Network Flow Applications

Admin

- Assignment 6 is due this Wed
- Assignment 7 will be released this week
- Midterm graded:
  - Mean: ~89, Median ~ 90
  - I don't release exam solutions, but happy to discuss feedback in person
- Lots of office and TA hours this week:
  - (Me) 2-3.30 pm today, 2-3 pm tomorrow, 1.30-3 pm Wed
  - TA hours: 9-11 pm today, 8-10 pm tomorrow

Health Days Next Week!

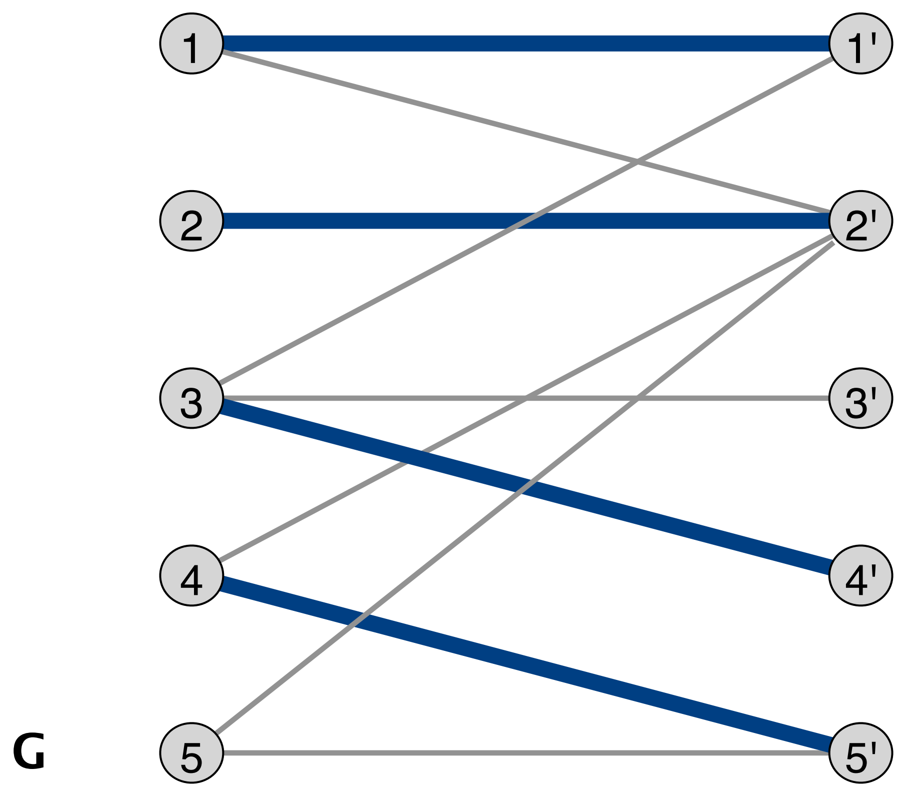
You are here

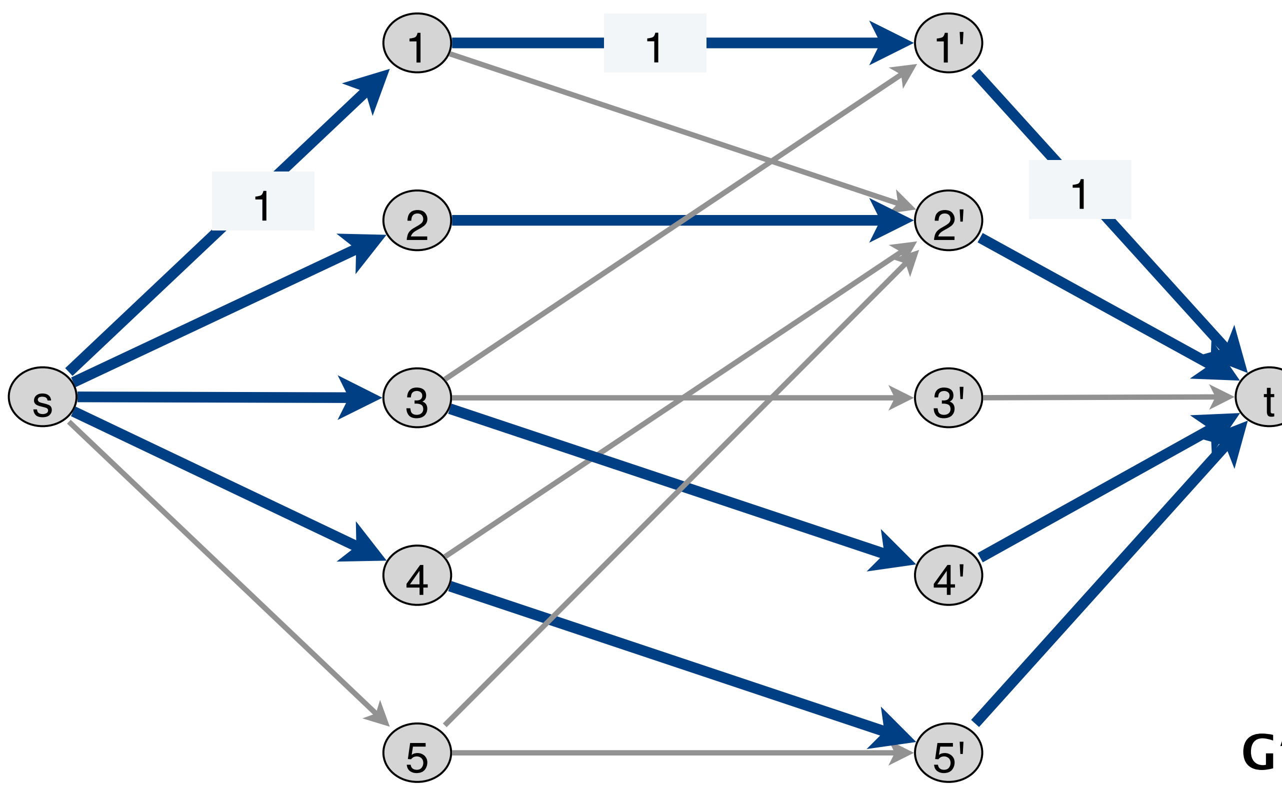
12Apr Flow Applications Reading: KT §7.6 ❘ E §11	13Apr	14Apr P vs NP and NP-hardness Reading: KT §8.1, 8.3 ❘ E §12.1–12.5 Assignment 7 out Assignment 6 due	15Apr	16Apr Problem Reductions Reading: KT §8.1, 8.3 ❘ E §12.1–12.5
19Apr NP-hard Reductions Reading: Reading: KT §8.2, 8.4 ❘ E §12.6–12.8	20Apr	21Apr Health Day	22Apr Health Day	23Apr Intractability Wrap Up Reading: KT §8.5–8.7; ❘ E §12.6–12.8

Rest and sunshine is here!

Bipartite Matching & Flow

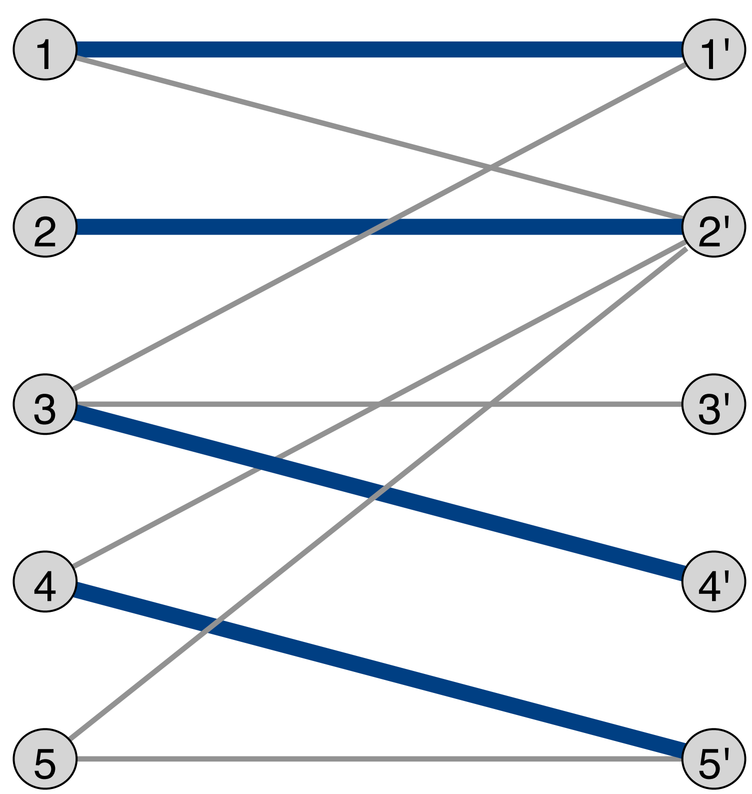
- Input: a bipartite graph (X, Y, E)
- Create a new directed graph $G' = (X \cup Y \cup \{s, t\}, E', c)$
- Add edge $s \rightarrow x$ to E' for all nodes $x \in X$
- Add edge $y \rightarrow t$ to E' for all nodes $y \in Y$
- Direct edge $x \rightarrow y$ in E' if $(x, y) \in E$
- Set capacity of all edges in E' to 1



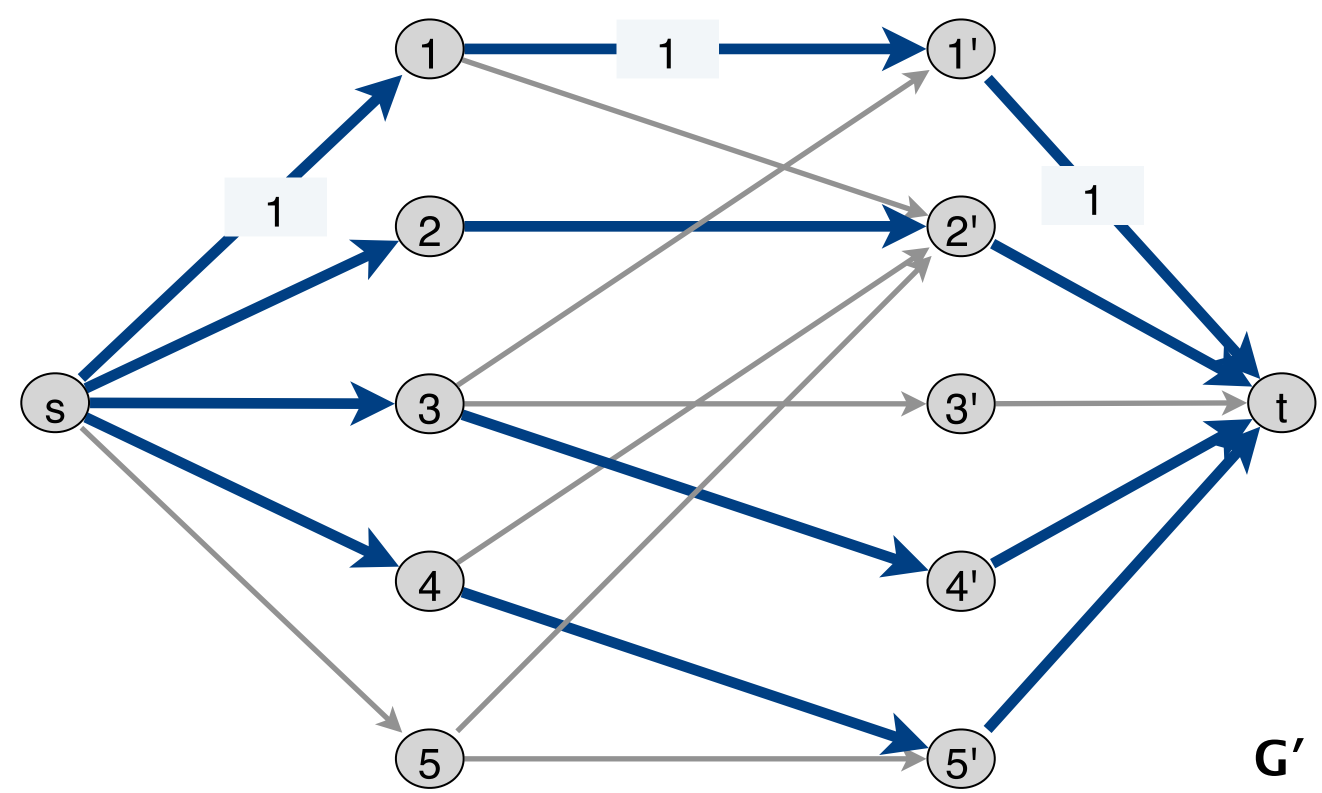


Bipartite Matching & Flow

- We showed that bipartite graph has a matching of size k if and only if the flow network has a flow of value k
- Now suppose $|X| = |Y| = n$
- If maximum flow is of value n, our original graph has a perfect matching!



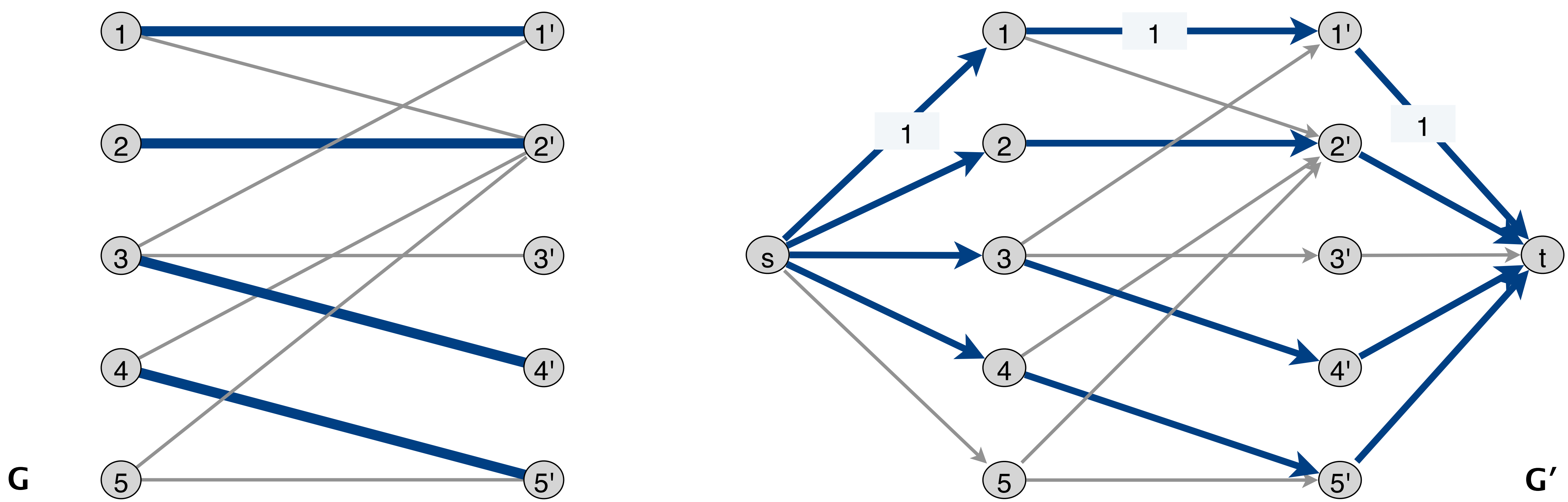
G



G′

Bipartite Matching & Flow

- We showed that bipartite graph has a matching of size k if and only if the flow network has a flow of value k
- Now suppose $|X| = |Y| = n$
- Suppose max flow is less than n, can we find a "certificate" that the original graph cannot have a perfect matching?

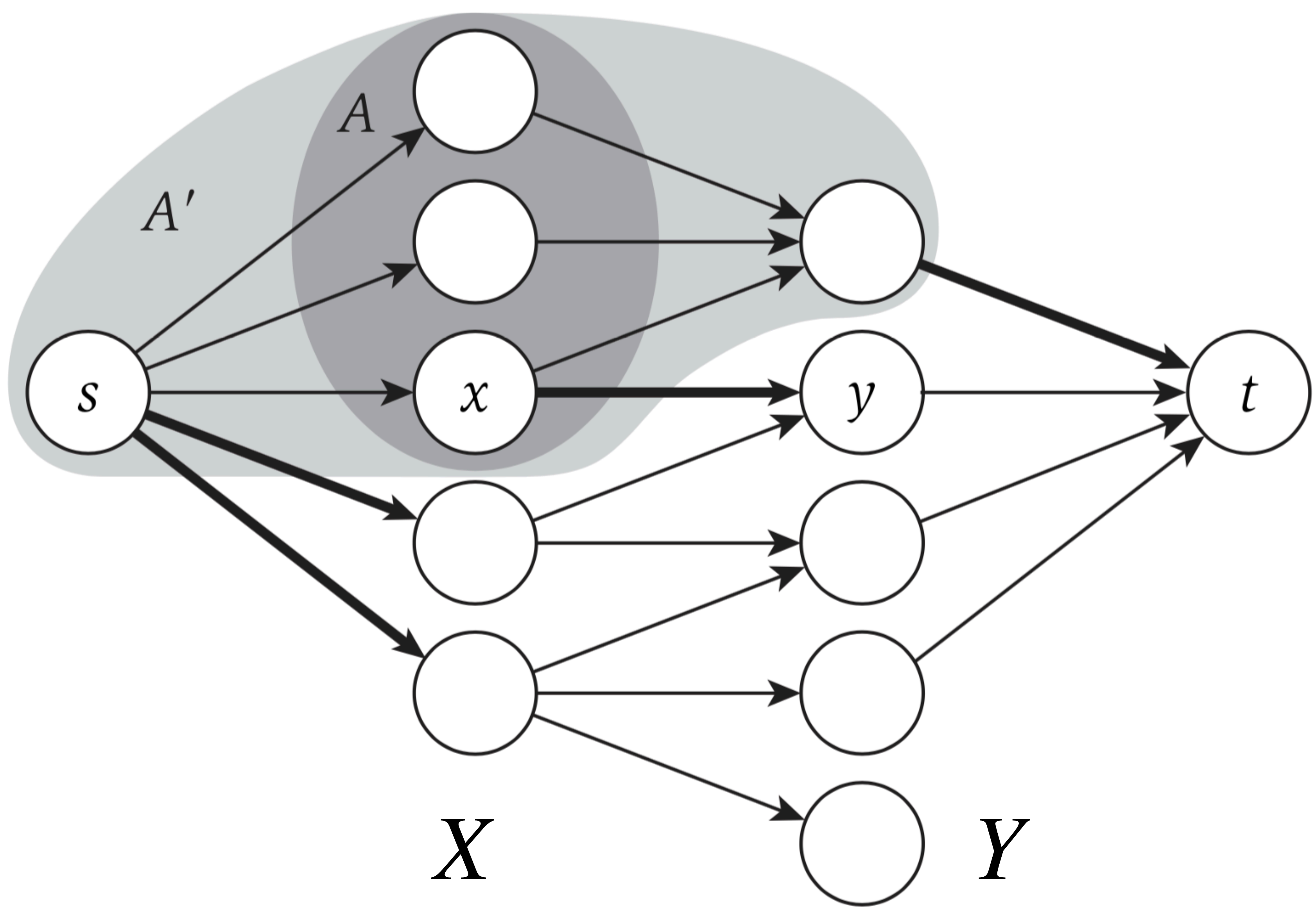


Hall's Theorem

- **Notation**. Let S be a subset of nodes in G, and let $N(S)$ be the set of nodes adjacent to nodes in S in G.
- **[Halls marriage theorem.]** Let $G = (X \cup Y, E)$ be a bipartite graph with $|X| = |Y|$. Then, graph G has a perfect matching iff $|N(S)| \geq |S|$ for all subsets $S \subseteq X \cup Y$.
- Proof.
- $(\Rightarrow)$. In a perfect matching, each node in S needs to be matched with a different node in $N(S)$
- $(\Leftarrow)$. Suppose G does not have a perfect matching
- We will find a subset S such that $|N(S)| < S$

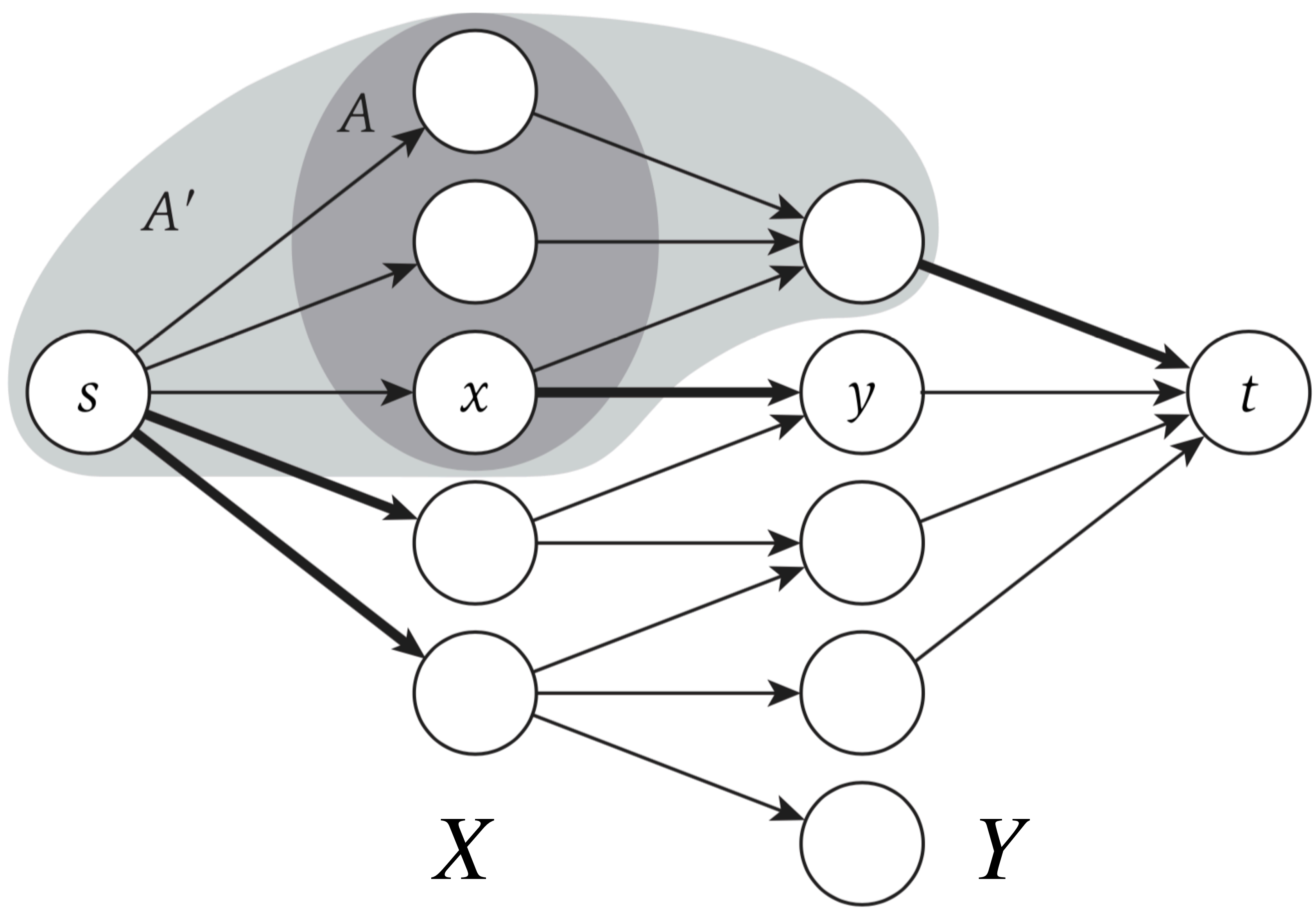
Perfect Matchings & Cuts

- ($\Leftarrow$). Suppose G does not have a perfect matching, then capacity of the min-cut (A', B') is less than n
- **Claim**. $A = X \cap A'$ has this property.



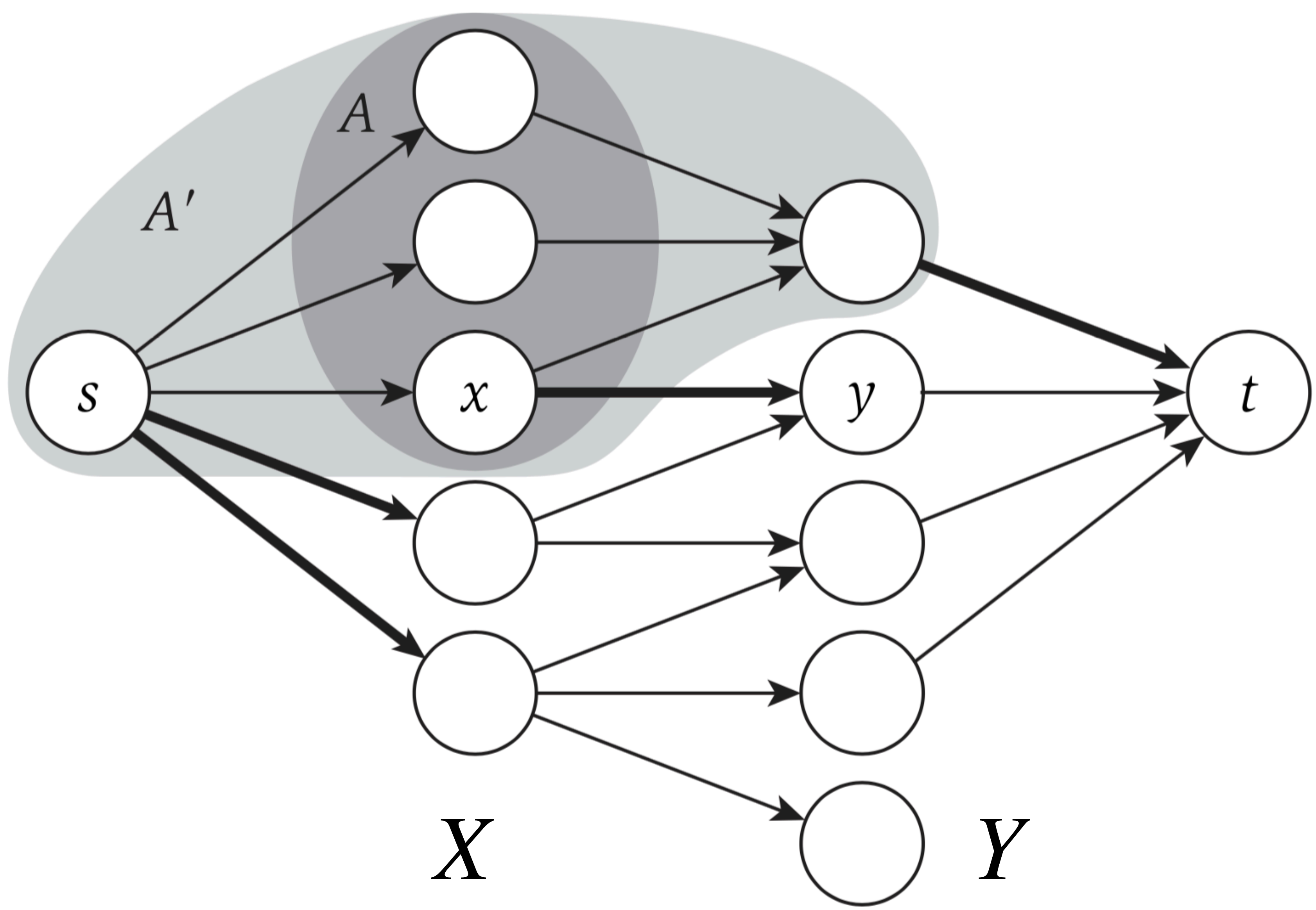
Perfect Matchings & Cuts

- ($\Leftarrow$). Suppose G does not have a perfect matching, then capacity of the min-cut (A', B') is less than n
- All edges are of capacity 1: $c(A', B')$ = # edges leaving cut A'



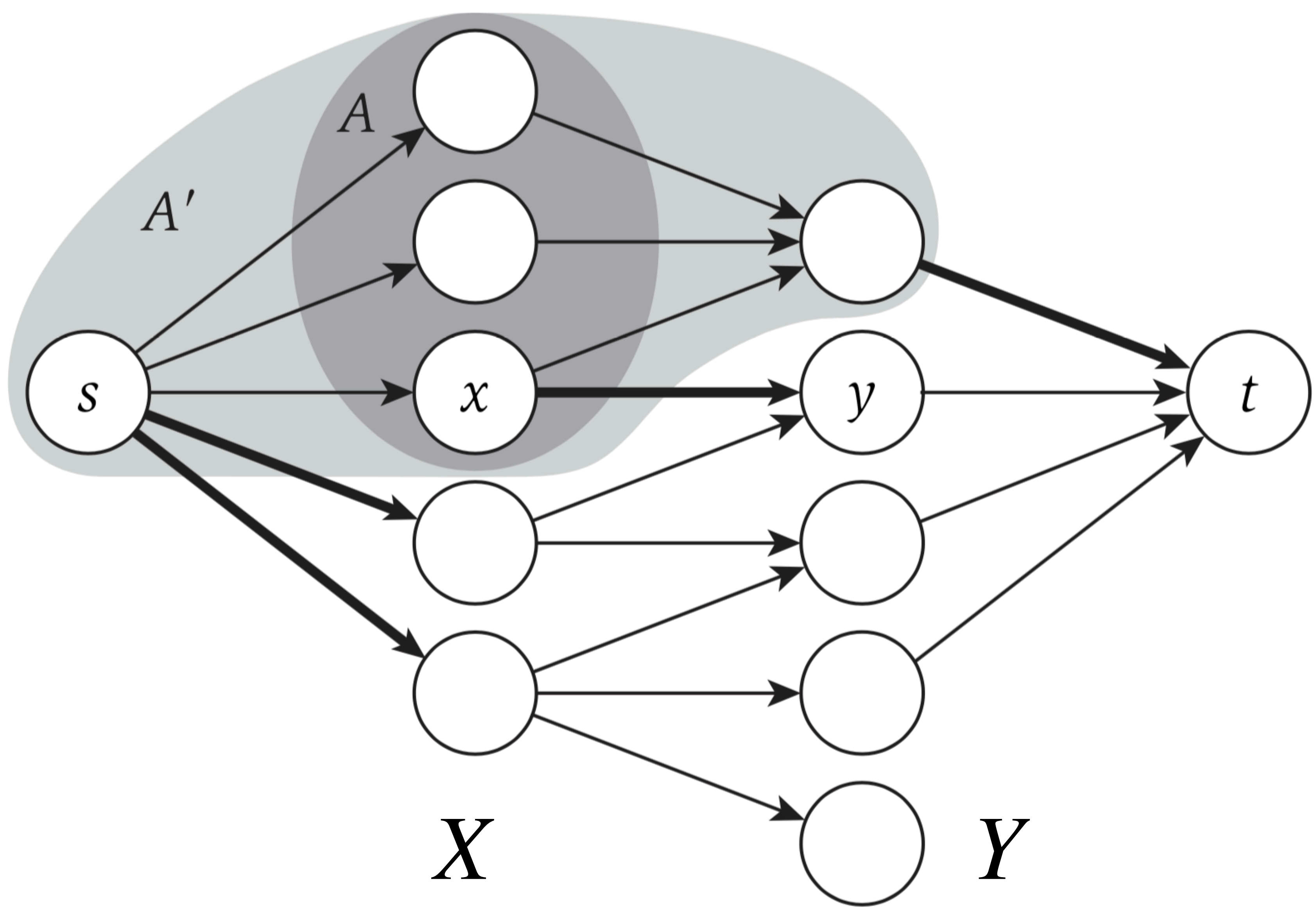
Perfect Matchings & Cuts

- $(\Leftarrow)$. Suppose G does not have a perfect matching, then capacity of the min-cut (A', B') is less than n
- $c(A', B') = n - |A| + |Y \cap A'| < n$



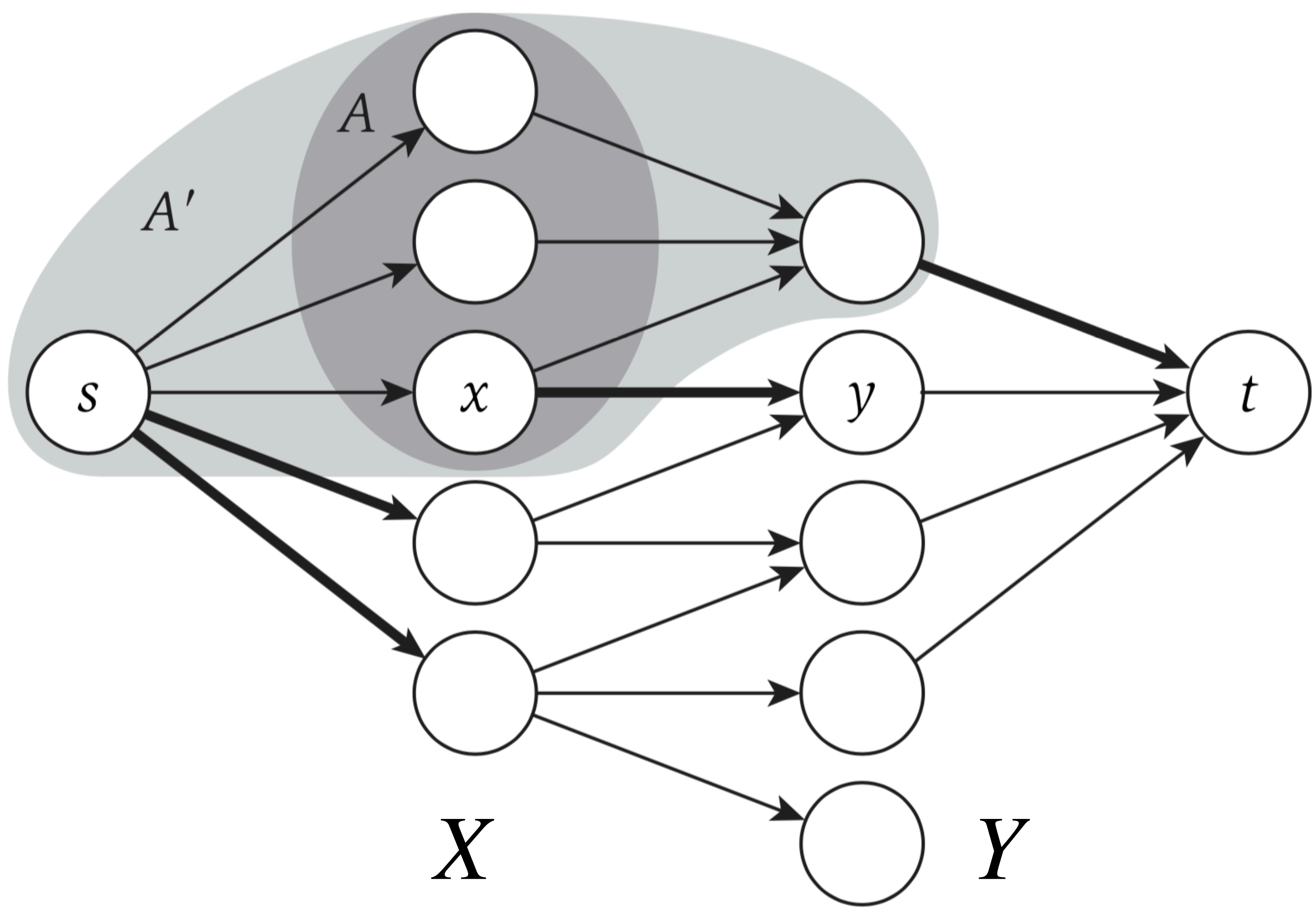
Perfect Matchings & Cuts

- $(\Leftarrow)$. Suppose G does not have a perfect matching, then capacity of the min-cut (A', B') is less than n
- $|Y \cap A'| < |A|$



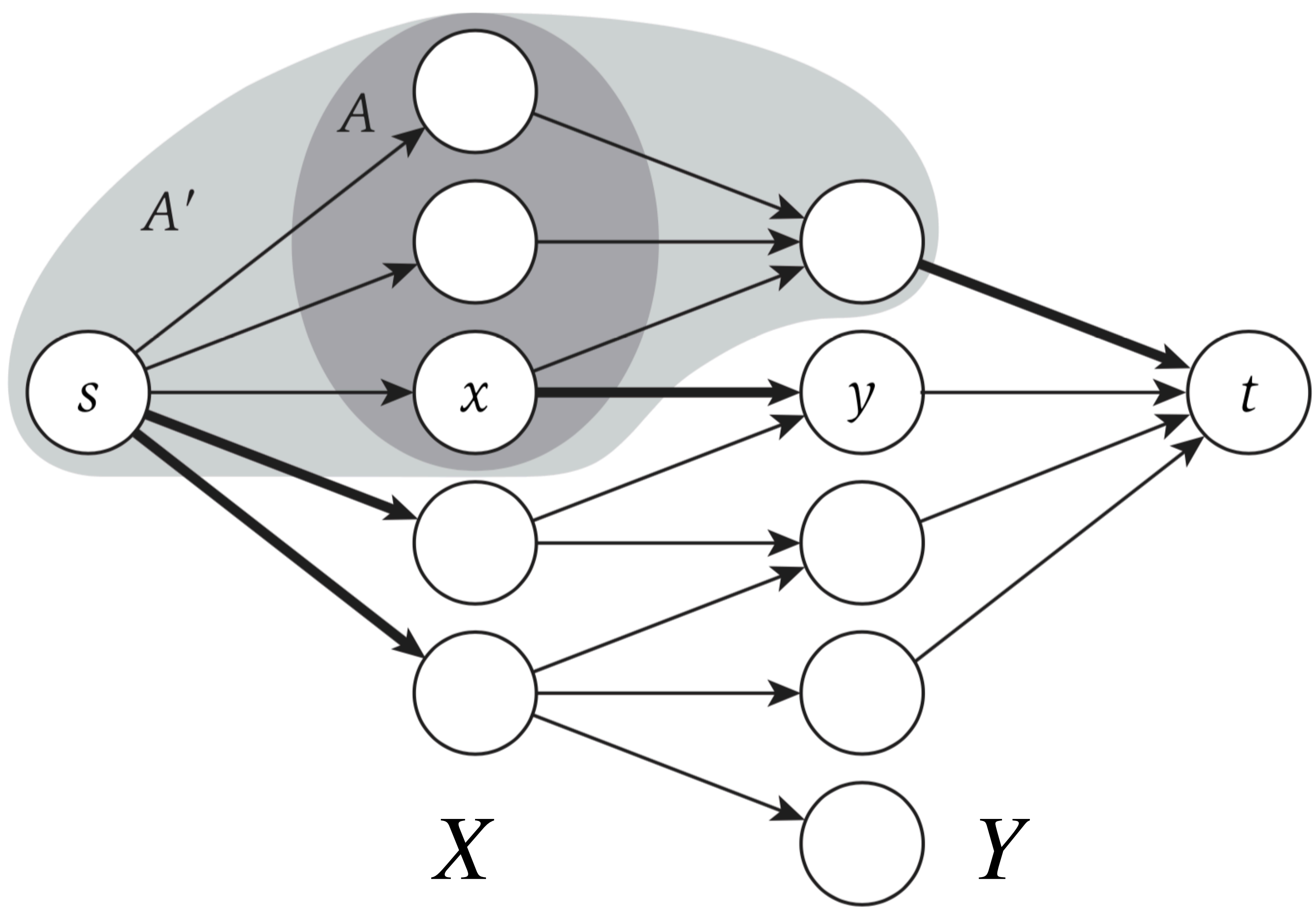
Perfect Matchings & Cuts

- $(\Leftarrow)$. Suppose G does not have a perfect matching, then capacity of the min-cut (A', B') is less than n
- $|Y \cap A'| < |A|$, need to show $|N(A)| < |A|$



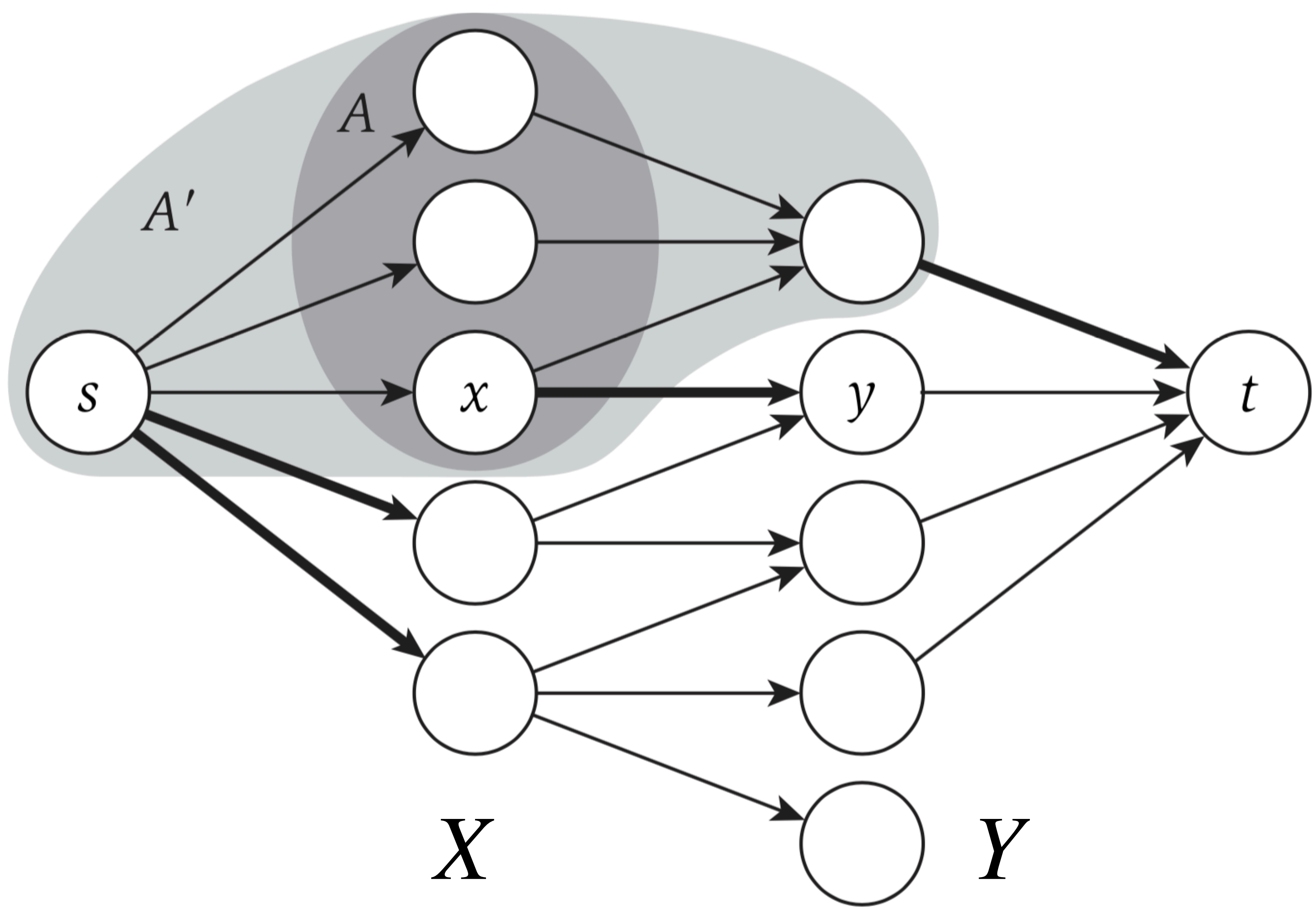
Perfect Matchings & Cuts

- $(\Leftarrow)$. Suppose G does not have a perfect matching, then capacity of the min-cut (A', B') is less than n
- $|Y \cap A'| < |A|$, enough to show $N(A) \subseteq Y \cap A'$



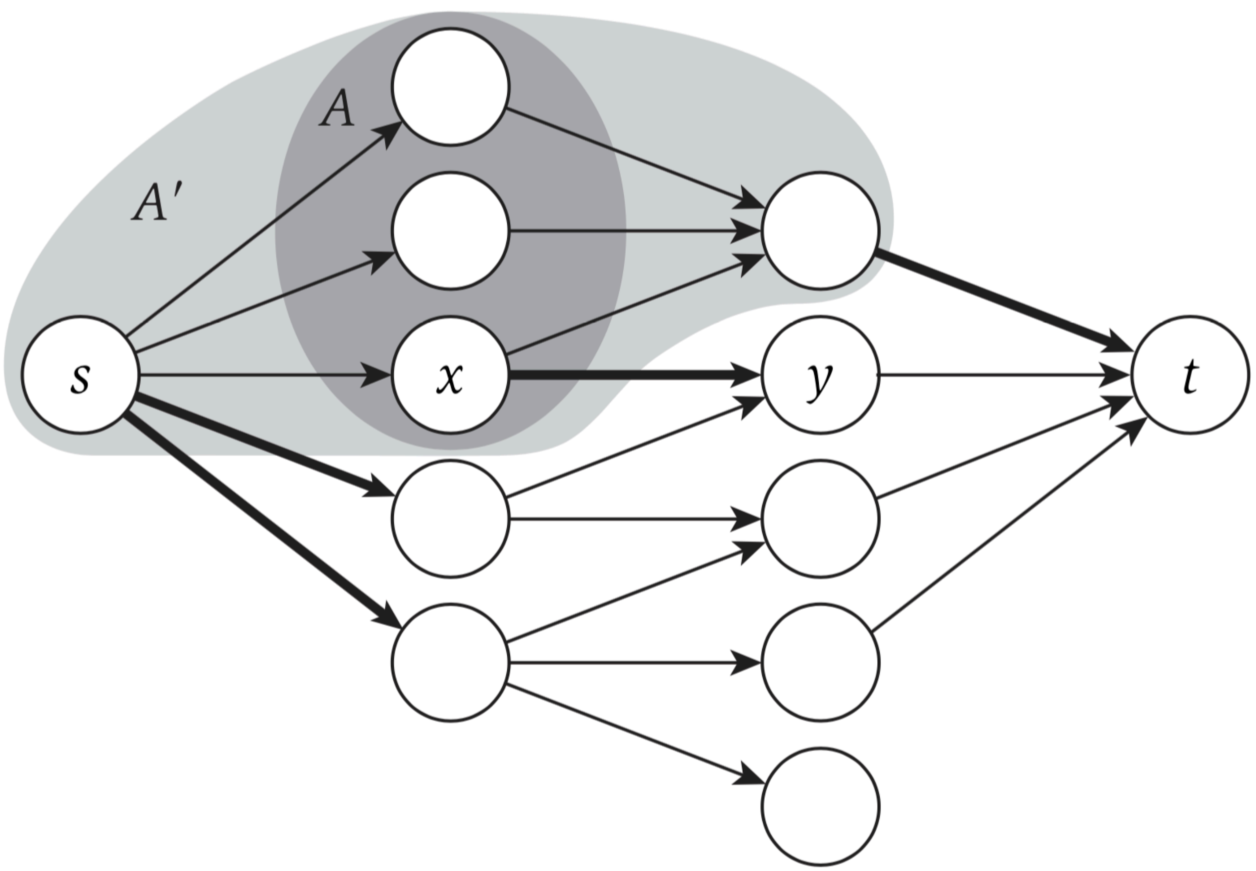
Perfect Matchings & Cuts

- ($\Leftarrow$). Suppose G does not have a perfect matching, then capacity of the min-cut (A', B') is less than n
- $|Y \cap A'| < |A|$, enough to show $N(A) \subseteq A'$

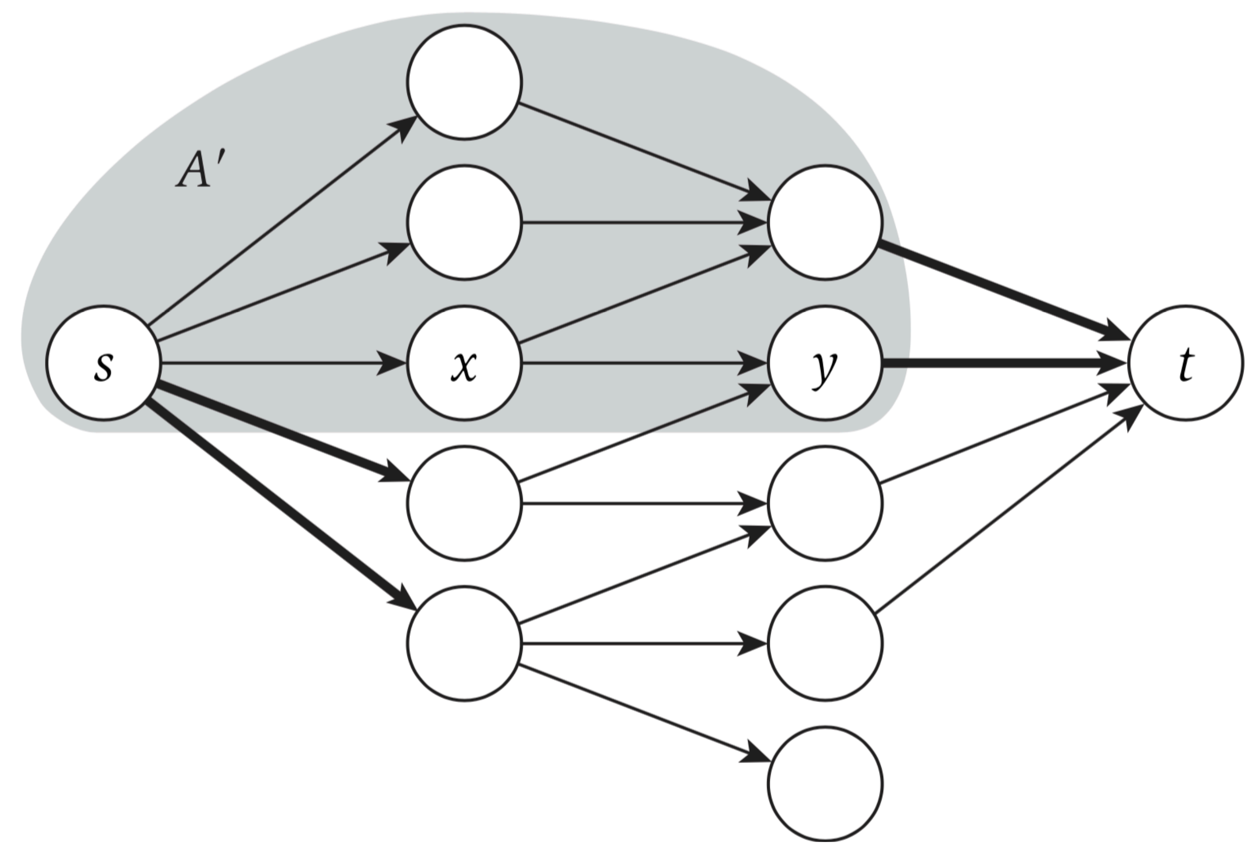


Perfect Matchings & Cuts

- We will show, if a mincut (A', B') doesn't have the property that $N(A) \subseteq A'$, we can find a new cut that does, that is, wlog we can assume $N(A) \subseteq A'$, where $A = X \cap A'$



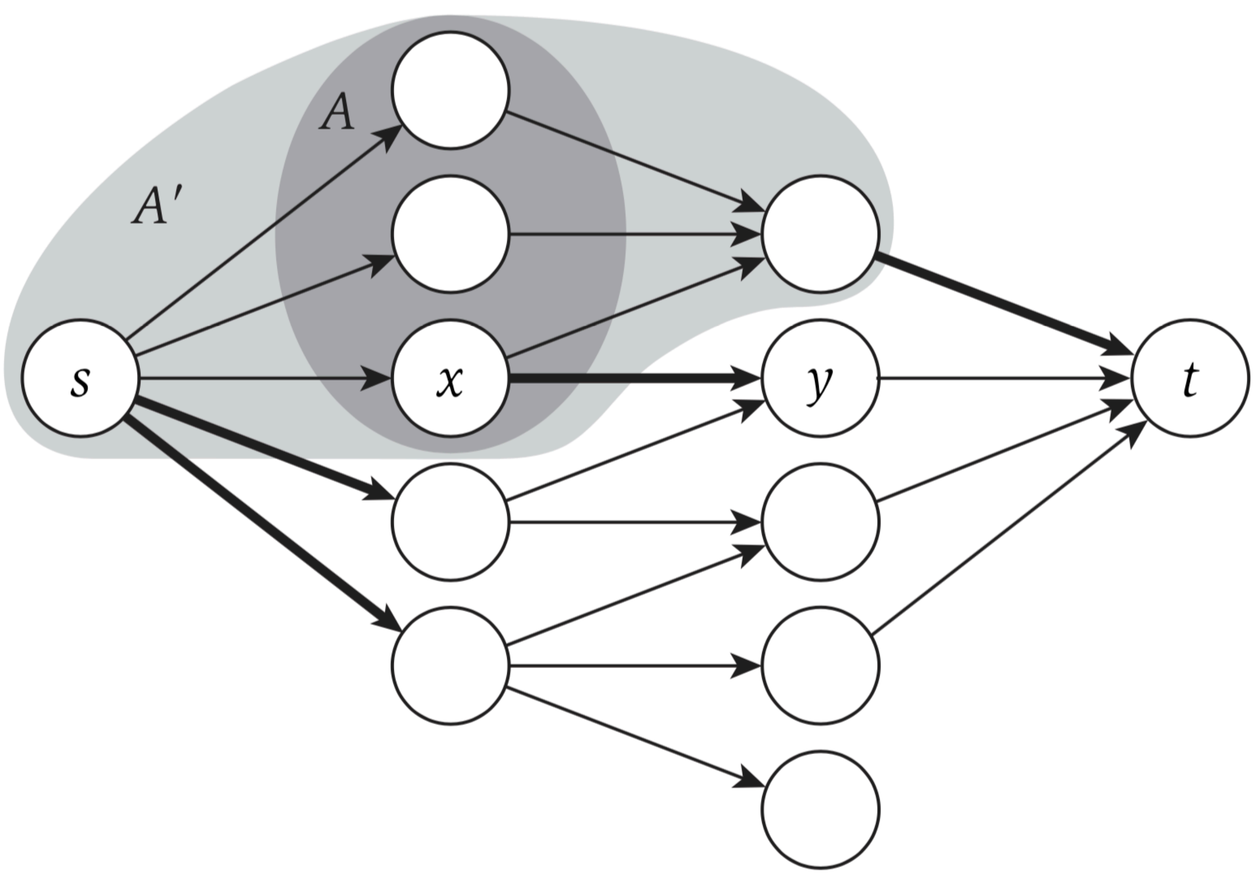
$N(A) \subsetneq A'$



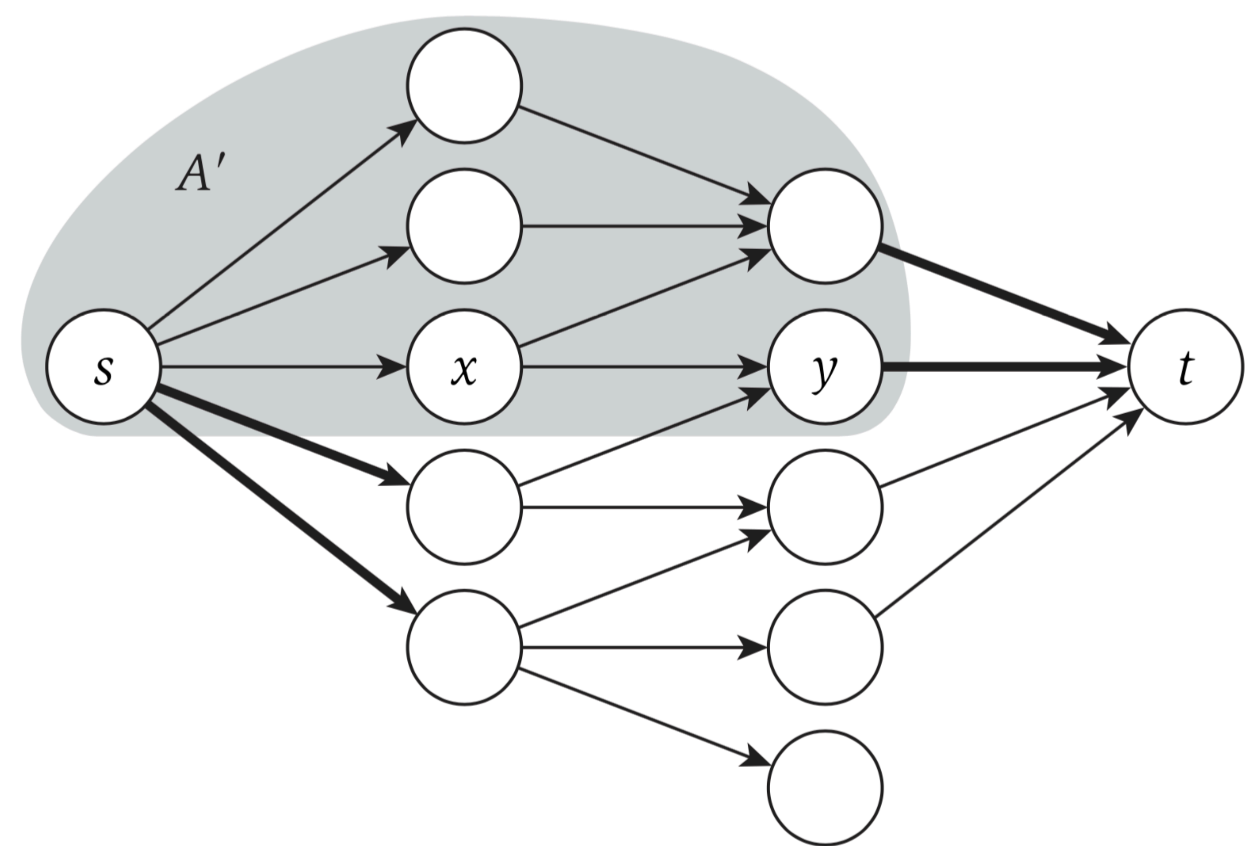
$N(A) \subseteq A'$

Perfect Matchings & Cuts

- Pick an edge (x, y) s.t. $x \in A$ and $y \notin A'$
- **Claim**: moving y to A' doesn't increase capacity of the cut ∎



$N(A) \subsetneq A'$



$N(A) \subseteq A'$

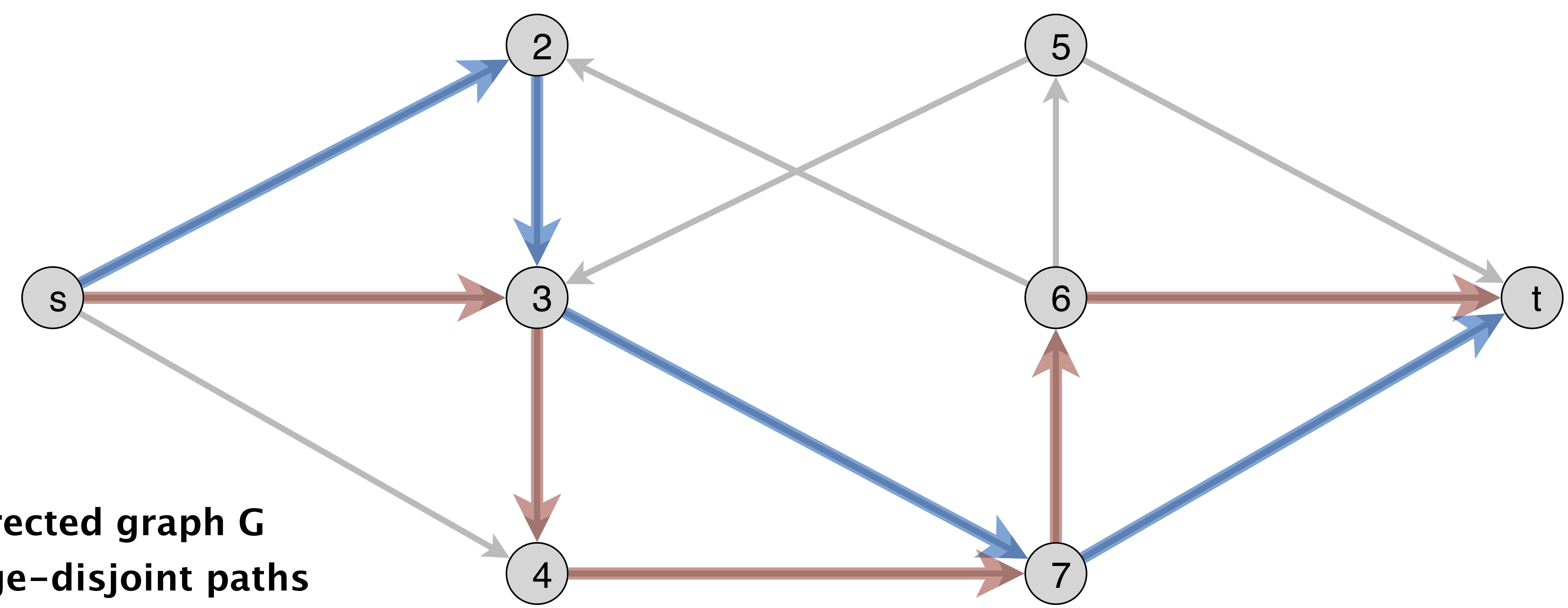
Summary: Flows and Matching

- We have proved Hall's theorem using network flows!
- **[Halls marriage theorem.]** Let $G = (X \cup Y, E)$ be a bipartite graph with $|X| = |Y|$. Then, graph G has a perfect matching iff $|N(S)| \geq |S|$ for all subsets $S \subseteq L$.
- If G has a perfect matching, we can find one using flow!
- If G doesn't have a perfect matching, we can find a certificate for this: a subset of nodes that violate Hall's condition!
- **Takeaway**. Algorithms can be useful in proving purely combinatorial math theorems!

Disjoint Paths Problem

Disjoint Paths Problem

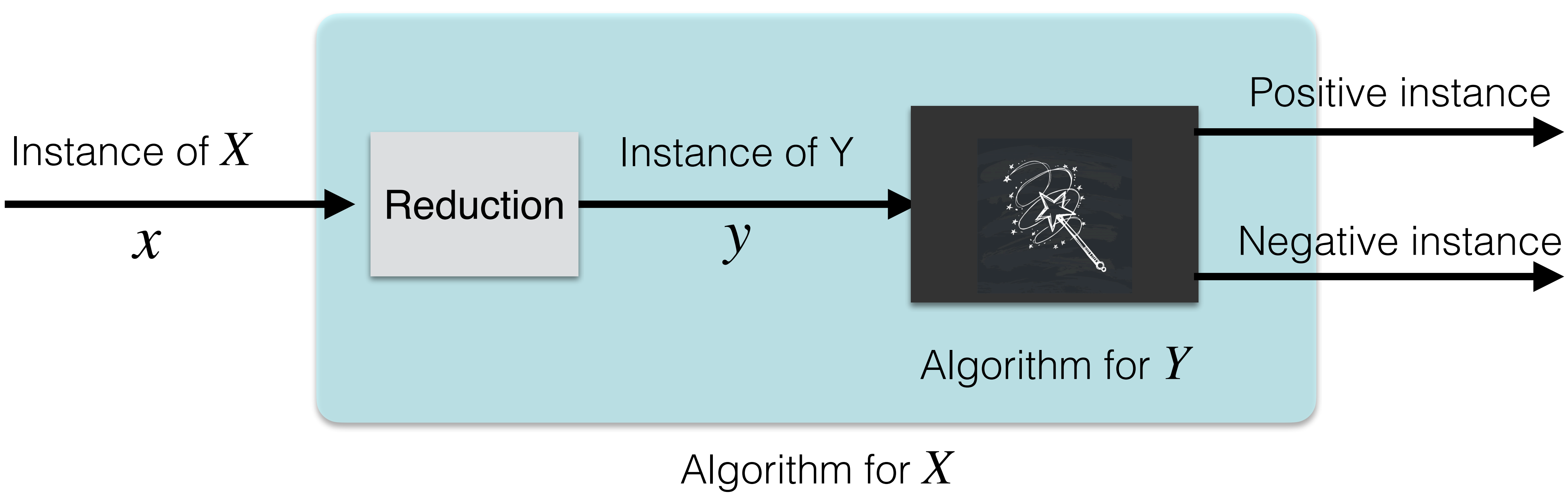
- **Definition.** Two paths are edge-disjoint if they do not have an edge in common.
- **Edge-disjoint paths problem.**
  Given a directed graph with two nodes s and t, find the max number of edge-disjoint $s \rightsquigarrow t$ paths.



Directed graph G
2 edge-disjoint paths

Towards Reduction

- Given: arbitrary instance x of disjoint paths problem (X): directed graph G, with source s and sink t
- **Goal**. create a special instance y of a max-flow problem (Y): flow network $G'(V', E', c)$ with s', t' s.t.
- **1-1 correspondence.** Input graph has k edge-disjoint paths iff flow network has a flow of value k



Reduction to Max Flow

- **Reduction.** G' : same as G with unit capacity assigned to every edge
- **Claim** [Correctness of reduction]. G has k edge disjoint $s \rightsquigarrow t$ paths iff G' has an integral flow of value k.
- Proof. $(\Rightarrow)$
- Set $f(e) = 1$ if e in some disjoint $s \rightsquigarrow t$, $f(e) = 0$ otherwise.
- We have $v(f) = k$ since paths are edge disjoint.
- $(\Leftarrow)$ Need to show: If G' has a flow of value k then there are k edge-disjoint $s \rightsquigarrow t$ paths in G

Correction of Reduction

- **Claim**. $(\Leftarrow)$ If f is a 0-1 flow of value k in G', then the set of edges where $f(e) = 1$ contains a set of k edge-disjoint $s \rightsquigarrow t$ paths in G.
- **Proof** [By induction on the # of edges k' with $f(e) = 1$]
- If $k' = 0$, no edges carry flow, nothing to prove
- IH: Assume claim holds for all flows that use $< k'$ edges
- Consider an edge $s \to u$ with $f(s \to u) = 1$
- By flow conservation, there exists an edge $u \to v$ with $f(u \to v) = 1$, continue "tracing out the path" until
- Case (a) reach t, Case (b) visit a vertex v for a 2nd time

Correction of Reduction

- **Case (a)** We reach t, then we found a $s \rightsquigarrow t$ path P
  - f' : Decrease the flow on edges of P by 1
  - $v(f') = v(f) - 1 = k - 1$
  - Number of edges that carry flow now $< k'$: can apply IH and find $k - 1$ other $s \rightsquigarrow t$ disjoint paths
- **Case (b)** visit a vertex v for a 2nd time: consider cycle C of edges visited btw 1st and 2nd visit to v
  - f' : decrease flow values on edges in C to zero
  - $v(f') = v(f)$ but # of edges in f' that carry flow $< k'$, can now apply IH to get k edge disjoint paths

$\blacksquare$

Summary & Running Time

- Proved k edge-disjoint paths iff flow of value k
- Thus, max-flow iff max # of edge-disjoint $s \rightsquigarrow t$ paths
- Running time of algorithm overall:
    - Running time of reduction + running time of solving the max-flow problem (dominates)
- What is running time of Ford–Fulkerson algorithm for a flow network with all unit capacities?
    - $O(nm)$
- Overall running time of finding max # of edge-disjoint $s \rightsquigarrow t$ paths: $O(nm)$

Survey Design

Survey Design

- Design survey asking n consumers about m products
- Can survey consumer i about product j only if they own it
- Ask consumer i at least a_i and at most b_i questions
- Ask at least p_j and at most q_j customers about product j

- **Problem.** Given an instance of this problem, determine if it is possible to design a survey that satisfies these requirements

- Challenge. We have lower bounds now in addition to "capacities" which serve as upper bounds
    - How do we handle that using flows?

Survey Design

- Design survey asking n consumers about m products
- Can survey consumer i about product j only if they own it
- Ask consumer i at least a_i and at most b_i questions
- Ask at least p_j and at most q_j customers about product j

- **Problem.** Given an instance of this problem, determine if it is possible to design a survey that satisfies these requirements

- Note. If $a_i = b_i = 1$ and $p_i = q_i = 1$, what can we say about this problem?
    - Same as finding perfect matching in bipartite graph!

Survey Design

- Design survey asking n consumers about m products
- Can survey consumer i about product j only if they own it
- Ask consumer i at least a_i and at most b_i questions
- Ask at least p_j and at most q_j customers about product j
- First step in reduction to network flow
    - What would the directed graph look like?
    - n nodes for consumers
    - m nodes for products
    - When is there an edge between a consumer & product?
        - Add $(i \rightarrow j)$ if consumer i owns product j

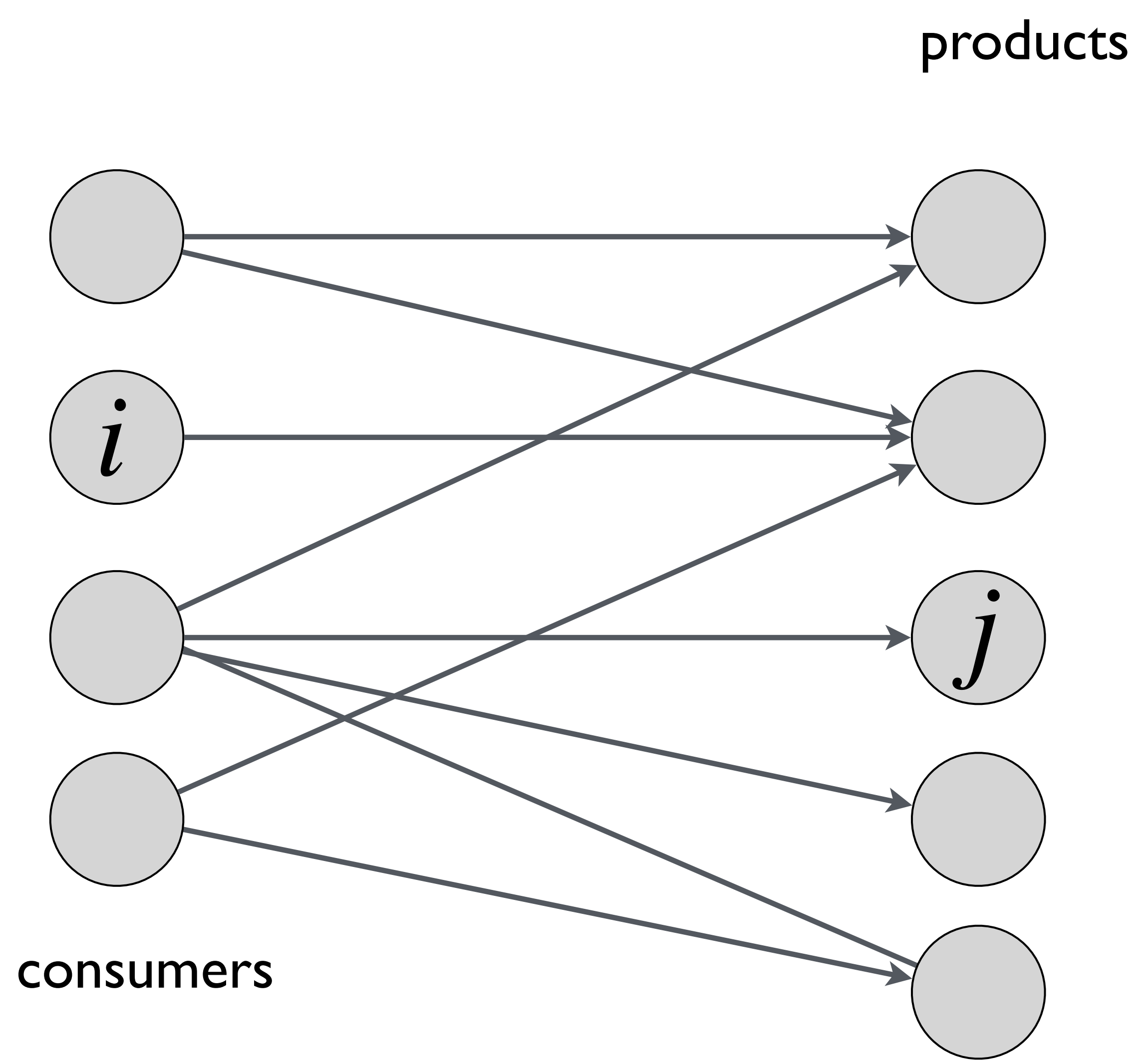
Survey Design

- Design survey asking n consumers about m products
- Can survey consumer i about product j only if they own it
- Ask consumer i at least a_i and at most b_i questions
- Ask at least p_j and at most q_j customers about product j
- Second step:
    - Find the mapping between problem and integral flow
- Either consumer i is asked about product j or not
- How can we map this to a flow on edge $i \to j$?
    - Either $f(i \to j) = 1$ or $f(i \to j) = 0$ respectively
- Next step: think about what the upper/lower bounds mean for flow coming in and out of these nodes

Survey Design

$f_{out}(i) =$ number of questions i is asked

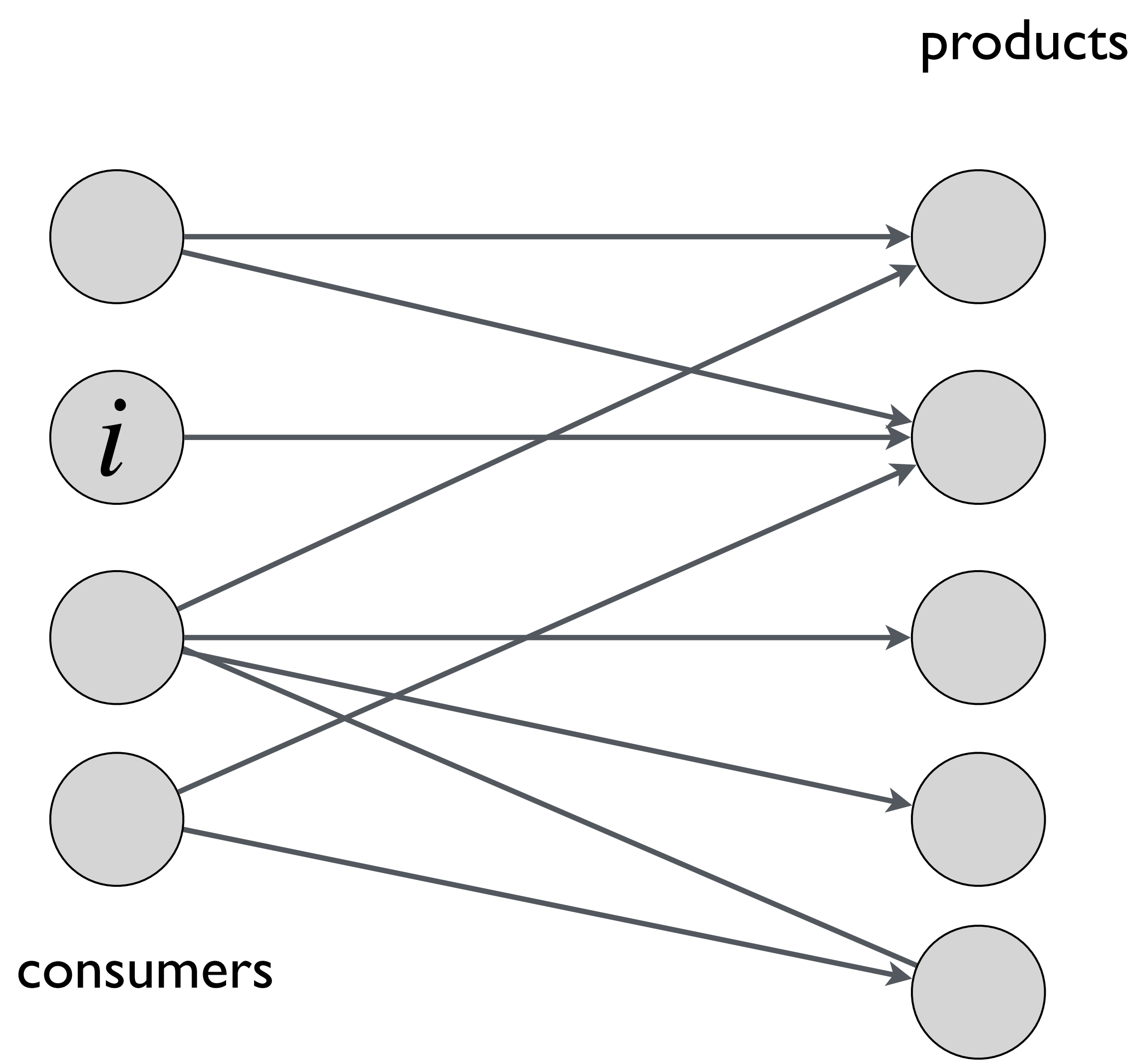
$f_{in}(j) =$ number of consumers asked about product j



$f(i \to j) = 1$ iff consumer i is asked about product j

Survey Design

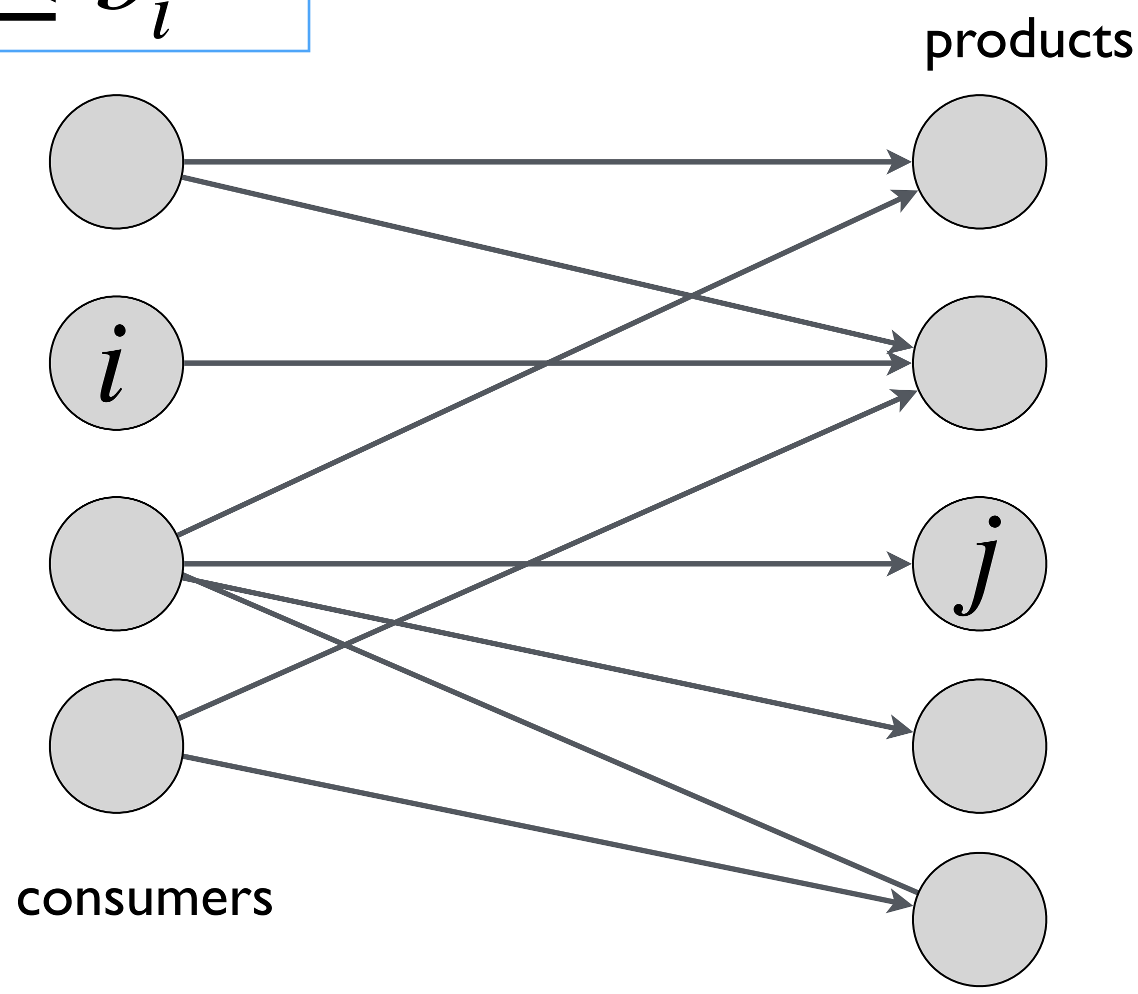
Consumer i should be asked at least a_i questions and at most b_i questions



$f(i \to j) = 1$ iff consumer i is asked about product j

Survey Design

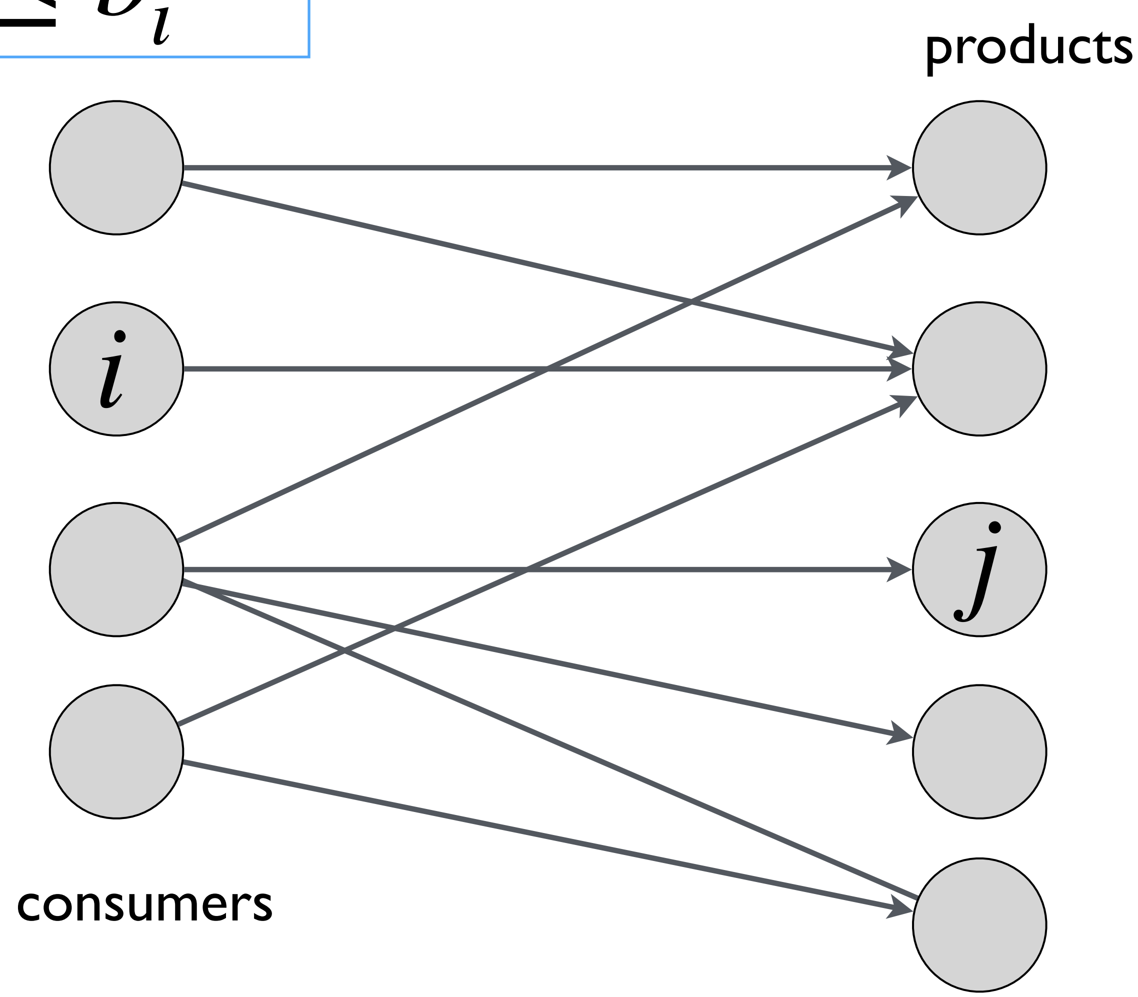
$$a_i \leq f_{out}(i) \leq b_i$$



$f(i \to j) = 1$ iff consumer i is asked about product j

Survey Design

$$\boxed{a_i \leq f_{out}(i) \leq b_i}$$

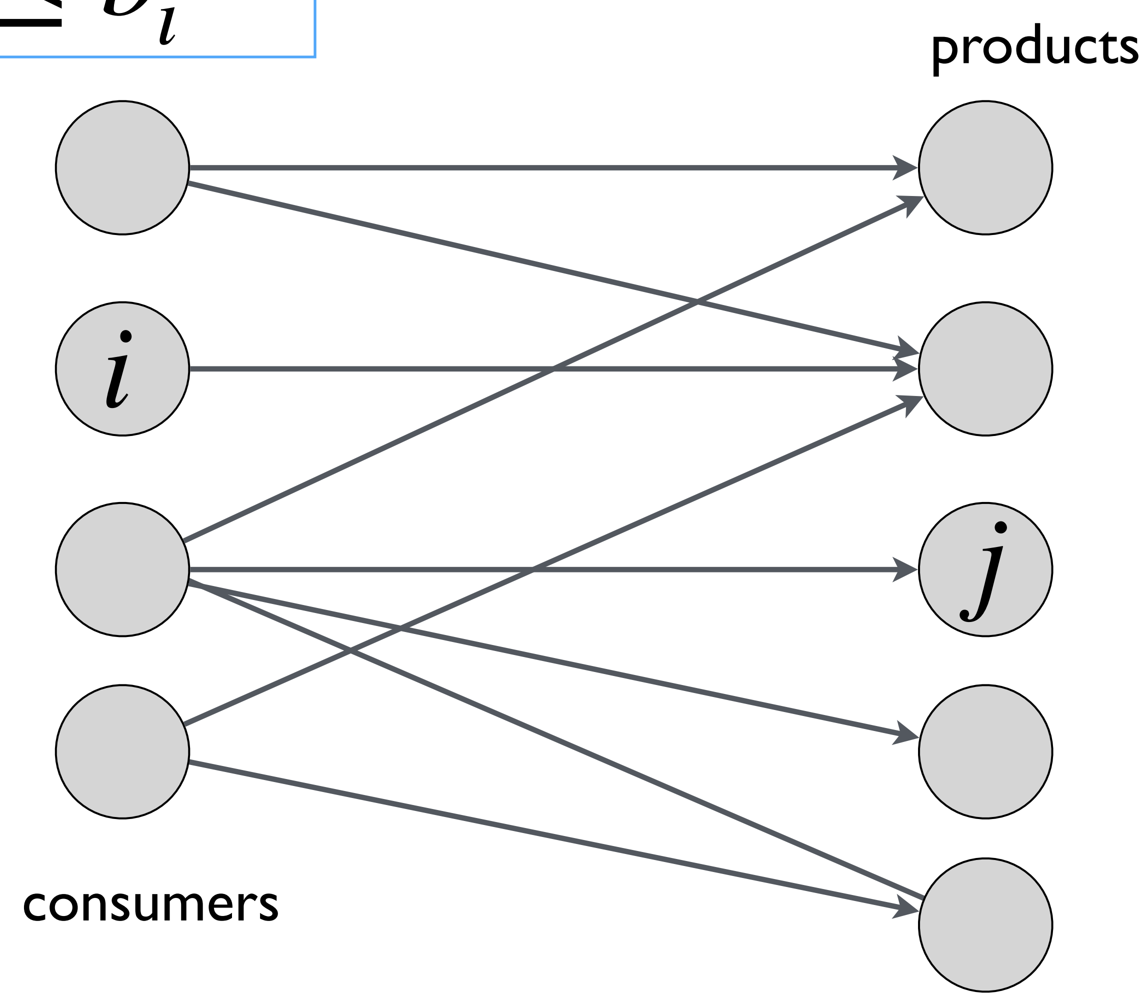


$f(i \to j) = 1$ iff consumer i is asked about product j

Survey Design

Ask at least p_j and at most q_j people about product j

$a_i \leq f_{out}(i) \leq b_i$

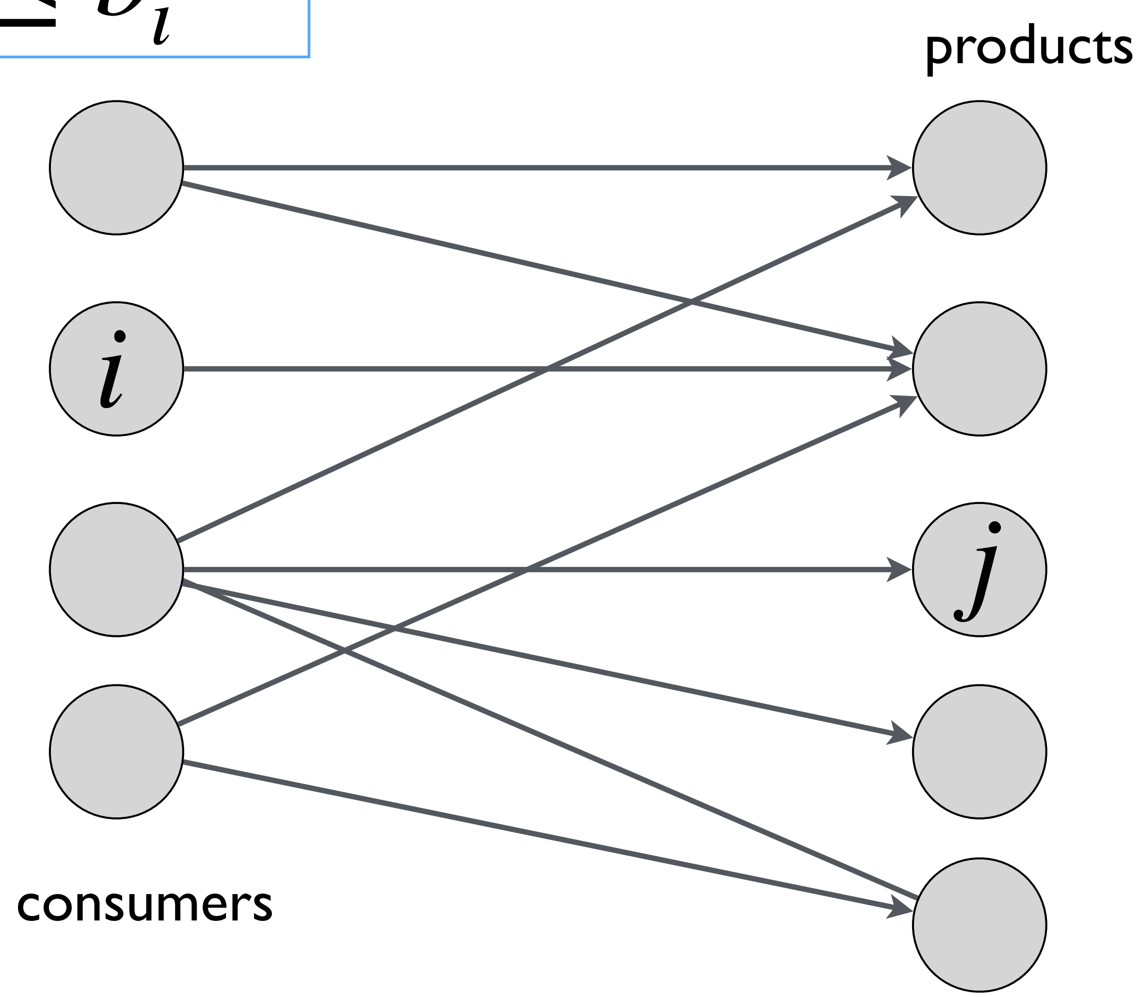


$f(i \rightarrow j) = 1$ iff consumer i is asked about product j

Survey Design

$$p_j \leq f_{in}(j) \leq q_j$$

$$a_i \leq f_{out}(i) \leq b_i$$

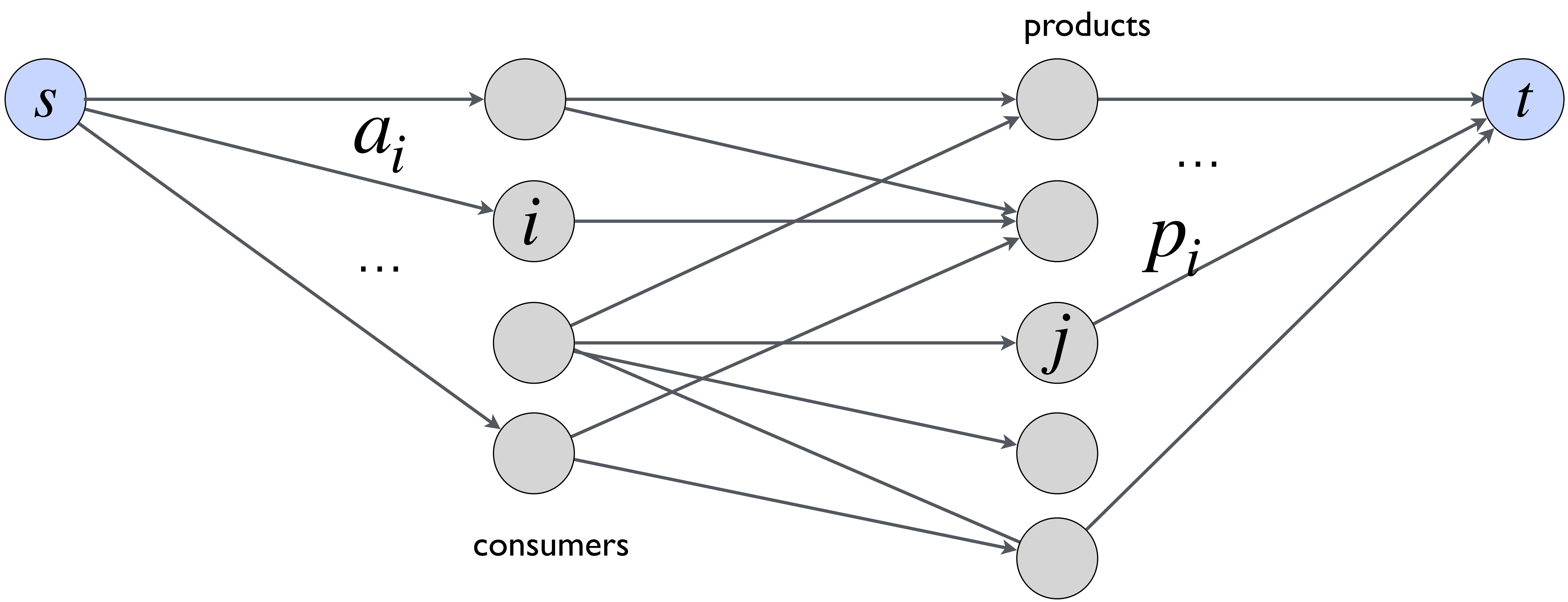


$f(i \to j) = 1$ iff consumer i is asked about product j

Survey Design

$$a_i \le f_{out}(i) \le b_i$$

$$p_j \le f_{in}(j) \le q_j$$

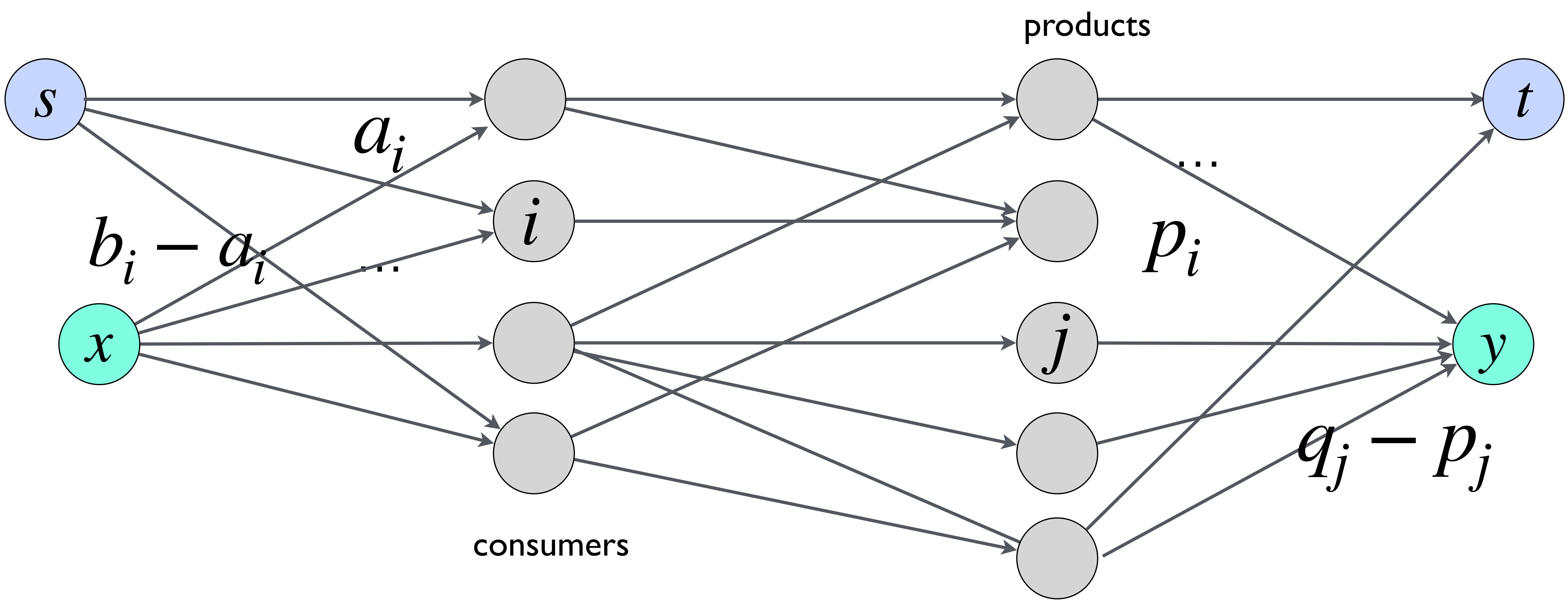


$f(i \to j) = 1$ iff consumer i is asked about product j

Survey Design

$$a_i \leq f_{out}(i) \leq b_i$$

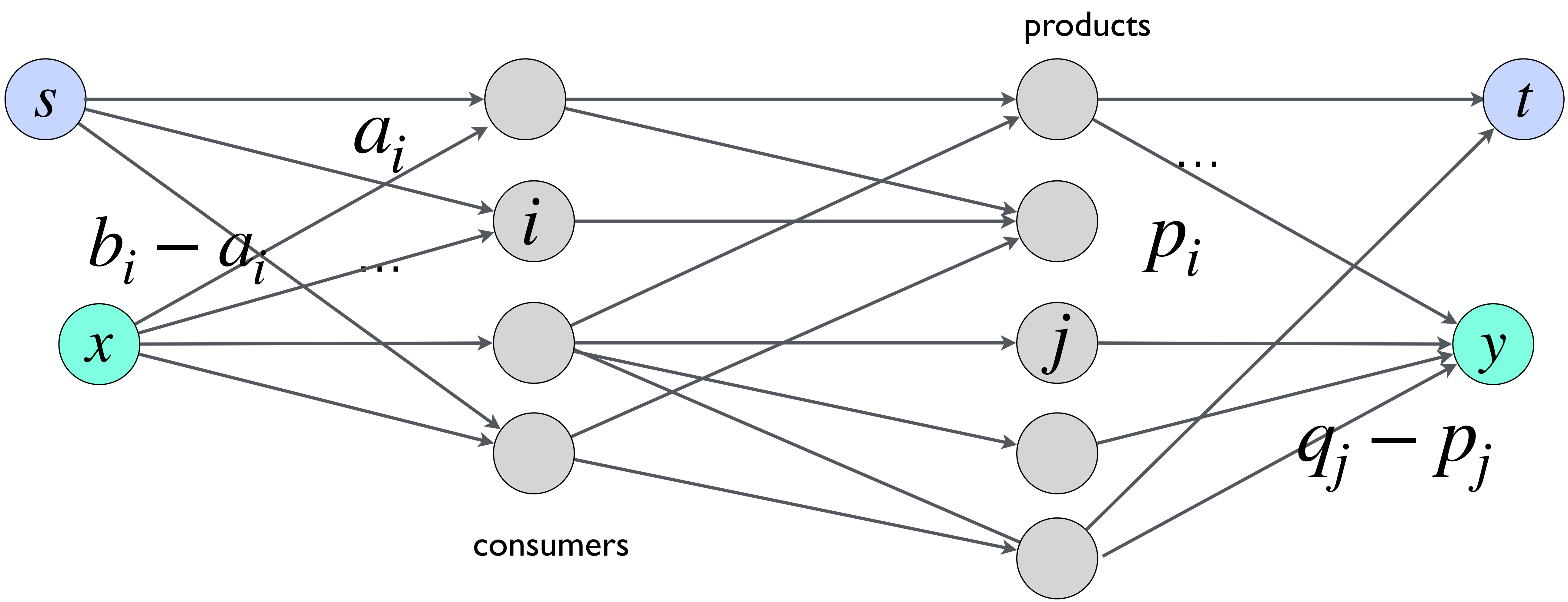
$$p_j \leq f_{in}(j) \leq q_j$$



$f(i \to j) = 1$ iff consumer i is asked about product j

Survey Design

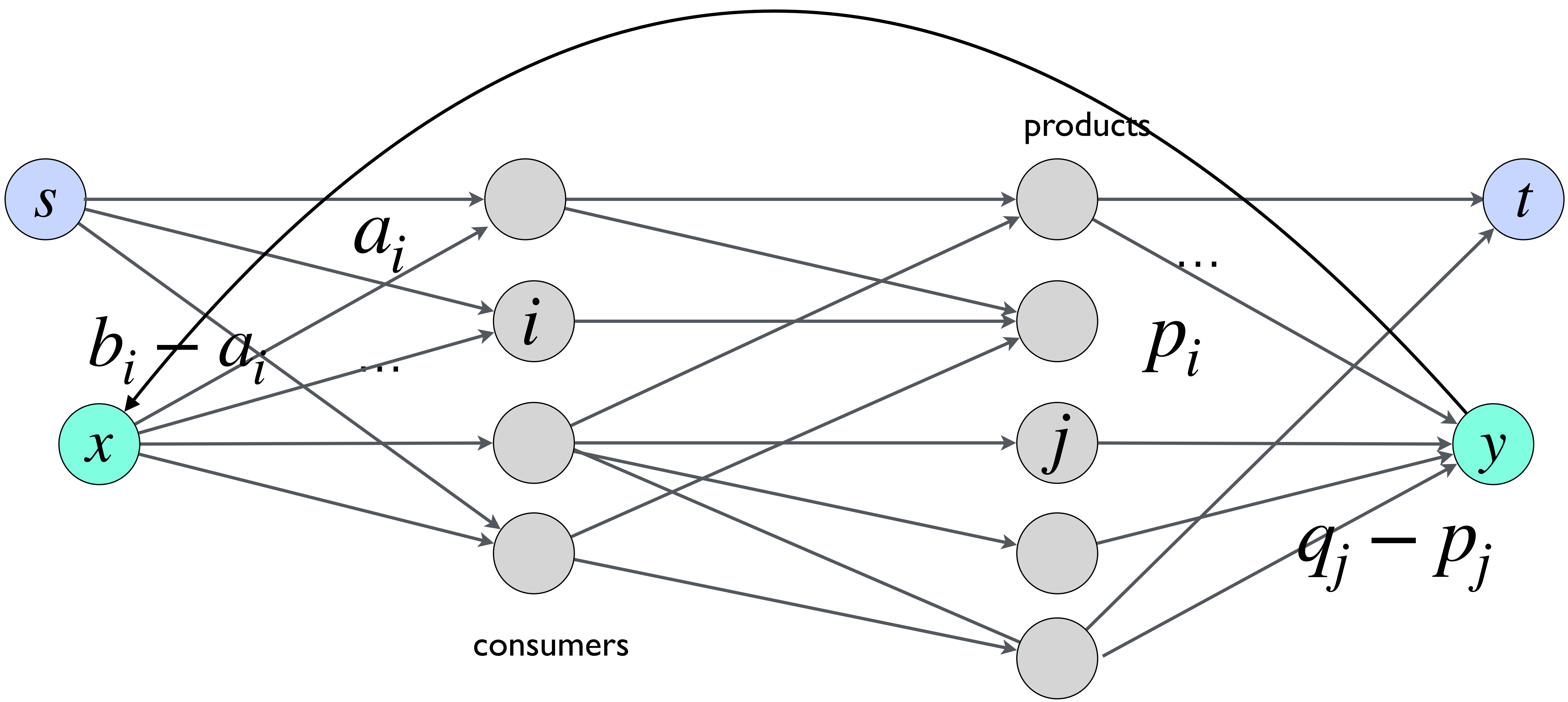
Assume $\sum_i a_i = \sum_j p_j$



$f(i \to j) = 1$ iff consumer i is asked about product j

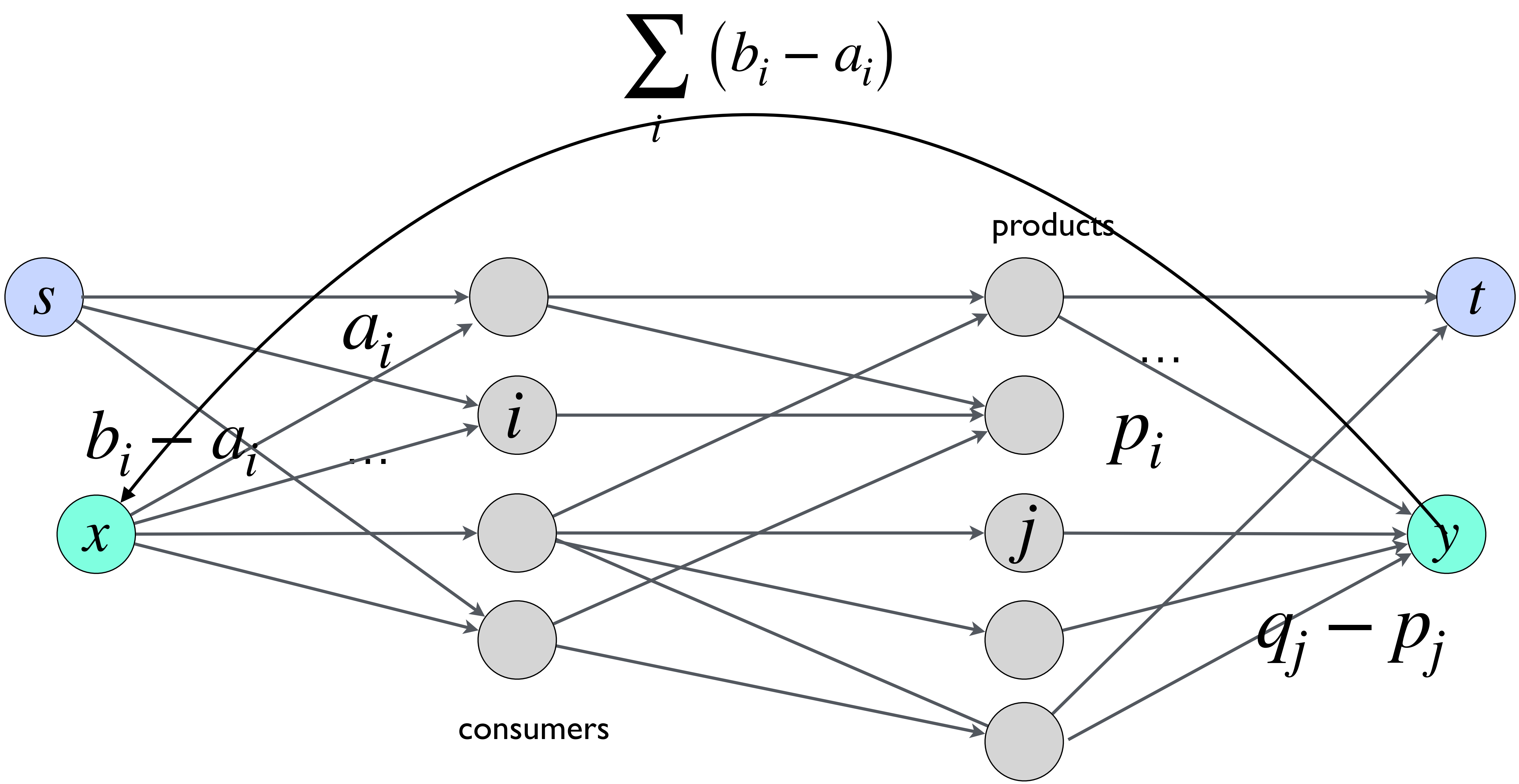
$$\text{Assume} \sum_i a_i = \sum_j p_j$$

number of extra questions



$f(i \to j) = 1$ iff consumer i is asked about product j

Assume $\sum_i a_i = \sum_j p_j$



$f(i \to j) = 1$ iff consumer i is asked about product j

Survey Design: Reduction

- Nodes in flow network: s, t, x, y and a node i for each consumer, node j for each product
- Edges and capacities:
  - Edge $i \to j$ with capacity 1 for each consumer i and product j if consumer i owns product j
  - Edge $s \to i$ for each consumer i with capacity a_i
  - Edge $j \to t$ for each product j with capacity p_i
  - Edge $x \to i$ for each consumer i with capacity $a_i - b_i$
  - Edge $j \to y$ for each product j with capacity $q_i - p_i$
  - Edge $y \to x$ with capacity $\sum_i (b_i - a_i)$

Survey Design: Reduction

- **Claim**. It is possible to design a survey satisfying the constraints of the problem iff the corresponding flow network has an integral max flow of value $\sum_i a_i$
- $(\Rightarrow)$ Suppose it is possible to design such a survey
- Let $f(s \to i) = a_i$, $f(j \to t) = p_i$ for each i, j
- Let $f(i \to j) = 1$ iff consumer i is asked a question about product j
- Let $f(x \to i) =$ total # questions i is asked $- a_i$
- Let $f(j \to y) =$ total # questions about product $j - p_j$
- Let $f(y \to x) =$ total # of questions overall $- \sum_i a_i$

Survey Design: Reduction

- **Claim**. It is possible to design a survey satisfying the constraints of the problem iff the corresponding flow network has an integral max flow of value $\sum_i a_i$
- $(\Rightarrow)$ Suppose it is possible to design such a survey
- Value of such a flow is $\sum_i a_i = \sum_j p_j$
- Check flow conservation at each node and capacity constraints using the constants on questions asked in the survey problem
    - Convince yourself at home

Survey Design: Reduction

- **Claim**. It is possible to design a survey satisfying the constraints of the problem iff the corresponding flow network has an integral max flow of value $\sum_i a_i$
- $(\Leftarrow)$ Suppose the max flow has value $\sum_i a_i$
- Ask consumer i a question about product j iff $f(i \to j) = 1$
- Check: each consumer is asked between a_i and b_i questions
- Check: between p_j and q_j consumers are asked about product j

Acknowledgments

- Some of the material in these slides are taken from
  - Kleinberg Tardos Slides by Kevin Wayne (https://www.cs.princeton.edu/~wayne/kleinberg-tardos/pdf/04GreedyAlgorithmsI.pdf)
  - Jeff Erickson's Algorithms Book (http://jeffe.cs.illinois.edu/teaching/algorithms/book/Algorithms-JeffE.pdf)