Matchings

Daniel Penazzi

14 de mayo de 2021

Tabla de Contenidos

Matching

Definición

Dado un grafo G un matching en G es un subgrafo M de G tal que el grado (en M) de cada vértice de M es uno.

- En otras palabras, un matching es un conjunto de lados disjuntos.
- El problema a resolver será:
- Dado un grafo G bipartito, encontrar un matching en G con la mayor cantidad de lados posible.
- Un matching que tenga la mayor cantidad posible de lados se dice matching maximal

Matching maximales

- Si bien hay problemas de matchings en grafos generales, nos concentraremos en matchings en grafos bipartitos, pues:
  1. Muchos problemas de aplicación sólo requieren un grafo bipartito
  2. El algoritmo para grafos no bipartitos se “apoya" en el algoritmo para bipartitos, asi que hay que conocer este de todos modos.
- Nota: la persona que generalizó el algoritmo que daremos para grafos bipartitos a grafos no bipartitos es....Edmonds. El algoritmo es mucho mas complicado y no lo daremos.

Solución al problema de Matching maximales

- Ejemplo: G es un grafo cuyos vértices son tareas que hay que hacer y trabajadores que pueden hacer esas tareas.
- Se une un trabajador con una tarea si el trabajador es capaz de hacer esa tarea.
- Queremos encontrar la mayor cantidad posible de emparejamientos tarea/trabajador.
- Es decir, un matching maximal.
- Mas adelante veremos una generalización de este problema, cuando pej asociamos a cada par tarea/trabajados un costo que puede cambiar dependiendo de cual trabajador hace cual tarea, y no sólo querramos un matching maximal sino un matching maximal de costo mínimo.

Solución al problema de Matching maximales

- La solución original a este problema no es exactamente la que daremos, pero la daremos de la forma en que la daremos para aprovechar lo que hemos aprendido:
  1. Transformamos el grafo bipartito no dirigido en un network (el cual tendrás todas sus capacidades enteras)
  2. Encontramos un flujo entero maximal en ese network, que sabemos por el teorema de la Integralidad que existe.
  3. A partir de ese flujo maximal obtenemos un matching maximal
- Sabemos como hacer la segunda de esas cosas, pues si las capacidades son enteras cualquier especificación de Ford-Fulkerson como Edmonds-Karp o Dinitz lo encuentra.
- Veamos como hacer las otras dos.

Transformación del grafo en network

- Sea G bipartito, y X, Y sus partes. Es decir:
- $V(G) = X \cup Y$, $X \cap Y = \emptyset$, y no hay lados entre vértices de X o entre vértices de Y.
  - Por lo tanto los únicos lados son entre algún vértice de X y algún vértice de Y.
- Para construir el network N, agregamos a los vértices dos vértices nuevos, que denotaremos por s y t.
- Los lados del network sera la unión de todos los siguientes lados:
  - $\overrightarrow{xy}$ para todo $x \in X,\ y \in Y,\ xy \in E(G)$
  - $\overrightarrow{sx}$ para todo $x \in X$
  - $\overrightarrow{yt}$ para todo $y \in Y$
- Con capacidad 1 en todos los lados

Relación entre matchings maximales en G y flujos maximales enteros en N

- Supongamos que tenemos un flujo f entero.
- Como la capacidad de todos los lados es 1, tenemos:
  - Si $x \in X$, como sólo hay un lado que entre a x (el $\overrightarrow{sx}$) entonces $in_f(x) = f(\overrightarrow{sx})$.
  - Como f es entero, y la capacidad de $\overrightarrow{sx}$ es uno, entonces $f(\overrightarrow{sx})$ es 1 o 0, y por lo de arriba, entonces $in_f(x)$ es 1 o 0.
  - Si un vértice $x \in X$ como $out_f(x) = in_f(x)$, debe ser $out_f(x)$ tambien 1 o 0.
  - Asi que si $out_f(x) > 0$, debe ser 1, y entonces, como f es entero, debe haber un único $y \in Y$ con $f(\overrightarrow{xy}) = 1$

Relación entre matchings maximales en G y flujos maximales enteros en N

- Un razonamiento similar, usando t, se puede hacer con los $y \in Y$: o bien $out_f(y) = 0$, o es 1, y en este último caso existe un único $x \in X$ tal que $f(\overrightarrow{xy}) = 1$.
- Concluimos que dado un flujo f entero, entonces si tomamos:
  - $W = \{x \in X : in_f(x) = 1\}$
  - $U = \{y \in Y : out_f(y) = 1\}$
  - $F = \{xy \in E(G) : x \in W,\ y \in U : f(\overrightarrow{xy}) = 1\}$
- Entonces $(W \cup U, F)$ es un subgrafo de G que es un matching.
- Pues para cada $x \in W$ hay un único $y \in U$ con $xy \in F$ y para cada $y \in U$ hay un único $x \in W$ con $xy \in F$.

Relación entre matchings maximales en G y flujos maximales enteros en N

- Es decir, podemos construir matchings a partir de flujos.
- Observemos que la cantidad de lados del matching será igual a la cantidad de $x \in X$ con $in_f(x) = 1$,
- Pero esto es igual a $out_f(s) = v(f)$.
- Asi que la cantidad de lados del matching es igual al valor del flujo.

Relación entre matchings maximales en G y flujos maximales enteros en N

- Viceversa, dado un matching, simplemente definimos f igual a 1 en los lados del matching, 0 en los otros lados $\overrightarrow{xy}$, y definimos $f(\overrightarrow{sx}) = 1$ si x es un vértice del matching y $f(\overrightarrow{yt}) = 1$ si y es un vértices del matching.
- Es trivial ver que f es flujo (facil: lo costruimos camino a camino con caminos de la forma $sxyt$)
- Y como antes, el valor de f es la cantidad de lados del matching
- Por lo tanto flujos maximales enteros se corresponden con matching maximales.

Algoritmo

- Con lo cual, para resolver el problema del matching maximal basta con:
  - Construir el network asociado.
  - Encontrar flujo maximal entero f en el network. (como dijimos, pej Edmonds-Karp o Dinitz encuentran un flujo maximal entero si todas las capacidades son enteras)
  - Tomar como matching aquellos lados xy con $f(\overrightarrow{xy}) = 1$.
- En realidad, ya sea que se use Dinitz o Edmonds-Karp, la primera parte es siempre “con Dinitz".
- Lo pongo entre comillas porque ni siquiera usamos Dinitz, pero es equivalente.

Algoritmo

- Observemos que si hicieramos Dinitz, el primer network auxiliar tendría nivel 0 igual a $\{s\}$, nivel 1 igual a X, nivel 2 igual a Y y nivel 3 igual a $\{t\}$.
- Todo camino aumentante en ese primer network auxiliar será de la forma $sxyt$ con $x \in X, y \in Y$, y corresponderá a agregar xy al matching.
- Asi que la primera parte simplemente consiste en buscar un matching inicial agregando todos los lados que se pueda sin romper la condición de matching.
- Y esto es equivalente a correr Dinitz en el primer network auxiliar.
- Y luego de esa primera parte, se trata de ir extendiendo el matching inicial.

Representación de un grafo bipartito

- Veamos un ejemplo. Pero antes hablemos un poco de cómo representaremos grafos bipartitos para estos problemas.
- Sea G grafo bipartito con partes X e Y.
- Si bien en general es mas eficiente representar la estructura de vecinos de un grafo usando una lista o un array de vécinos, en muchos de estos problemas en realidad estan muchos de los vértices de X unidos a muchos de los vértices de Y.
- Asi que en realidad en estos problemas es común usar una matriz de adyacencia en vez de una lista o array.

Representación de un grafo bipartito

- Como G es bipartito, una matriz de adyacencia para G, va a “lucir" como:

$$\begin{bmatrix} 0 & A \\ A^t & 0 \end{bmatrix}$$

- puesto que no hay lados entre vértices de X o vértices de Y.
- Asi que en vez de dar toda la matriz de adyacencia se suele dar solamente A, cuyas filas representan elementos de X y cuyas columnas elementos de Y.

Representación de un grafo bipartito

- En el caso de un grafo sin pesos, las entradas son 1 o 0, indicando simplemente si x es vécino de y o no.
- En el caso de un grafo con pesos en los lados (que veremos mas adelante) se suele poner en la entrada (x, y) el peso del lado xy si existe un lado entre x e y, o bien ∞ si no existe lado.
- Pej, podria ser una matriz asi:

$$\begin{bmatrix} 1 & 7 & \infty & 4 \\ \infty & 5 & \infty & 7 \\ 4 & 7 & 4 & 8 \\ 9 & 4 & 7 & 5 \end{bmatrix}$$

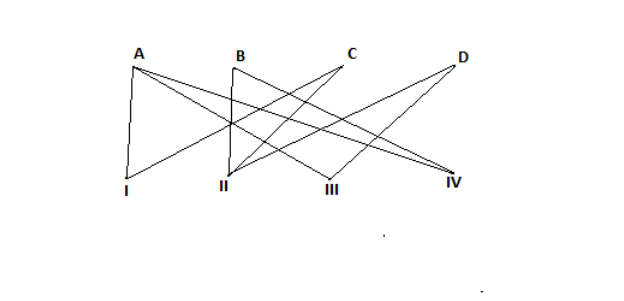
Ejemplo

- Veamos un ejemplo del algoritmo.
- Sea G bipartito con $X = \{A, B, C, D\}$, $Y = \{I, II, III, IV\}$ dado por el siguiente grafo:
- Lo podemos representar por la matriz con filas A, B, C, D y columnas, I, II, III, IV:

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

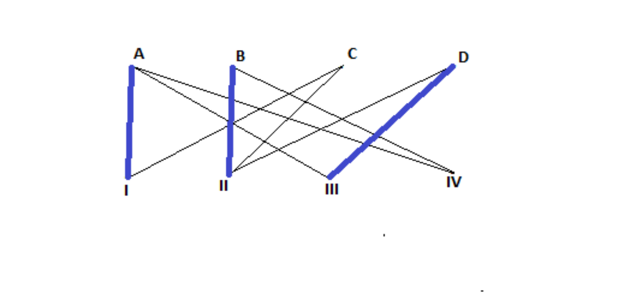
Ejemplo

- Un matching inicial, simplemente mirando



Ejemplo

- Un matching inicial, simplemente mirando se puede construir como *AI*, *BII*, *DIII*.



- No podemos asignarle a *C* porque sus unicos vécinos son I y II, y ya estaban en el matching con A y B.

Ejemplo

- En este ejemplo tan chico, es obvio como “arreglar" esto “a ojo", pero estamos queriendo ejemplificar el algoritmo, asi que sigamos.
- Supongamos que usamos Edmonds-Karp.
- Recordemos que al construir la cola ibamos guardando tanto quien fue el vértice que pone al nuevo vértice en la cola, como el “$\epsilon(x)$" que vamos teniendo hasta ese momento.
- Pero en nuestro caso, como todas las capacidades son uno, esto último es irrelevante y no hace falta hacerlo.

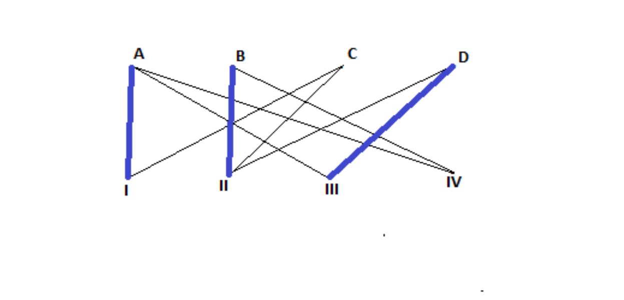
Ejemplo

- Asi que la cola empieza:

  sC

  s

- Pues al estar A, B, D en el matching es lo mismo que decir que $f(\overrightarrow{sA}) = f(\overrightarrow{sB}) = f(\overrightarrow{sD}) = 1$ asi que sólo queda C para enviarle flujo.

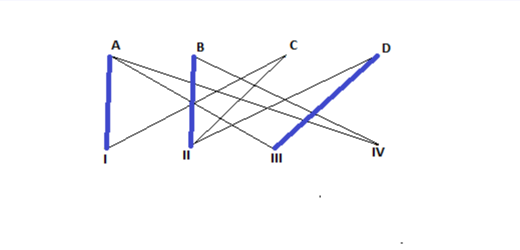


Ejemplo

- C tiene vécinos a I, II
- Asi que la cola sigue:

$$\begin{array}{c} sC\ I\ II \\ s\,CC \end{array}$$

- Tanto I como II está en el matching, asi que $f(\overrightarrow{It}) = f(\overrightarrow{IIt}) = 1$ y ninguno puede agregar a t a la cola.

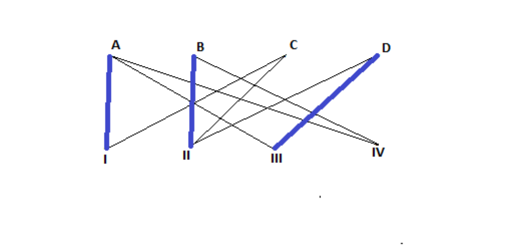


Ejemplo

- Pero justamente, como están en el matching, ambos tienen un vértice en X con el cual están matcheados.
- A en el caso de I, B en el caso de II.
- Esto dice que $f(\overrightarrow{AI}) = f(\overrightarrow{BII}) = 1$ y por lo tanto I puede agregar a A backward y II puede agregar a B backward.
- La cola queda:

$$\begin{array}{cccccc} s & C & I & II & A & B \\ & s & C & C & I^- & II^- \end{array}$$

Ejemplo



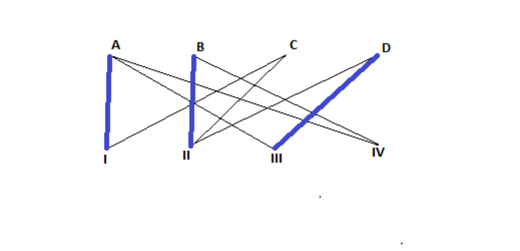
- A tiene vecinos a I,III,IV.

Ejemplo

- A tiene vecinos a I,III,IV.
- El primero ya está en la cola pero los otros dos no, asi que se agregan:

s	C	I	II	A	B	III	IV
	s	C	C	I^-	II^-	A	A

Ejemplo



- A tiene vecinos a I,III,IV.
- El primero ya está en la cola pero los otros dos no, asi que se agregan:
- B tiene a II, IV como vecinos,

Ejemplo

- A tiene vecinos a I,III,IV.
- El primero ya está en la cola pero los otros dos no, asi que se agregan:

s	C	I	II	A	B	III	IV
	s	C	C	I^-	II^-	A	A

- B tiene a II, IV como vecinos, ambos están, no se agrega nada.

Ejemplo

- III forma parte del matching, lo que significa, como vimos en el caso de I,II, que sólo puede agregar (backwards) al vértice de X con el cual forma matching, en este caso, D:

$$\begin{array}{l} \mathrm{sC\ I\ II\ A\ B\ \ III\ IV\ \ \ D} \\ \mathrm{s\,C\,C}I^-\,II^-\,\mathrm{A\ \ A}\,III^- \end{array}$$

- IV no forma parte del matching.
- Esto significa que $f(\overrightarrow{IVt}) = 0$ y puede agregar a t a la cola, con lo que terminamos.

$$\begin{array}{l} \mathrm{sC\ I\ II\ A\ B\ \ III\ IV\ \ \ D\ \ t} \\ \mathrm{s\,C\,C}I^-\,II^-\,\mathrm{A\ \ A}\,III^-\,\mathrm{IV} \end{array}$$

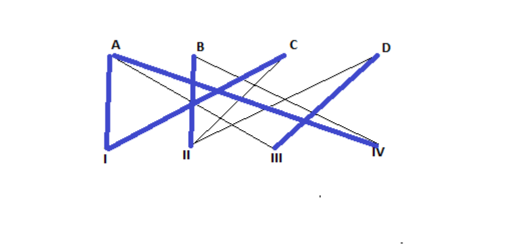
- Y reconstruimos el camino como $sC\overleftarrow{IA}IVt$

Ejemplo

- $sC\overleftarrow{IA}IVt$
- Cambiando el flujo de acuerdo con ese camino, vemos que tenemos que sumarle 1 a los lados $\overrightarrow{sC}, \overrightarrow{CI}, \overrightarrow{AIV}, \overrightarrow{IVt}$ y restarle 1 al lado $\overrightarrow{AI}$.
- Esto ¿cómo se traduce en términos del matching?
- $\overrightarrow{sC}, \overrightarrow{IVt}$ son sólo auxiliares de la construcción que hemos hecho.
- Asi que lo importante es que le sumamos 1 a $\overrightarrow{CI}, \overrightarrow{AIV}$ y restamos 1 al lado $\overrightarrow{AI}$.
- Que significa que agregamos CI, AIV al matching y eliminamos el lado AI del matching.

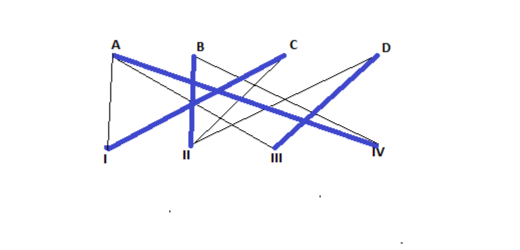
Ejemplo

- Agregamos CI, AIV al matching:



Ejemplo

- Agregamos CI, AIV al matching y eliminamos el lado AI del matching:



- El matching queda AIV, BII, CI, $DIII$ y claramente es maximal.

Estructura general de los caminos aumentantes

- Algo similar a esto, por la estructura del grafo, es algo que va a pasar siempre:
- El camino aumentante será de la forma $sx_{i_1}\overleftarrow{y_{i_1}x_{i_2}}\,\overleftarrow{y_{i_2}x_{i_3}}\ldots\overleftarrow{y_{i_{r-1}}x_{i_r}}y_{i_r}t$.
- Por eso se le suele llamar un camino alternado (porque alterna entre elementos de X y elementos de Y)
- Y respecto del matching significará que:
  1. Se agregarán al matching los lados $x_{i_j}y_{i_j}$, $j = 1, 2, ..., r$
  2. Se borrarán del matching los lados $x_{i_j}y_{i_{j-1}}$, $j = 2, ..., r$
- Como se agregan r lados y se borran $r - 1$, el matching aumenta en un lado y se continua de esta forma hasta que no se pueda seguir mas.

Construcción de los caminos aumentantes

- Una cosa que se puede hacer al hacerlo a mano es ir escribiendo en la misma matriz tanto el matching (marcando por ejemplo con un circulo las entradas) como las etiquetas de la cola, si se usa Edmonds-Karp, en cada fila o columna correspondiente.
- En vez de marcar con un circulo, aca le cambiare el color a los elementos del matching.
- Matriz era:

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Construcción de los caminos aumentantes

- Una cosa que se puede hacer al hacerlo a mano es ir escribiendo en la misma matriz tanto el matching (marcando por ejemplo con un circulo las entradas) como las etiquetas de la cola, si se usa Edmonds-Karp, en cada fila o columna correspondiente.
- En vez de marcar con un circulo, aca le cambiare el color a los elementos del matching.
- Matching inicial AI,BII,DIII:

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Construcción de los caminos aumentantes

- Ahora empezamos la cola, marcando con una “s” las filas no matcheadas:

Construcción de los caminos aumentantes

- Ahora empezamos la cola, marcando con una “s” las filas no matcheadas:

$$\begin{array}{cc} \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} & \begin{array}{c} \\ \\ s \\ \\ \end{array} \end{array}$$

- Recorremos esa fila, buscanco vecinos, es decir, “1”.
- Encontramos las columnas *I*, *II*.
- Esto es lo mismo que antes cuando poniamos:

Construcción de los caminos aumentantes

- Ahora empezamos la cola, marcando con una “s” las filas no matcheadas:

$$\begin{array}{cc} \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} & \begin{array}{c} \\ \\ s \\ \\ \end{array} \end{array}$$

- Recorremos esa fila, buscanco vecinos, es decir, “1”.
- Encontramos las columnas *I*, *II*.
- Esto es lo mismo que antes cuando poniamos:

sC I II
sCC

Construcción de los caminos aumentantes

- Ahora empezamos la cola, marcando con una “s” las filas no matcheadas:

$$\begin{array}{cc} \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} & \begin{array}{c} \\ \\ s \\ \\ \end{array} \end{array}$$

- Recorremos esa fila, buscanco vecinos, es decir, “1”.
- Encontramos las columnas I, II.
- Esto es lo mismo que antes cuando poniamos:
- Pero en esa cola, marcabamos con una C a I,II. ¿Cómo hacemos ahora?

Construcción de los caminos aumentantes

- Simplemente marcamos las columnas I y II con “C”

$$\begin{array}{cccc c}
\left[\begin{array}{cccc} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{array}\right] & s \\
\begin{array}{cccc} C & C & & \end{array} &
\end{array}$$

- En la siguiente etapa de la cola, mirabamos ambos vértices I,II, veiamos que no podiamos llegar a t, pero podiamos devolver flujo.
- En la matriz es equivalente a revisar la columna correspondiente y ver si hay “1” marcado o no.
- Si no hay, es porque la columna no forma parte del matching y podemos extender el matching.
- Si hay un “1” marcado, podemos agregar la fila donde esta ese 1 a la cola.

Construcción de los caminos aumentantes

- En este caso en las dos columnas hay 1 marcados, asi que agregamos las filas correspondientes a la cola.

Construcción de los caminos aumentantes

- En este caso en las dos columnas hay 1 marcados, asi que agregamos las filas correspondientes a la cola. Previamente lo hicimos asi:

$$\begin{array}{ccccc} s & C & I & II & A \quad B \\ & s & C & C & I^{-} \; II^{-} \end{array}$$

Construcción de los caminos aumentantes

- En este caso en las dos columnas hay 1 marcados, asi que agregamos las filas correspondientes a la cola. Ahora:

$$\begin{array}{cc} \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} & \begin{array}{c} \\ \\ s \\ \\ \end{array} \\ \begin{array}{cccc} C & C & & \end{array} & \end{array}$$

Construcción de los caminos aumentantes

- En este caso en las dos columnas hay 1 marcados, asi que agregamos las filas correspondientes a la cola. Ahora:

$$\begin{array}{cc} \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} & \begin{array}{c} I \\ \\ s \\ \\ \end{array} \\ \begin{array}{cccc} C & C & & \end{array} & \end{array}$$

Construcción de los caminos aumentantes

- En este caso en las dos columnas hay 1 marcados, asi que agregamos las filas correspondientes a la cola. Ahora:

$$\begin{array}{cc} \begin{bmatrix}1&0&1&1\\0&1&0&1\\1&1&0&0\\0&1&1&0\end{bmatrix} & \begin{array}{c} I\\ II\\ s\\ \end{array} \\ \begin{array}{cccc} C & C & & \end{array} & \end{array}$$

- Ahora revisamos esas dos filas, buscando 1s y etiqueteando las colummnas correspondientes, si no estan ya etiqueteadas. (pues si lo estan, es que ya han sido puestas en la cola):

Construcción de los caminos aumentantes

- En este caso en las dos columnas hay 1 marcados, asi que agregamos las filas correspondientes a la cola.
- Ahora revisamos esas dos filas, buscando 1s y etiqueteando las colummnas correspondientes, si no estan ya etiqueteadas. (pues si lo estan, es que ya han sido puestas en la cola):

$$\begin{array}{cc}\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} & \begin{matrix} I \\ II \\ s \\ \end{matrix} \\ \begin{matrix} C & C & A & A \end{matrix} & \end{array}$$

Construcción de los caminos aumentantes

- La columna IV ha sido etiqueteada y no tiene 1 marcados.

$$\begin{array}{cc} \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} & \begin{matrix} I \\ II \\ s \\ \end{matrix} \\ \begin{matrix} C & C & A & A \end{matrix} & \end{array}$$

- Podemos extender el matching entonces.
- Nos fijamos en la etiqueta que tiene la columna “libre” , A en este caso.

Construcción de los caminos aumentantes

$$\begin{array}{cc} \begin{bmatrix} {\color{blue}1} & 0 & 1 & 1 \\ 0 & {\color{blue}1} & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & {\color{blue}1} & 0 \end{bmatrix} & \begin{matrix} I \\ II \\ s \\ \end{matrix} \\ \begin{matrix} C & C & A & {\color{blue}A} \end{matrix} & \end{array}$$

- Esto representa la etiqueta en la cola que dice quien lo puso en la cola.
- Por lo que el lado $\overrightarrow{AIV}$ formará parte del camino y por lo tanto mandaremos flujo por el.
- Es decir, agregaremos AIV al matching.

Construcción de los caminos aumentantes

$$\begin{array}{c c}
\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} & \begin{array}{c} I \\ II \\ s \\ \end{array} \\
\begin{array}{cccc} C & C & A & A \end{array} &
\end{array}$$

- Agregamos *AIV* al matching.
- Esto hace que la fila A tenga ahora dos 1 marcados.
- Borramos el “otro” 1, que podemos ver por la etiqueta del costado.
- Esto deja libre la columna I, asi que miramos su etiqueta, C .
- Agregamos el 1 correspondiente al matching y listo.

Construcción de los caminos aumentantes

- Veremos mas ejemplos la clase que viene.
- Pero la idea básica es esa: construir un matching inicial, luego para extenderlo en cada caso vamos revisando filas etiqueteadas, etiqueteando las columnas donde hay 1s, y luego miramos columnas etiqueteadas, mirando si hay 1s marcados. Si los hay, etiqueteamos las filas correspondientes, y si no, terminamos.
- Si tenemos una columna libre, le miramos la etiqueta, agregamos el 1 correspondiente al matching, liberamos el "otro"1, esto deja una columa libre, repetimos hasta llegar a una fila marcada con "s" , con lo cual habremos extendido el matching en un lado
- Seguimos hasta que no podemos mas.
- En el ejemplo terminamos matcheando todas las filas, pero no siempre sucederá eso.

Matchings perfectos y completos

- Problemas relacionados con el de matching maximal son:
  1. ¿Existe un matching que “cubra" todos los vértices?
     - (Es decir, que el conjunto de vértices del matching sea igual al conjunto de vértices de G)
     - Un matching que cubra todos los vértices se dice perfecto
  2. ¿Existe un matching que “cubra" todos los vértices de una de las partes del grafo?
     - Pej. si cubre todos los vértices de X, se dice que es completo sobre X, o completo de X a Y.

Matchings perfectos y completos

- Si X e Y son las partes del grafo es obvio que una condición necesaria para que exista un matching perfecto es que $|X| = |Y|$.
- Tambien es obvio que una condición necesaria para que un matching sea completo sobre una de las partes, pej X, es que $|X| \leq |Y|$, porque si vamos a cubrir todos los vértices de X, necesitamos para cada uno de ellos un vértice en Y correspondiente.
- Obviamente, un matching completo sobre X y completo sobre Y es perfecto.
- Ademas, si $|X| = |Y|$, cualquier matching completo, sobre X o Y, es perfecto.

Matchings perfectos y completos

- Pej, si tenemos tareas y trabajadores que hagan las tareas, un matching perfecto es un matching en el cual todas las tareas se hacen y todos los trabajadores tienen un trabajo.
- Un matching completo sobre los trabajadores es un matching tal que ningún trabajador queda sin empleo, pero podria quedar alguna tarea sin realizar.
- Un matching completo sobre las tareas es un matching tal que todas las tareas se realizan pero puede quedar algún trabajador sin empleo.

Matchings perfectos y completos

- Observar que no podemos elegir entre tener un matching completo sobre los trabajadores o tener uno completo sobre las tareas, pues hay condiciones estructurales que nos pueden impedirlo:
  - si la cantidad de trabajadores es mayor que la cantidad de tareas, no puede haber un completo sobre los trabajadores
  - si la cantidad de tareas es mayor que la cantidad de trabajadores, no puede haber un completo sobre las tareas
  - mientras que si la cantidad de trabajadores y tareas son iguales, todo completo sobre las tareas será perfecto y completo sobre los trabajadores.

Matchings perfectos y completos

- Ahora bien, aunque la cantidad sea igual, podria suceder que no exista un matching perfecto, o en el caso de no ser iguales, que no exista un completo sobre ninguna de las 2 partes.
- Por ejemplo, si tenemos 2 tareas A y B y 10 trabajadores, pero ninguno de ellos puede hacer la tarea B, entonces no puede haber un matching completo sobre las tareas. (ni sobre los trabajadores)
- Para resolver estos problemas simplemente hallamos un matching maximal y chequeamos si es completo o perfecto.

Matchings perfectos y completos

- Pero ademas se pueden probar cosas teóricas que se usan bastante.
- Vimos que la condición $|X| \leq |Y|$ es necesaria para que exista un matching completo sobre X, pero con el ejemplo anterior vimos que no es suficiente.
- Esto es porque no es suficiente que la cardinalidad de X sea menor que la de Y, sino que necesitamos que existan suficientes vecinos para los elementos de X.
- En el ejemplo anterior, B no tenia ningún vecino.

Matchings perfectos y completos

- Pero aún si todos los vértices tienen vecinos, podria no haber un matching completo.
- Por ejemplo, supongamos que tenemos $X = \{A, B, C, D, E\}$, $Y = \{1, 2, 3, 4, 5, 6, 7\}$ y lados $E = \{A1, A4, B2, B3, B4, B6, B7, C1, C5, D1, D4, D5, E4, E5\}$.
- Si queremos encontrar un matching completo sobre X no podriamos.
- Analizemos rapidamente porqué:

Matchings perfectos y completos

- Miremos el subconjunto $S = \{A, C, D, E\}$.

Matchings perfectos y completos

- Miremos el subconjunto $S = \{A, C, D, E\}$.
- La unión de todos los vécinos de los vértices de S es $\{1, 4, 5\}$. ($E = \{A1, A4, B2, B3, B4, B6, B7, C1, C5, D1, D4, D5, E4, E5\}$)

Matchings perfectos y completos

- Miremos el subconjunto $S = \{A, C, D, E\}$.
- La unión de todos los vécinos de los vértices de S es $\{1, 4, 5\}$.
- Por lo tanto no hay suficiente cantidad de vécinos para el conjunto S, lo cual dice que S no puede ser cubierto, mucho menos X.

Condición de Hall

- Dado un subconjunto S de vértices definimos

$$\Gamma(S) = \cup_{x \in S}\Gamma(x) = \{z : \exists x \in S : xz \in E\}$$

- Si hubiese un matching completo sobre X, y $S \subseteq X$, tendriamos que restringiendo el matching a S deberiamos tener al menos $|S|$ vécinos de los elementos de S.
- Y esos vecinos estan todos en $\Gamma(S)$.
- Por lo tanto una condición necesaria para que exista matching completo sobre $|X|$ es que $|S| \leq |\Gamma(S)|$ para todo $S \subseteq X$.
- Esto se llama la condición de Hall.
- Y resulta que no sólo es necesaria sino suficiente.

Teorema de Hall

Teorema (Hall)

Si G es un grafo bipartito con partes X e Y y $|S| \leq |\Gamma(S)|$ para todo $S \subseteq X$, entonces existe un matching completo sobre X.

- Prueba:
- Supongamos que el teorema no es cierto, es decir que $|S| \leq |\Gamma(S)|$ para todo $S \subseteq X$ pero que no existe un matching completo sobre X.
- Probaremos que el hecho de que no exista un matching completo sobre X implica que existe $S \subseteq X$ con $|S| > |\Gamma(S)|$ lo cual contradice la hipotesis.

Prueba del Teorema de Hall

- La prueba de que existe tal S la haremos analizando el algoritmo que dimos antes.
- Como estamos suponiendo que no existe matching completo sobre X, entonces corriendo el algoritmo para hallar matching maximal, usando Edmonds-Karp, obtenemos un matching maximal que NO cubre todos los vértices de X.
- Sea S_0 aquellos vértices de X que, al terminar el algoritmo, han quedado sin cubrir. ($S_0 \neq \emptyset$ porque estamos suponiendo que no hemos cubierto a todos los vértices de X)

Prueba del Teorema de Hall

- Entonces $x \in S_0$ si y solo si no existe $y \in Y$ tal que xy sea lado del matching.
- Esto significa que $f(\overrightarrow{xy}) = 0$ para todo y, donde f es el flujo maximal encontrado.
- Por lo tanto $out_f(x) = 0$ para todo $x \in S_0$.
- Como f es flujo, eso significa que $in_f(x) = 0$ para todo $x \in S_0$.

Prueba del Teorema de Hall

- Viceversa, supongamos que x es un vértice tal que $in_f(x) = 0$.
- Entonces $out_f(x) = 0$ y por lo tanto $f(\overrightarrow{xy}) = 0$ para todo y lo cual significa que x no forma parte del matching.
- Conclusión: $S_0 = \{x \in X : in_f(x) = out_f(x) = 0\}$.
- En el último paso del algoritmo, Edmonds-Karp encuentra un corte minimal.
- Antes denotabamos los cortes con la letra S, pero ahora usaremos otra letra porque el corte minimal no es el "S" que estamos buscando. Digamos que el corte es C.

Prueba del Teorema de Hall

- Ese corte C incluye a s, y los otros elementos de C son vértices de G, que pueden estar en X o en Y.
- Asi que existen $S \subseteq X$, $T \subseteq Y$ tal que $C = \{s\} \cup S \cup T$.
- Recordemos que C se obtiene al hacer la ultima cola BFS: C consiste de todos los vértices que en algún momento estan en esa última cola.
- Como $x \in S_0 \iff in_f(x) = 0 \iff f(\overrightarrow{sx}) = 0$, entonces los vértices de S_0 son precisamente todos los vértices que s agrega a la cola.
- Por lo tanto, $S_0 \subseteq S$.

Prueba del Teorema de Hall

- Ahora bien, todos los otros elementos de S (los que no estan en S_0, si es que hay alguno) deben haber sido agregados a la cola por algún elemento distinto de s.
- Pero un vértice solo puede agregar a algún vecino, ya sea vecino hacia adelante o hacia atras.
- Esto quiere decir que los vértices de S no pueden haber sido agregado por otros vértices de S, pues como $S \subseteq X$, no hay lados entre ellos.
- Asi que los vértices de S que no están en S_0 deben haber sido agregados por vértices de T.

Prueba del Teorema de Hall

- Pero $T \subseteq Y$ y todo vértice y de Y tiene $\Gamma^+(y) = \{t\}$.
- Asi que T sólo puede agregar a un vértice de X a la cola si lo agrega por medio de un lado backward
- Para que y agregue a x como backward, debe pasar que $f(\overrightarrow{xy}) > 0$.
- En nuestro caso, esto implica $f(\overrightarrow{xy}) = 1$ y que xy sea parte del matching.
- Por lo tanto, cada y que pueda agregar a la cola a algún vértice en forma backward, puede agregar UN SÓLO tal vértice.

Prueba del Teorema de Hall

- Entonces cada vértice de T puede agregar un sólo vértice a la cola.
- Pero ademas, cada vértice de T efectivamente agrega un vértice a la cola.
- ¿Por qué?
- Supongamos que no, que exista $y \in T$ que no agrega a nadie a la cola.
- Como C es un corte, $t \notin C$, asi que y no deberia poder agregar a t a la cola.
- Con lo cual, debemos tener que $f(\overrightarrow{yt}) = 1$.
- Entonces $out_f(y) = 1$ y por lo tanto $in_f(y) = 1$ y debe existir $x \in X$ tal que $f(\overrightarrow{xy}) = 1$.
- Pero entonces x es “candidato" a ser agregado a la cola como backward por y.

Prueba del Teorema de Hall

- Como estamos suponiendo que y no agrega a nadie a la cola, debe haber una razón para que y no agregue a x.
- La única razón posible es que x ya esté en la cola.
- Pero ¿quien lo agregó?
- No lo puede haber agregado alguien de T pues si fuese asi:
  - y no lo agrega asi que debería ser otro vértice z de T, agregandolo backward.
  - Pero esto significaria que tanto xz como xy estan en el matching, absurdo.
- Entonces la única posibilidad que queda es que haya sido agregado por s, es decir, que x esté en S_0.

Prueba del Teorema de Hall

- Pero si $x \in S_0$, entonces $out_f(x) = 0$ lo que contradice que $f(\overrightarrow{xy}) = 1$.
- Concluimos entonces que todo y en T agrega a alguien a la cola, y por lo que vimos antes, agrega exactamente un tal alguien a la cola.
- Esos elementos agregados son los de $S - S_0$, asi que concluimos que $|S - S_0| = |T|$.

Prueba del Teorema de Hall

- Analicemos ahora un poco a los vértices de T.
- Todos ellos deben haber sido agregados por alguien de X de la cola, es decir, por alguien de S.
- Esto sólo puede pasar si cada vértice de T es un vécino de algún vértice de S.
- Por lo tanto, $T \subseteq \Gamma(S)$.

Prueba del Teorema de Hall

- Habiendo visto que $T \subseteq \Gamma(S)$, ahora nos preguntamos si son iguales o no.
- Si no son iguales, entonces existe $y \in \Gamma(S)$ tal que $y \notin T$.
- Como $y \in \Gamma(S)$ entonces existe $x \in S$ con $xy \in E$.
- Como $x \in S$, x está en C.
- Pero $y \notin T$, asi que y no está en C.
- Asi que x nunca agregó a y.
- Pero $xy \in E$, asi que para que x no agregue a y, debe pasar que $f(\overrightarrow{xy}) = 1$.

Prueba del Teorema de Hall

- Pero entonces, ¿quien agregó a x?
- y no fue porque nunca estuvo en la cola.
- No puede ser ningún otro elemento de T pues $f(\overrightarrow{xy}) = 1$ implica que no existe ningún otro z con $f(\overrightarrow{xz}) = 1$.
- Debe haber sido agregado por s, pero esto implica que $out_f(x) = 0$ absurdo pues $f(\overrightarrow{xy}) = 1$
- El absurdo vino de suponer que T y $\Gamma(S)$ no eran iguales, asi que concluimos que $T = \Gamma(S)$.

Prueba del Teorema de Hall

- Entonces:

$$|\Gamma(S)| = |T| = |S - S_0| = |S| - |S_0| < |S|$$

- (la ultima desigualdad pues $S_0 \neq \emptyset$.
- Hemos probado que $|\Gamma(S)| < |S|$, lo que contradice la hipotesis de que $|S| \leq |\Gamma(S)|$ para todo $S \subseteq X$.
- Fin

Teorema del matrimonio de Kőnig

- OBSERVACIÓN: A veces al teorema de Hall se le llama “teorema del matrimonio" pero en realidad ese nombre se suele reservar para otro teorema, del húngaro Dénnes Kőnig.

Teorema del matrimonio de Kőnig

- OBSERVACIÓN: A veces al teorema de Hall se le llama "teorema del matrimonio" pero en realidad ese nombre se suele reservar para otro teorema, del húngaro Dénnes Kőnig.

Teorema del matrimonio (Kőnig, 1914)

Todo grafo bipartito regular tiene un matching perfecto.

- Antes de dar la prueba, hay que explicar un poco porqué se llama "teorema del matrimonio".
- El nombre viene de una de las aplicaciones del teorema.

Teorema del matrimonio

- Supongamos que tenemos un conjunto de hombres y mujeres que quieren casarse.
- Nota: como esto fue hecho hace decadas, obvio que no existia el matrimonio homosexual, y por eso vamos a tener un grafo bipartito, de lo contrario tendriamos que usar otro teorema.
- asi que en la epoca actual tendremos que agregar la hipotesis que todos los involucrados son heterosexuales.
- Cada hombre hace una lista de las mujeres con las cuales aceptaria casarse, y cada mujer hace una lista de los hombres con los cuales aceptaria casarse.

Teorema del matrimonio

- Luego para cada mujer x hacemos una lista reducida de hombres compatibles: aquellos hombres y en la lista de x que tienen a x en la lista de y.
- Y hacemos lo mismo para cada hombre y: una lista de mujeres x que estan en la lista de y tales que y está en la lista de x.
- Es decir, x está en la lista reducida de y si y sólo si y está en la lista reducida de x si y solo si x e y son mutuamente compatibles.
- Entonces el teorema dice que si cada hombre es compatible con exactamente k mujeres y cada mujer es compatible con exactamente k hombres, con $k \neq 0$, entonces se pueden casar a todos los hombres y a todas las mujeres de forma compatible.

Teorema del matrimonio

- Si lo prefieren pueden pensar en terminos de tareas/trabajadores que si cada tarea puede ser realizada por exactamente k trabajadores y cada trabajador puede realizar exactamente k tareas, con $k > 0$, entonces se pueden realizar todas las tareas y ningún trabajador se queda sin trabajo.
- En esta formulación podría llamarse el teorema de la distribución de tareas o algo asi, pero bueno, no es asi como se lo conoce.

Prueba del Teorema del matrimonio

- Vayamos a la prueba.
- Dado un conjunto $W \subseteq V$, definamos $E_W = \{zw \in E | w \in W\}$
- Sean X e Y las partes del grafo bipartito.
- Supongamos que $W \subseteq X$. Entonces:
- $|E_W| = |\{zw \in E | w \in W\}| = \sum_{w \in W} |\{z : zw \in E\}|$
- La última igualdad pues como $W \subseteq X$ entonces no estamos contando un lado zw dos veces, pues $zw \in E$ implica que $z \in Y$ y por lo tanto no puede estar en W.

Prueba del Teorema del matrimonio

- Entonces $|E_W| = \sum_{w \in W} |\{z : zw \in E\}| = \sum_{w \in W} d(w)$
- Como G es regular, $d(w) = \Delta$ para todo w, asi que tenemos:
- $|E_W| = \sum_{w \in W} d(w) = \sum_{w \in W} \Delta = \Delta.|W|$.
- Con un razonamiento similar, tenemos que si $W \subseteq Y$ entonces tambien vale que $|E_W| = \Delta.|W|$.
- Apliquemos esta propiedad para diversos W.
- Si tomamos $W = X$ tendremos que $|E_X| = \Delta.|X|$.
- Pero G es bipartito, asi que $E_X = \{xy \in E : x \in X\} = E$
- Concluimos que $|E| = \Delta.|X|$.

Prueba del Teorema del matrimonio

- Tomando $W = Y$ concluimos $|E_Y| = \Delta.|Y|$.
- Pero $E_Y = E$ asi que tenemos que $|E| = \Delta.|Y|$.
- Entonces $|E|$ es igual a $\Delta.|X|$ y a $\Delta.|Y|$.
- Por lo tanto $\Delta.|X| = \Delta.|Y|$ asi que $|X| = |Y|$.
- (observar que estamos usando $\Delta \neq 0$. Esto es asi porque la definición de “bipartito" que dimos el primer dia de clase era que $\chi(G) = 2$, asi que esto implica que hay al menos un lado y $\Delta \neq 0$)
- Entonces, como $|X| = |Y|$, para probar que existe un matching perfecto basta probar que existe un matching completo sobre X.
- Para probar esto, usaremos el teorema de Hall.

Prueba del Teorema del matrimonio

- Sea entonces $S \subseteq X$.
- Sea $\ell \in E_S$.
- Entonces existe $x \in S$, $y \in Y$ tal que ℓ es de la forma $\ell = xy$.
- Por lo tanto $y \in \Gamma(x) \subseteq \Gamma(S)$ (pues $x \in S$).
- Entonces ℓ es de la forma xy con $y \in \Gamma(S)$, lo cual implica que $\ell \in E_{\Gamma(S)}$.
- Como ℓ era cualquier elemento de E_S esto dice que $E_S \subseteq E_{\Gamma(S)}$.
- Por lo tanto $|E_S| \leq |E_{\Gamma(S)}|$

Prueba del Teorema del matrimonio

- Como $S \subseteq X$, entonces lo que probamos al principio de la prueba dice que $|E_S| = \Delta.|S|$.
- Como $\Gamma(S) \subseteq Y$, entonces $|E_{\Gamma(S)}| = \Delta.|\Gamma(S)|$.
- Como en la pagina anterior probamos que $|E_S| \leq |E_{\Gamma(S)}|$, concluimos que $\Delta.|S| \leq \Delta.|\Gamma(S)|$.
- Por lo tanto $|S| \leq |\Gamma(S)|$ y como esto vale para cualquier $S \subseteq X$, Hall nos dice que existe un matching completo sobre X lo cual completa la prueba.

Mas sobre Kőnig

- Kőnig fue un “peso pesado" en teoría de grafos.
- Publicó el primer libro de texto sobre teoría de grafos y tiene varios teoremas en el area.
- Uno de sus teoremas dice que el número de lados de un matching maximal es igual al número de vértices de una cobertura minimal de vértices. (i.e., un conjunto de vértices tal que todo lado tiene a uno de esos vértices como extremo).
- Este teorema es equivalente al teorema de Hall.
- En octubre de 1944, durante el exterminio del 75% de los judios de Hungria, Kőnig, que era judio, se suicidó para evitar caer en las garras de los nazis.

Otro teorema de Kőnig

- En honor de Kőnig, probaremos otro teorema tambien de el.
- Este teorema se puede probar de varias formas pero la mas fácil es usar el teorema del matrimonio.
- Y encaja bastante bien con otros temas del curso.
- Hemos visto coloreo de vértices de un grafo, pero los lados tambien se pueden colorear.
- Un coloreo propio de los lados de un grafo es un coloreo de los lados tal que dos lados que tengan un extremo en común tengan colores distintos.
- El índice cromático de un grafo es la menor cantidad de colores de un coloreo propio de los lados y se denota por $\chi'(G)$.

Coloreos de lados

- Una cota inferior obvia es $\Delta \leq \chi'(G)$.
- Pues si $d(x) = \Delta$, entonces existen Δ lados con extremo x, todos los cuales deben llevar colores distintos.
- Vizing probó en 1964 que una cota superior es....
- $\Delta + 1$ (!!!!)
- Es decir $\Delta \leq \chi'(G) \leq \Delta + 1$ para todo grafo G, asi que sólo hay dos opciones.
- Pero determinar en general si $\chi'(G)$ es Δ o $\Delta + 1$ es NP-completo!
- Para algunos grafos sin embargo, se puede determinar a que categoria pertenecen fácilmente

Teorema de Kőnig

Teorema (Kőnig)

Si G es bipartito, entonces $\chi'(G) = \Delta$

- Prueba.
- Supongamos primero que G es regular y probemoslo por inducción en Δ.
- Si $\Delta = 1$, entonces G es una colección de lados disjuntos, y podemos colorearlos a todos con el color 1.
- Supongamos ahora que $\Delta(G) > 1$ y que el teorema vale para grafos bipartitos regulares con grado máximo $\Delta(G) - 1$.

Teorema de Kőnig

- Usando el teorema del matrimonio, tenemos un matching perfecto, pues G es bipartito regular.
- Sean E_1 los lados de ese matching, y borremoslos de G, obteniendo un grafo $G*$.
- Como son lados de un matching, el grado de cada vértice de G disminuye en 1.
- Asi, $G*$ tambien es regular y $\Delta(G*) = \Delta(G) - 1$ y por hipotesis inductiva, sus lados se pueden colorear con $\Delta(G*)$ colores.
- Dandole a los lados de E_1 un color distinto de esos $\Delta(G*)$ colores, tenemos coloreados todos los lados de G con $\Delta(G*) + 1 = \Delta(G)$ colores, lo cual prueba el paso inductivo.

Teorema de Kőnig

- Para finalizar la prueba en el caso general, se puede probar que todo grafo bipartito G puede incluirse en un grafo bipartito H regular tal que $\Delta(G) = \Delta(H)$.
- Entonces $\chi'(G) \leq \chi'(H)$

Teorema de Kőnig

- Para finalizar la prueba en el caso general, se puede probar que todo grafo bipartito G puede incluirse en un grafo bipartito H regular tal que $\Delta(G) = \Delta(H)$.
- Entonces $\chi'(G) \leq \chi'(H)$
  pues $G \subseteq H$

Teorema de Kőnig

- Para finalizar la prueba en el caso general, se puede probar que todo grafo bipartito G puede incluirse en un grafo bipartito H regular tal que $\Delta(G) = \Delta(H)$.
- Entonces $\chi'(G) \leq \chi'(H) = \Delta(H)$
  pues H es bipartito regular y ya probamos el teorema para bipartitos regulares

Teorema de Kőnig

- Para finalizar la prueba en el caso general, se puede probar que todo grafo bipartito G puede incluirse en un grafo bipartito H regular tal que $\Delta(G) = \Delta(H)$.
- Entonces $\chi'(G) \leq \chi'(H) = \Delta(H) = \Delta(G)$
- Lo que quedaria es probar la existencia de H.
- La idea es ir construyendo grafos cada vez mas grandes tal que el δ crezca pero el Δ se mantenga igual, terminando eventualmente con un grafo con $\delta = \Delta$, es decir, regular.
- Asi que basta probar que existe $\tilde{G}$ con $G \subseteq \tilde{G}$ tal que $\tilde{G}$ es bipartito con $\Delta(G) = \Delta(\tilde{G})$ y $\delta(\tilde{G}) = \delta(G) + 1$.

Teorema de Kőnig

- Si X e Y son las partes de G, definimos X^*, Y^* copias de X, Y respectivamente. (pej, podriamos definir para cada x de X un elemento $x^* = (x, 1)$, y similar para y pero la forma exacta de definir x^* es irrelevante)
- Tomamos $\tilde{X} = X \cup Y^*$, $\tilde{Y} = Y \cup X^*$.
- Sea $\Delta = \Delta(G)$ y:

$$\begin{aligned} \tilde{E} &= E \cup \{x^* y^* : xy \in E\} \cup \{xx^* : d(x) < \Delta\} \cup \\ &\cup \{yy^* : d(y) < \Delta\} \end{aligned}$$

- Queda claro que $\tilde{G}$ es bipartito con partes $\tilde{X}$, $\tilde{Y}$.

Teorema de Kőnig

- Ademas, el grado de cualquier vértice z con $d(z) = \Delta$ no cambia, y el grado del z^* correspondiente tambien es Δ.
- Mientras que para todos los otros vértices, el grado sube exactamente en uno, pues existe un lado extra zz^*.
- Fin.