

Teorema di Berge

Teorema (Berge, 1957) Un abbinamento A di G è massimo **se e solo se** G non ammette cammini alternanti rispetto ad A che siano anche aumentanti.

Un possibile algoritmo

```
A =  $\emptyset$ ; //Inizializzazione
trovato = TRUE;
while (trovato) {
    search (A, &trovato);
    if (trovato)
        aumenta (G, &A);
}
```

Come è fatta `search (G, A, &trovato)`?

Teorema del cammino aumentante

Teorema

Sia v un vertice **esposto** in un abbinamento **A** . Se non esiste un cammino aumentante per A che parte da v , allora esiste un abbinamento massimo avente v esposto

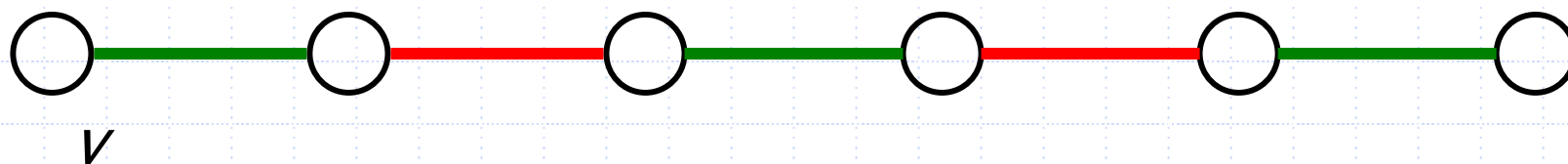
Dimostrazione

Sia **A^*** un abbinamento massimo in cui v è accoppiato.

Consideriamo **$A \oplus A^*$** .

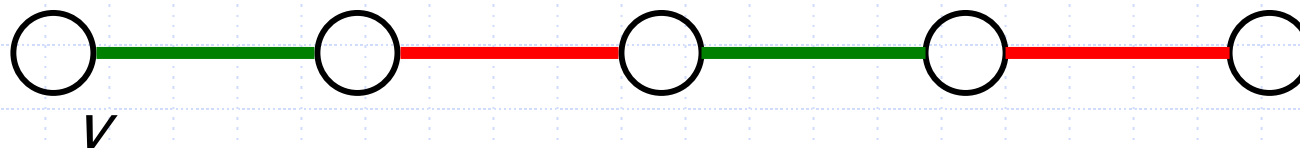
Dimostrazione

Poiché A e A^* sono due abbinamenti e v è accoppiato in A^* , si ha che $A \oplus A^*$ *non può* contenere un cammino



perché aumentante per A .

Però, contiene un cammino



Pertanto, scambiando gli spigoli verdi con quelli rossi, posso ottenere da A^* un altro abbinamento massimo, ma con v esposto.

Un possibile algoritmo (II)

```
A =  $\emptyset$ ; //Inizializzazione
for (v  $\in$  V) {
    if (v è esposto) {
        search (v, A, &trovato, &q);
        if (trovato)
            aumenta (q, v, &A);
        else
            //cancella v e tutti gli
            //spigoli incidenti in v
            cancella (v, &G);
    }
}
```

Ricerca di cammini aumentanti

Scopo della funzione search

trovare un cammino aumentante rispetto A , oppure dire che non esiste

Parametri

v nodo esposto,

A abbinamento,

trovato, variabile booleana

q , vertice estremo del cammino aumentante

Introduciamo un'etichetta per i vertici di V

label (w) = **PARI**
 = **DISPARI**
 = **NULL**

Ricerca di cammini aumentanti (II)

```
search (v, A, *q, *trovato) {
    for (i ∈ V)
        label (i) = NULL;
    LIST = {v};
    label {v} = PARI;
    while (LIST != ∅) {
        pop (&i, LIST);
        if (label (i) == PARI)
            esplora_pari (i, A, q, trovato,
                &LIST);
        else
            esplora_dispari (i, A, &LIST);
        if (trovato)
            return;
    }
}
```

Ricerca di cammini aumentanti (III)

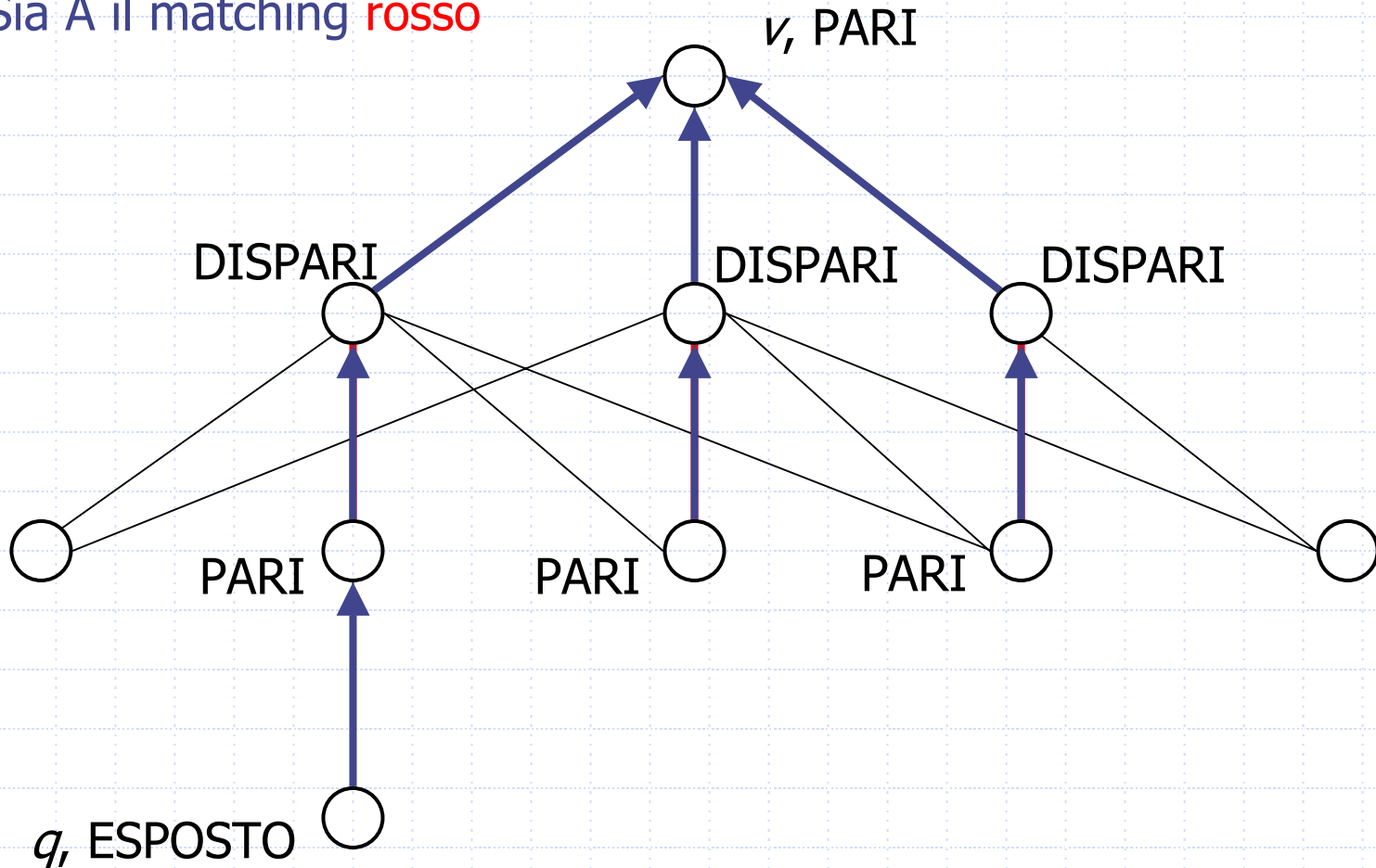
```
esplora_pari (i, A, *q, *trovato, *LIST) {  
    for (j  $\in$   $\delta(i)$ ) {  
        if (j  $\notin$  A) {  
            *q = j;  
            *trovato = TRUE;  
            pred (q) = i;  
            return;  
        }  
        if (j  $\in$  A && label (j) == NULL) {  
            pred (j) = i;  
            label (j) = DISPARI;  
            push (j, LIST);  
        }  
    }  
}
```

Ricerca di cammini aumentanti (IV)

```
esplora_dispari (i, A, *LIST)
{
  j = vertice accoppiato ad i in A;
  if (label (j) == NULL) {
    pred (j) = i;
    label (j) = PARI;
    push (j, LIST);
  }
}
```

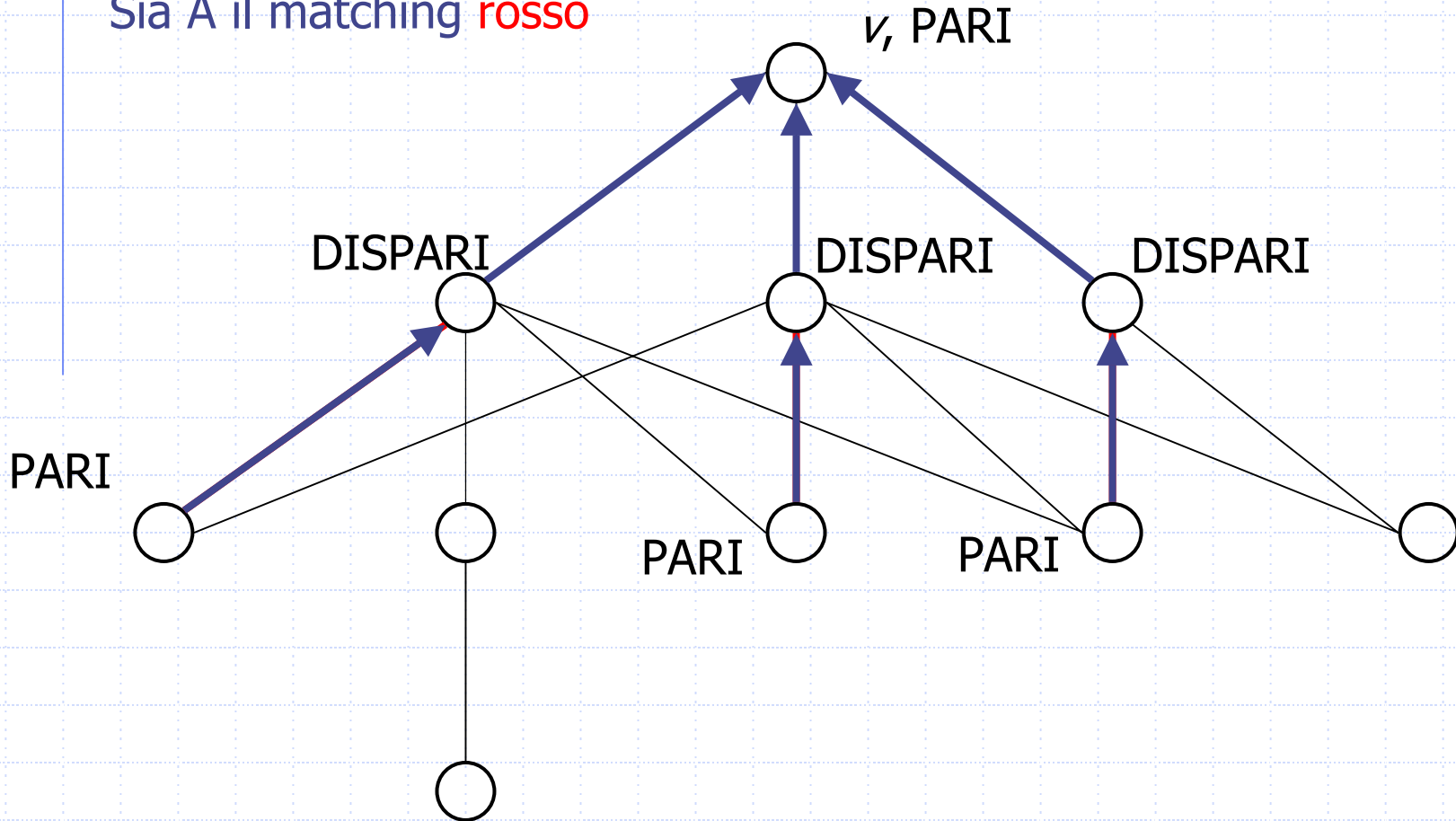
Esempio

Sia A il matching **rosso**

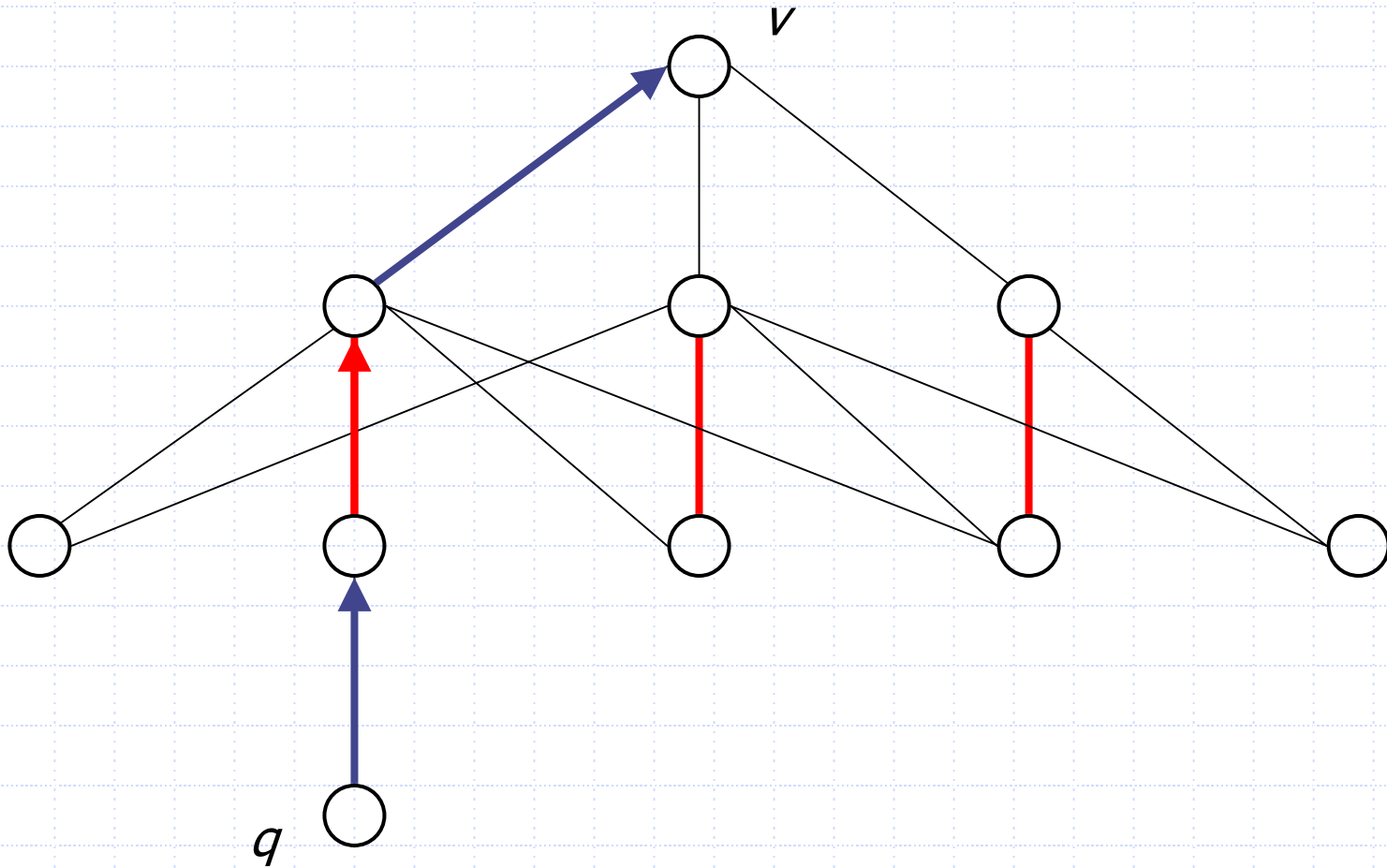


Esempio

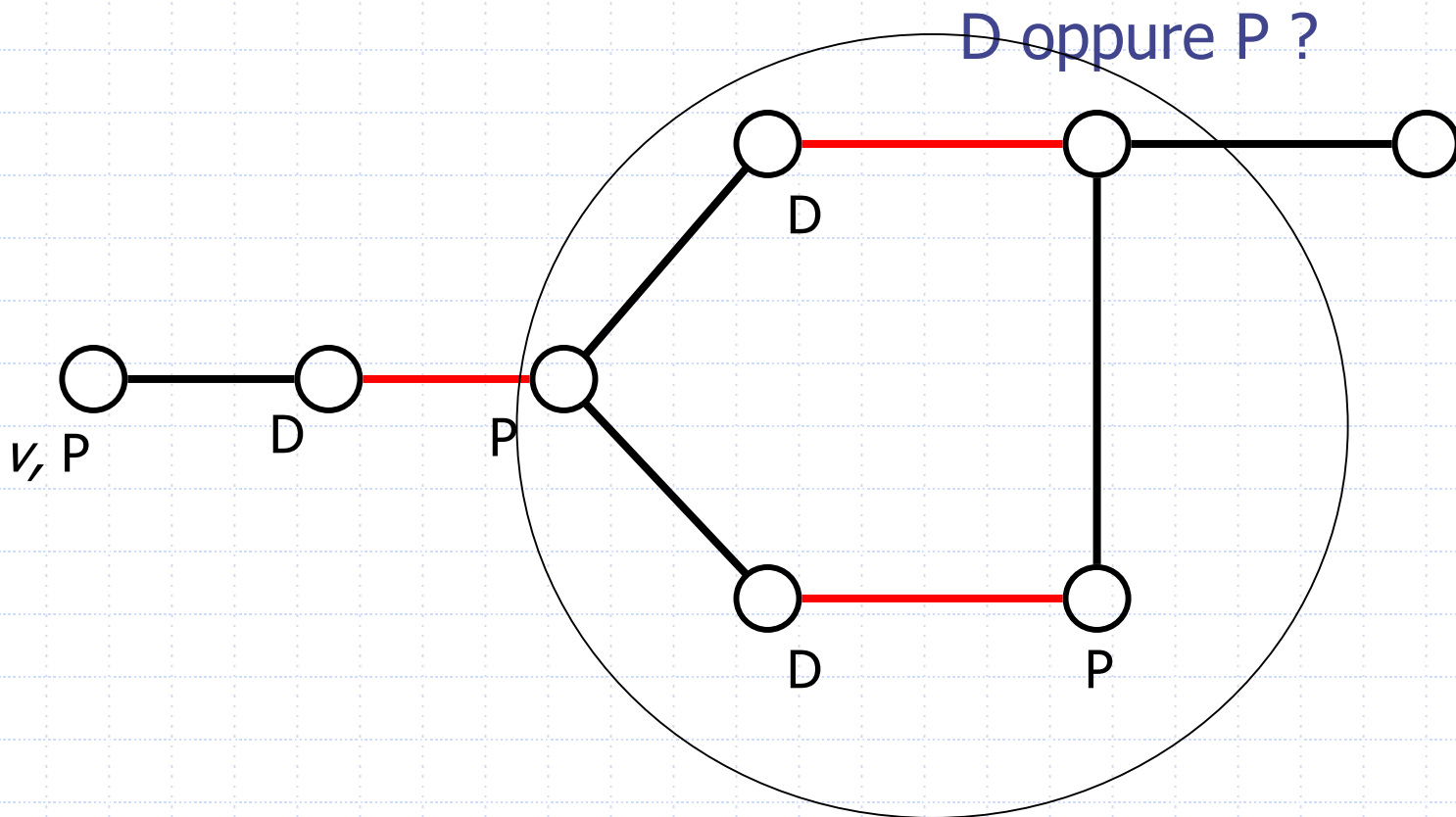
Sia A il matching rosso



Esempio



Un problema



Teorema

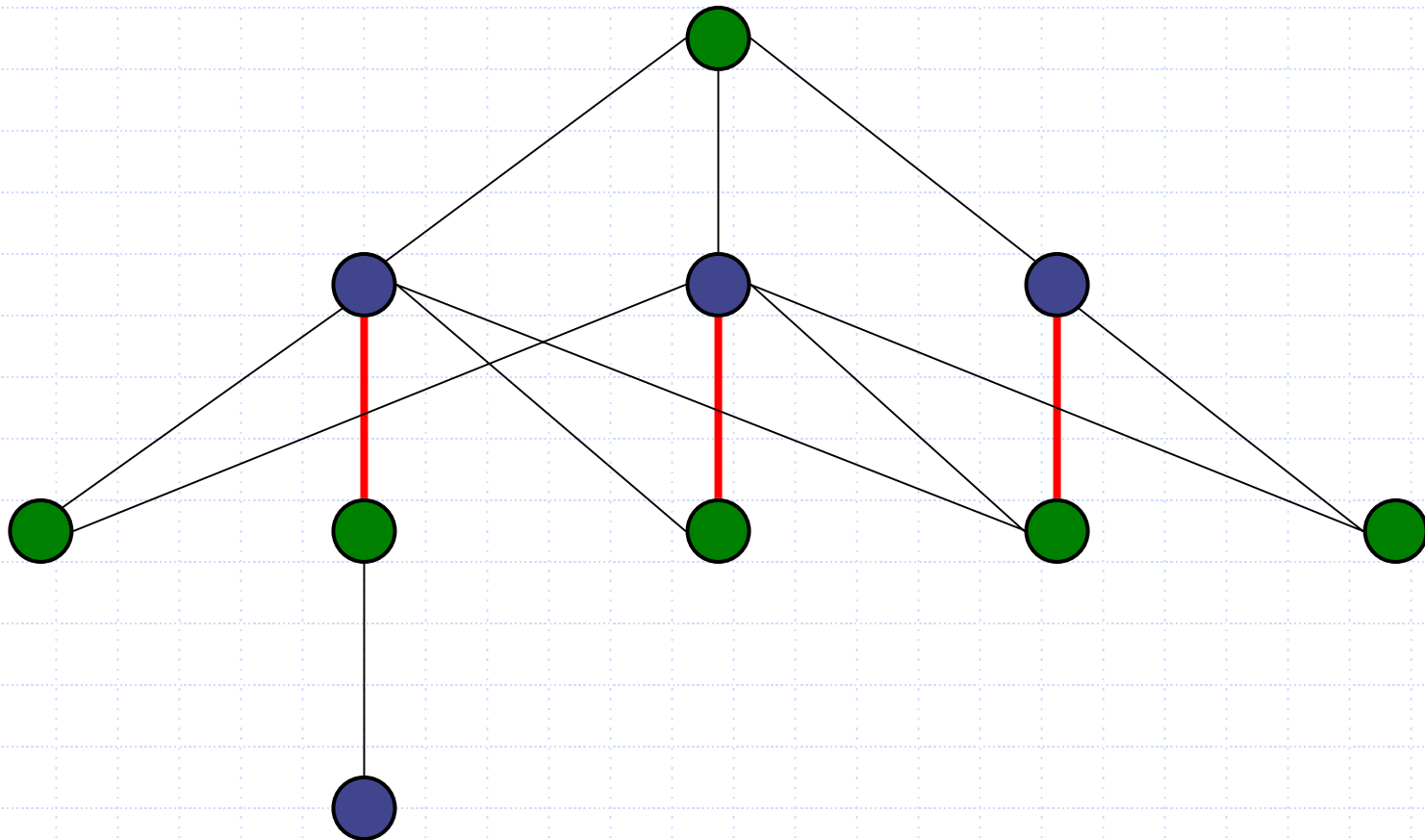
Se i vertici di G sono etichettati in modo "unico" dalla procedura search rispetto ad un abbinamento A , allora search termina con un cammino aumentante, se esso esiste.

Domanda

Esistono grafi che ammettono sempre la proprietà di unicità delle etichette?

Grafi bipartiti

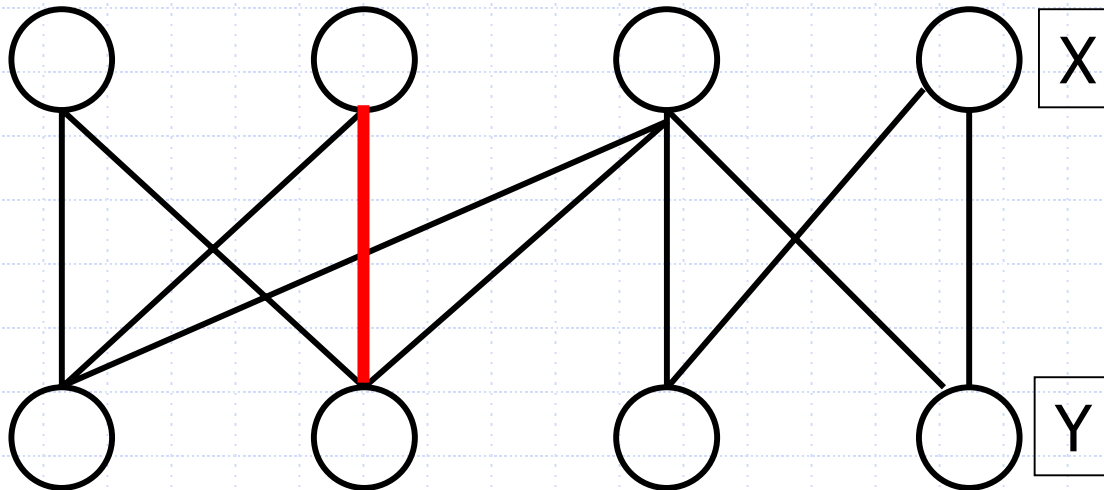
$G(X, Y, E)$ grafo bipartito



L'algoritmo ungherese

Sia $G(X, Y, E)$ un **grafo bipartito**

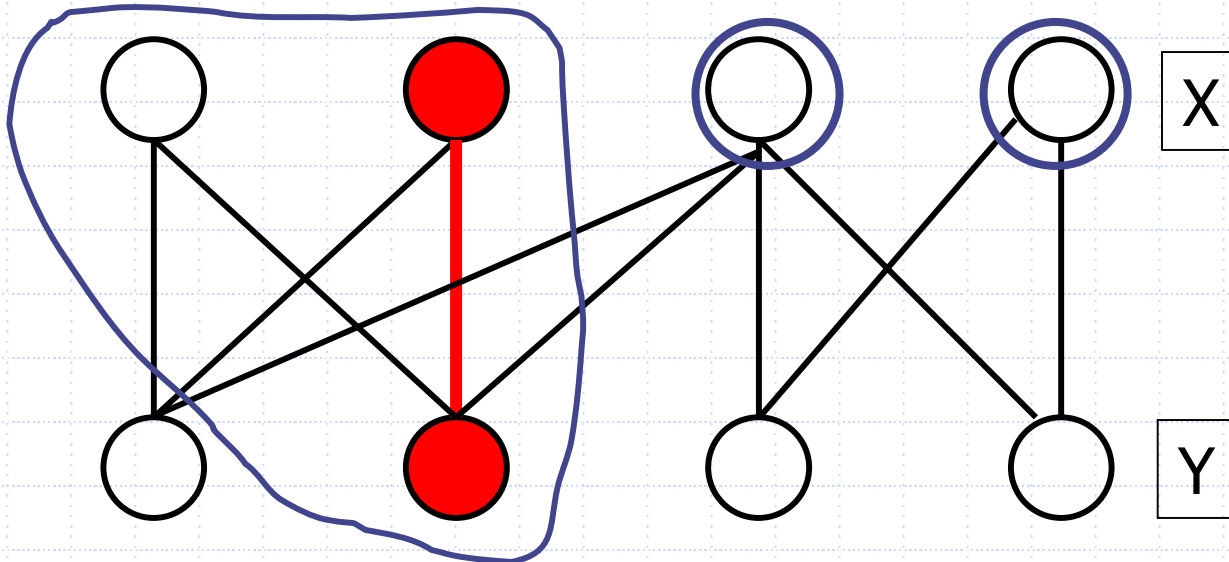
1. Scegli un abbinamento A qualsiasi



L'algoritmo ungherese

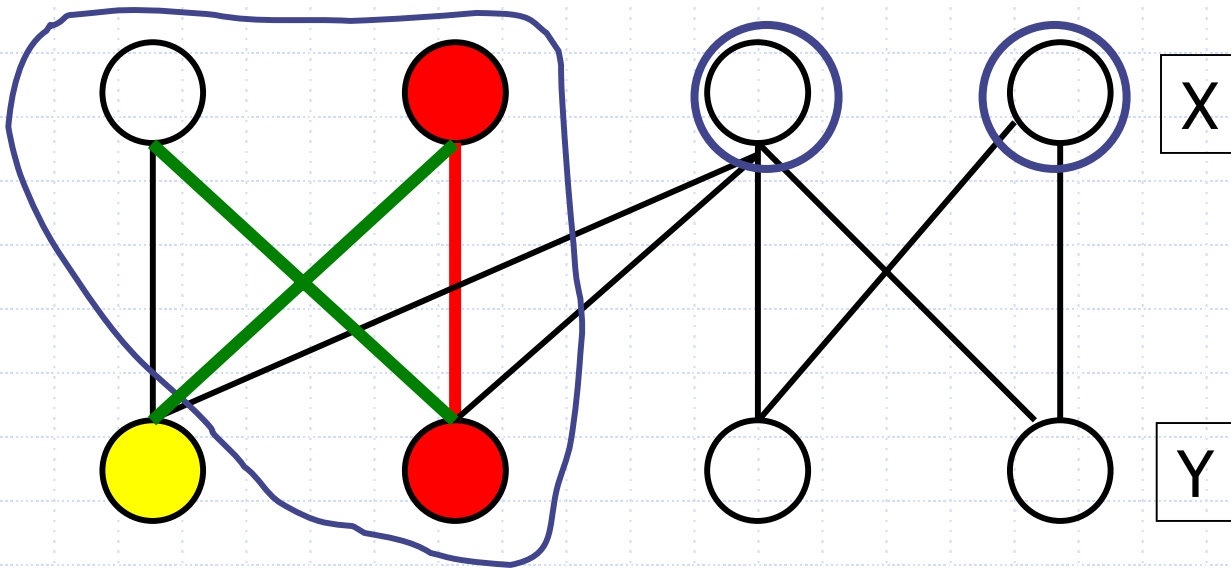
2. Forma una foresta massimale F con le seguenti caratteristiche:

- ogni vertice y di Y in F ha grado 2
- in y incide uno spigolo di A
- ogni componente di F contiene un vertice esposto di X



L'algoritmo ungherese

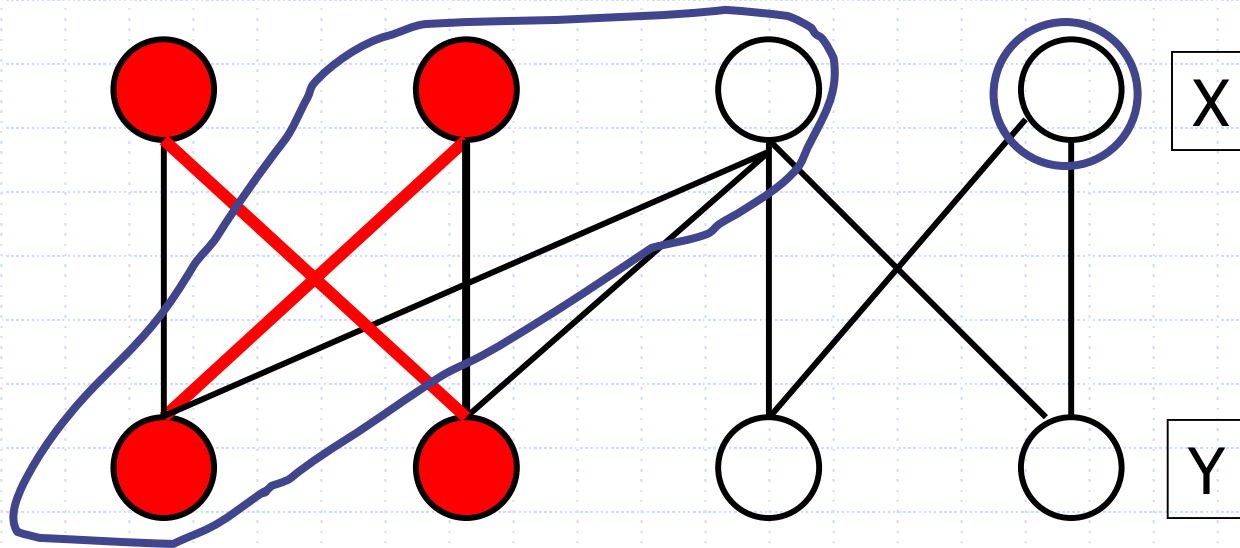
3. Se esiste uno spigolo che unisce $V(F) \cap X$ con un vertice esposto di Y , allora abbiamo trovato un cammino aumentante. Aumenta A e vai al passo 2.



L'algoritmo ungherese

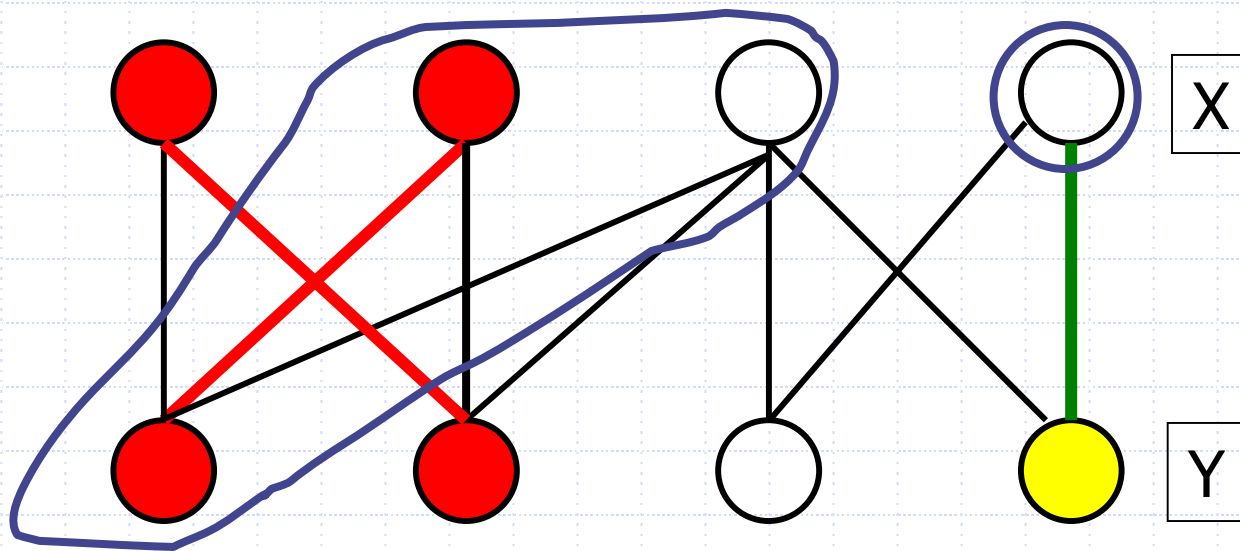
2 (2). Forma una foresta massimale F con le seguenti caratteristiche:

- ogni vertice y di Y in F ha grado 2
- in y incide uno spigolo di A
- ogni componente di F contiene un vertice esposto di X



L'algoritmo ungherese

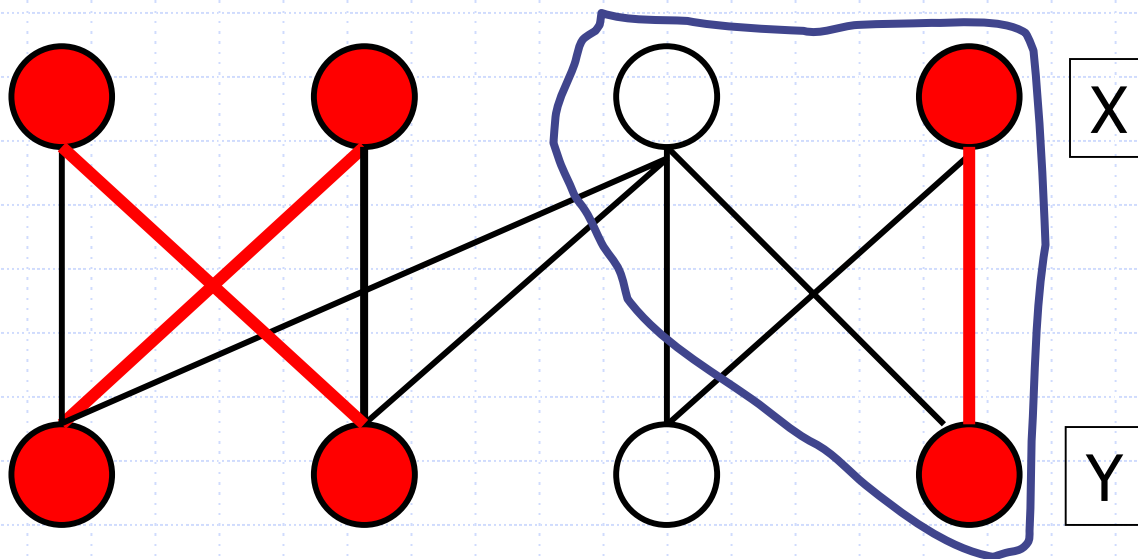
3 (2) Se esiste uno spigolo che unisce $V(F) \cap X$ con un vertice esposto di Y , allora abbiamo trovato un cammino aumentante. Aumenta A e Vai al passo 2.



L'algoritmo ungherese

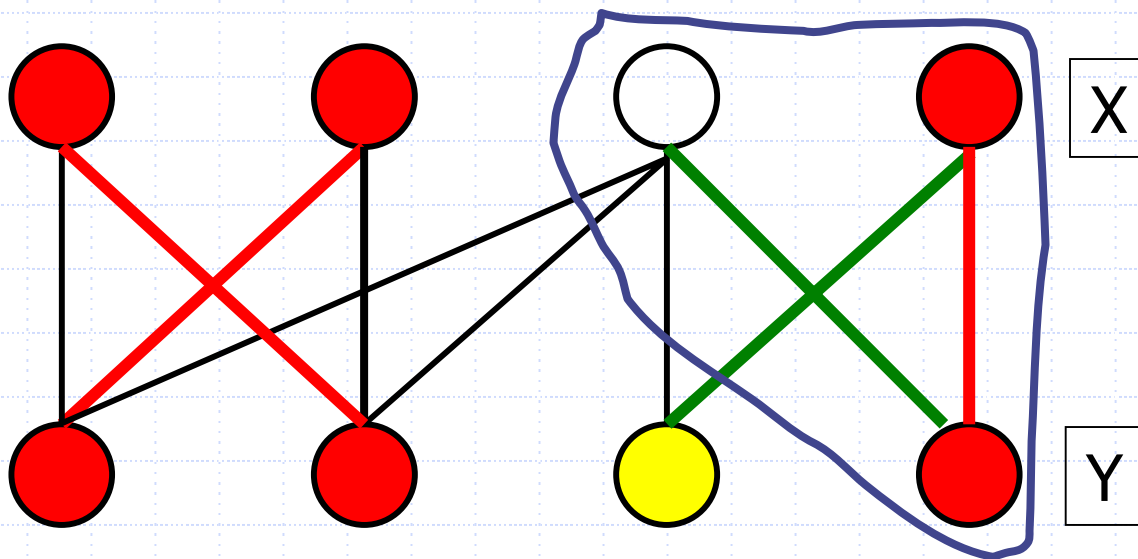
2 (3) Forma una foresta massimale F con le seguenti caratteristiche:

- ogni vertice y di Y in F ha grado 2
- in y incide uno spigolo di A
- ogni componente di F contiene un vertice esposto di X



L'algoritmo ungherese

3 (3) Se esiste uno spigolo che unisce $V(F) \cap X$ con un vertice esposto di Y , allora abbiamo trovato un cammino aumentante. Aumenta A e vai al passo 2.



L'algoritmo ungherese

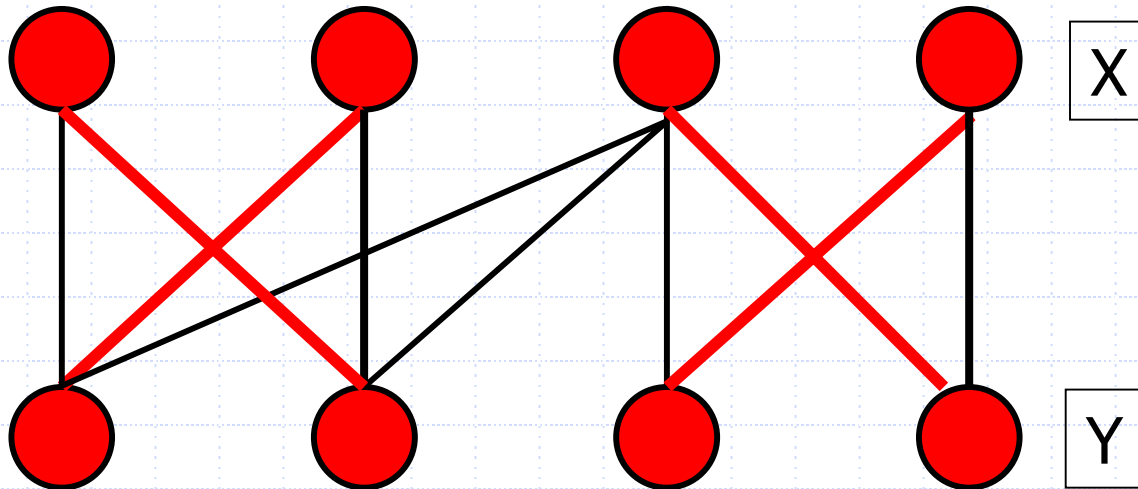
La foresta è vuota

STOP

altrimenti

La foresta non è vuota, ma non ci sono spigoli che uniscono vertici $V(F) \cap X$ con vertici esposti di Y , A è massimo

STOP



Teorema di König

Teorema (König 1931).

Se $G = (X, Y, E)$ è un grafo bipartito allora $\mu(G) = \tau(G)$

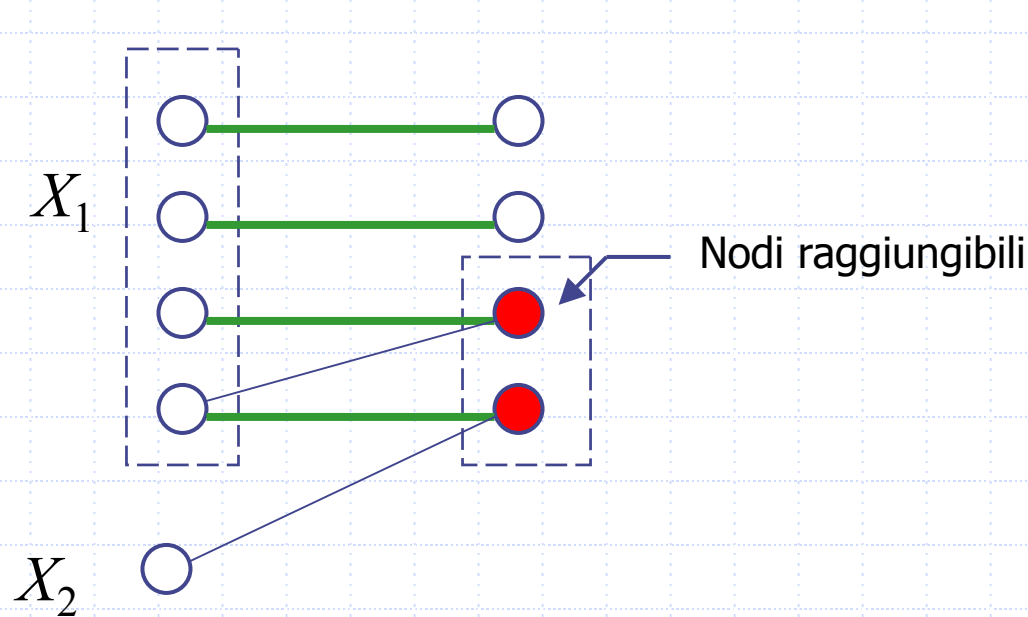
Dimostrazione

Sia A un abbinamento massimo, e siano

X_1 : insieme dei nodi x di X saturi rispetto ad A

X_2 : insieme dei nodi x di X esposti rispetto ad A

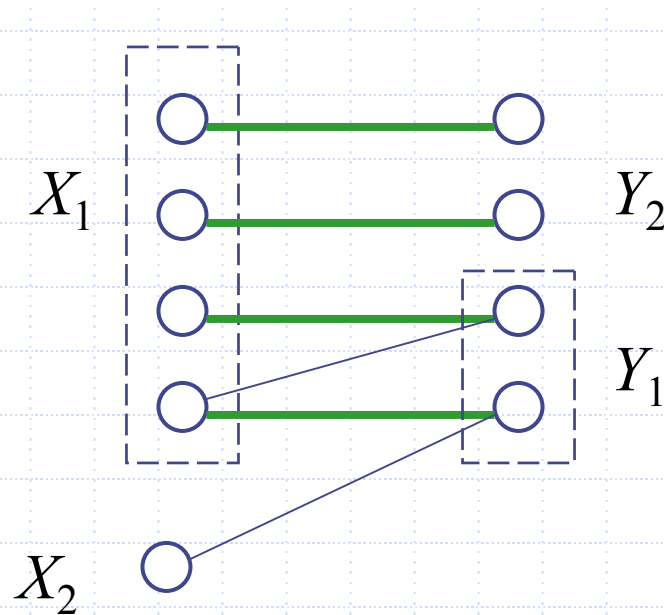
Teorema di König



Definizione: Un nodo $y \in Y_1$ è **raggiungibile** se esiste P **alternante** rispetto ad A da x in X_2 tale che l'ultimo arco **non appartiene** ad A

Y_1 : insiemi dei nodi y di Y raggiungibili da x in X_2

Teorema di König

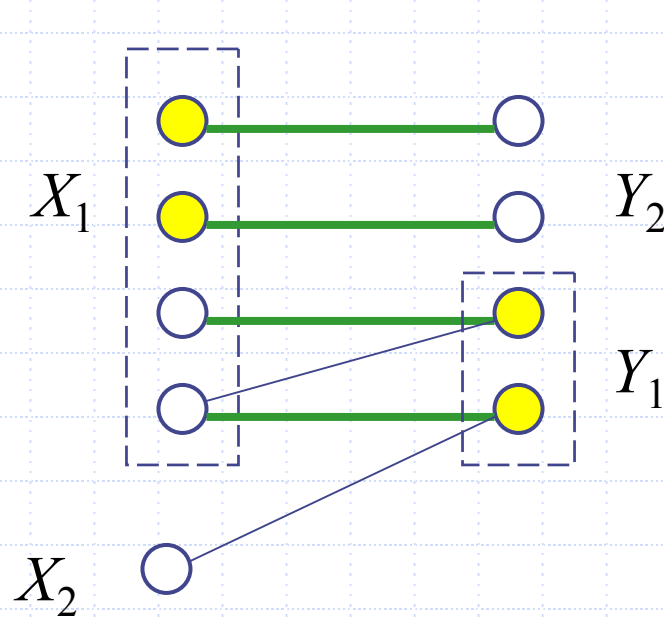


Osservazione

Per definizione i nodi in Y_1 sono saturi, altrimenti A non sarebbe massimo

Infine: $Y_2: Y - Y_1$

Teorema di König



Consideriamo il seguente insieme di nodi

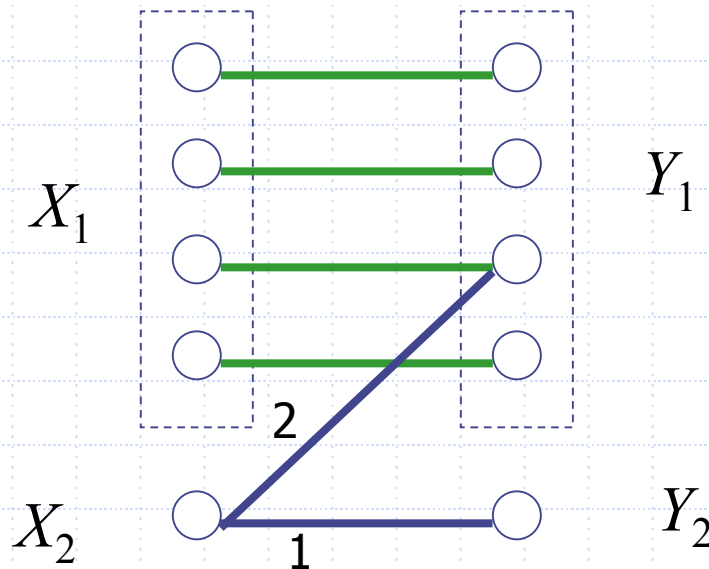
$$Z = \{z_1, z_2, \dots, z_{\mu(G)}\} \quad \text{con} \quad \begin{array}{ll} z_i = y_i & \text{se } y_i \text{ è raggiungibile} \\ z_i = x_i & \text{altrimenti} \end{array}$$

e dimostriamo che è un trasversale

Dimostrazione (I)

Dimostriamo che non esistono archi da nodi in X_2 verso nodi in Y non coperti da Z . Difatti,

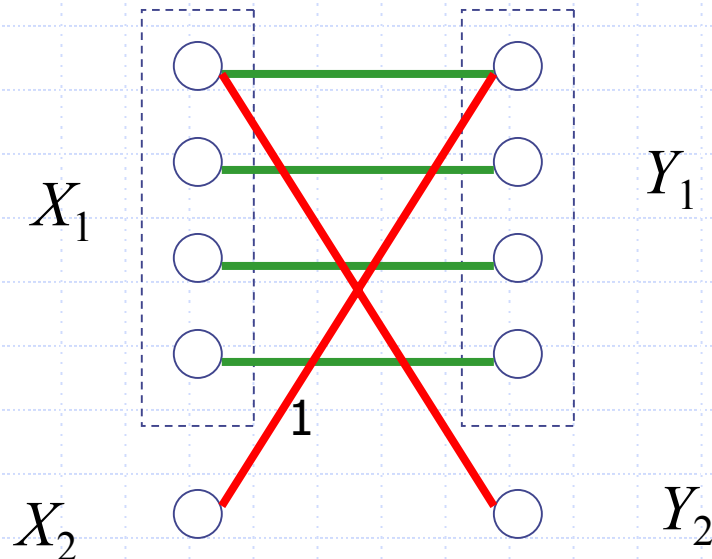
- 1) Non può esistere un arco non coperto da Z tra un nodo in X_2 e un nodo in Y_2 , altrimenti il matching non sarebbe massimo
- 2) Non può esistere un arco non coperto da Z tra un nodo in X_2 e un nodo in Y_1 perché i nodi in Y_1 sono raggiungibili e quindi l'arco sarebbe coperto



Dimostrazione (II)

Dimostriamo che non esistono archi da nodi in Y verso nodi in X_1 non coperti da Z .

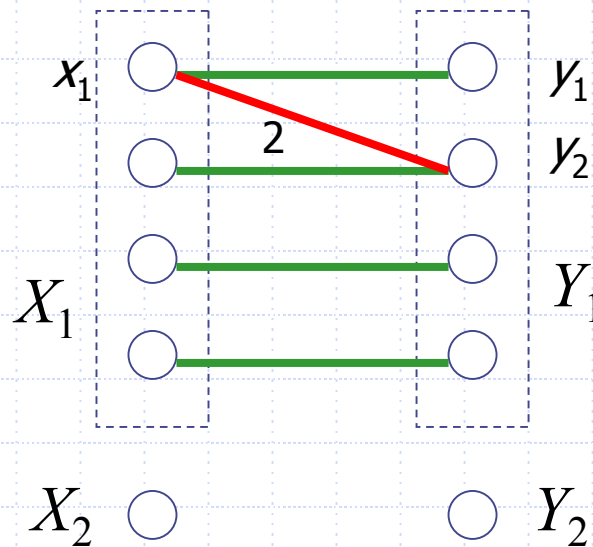
Difatti, consideriamo l'arco 1, da Y_2 a X_1 . Se non fosse coperto significa che il nodo y_1 , estremo dell'arco del matching è raggiungibile. Ma allora esisterebbe un cammino aumentante e il matching non sarebbe massimo.



Dimostrazione (III)

Consideriamo un arco da X_1 a Y_1 , ad esempio l'arco 2.

Se non fosse coperto significa che il nodo y_2 non è raggiungibile, ovvero non appartiene ad Y_1 (contraddizione).



Pertanto Z è un trasversale di cardinalità pari a $\mu(G)$.