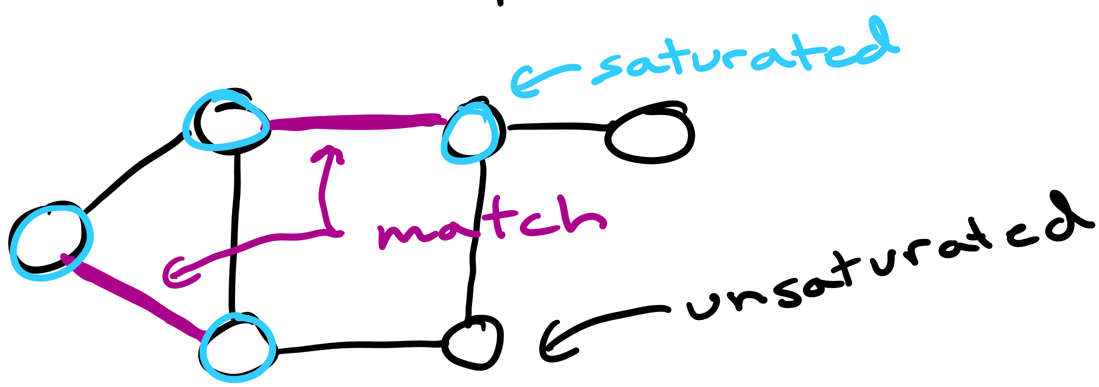
Lecture 10 - Bipartite Matching

Sunday, February 11, 2024 4:41 PM

<u>Match</u> on G = a set of non-loop unique edges with no shared endpoint vertices

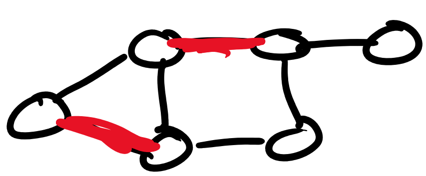


saturated vertex - vertex with an incident matched edge

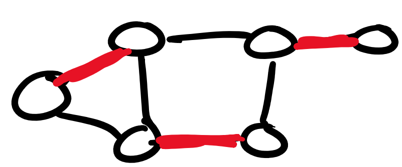
unsaturated vertex - vertex with no incident matched edge

maximum matching - largest possible match in terms of cardinality

maximal - a match that can not be made larger



maximal match



maximum match
perfect match

perfect match

perfect match - a match that saturates all vertices

$$|M_p| = \frac{|V(G)|}{2} \qquad M_p = \{e_i \dots e_j\}$$

$$e_\ell \in E(G)$$

↑ perfect match

Does G have a perfect match?

Necessary conditions

→ $|V(G)|$ must be even

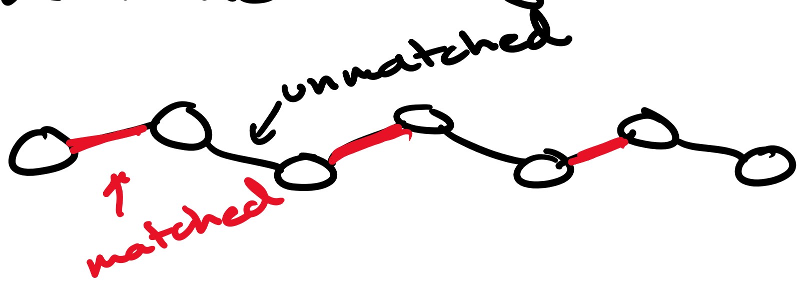
→ $\forall v \in V(G) : d(v) \geq 1$

Sufficient conditions?

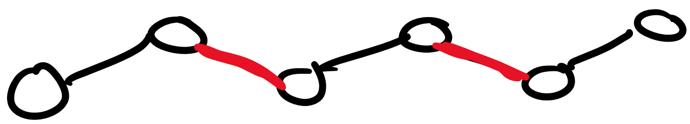
A lot tougher to say

Alternating paths

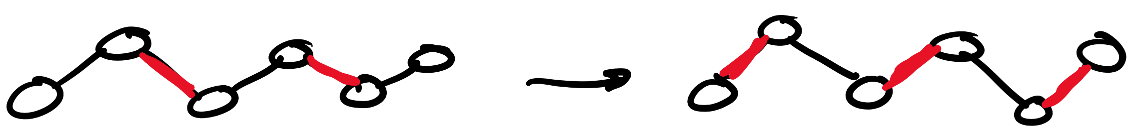
M-alternating path - given some match M on graph G, an M-alt. path is a path subgraph that alternates between matched and unmatched edge unmatched edge



M-augmenting path - an M-alt. path that starts and ends at unsaturated vertices



Note: along an M-aug. path, we can increase our match by swapping matched an unmatched edges



=> this implies that if G has an M-aug. path, then M is not maximum

Q: does every non-maximum match have an M-aug. path?

Q: does every non-maximum match have an M-aug. path?

Berge: a matching M on G is maximum iff there exists no M-aug paths

Contrapositive

P implies Q

Contrapositive: $P \rightarrow Q$

$\neg Q \rightarrow \neg P$

logically equivalent

not Q implies not P

$P \Longleftrightarrow Q$

$\neg P \Longleftrightarrow \neg Q$

Contrapositive of Berge:

M is not maximum $\Longleftrightarrow$ $\exists$ M-aug. path

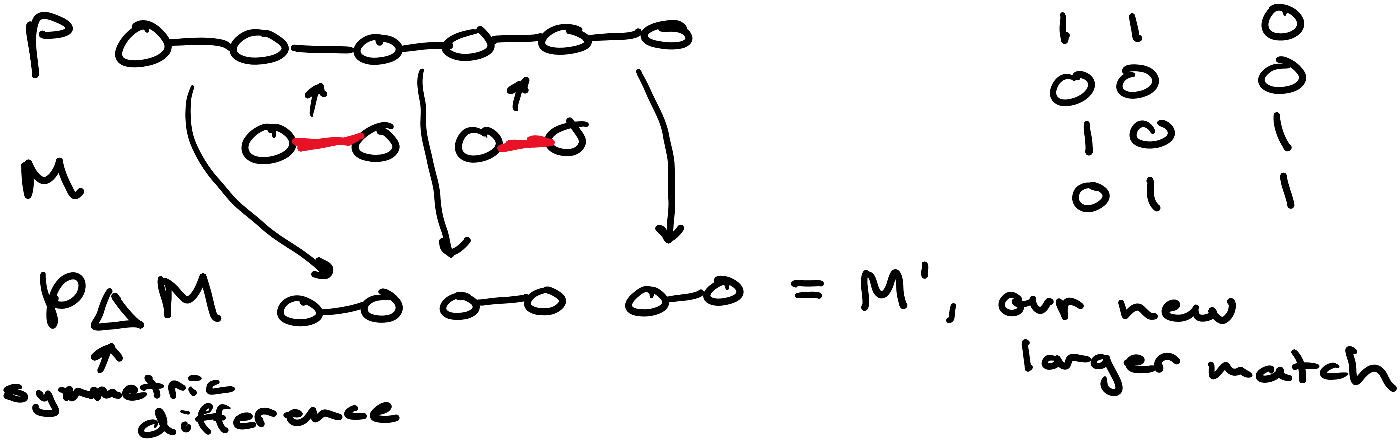
($\Leftarrow$) we already showed how to increase a match via an M-aug. path

Note: to do so, we taking the symmetric difference of

Note: To do so, we taking the <u>symmetric difference</u> of match M and path P

→ an exclusive or → XOR

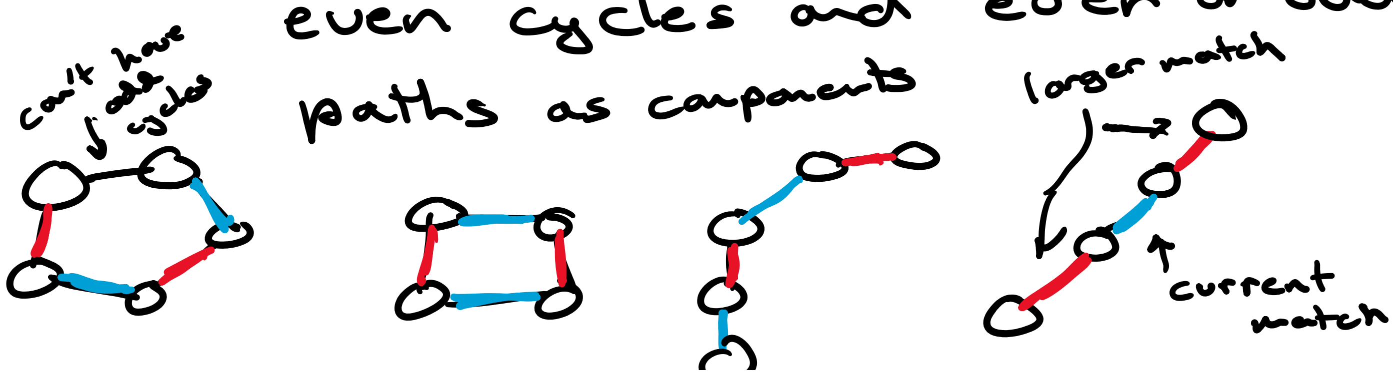
A	B	XOR
1	1	0
0	0	0
1	0	1
0	1	1

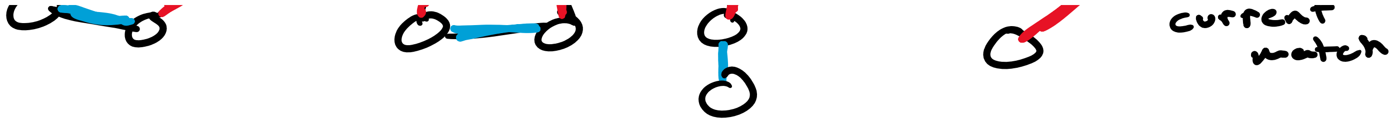


(=>) Consider M as our current match and M' as a larger match where $|M'| > |M|$

Consider $F = M' \triangle M$

↳ F is going to be comprised of even cycles and even or odd paths as components





Consider our odd paths in F
→ we know at least one must exist as $|M'| > |M|$
⇒ this is our M-aug. path □

Maximum Bipartite Matching

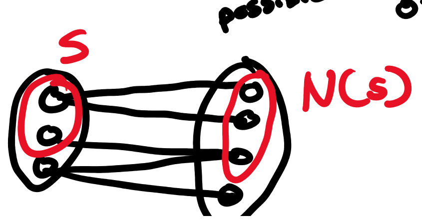
↳ we consider optimality with respect to $|X|$ wlog where $|X| \le |Y|$ in G_{XY} bipartite graph

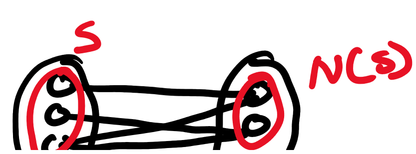
if $\exists M$ that saturates all $v \in X$, then M is maximum
(X-saturating match)

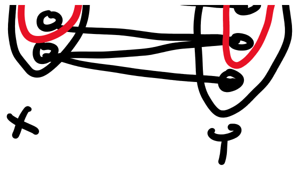
Hall: $\exists M$ that fully saturates X in bigraph G_{XY} and $|X| \le |Y|$
iff $\forall S \subseteq X : |N(S)| \ge |S|$

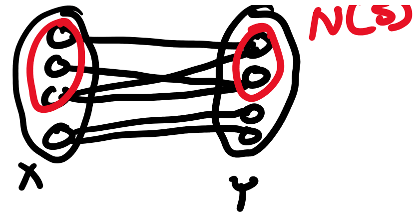
all possible subset of X

neighborhoods of all in S





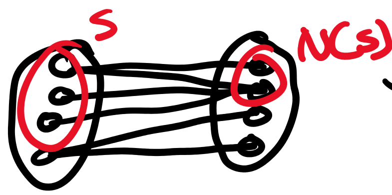




☆ Contrapositive ☆

there is no X-saturating M
iff
$\exists S \subseteq X : |S| > |N(S)|$

($\Leftarrow$) We demonstrated that above



→ there are not enough 'partners' in Y for S to match with

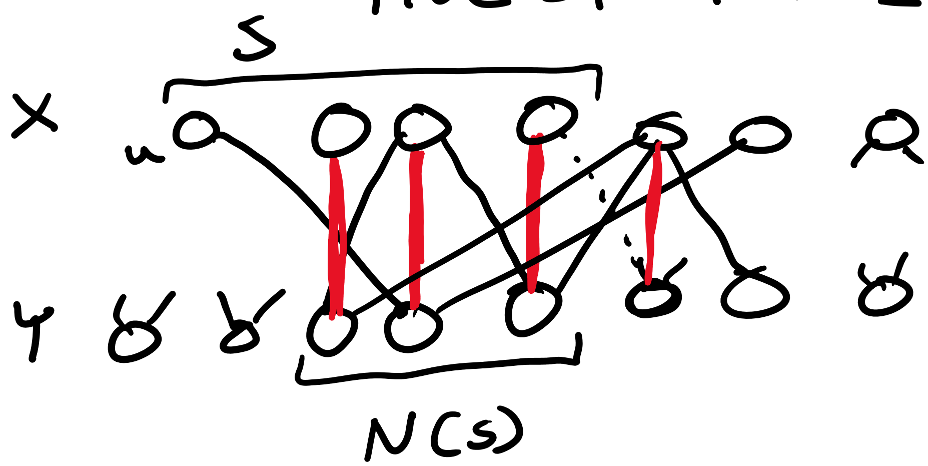
($\Rightarrow$)
- consider some maximum M
- consider some $u \in X$, $u \notin V(M)$ ← unsaturated
- consider S = all $v \in X$ that are reachable via a u,v-M-alt path

Note: $N(S)$ is fully saturated
→ otherwise, we have an M-aug path and M is not optimal ✗contradiction✗

path and M is not optimal

Finally: considering a bijection from $N(S) \leftrightarrow S-u$, we have

$$|N(S)| = |S| - 1 < |S| \quad \square$$



Q: How can we determine or find a maximum match on a bipartite graph?

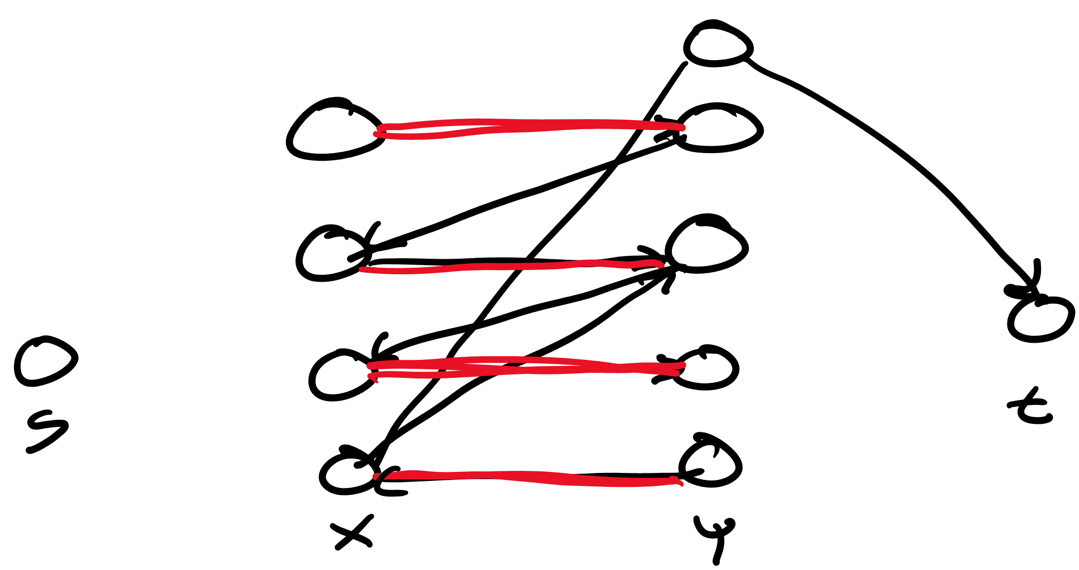
In general: keep finding M-aug paths via M-alt paths from unsaturated vertices (in X)

Maximum Bipartite Matching Algorithm

Iteratively find M-aug paths to increase $|M|$

To do so:

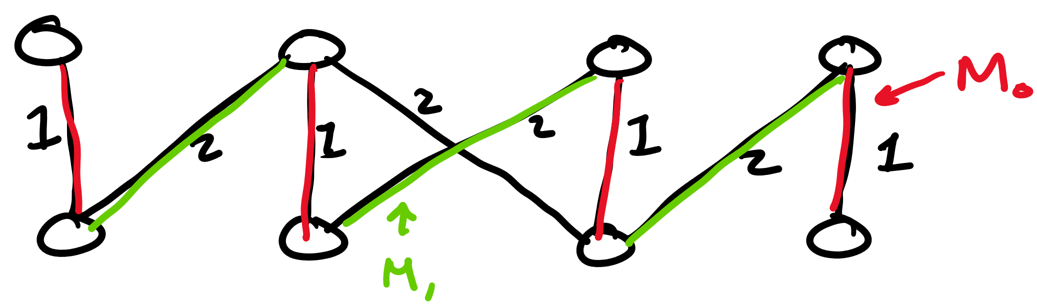
- add vertex s, which has directed edges to all unsaturated vertices in X
- add vertex t, which has directed edges from all unsaturated vertices in Y
- direct all matched edges $(x, y) : x \in X, y \in Y$ to point from y to x
- direct all unmatched edges from x to y



Mo' Matching

Maximum weight matching

↳ we optimize the sum of edge weights in our match instead of cardinality $|M|$



$|M_0| = 4$ $\qquad$ $|M_1| = 3$

$\underbrace{\sum_{e \in M_0} w(e) = 4}_{\text{max cardinality}}$ $\qquad$ $\underbrace{\sum_{e \in M_1} w(e) = 6}_{\text{max weight}}$

Stable Matching

↳ optimize our match for given preferences

Classic "Marriage Problem"

- n men and n women
- each specify a preference

- each specify a preference order for potential partners

unstable pair: man x and woman a are not matched but both prefer match (x, a) over their current partner

stable match: a match with no unstable

Gale-Shapley Proposal Algorithm:

Input: preferences for all n men and n women

while not done:

Each man proposes to their highest preference woman who has no yet rejected them

If each woman gets exaclty 1 proposal → done

Else women with 2+ proposals reject all

proposals reject all
but their highest preference

Men (x, y, z, w) Women (a, b, c, d)

$x: a > b > c > d$ $a: z > x > y > w$

$y: \cancel{a} > \cancel{c} > b > d$ $b: y > w > x > z$

$z: \cancel{c} > d > a > b$ $c: w > x > y > z$

$w: c > b > a > d$ $d: x > y > z > w$

Iterate:

(x, a) (y, a) (z, c) (w, c)

reject reject

(x, a) (y, c) (z, d) (w, c)

reject

(x, a) (y, b) (z, d) (w, c) ✓