Probability in Computing

LECTURE 4: CONDITIONAL PROBABILITY, APPLICATIONS, GEOMETRIC DISTRIBUTION

Agenda

- Application: Verification of Matrix Multiplication
- Application: Randomized Min-Cut
- Geometric Distribution
- Coupon Collector's Problem

Application: Verifying Matrix Multiplication

- Consider matrix multiplication AB = C (integers modulo 2)
  - Simple algorithm takes $O(n^3)$ operations.
  - Want to check if a given matrix multiplication program works correctly
- Randomized Algorithm:
  - Choose a random vector $r = (r_1, r_2, \dots, r_n)$ in $\{0,1\}^n$.
  - Compute A(Br) and Cr then comparer the two values: if equal return yes AB=C, else no.
- Note on the above randomized algorithm:
  - 1-side error
  - Complexity = $O(n^2)$
  - Accuracy depends on P(ABr = Cr) when AB!=C

Analysis of P(ABr = Cr)

- Choosing r randomly is equivalent to choosing r_i randomly and independently. (1)
- Let D = AB – C ≠ 0. Since Dr = 0, there must be some non-zero entry. Let that be d_{11}.
  - $Dr = 0 \rightarrow \Sigma d_{1j}r_j = 0 \rightarrow r_1 = -\Sigma d_{1j}r_j / d_{11}$.
  - Since r1 can take 2 values, combine with (1), we have ABr = Cr with probability of at most ½
  - Refer to book for formal proof (using Law of Total Probability)
- Principle of Deferred Decisions: when there are several random variables, it often helps to think of some of them as being set at one point in the algorithm with the rest of them being left random (or deferred) until some further point in the analysis.
- We can attempt this verification k times to obtain accurate answer with $p = 2^{-k}$ and efficiency = $O(kn^2) = O(n^2)$

Theorems

- Law of Total Probability: Assume E_1, E_2, …, E_n be mutually disjoint events in the sample space Ω and union of E_i = Ω. Then
  - $Pr(B) = \sum Pr(B \text{ and } E_i) = \sum Pr(B|E_i)Pr(E_i)$

- Bayes' Law: Assume E_1, E_2, …, E_n be mutually disjoint events in the sample space Ω and union of E_i = Ω. Then

  $Pr(E_j|B) = Pr(E_j \text{ and } B)/Pr(B)$
  $= Pr(B|E_j)Pr(E_j) \,/\, \sum Pr(B|E_i)Pr(E_i)$

  - Notice the model transformation from prior probability to posterior probability.

Gradual Change in Our Confidence in Algorithm Correctness

- In matrix verification case:
  - E = the identify is correct
  - B = test returns that the identity is correct
- Prior assumption: Identity = ½
  - How does this assumption change after each run?
- We start with $\Pr(E) = \Pr(E^c) = ½$
- Since the test has error bounded by ½, $\Pr(B|E^c) \leq ½$. Also, $\Pr(B|E) = 1$
- Now by Bayes' Law:

$\Pr(E|B) = \Pr(B|E)\Pr(E) / \{\Pr(B|E)\Pr(E) + \Pr(B|E^C)\Pr(E^C)\} \geq$

$½ / \{1.½ + ½. ½\} = 2/3$

Gradual Change in Our Confidence in Algorithm Correctness

- The prior model is revised:
  - $\Pr(E) \geq 2/3$ and $\Pr(E^c) \leq 1/3$.
- Applying Bayes' Law again will yeild $\Pr(E|B) \geq 4/5$
- In general, at i^th iteration, $\Pr(E|B) \geq 1 - 1/(2^i+1)$
- After 100 calls, test returns that identity is correct, then our confidence in the correctness of this identity is at least $1 - 1/2^{100}+1)$

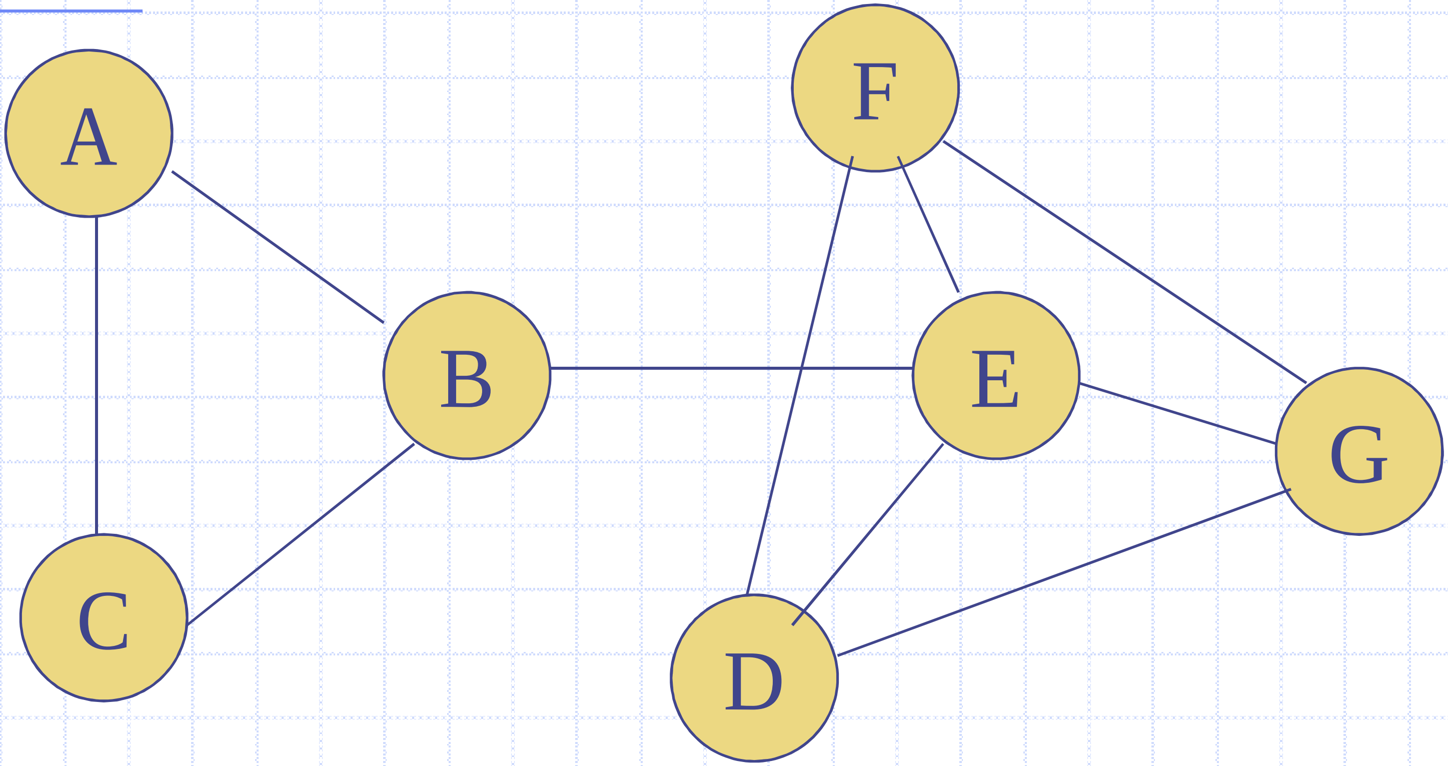
Application: Randomized Min Cut

- Cut-set: Set of edges whose removal breaks the graph into two or more connected components.

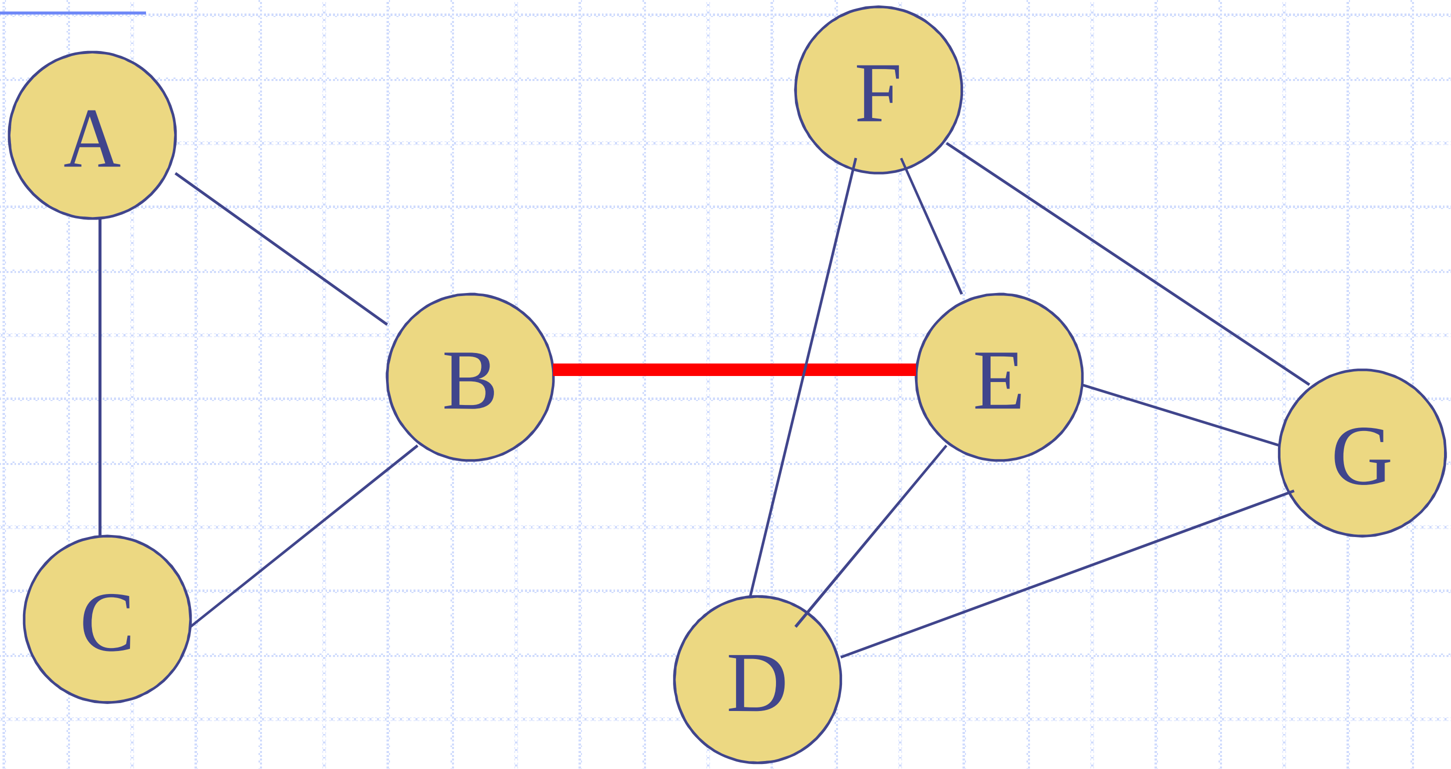
- Min-cut: Cut-set with minimum cardinality.

- Applications:
  - Network reliability.
  - Clustering problems
  - Al-Qaeda

Example



Example



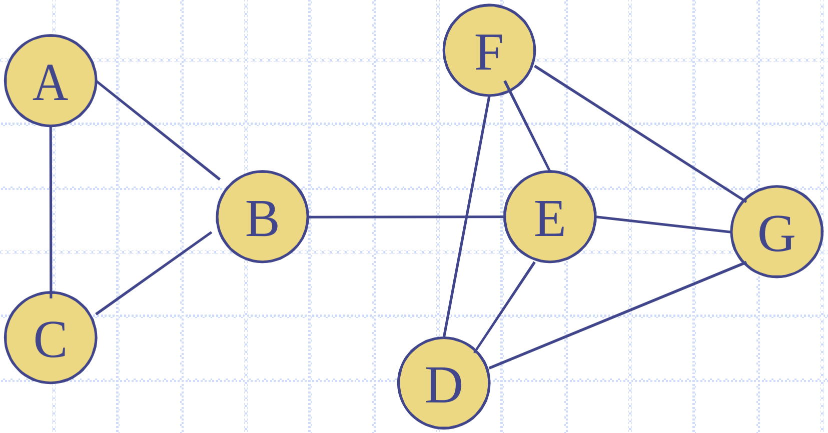
$$E_1 = \left\{ \overline{BE} \right\}$$

Karger algorithm

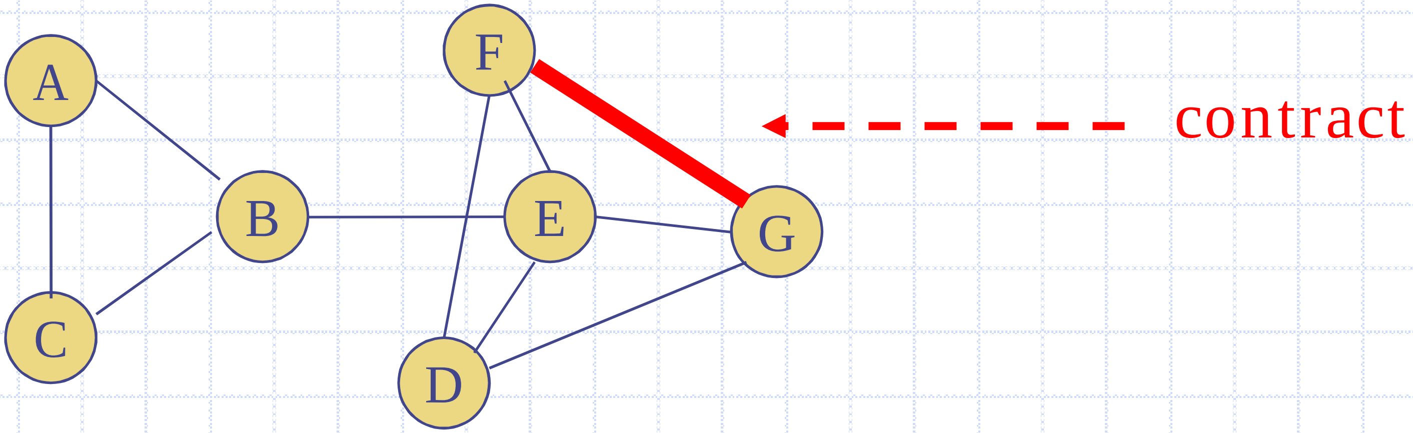
- Edge contraction (collapsing an edge): To collapse edge {u,v}, we
  - Create a new node uv
  - Replace any edge of form u, w or v, w with new edge uv, w
  - Delete original vertices u and v.
  - Resulting graph is denoted G/{u,v}.

- Repeat until 2 nodes left (n-2 iterations):
  - Choose edge at random
  - “Contract” edge.
- Take all the edges between them as min-cut

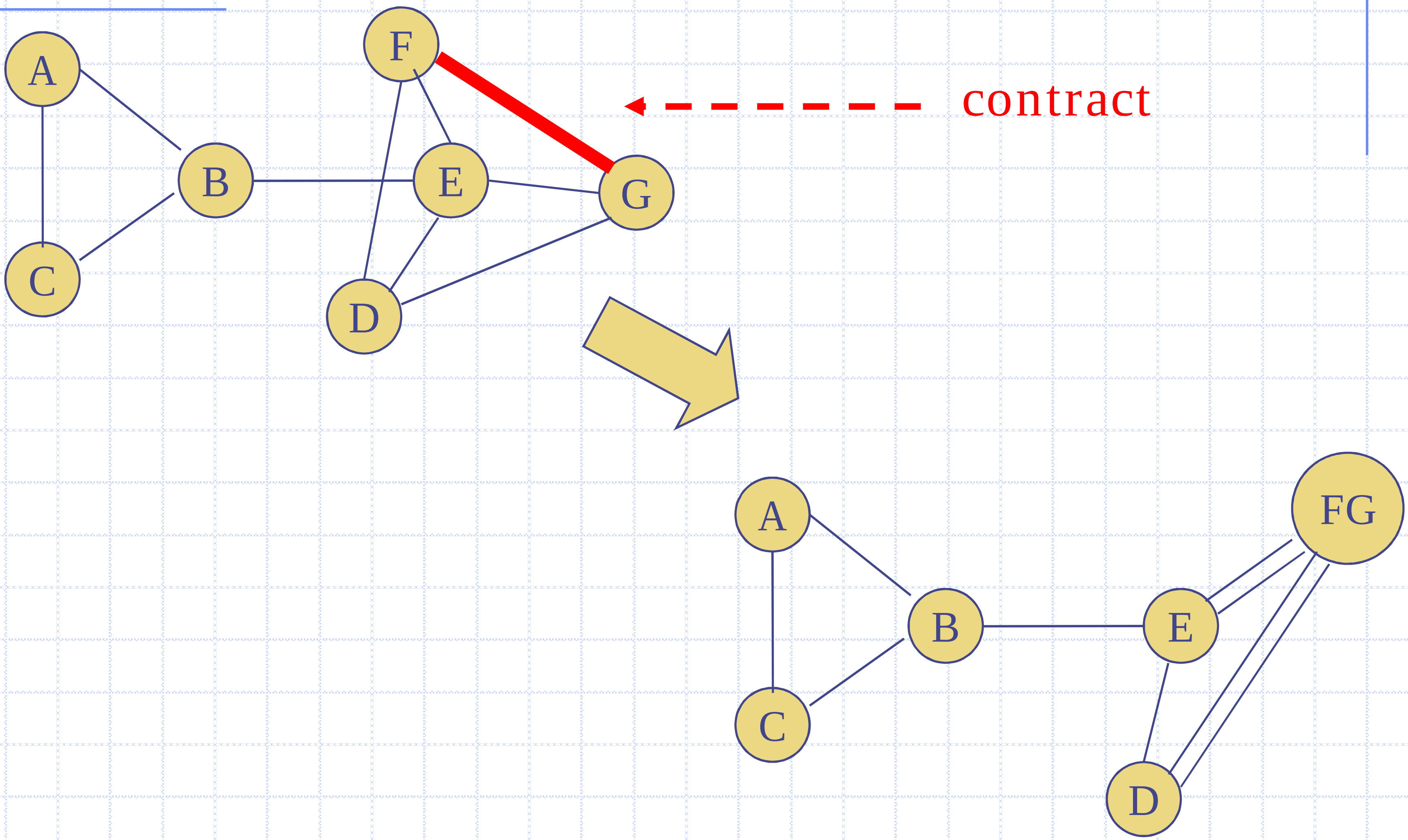
Example



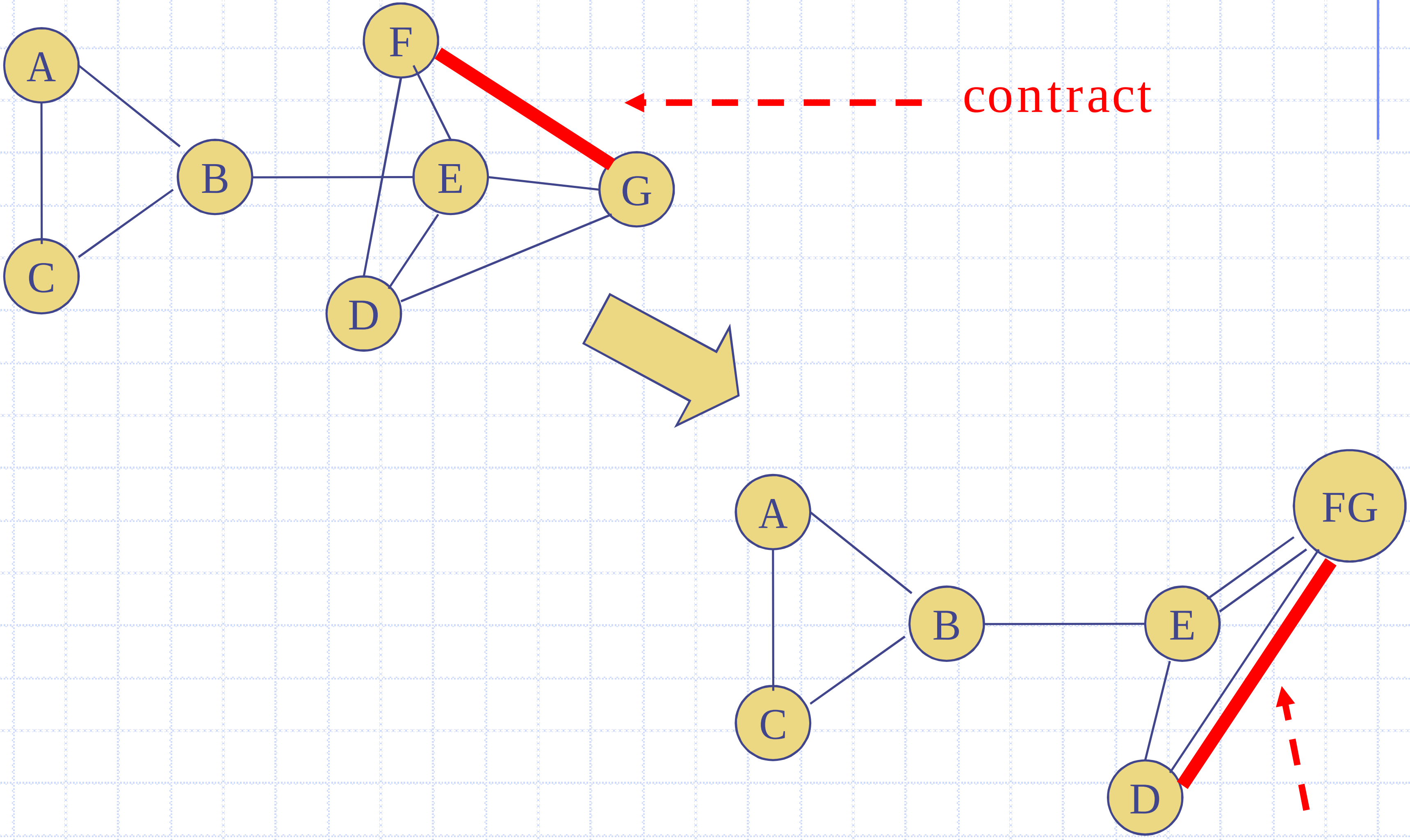
Example



Example



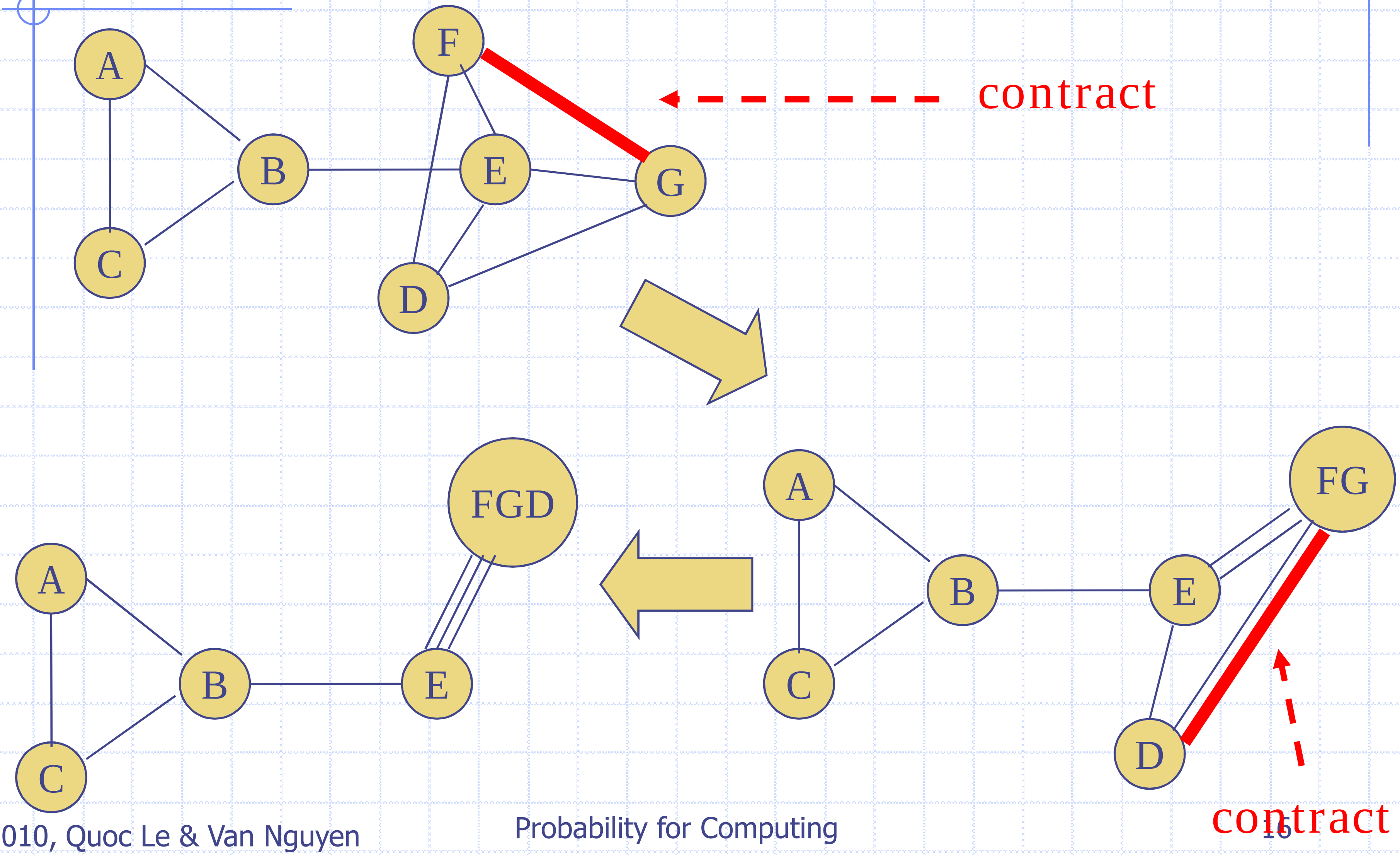
Example



contract

contract

Example



Analysis

- Let k be the size of the min-cut set of G
  - We want to compute the probability of finding one such set C.
- C partition V (set of vertices) in to S and V-S
  - If the algorithm never choose an edge in C in its n-2 iterations, then the algorithm will return C as minimum cut-set.
- Let E_i = edge contracted in iteration i is not in C
- Let F_i = Union of E_j ($j = 1 \rightarrow i$) = no edge of C was contracted in the first i iterations.
- We need to compute $\Pr(F_{n-2})$

Analysis

- All vertices have degree k or larger → graph must have ≥ nk/2 edges.

  $\Pr(F_1) = \Pr(E_1) \geq 1 – k/ (nk/2) = 1 – 2/n$

- Conditionally:
  - $\Pr(E_2|F_1) \geq 1 – k/ (k(n-1)/2) = 1 – 2/(n-1)$.
  - Similarly: $\Pr(E_i|F_{i-1}) \geq 1 – 2/(n-i+1)$

- Therefore:

  $\Pr(F_{n-2}) = \Pr(E_{n-2}|F_{n-3})* \ \Pr(F_{n-3})+ \Pr(E_{n-2}|F_{n-3}^c)* \ \Pr(F_{n-3}^c)$

  $= \Pr(E_{n-2}|F_{n-3})* \ \Pr(F_{n-3})$ since $\Pr(E_{n-2}|F_{n-3}^c)=0$

- So,

  $\Pr(F_{n-2}) = \Pr(E_{n-2}|F_{n-3})* \ \Pr(F_{n-3}) = \Pr(E_{n-2}|F_{n-3})* \Pr(E_{n-3}|F_{n-4}) * \Pr(F_{n-4}) =$

  $\Pr(E_{n-2}|F_{n-3})* \Pr(E_{n-3}|F_{n-4}) *\dots * \Pr(E_2|F_1) * \Pr(F_1) = 2/n(n-1)$.

What's next

- Karger: Use of idea of amplification – Run algorithm many times and return the smallest guess.
- Our probability of success $\geq 1 - (1 - 2/n(n-1))^N \geq 1 - e^{-2N/n(n-1)}$. (due to $1\text{-}x \leq e^{-x}$)
- Choose N = c(n choose 2) ln(n), for some constant c, then it is correct with probability at least $1 - 1/n^c$.
- Complexity = $O(n^4 \log n)$.
- We can reduce the time complexity by an order of n2 to obtain $O(n^2(\log n)^3)$ - http://www.cs.dartmouth.edu/~ac/Teach/CS105-Winter05/Handouts/05-mincut.pdf

Geometric Distribution

- Flip a coin until it lands on head. What is the distribution of the number of flips?
- Perform a sequence of independent trials until the first success, where each trial succeeds with prob. = p.
- Def: A geometric random variable X with parameter p is given by the following probability distribution:

$$\Pr(X=n)=(1-p)^{n-1}.p$$

Properties

- Lemma 1: (Memory-less): $\Pr(X=n+k|X>k) = \Pr(X=n)$

- Lemma 2: Let X be a discrete random variable that takes on only non-negative integer values. Then: $E[X] = \sum \Pr(X \geq i)$ $(i = 1 \rightarrow \infty)$
  - For a geometric random variable X(p), $\Pr(X \geq i) = (1-p)^{i-1}$ → $E[X] = 1/p$.

Coupon Collector Problem

- Problem: Suppose that each box of cereal contains one of n different coupons. Once you obtain one of every type of coupon, you can send in for a prize.

- Question: How many boxes of cereal must you buy before obtaining at least one of every type of coupon.

- Let X be the number of boxes bought until at least one of every type of coupon is obtained.
- Let X_i be number of boxes bought while you had exactly i-1 different coupons.
- Then: $X = \sum X_i$.

Coupon Collector's Problem

- Probability of obtaining a new coupon when i -1 coupons have been found: $p_i = 1 – (i\text{-}1)/n$.

- Hence, $E[X_i] = 1/p_i = n/(n\text{-}i+1)$

- $E[X] = \Sigma E[X_i] = n \Sigma 1/i = nH(n)$ (harmonic number) = $n\ln(n)$.

Application: Packet Sampling

- Sampling packets on a router with probability p
  - The number of packets transmitted after the last sampled packet until and including the next sampled packet is geometrically distributed.
- From the point of destination host, determining all the routers on the path is like a coupon collector's problem.
- If there's n routers, then the expected number of packets arrived before destination host knows all of the routers on the path = nln(n).