Aplicaţie
Flux maxim → cuplaj maxim
în grafuri bipartite

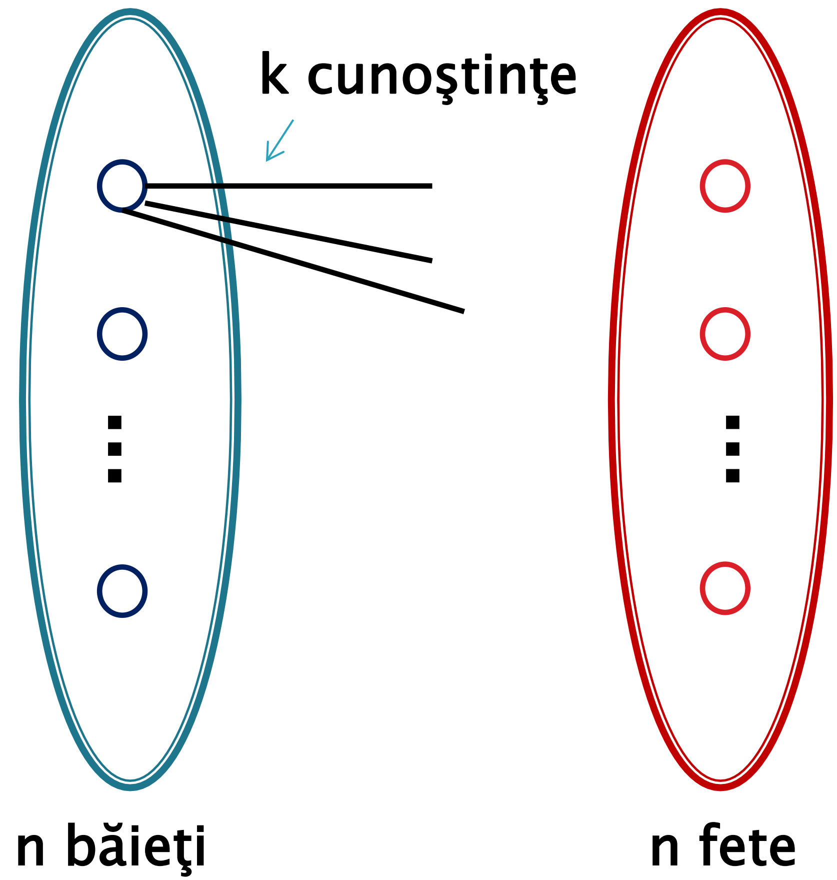
Cuplaje

Repere istorice. Aplicaţii

- **Problema seratei (perechilor)** – sec XIX
  - n băieţi, n fete
  - Un băiat cunoaşte exact k fete
  - O fată cunoaşte exact k băieţi

Repere istorice. Aplicaţii

- **Problema seratei (perechilor)**



Repere istorice. Aplicaţii

- **Problema seratei (perechilor)** – sec XIX
  - Se poate organiza o repriză de dans astfel încât fiecare participant să danseze cu o cunoştinţă a sa?

Repere istorice. Aplicaţii

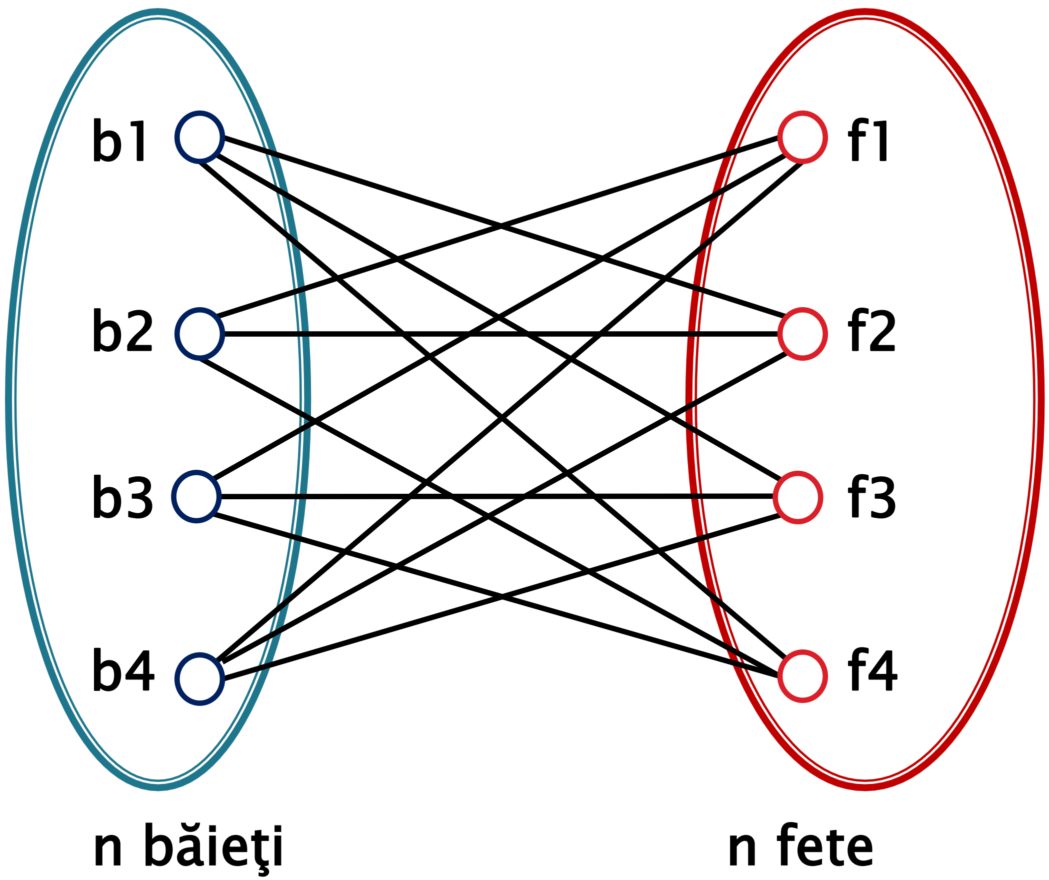
- **Problema seratei (perechilor)** – sec XIX
  - Se poate organiza o repriză de dans astfel încât fiecare participant să danseze cu o cunoştinţă a sa?
  - Se pot organiza **k** reprize de dans în care fiecare participant să danseze câte un dans cu fiecare cunoştinţă a sa? – **vom reveni la cursuri dedicate cuplajelor**

Repere istorice. Aplicaţii

- **Problema seratei (perechilor)** – sec XIX

n=4

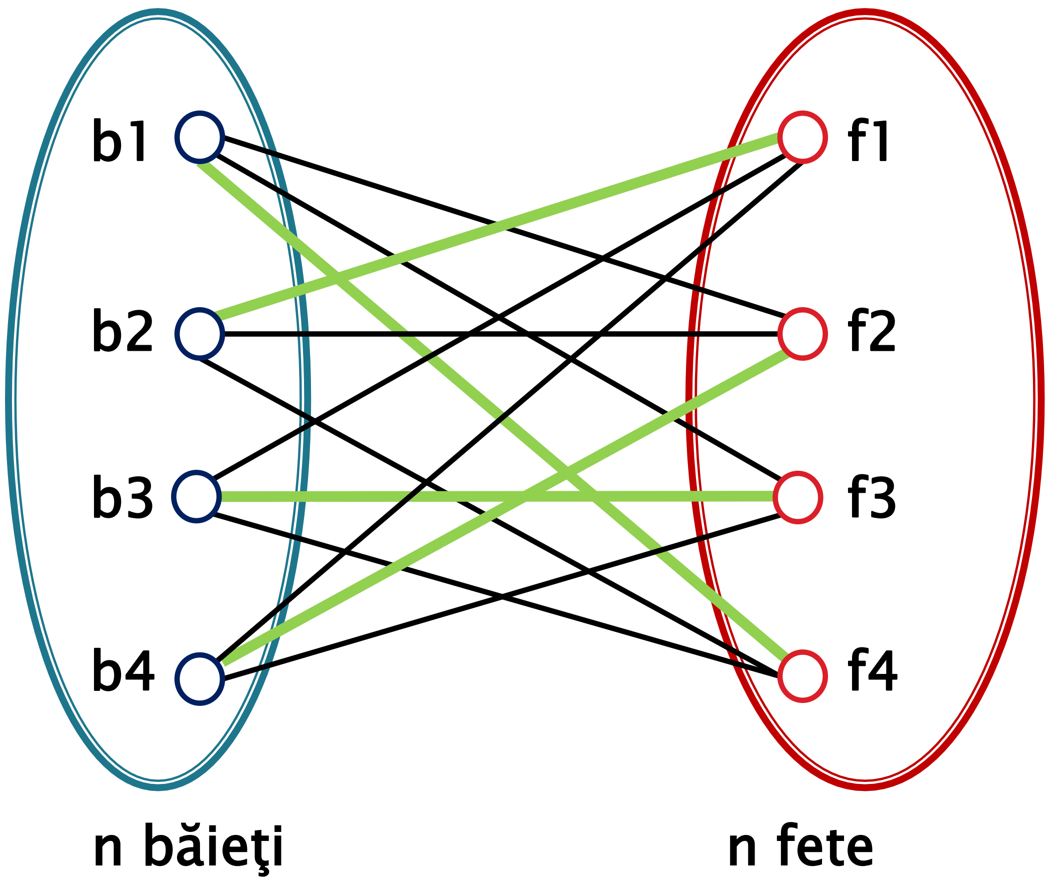
k=3



Repere istorice. Aplicaţii

- O repriză de dans

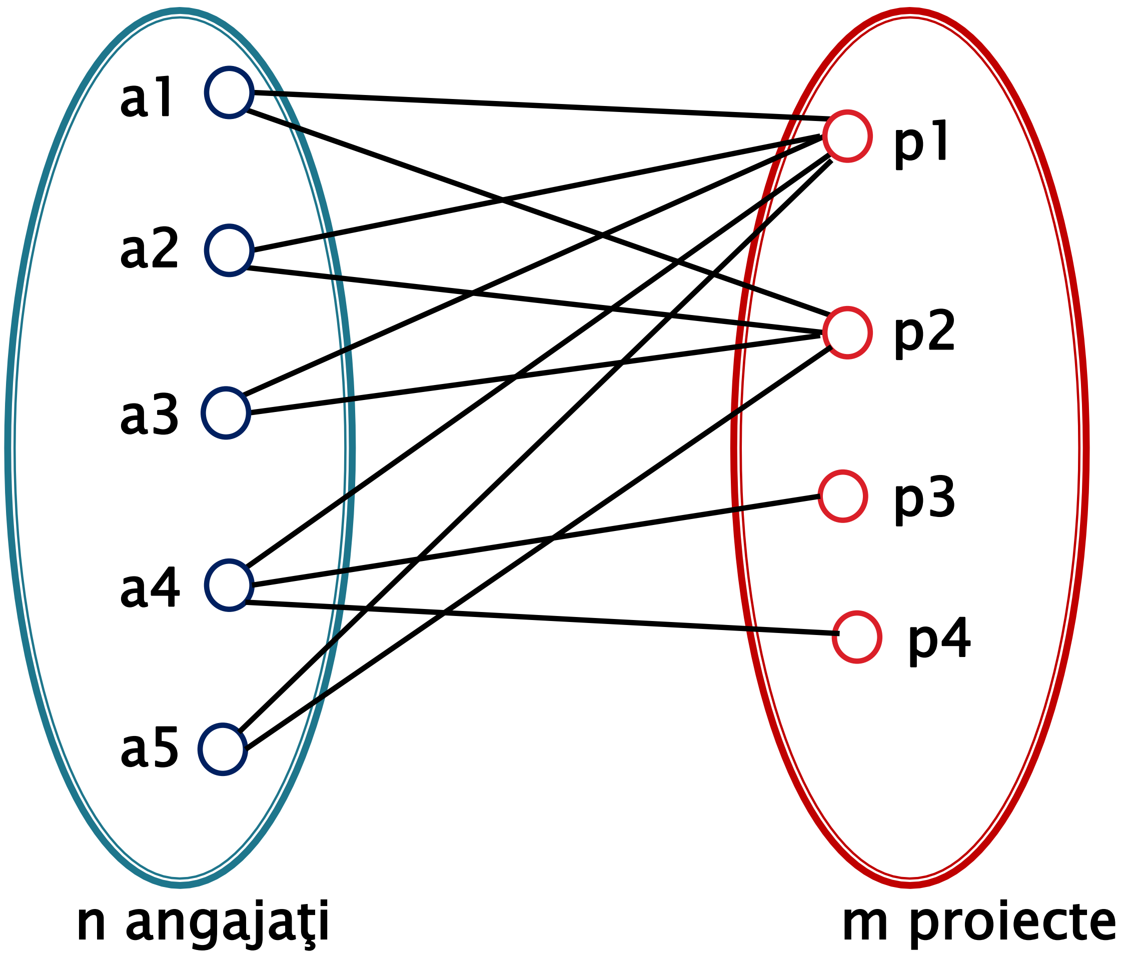
b1,f4
b2,f1
b3,f3
b4,f2



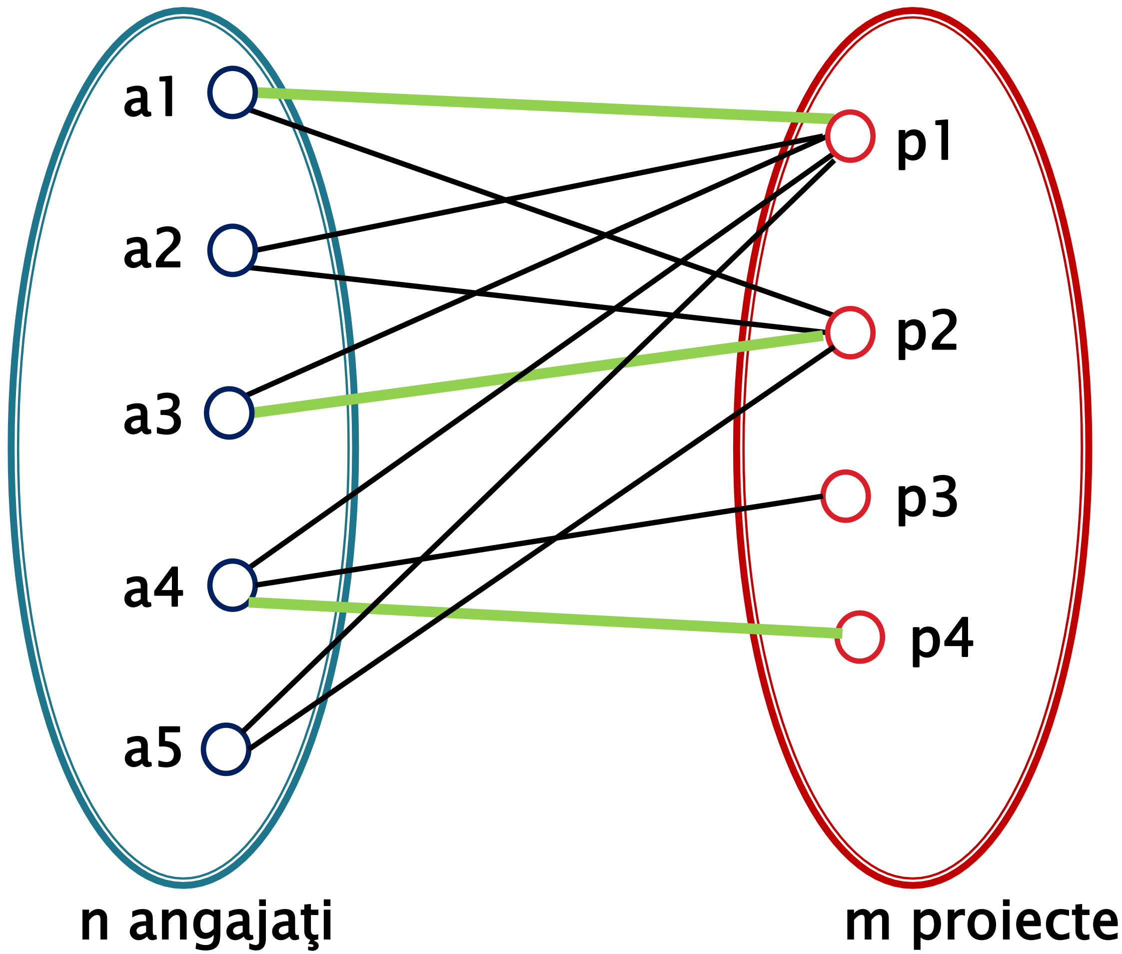
Repere istorice. Aplicaţii

- **Organizare de competiții**
- **Probleme de repartiţie**
  - lucrători – locuri de muncă
  - profesori – examene /conferinţe
    - **Problema orarului**

Alte aplicaţii

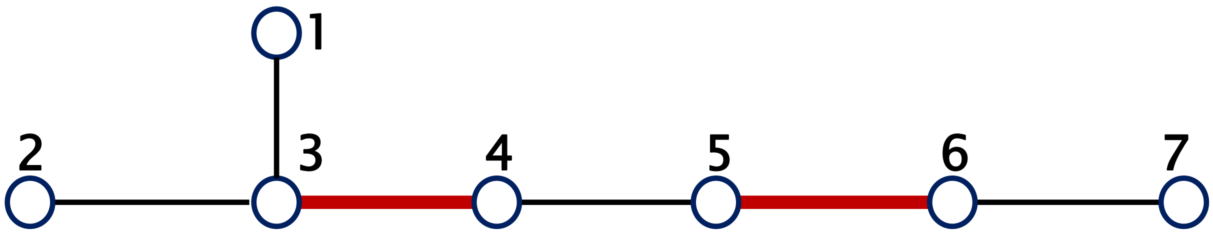


Alte aplicaţii



Fie G = (V, E) un graf şi **M ⊆ E**.

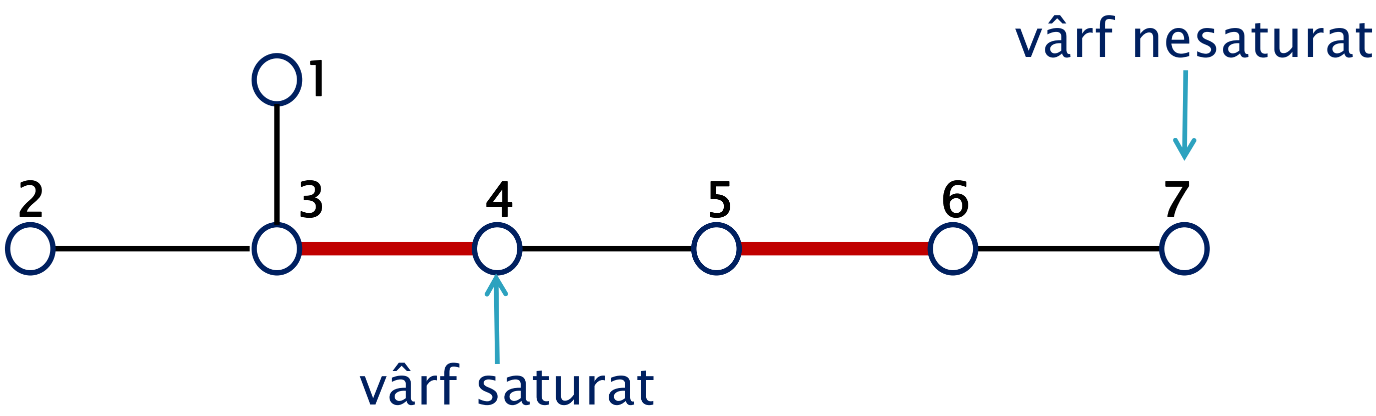
- M s.n **cuplaj** dacă orice două muchii din M sunt neadiacente



cuplaj M={ {3,4}, {5,6} }

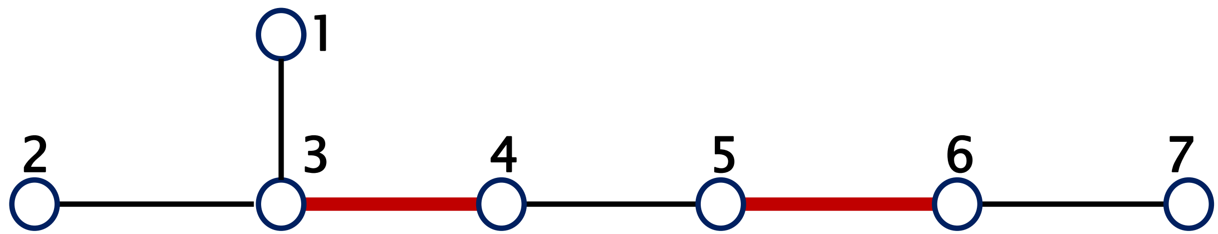
Fie G = (V, E) un graf şi **M ⊆ E.**

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

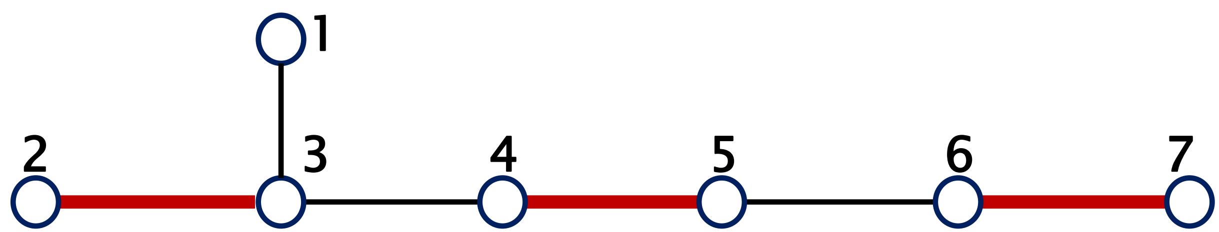


- Un cuplaj M^* s.n **cuplaj de cardinal maxim** (**cuplaj maxim**):

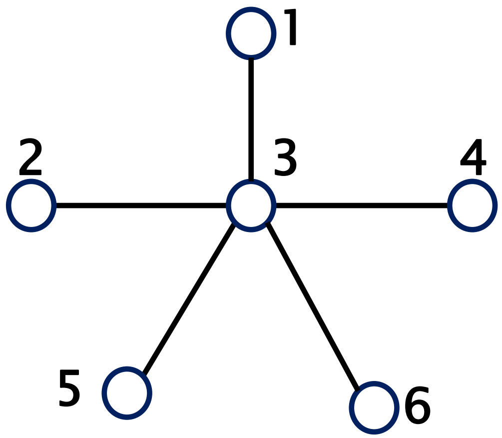
$$|M^*| \geq |M|, \forall M \subseteq E \text{ cuplaj}$$



cuplaj



cuplaj de cardinal maxim

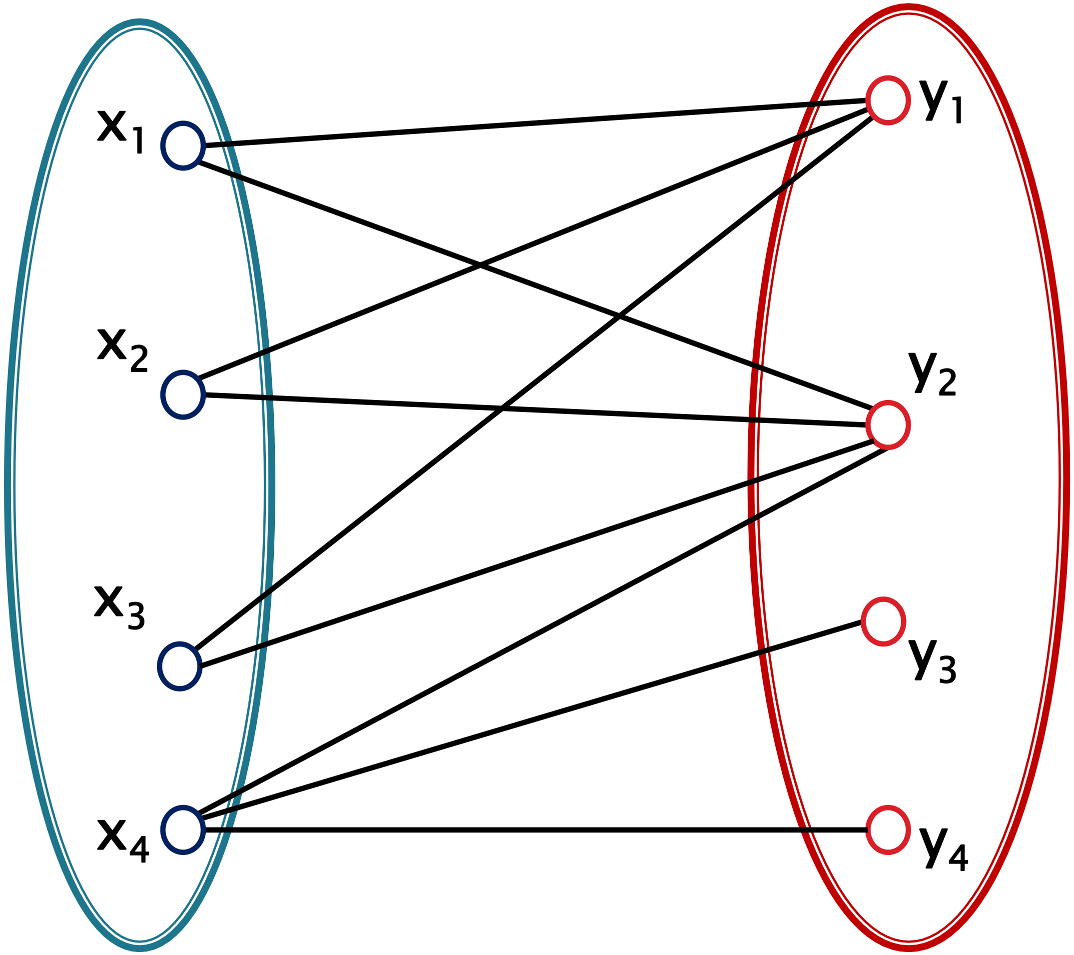


cuplaj de cardinal maxim ?

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

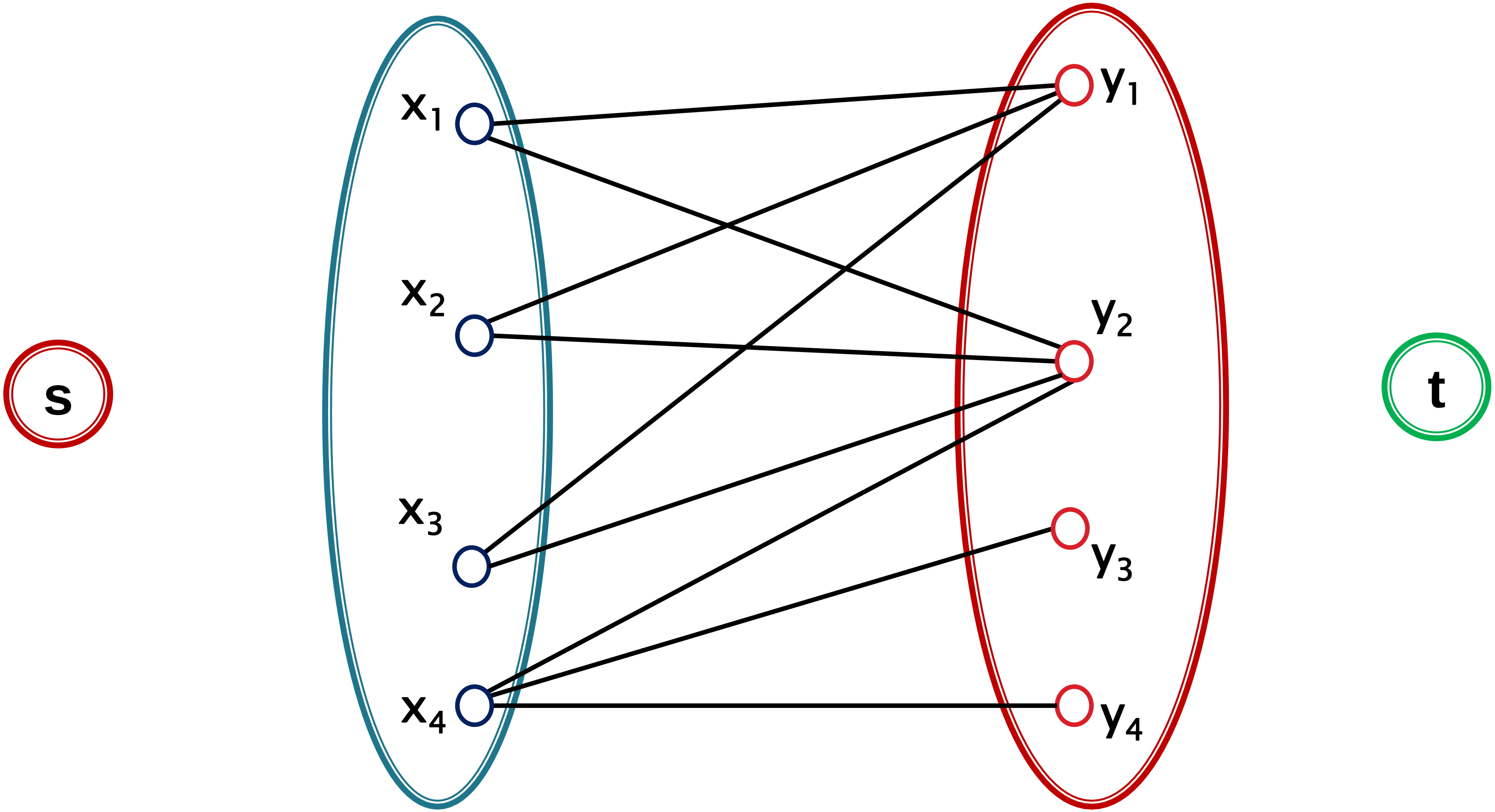
- **Algoritm de determinare a unui cuplaj maxim într-un graf bipartit**
  - Reducem problema determinării unui cuplaj maxim într-un cuplaj bipartit G la determinarea unui flux maxim într-o reţea de transport asociată lui G

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

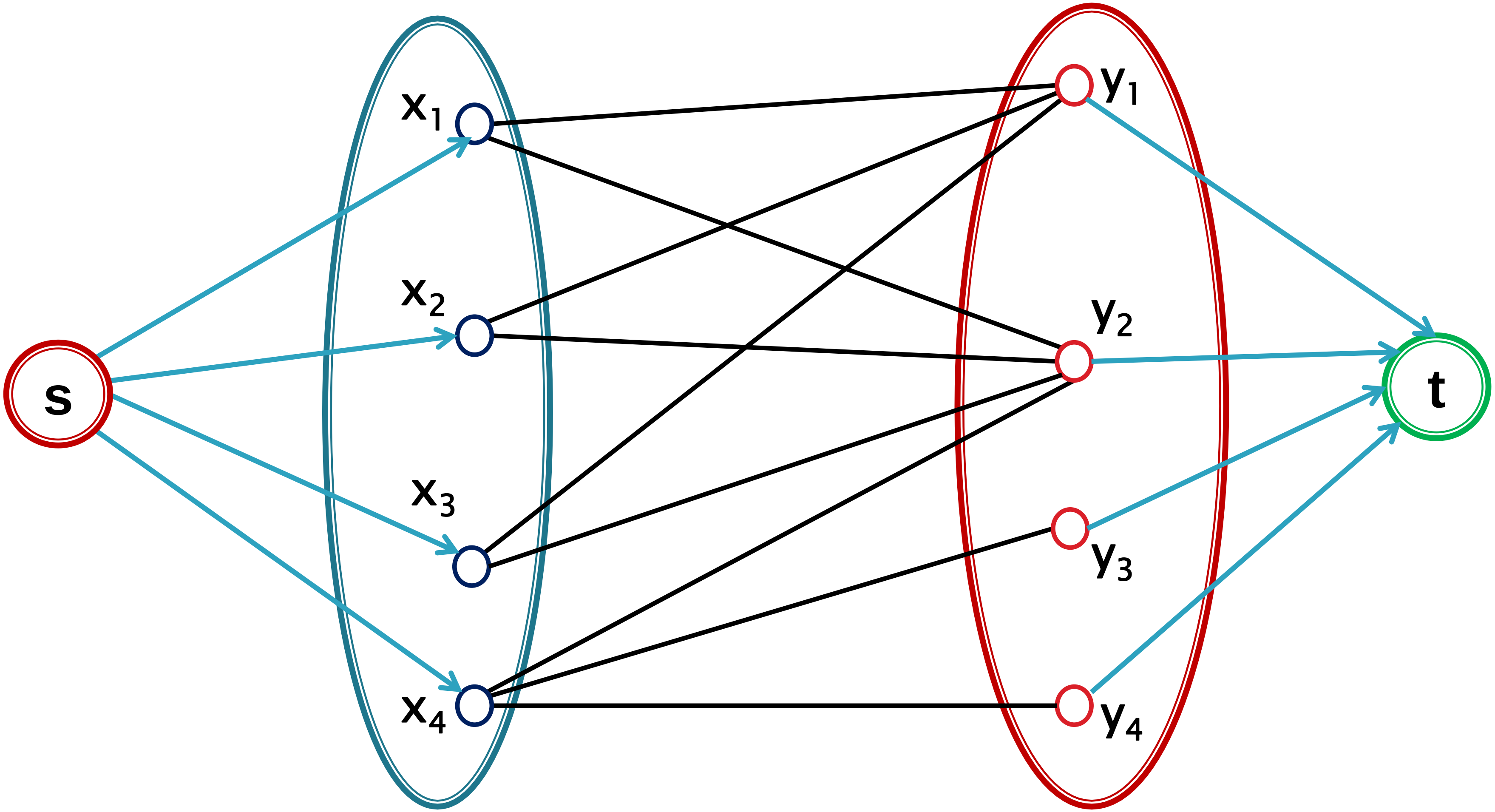


x_1
x_2
x_3
x_4
y_1
y_2
y_3
y_4

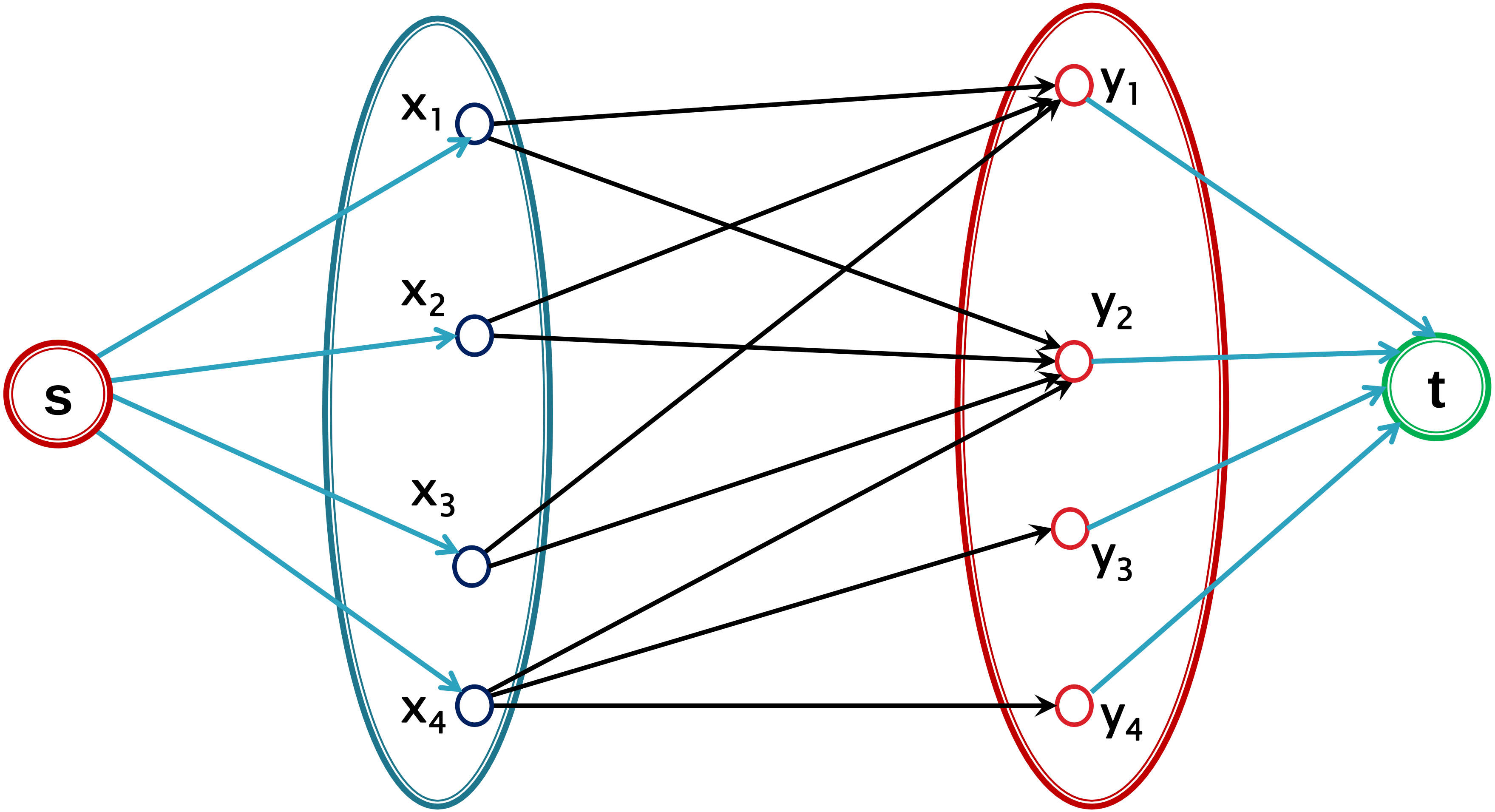
- Adăugăm două noduri noi s şi t



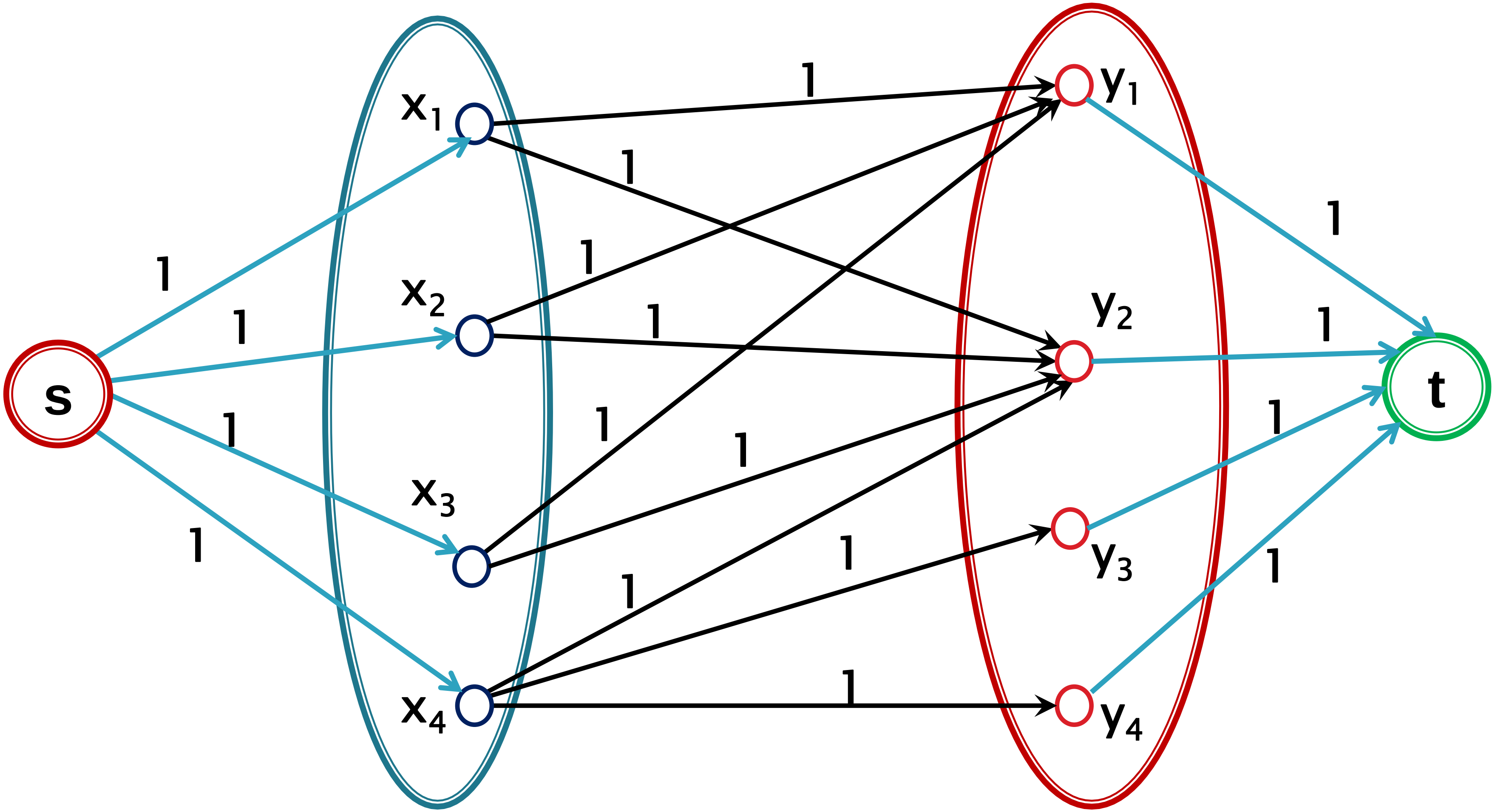
- Adăugăm arce (s,x_i), pentru $x_i \in X$ şi (y_j, t), $y_j \in Y$



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



- Asociem fiecărui arc capacitatea 1

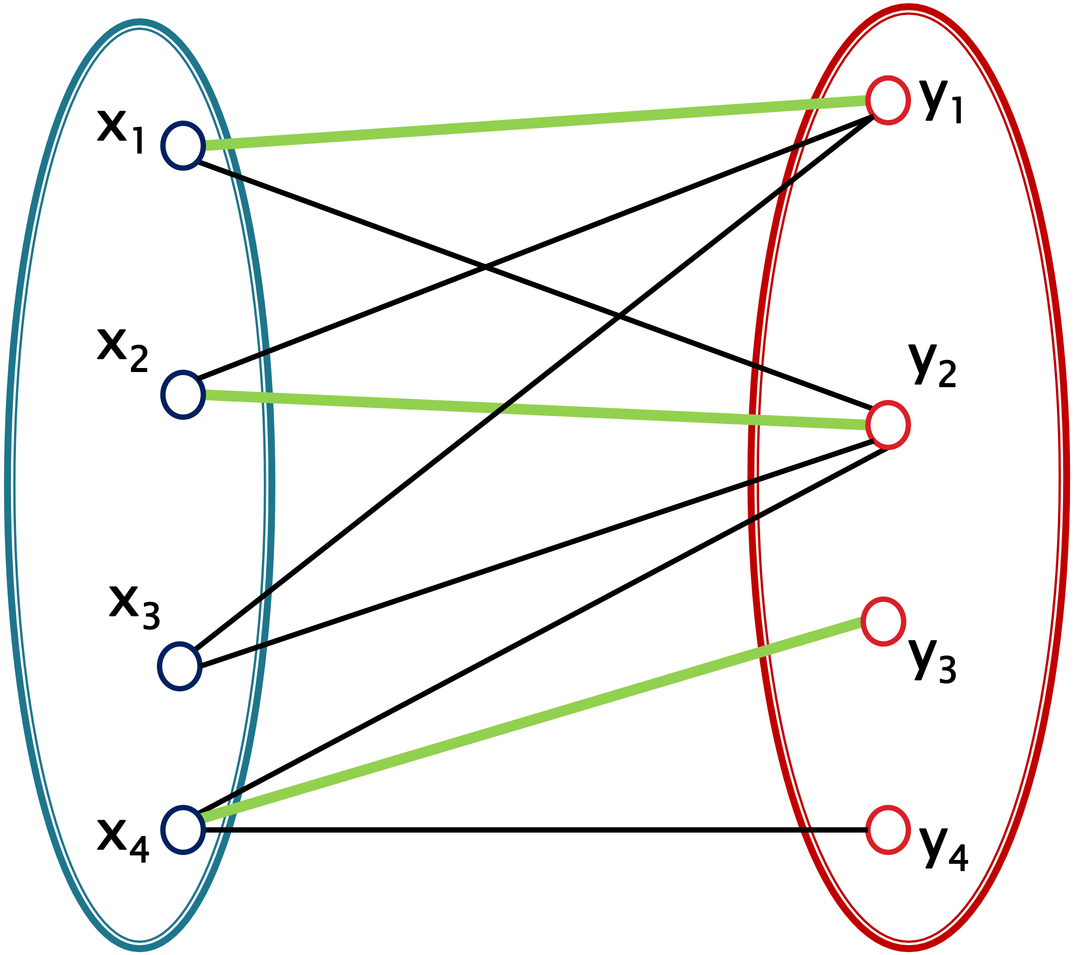


- **Proprietatea 1**

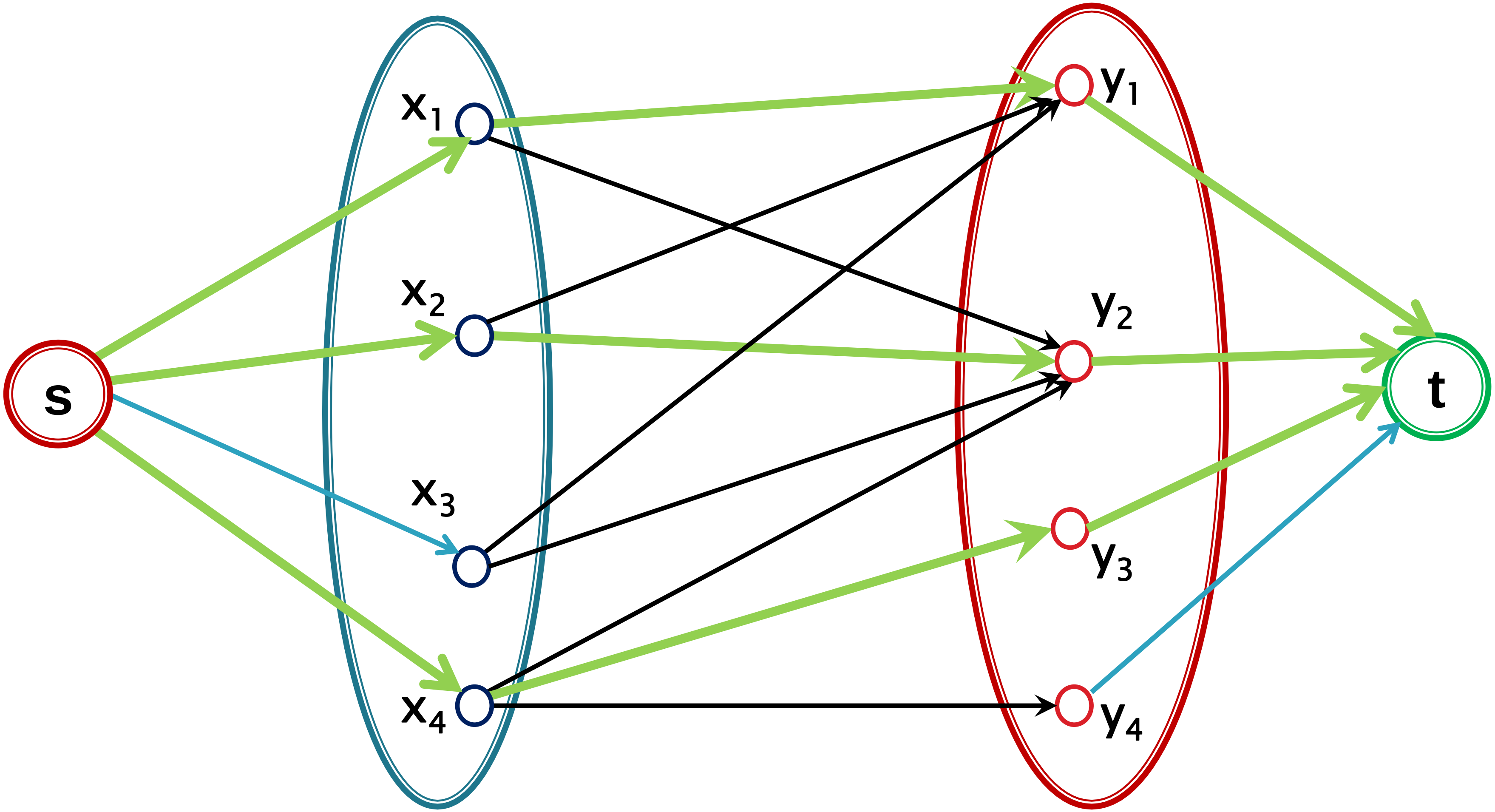
  Fie $G=(X \cup Y, E)$ un graf bipartit şi M un cuplaj în G. Atunci există un flux f în reţeaua de transport asociată N cu

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

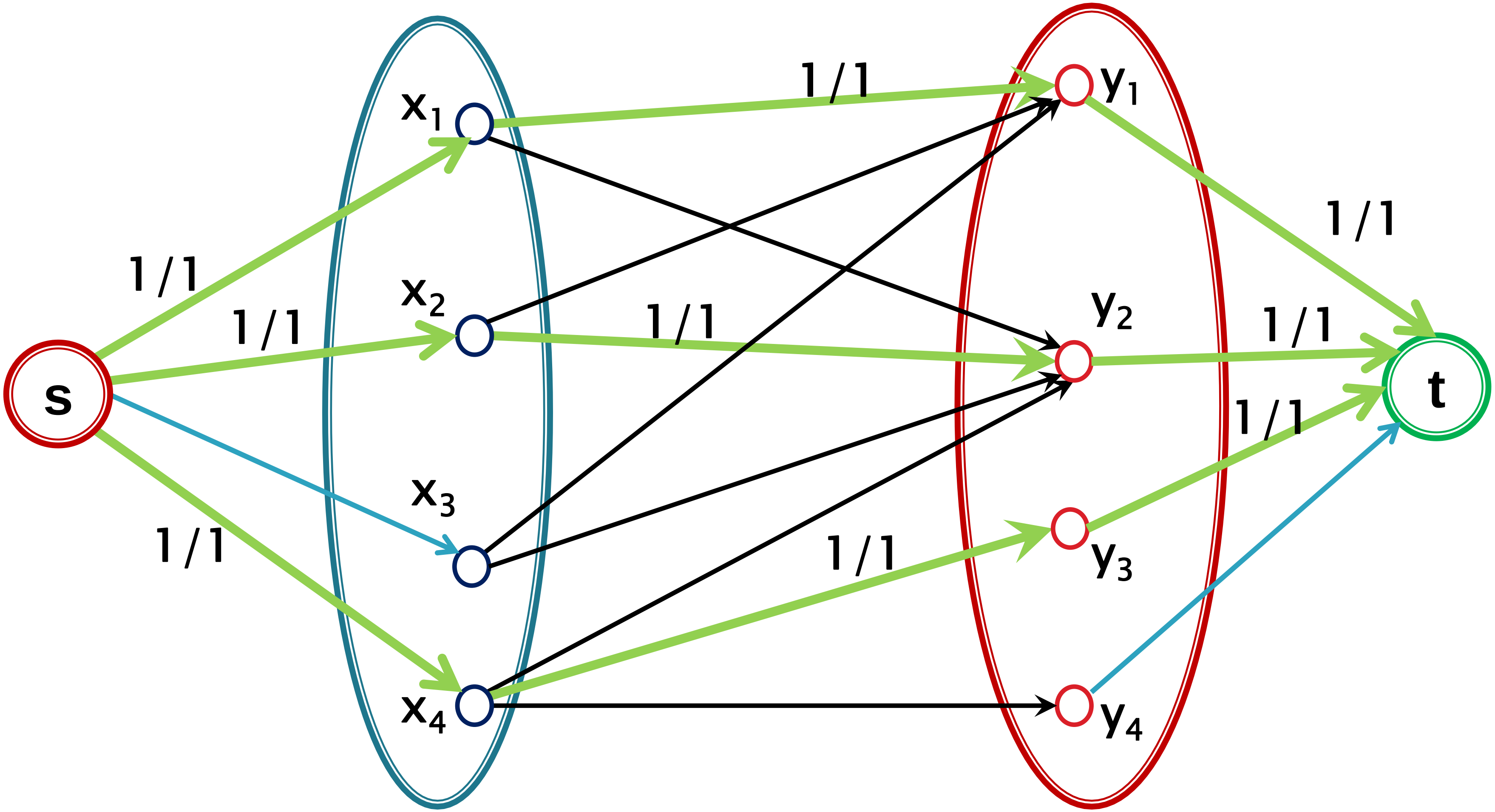
- **Cuplaj în graf $\Rightarrow$ flux în reţea**



- Cuplaj în graf $\Rightarrow$ flux în reţea

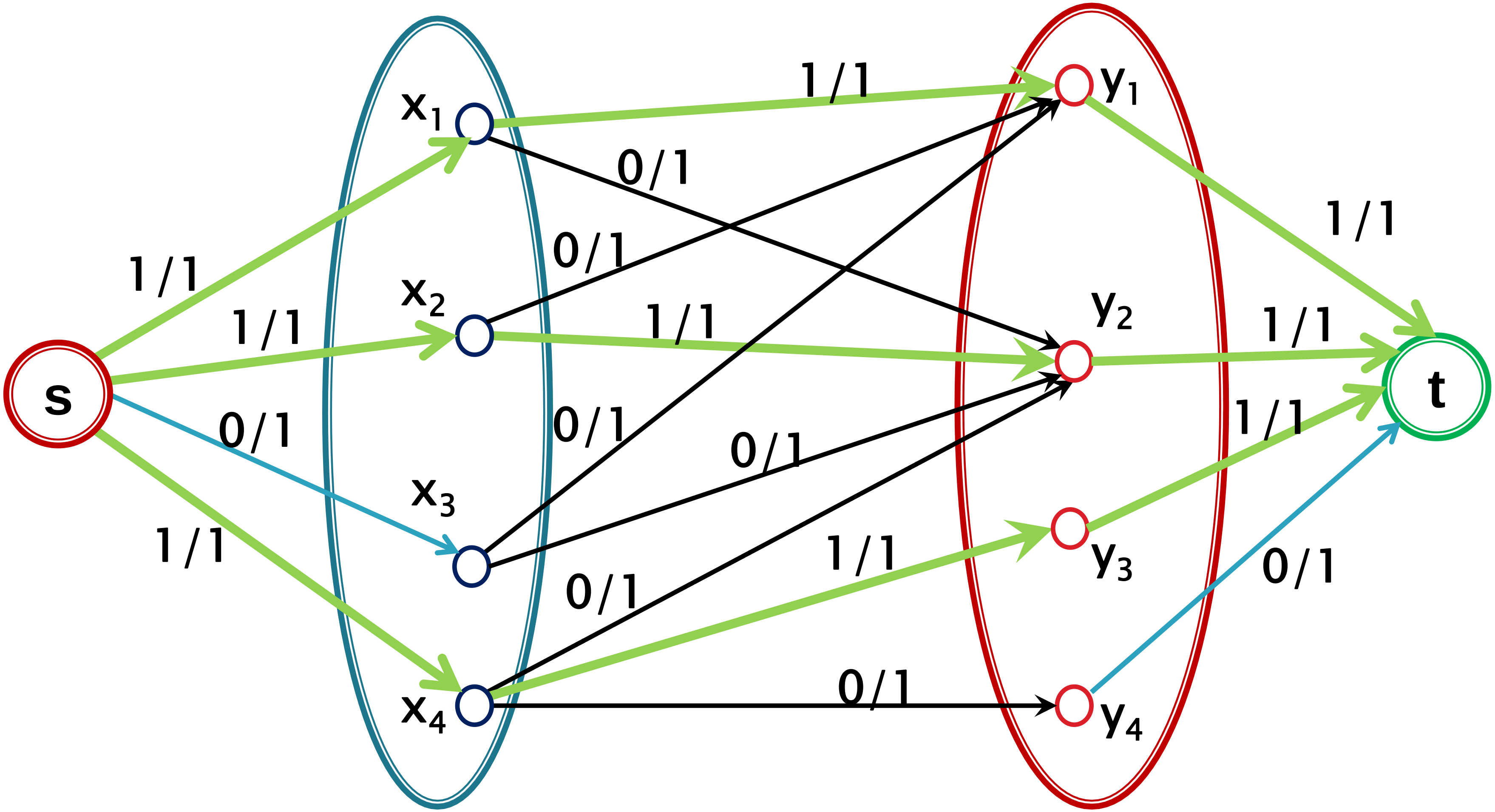


- Cuplaj în graf $\Rightarrow$ flux în reţea



Celelalte arce au flux 0 și capacitate 1

- Cuplaj în graf $\Rightarrow$ flux în reţea

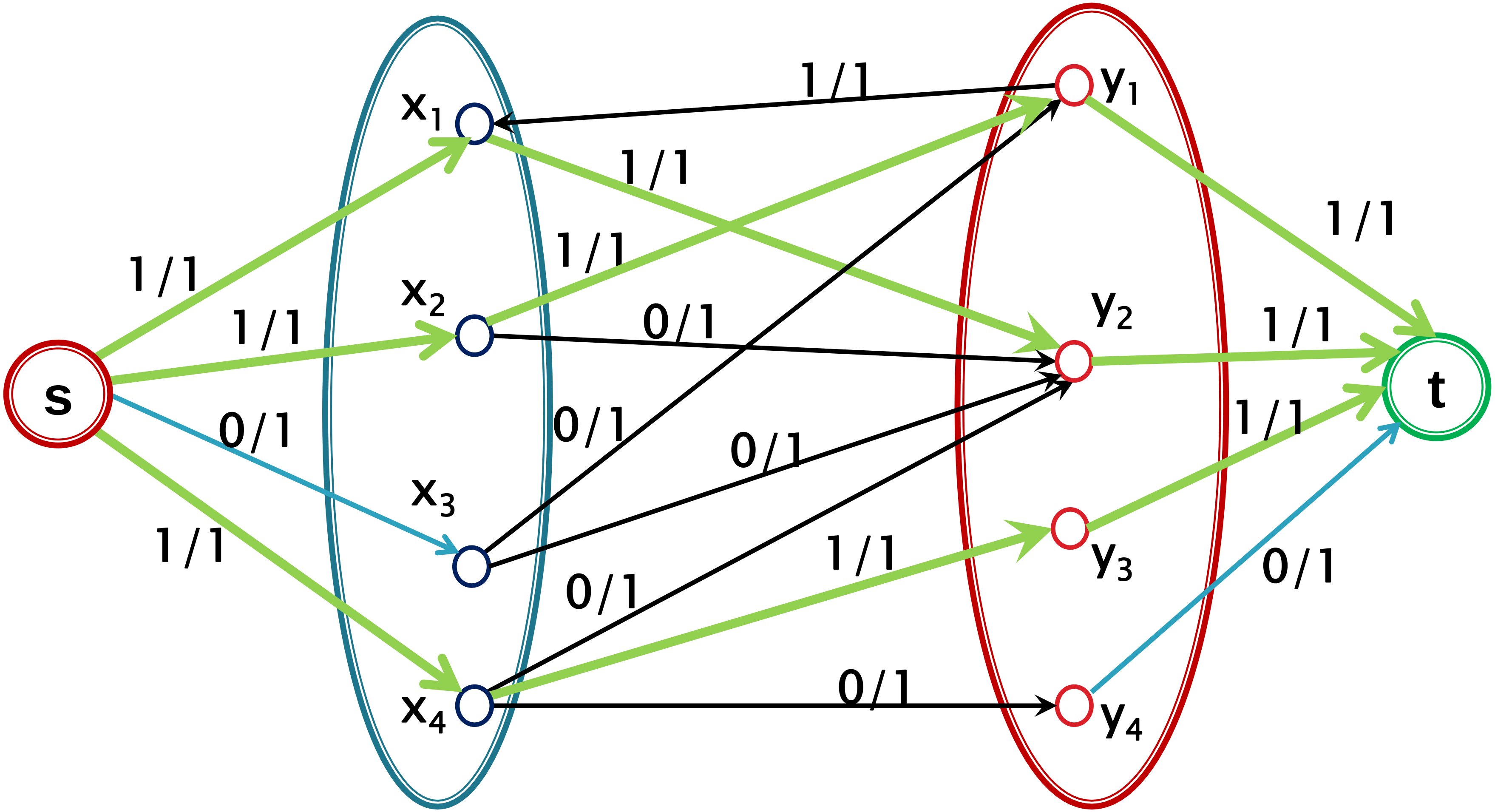


- **Proprietatea 2**

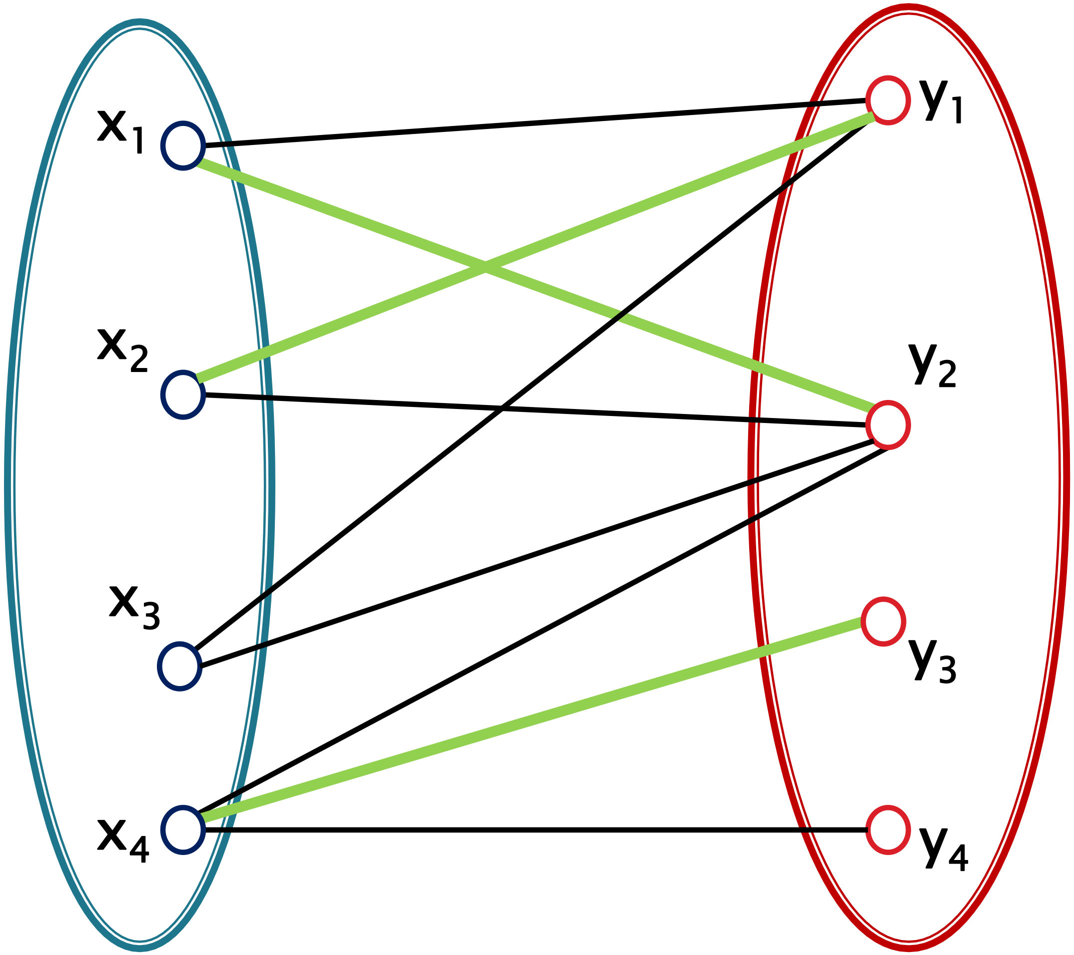
  Fie $G=(X\cup Y, E)$ un graf bipartit şi f un flux în reţeaua de transport N asociată. Atunci există M un cuplaj în G cu

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

- Flux în reţea ⇒ cuplaj în graf

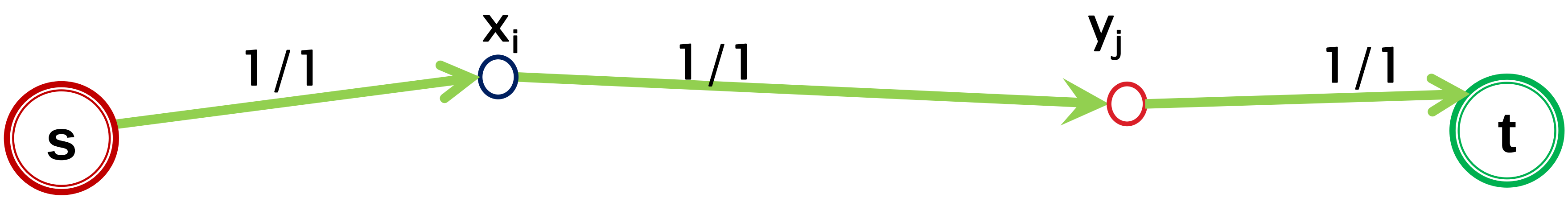


- Flux în reţea ⇒ cuplaj în graf

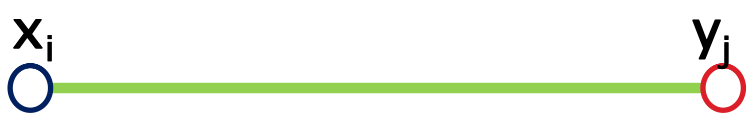


- Concluzie: Flux în reţea $\Leftrightarrow$ cuplaj în graf

Drum cu o unitate de flux



Muchie în cuplaj



Consecinţă

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 reţeaua 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 reţeaua 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 reţeaua 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 reţeaua 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 reţeaua 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: $L= c^+(s) \le n \quad \Rightarrow \quad O(mn)$

Aplicaţie
Construcţia unui graf orientat din secvenţele de grade

- Se dau secvenţele
  - $s_0^+ = \{d_1^+, \ldots, d_n^+\}$
  - $s_0^- = \{d_1^-, \ldots, d_n^-\}$

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

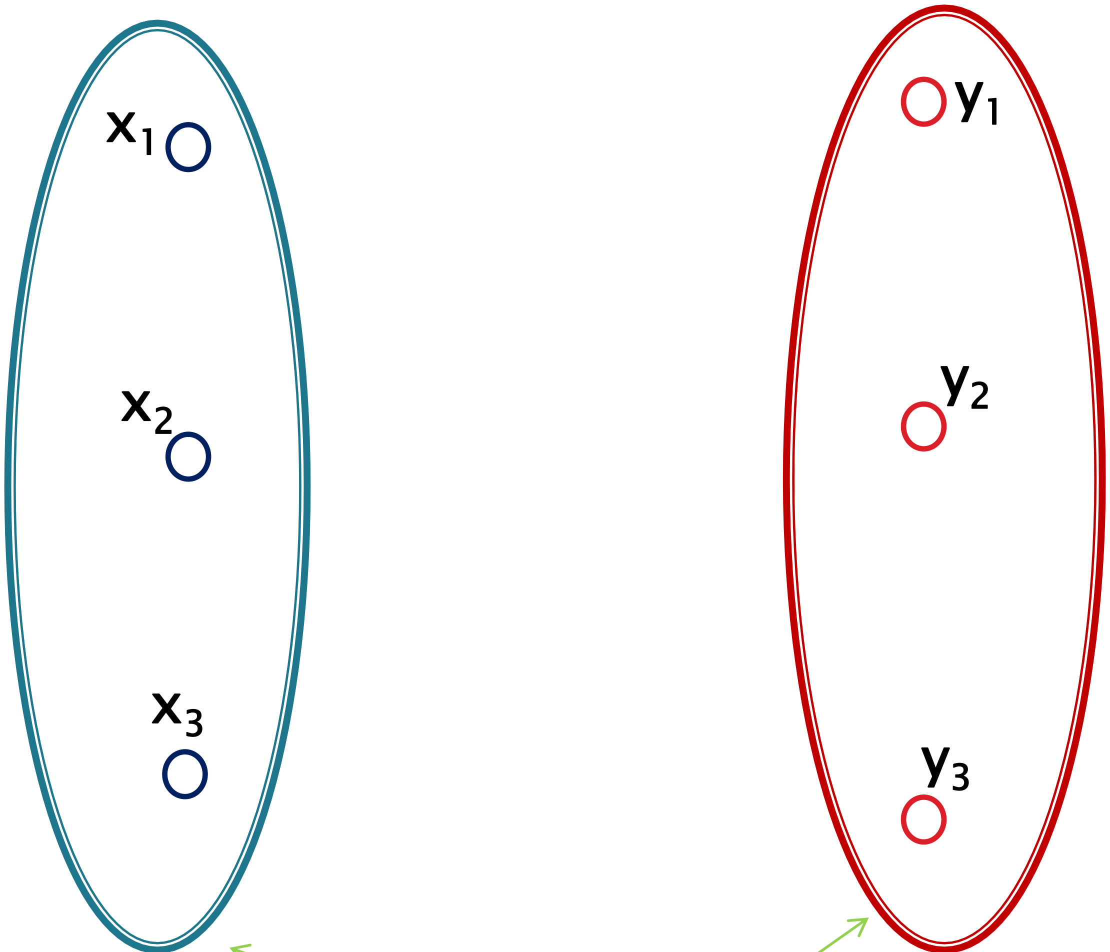
- Se dau secvenţele
  - $s_0^+ = \{d_1^+, \ldots, d_n^+\}$
  - $s_0^- = \{d_1^-, \ldots, d_n^-\}$

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

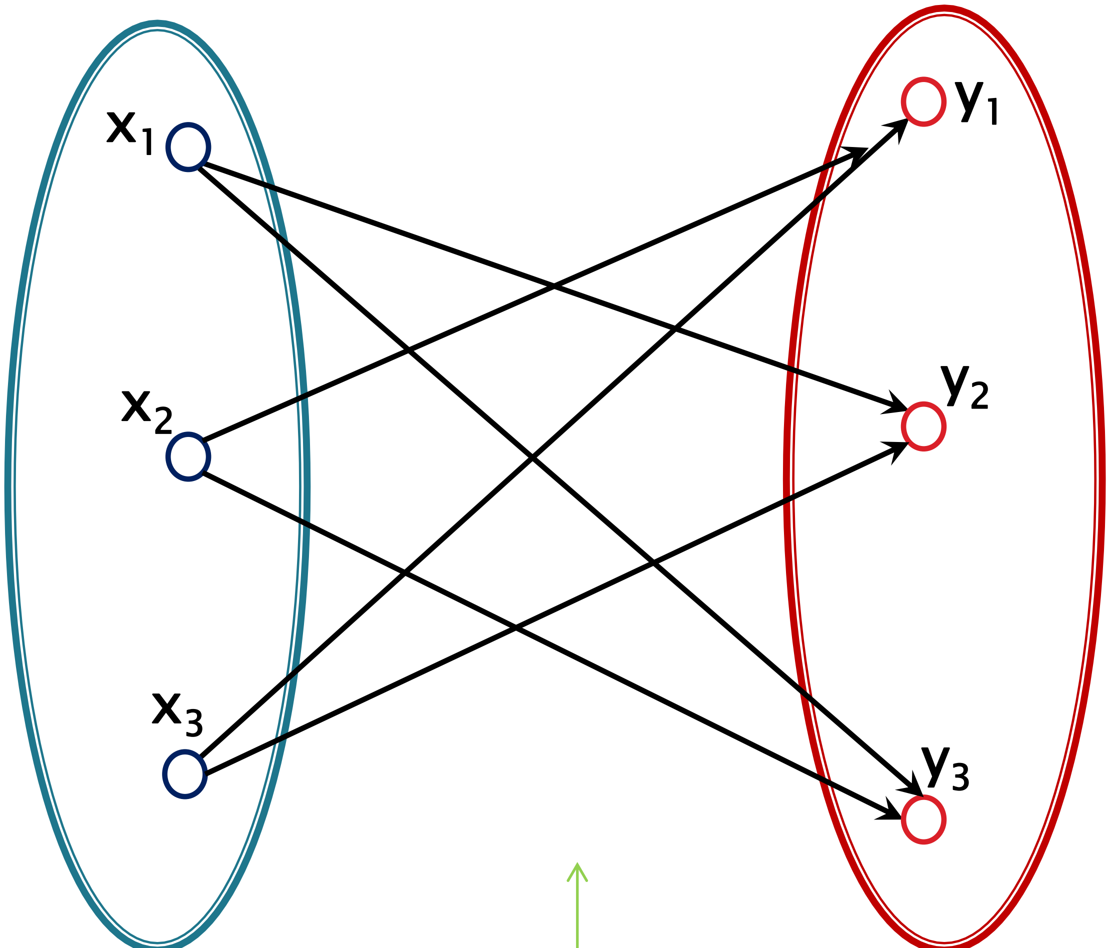
- **Exemplu**
  - $s_0^+ = \{1, 0, 2\}$
  - $s_0^- = \{1, 1, 1\}$

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

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

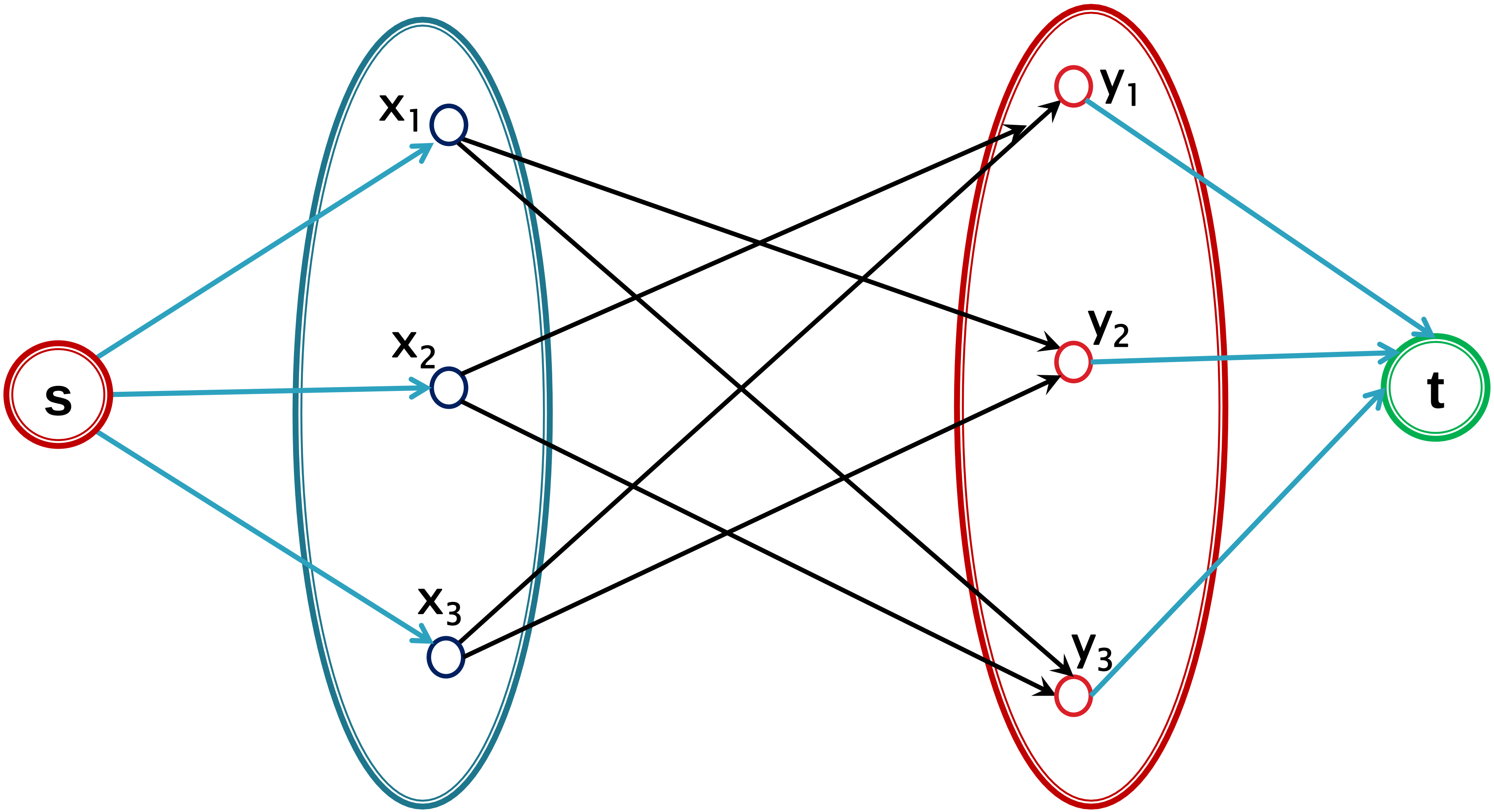


Vârfurile 1, 2,..., n se pun în ambele clase ale bipartiţiei (câte o copie)

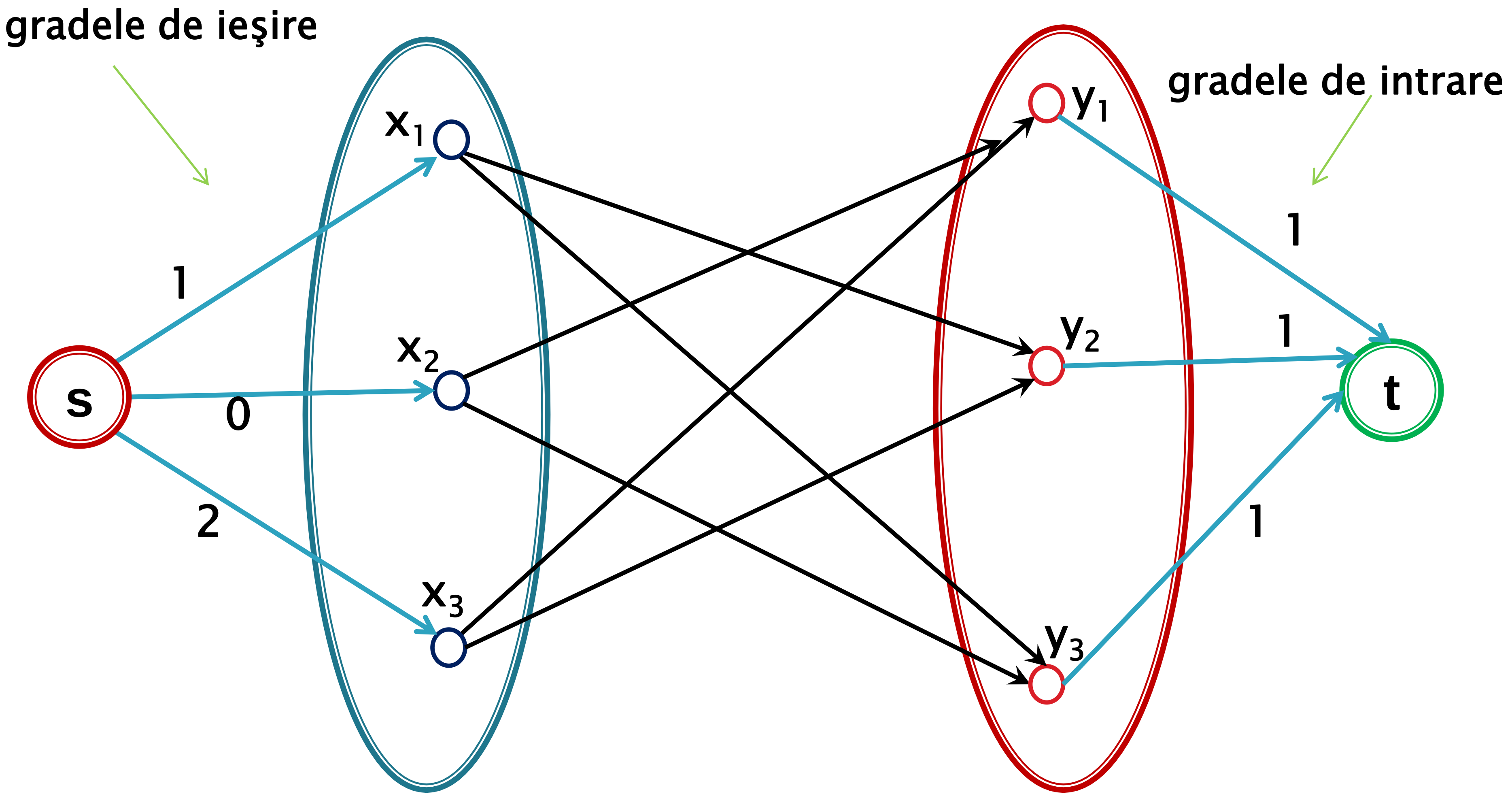


arce $x_i y_j$ cu $i \neq j$

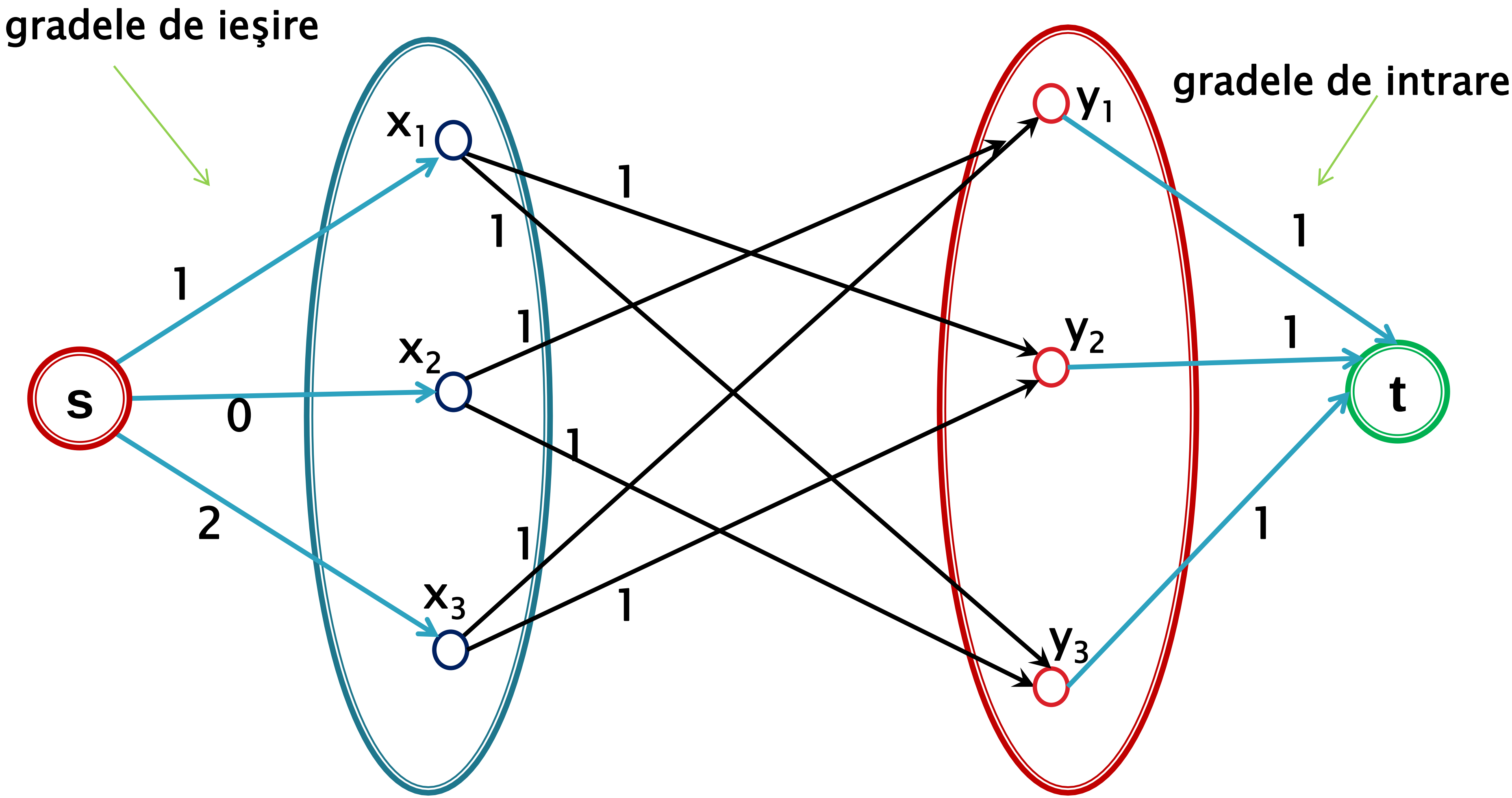
(fluxul pe arcul $x_i y_j$ va fi nenul $\Leftrightarrow$ $ij \in E(G)$)



s
x1
x2
x3
y1
y2
y3
t



gradele de ieşire
gradele de intrare
s
t
x_1
x_2
x_3
y_1
y_2
y_3
1
0
2
1
1
1


gradele de ieşire
gradele de intrare
s
t
x_1
x_2
x_3
y_1
y_2
y_3
1
0
2
1
1
1
1
1
1
1
1
1

- **Proprietate**

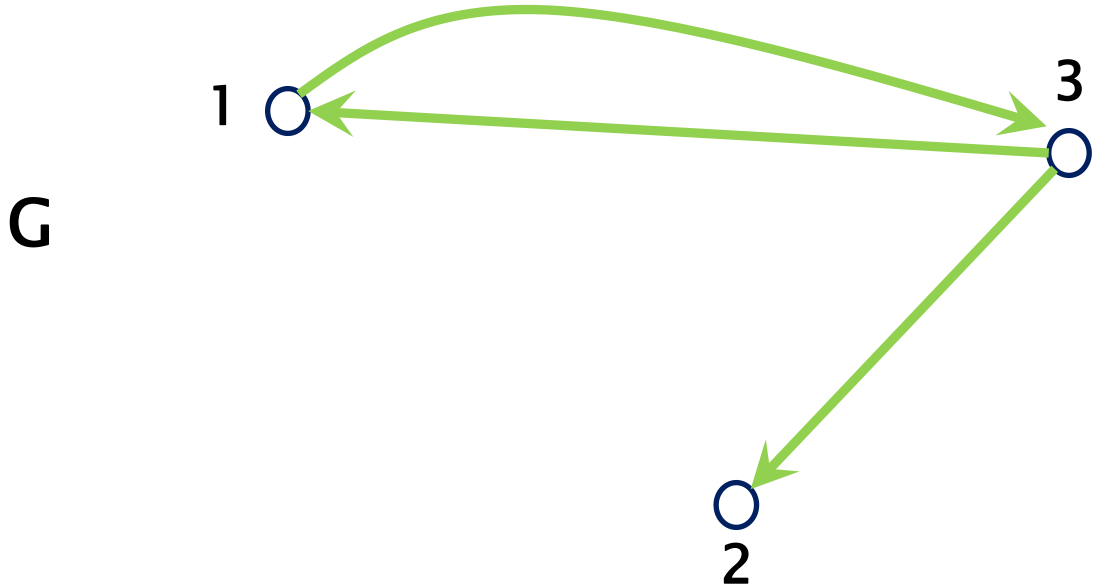
Există graf cu secvenţele date $\Leftrightarrow$ în graful asociat **fluxul de valoare maximă** are

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

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

tăieturile ({s},V−{s}) , (V−{t}, {t}) sunt minime

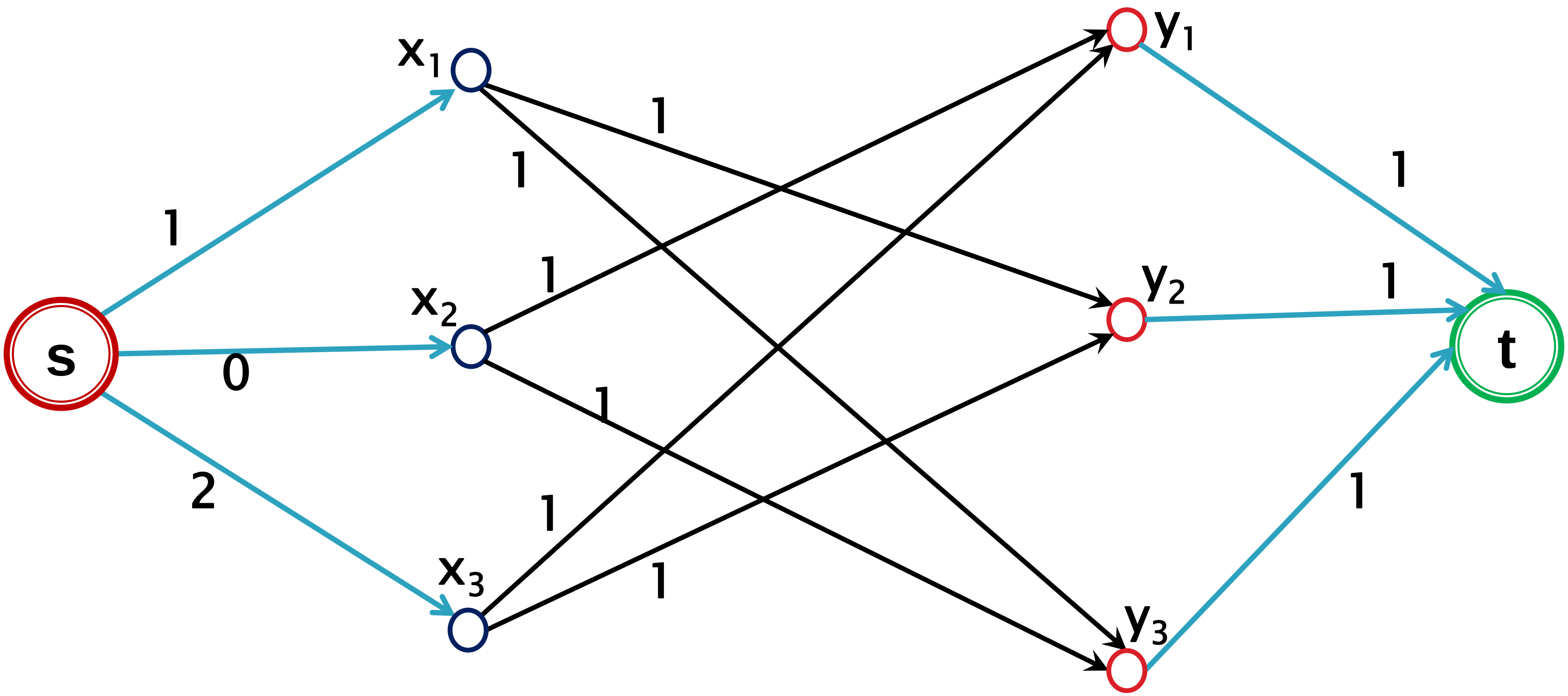
G ⇒ flux în reţea

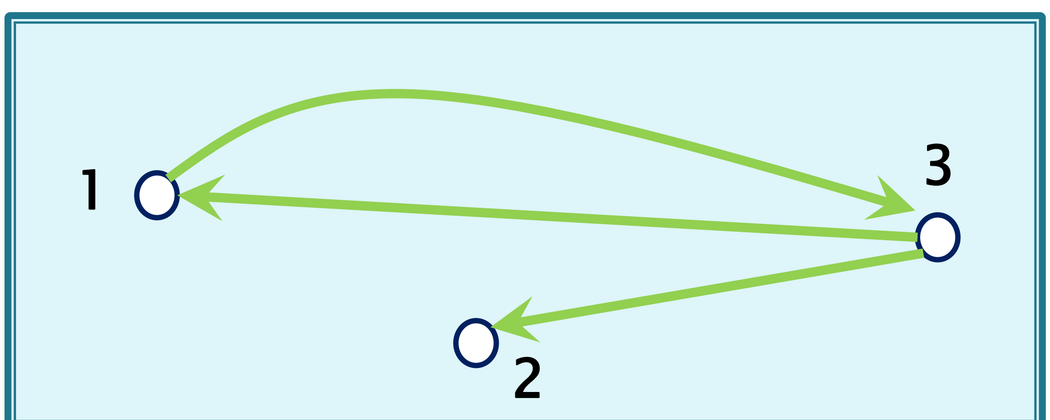


$$s_0^+ = \{1,\ 0,\ 2\}$$

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

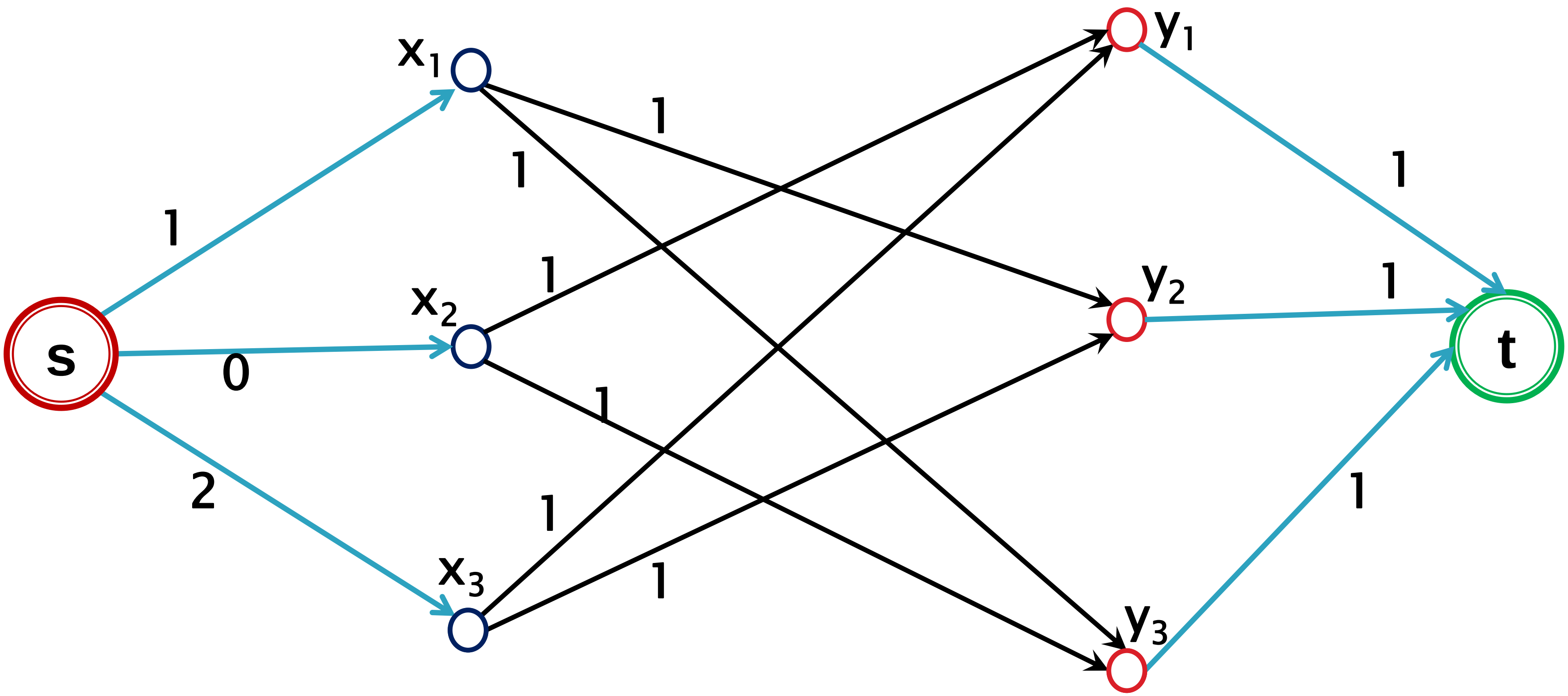
G ⇒ flux în reţea

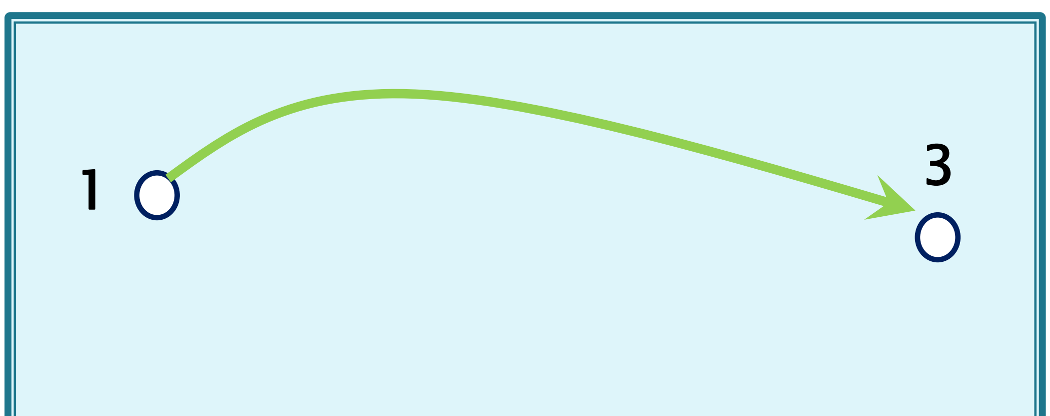




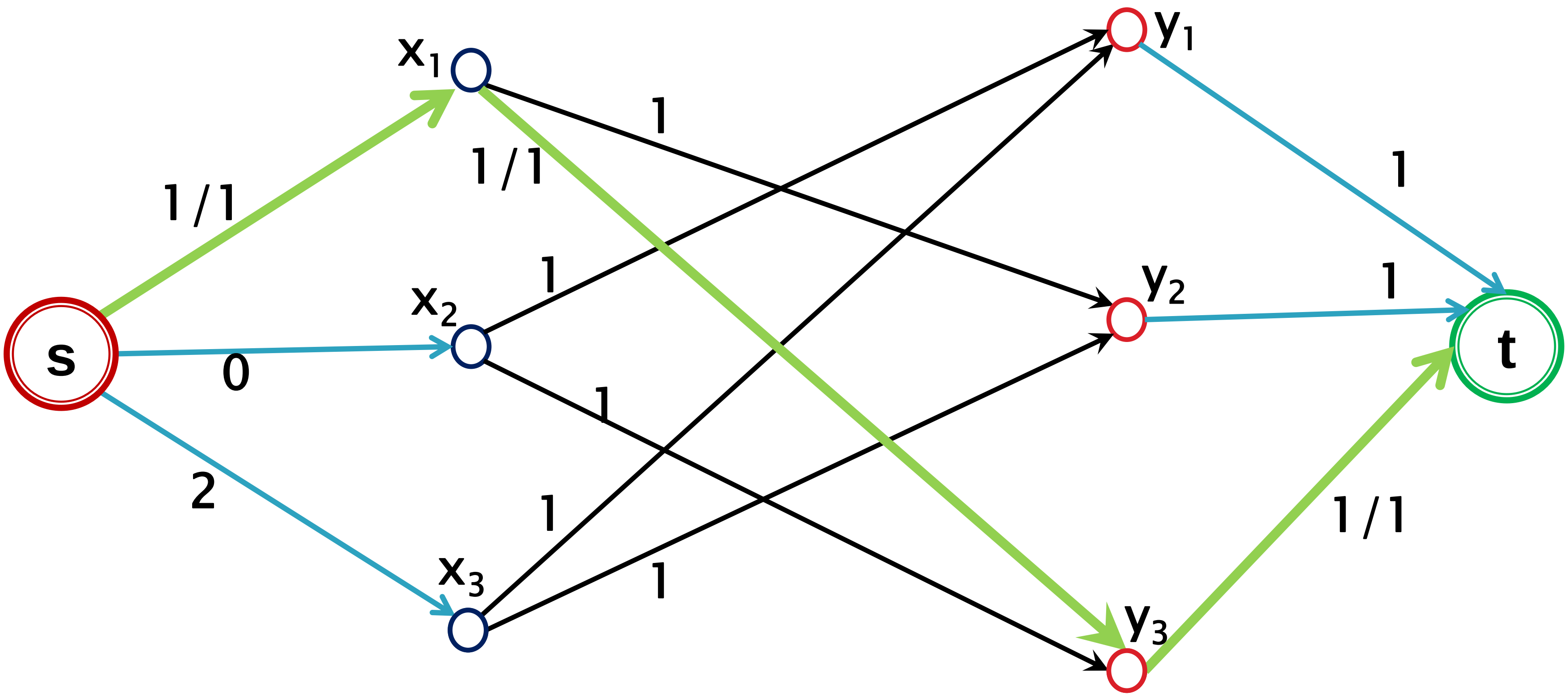
G

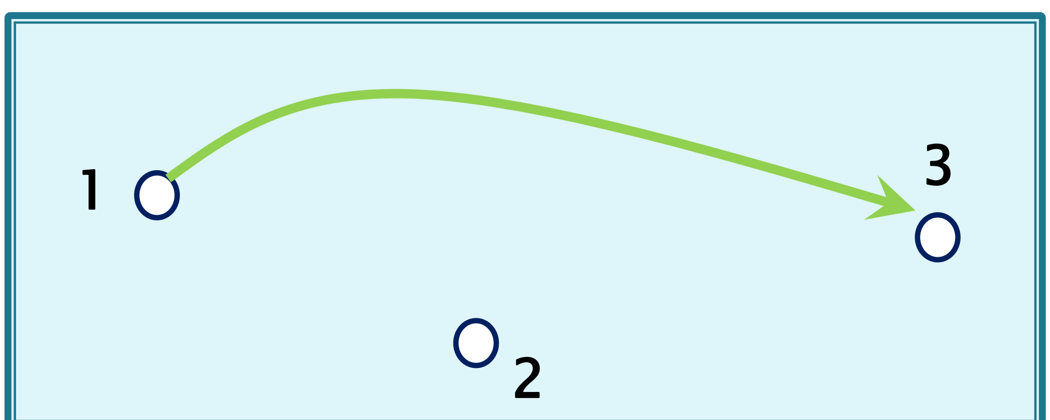
$G \Rightarrow$ flux în reţea



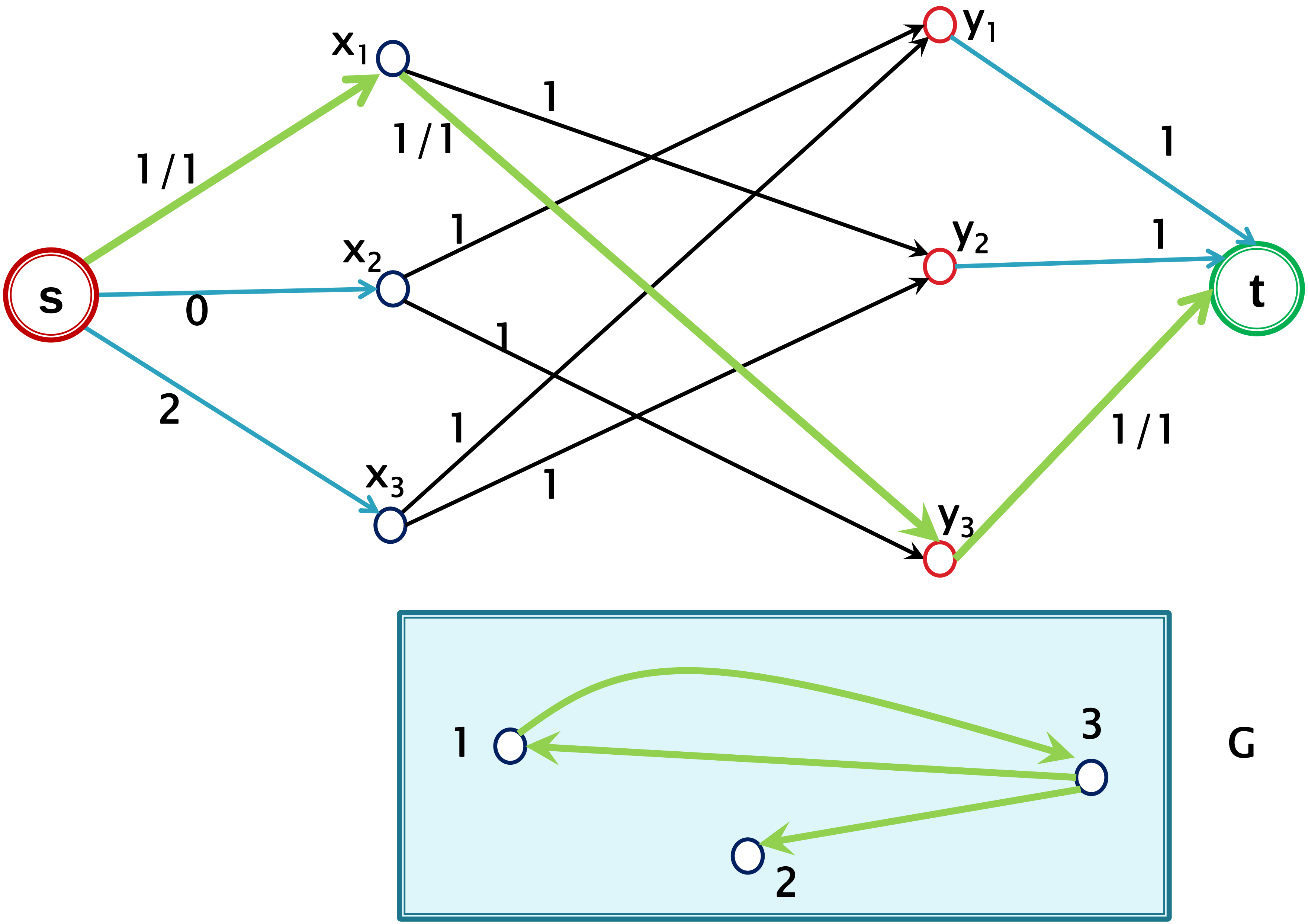


G ⇒ flux în reţea



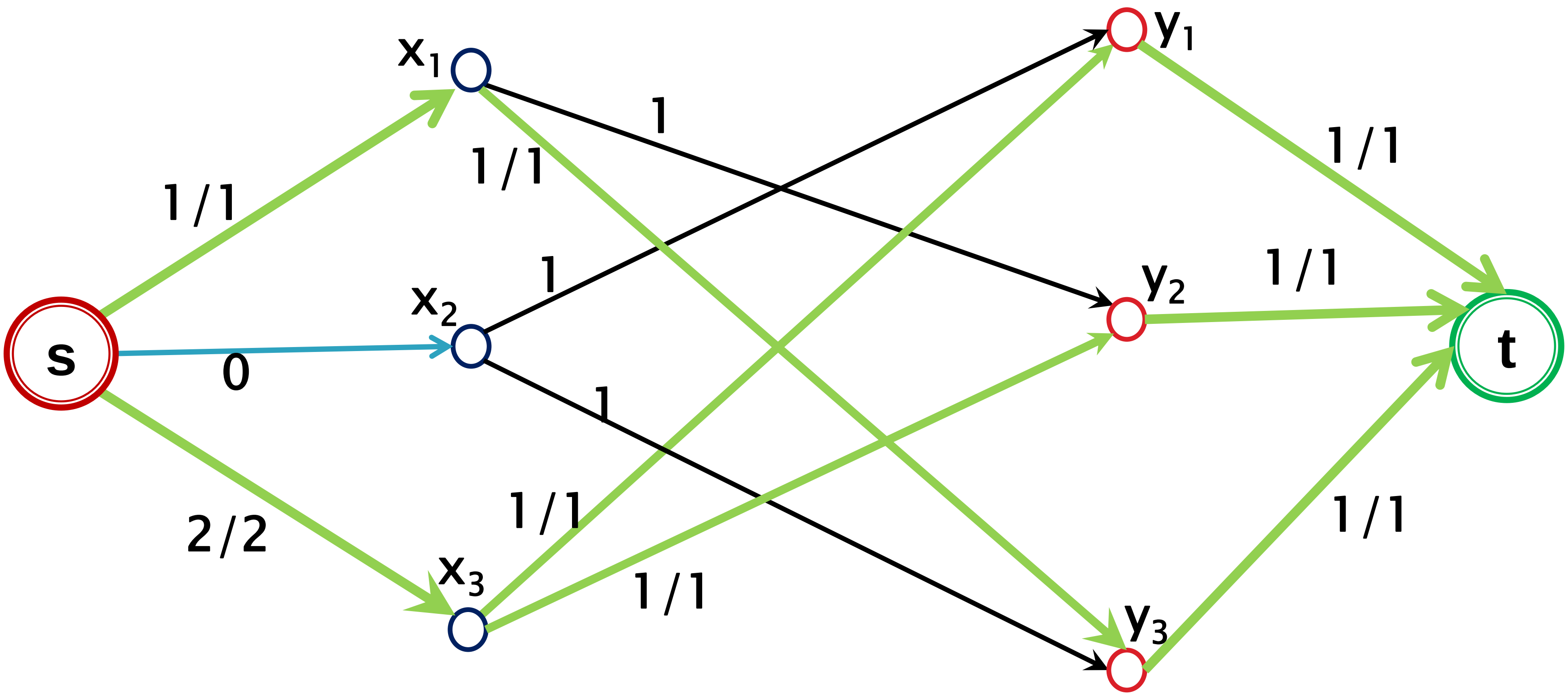


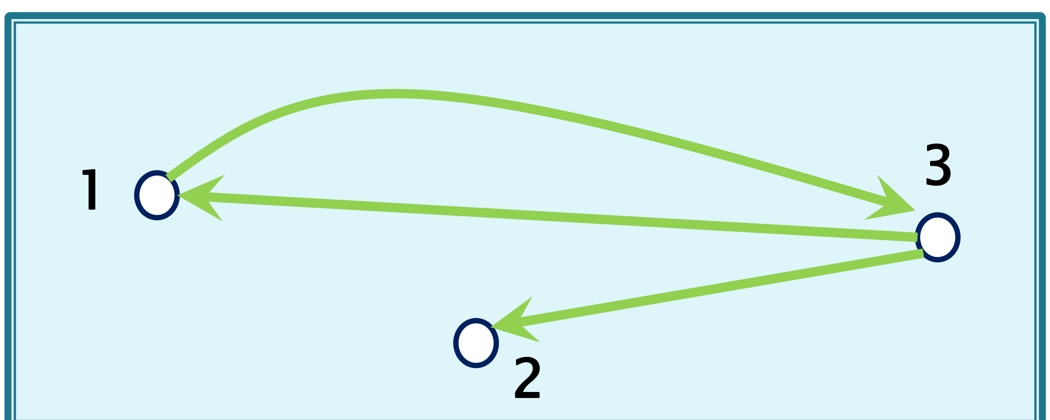
$G \Rightarrow$ flux în reţea



G

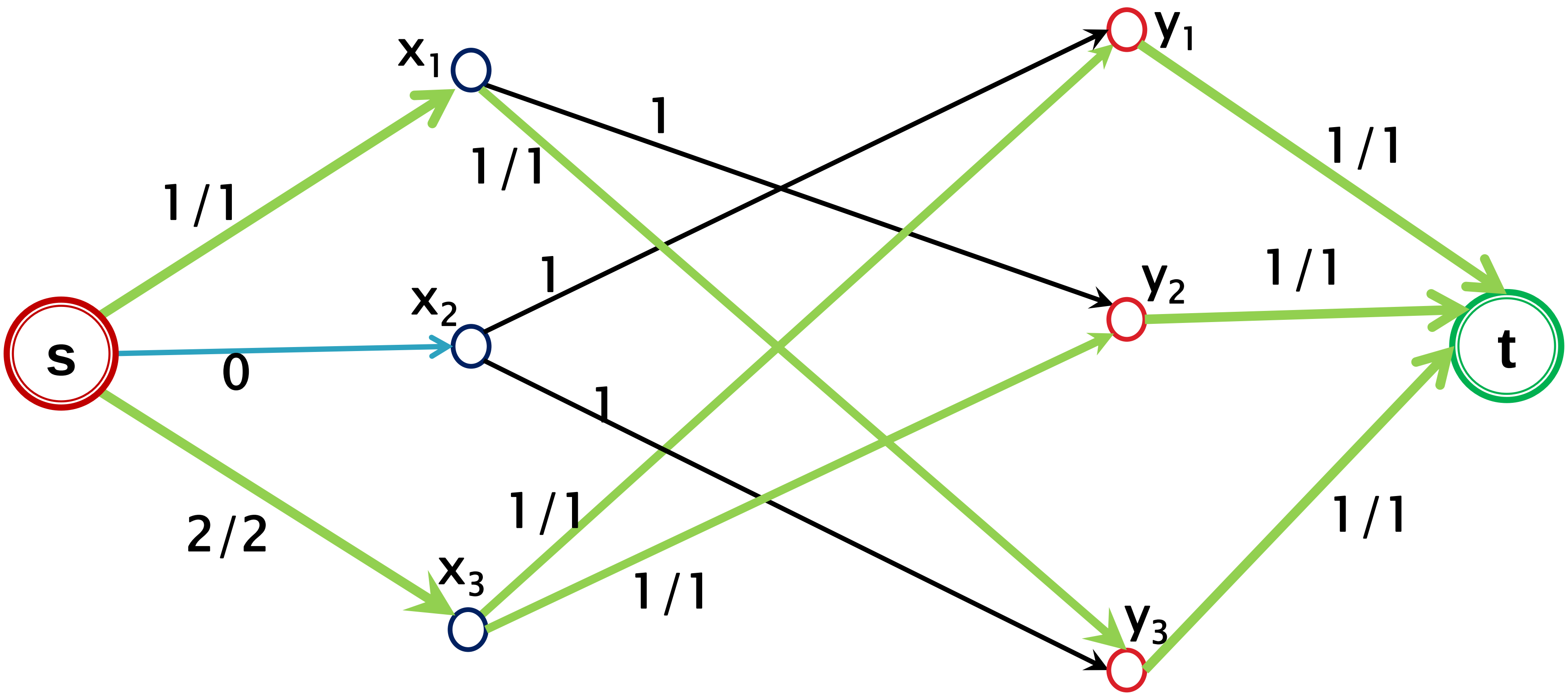
$G \Rightarrow$ flux în reţea





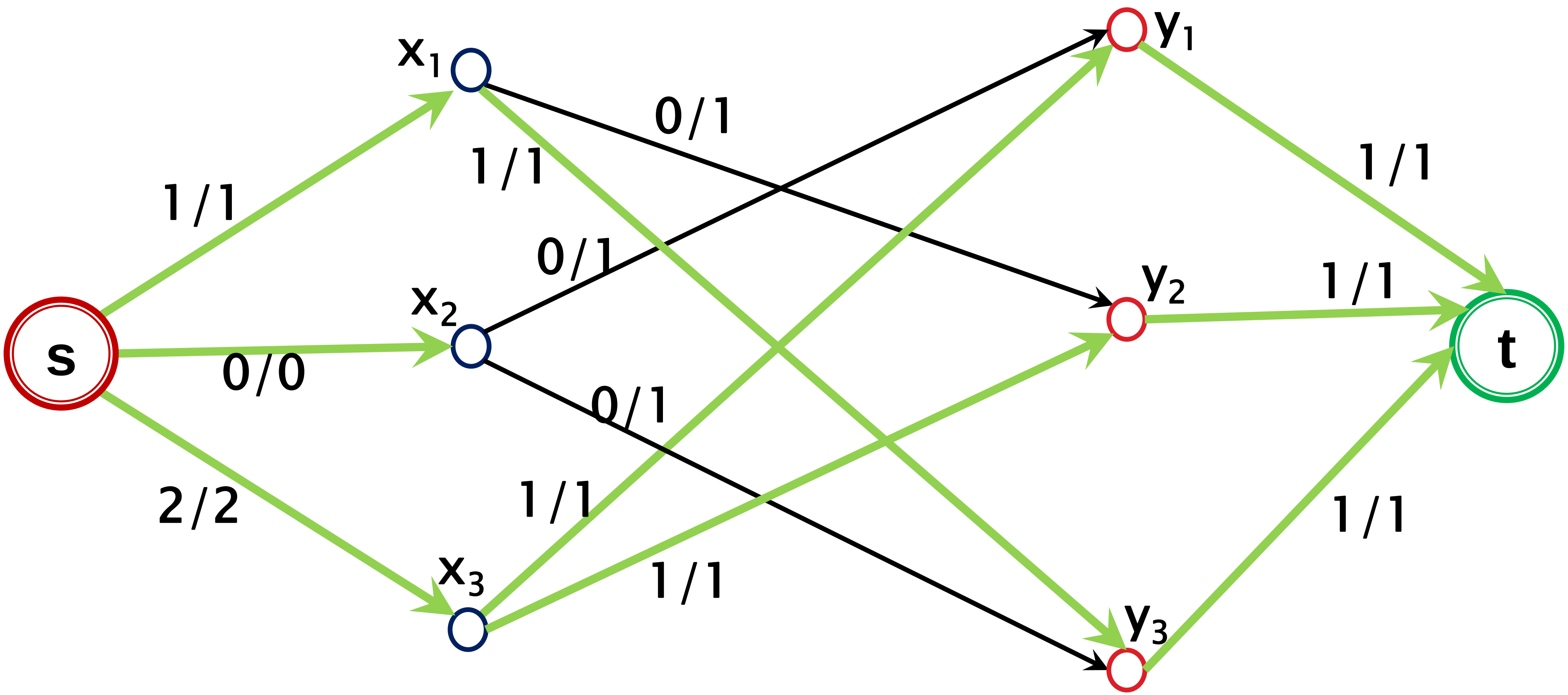
G

G ⇒ flux în reţea



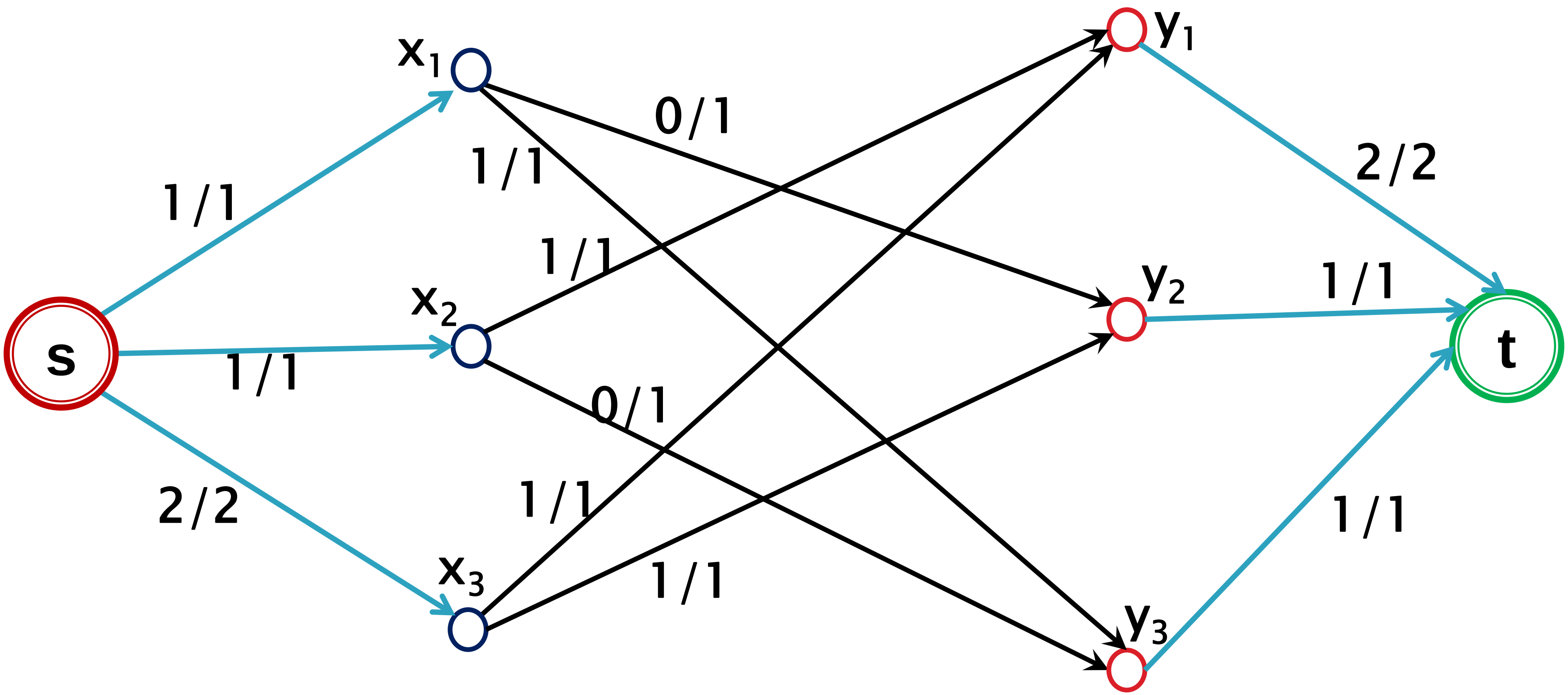
Restul arcelor au fluxul 0

G ⇒ flux în reţea

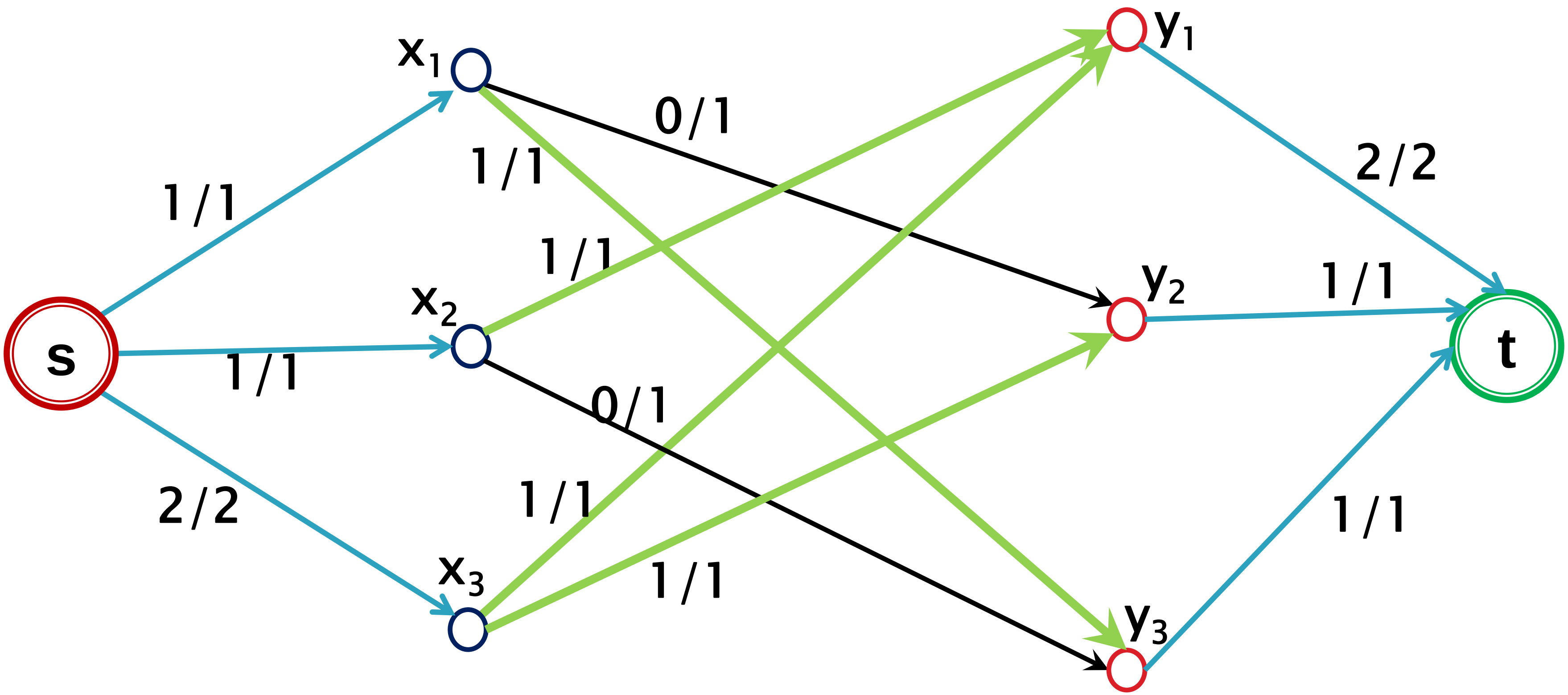


Reciproc

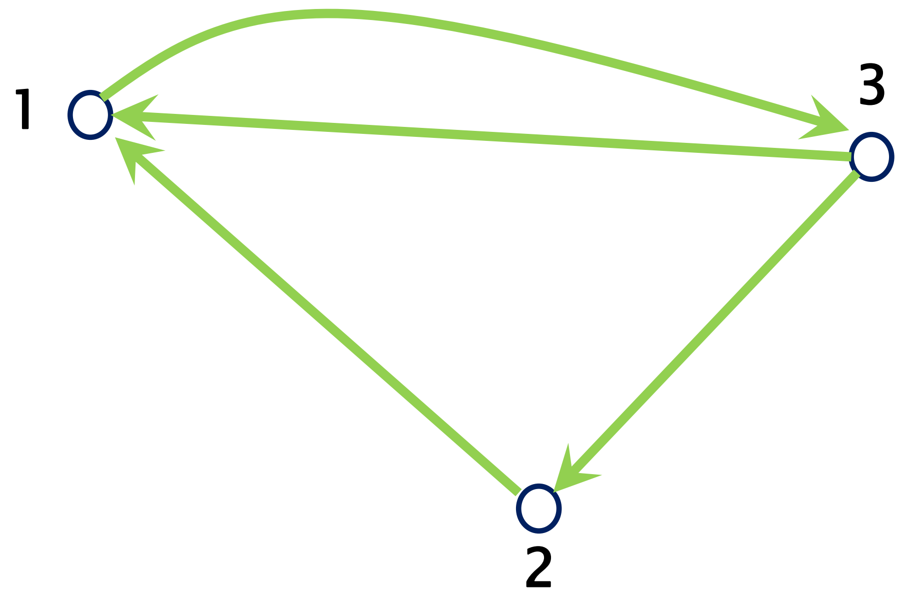
flux în reţea care saturează arcele din s și t ⇒ G



flux în reţea care saturează arcele din s și t ⇒ G



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



Algoritm de determinare a unui graf orientat G cu

$s^+(G) = s_0^+$ **şi** $s^-(G) = s_0^-$

1. **Construim N reţeaua de transport asociată**

2. **Determinăm f* flux maxim în N**

Algoritm de determinare a unui graf orientat G cu

$s^+(G) = s_0^+$ **şi** $s^-(G) = s_0^-$

1. **Construim N reţeaua de transport asociată**
2. **Determinăm f^* flux maxim în N**
3. **Dacă val(f*)<** $d_1^+ + \ldots + d_n^+$ **atunci**

   Nu există G. STOP

Algoritm de determinare a unui graf orientat G cu

$s^+(G) = s_0^+$ şi $s^-(G) = s_0^-$

1. Construim N reţeaua de transport asociată
2. Determinăm f^* flux maxim în N
3. Dacă val(f*)< $d_1^+ + \ldots + d_n^+$ atunci

    Nu există G. STOP

4. V(G) = {1,…, n}

   E(G) = {ij| x_iy_j ∈N cu $f^*(x_iy_j)$ = 1}

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

Aplicaţii

Alte probleme de asociere

Problema de asociere (temă)

- Se dau 2 mulțimi de obiecte, spre exemplu produse și clienți (joburi/maşini, pagini web/servere, 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/maşini, pagini web/servere, 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/maşini, pagini web/servere, 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)

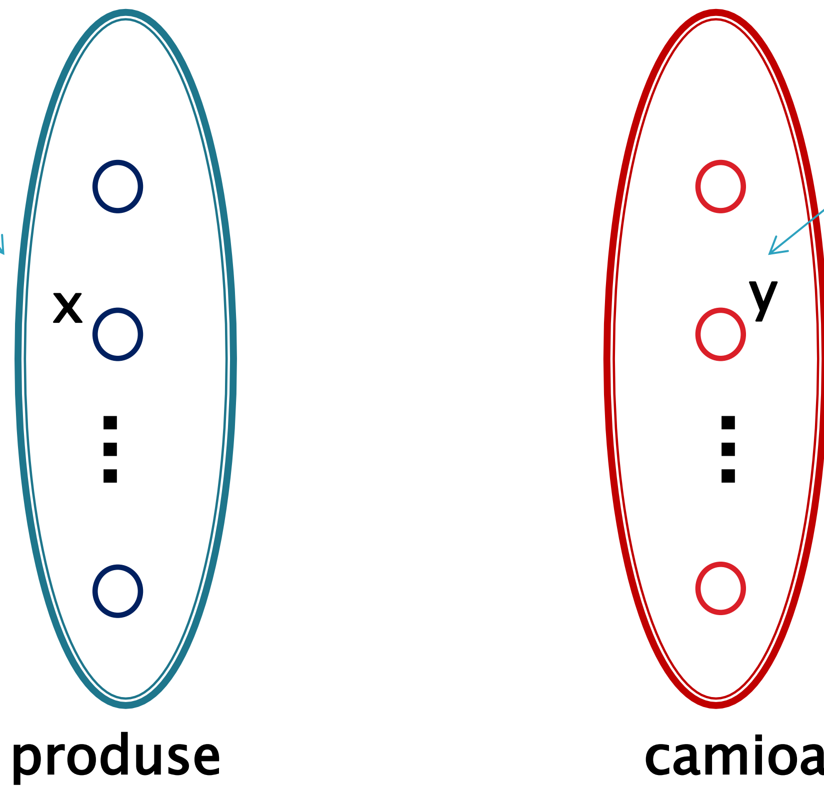
- Se dau 2 mulțimi de obiecte, spre exemplu produse și clienți (joburi/masini, pagini web/servere, echipe turneu etc).
  - Pentru fiecare produs x se cunoaște **c(x)** = numărul de unități disponibile din produsul x
  - Pentru fiecare client y se cunoaște **c(y)** = numărul maxim de unități de produse 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 (unităţi de produse) clienților cu respectarea constrângerilor

Problema de asociere (temă)

c(x) = numărul de kg de produs x (cantitatea)

c(y) = cate kg poate transporta camionul y (in total)

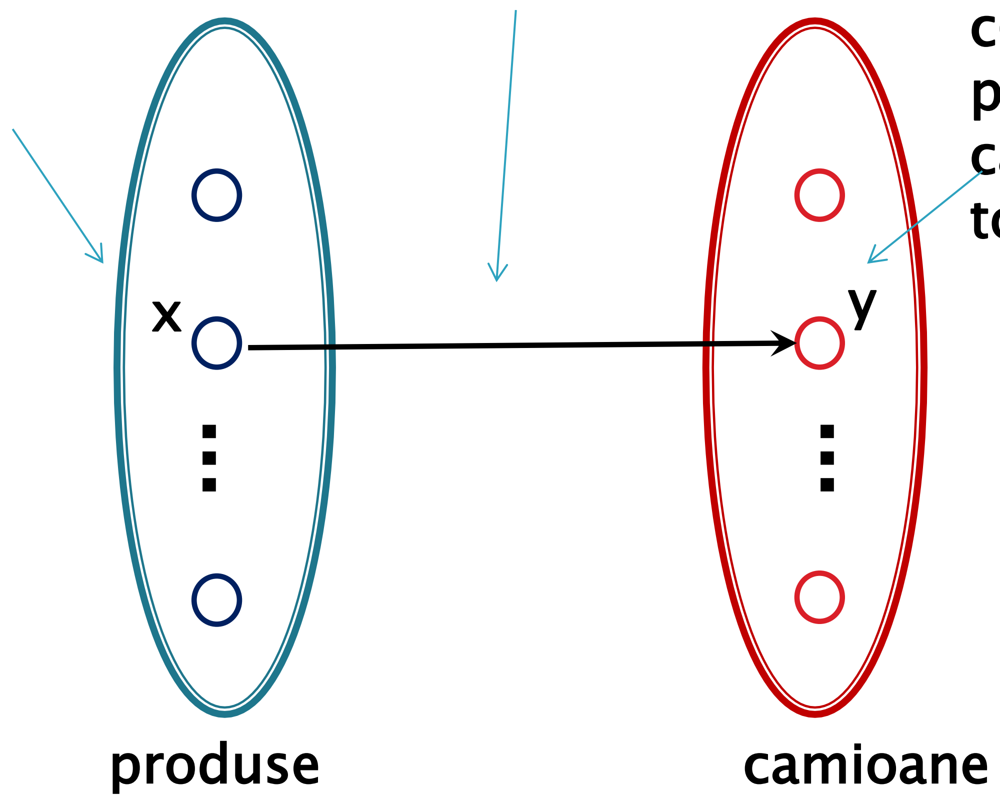


Problema de asociere (temă)

în camionul y pot fi încărcate cel mult c(x,y) kg din produsul y

c(x) = numărul de kg de produs x (cantitatea)

c(y) = cate kg poate transporta camionul y (in total)



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

Aplicaţie

Drumuri arc-disjuncte între două vârfuri.

Conectivitatea unui graf

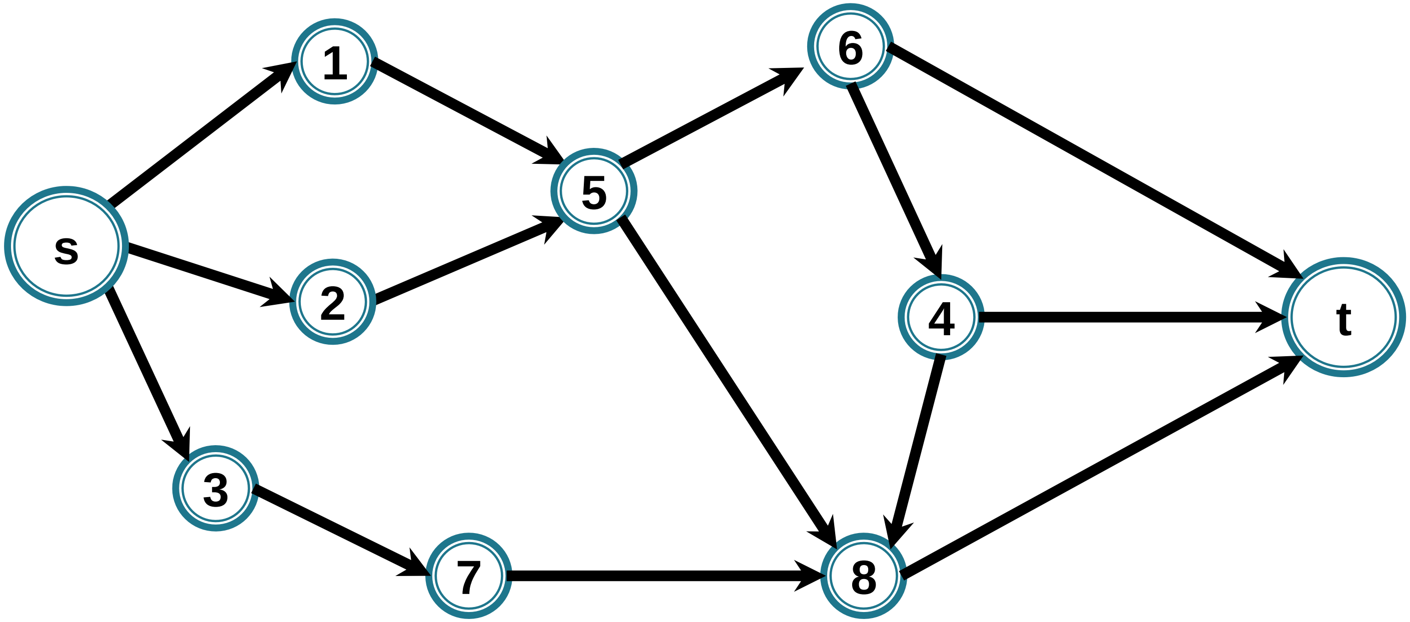
s-t drumuri arc-disjuncte

- $G = (V, E)$ - 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)**

- **Aplicaţii**

- Fiabilitatea reţelelor, conectivitate
- Probleme de strategie
- Măsuri de centralitate (a unui nod) în reţele 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)**

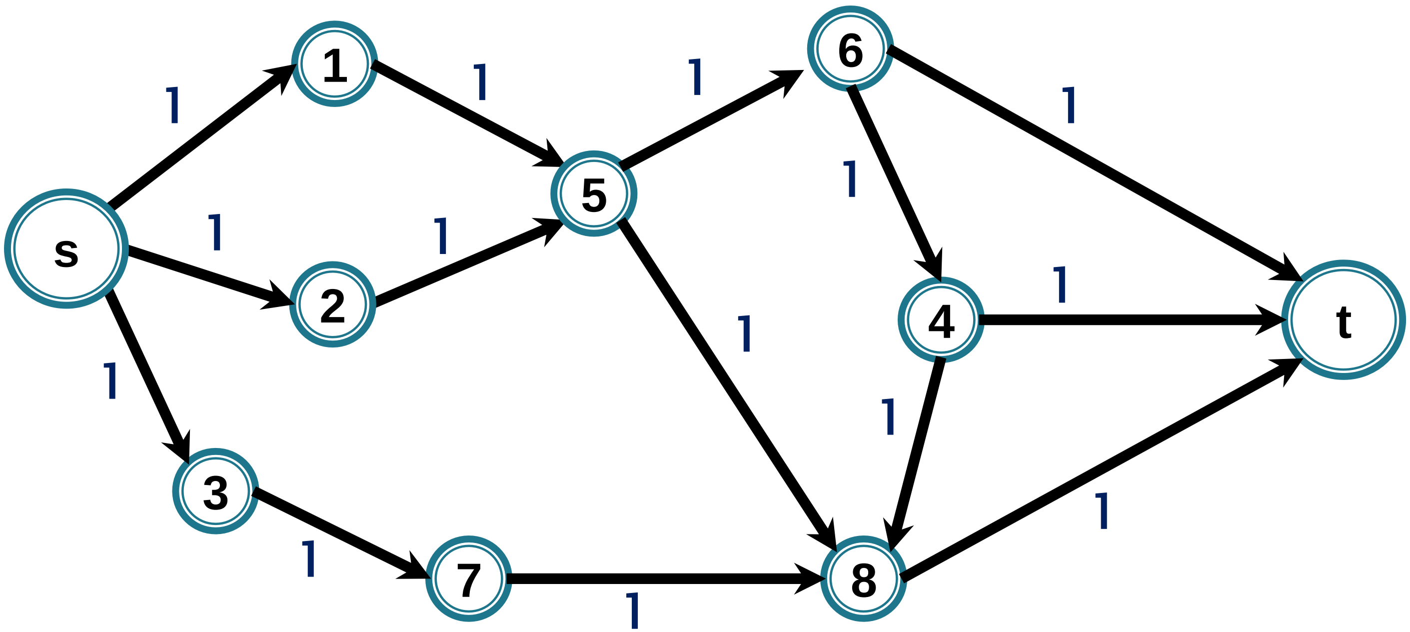
- Două drumuri P_1, P_2 s.n. **arc-disjuncte** dacă $E(P_1) \cap E(P_2) = \emptyset$

Temă (seminar)

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

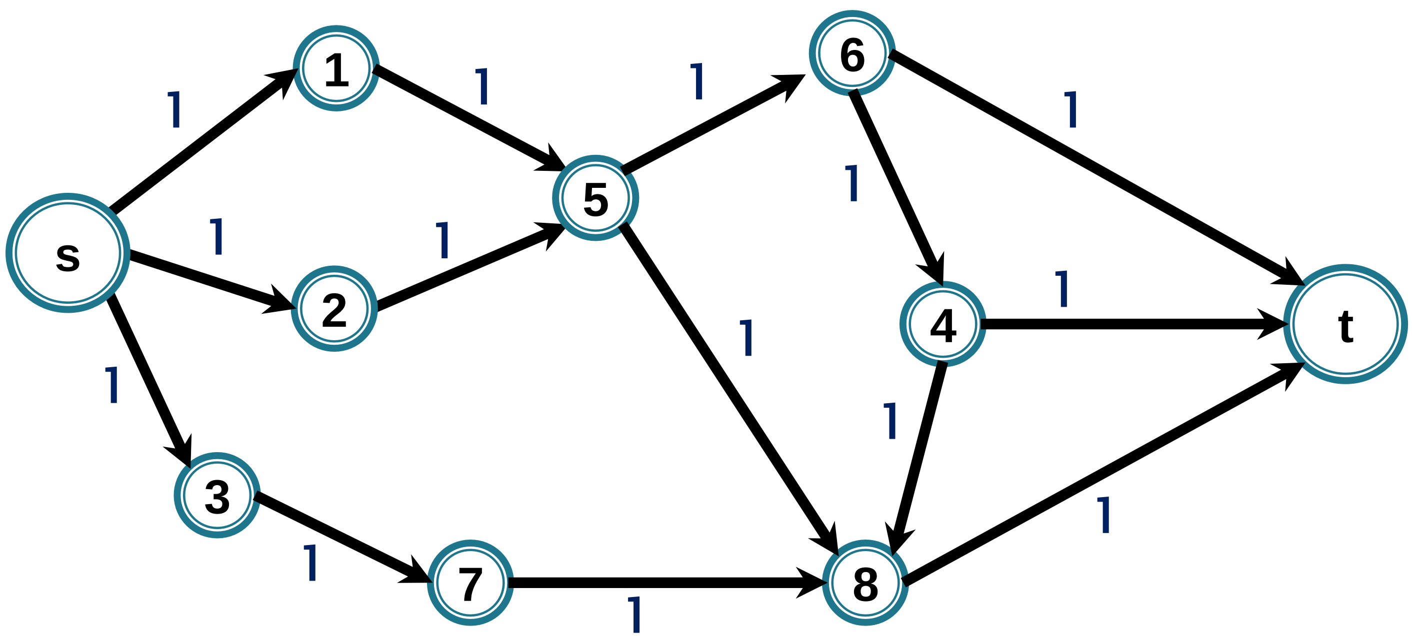
s-t drumuri arc-disjuncte

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

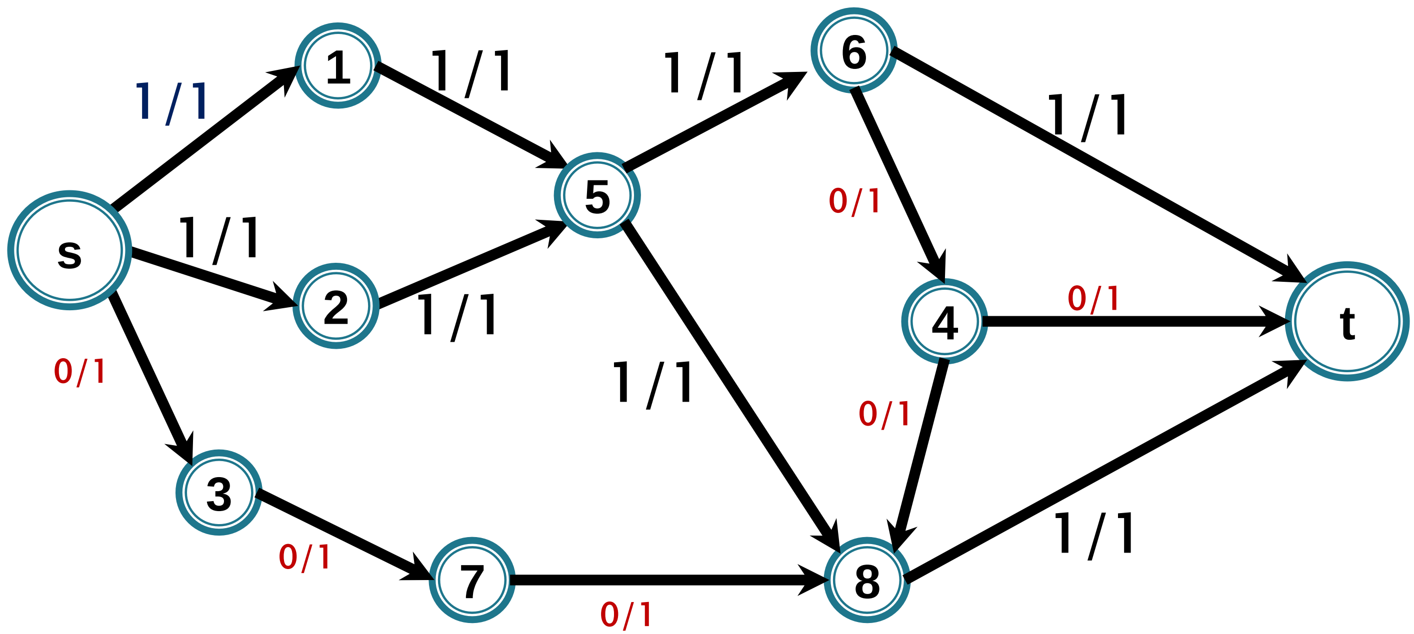


s-t drumuri arc-disjuncte

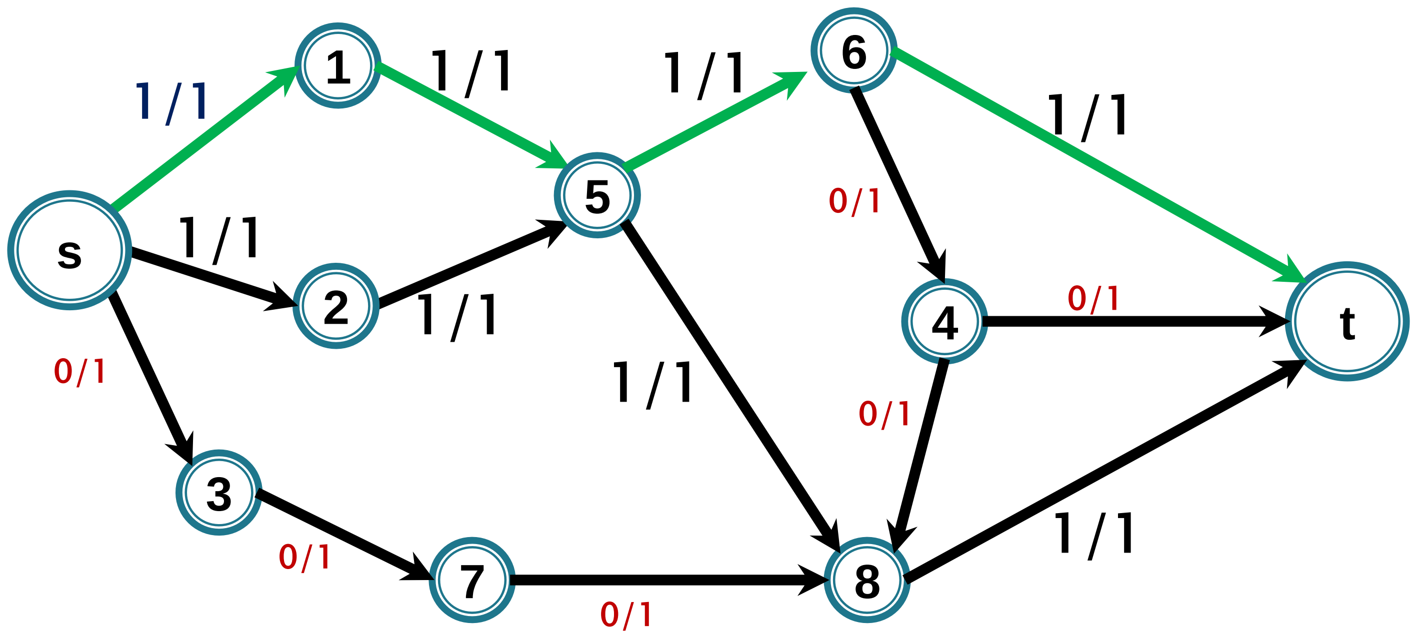
- **Intuitiv**:
  - Asociem fiecărui arc capacitatea 1
  - Fluxul maxim: f(e) $\in$ {0, 1}
  - Un drum de la s la t= **traseul parcurs de o unitate de flux de la s la t**
  - **Numărul de s-t drumuri arc-disjuncte**= valoarea fluxului maxim



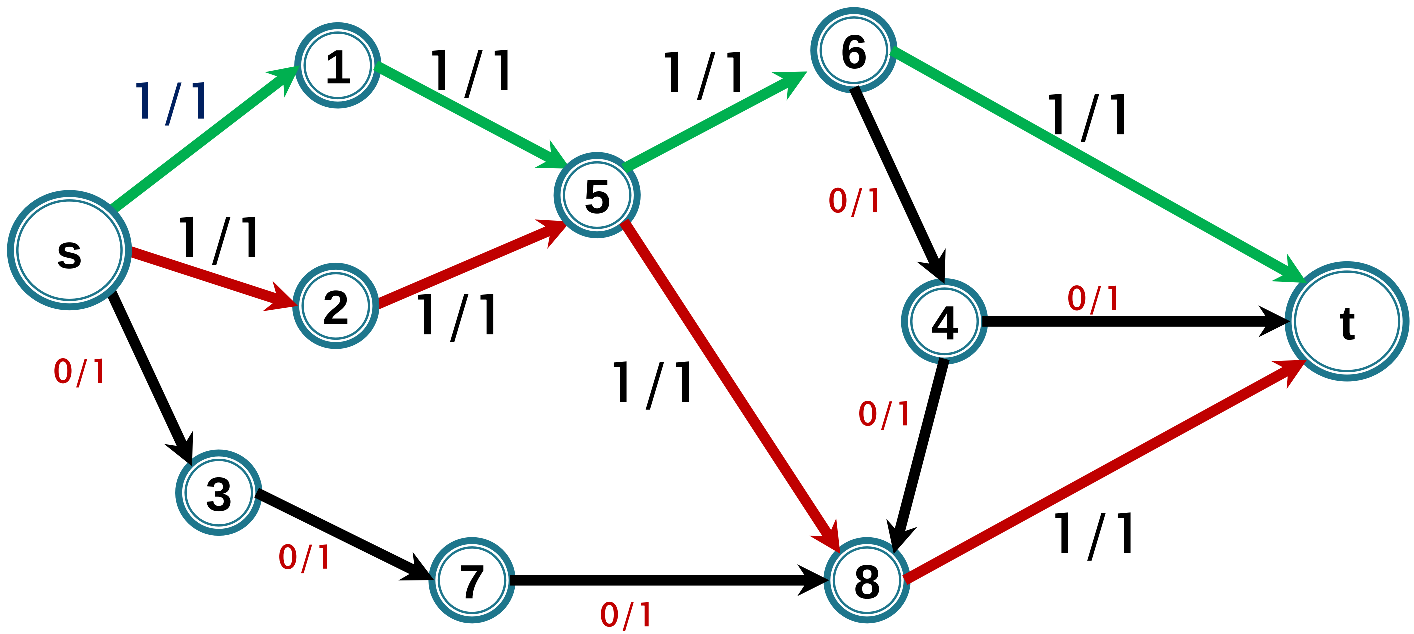
s-t drumuri arc-disjuncte



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 şi 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

- <u>**Teorema lui Menger**</u>

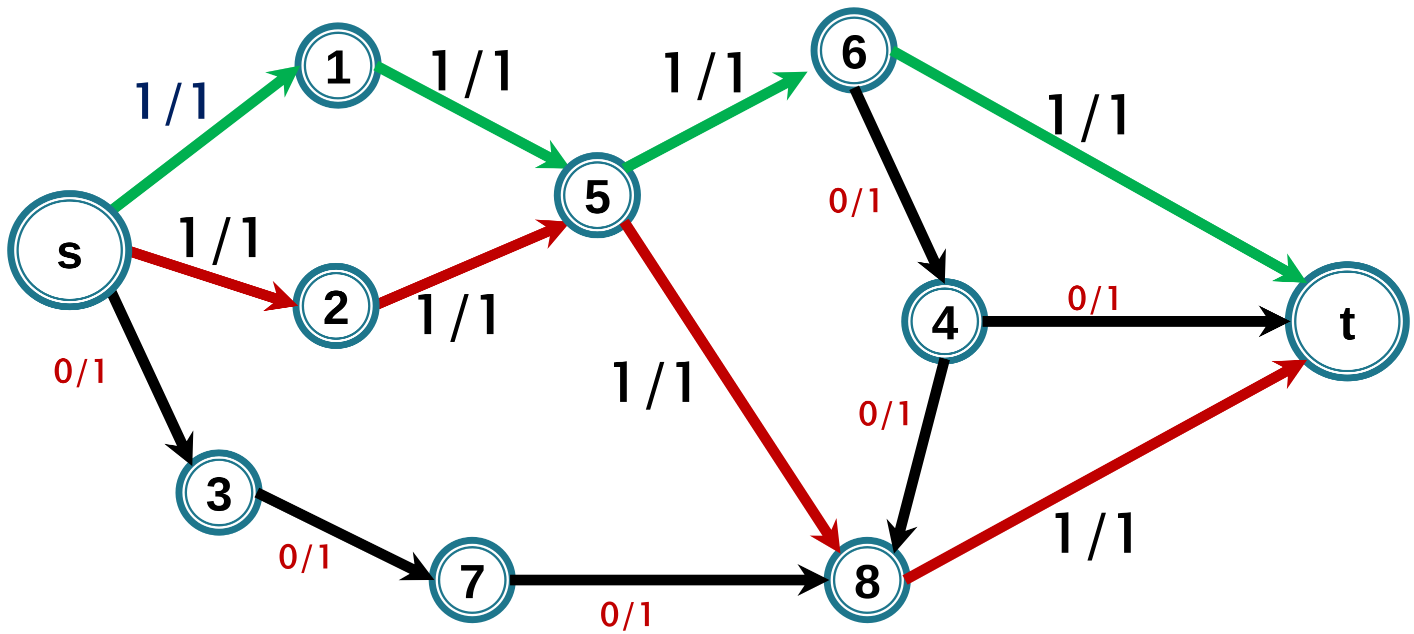
  Fie G graf orientat, s, t două vârfuri distincte în G.

  Numărul minim de arce care trebuie eliminate pentru ca s şi 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 mulţime de arce se poate determina cu algoritmul Ford Fulkerson?

s-t drumuri arc-disjuncte



s-t drumuri arc-disjuncte

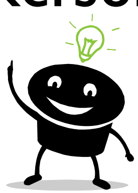
- <u>**Teorema lui Menger**</u>

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

  Numărul minim de arce care trebuie eliminate pentru ca s şi t să nu mai fie conectate printr-un drum (să fie **separate**) =

  numărul maxim de drumuri arc-disjuncte de la s la t

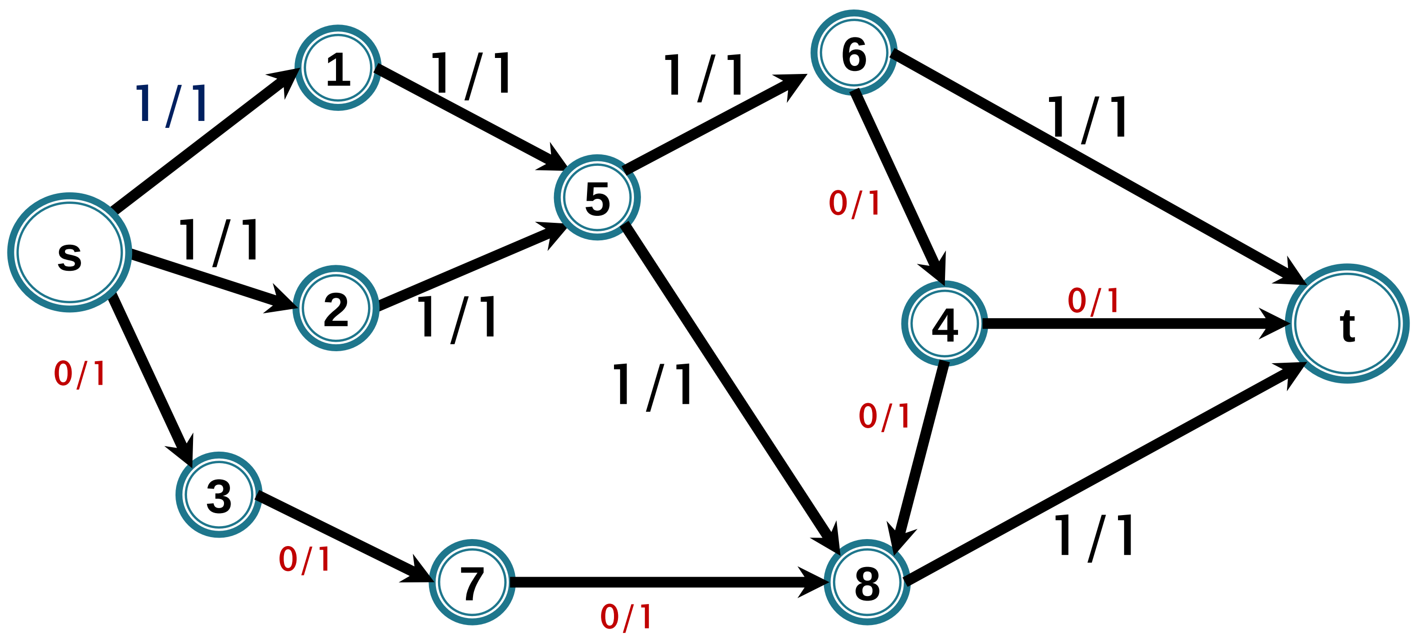
O astfel de mulţime de arce se poate determina cu algoritmul Ford Fulkerson



Sunt arcele directe ale tăieturii minime

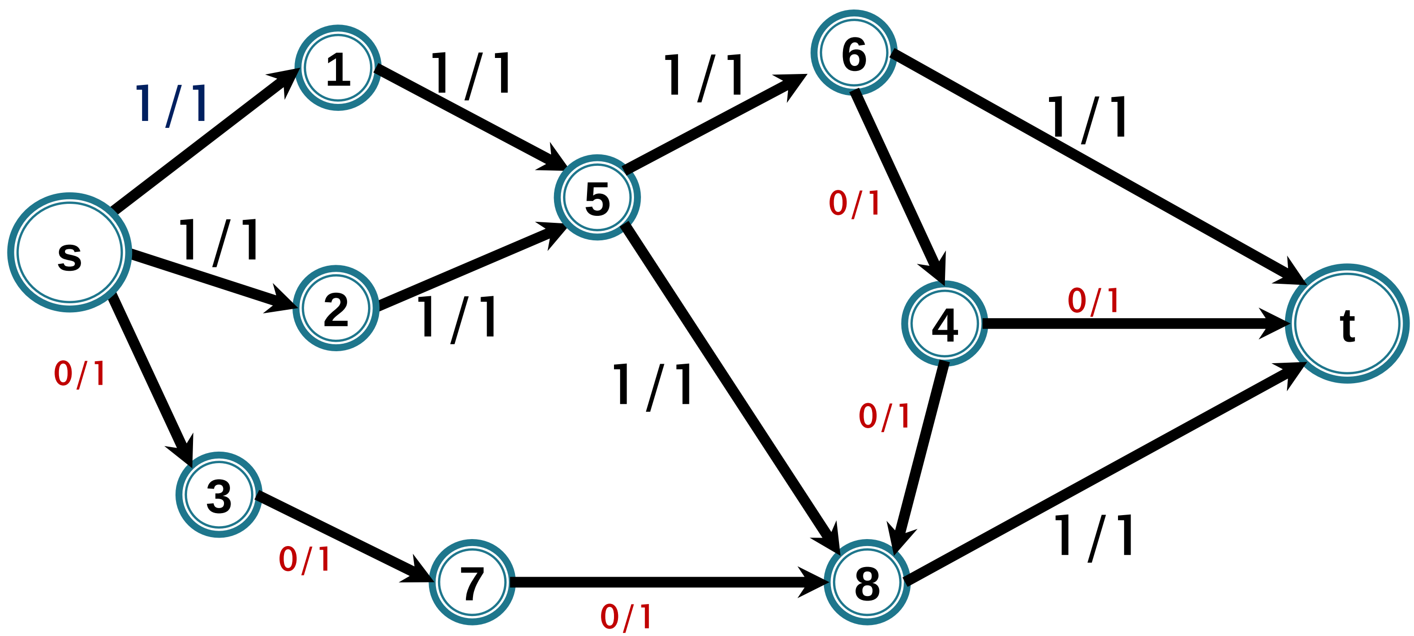
s-t drumuri arc-disjuncte

Cum determinăm tăietura minimă?

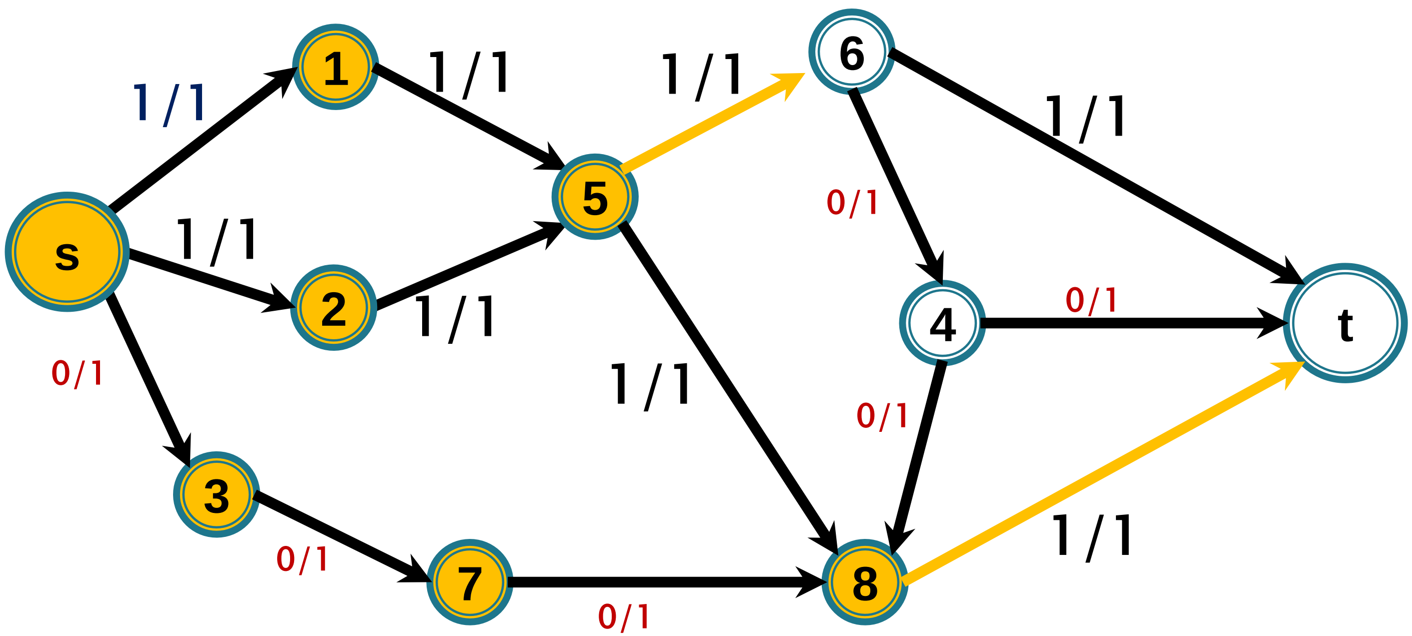


s-t drumuri arc-disjuncte

Mulţimea vârfurilor accesibile din s prin lanţuri f-nesaturate



s-t drumuri arc-disjuncte



s-t drumuri arc-disjuncte

Variante

- Aceeaşi problemă pentru
  - G= (V, E) – **neorientat** conex, |E|>2

- Aceeaşi 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 mulţimi de muchii $F \subseteq E$ cu proprietatea că

  G – F nu mai este conex

- Dacă k' (G) $\geq$ t, G se numeşte **t-muchie conex**

s-t drumuri arc-disjuncte

- **Muchie-conectivitatea lui G k'(G) =** cardinalul **minim** al unei mulţimi de muchii $F \subseteq E$ cu proprietatea că

  G – F nu mai este conex

- Dacă $k'(G) \geq t$, G se numeşte **t-muchie conex**

  - **Amintim (laborator+seminar):**
    - există muchie critică ⇒ G este 1-conex
    - Nu există muchie critică ⇒ G este 2-conex

- **Cu ajutorul algoritmului de flux maxim putem determina (muchie)-conectivitatea unui graf**