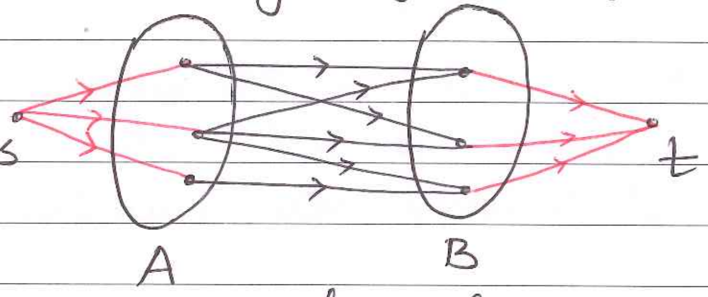
Lecture 8

We will use Ford-Fulkerson algorithm to find a max-size matching in a bipartite graph.
- we need to transform G into a directed graph
- we need a source s and a sink t.

So let us add vertices s and t as follows and direct all edges from left to right



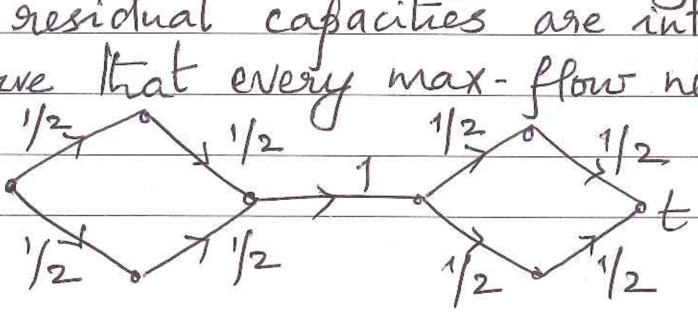
That is, we add edges (s, a) for all $a \in A$ and edges (x, t) for all $x \in B$.
- Set every edge capacity to 1. So $c(e) = 1 \ \forall e$.

* Compute a max-flow in the above graph using Ford-Fulkerson algorithm.

Let f be the flow computed by Ford-Fulkerson algorithm. The following property of Ford-Fulkerson algorithm will be important here:
- when all edge capacities are integers, $f(e) \in \mathbb{Z}$ for all $e \in E$.

This is because this algorithm finds an s-t path in G_f and sends as much flow along it as possible. So residual capacities are integral and so on. Observe that every max-flow need not be integral.

Let all edges capacities be 1.



This is a max-flow that is not integral.

Going back to Ford-Fulkerson algorithm, we have $0 \leq f(e) \leq 1$ and since $f(e) \in \mathbb{Z}$, this means $f(e)$ is either 0 or 1.

Let $M = \{e \in E : f(e) = 1\}$.

Exercise. Show that M is a max-size matching in the given bipartite graph.

Call a matching M in $G = (A \cup B, E)$ A-perfect if $|M| = |A|$. That is, every vertex in A has a matching edge incident to it.

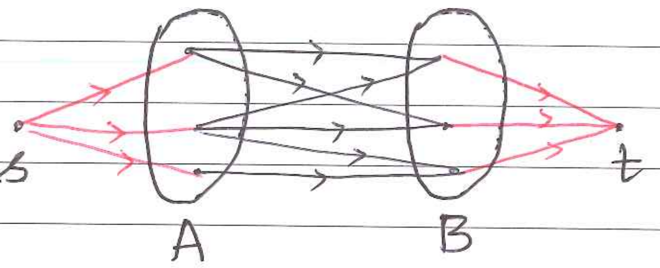
Hall's theorem gives a necessary and sufficient condition for G to have an A-perfect matching. There are many proofs of this theorem – we will now see one based on max flow - min cut theorem.

Hall's theorem: A bipartite graph $G = (A \cup B, E)$ has an A-perfect matching if and only if for all $S \subseteq A$: $|Nbr(S)| \geq |S|$.
– here $Nbr(S)$ is the set of neighbours of vertices in S.

Proof. It is easy to show one side of the above statement. Suppose G has an A-perfect matching M. Let S be any subset of A. $S = \{a_1, .., a_k\}$. Since M is A-perfect, each of $a_1, .., a_k$ has an edge of M incident to it.
So $Nbr(S) \supseteq \{M(a_1), .., M(a_k)\}$.
Thus $|Nbr(S)| \geq k = |S|$.

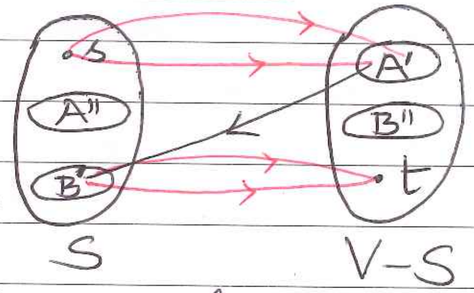
We now need to show the converse. That is, if $|Nbr(S)| \geq |S|$ $\forall S \subseteq A$ then G has an A-perfect matching.

Suppose G does not have an A-perfect matching. Consider the following network:



It will be convenient for us to set $c(e) = \infty$ for all edges e in $A \times B$. For edges e outgoing from s or incoming into t we still have $c(e) = 1$. Please check that what was discussed earlier holds true here as well — an integral max-flow projects to a max-size matching in G.

By our assumption above, there is no matching of size $|A|$ in G. So value of max-flow in the above network is $< |A|$. By max-flow min-cut theorem, the s-t mincut in this network has capacity $< |A|$



Let $(S, V-S)$ be an s-t min cut here. Note that there is no edge from A'' to B'' since such an edge e has $c(e) = \infty$. That would make the capacity of this cut ∞, however the capacity of s-t min cut here $\leq |A|$. Thus $Nbr(A'') \subseteq B'$.

$$\text{Capacity of this cut} = \underbrace{|A'|}_{\substack{\text{edges from}\\ s \text{ to } A'}} + \underbrace{|B'|}_{\substack{\text{edges from}\\ B' \text{ to } t}} < |A|$$

by our assumption

So $|B'| < |A| - |A'|$
$= |A''|$

Since $Nbr(A'') \subseteq B'$ this contradicts our hypothesis that $|Nbr(S)| \geq |S| \ \forall \, S \subseteq A$. □

Improving Ford-Fulkerson algorithm

We would like to bound the number of repeat-loop iterations by a polynomial in m, n.
Let us try the following approach

1. Initialize $f(e) = 0 \quad \forall\, e \in E$.
2. Repeat
   - augment f such that $d(s,t)$ in the new G_f $>$ $d(s,t)$ in the old G_f

   until there is no s-t path in G_f.
3. Return f.

- $d(s,t)$ = number of edges in the shortest s-t path in G_f.

The termination condition of Ford-Fulkerson algo is that there is no s-t path in G_f. Suppose we can ensure that in each iteration of the repeat-loop, s-t distance in G_f is "worsening".
- at the beginning $d(s,t) \geq 1$.
- at the end, $d(s,t) = \infty$ since there is no s-t path in G_f.

For how many iterations can the above repeat-loop run if we ensure that in each iteration $d(s,t)$ is increasing by at least 1?
- note that if there is an s-t path in G_f then $d(s,t) \leq$ **n-1** (since the total number of vertices is n)

So the number of repeat-loop iterations is at most n if we are able to find a flow in G_f in each iteration such that $\underbrace{d_{i+1}(s,t)}_{s\text{-}t \text{ distance in } (i+1)\text{-th itn.}} > \underbrace{d_i(s,t)}_{s\text{-}t \text{ distance in } i\text{-th itn.}}$

Goal: find a flow f_b in G_f^i such that augmenting f along f_b makes $d_{i+1}(s,t) > d_i(s,t)$.

- G_f^i is the residual graph in the i-th iteration.

and $d_i(s,t)$ = number of edges in the shortest s-t path in G_f^i

What properties should f_b have?

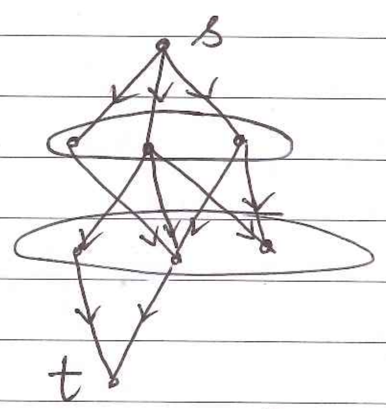
- every shortest s-t path in G_f^i should have at least 1 edge saturated by f_b

Layered network: This uses the BFS tree rooted at s in the graph G_f

- all edges from layer i to layer $i+1$ in G_f are present here.

layers are given by the BFS tree

L_f:



Idea: Find a flow f_b in L_f such that every shortest path in G_f, i.e. every s-t path in L_f, has at least 1 edge saturated by f_b

Such a flow f_b is called a blocking flow.

Let $D_i(v)$ = number of edges in a shortest s-v path in G_f^i

We need to show that $D_{i+1}(t) > D_i(t)$ if we use the above idea.

Let us first write down the improved algorithm for max-flow. This is called Dinic's algorithm.

1. Initialize $f(e) = 0 \quad \forall e \in E$.
2. Repeat
   - construct L_f and find a blocking flow f_b.

– augment f along f_b; update G_f

until there is no s-t path in G_f

3. Return f

The correctness of the above algorithm follows from max flow – min cut theorem. If the above algorithm returns f then f is a max flow in G (since there is no s-t path in G_f).

We claim the number of repeat-loop iterations is $\leq n$ since $D_{i+1}(t) > D_i(t)$ in every iteration i.

Let us prove this now. Suppose p is a shortest s-t path in G_f^{i+1}.

Case 1. Every edge in p is also present in G_f^i.
In this case, p cannot be a shortest s-t path in G_f^i – this is because at least 1 edge has been saturated in every shortest s-t path in G_f^i and the residual capacity of a saturated edge is 0.
– thus p is an s-t path in G_f^i but not a shortest s-t path in G_f^i.
So $|p| > D_i(t)$ and we also have $|p| = D_{i+1}(t)$.
number of edges in p
Thus $D_{i+1}(t) > D_i(t)$.

Case 2. The path p has some edges not in G_f^i.

p: 

Let (u,v) be a "new" edge in G_f^{i+1}, i.e., $(u,v) \notin G_f^i$.
We have $D_{i+1}(u) \geq D_i(u)$ and $D_{i+1}(v) = D_{i+1}(u) + 1$.
Show that $D_i(u) = D_i(v) + 1$
and hence $D_{i+1}(v) \geq D_i(v) + 2$. We'll continue this in next lecture.