MATH 340 - Discrete Structures 2

McGill University - Winter 2013

Last Updated: April 15, 2013

Contents

Information

- Instructor: Bruce Shepherd
- LaTeX: Ehsan Kia
- Notes: Catherine Hilgers

1 Summary of Graph Theory Terms

A (simple) graph G is an ordered pair $(V(G), E(G))$, sometimes written (V, E), where $V(G)$ is a finite set of vertices (aka nodes), and $E(G)$ is a finite set of edges.

Each edge is of the form $\{u, v\}$ sometimes written uv, where $u \neq v$ are two vertices that are the end points of the edge. An edge $e \in E$ is *incident* to a vertex $v \in V$ if $e = (u, v)$ for some $u \in V$. A vertex $v \in V$ is called *adjacent* to a vertex $u \in V$ if $(u, v) \in E$.

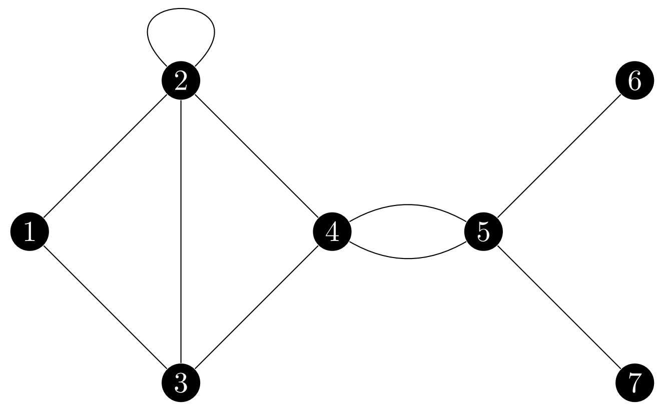
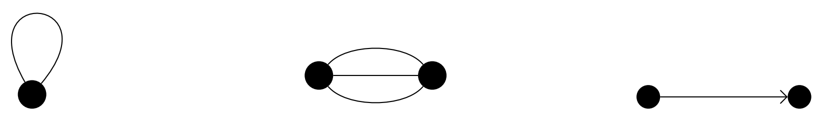


Figure 1: Example of a graph where $V = \{1, 2, 3, 4, 5, 6, 7\}$

NOTE: simple, undirected graph mean that we have no:



(a) Loops

(b) Multiple edges

(c) Direction

Suppose we have $H = (V(H), E(H))$ such that:

i) $V(H) \subseteq V(G)$

ii) $E(H) \subseteq E(G)$

iii) $\forall\ e = (u, v) \in E(H) : u, v \in V(H)$

Then, H is a *subgraph* of G.

Given a set $S \subset V$, we define the *subgraph induced by* S to be the graph denoted by $G[S]$ to be a subgraph of G whose vertex set is S and whose edge set is the set of edges with both ends in S.

Similarly, for $F \subset E$, define the subgraph induced by F, denoted $G[F]$, to be the subgraph of G whose edge set is F and whose vertex set is the set of all endpoints in F.

1.1 Terminology

The *degree* of a vertex v is the number of edges of which it is an endpoint, denoted by $deg_G(v)$.

A *walk* of a graph G is a sequence of alternating vertices and edges $v_0 e_1 v_1 e_2 ... v_{n-1} e_n v_n$ such that e_i is incident to v_{i-1} and v_i, $\forall\ i = 1, ..., n$, where n is the length of the walk.

A *trail* is a walk in which the edges are distinct.

A *path* is a trail in which vertices are distinct.

A *cycle* is a trail of length at least 1 in which the vertices are distinct, except v_0 and v_n which are the same.

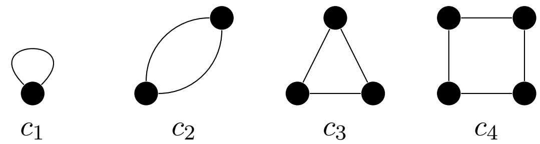
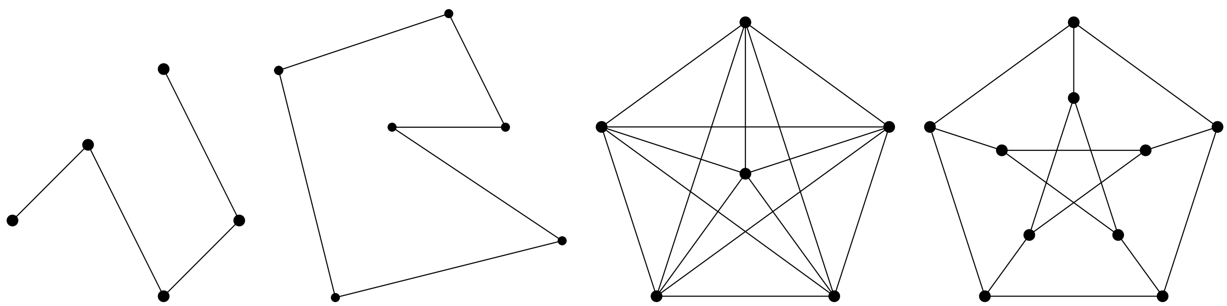


Figure 2: Cycles of size 1 to 4.

A graph is *connected* if $\exists$ a path between any two vertices. Else, it's disconnected.

A *component* of G is a maximal connected subgraph.

1.2 Special Graphs



(a) Path (b) Cycle (c) Complete graph (d) Petersen graph

Figure 3: Examples of simple graphs

A *tree* is a connected graph with no cycles (Figure 4).

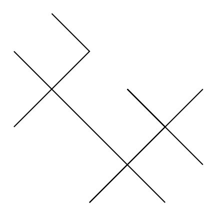


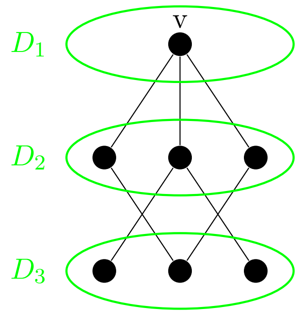
Figure 4: A tree

A graph G is *bipartite* if $\exists$ a partition (X, Y) of $V(G)$ such that for every edge $e \in E(G)$, e has one endpoint in X and the other in Y. X and Y are called the *parts* of G and (X, Y) is called the bipartition.

Theorem: G is bipartite $\Leftrightarrow$ G contains no odd cycles.

Proof: Without loss of generality, assume G is connected, since G is bipartite $\Rightarrow$ each of its components are.

($\Rightarrow$) Suppose G is bipartite, with bipartition (X, Y). Let $v_0e_1v_1e_2...e_nv_n$ be an odd cycle (n is odd). Assume $v_0 \in X$. We then show that for $0 \leq k < \frac{n}{2}$, $v_{2k} \in X$. Assume inductively that $V_{2k-2} \in X$, where $k \geq 1$. Then v_{2k-1} lies in Y, since e_{2k-1} has endpoints in both X and Y. But v_{2k-1} inplies $v_{2k} \in X$ for the same reason. In particular, $v_{n-1} \in X$, but this means the two endpoints of e_n, v_0 and v_{n-1}, both lie in X. This contradict the fact that G is bipartite.



($\Leftarrow$) Suppose G contains no odd cycles. Let $v \in V$, and for all $u \in V$, define $d(v)$ = length of the shortest path from u to v. Let $D_i = \{u \in V : d(u) = i\}$.

Claim 1: $j \geq i+2 \Rightarrow$ there are no edges with endpoints in D_i or D_j.

Claim 2: any $i \geq 0$, there are no edges with both endpoints in D_i.

Then, letting $X = \bigcup_{i \text{ even}} D_i$ and $Y = \bigcup_{i \text{ odd}} D_i$, then (X, Y) forms a bipartition of G.

Proof of claim 1: Suppose there were some vertices u, v, and integers i, j, such that $j \geq i+2$, $u \in D_i$, $w \in D_j$, and $uw \in E$. Then, a shortest path from v to w is no longer than the path by adjoining uw to the shortest path from v to u. So, $d(w) \leq i+1$. This contracting the fact that $w \in D_j$, that is, $d(w) \geq qi+1$.

Proof of claim 2: Suppose there were some $i \geq 0$ and vertices $u, w \in D_i$ such that $uw \in D_i$. Then, $\exists$ two paths: $P_1 = (v = a_0, a_1, a_2, ..., a_{i-1}, u = a_i)$ and $P_2 = (v = b_0, b_1, b_2, ..., b_{i-1}, w = b_i)$. Let m be thte largest index such that $a_k \neq b_k \; \forall \; m+1 \leq k \leq i$. Then, $a_m a_{m+1} ... a_{i-1} u w b_{i-1} ... b_{m+1} b_m$ is a cycle of length 2(i-m)+1, which is odd. $\Rightarrow\Leftarrow$.

■

2 Stable Marriages

We have n boys and n girls. Each boy has an ordered list of girls and vice versa.

A set M of marriages is *stable* if there is no boy-girl pair who prefer each other to their current pairings in M. We call this situation an unstable (unblocking) pair [Figure 5].

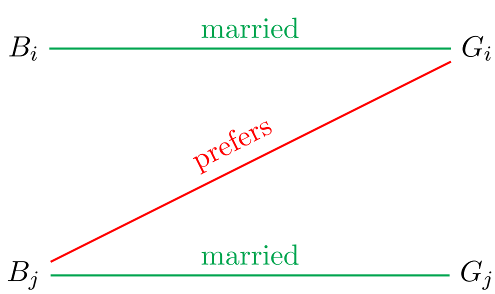


Figure 5: **Unstable pair** B_j prefers G_i to G_j and G_i prefers B_j to B_i

2.1 Example

In the following example [Figure 6], we have 3 boys and 3 girls, each with their own preference list, but the given matching isn't a stable marriage.

But when trying again, we can easily find two stable configuations [Figure 7]

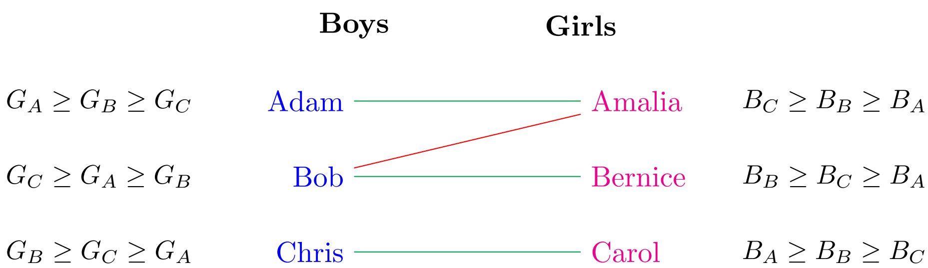


Figure 6: Unstable because Amalia and Bob prefer each other over their current partner

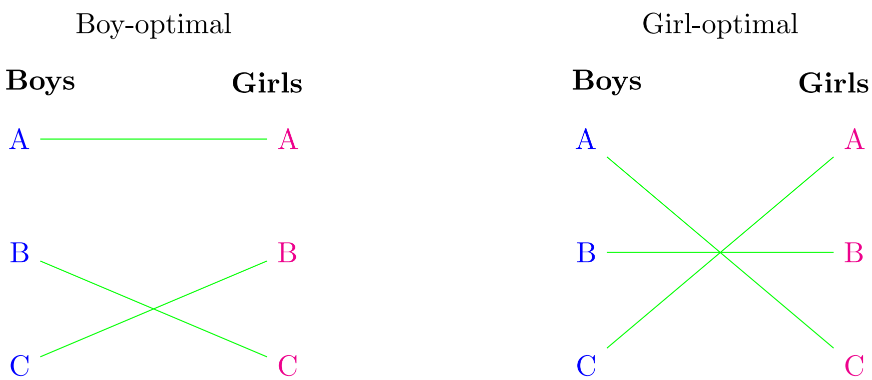


Figure 7: These work because each boy prefers a different girl, and each girl prefers a different boy.

2.2 Gale-Shapley

Do stable matchings exist in general?

Theorem (Gale & Shapley): A stable matching always exists

Proof (by algorithm): While there is some "single" boy B, B proposes to the next girl on his list, call her G. Girl G accepts if she is single or prefers B to her current fiancé. Claim is that the algorithm terminates for any set of lists with a stable matching.

NOTE: as the algorithm proceeds, girls' choices only get better and mens' only get worse. Each time a girl changes fiancè, she trades up. A boy only changes if he gets dumped by G and he then proposes to the next girl on his list.

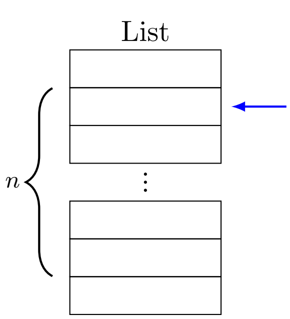


Figure 8: Preference list

Corollary: The algorithm terminates. Say boy B has his list. The pointer aims at his current match. There are n boys and n possible pointers into their lists [Figure 8]. each dumping moves the pointer down the list by one. We have $\leq n^2$ total dumpings. The algorithm terminates after $\mathcal{O}(n^2)$.

The matching returned by the algorithm is stable. Suppose M is the output matching, and has unstable pair (B_i, G_j), for a contradiction:

- B_i prefers G_j to current match G_i

- G_j prefers B_i to current match B_j

Since B_i prefers G_j to G_i, he proposed to her earlier and she either rejected him, or accepted and dumped him later. In either case, she was at some point matched to some B_k she preferred to B_i. By observation, her partners only improved from that point on. Thus, she prefers B_j to B_k and B_k to B_i $\Rightarrow$ prefers B_j to B_i and (B_i, G_j) is not unstable. $\Rightarrow\Leftarrow$ (contradiction)

There can be many stable matchings. Let:

$$\mathcal{S} = \{M_1, M_2, ..., M_k\}$$

be the set of all stable matchings. Call G_j a *valid partner* for B_i if (B_i, G_j) are matched in some $M_i \in \mathcal{S}$. For each B, let $G^+(B)$ be his most preferred valid partner.

Remarkably, the boy-proposal algorithm matches each boy B to $G^+(B)$. To show this, we require a lemma:

Lemma: a girl never rejects a valid partner

Proof (by contradiction): Suppose not. Consider the first time G_j rejects a valid partner B_i. Say (B_i, G_j) were matched in $M_t \in \mathcal{S}$. Say G_j dumps B_i for B_j at that time. Say (B_j, G_k) is a match in M_t [Figure 9].

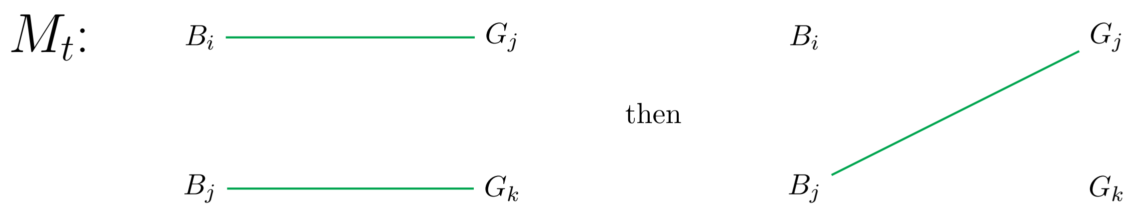


Figure 9: A valid partner being dumped by a girl in boy-proposal

Since B_i is the first valid partner to be dumped, we claim B_j prefers G_j to G_k. Why? Supposed B_j prefers G_k to G_j. Thus he proposes first to G_k. But $(G_k, B_j) \in M_t$, and therefore G_k is valid for B_j. But B_j was as we supposed in the beginning the first valid person to be dumped, which means B_j did not get dumped and $B - j$ is not free to propose to G_j. $\Rightarrow\Leftarrow$

So B_j prefers G_j to G_k and G_j prefers B_j to B_i, therefore (B_i, G_j) is unstable in M_t. But $M_t \in \mathcal{S}$ and in thus stable. $\Rightarrow\Leftarrow$. Hence a girl never rejects a valid partner.

$\blacksquare$

Now we will show that the boy-proposal algorithm matches each boy B with $G^+(B)$.

Proof: If B_i is matched by algorithm to G_j, who he doesn't like as much as $G^+(B_i)$, then he proposed to $G^+(B_i)$ first. But $G^+(B_i)$ and B_i are valid, hence $G^+(B_i)$couldn't have rejected him. $\Rightarrow\Leftarrow$

Let $B^-(G_j)$ be the worst partner for G_j amongst all stable matchings.

Lemma: The boy-proposal algorithm matches each G_j to $B^-(G_j)$.

Proof: Supposed B_j and G_j are matched, whom she prefers to $B^-(G_j)$. Say $(G_j, B^-(G_j)) \in M_r$ and $(G_i, B_j) \in M_r$ [Figure 10].

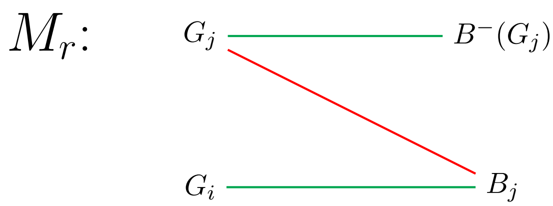


Figure 10: By the previous, B_j gets $G^+(B_j)$, so $G_j = G^+(B_j)$

Thus, B_j prefers G_j to G_i and G_j prefers B_j to $B^-(G_j)$, therefore the valid pair (B_i, G_j) is unstable in M_r. $\Rightarrow\Leftarrow$. It follows that G_j gets $B^-(G_j)$ with boy proposal.

■

2.3 Matching

A *matching* in a graph $G(V, E)$ is a set $M \subseteq E$ of vertex-disjoint edges, i.e., each vertex of G is the endpoint of at most one edge in M.

we say $v \in V$ is *matched* (or *saturated*) by M if it is the endpoint of some edges in M. Otherwise, it is *unmatched*. A path P is *M-alternating* if its edges are alternatively in M and not in M.

An alternating path is *M-augmenting* if its endpoints are unmatched.

Theorem: A matching in G is of maximum cardinality $\iff$ there is no M-augmenting path.

Proof:($\Rightarrow$) Suppose P is an M-augmenting path, then switching the edges in P produces a larger matching. Let $M' = M \oplus E(P)$ (Symmetric difference of M and the edges in the path P).

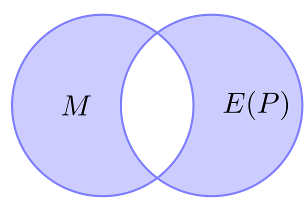
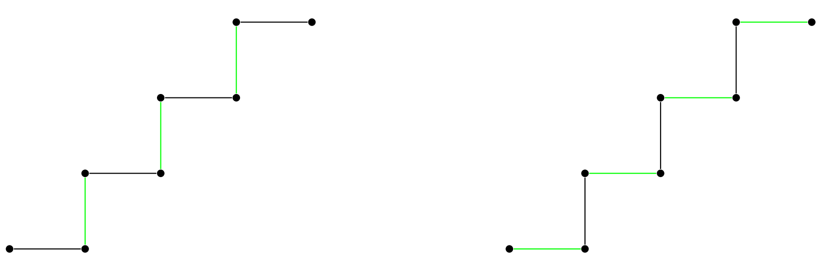


Figure 11: Symmetric difference of M and $E(P)$

$$
\begin{aligned}
M \oplus E(P) &= (M \cup E(P)) - (M \cap E(P)) \\
&= (M - E(P)) \cup (E(P) - M)
\end{aligned}
$$



(a) M has matching of size 3

(b) M' has matching of size 4

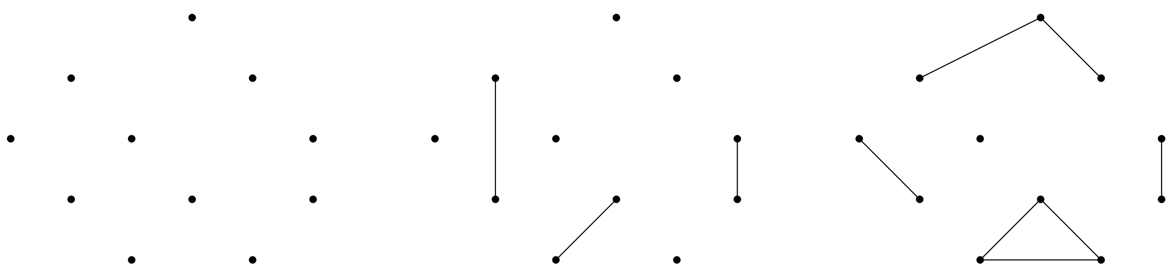
($\Leftarrow$) Suppose M has no augmenting path. Claim that it is a maximum matching. Suppose not, and that M^* is a maximum matching where $|M^*| > |M|$. Consider $M \oplus M^*$. Let H be the subgraph induced by the edges.

Claim:

$$\begin{aligned} |M| &= \# \text{ of } M \text{ -edges } \in H + |M \cap M^*| \\ &= \# \text{ of } M^*\text{-edges} \in H + |M \cap M^*| \end{aligned}$$

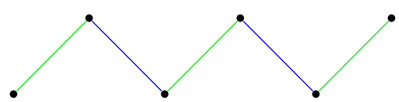
What is the degree of any vertex in H? It's at most two, since each vertex is incident to at most one edge in M and at most one edge in M^*. $deg_H(v) \in \{0, 1, 2\}$.

What does H look like?

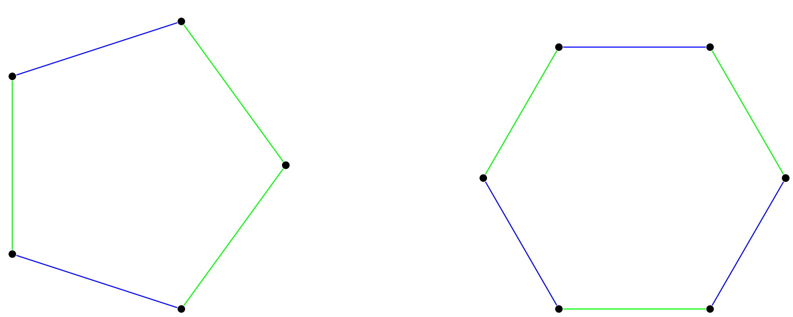


(a) $d_H(v) = 0 \ \forall \ v$ (b) $d_H(v) \in \{0, 1\} \ \forall \ v$ (c) $d_H(v) \in \{0, 1\} \ \forall \ v$

Say blue is M and green is M^*. They alternate:



This means that a cycle must be even:



(a) No (b) Yes

Each component is either

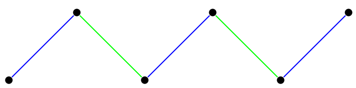
- an even cycle

- a path

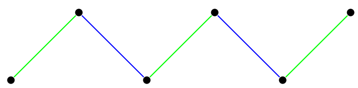
Since alternating an even cycle doesn't change the size of M nor M^*, we will focus on paths.

Consider the 3 following types of paths:

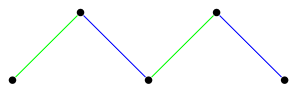
1. M^*-augmenting
2. M-augmenting
3. augments nothing



(a) Type 1



(b) Type 2



(c) Type 3

There are no type 1 paths since they are M^*-augmenting and we assumed M^* was maximum! (See $\Rightarrow$ path of the proof). Each type 3 path, similarly to the cycle components, have the same number M and M^* edges. But, by the claim, H must have more M^* edges than M-edges. Therefore there is a type 2 component, and thus is an M-augmenting path. $\Rightarrow\Leftarrow$

$\blacksquare$

NOTE: This theorem holds for all graphs.

2.4 Matching in Bipartite Graph

G is bipartite if there is a partition $V(G) = X \cup Y$, such that each edge has one endpoint in Z and the other in Y (figure 12).

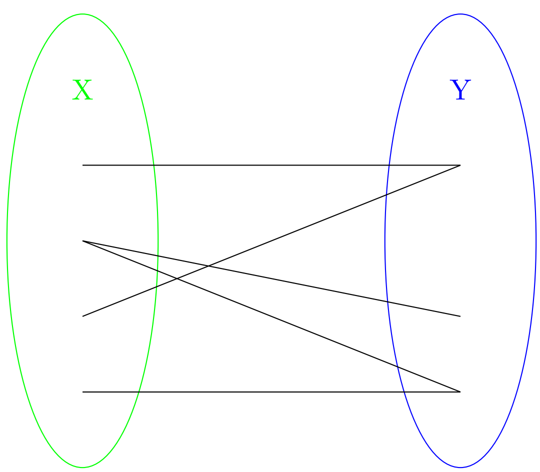
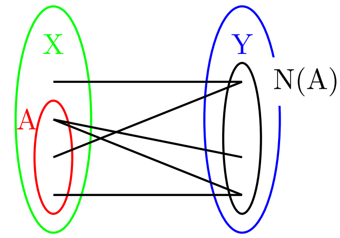


Figure 12: Bipartite graph partitioned into vertex set X and Y

Definition: A matching is perfect if it matches each vertex of G (we can only have degree 1 matching here).



Fundamental question: "When does a graph have a perfect matching?"

Definition: For $A \in V$, denote by $N(A)$ the set of neighbors of A, i.e., $N(A) = \{v \notin A : \exists\, uv \in E, u \in A\}$

Theorem (Hall's): A bipartite graph G with $|X| = |Y|$ has a perfect matching $\iff$ $|N(A) \geq |A| \;\forall\; A \subseteq X$. (Known as Hall's condition)

Proof: ($\Rightarrow$) Trivially holds since we can't have this:

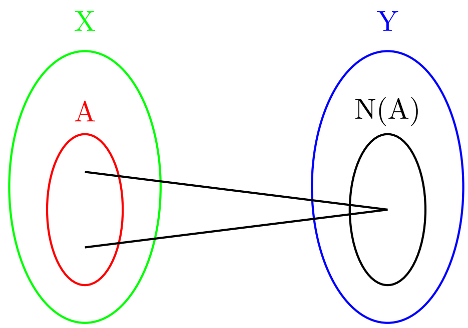


Figure 13: The two vertices in A have only one possible vertex they can match with, therefore there is no perfect matching that would match both.

($\Leftarrow$) If we have some matching M with unmatched vertex u, then we showed how to find an M-augmenting path from u. This gives a new matching which is larger. Repeat until you get a perfect matching.

Algorithm to find M-augmenting path from u:

Let $A = \{u\}, B = \emptyset$. Maintain two properties of A and B as we proceed.

i) $A \subset X$, $B \subset Y$. $A - u$ matched to B by M. (N.B. $|A| = |B| + 1$)

ii) There is an M-alternating path from u to any vertex in $A - u$ in the graph $G[A \cup B]$.

Repeat:

- Choose $v \in N(A) - B$. Let $e = wv, w \in A$. Combine an alternating path from u to w (by ii), with edge e, then get an M-augmenting path and quit.
- if v is matched to some $u' \in X - A$, and u' not in A by i), get $A \in A \cup \{u'\}$, $B \in B \cup \{v\}$. Clearly i) holds. Check that ii) holds (similar to previous argument).
- We can always find another vertex because Hall's condition implies that $N(A) \geq |A| = |B| + 1 > |B|$.

  This proof gives an algorithm for finding a perfect matching in G if it satisfies the Hall Condition.

NOTE: Runtime $\mathcal{O}(VE)$ steps. There exists faster algorithms.

Applications

A graph is *d-regular* if every vertex has degree d.

Theorem: Any d-regular bipartite graph can be decomposed into d perfect matchings, i.e., the edges $E = M_1 \cup M_2 \cup ... \cup M_d$ where each M_i is a perfect matching.

Proof: It is enough to show that we have one perfect matching in G, since if M is a perfect matching, then $G - M$ is a $(d-1)$-regular, and we can repeat.

First, note that since each edge has one end in X and one in Y:

$$\sum_{x \in X} deg(x) = |E| = \sum_{y \in Y} deg(y) \Rightarrow |X| = |Y|$$

We also have that $|E| = d|X| = d|Y|$ since $deg(x) = d \; \forall x \; inX$. Consider $A \subset X$:

$$\begin{aligned} d|A| &= \sum_{x \in A} deg(x) \\ &= \#\text{ of edges with 1 end in } A \\ &\leq \#\text{ of edges with one end in } N(A) \\ &= \sum_{y \in N(A)} deg(y) \\ &= d|N(A)| \end{aligned}$$

Thus $|A| \leq |N(A)|$. So G satisfies Hall's Condition and hence has a perfect matching.

Latin Squares

An $r \times n$ grid is a *Latin Rectangle* if the numbers in each row and column are distinct.

Theorem: Every $r \times n$ Latin rectangle with $r \leq n$ can be extended to an $n \times n$ Latin square.

Proof Define a bipartite graph with a vertex for each column (X), and a vertex for each number $1, 2, ..., n$ (Y). Add an edge from column j to number i if i does not appear in column j.

Each vertex in X will be connected with $n - r$ vertices on the other side. Hence, G is $(n-r)$-regular and has $n - r$ perfect matchings. These give $n - r$ rows which we can add to make the Latin square.

Example

1	2	3	4
2	3	4	1
4	1	2	3