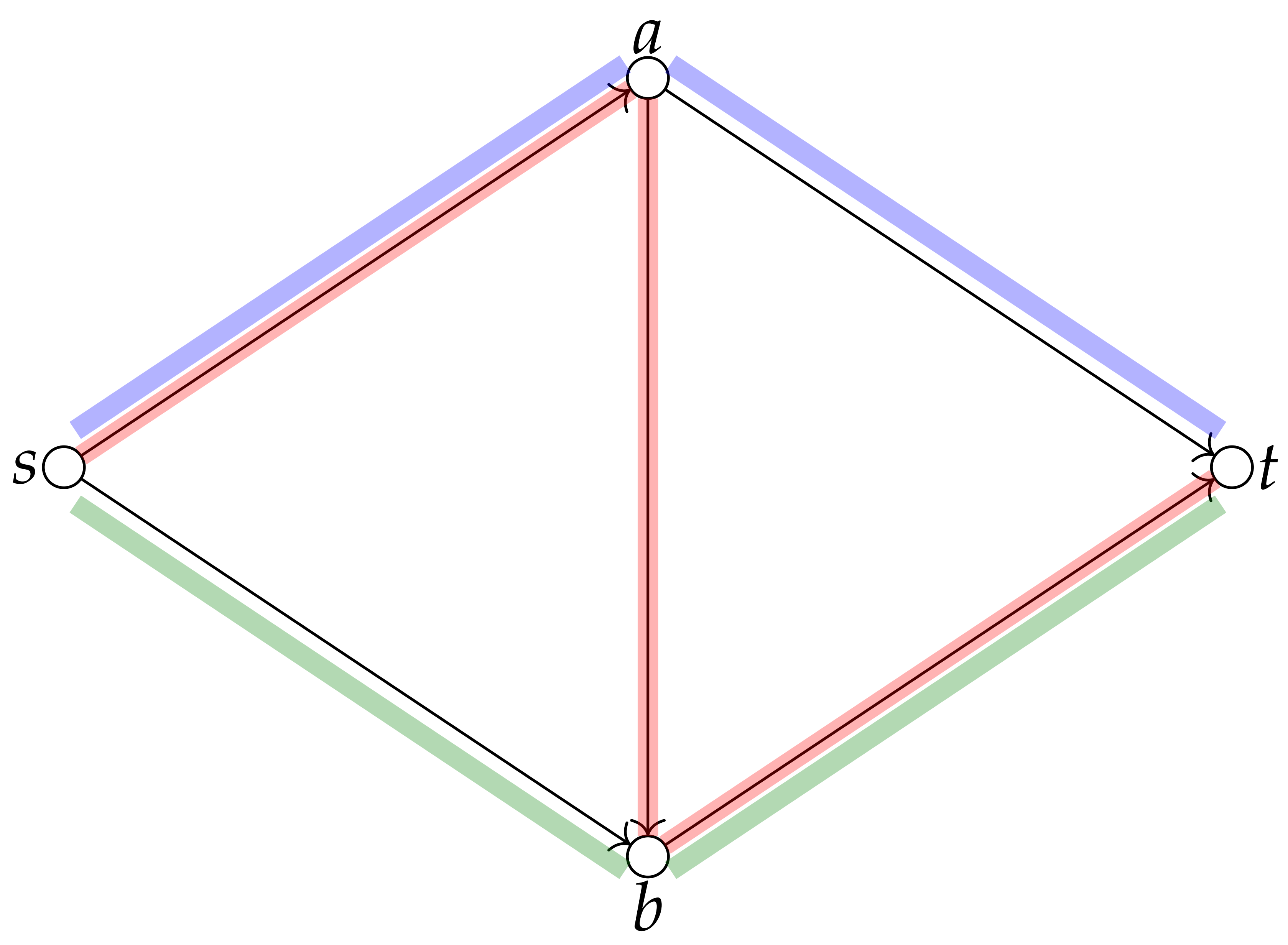
Residual Arcs

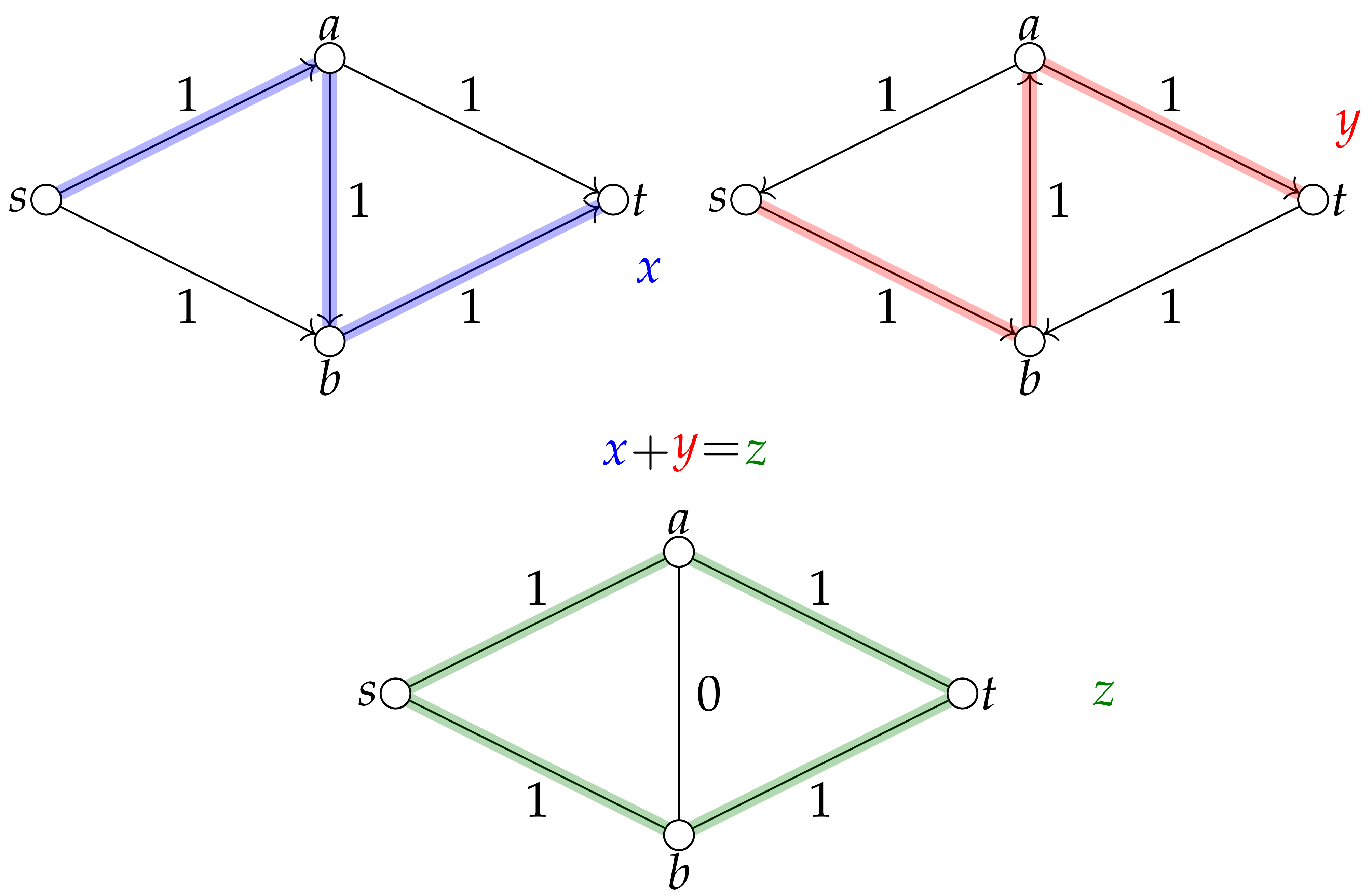


Residual Graph

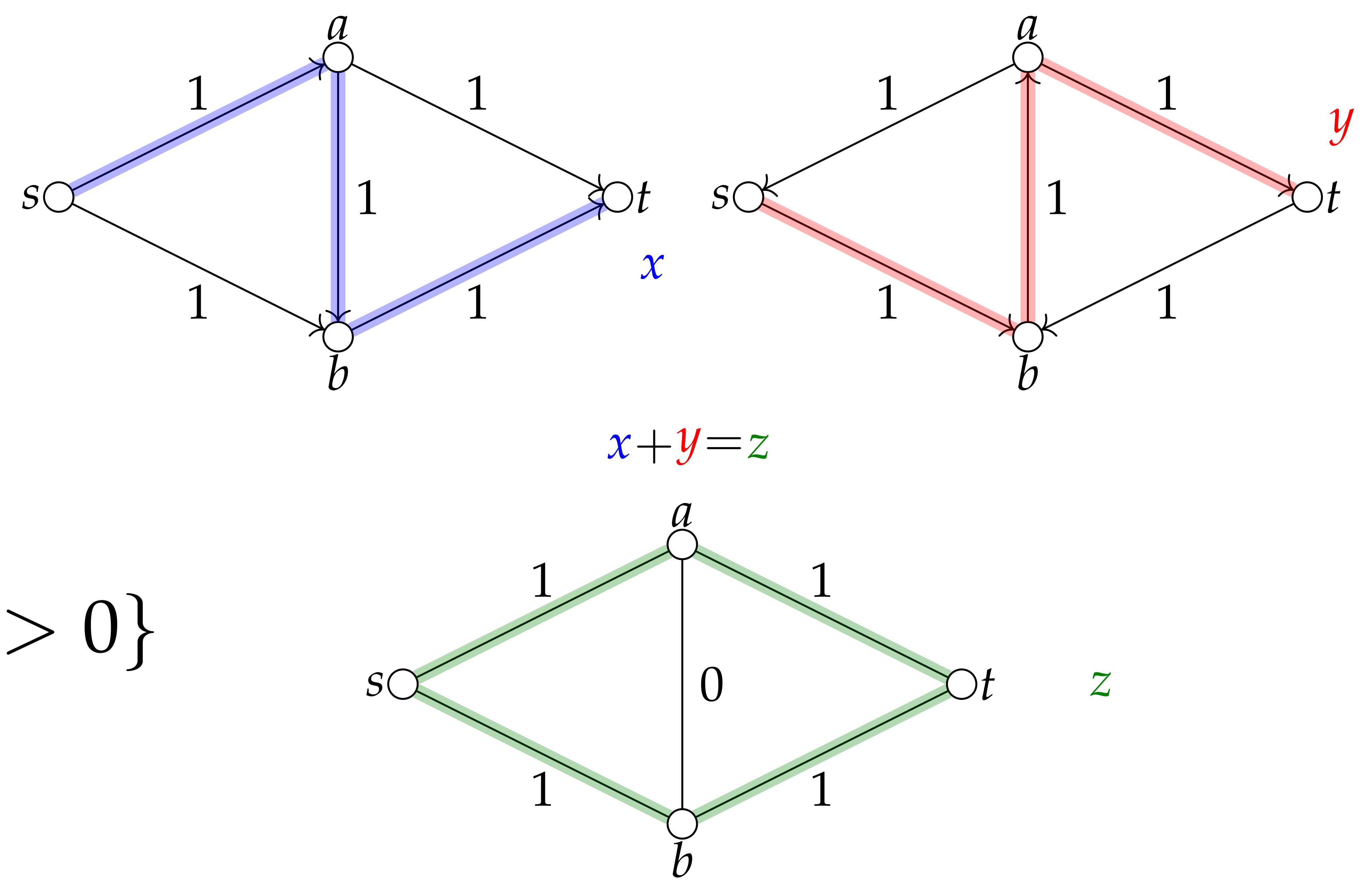


Residual Graph





Residual Graph $G_x(V, A_x)$, u_x



$$A_x := \{a \in A \mid x(a) < u(a)\} \cup \{a^{-1} \in A^{-1} \mid x(a) > 0\}$$

$$u_x(a) := \begin{cases} u(a) - x(a) & , a \in A \\ x(a^{-1}) & , a^{-1} \in A \end{cases}$$

Ford-Fulkerson Algorithm

1. Initialize by the zero flow: $x(a) = 0 \quad \forall a \in A$.

2. Construct residual graph G_x, u_x .

3. IF there is no path from source to sink in G_x: RETURN x.

4. ELSE find path P from source to sink in residual graph G_x, u_x and smallest residual capacity λ of an arc on P.

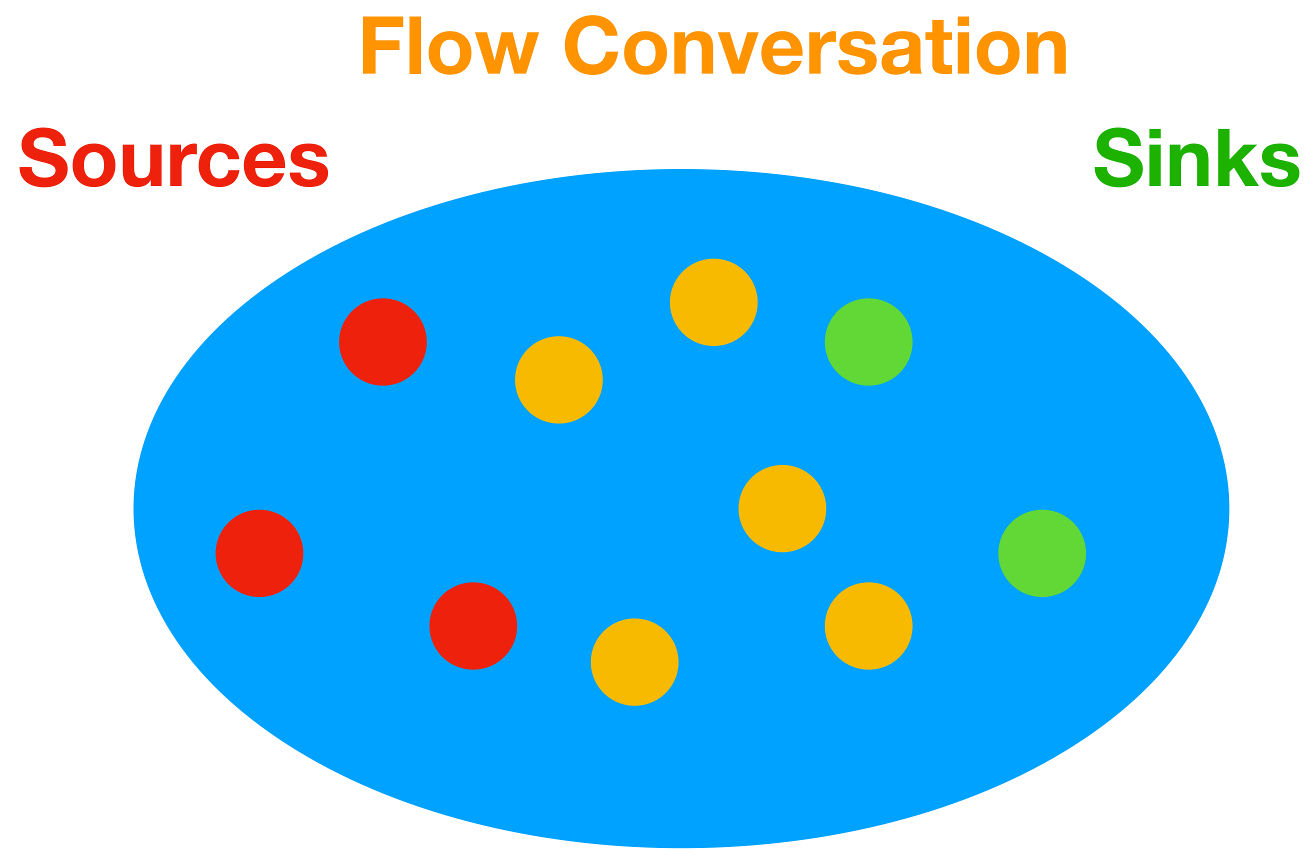
5. Augment x along P by λ. Goto 2.

Why does Ford-Fulkerson terminate with an optimal MaxFlow?

- Let x be the flow when FF terminates, i.e., there is no path from s to t in G_x.
- Consider the set S of vertices, from which the sink is still reachable in G_x.
- Because $\delta^+(S) = \emptyset$, the s-t-cut S has capacity equal to the flow value of x.
- As every cut's capacity is an upper bound on the value of every flow, the flow x must be maximal (and the s-t-cut S minimal).

MinCost Flow

b-Transshipment



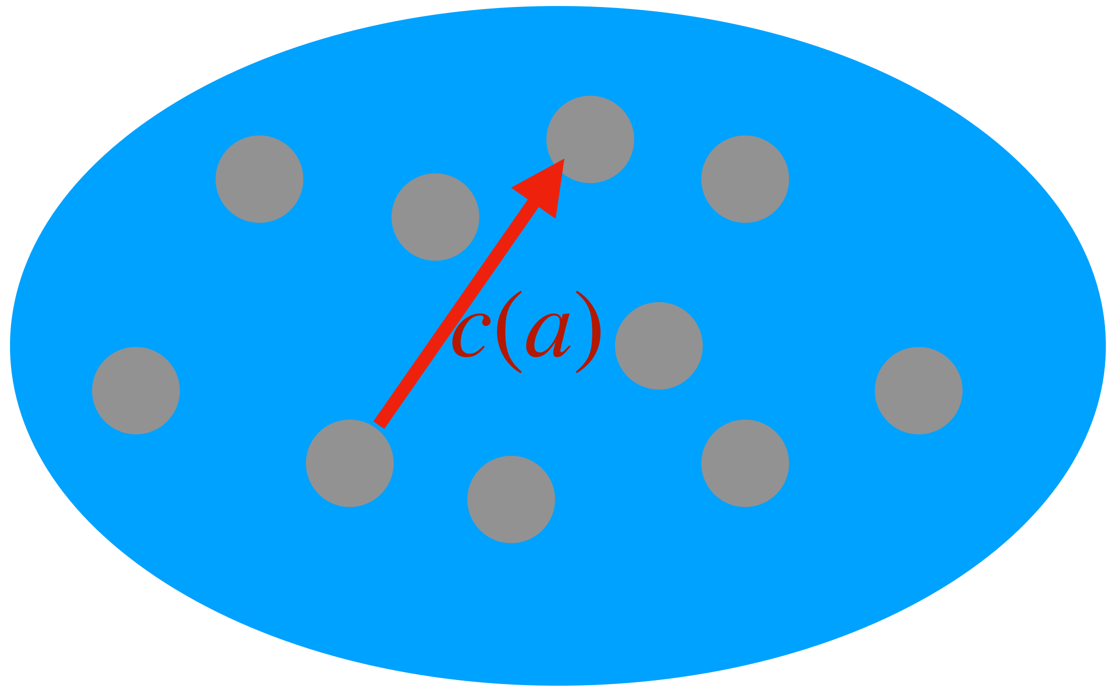
Additional data:

$$b : V \to \mathbb{R}$$

$$\mathrm{ex}_x(v) = b(v) \quad \forall v \in V$$

b-Transshipment

Additional data:

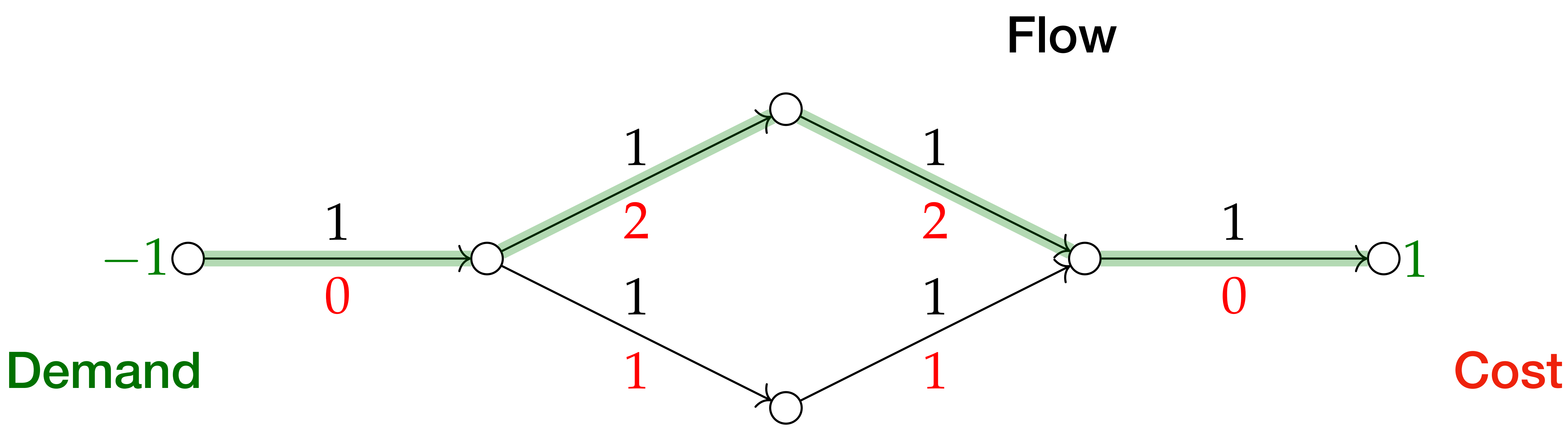


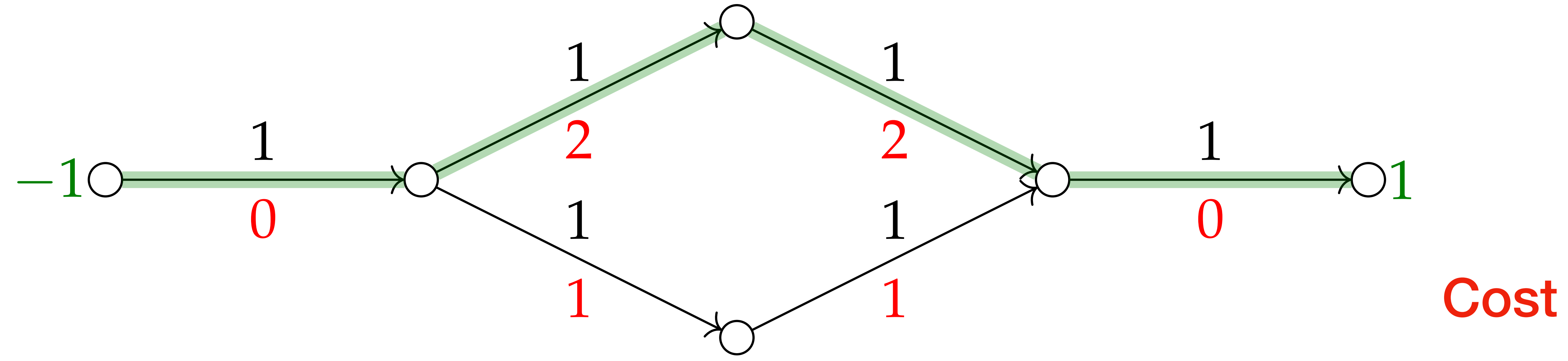
$$b : V \to \mathbb{R}$$

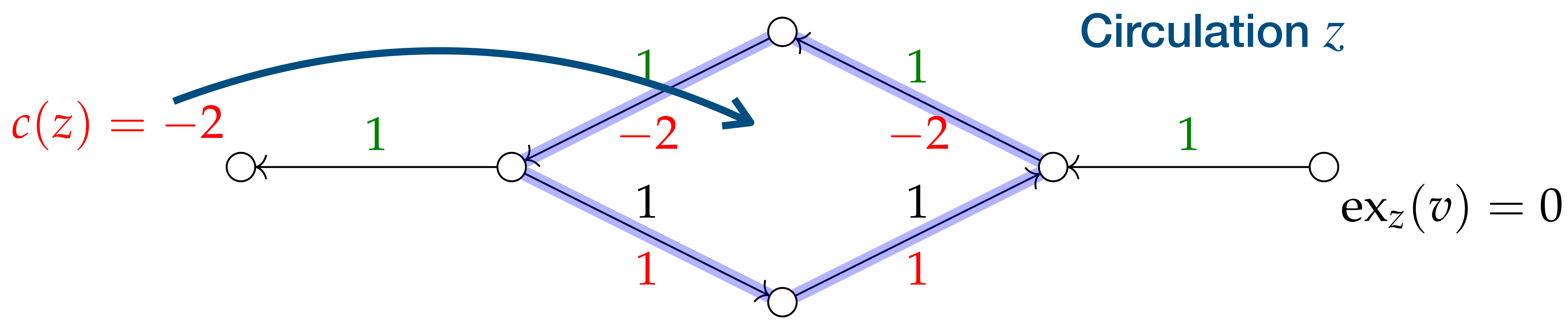
$$\mathrm{ex}_x(v) = b(v) \quad \forall v \in V$$

Cost of a flow x :

$$c(x) = \sum_{a \in A} c(a)x(a)$$


Flow
Demand
Cost
−1
1
1
0
1
2
1
2
1
1
1
1
1
0


−1
1
Cost

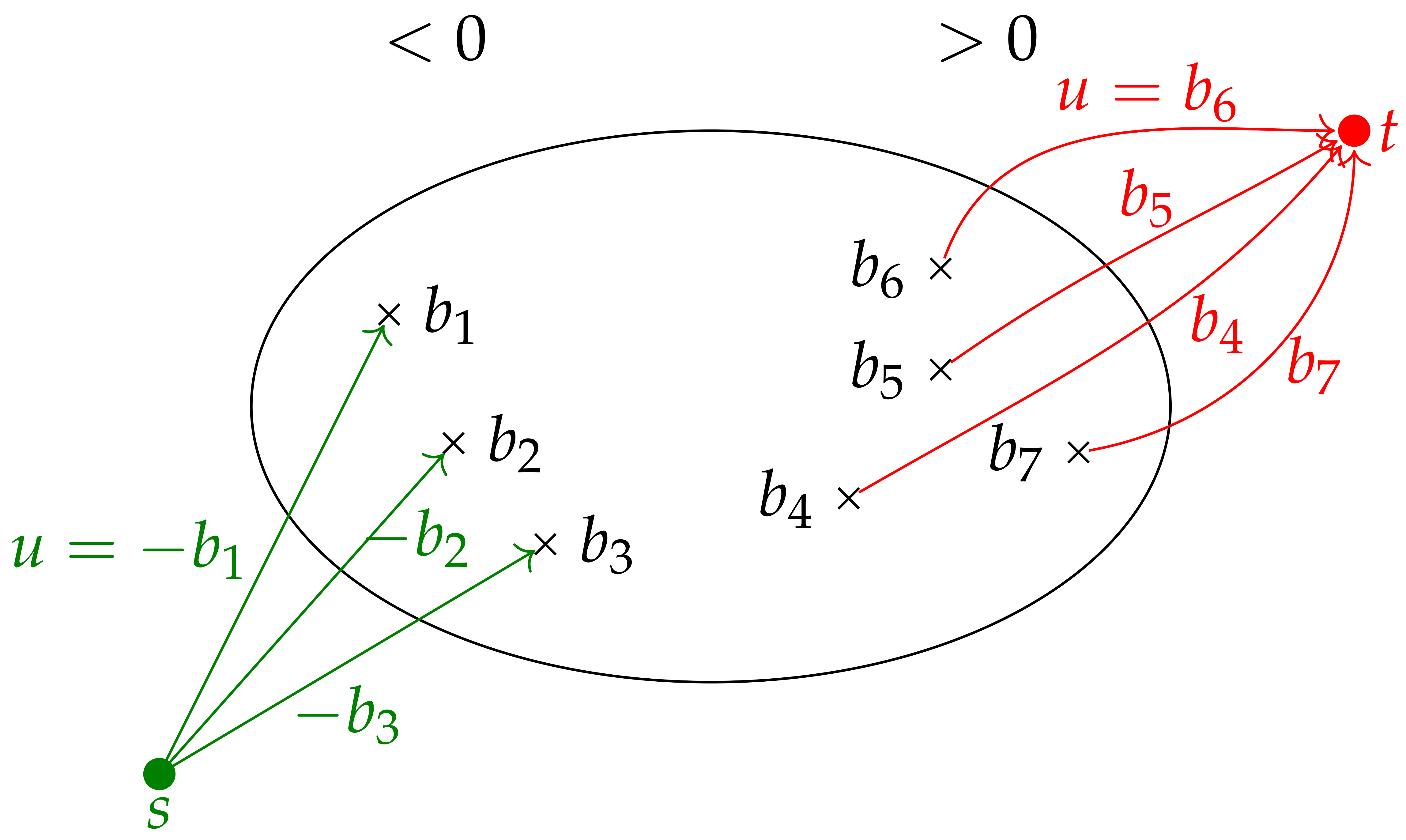

$c(z) = -2$
Circulation z
$\mathrm{ex}_z(v) = 0$

Additional: Residual cost

For any residual arc $a^{(-1)}$ (of an original arc a) we define its cost by:

$$c(a^{-1}) := -c(a).$$

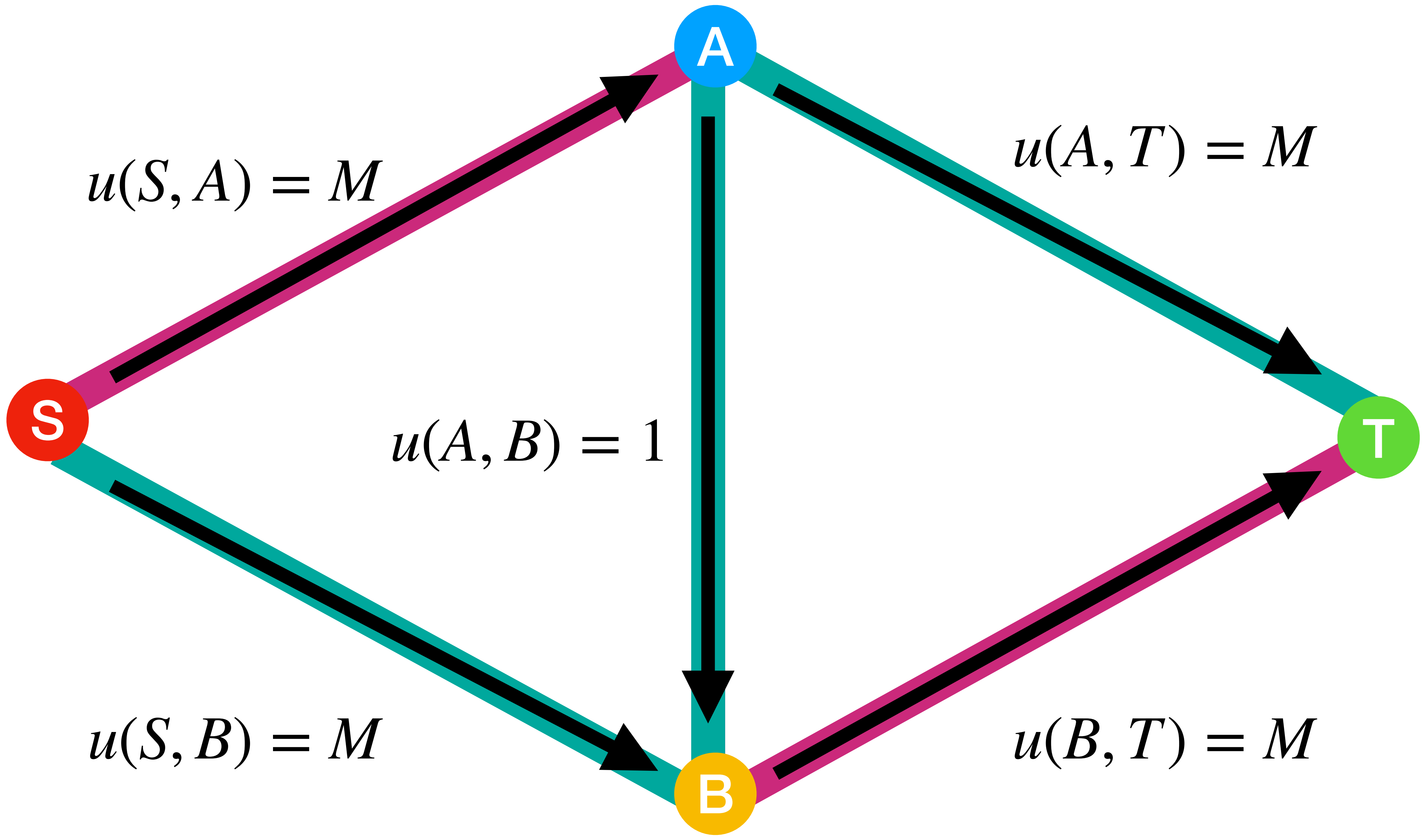
Initial solution



Negative Canceling Algorithm

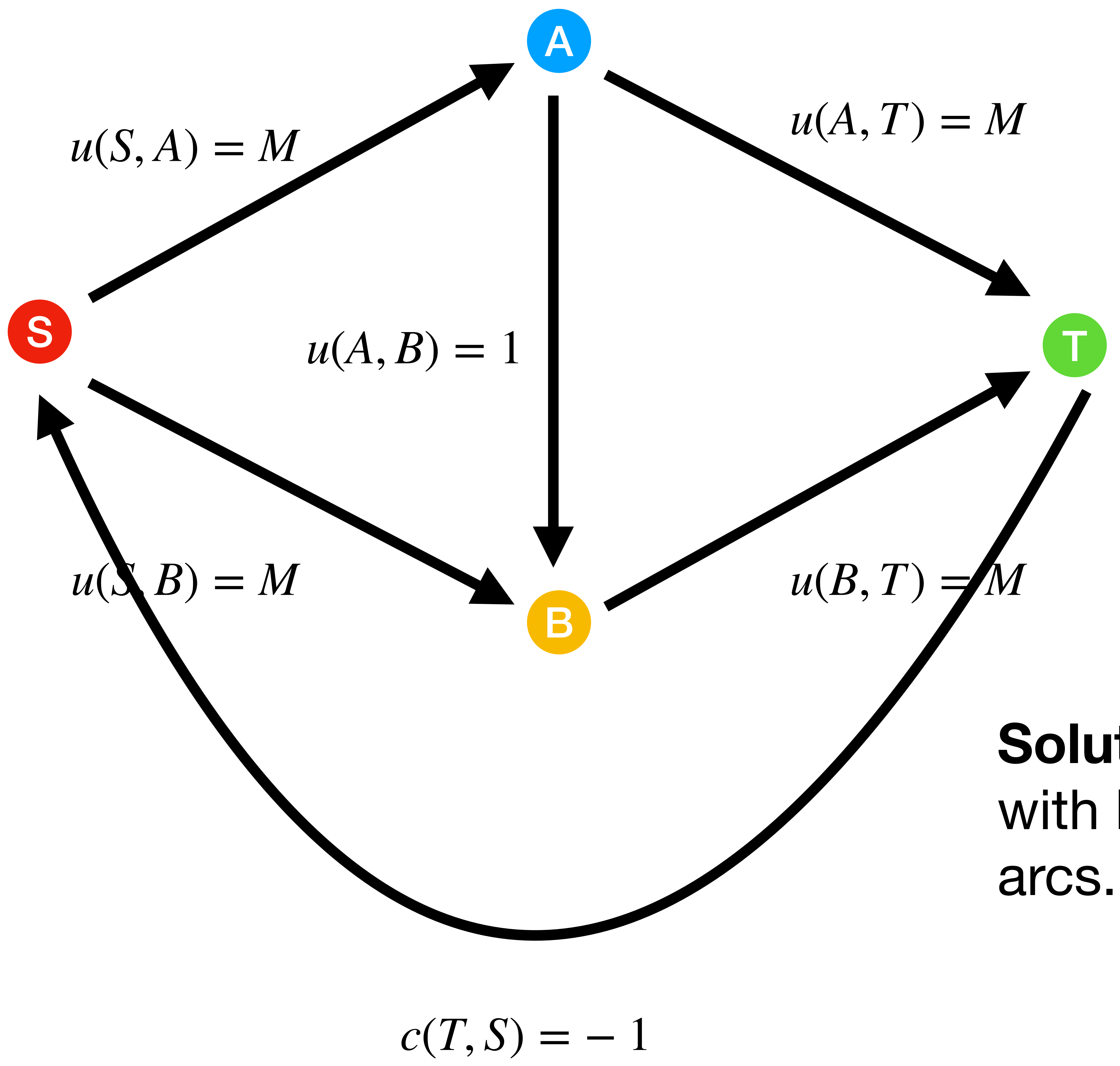
- Initialize: Calculate initial feasible b-transshipment x via MaxFlow.
- Search for negative directed circle C in residual graph G_x .
    - IF none exists: RETURN .
    - ELSE augment along circle C and iterate.

Problem with both algorithms:



Solution: Choose a path with least number of arcs. (Edmonds-Karp Algorithm)

Problem with both algorithms:



Solution: Choose a circle with least mean cost of arcs.