TTIC 31150/CMSC 31150 Mathematical Toolkit (Fall 2024)

Avrim Blum

Lecture 10: Probabilistic reasoning

Recap

- Definitions of sample space Ω, events, random variables, expectation, conditional probability, conditional expectation.
- Going back and forth between events and R.V.s with indicator R.V.s
- Linearity of expectation, examples.
- Independence of events and R.V.s. Mutual independence vs pairwise independence.
- Properties of independence: $E[XY]=E[X]E[Y]$ if independent.
- Application: universal hashing.
- Bernoulli R.V.s, Binomial R.V.s, Geometric R.V.s.

The Probabilistic Method

An approach to showing something exists by defining a probability space and showing it has non-zero probability of occurring.

Example: A graph with m edges must have a cut (a partitioning of the vertices into two sets A and B) such that at least $m/2$ edges cross the cut.

Proof: Consider randomly putting each vertex into A or B independently with prob. ½.

- Each edge (i, j) has probability ½ of crossing the cut.
- So, the expected number of edges that cross is $m/2$.
- So, there must exist a partitioning in which at least $m/2$ edges cross.

The Probabilistic Method

An approach to showing something exists by defining a probability space and showing it has non-zero probability of occurring.

Example #2: Suppose F is a k-CNF formula with $m < 2^k$ clauses, where every clause has size exactly k (and no variable can be repeated in a clause). Then F must be satisfiable.

Proof: Consider a random assignment x.

- Let C_j be the indicator R.V for the event that clause j is satisfied. $\mathbb{E}[C_j] = 1 - 1/2^k$.
- By linearity of expectation, the expected total number of clauses satisfied is $m(1 - 1/2^k) > m - 1$.
- So, there must be at least one assignment that satisfies all m clauses.

The Probabilistic Method

An approach to showing something exists by defining a probability space and showing it has non-zero probability of occurring.

More generally, for any value of m, there must exist an assignment that satisfies at least $\lceil m(1-1/2^k) \rceil$ clauses. E.g., for a 3-CNF, there must exist an assignment that satisfies at least a 7/8 fraction of the clauses.

Proof: Consider a random assignment x.

- Let C_j be the indicator R.V for the event that clause j is satisfied. $\mathbb{E}[C_j] = 1 - 1/2^k$.
- By linearity of expectation, the expected total number of clauses satisfied is $m(1 - 1/2^k) > m - 1$.
- So, there must be at least one assignment that satisfies all m clauses.

The Probabilistic Method

An approach to showing something exists by defining a probability space and showing it has non-zero probability of occurring.

More generally, for any value of m, there must exist an assignment that satisfies at least $\lceil m(1-1/2^k) \rceil$ clauses. E.g., for a 3-CNF, there must exist an assignment that satisfies at least a 7/8 fraction of the clauses.

- This gives an efficient randomized algorithm to find such an assignment, since the R.V. "# clauses satisfied" is a non-negative integer and bounded by m.

- On your homework, you will give an efficient deterministic algorithm.

The Coupon Collector Problem

Imagine we have n bins. At each time step we place a ball into a bin independently at random. How long will it take in expectation until all bins have at least one ball?

- Let X denote the time to fill all n bins. Let's decompose X as $X = \sum_i X_i$ where X_i denotes the time to fill the ith new bin.

- Each X_i is a geometric R.V. with parameter $\frac{n-i+1}{n}$, so $\mathbb{E}[X_i] = \frac{n}{n-i+1}$.

- Therefore, $\mathbb{E}[X] = \frac{n}{n} + \frac{n}{n-1} + \frac{n}{n-2} + \cdots + \frac{n}{1} = n \cdot H_n = n \cdot \ln n + \Theta(n)$.

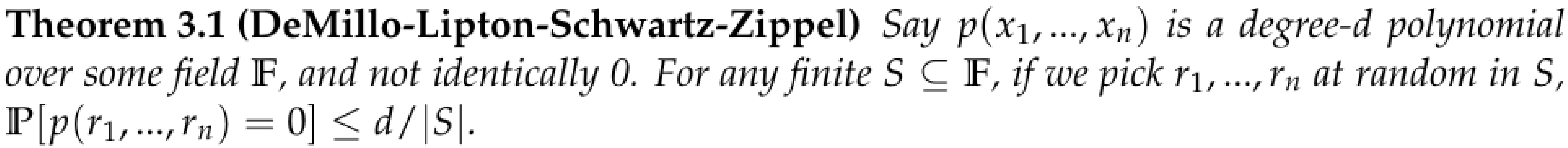
The DeMillo-Lipton-Schwartz-Zippel lemma

Most problems with randomized polynomial-time algorithms also have known deterministic polynomial-time algorithms. But there are a few hold-outs.

Here is one: like an algebraic version of the SAT problem.

- Say p is an n-variable polynomial of degree d, over a field $\mathbb{F}$ of size $\geq 2d$.
- Assume p is given in a form that can be evaluated efficiently, e.g.,
  $$(x_1 - 2x_2 + 3)(2x_1 - x_3) - (3x_1 + 2)(x_2 + x_3)$$
- Question: is p identically 0? Or, does there exist $x \in \mathbb{F}^n$ such that $p(x) \neq 0$?
- Equivalently, given two polynomials p_1, p_2, are they identical (does $p_1(x) - p_2(x) = 0$ always?

The DeMillo-Lipton-Schwartz-Zippel lemma



Theorem 3.1 (DeMillo-Lipton-Schwartz-Zippel) *Say $p(x_1, \dots, x_n)$ is a degree-d polynomial over some field $\mathbb{F}$, and not identically 0. For any finite $S \subseteq \mathbb{F}$, if we pick $r_1, \dots, r_n$ at random in S, $\mathbb{P}[p(r_1, \dots, r_n) = 0] \leq d/|S|$.*

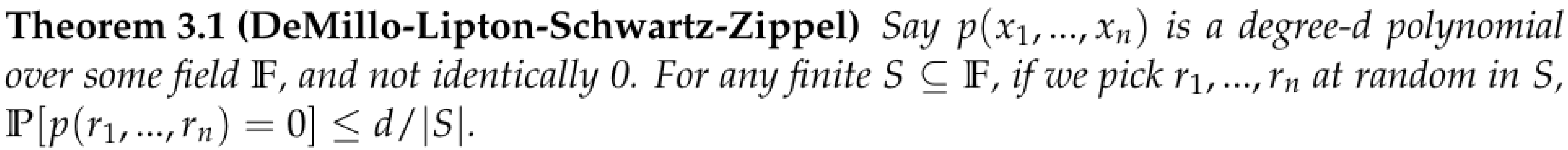
So, if $|\mathbb{F}| \geq 2d$ then we can just pick random inputs and try. See if we get 0 for k times in a row. If p was not identically 0, there would be at most a $1/2^k$ chance this would happen.

Will prove the theorem by induction on the number of variables n.

Base case: $n = 1$. This follows from the fact that a degree d polynomial in 1 variable has at most d roots.

What makes the inductive case harder is that a degree d polynomial in more than one variable could have an infinite number of roots. E.g., $x_1(x_2 - 1)$.

The DeMillo-Lipton-Schwartz-Zippel lemma



Theorem 3.1 (DeMillo-Lipton-Schwartz-Zippel) *Say $p(x_1, ..., x_n)$ is a degree-d polynomial over some field $\mathbb{F}$, and not identically 0. For any finite $S \subseteq \mathbb{F}$, if we pick $r_1, ..., r_n$ at random in S, $\mathbb{P}[p(r_1, ..., r_n) = 0] \leq d/|S|$.*

General case: Assume true for $n - 1$. Let i be the max degree of x_n.

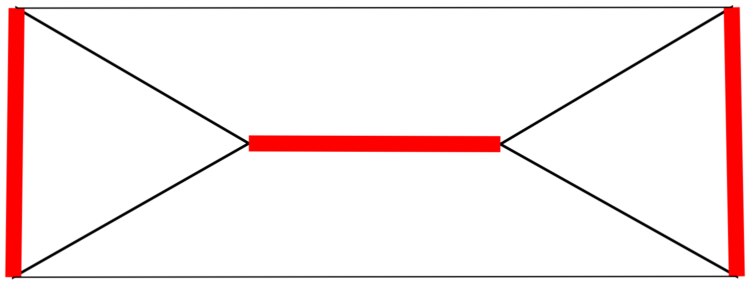
- Can write p as $x_n^i \cdot p_i\,(x_1, \ldots, x_{n-1}) + x_n^{i-1} \cdot p_{i-1}(x_1, \ldots, x_{n-1}) + \ \ldots$

  where p_i is not identically 0, and p_i has degree $\leq d - i$.
- Now, pick $x_1, \ldots, x_{n-1}$ independently at random in S. The probability we have set p_i to 0 is at most $(d - i)/|S|$ by induction.
- Assuming this does not happen, we have a degree i polynomial in one variable x_n.
- Pick x_n at random, the chance we get 0 is at most $i/|S|$.
- Overall, failure probability at most $(d - i)/|S| + i/|S| = d/|S|$.

Perfect matchings in general graphs

You may have seen in an algorithms class how to efficiently find perfect matchings in bipartite graphs (e.g., using network flow).

You might not have seen how to do it in general graphs, e.g.,



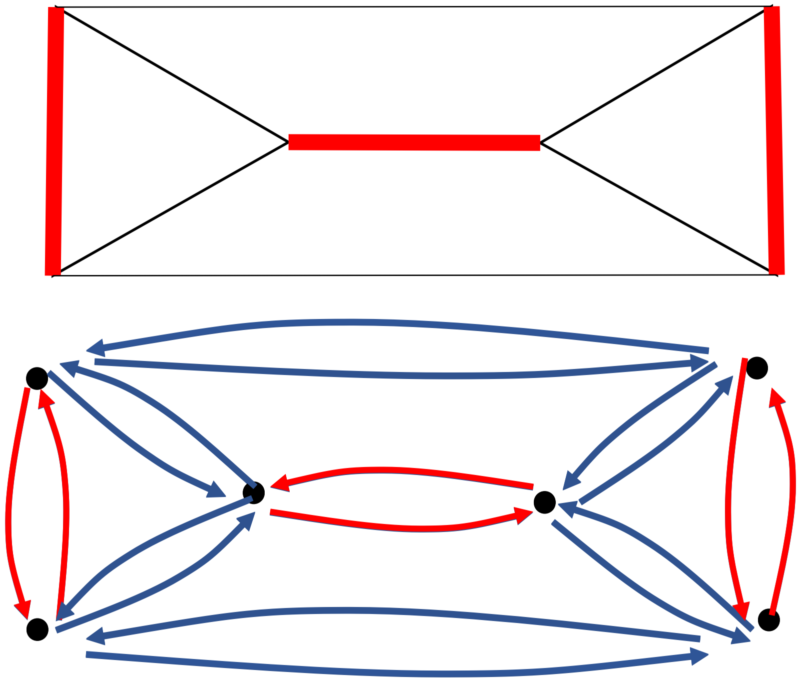
Here is a simple randomized algorithm.

(Efficient deterministic algorithms exist too, but they’re more complicated)

Perfect matchings in general graphs

First, given graph G, think of it as a directed graph G' where each undirected edge in G is replaced by two directed edges, one in each direction.

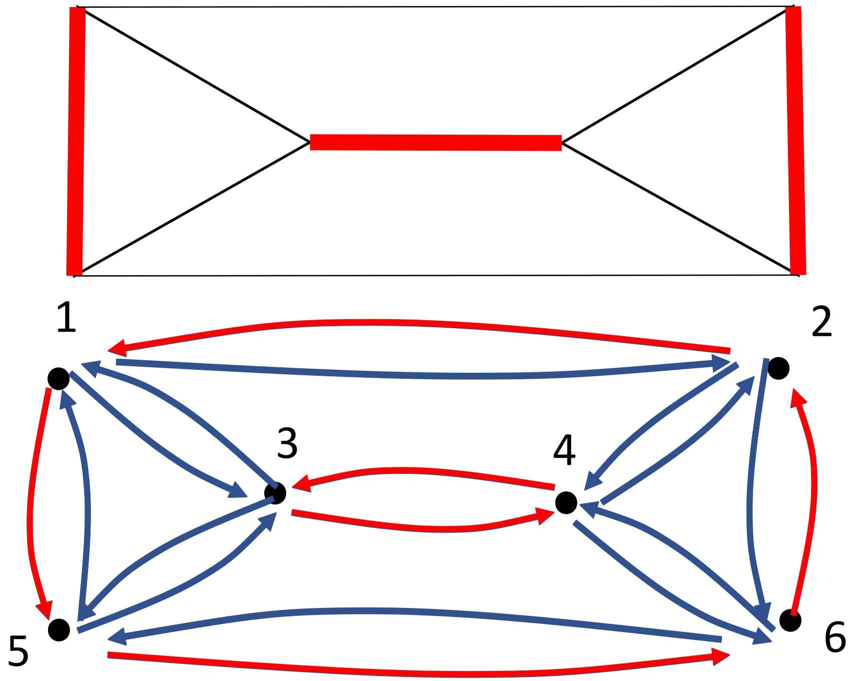
G has a perfect matching iff G' has a cycle cover (a collection of disjoint cycles that cover all vertices) where all cycles have even length.



Perfect matchings in general graphs

First, given graph G, think of it as a directed graph G’ where each undirected edge in G is replaced by two directed edges, one in each direction.

G has a perfect matching iff G’ has a cycle cover (a collection of disjoint cycles that cover all vertices) where all cycles have even length.



Note: a cycle cover is equivalent to a permutation of the vertices that is consistent with the edges.

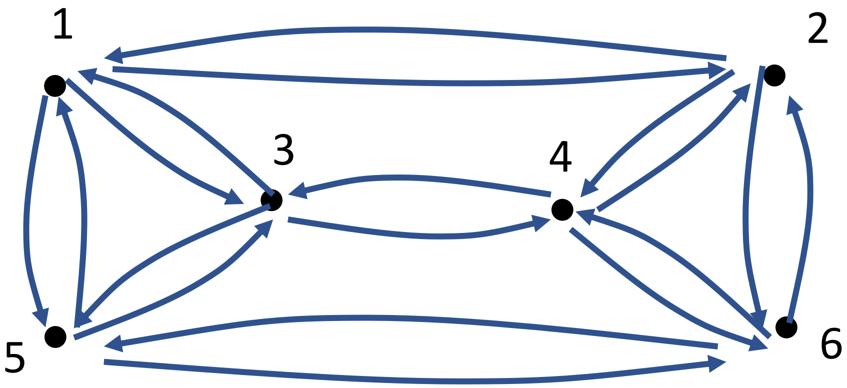
Perfect matchings in general graphs

Now, given G, create “Tutte matrix” M. For every edge $(i, j), i < j$, put variable x_{ij} in entry ij (above the diagonal) and $-x_{ij}$ in entry ji (below the diagonal). The rest are 0.

0	x_{12}	x_{13}	0	x_{15}	0
$-x_{12}$	0	0	x_{24}	0	x_{26}
$-x_{13}$	0	0	x_{34}	x_{35}	x_{36}
0	$-x_{24}$	$-x_{34}$	0	0	x_{46}
$-x_{15}$	0	$-x_{35}$	0	0	x_{56}
0	$-x_{26}$	0	$-x_{46}$	$-x_{56}$	0

Now, consider $\det(M)$. This is a polynomial of degree $\leq n$ in at most n^2 variables, and it can be efficiently computed on any given assignment.

Claim: this polynomial is identically 0 iff G has no perfect matching.



So, can use DLSZ lemma to solve the decision question, and then do an easy self-reduction.

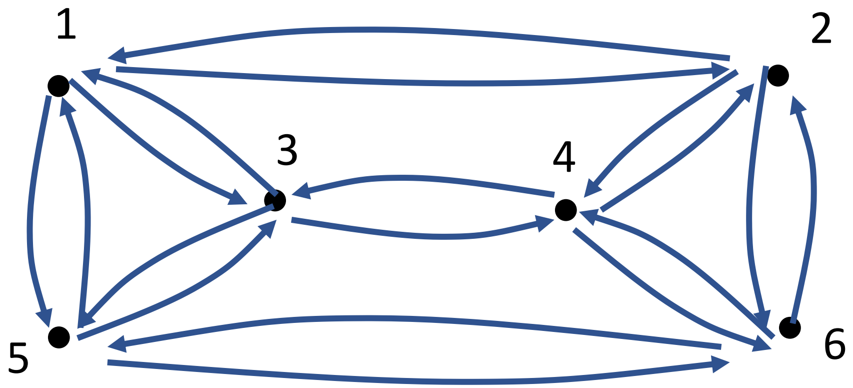
Perfect matchings in general graphs

Now, given G, create “Tutte matrix” M. For every edge $(i,j), i<j$, put variable x_{ij} in entry ij (above the diagonal) and $-x_{ij}$ in entry ji (below the diagonal). The rest are 0.

0	x_{12}	x_{13}	0	x_{15}	0
$-x_{12}$	0	0	x_{24}	0	x_{26}
$-x_{13}$	0	0	x_{34}	x_{35}	x_{36}
0	$-x_{24}$	$-x_{34}$	0	0	x_{46}
$-x_{15}$	0	$-x_{35}$	0	0	x_{56}
0	$-x_{26}$	0	$-x_{46}$	$-x_{56}$	0

Claim: $\det(M)$ is identically 0 iff G has no perfect matching.

- If we write $\det(M)$ as a sum of terms, each term is a permutation consistent with the edges, i.e., a cycle cover of G’.
- Two cycle covers of G’ will give $\pm$ the same term iff they have the same edges of G (possibly used in different directions)



- If you sum over all cycle covers giving $\pm$ the same term, you will get 0 iff it includes an odd cycle.

Perfect matchings in general graphs

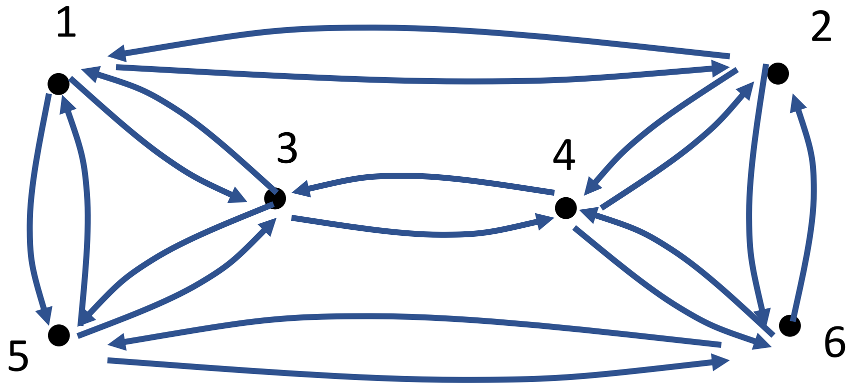
Now, given G, create "Tutte matrix" M. For every edge $(i, j), i < j$, put variable x_{ij} in entry ij (above the diagonal) and $-x_{ij}$ in entry ji (below the diagonal). The rest are 0.

0	x_{12}	x_{13}	0	x_{15}	0
$-x_{12}$	0	0	x_{24}	0	x_{26}
$-x_{13}$	0	0	x_{34}	x_{35}	x_{36}
0	$-x_{24}$	$-x_{34}$	0	0	x_{46}
$-x_{15}$	0	$-x_{35}$	0	0	x_{56}
0	$-x_{26}$	0	$-x_{46}$	$-x_{56}$	0

Claim: $\det(M)$ is identically 0 iff G has no perfect matching.

- This because a cycle of length k will have some j negative entries and $k - j$ positive entries, and its reverse will have $k - j$ negative entries and j positive entries. These have opposite sign iff k is odd.

- Also using fact that sign of permutation doesn't change when you reverse a cycle
- Also using assumption that $2^{\#(cycles\ of\ length > 2)} \neq 0$ in $\mathbb{F}$.



- If you sum over all cycle covers giving $\pm$ the same term, you will get 0 iff it includes an odd cycle.