

Aplicație

Flux maxim \rightarrow cuplaj maxim
în grafuri bipartite

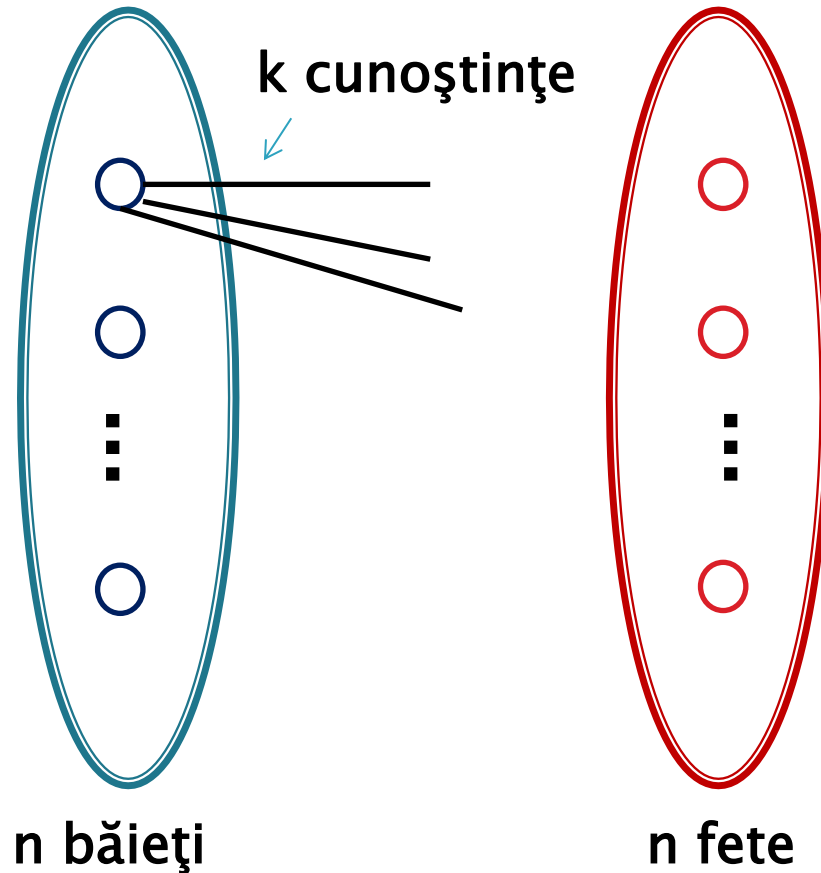
Cuplaje

Repere istorice. Aplicații

- ▶ **Problema seratei (perechilor) – sec XIX**
 - n băieți, n fete
 - Un băiat cunoaște exact k fete
 - O fată cunoaște exact k băieți

Repere istorice. Aplicații

► Problema seratei (perechilor)



Repere istorice. Aplicații

► Problema seratei (perechilor) – sec XIX

- ❖ Se poate organiza o repriză de dans astfel încât fiecare participant să danseze cu o cunoștință a sa?

Repere istorice. Aplicații

► Problema seratei (perechilor) – sec XIX

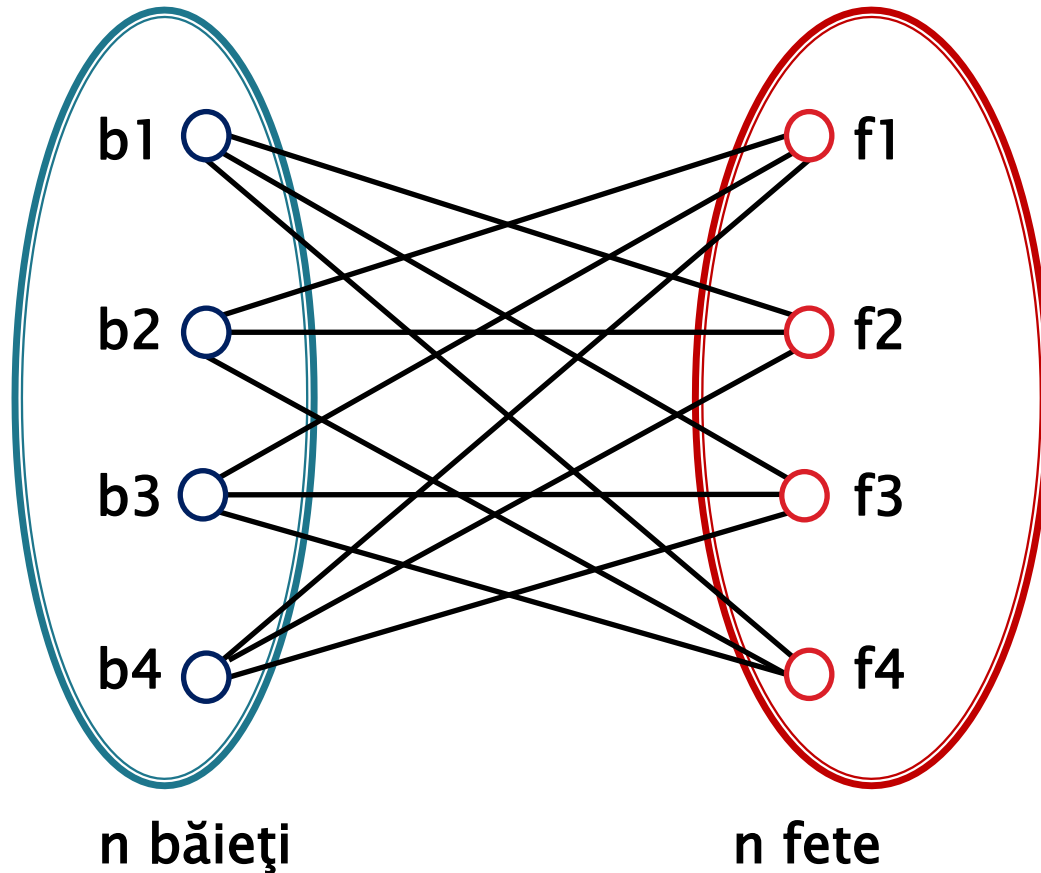
- ❖ Se poate organiza o repriză de dans astfel încât fiecare participant să danseze cu o cunoștință a sa?
- ❖ Se pot organiza **k** reprize de dans în care fiecare participant să danseze câte un dans cu fiecare cunoștință a sa? – **vom reveni la cursuri dedicate cuplajelor**

Repere istorice. Aplicații

► Problema seratei (perechilor) – sec XIX

$n=4$

$k=3$



Repere istorice. Aplicații

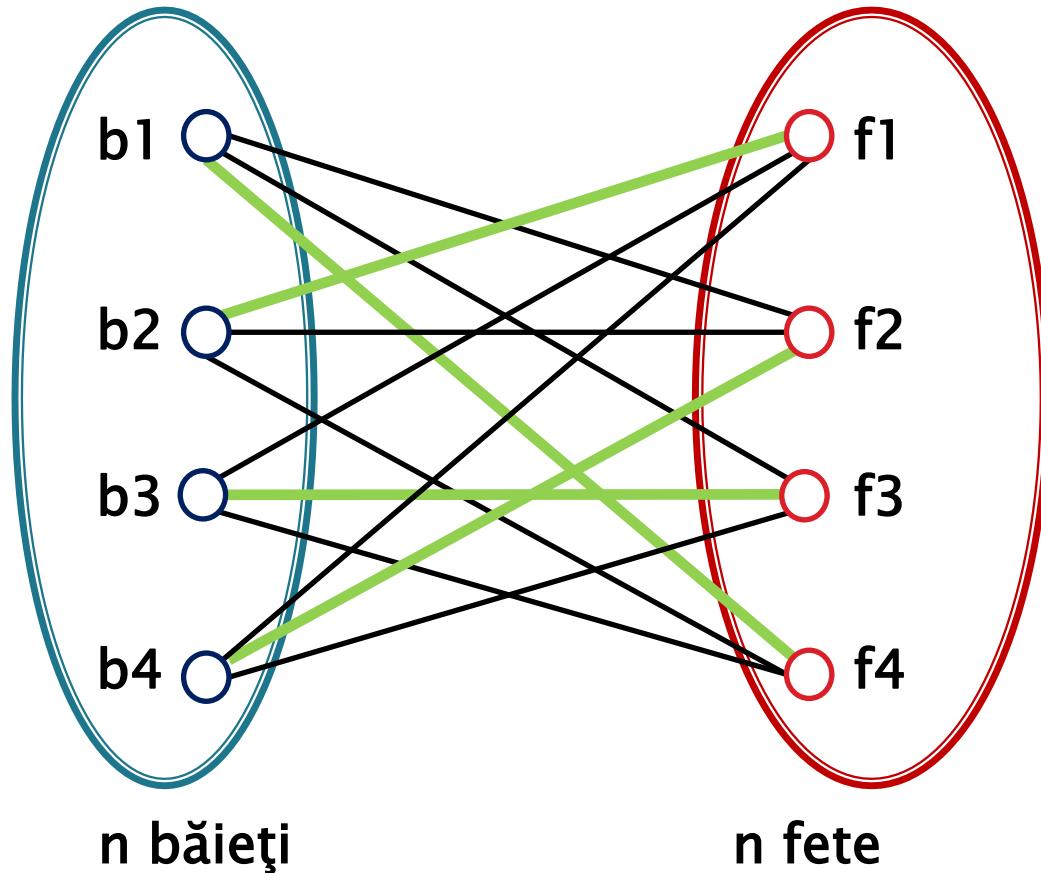
► O repriză de dans

b1,f4

b2,f1

b3,f3

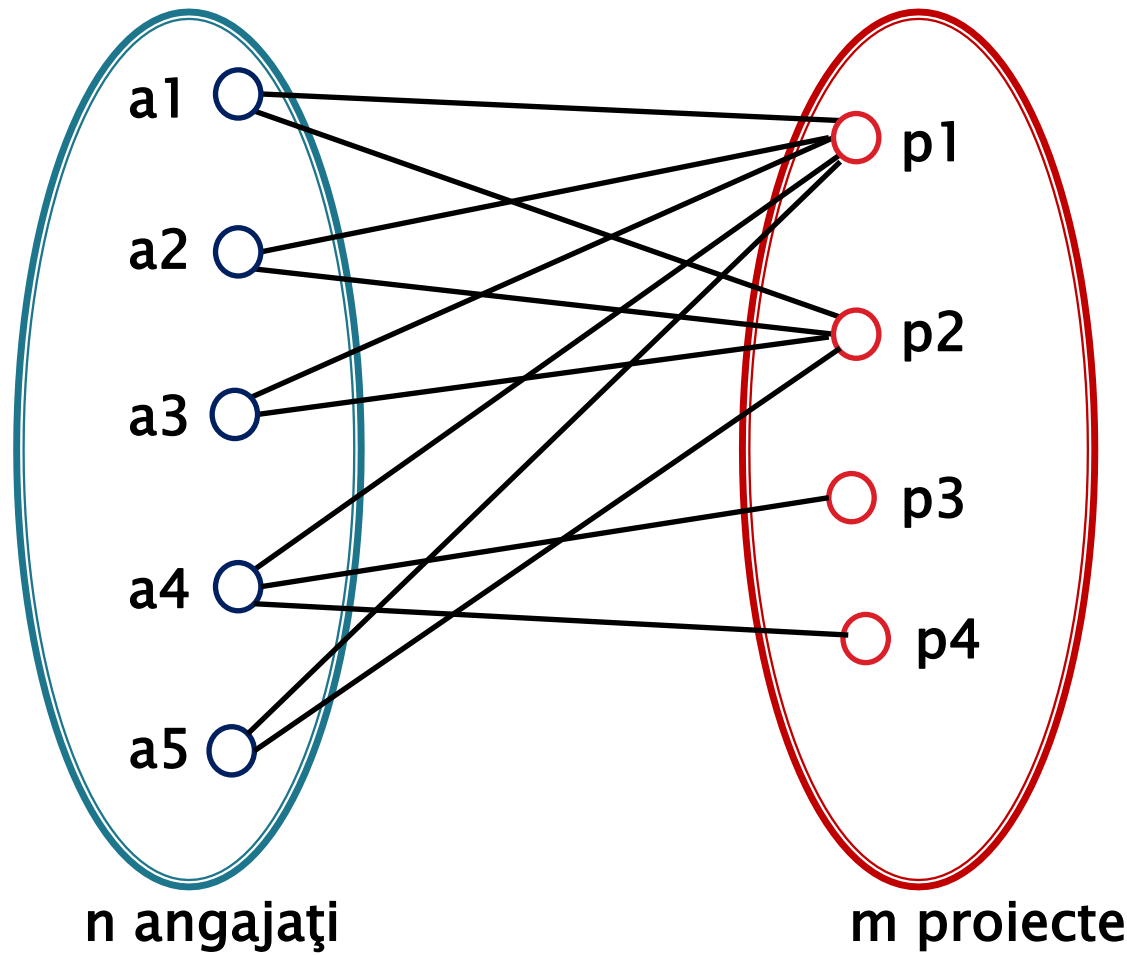
b4,f2



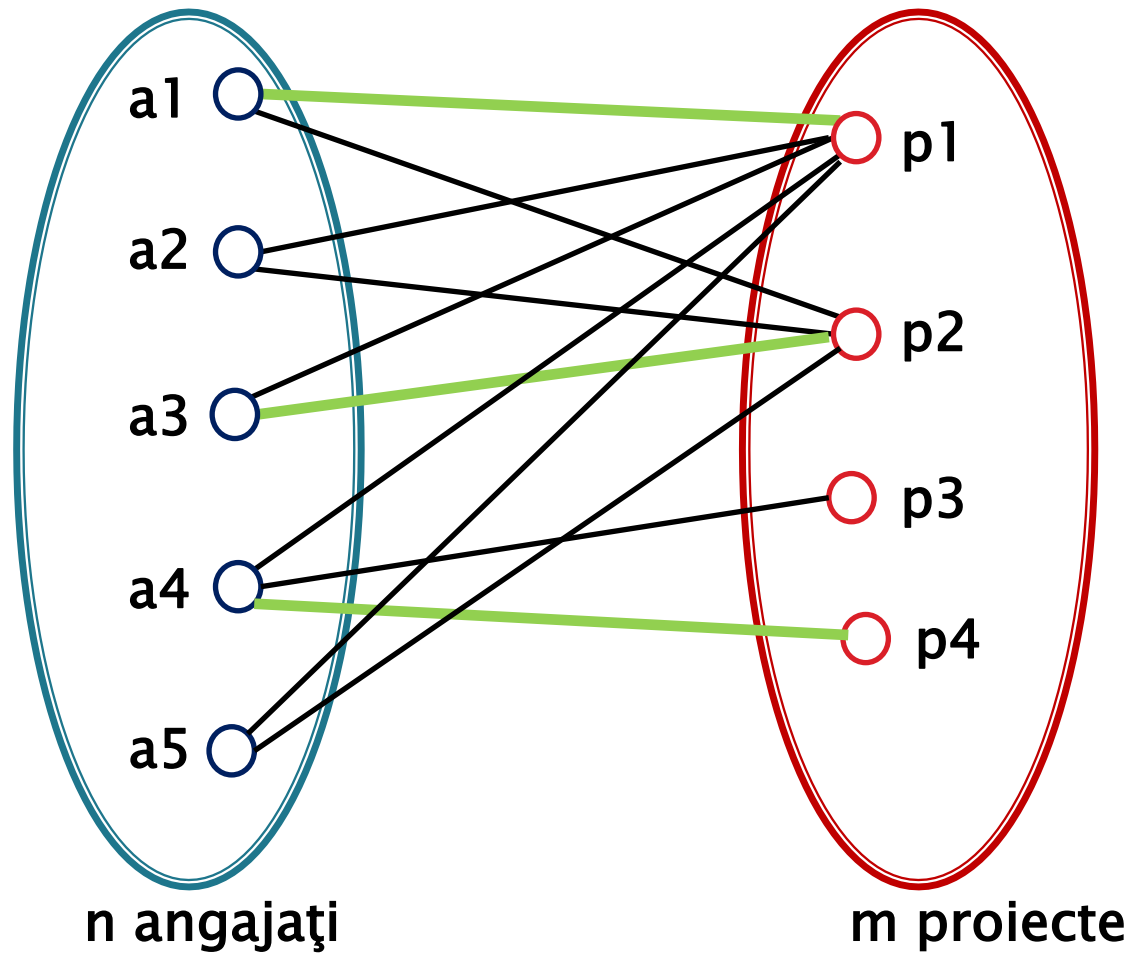
Repere istorice. Aplicații

- ▶ **Organizare de competiții**
- ▶ **Probleme de repartiție**
 - lucrători – locuri de muncă
 - profesori – examene /conferințe
 - **Problema orarului**

Alte aplicații

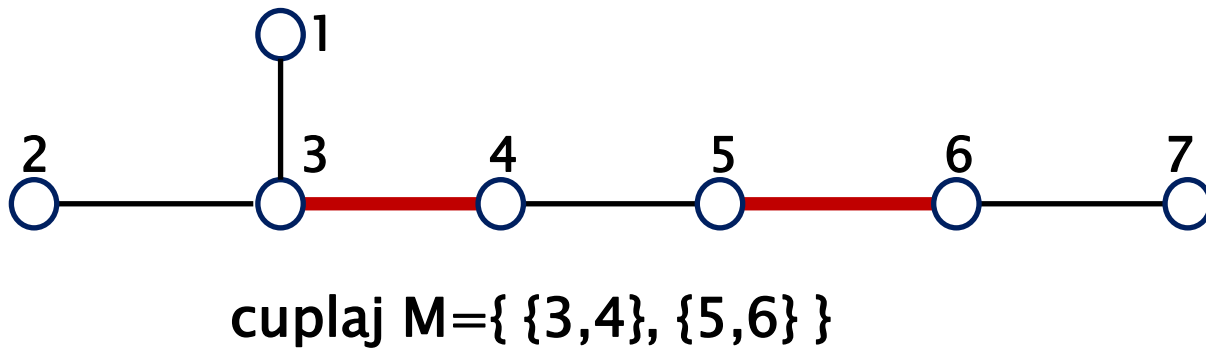


Alte aplicații



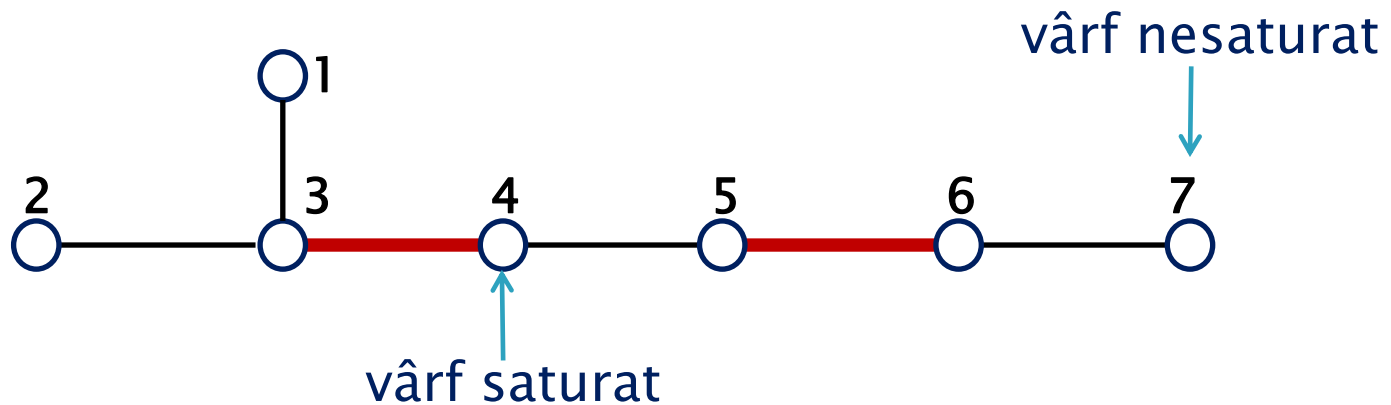
Fie $G = (V, E)$ un graf și $M \subseteq E$.

- ▶ M s.n **cuplaj** dacă orice două muchii din M sunt neadiacente



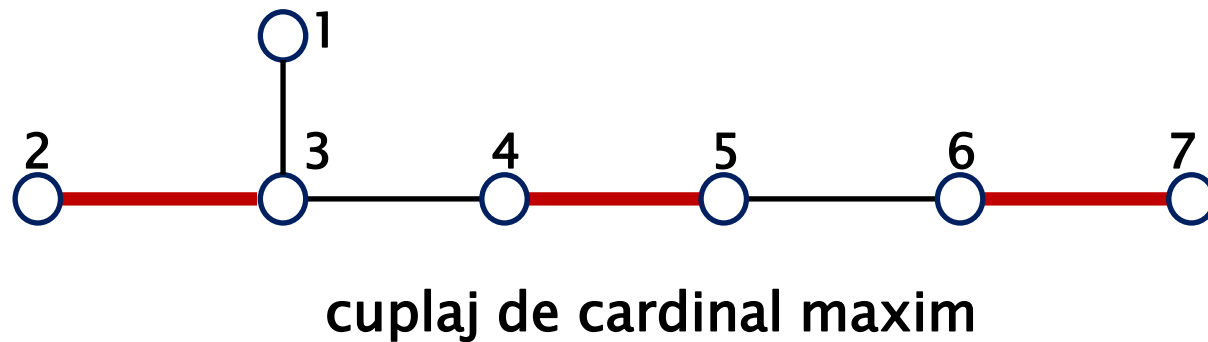
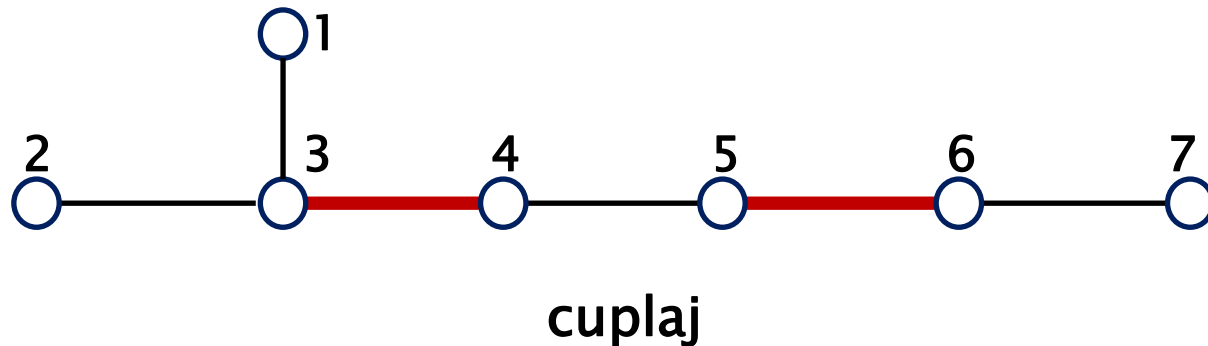
Fie $G = (V, E)$ un graf și $M \subseteq E$.

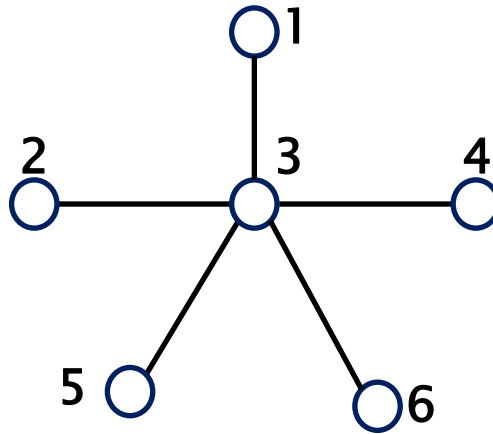
- ▶ M s.n **cuplaj** dacă orice două muchii din M sunt neadiacente
- ▶ $V(M)$ = mulțimea vârfurilor **M -saturate**
- ▶ $V(G) - V(M)$ = mulțimea vârfurilor **M -nesaturate**



- ▶ Un cuplaj M^* s.n **cuplaj de cardinal maxim** (**cuplaj maxim**):

$$|M^*| \geq |M|, \forall M \subseteq E \text{ cuplaj}$$





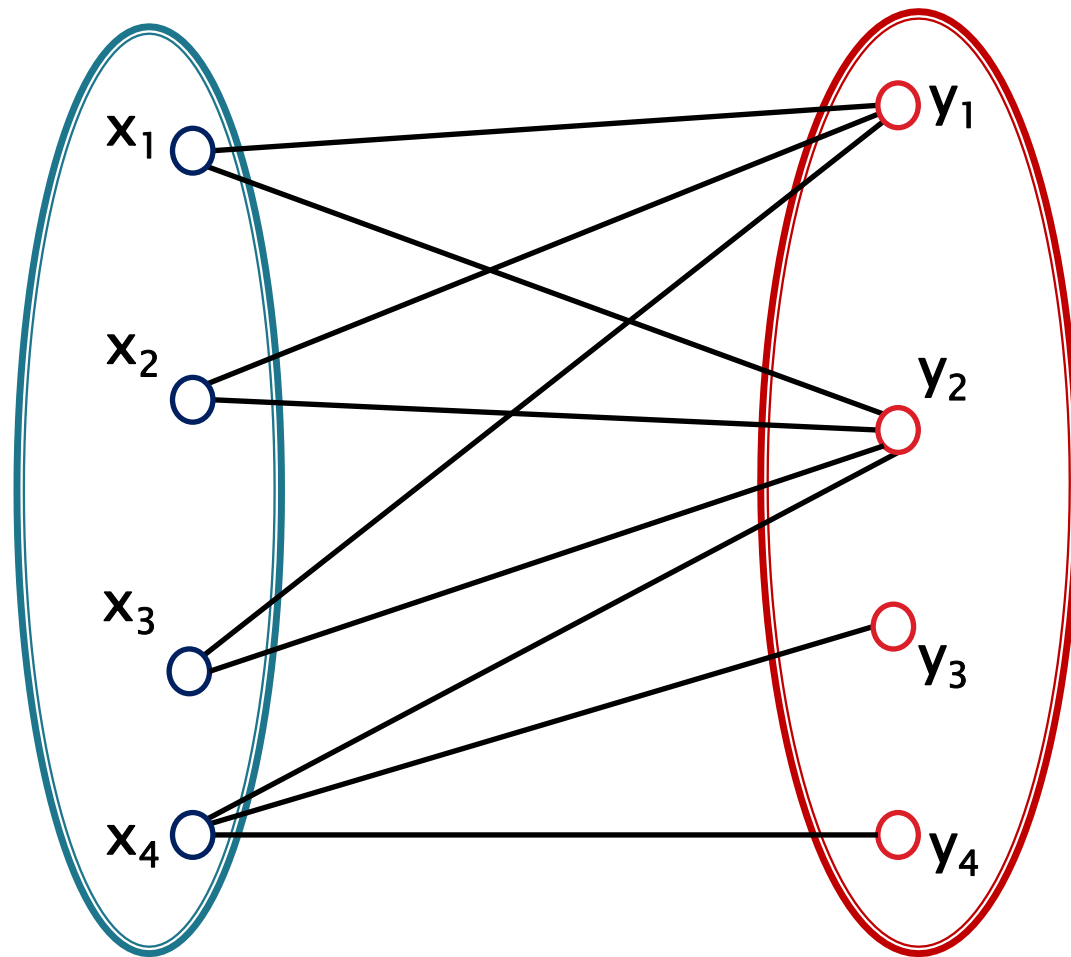
cuplaj de cardinal maxim ?

- ▶ **Algoritm de determinare a unui cuplaj maxim într-un graf bipartit**

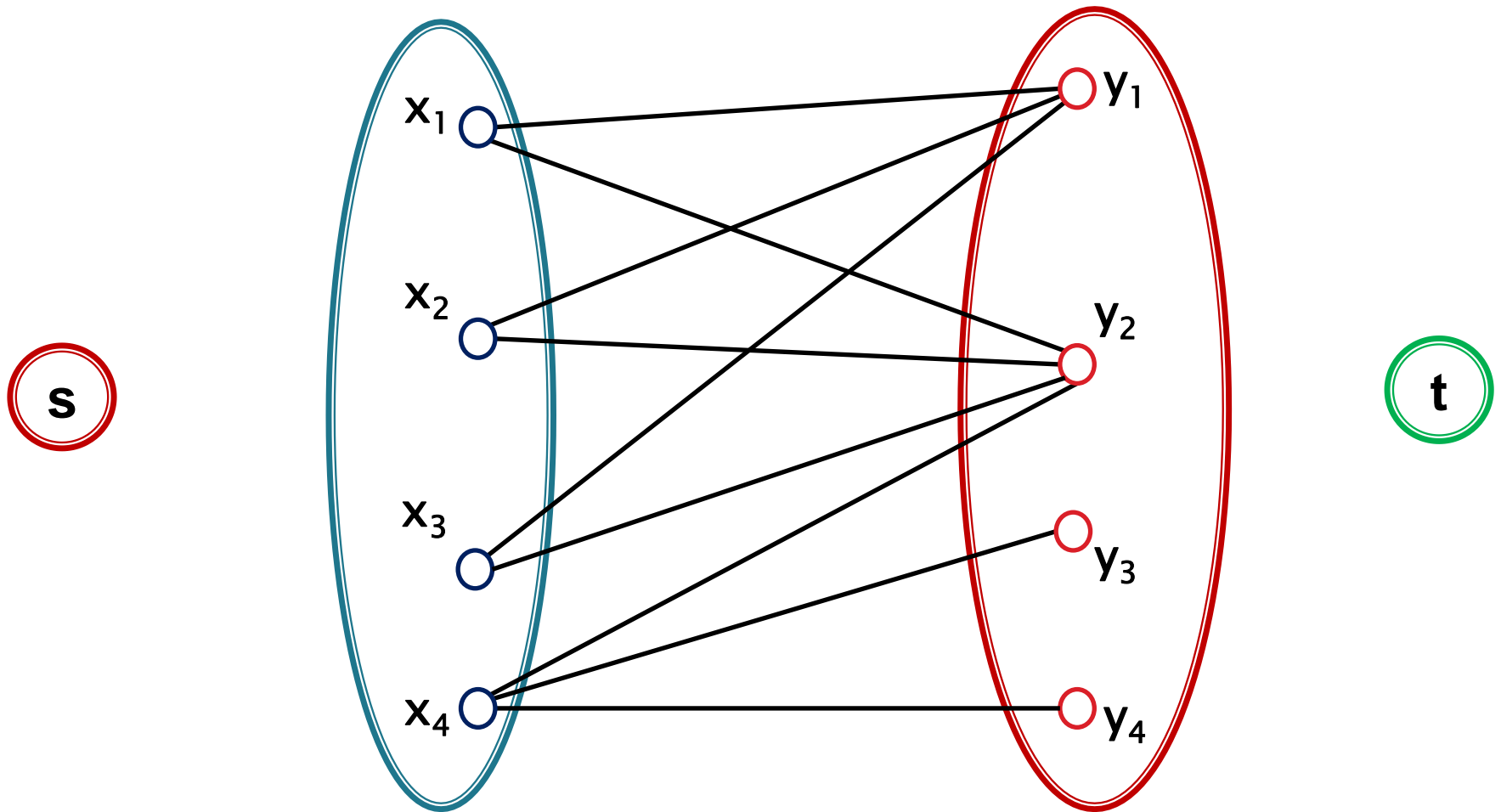
▶ Algoritm de determinare a unui cuplaj maxim într-un graf bipartit

- Reducem problema determinării unui cuplaj maxim într-un cuplaj bipartit G la determinarea unui flux maxim într-o rețea de transport asociată lui G

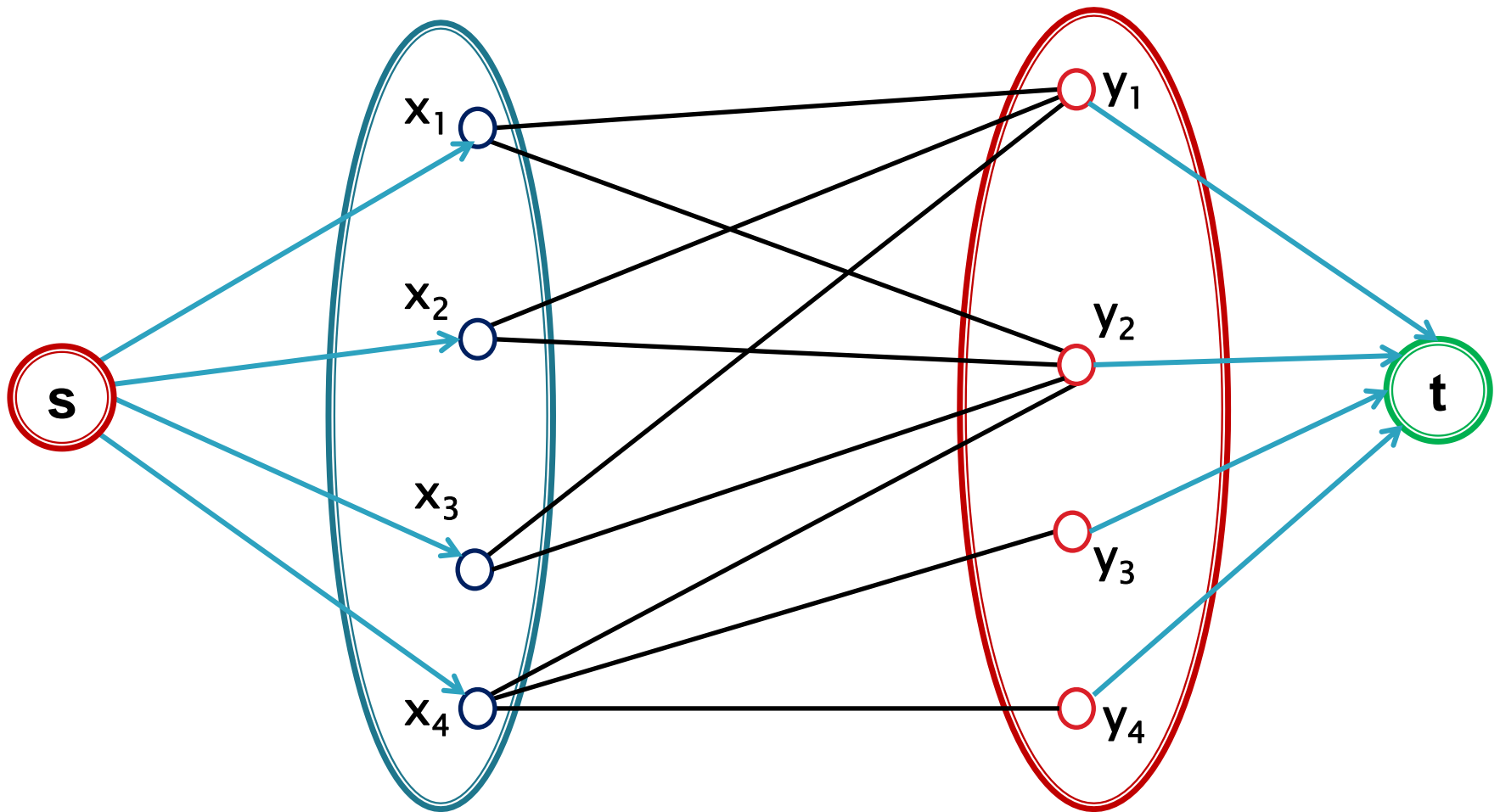
- ▶ **Algoritm de determinare a unui cuplaj maxim într-un graf bipartit**
 - Construim rețeaua de transport N asociată lui G :



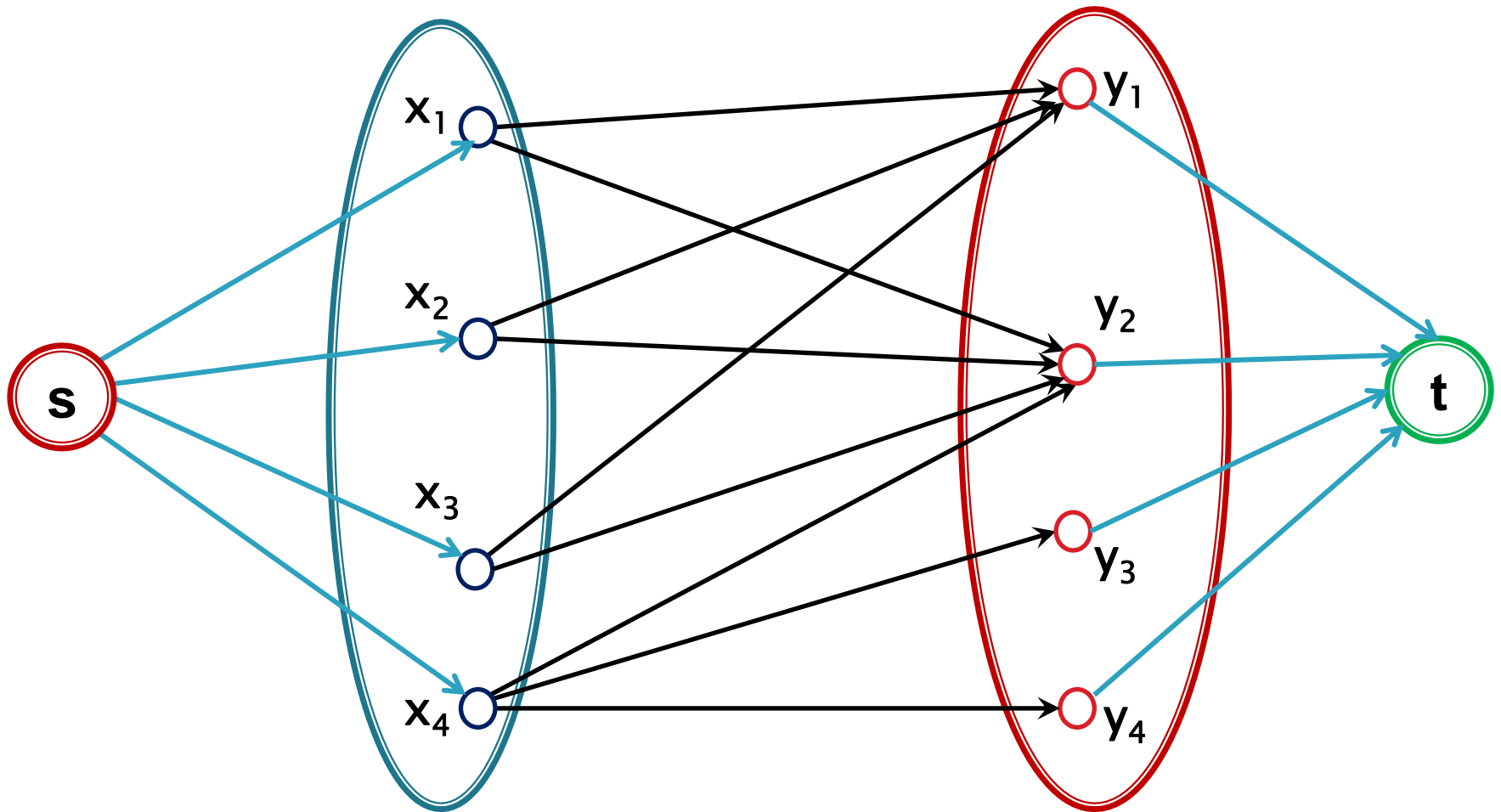
- ▶ Adăugăm două noduri noi s și t



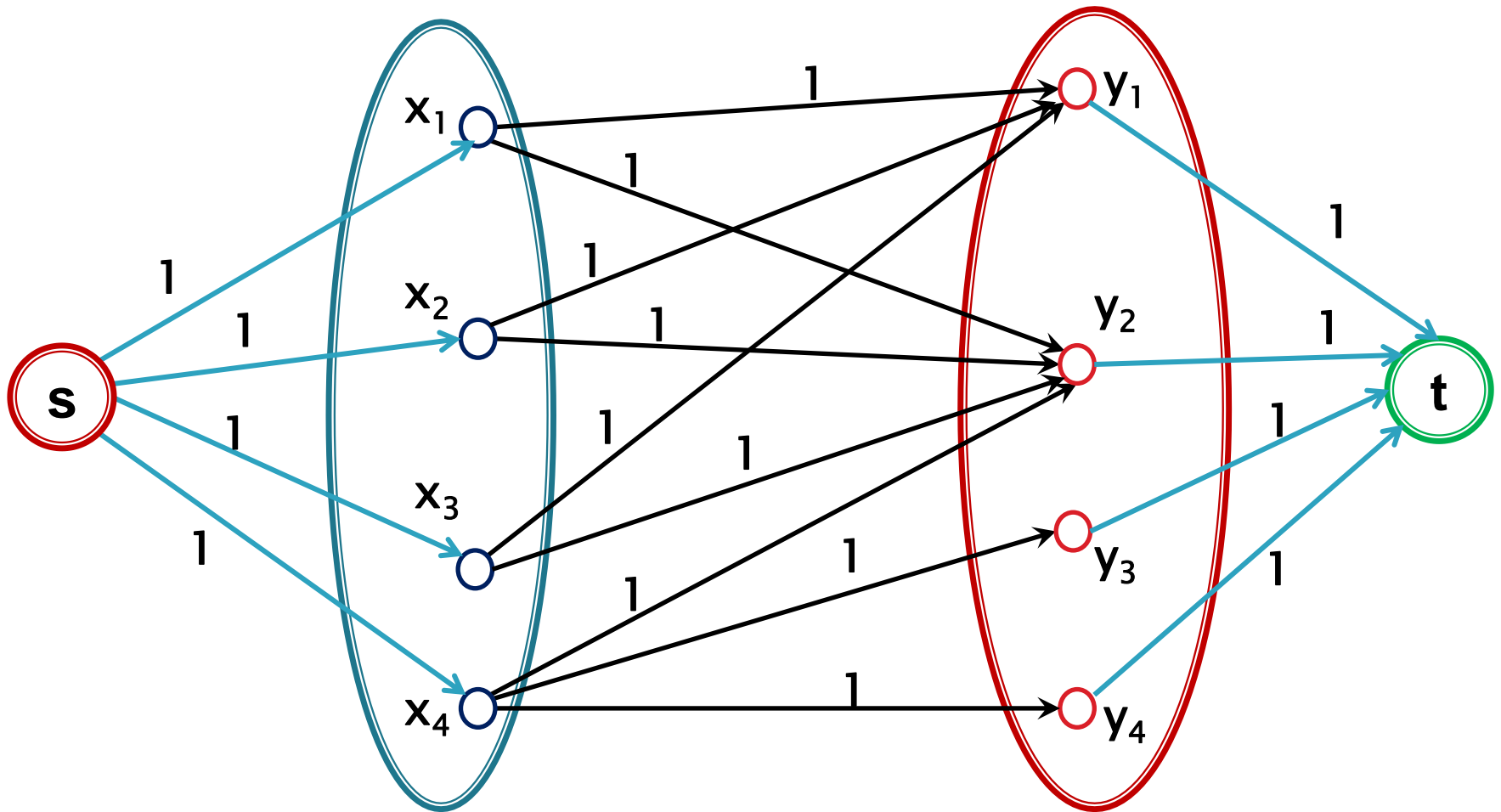
- ▶ Adăugăm arce (s, x_i) , pentru $x_i \in X$ și (y_j, t) , $y_j \in Y$



- Transformăm muchiile $x_i y_j$ în arce (de la X la Y)



- Asociem fiecărui arc capacitatea 1

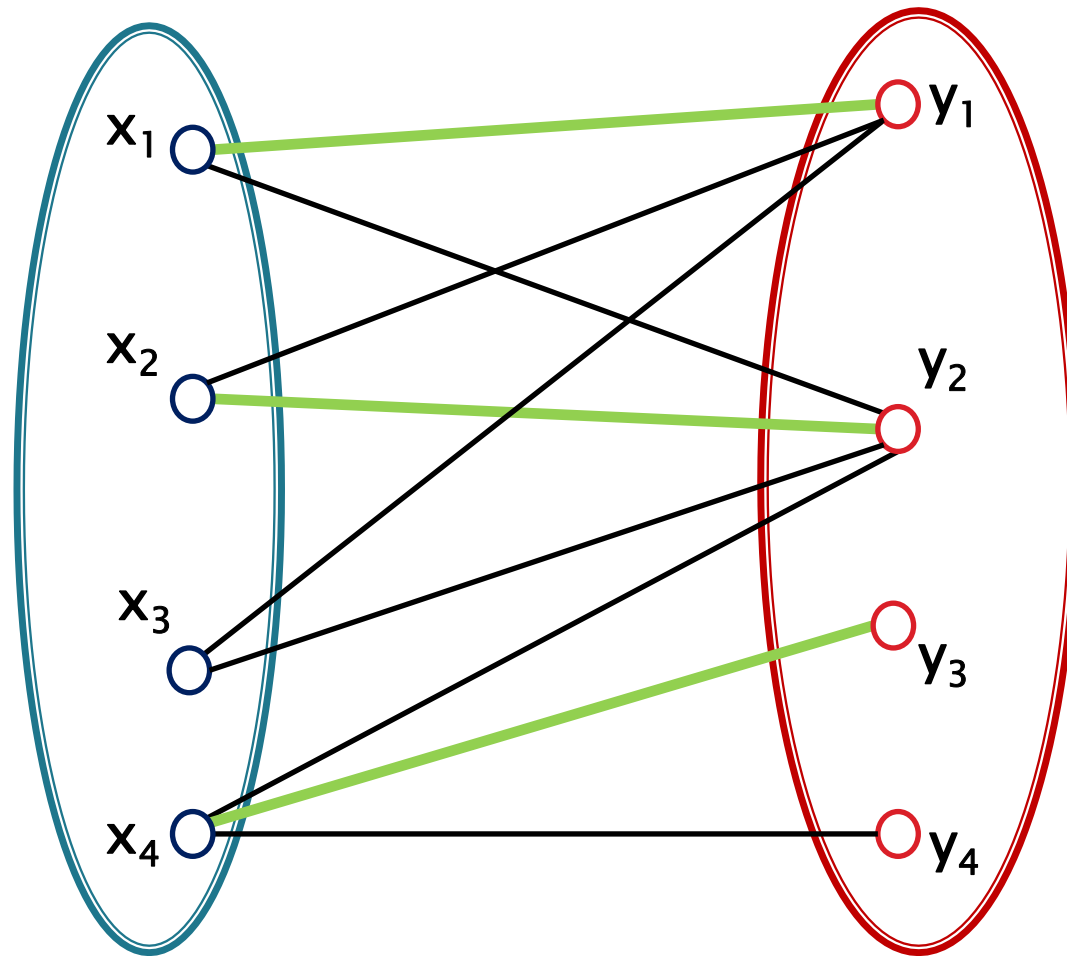


► Proprietatea 1

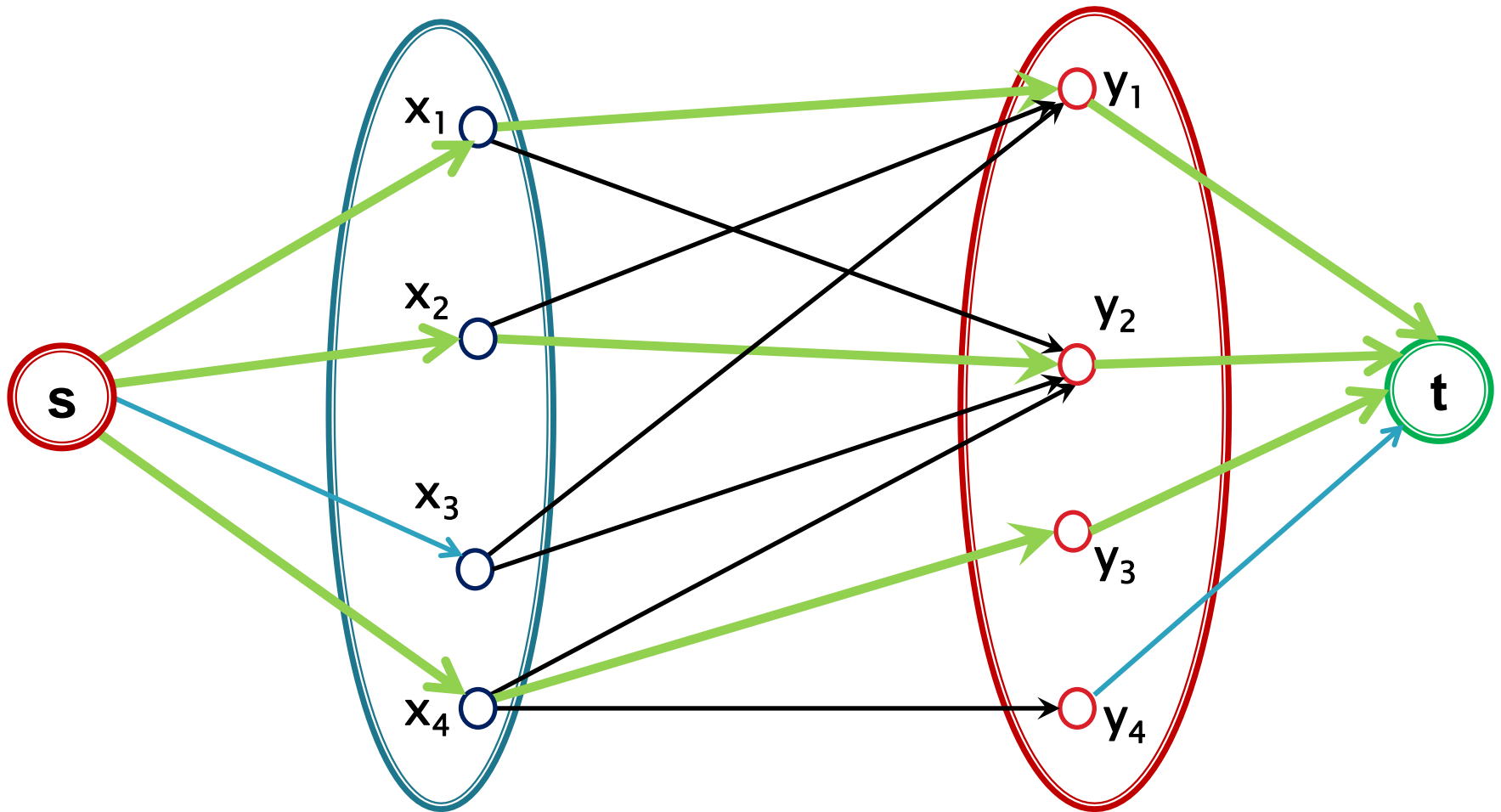
Fie $G=(X\cup Y, E)$ un graf bipartit și M un cuplaj în G .
Atunci există un flux f în rețeaua de transport asociată N cu

$$\text{val}(f) = |M|$$

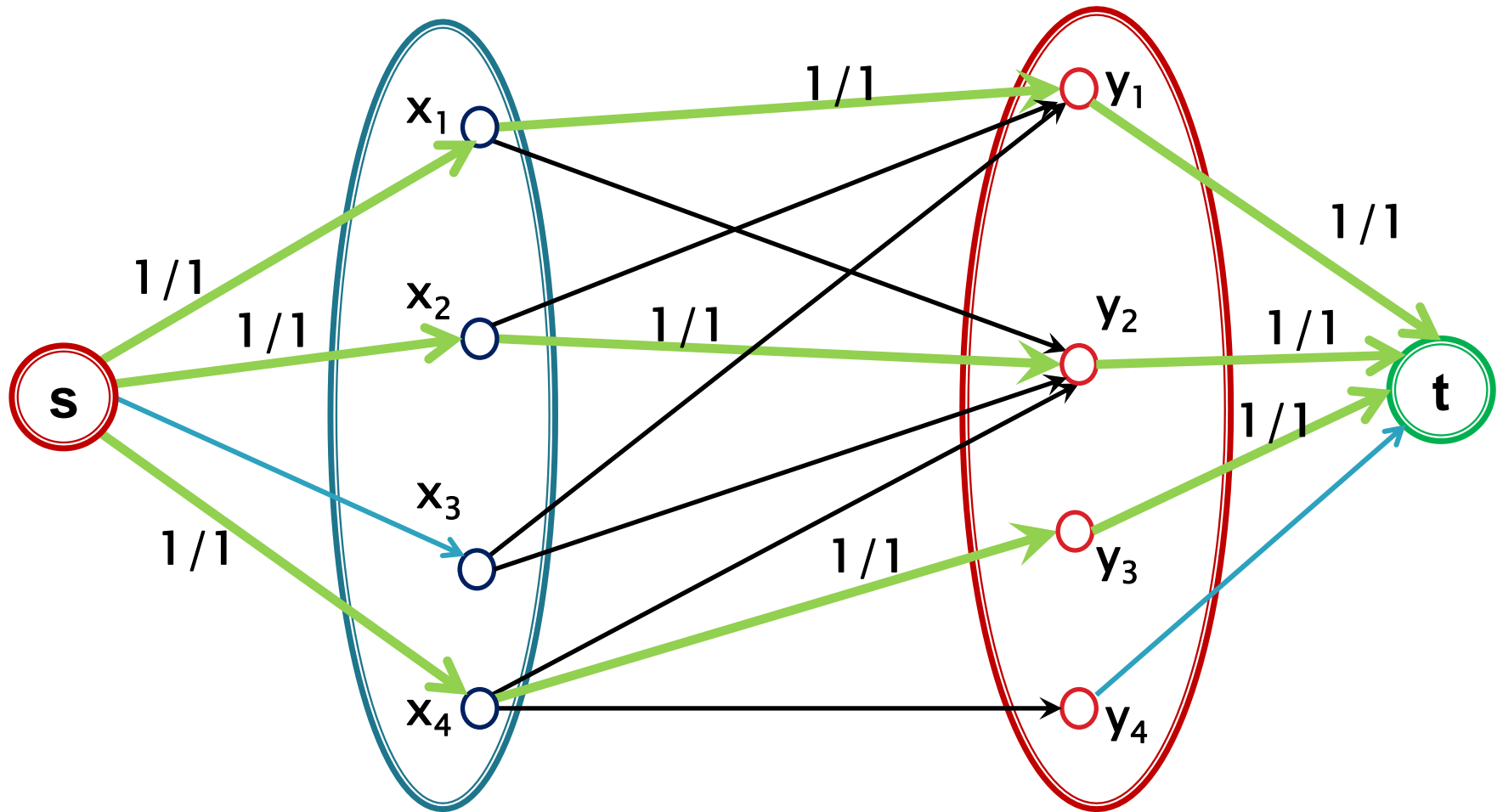
► Cuplaj în graf \Rightarrow flux în rețea



► Cuplaj în graf \Rightarrow flux în rețea

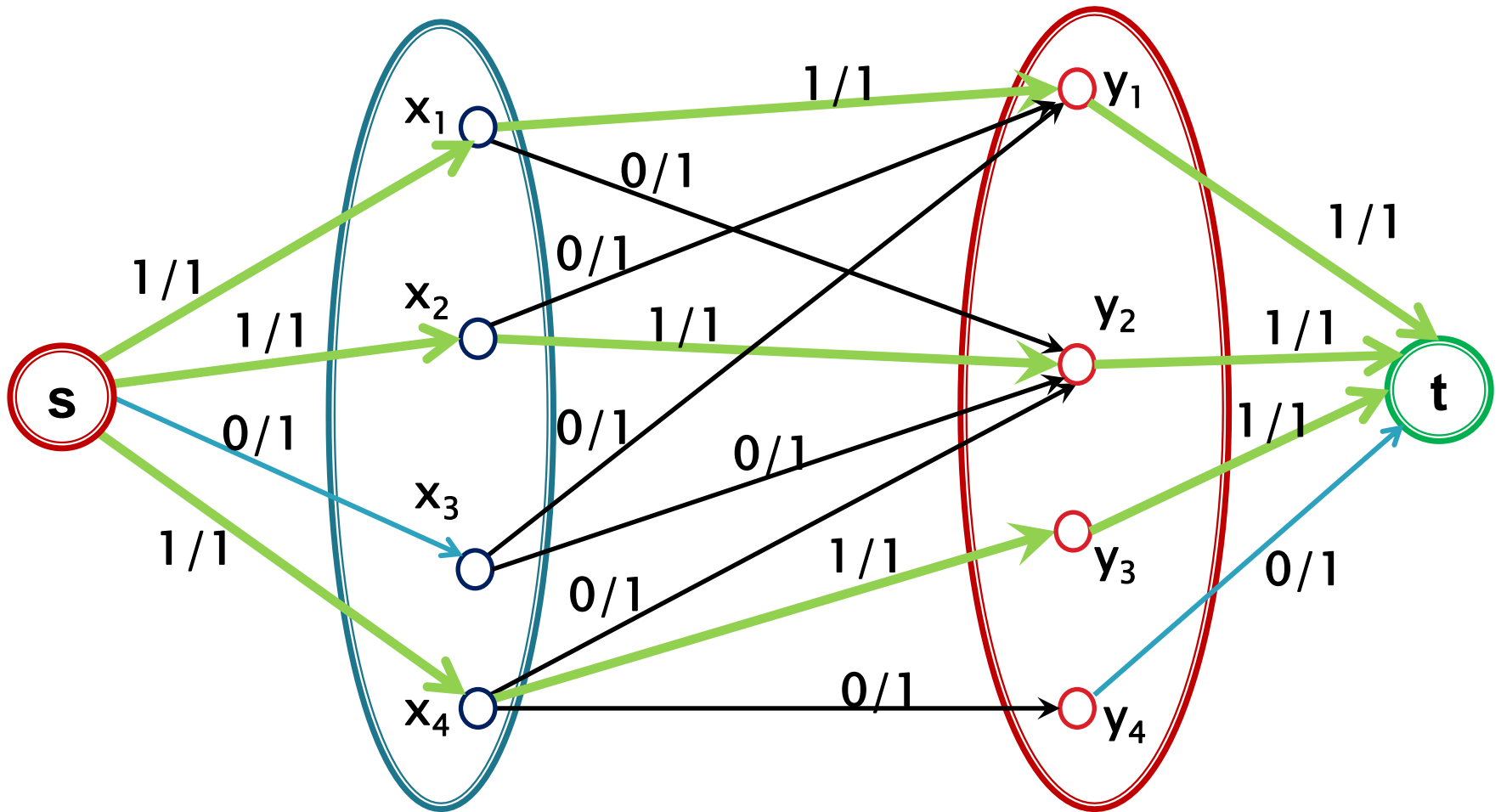


► Cuplaj în graf \Rightarrow flux în rețea



Celelalte arce au flux 0 și capacitate 1

► Cuplaj în graf \Rightarrow flux în rețea

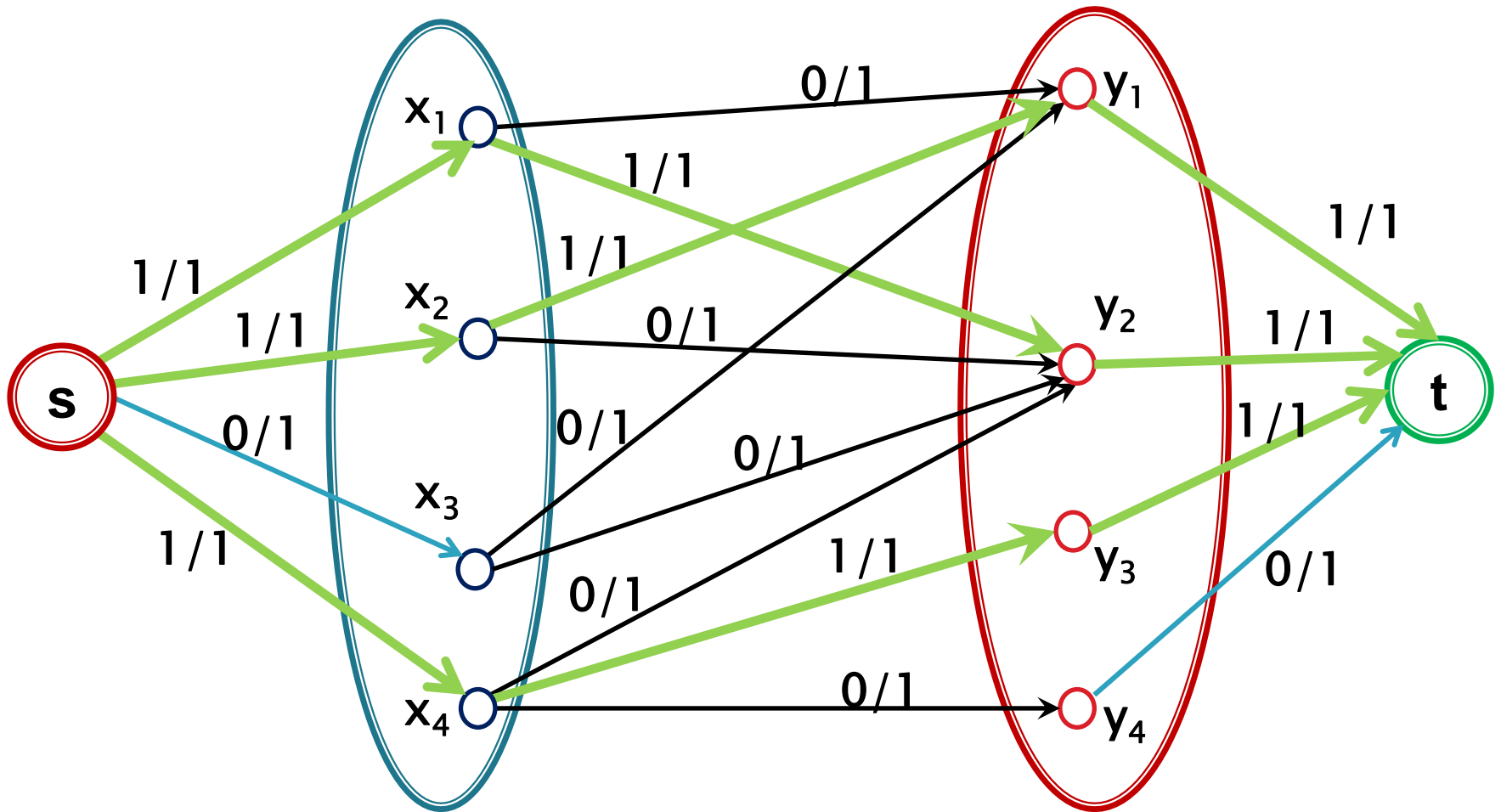


► Proprietatea 2

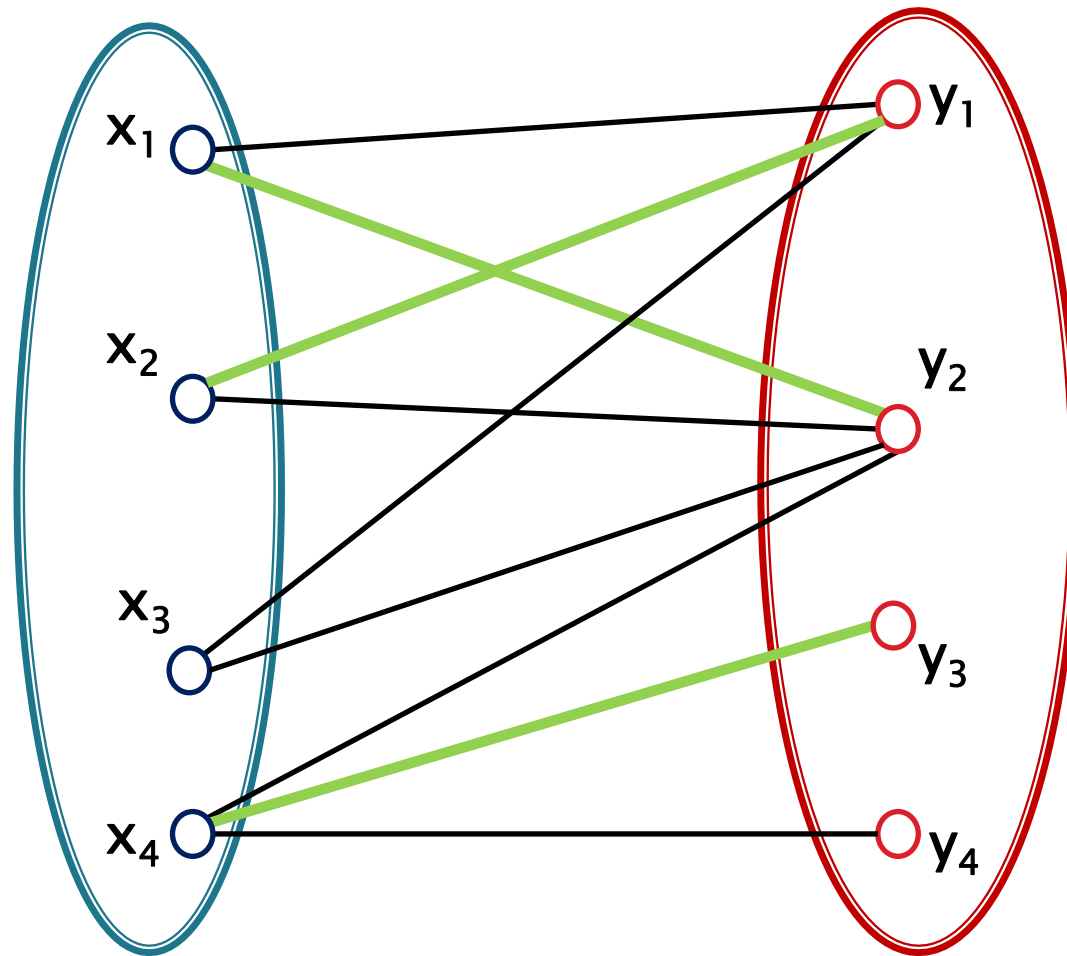
Fie $G=(X\cup Y, E)$ un graf bipartit și f un flux în rețeaua de transport N asociată. Atunci există M un cuplaj în G cu

$$\text{val}(f) = |M|$$

► Flux în rețea \Rightarrow cuplaj în graf

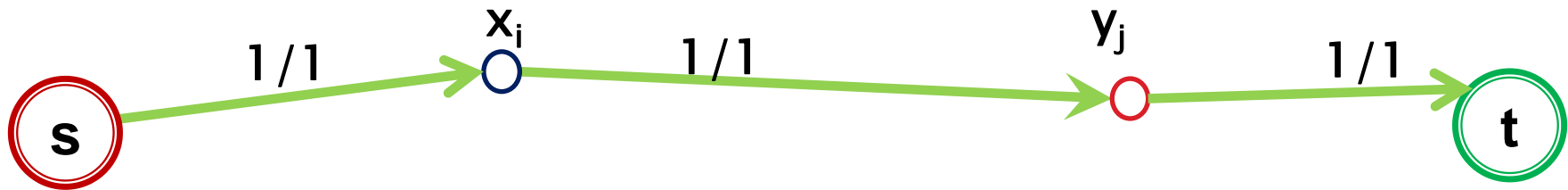


► Flux în rețea \Rightarrow cuplaj în graf

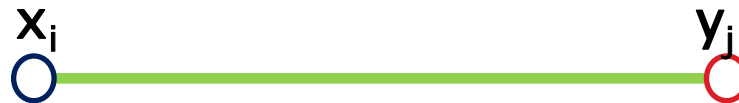


► Concluzie: Flux în rețea \Leftrightarrow cuplaj în graf

Drum cu o unitate de flux



Muchie în cuplaj



► Consecință

f^* flux maxim în $N \Rightarrow$ cuplajul corespunzător M^* este cuplaj maxim în G

A determina un **cuplaj maxim** într-un graf bipartit \Leftrightarrow
a determina un **flux maxim** în rețeaua asociată

Algoritm de determinare a unui cuplaj maxim în $G=(X\cup Y,E)$:

Algoritm de determinare a unui cuplaj maxim în $G=(X\cup Y,E)$:

1. Construim N rețeaua de transport asociată
2. Determinăm f^* flux maxim în N

Algoritm de determinare a unui cuplaj maxim în $G=(X\cup Y,E)$:

1. Construim N rețeaua de transport asociată
2. Determinăm f^* flux maxim în N
3. Considerăm $M = \{xy \mid f^*(xy)=1, x\in X, y\in Y, xy\in N\}$

(pentru fiecare arc cu flux nenul xy din N care nu este incident în s sau t , muchia xy corespunzătoare din G se adaugă la M)

4. return M

Algoritm de determinare a unui cuplaj maxim în $G=(X\cup Y,E)$:

1. Construim N rețeaua de transport asociată
2. Determinăm f^* flux maxim în N
3. Considerăm $M = \{xy \mid f^*(xy)=1, x\in X, y\in Y, xy\in N\}$

(pentru fiecare arc cu flux nenul xy din N care nu este incident în s sau t , muchia xy corespunzătoare din G se adaugă la M)

4. return M

Complexitate?

Algoritm de determinare a unui cuplaj maxim în $G=(X \cup Y, E)$:

1. Construim N rețeaua de transport asociată
2. Determinăm f^* flux maxim în N
3. Considerăm $M = \{xy \mid f^*(xy)=1, x \in X, y \in Y, xy \in N\}$

(pentru fiecare arc cu flux nenul xy din N care nu este incident în s sau t , muchia xy corespunzătoare din G se adaugă la M)

4. return M

Complexitate: $L = c^+(s) \leq n \Rightarrow O(mn)$

Aplicație

Construcția unui graf orientat
din secvențele de grade

► Se dau secvențele

- $s_0^+ = \{d_1^+, \dots, d_n^+\}$
- $s_0^- = \{d_1^-, \dots, d_n^-\}$

Să se construiască, **dacă se poate**, un graf orientat G
cu $s^+(G) = s_0^+$ și $s^-(G) = s_0^-$

► Se dau secvențele

- $s_0^+ = \{d_1^+, \dots, d_n^+\}$
- $s_0^- = \{d_1^-, \dots, d_n^-\}$

Să se construiască, dacă se poate, un graf orientat G cu $s^+(G) = s_0^+$ și $s^-(G) = s_0^-$

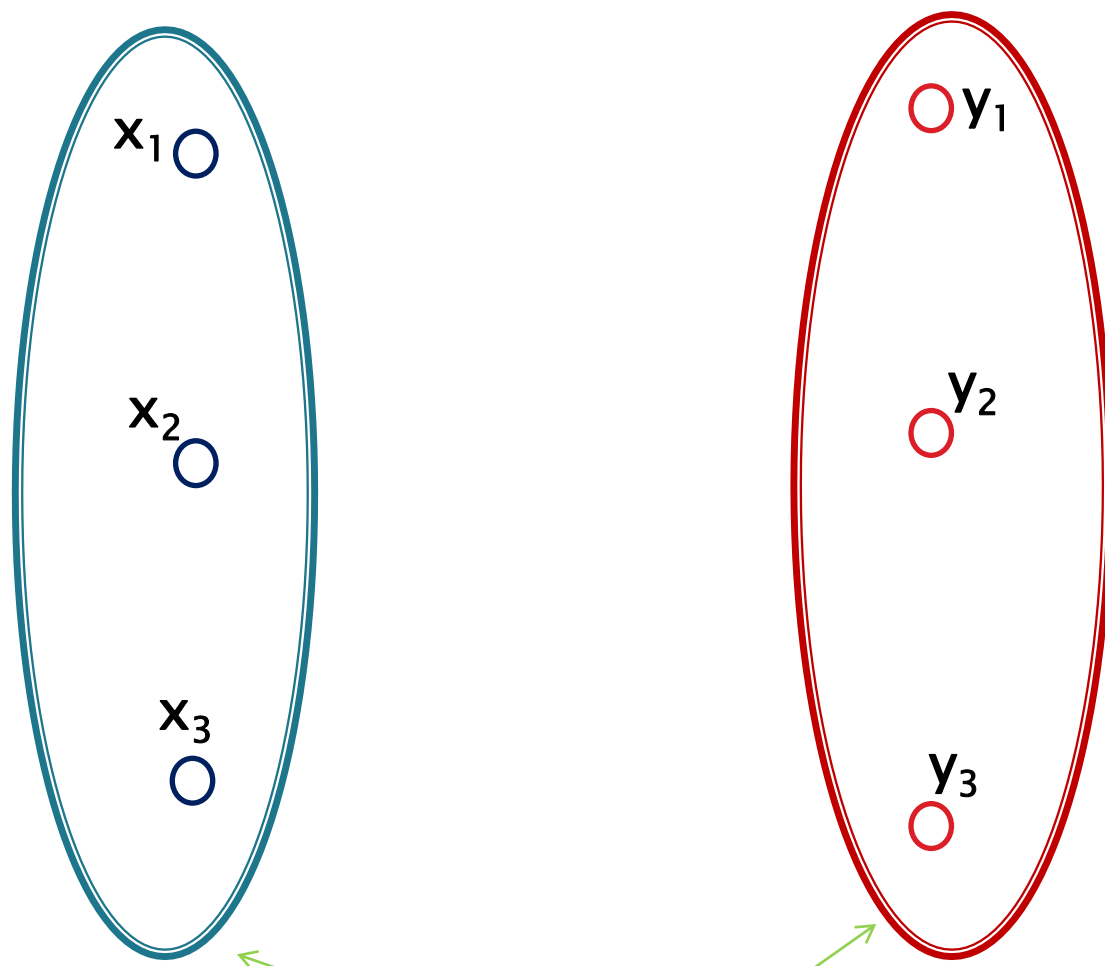
► Exemplu

- $s_0^+ = \{1, 0, 2\}$
- $s_0^- = \{1, 1, 1\}$

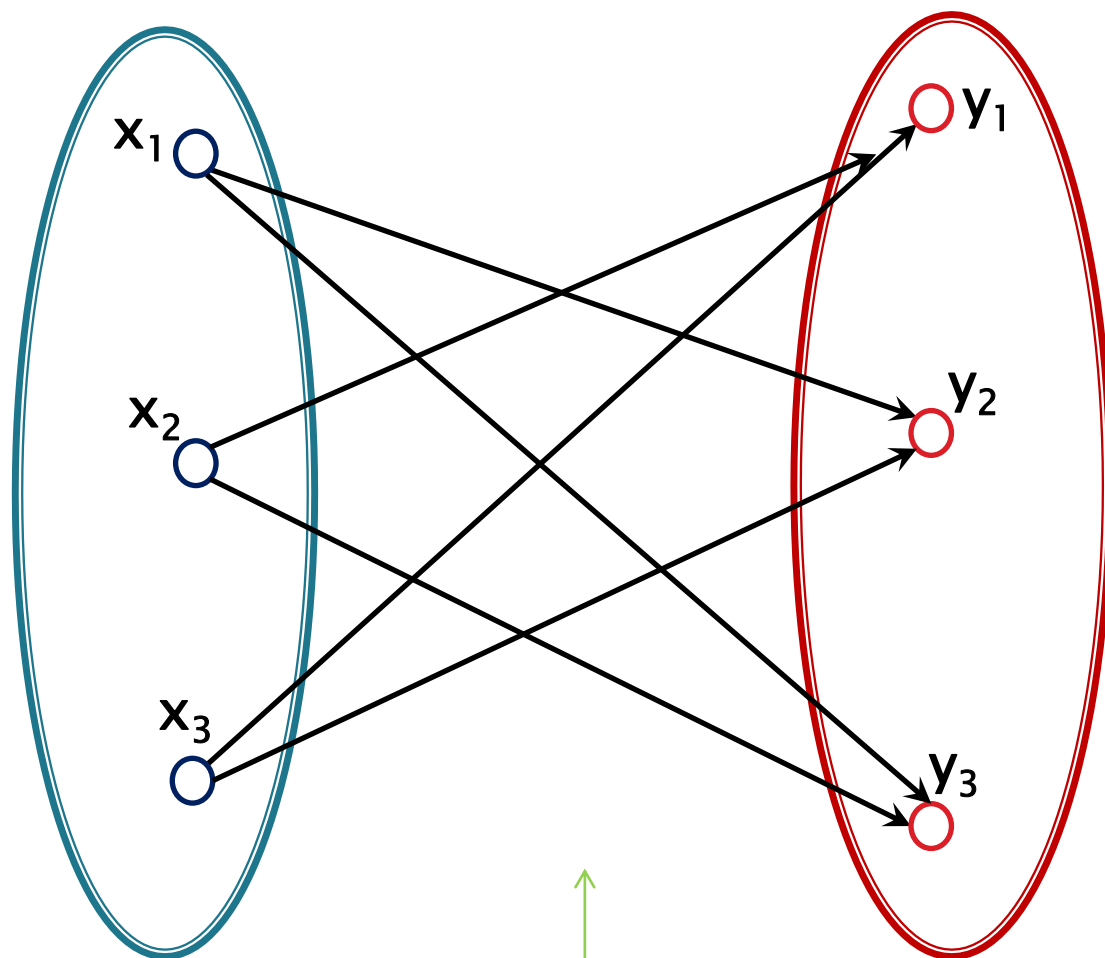
▶ Exemplu

- $s_0^+ = \{1, 0, 2\}$
- $s_0^- = \{1, 1, 1\}$

- ▶ **Construim o rețea asociată celor două secvențe a.î.**
din fluxul maxim în rețea să putem **deduce** dacă G se
poate construi + arcele grafului G (în caz afirmativ)

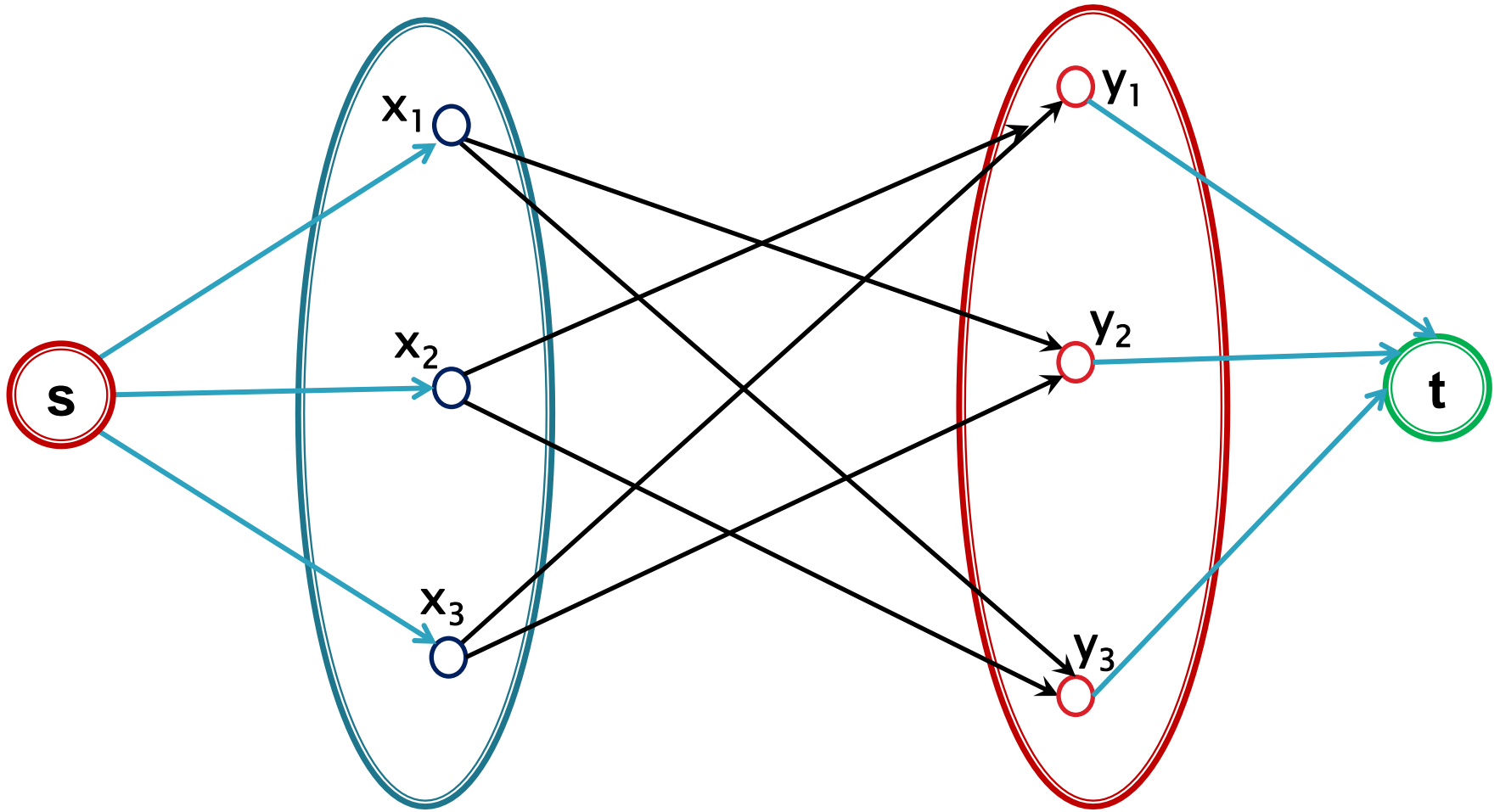


Vârfurile 1, 2,..., n se pun în ambele clase ale bipartiției (câte o copie)

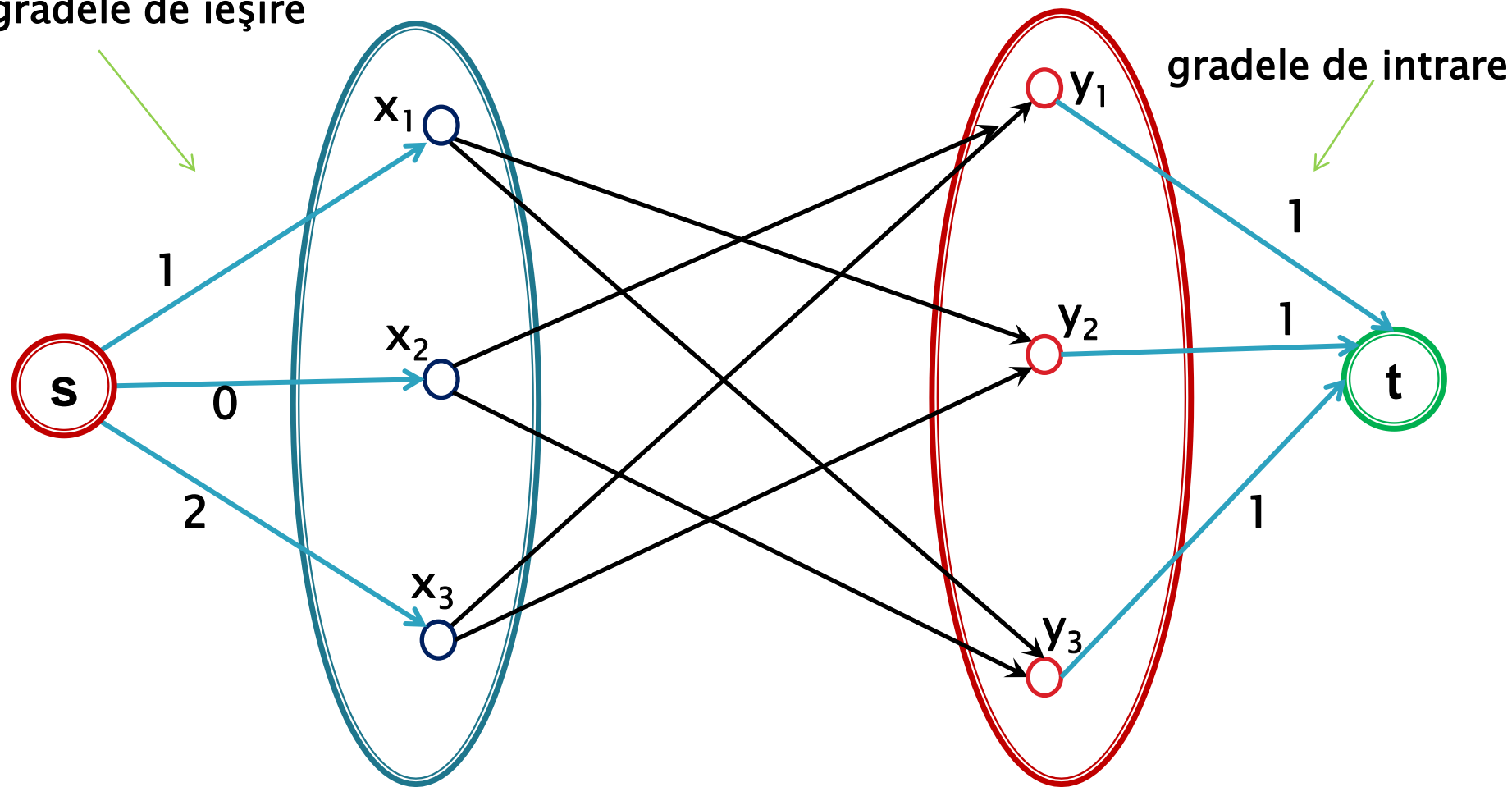


arce $x_i y_j$ cu $i \neq j$

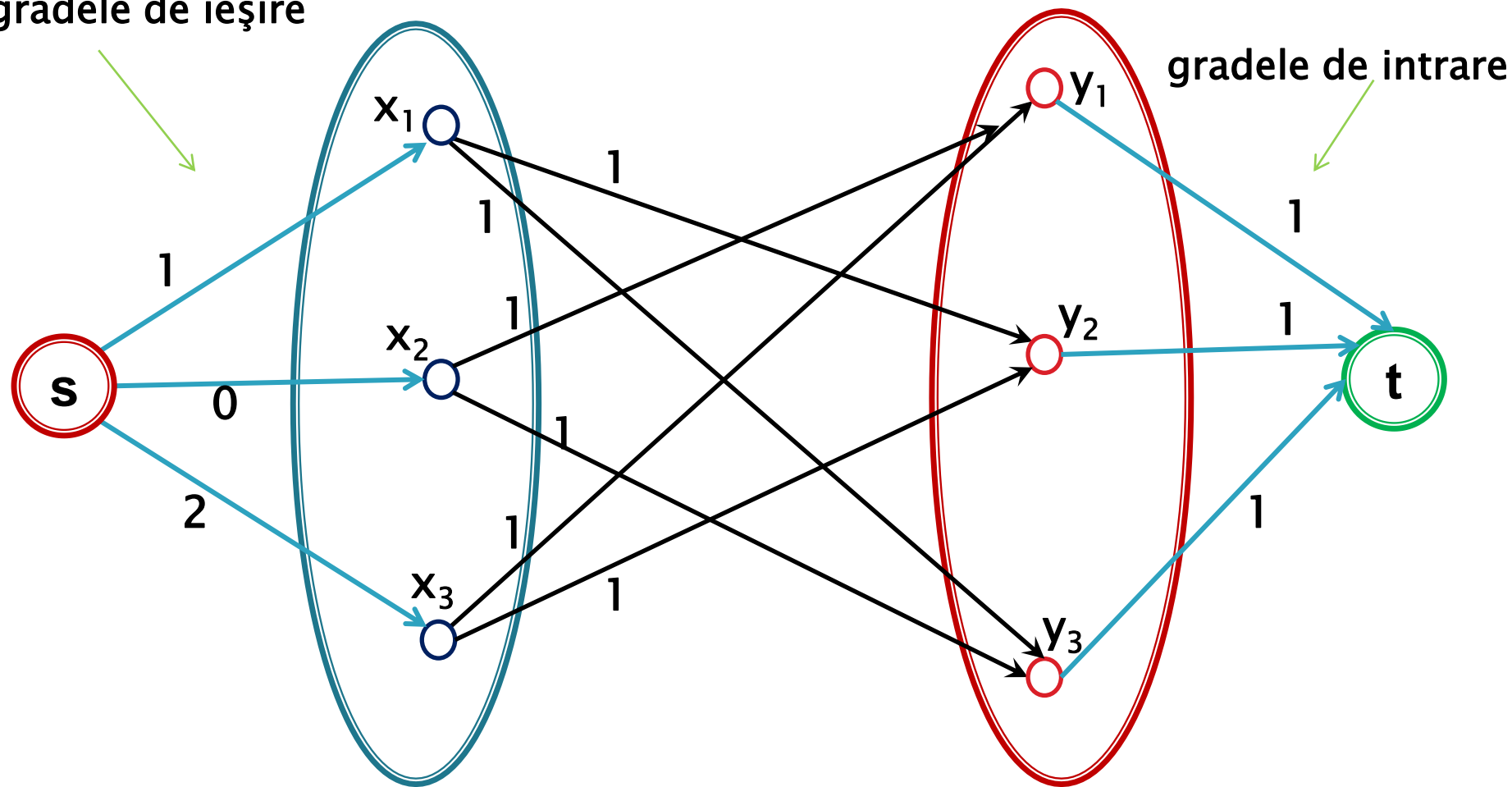
(fluxul pe arcul $x_i y_j$ va fi nenul $\Leftrightarrow ij \in E(G)$)



gradele de ieșire



gradele de ieșire



gradele de intrare

► Proprietate

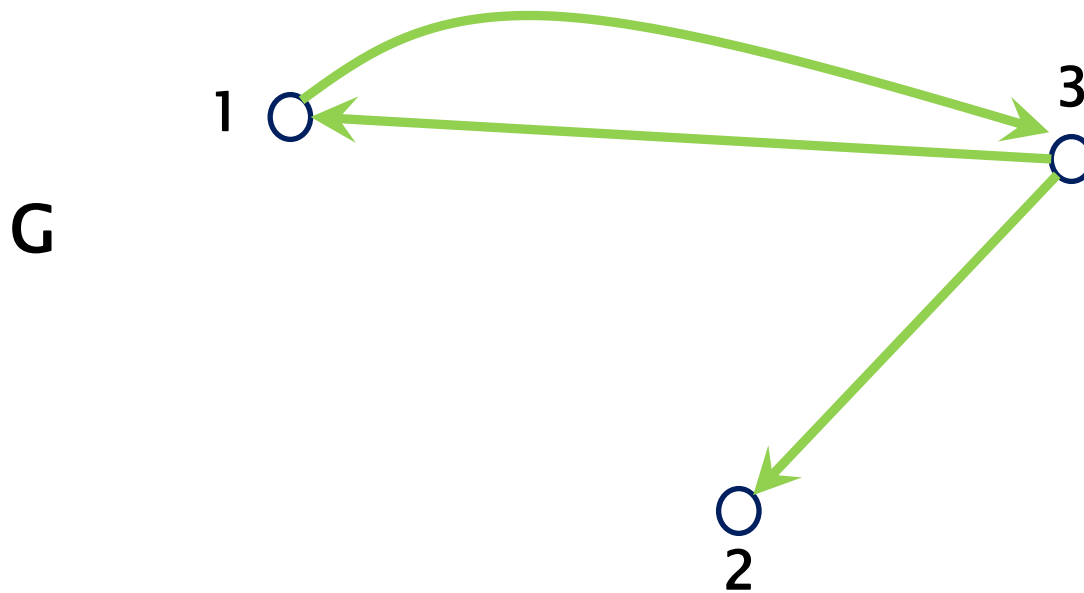
Există graf cu secvențele date \Leftrightarrow în graful asociat fluxul de valoare maximă are

$$\text{val}(f) = d_1^+ + \dots + d_n^+ = d_1^- + \dots + d_n^-$$

(saturează toate arcele care ies din s + toate arcele care intră în t)

tăieturile $(\{s\}, V - \{s\})$, $(V - \{t\}, \{t\})$ sunt minime

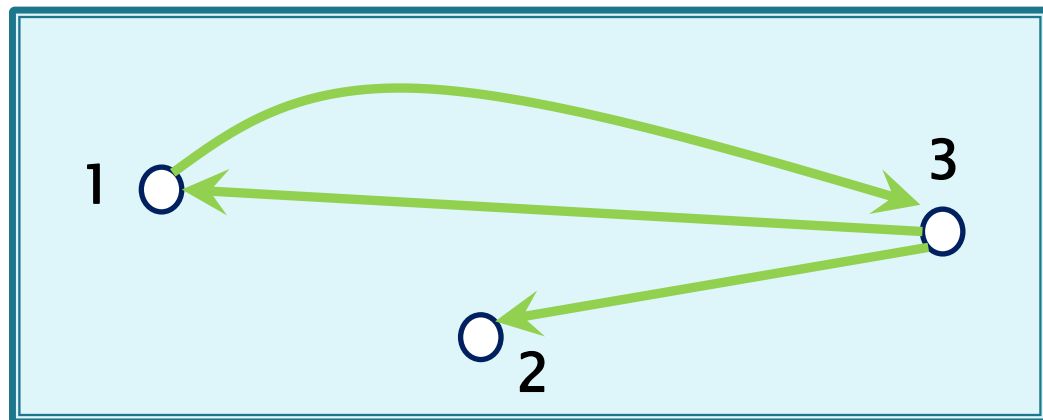
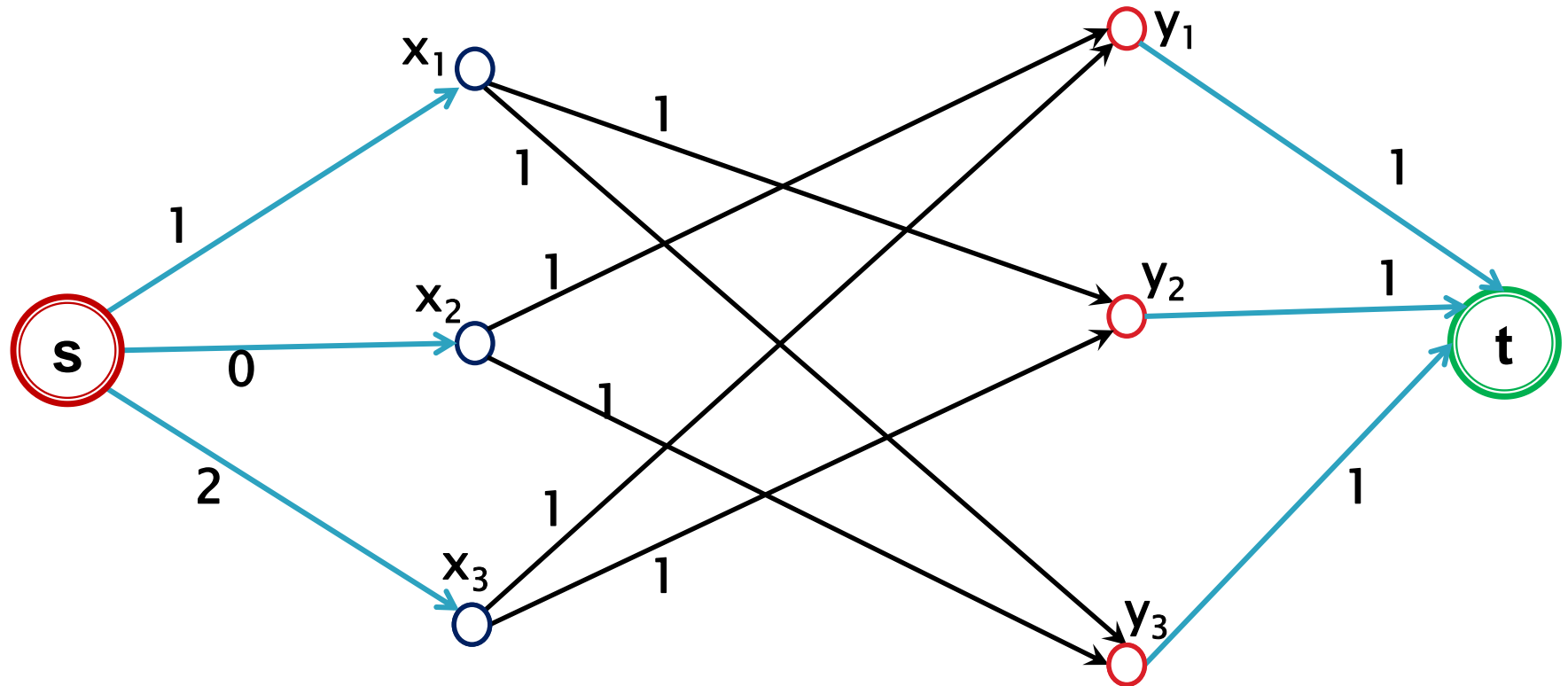
$G \Rightarrow$ flux în rețea



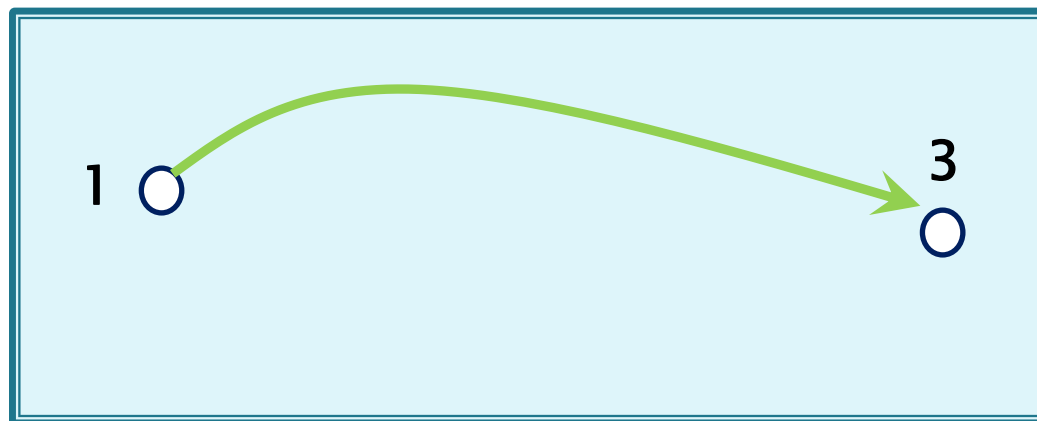
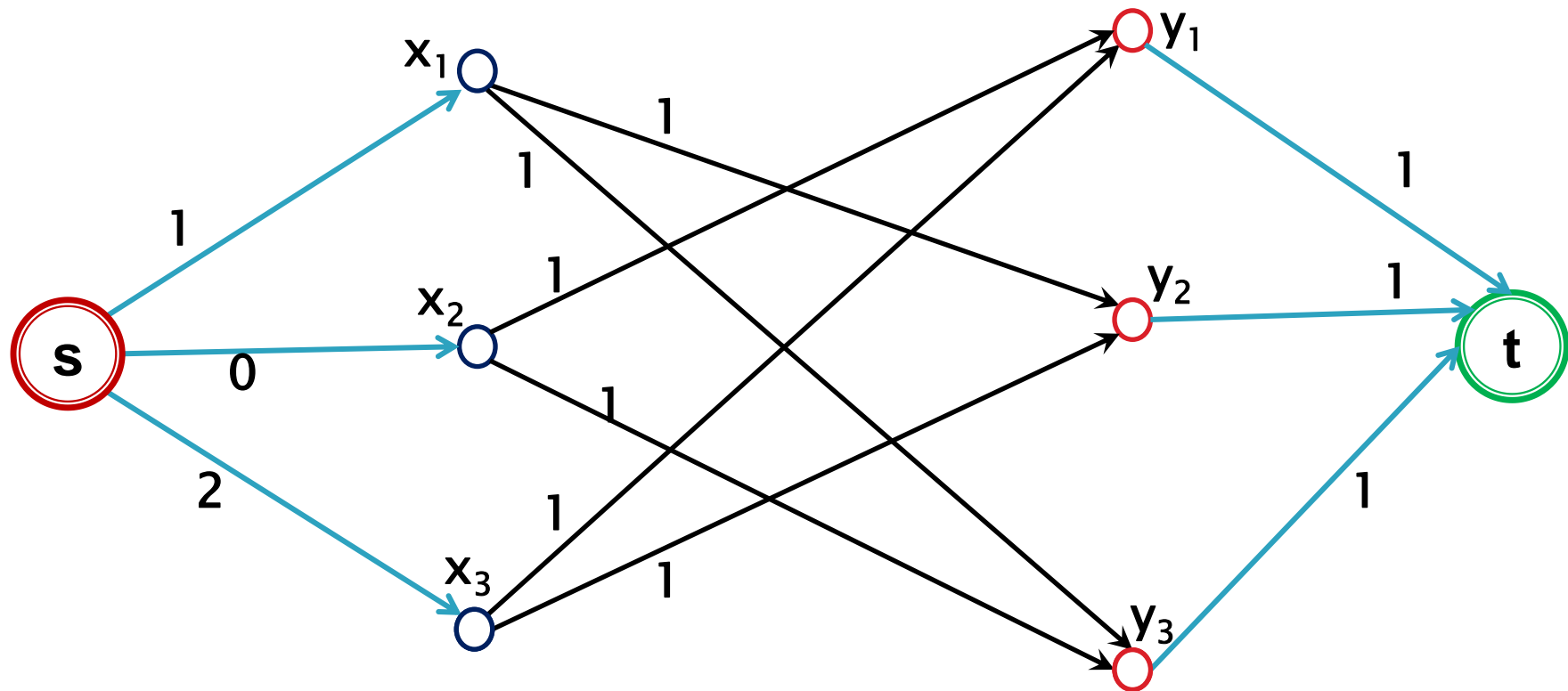
$$s_0^+ = \{1, 0, 2\}$$

$$s_0^- = \{1, 1, 1\}$$

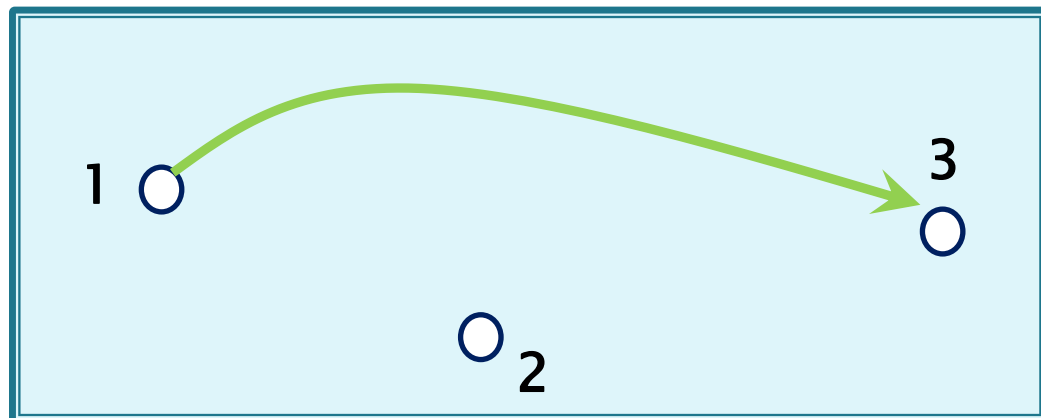
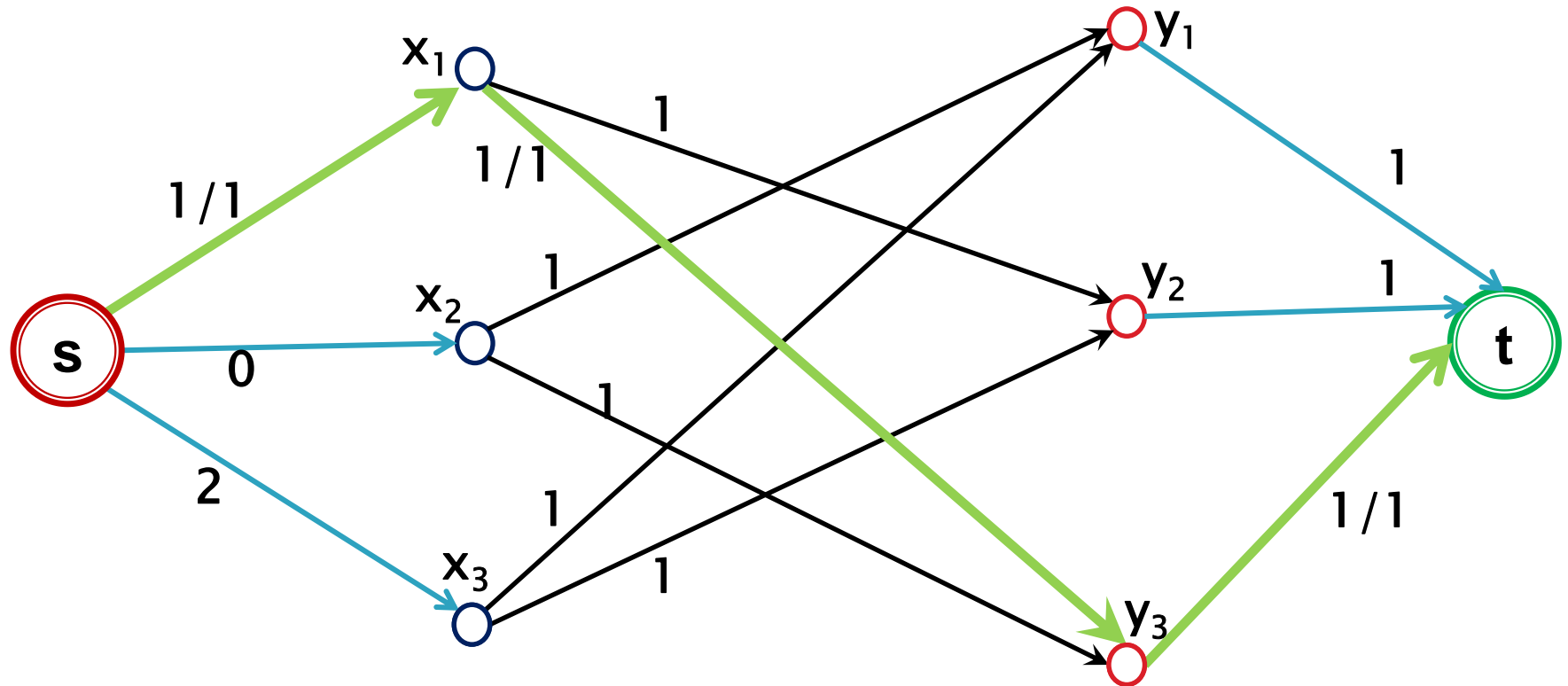
$G \Rightarrow$ flux în rețea



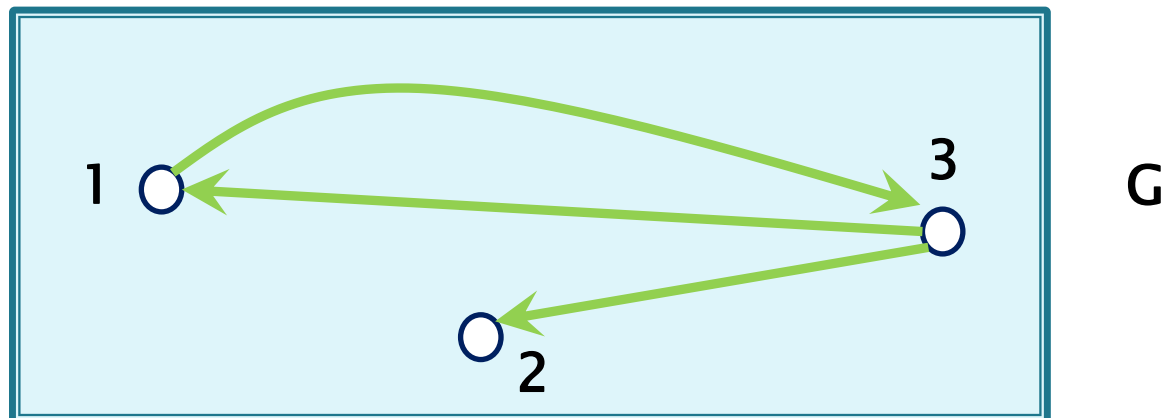
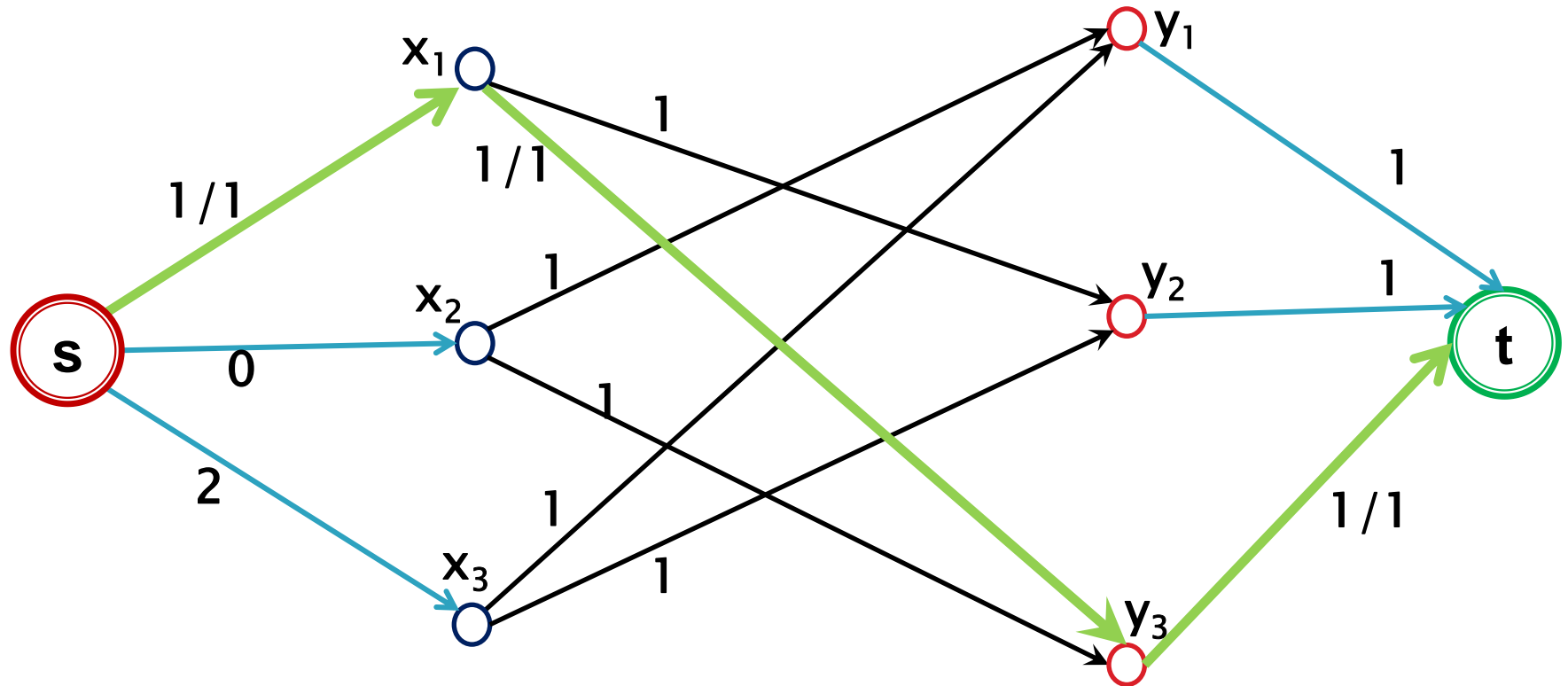
$G \Rightarrow$ flux în rețea



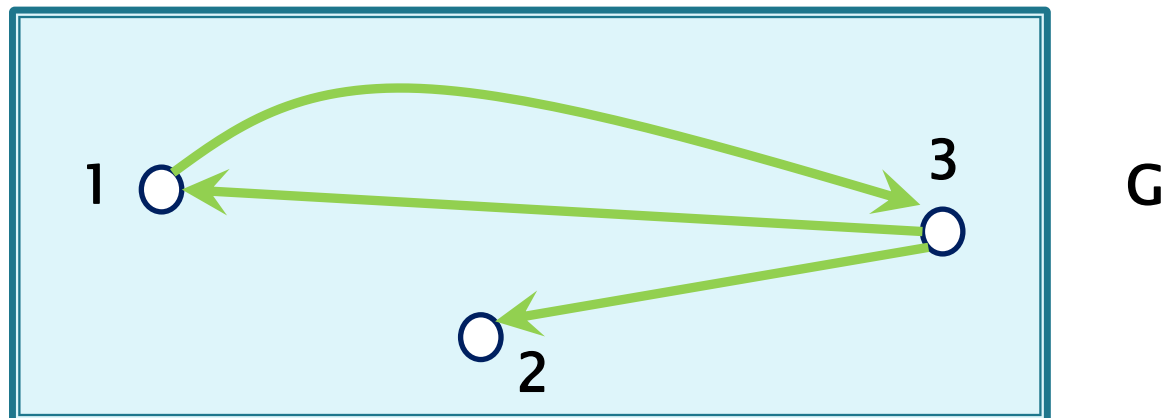
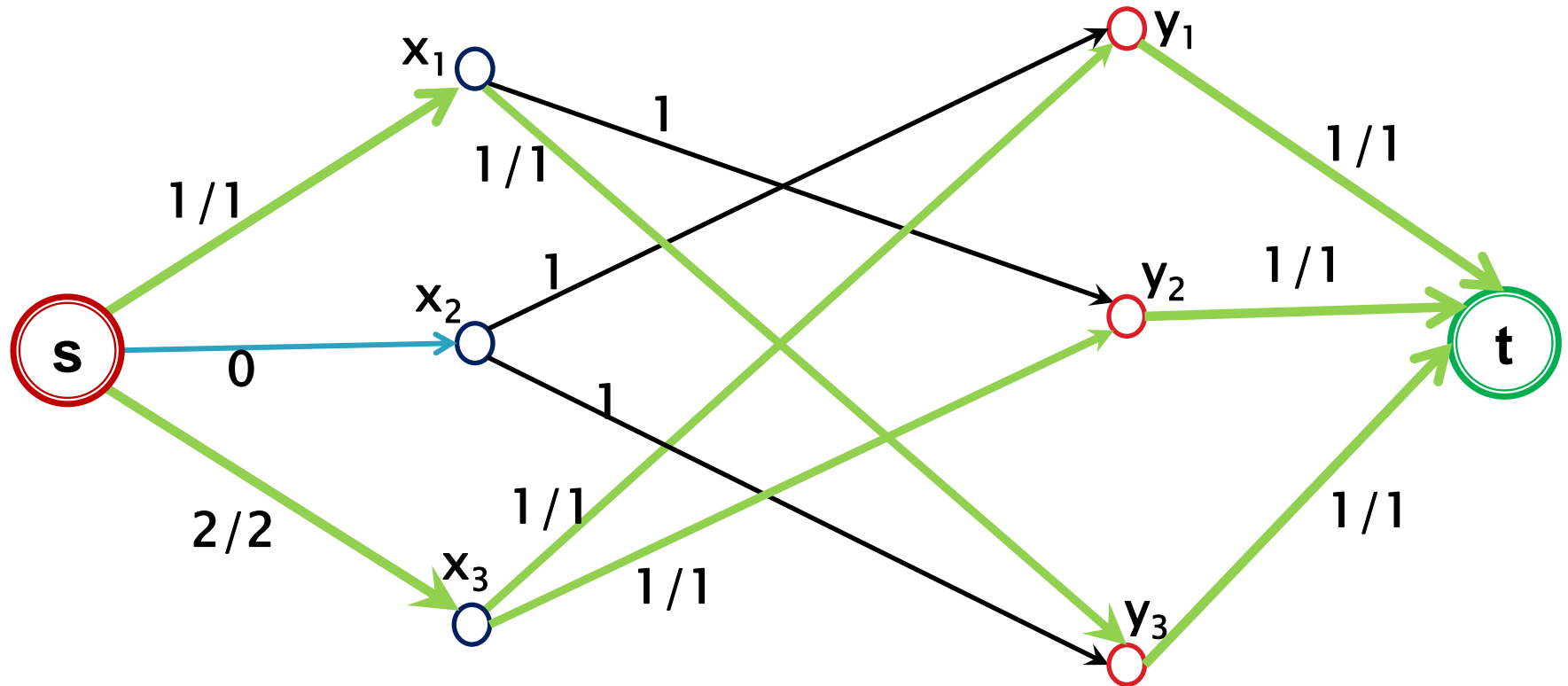
$G \Rightarrow$ flux în rețea



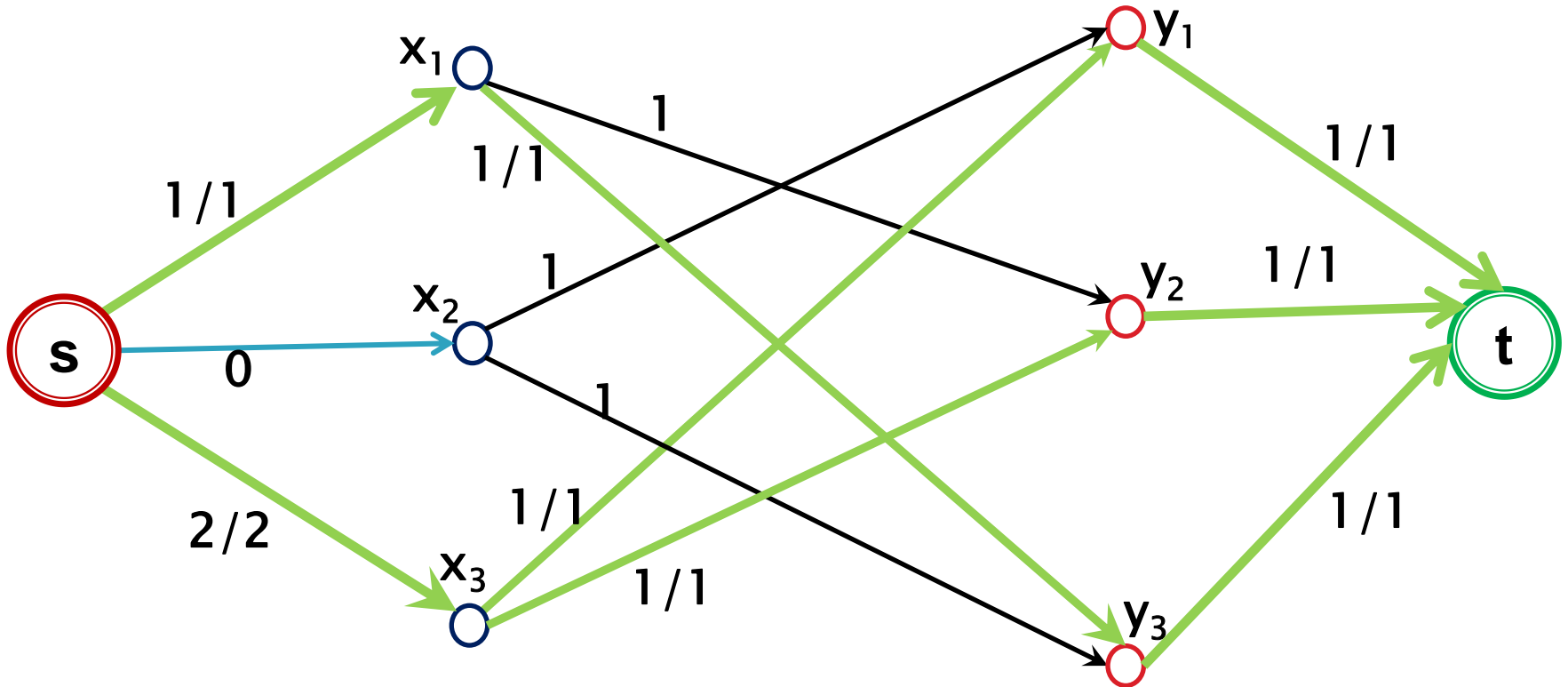
$G \Rightarrow$ flux în rețea



$G \Rightarrow$ flux în rețea

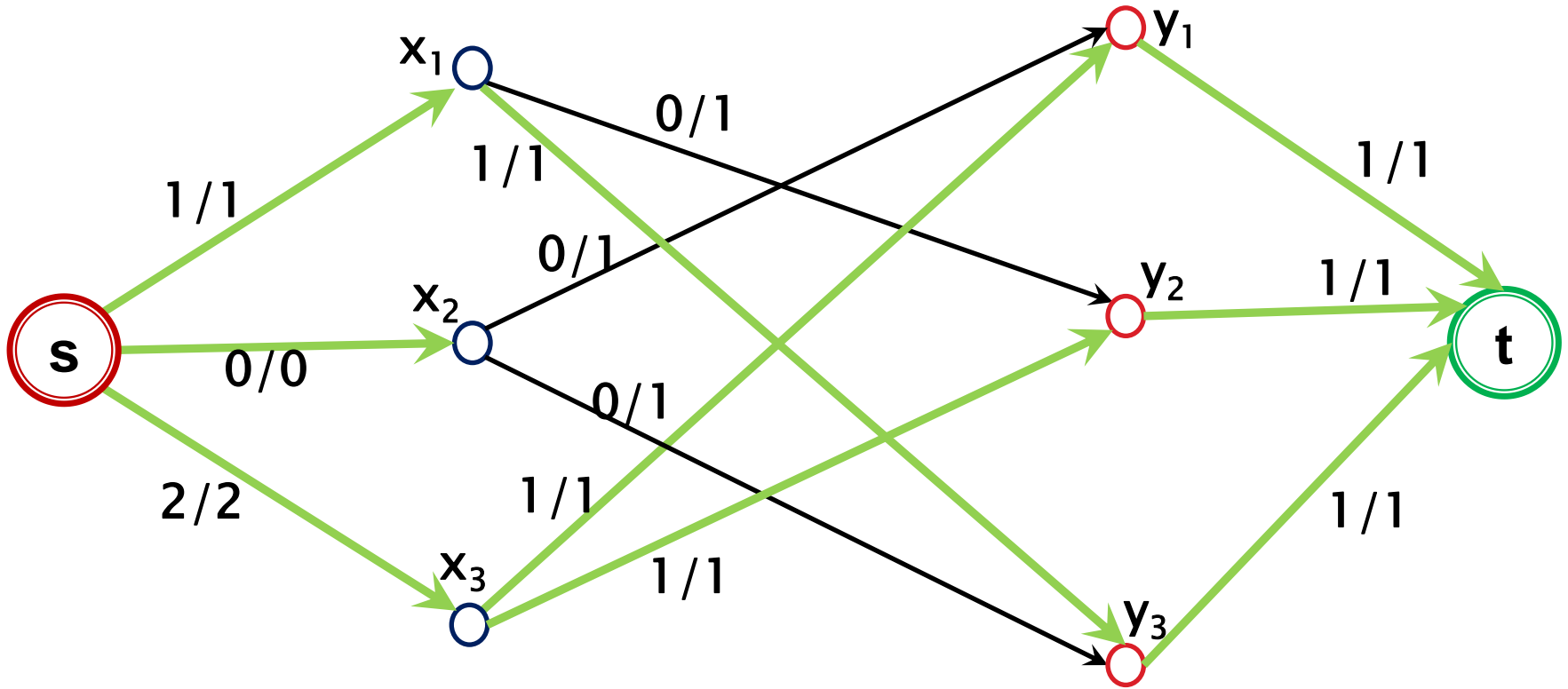


$G \Rightarrow$ flux în rețea

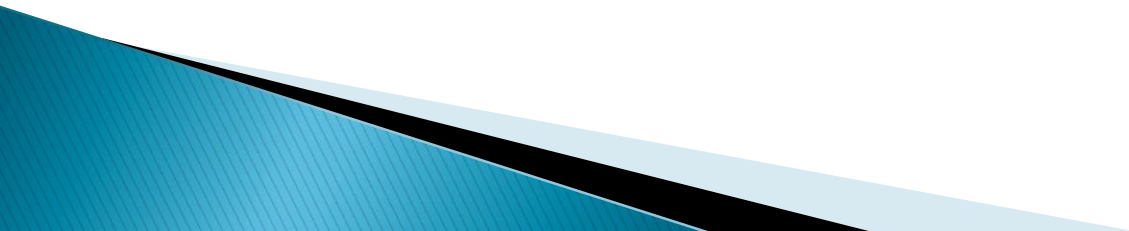


Restul arcelor au fluxul 0

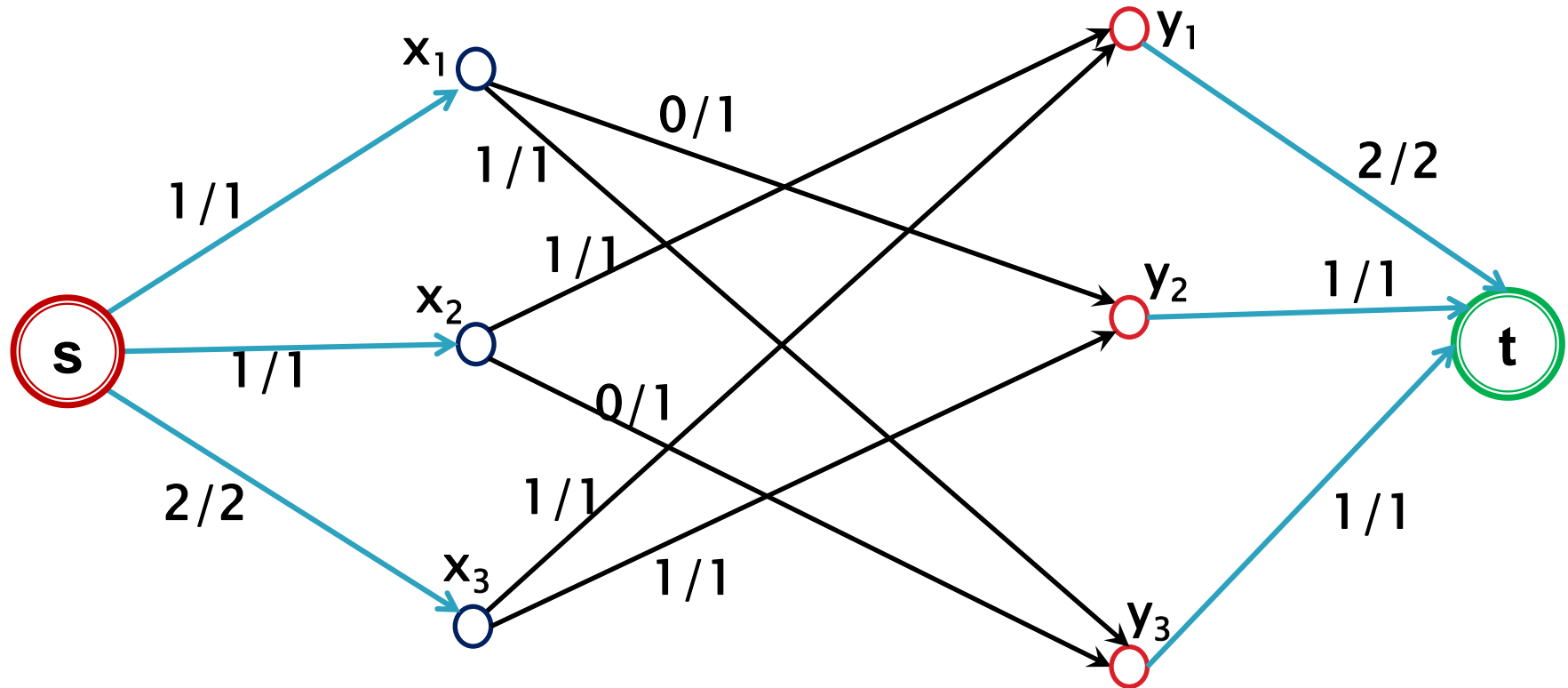
$G \Rightarrow$ flux în rețea



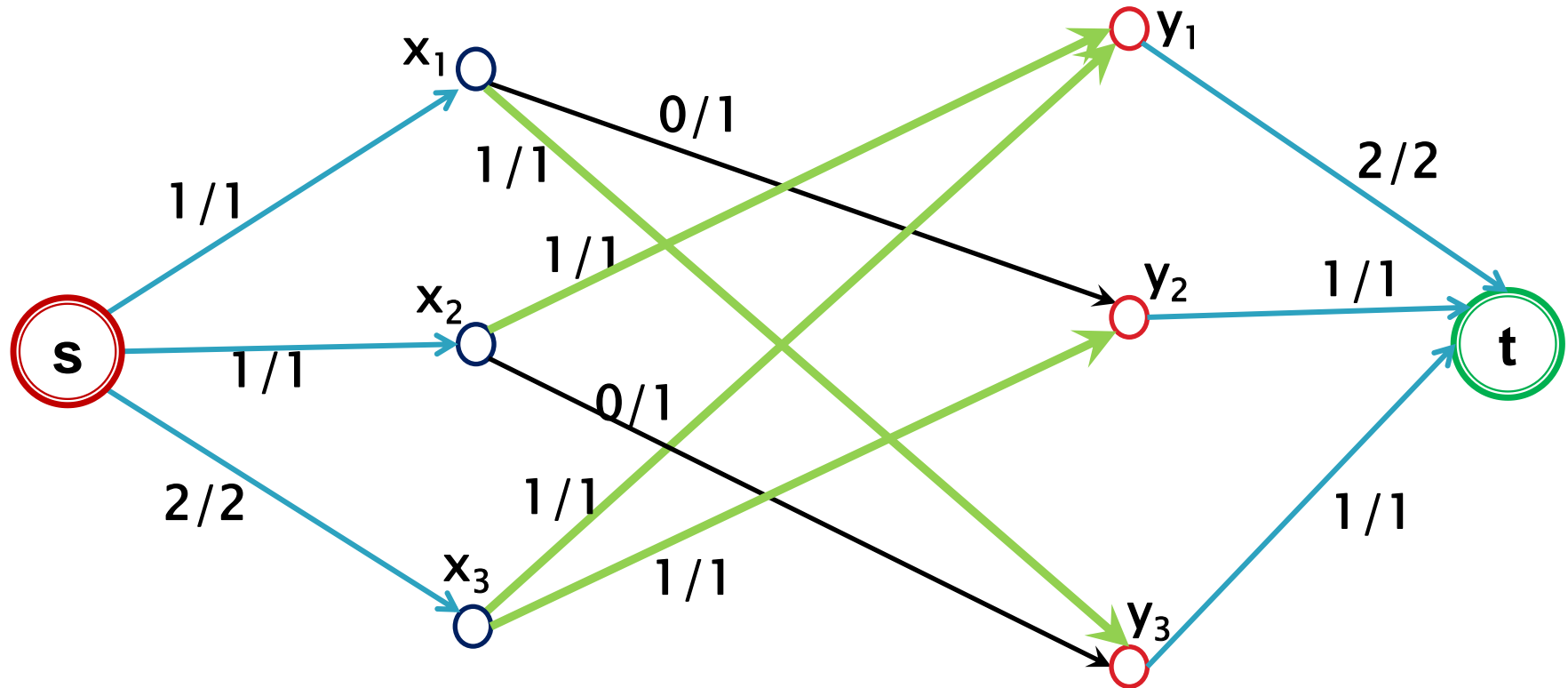
Reciproc



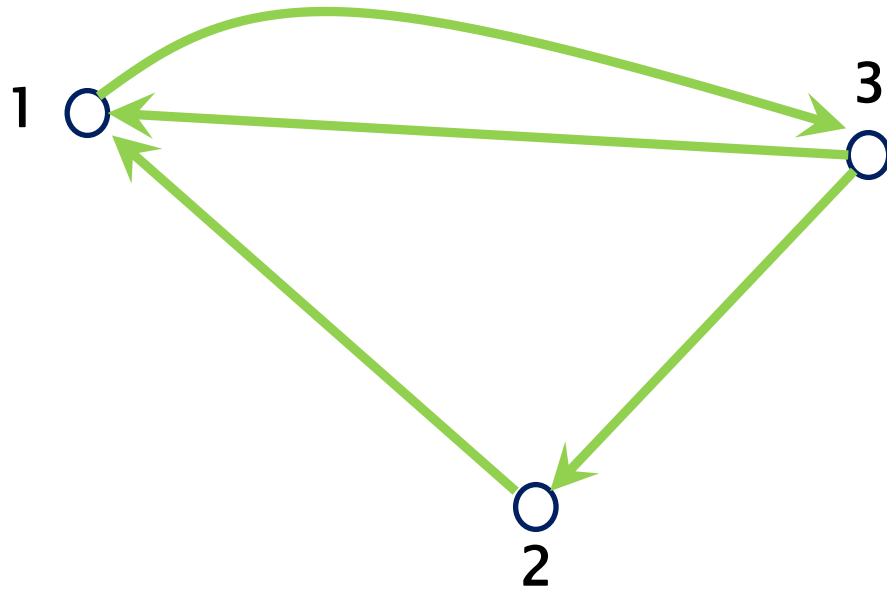
flux în rețea care saturează arcele din s și $t \Rightarrow G$



flux în rețea care saturează arcele din s și $t \Rightarrow G$



flux în rețea care saturează arcele din s și $t \Rightarrow G$



Algoritm de determinare a unui graf orientat G cu

$$s^+(G) = s_0^+ \text{ și } s^-(G) = s_0^-$$

1. Construim N rețeaua de transport asociată
2. Determinăm f^* flux maxim în N

Algoritm de determinare a unui graf orientat G cu

$$s^+(G) = s_0^+ \text{ și } s^-(G) = s_0^-$$

1. Construim N rețeaua de transport asociată
2. Determinăm f^* flux maxim în N
3. Dacă $\text{val}(f^*) < d_1^+ + \dots + d_n^+$ atunci

Nu există G . STOP

Algoritm de determinare a unui graf orientat G cu

$$s^+(G) = s_0^+ \text{ și } s^-(G) = s_0^-$$

1. Construim N rețeaua de transport asociată

2. Determinăm f^* flux maxim în N

3. Dacă $\text{val}(f^*) < d_1^+ + \dots + d_n^+$ atunci

Nu există G . STOP

4. $V(G) = \{1, \dots, n\}$

$$E(G) = \{ij \mid x_i y_j \in N \text{ cu } f^*(x_i y_j) = 1\}$$

Complexitate: $L = c^+(s) = d_1^+ + \dots + d_n^+ = m \Rightarrow O(m^2)$

Aplicații

Alte probleme de asociere

Problema de asociere (temă)

- ▶ Se dau 2 mulțimi de obiecte, spre exemplu produse și clienți (joburi/mașini, pagini web/serve, echipe turneu etc).
 - Pentru fiecare produs x se cunoaște $c(x)$ = numărul de unități disponibile din produsul x

Problema de asociere (temă)

- ▶ Se dau 2 mulțimi de obiecte, spre exemplu produse și clienți (joburi/mașini, pagini web/serve, echipe turneu etc).
 - Pentru fiecare produs x se cunoaște $c(x)$ = numărul de unități disponibile din produsul x
 - Pentru fiecare client y se cunoaște $c(y)$ = numărul maxim de unități de produse pe care le poate primi (în total, din toate produsele)

Problema de asociere (temă)

- ▶ Se dau 2 mulțimi de obiecte, spre exemplu produse și clienți (joburi/mașini, pagini web/serve, echipe turneu etc).
 - Pentru fiecare produs x se cunoaște $c(x)$ = numărul de unități disponibile din produsul x
 - Pentru fiecare client y se cunoaște $c(y)$ = numărul maxim de unități de produse pe care le poate primi (în total, din toate produsele)
 - Pentru fiecare pereche produs–client (x,y) se cunoaște $c(x,y)$ = numărul maxim de unități din produsul x pe care le poate primi clientul y

Problema de asociere (temă)

- ▶ Se dau 2 mulțimi de obiecte, spre exemplu produse și clienți (joburi/masini, pagini web/serve, echipe turneu etc).
 - Pentru fiecare produs x se cunoaște $c(x)$ = numărul de unități disponibile din produsul x
 - Pentru fiecare client y se cunoaște $c(y)$ = numărul maxim de unități de produse pe care le poate primi (în total, din toate produsele)
 - Pentru fiecare pereche produs–client (x,y) se cunoaște $c(x,y)$ = numărul maxim de unități din produsul x pe care le poate primi clientul y

Să se determine o modalitate de a distribui cât mai multe produse (unități de produse) clienților cu respectarea constrângerilor

Problema de asociere (temă)

- ▶ **Observație** – Problema determinării unui cuplaj de cardinal maxim într-un graf bipartit $G=(X \cup Y, E)$ este un caz particular al acestei probleme, pentru
 - $c(x) = c(y) = 1, \forall x \in X, y \in Y$
 - $c(x, y) = 1$, dacă $xy \in E$
 - $c(x, y) = 0$, dacă $xy \notin E$
- <http://jeffe.cs.illinois.edu/teaching/algorithms/2009/notes/17-maxflowapps.pdf>

Aplicație

Drumuri arc-disjuncte între două vârfuri.

Conectivitatea unui graf

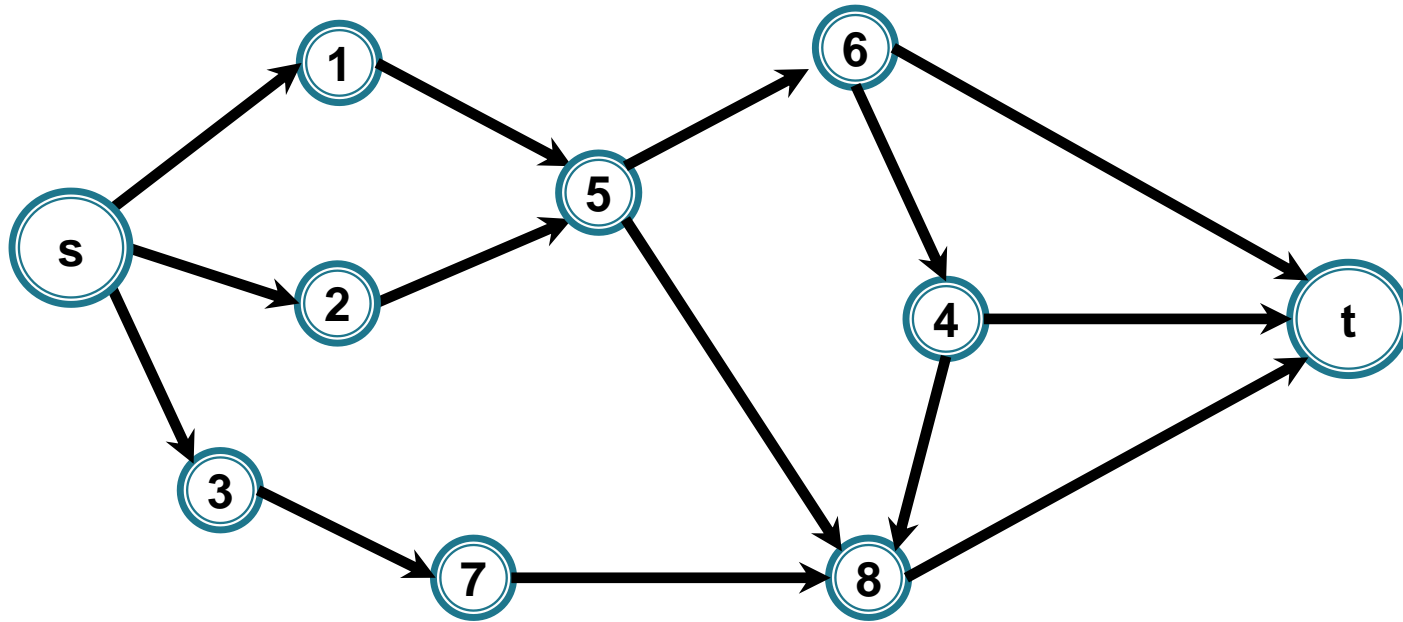
s-t drumuri arc-disjuncte

- ▶ $G = (V, E)$ – orientat, conex (graful neorientat suport)
- ▶ s, t – două vârfuri

Să se determine numărul maxim k de **s-t drumuri elementare arc-disjuncte** (+ k astfel de drumuri)

- Două drumuri P_1, P_2 s.n. **arc-disjuncte** dacă $E(P_1) \cap E(P_2) = \emptyset$

s-t drumuri arc-disjuncte



s-t drumuri arc-disjuncte

- ▶ $G = (V, E)$ – orientat, conex (graful neorientat suport)
- ▶ s, t – două vârfuri

Să se determine numărul maxim de **s-t drumuri arc-disjuncte**
+ k astfel de drumuri)

- ▶ **Aplicații**
 - Fiabilitatea rețelelor, conectivitate
 - Probleme de strategie
 - Măsuri de centralitate (a unui nod) în rețele sociale

s-t drumuri arc-disjuncte

- ▶ $G = (V, E)$ – orientat, conex (graful neorientat suport)
- ▶ s, t – două vârfuri

Să se determine numărul maxim k de **s-t drumuri elementare arc-disjuncte** (+ k astfel de drumuri)

- Două drumuri P_1, P_2 s.n. **arc-disjuncte** dacă $E(P_1) \cap E(P_2) = \emptyset$

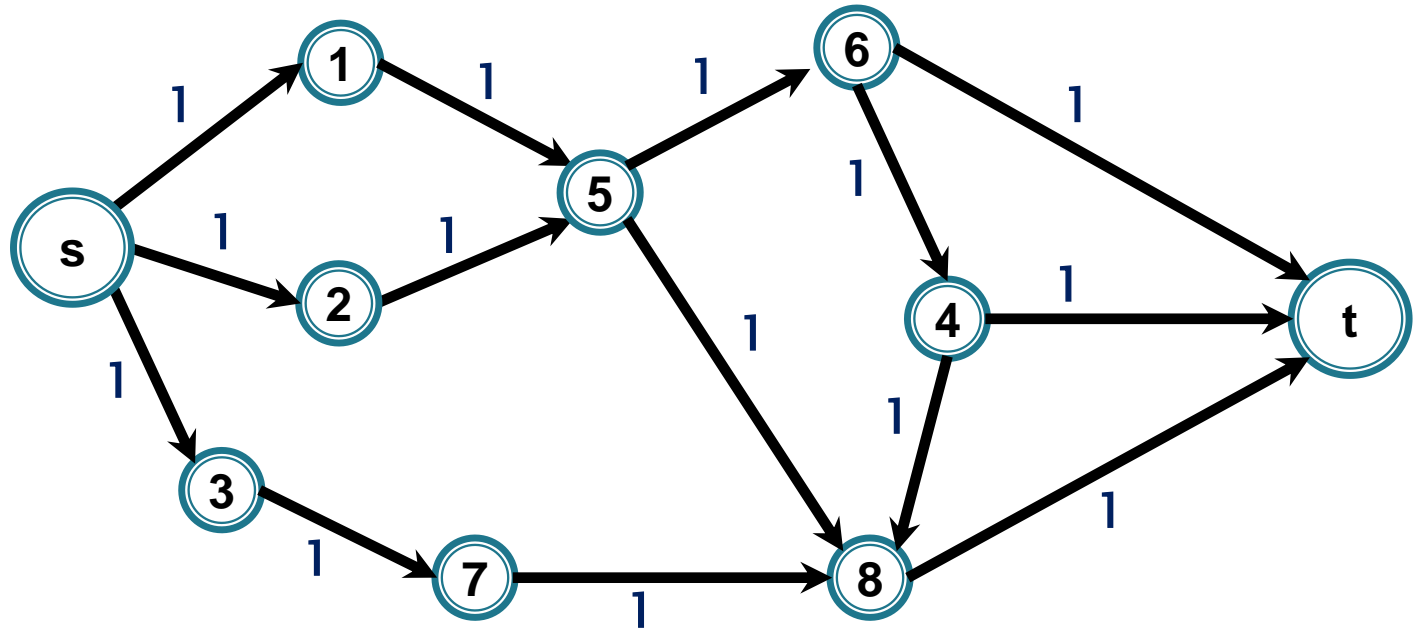
Temă (seminar)

- Jon Kleinberg, Éva Tardos, **Algorithm Design**, Addison-Wesley 2005

s-t drumuri arc-disjuncte

► Intuitiv:

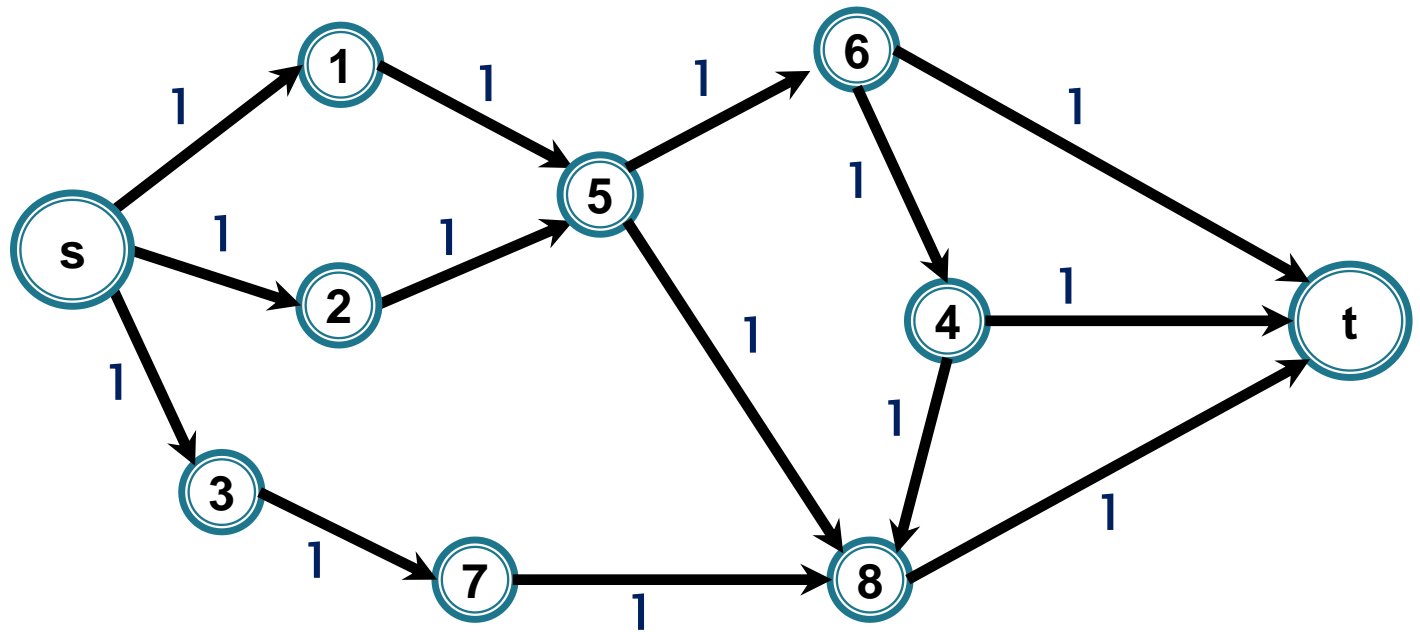
- Asociem fiecărui arc capacitatea 1
- Fluxul maxim: $f(e) \in \{0, 1\}$



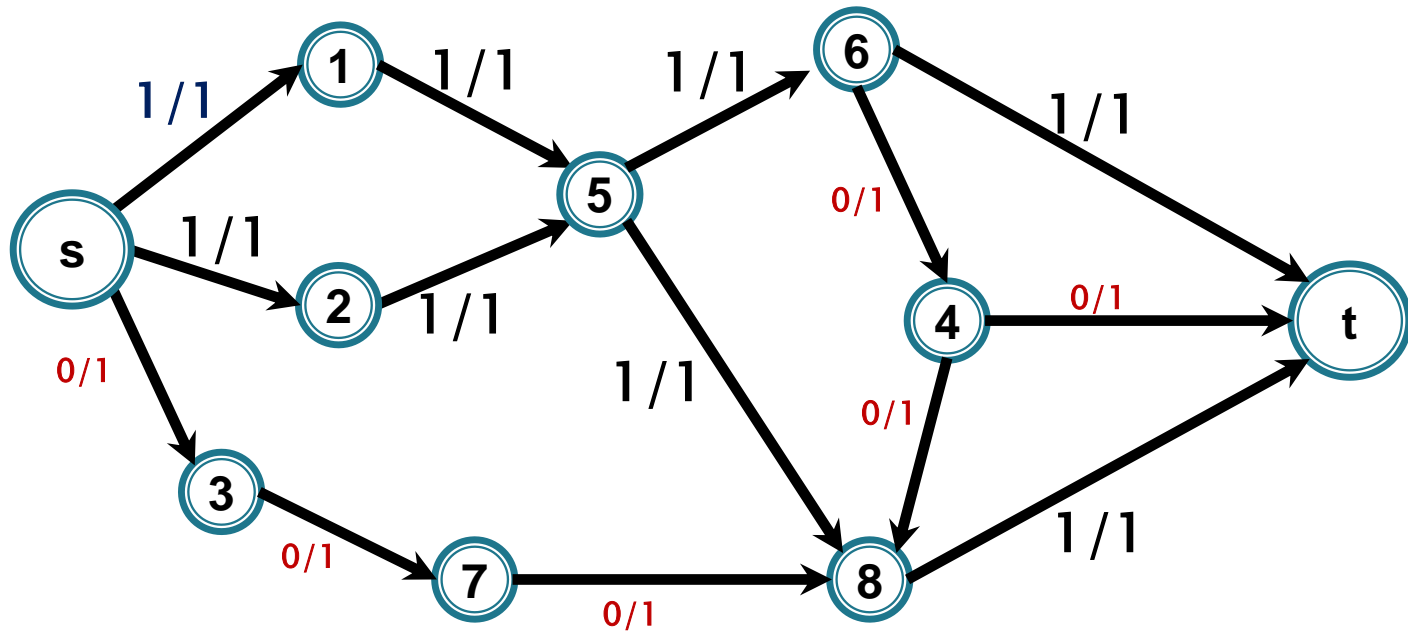
s-t drumuri arc-disjuncte

► Intuitiv:

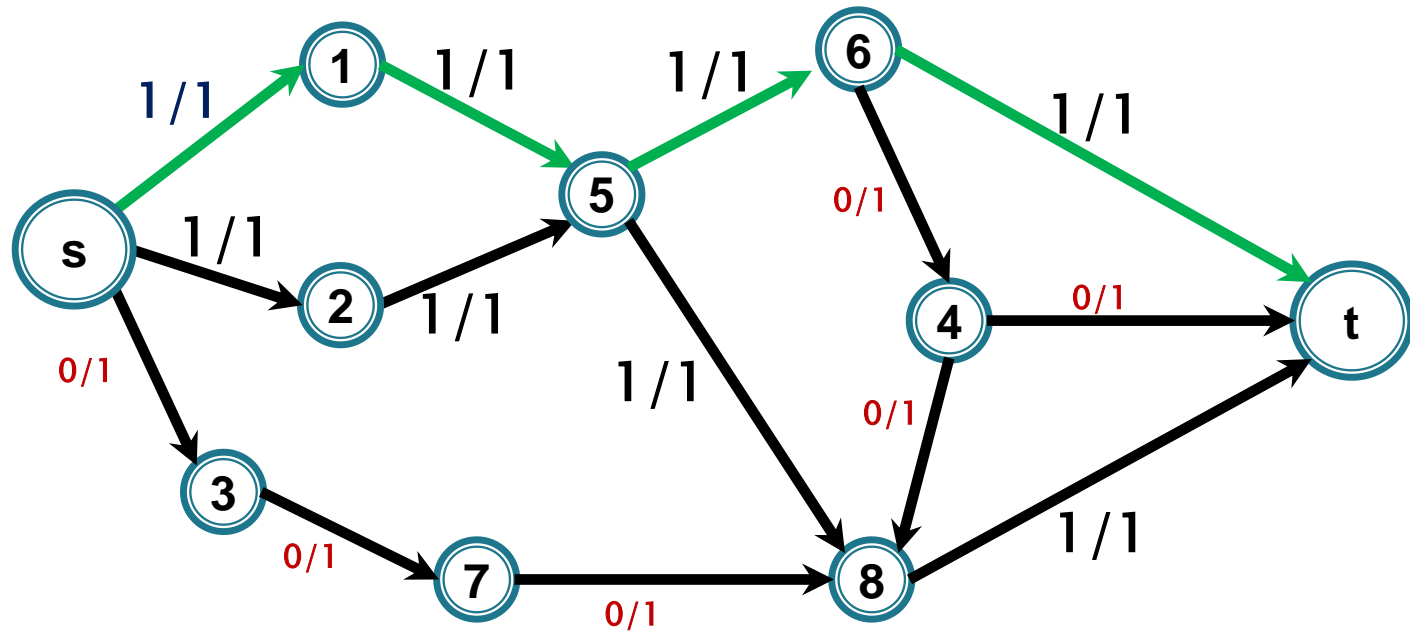
- Asociem fiecărui arc capacitatea 1
- Fluxul maxim: $f(e) \in \{0, 1\}$
- Un drum de la s la t = **traseul parcurs de o unitate de flux de la s la t**
- Numărul de s-t drumuri arc-disjuncte = valoarea fluxului maxim



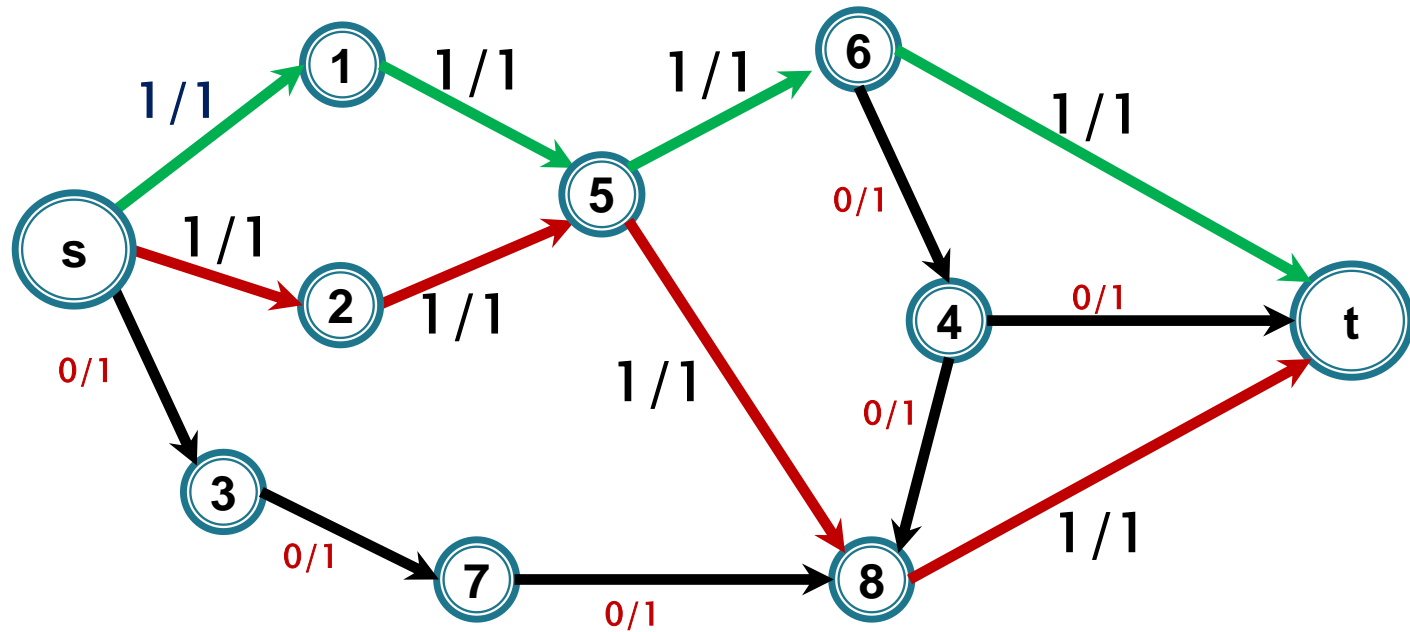
s-t drumuri arc-disjuncte



s-t drumuri arc-disjuncte



s-t drumuri arc-disjuncte



s-t drumuri arc-disjuncte

► Teorema lui Menger

Fie G graf orientat, s, t două vârfuri distincte în G .

Numărul minim de arce care trebuie eliminate pentru ca s și t să nu mai fie conectate printr-un drum (să fie **separate**) =
numărul maxim de drumuri arc-disjuncte de la s la t

s-t drumuri arc-disjuncte

► Teorema lui Menger

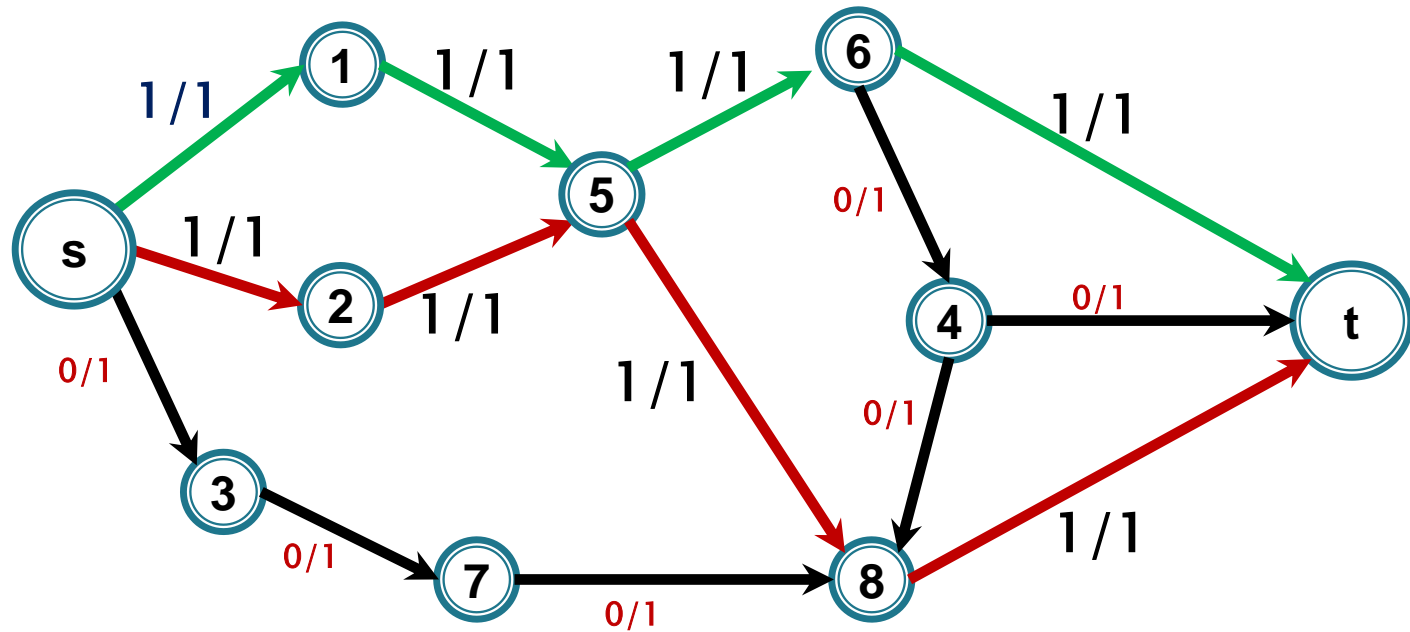
Fie G graf orientat, s , t două vârfuri distincte în G .

Numărul minim de arce care trebuie eliminate pentru ca s și t să nu mai fie conectate printr-un drum (să fie **separate**) =
numărul maxim de drumuri arc-disjuncte de la s la t



O astfel de mulțime de arce se poate determina cu algoritmul Ford Fulkerson?

s-t drumuri arc-disjuncte



s-t drumuri arc-disjuncte

► Teorema lui Menger

Fie G graf orientat, s , t două vârfuri distincte în G .

Numărul minim de arce care trebuie eliminate pentru ca s și t să nu mai fie conectate printr-un drum (să fie **separate**) =
numărul maxim de drumuri arc-disjuncte de la s la t

O astfel de mulțime de arce se poate determina cu algoritmul Ford Fulkerson

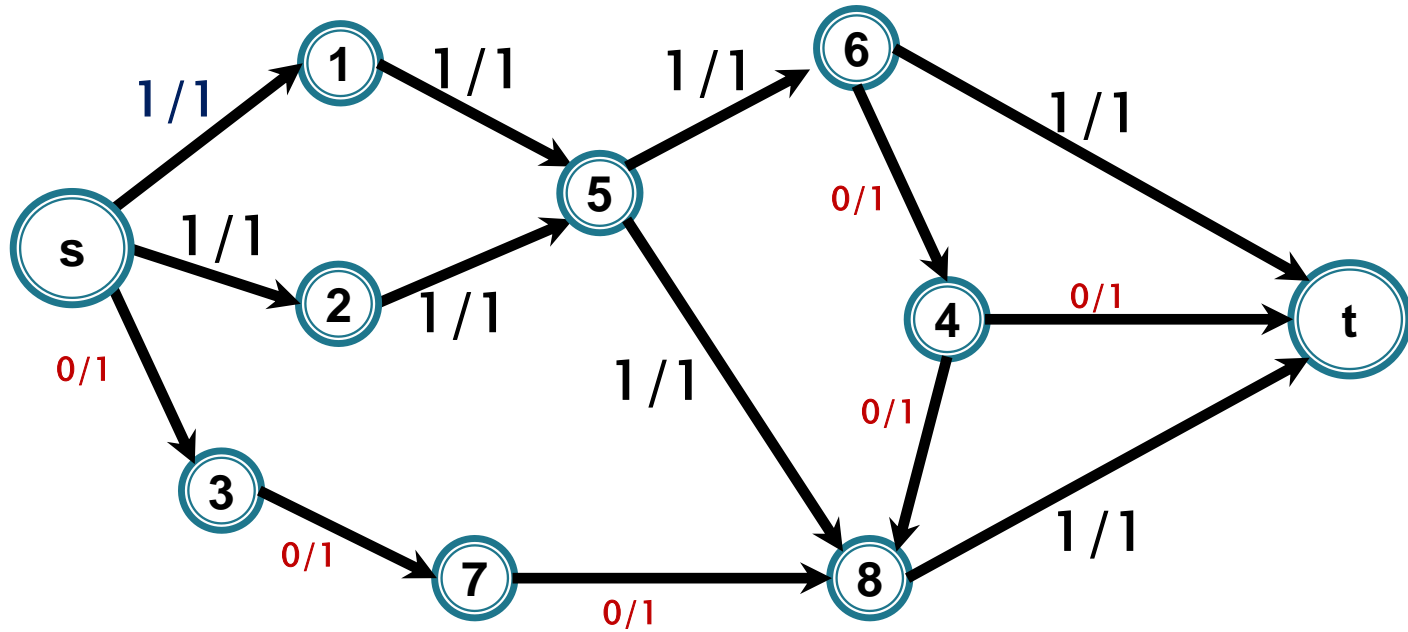


Sunt arcele directe ale tăieturii minime

s-t drumuri arc-disjuncte



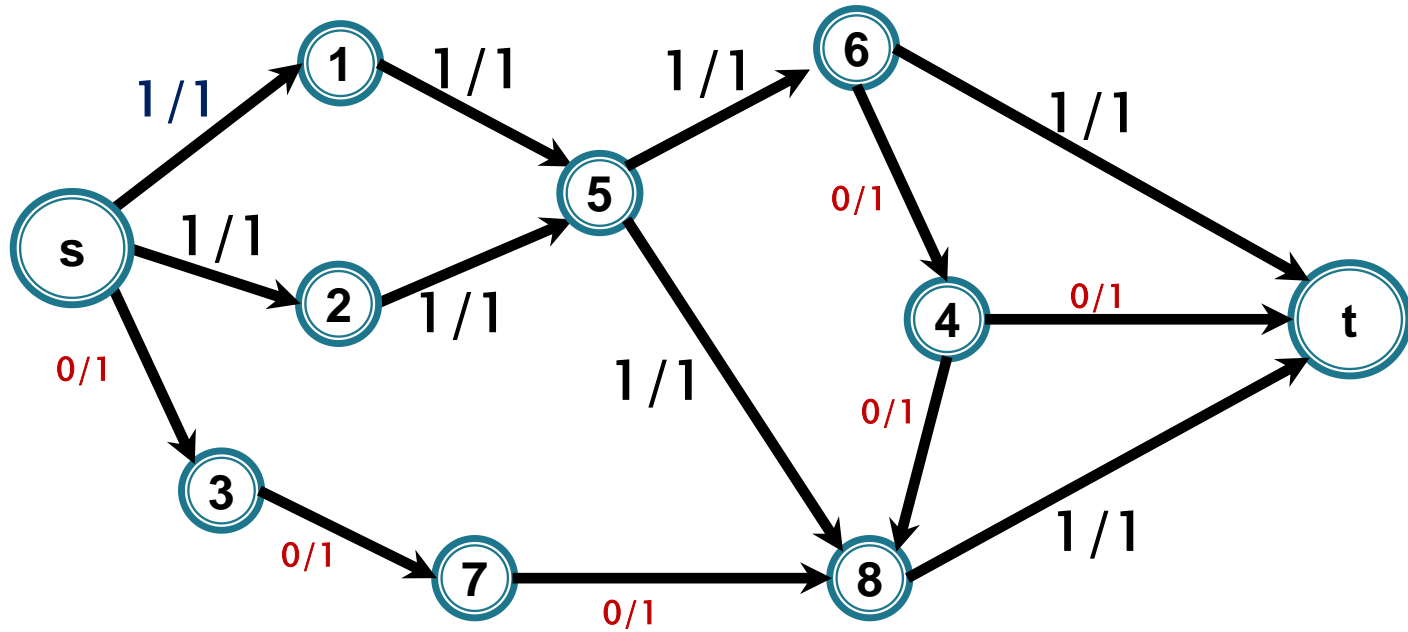
Cum determinăm tăietura minimă?



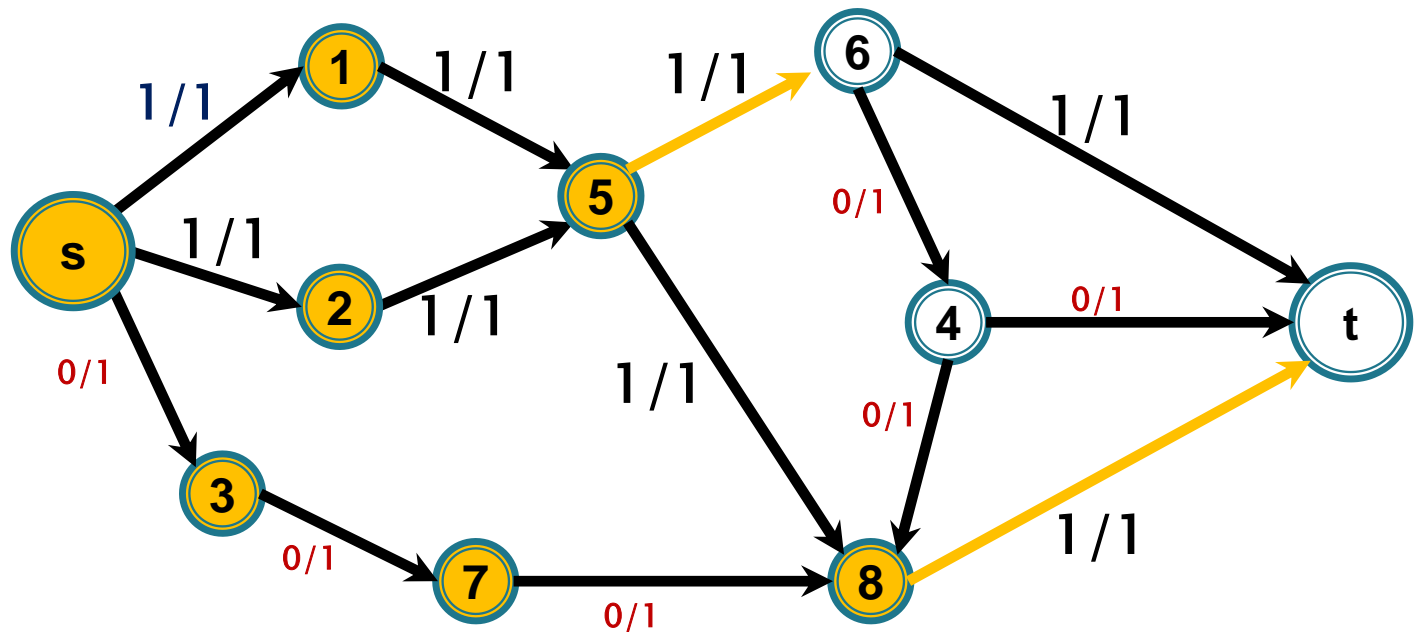
s-t drumuri arc-disjuncte



Mulțimea vârfurilor accesibile din s prin lanțuri f-nesaturate



s-t drumuri arc-disjuncte



s-t drumuri arc-disjuncte

Variante

- ▶ Aceeași problemă pentru
 - $G = (V, E)$ – neorientat conex, $|E| > 2$
- ▶ Aceeași problemă pentru **vârfuri** (s-t drumuri care nu au vârfuri interne în comun)

s-t drumuri arc-disjuncte

- ▶ Muchie-conectivitatea lui G $k'(G) =$ cardinalul minim al unei mulțimi de muchii $F \subseteq E$ cu proprietatea că $G - F$ nu mai este conex
- ▶ Dacă $k'(G) \geq t$, G se numește **t-muchie conex**

s-t drumuri arc-disjuncte

- ▶ **Muchie-conectivitatea lui G** $k'(G) =$ cardinalul minim al unei mulțimi de muchii $F \subseteq E$ cu proprietatea că $G - F$ nu mai este conex
- ▶ Dacă $k'(G) \geq t$, G se numește **t-muchie conex**
 - Amintim (laborator+seminar):
 - există muchie critică $\Rightarrow G$ este 1-conex
 - Nu există muchie critică $\Rightarrow G$ este 2-conex
- ▶ **Cu ajutorul algoritmului de flux maxim putem determina (muchie)-conectivitatea unui graf**