Circuits and Derandomization.

CS254

Chris Pollett

Nov. 20, 2006.

Outline

- Polynomial size circuits
- Derandomization

Polynomial Size Circuits

- We have already defined what a Boolean circuit is.
- The *size* of a circuit is the number of gates in it.
- We next would like to define what it means for a family of circuits to recognize a language.

Defn. A family of circuits is an infinite sequence $(C_0, C_1, \ldots)$ of Boolean circuits, where C_n has n input variables. We say a language L has polynomial size circuits, if there is a polynomial p such that $\text{size}(C_n) \leq p(n)$ and C_n accepts exactly those strings in L of length n.

P is in P/Poly

- We call the class of languages with polynomial circuits P/poly.

Thm. All languages in P have polynomial size circuits.

Proof. This essentially follows from our proof that CVP is P-complete -- however, rather than encode a particular x into the inputs we instead let its value come from variables.

Uniformity

Defn. We call a circuit family $(C_0, C_1, \ldots)$ ***uniform*** if there is a log n-space machine N which on input 1^n outputs C_n. We say that a language L has ***uniformly polynomial circuits*** if there is a uniform family of p-size circuits that decides L.

Thm. A language L has uniformly polynomial circuits iff L is in P.

Proof. One direction follows from the theorem of the last slide recall completeness of CVP was logspace computable. For the other direction suppose that L has uniformly polynomial circuits. In p-time we can decide x in L by first running the logspace machine to get $C_{|x|}$ then doing circuit evaluation on x in p-time.

Advice Classes

- An **advice string** is a map from positive integers to strings.
- We say a machine M **decides a language L with advice string** A(n) if x in L implies M(x, A(|x|)) output yes. And if x is not in L then M(x, A(|x|)) outputs "no".
- Let **poly** denote the set of advice strings A(n) such that $|A(n)| \leq p(n)$ for some polynomial n.
- We say a language L is in **P/poly** if there is a a p-time M that decides L using an advice string in poly.

Prop. This and our previous definition of P/poly are equivalent.

Some Conjectures

- **Conjecture A:** NP-complete problems have no uniformly polynomial circuits.
- This can be viewed as a restatement of P≠NP.
- **Conjecture B:** NP-complete problems have no polynomial circuits, uniform or not.
- So if Conjecture B is true, proving circuit lower bounds for problems in NP might be an approach to the P versus NP problem.
- The next result show that circuit lower bounds are useless in proving P≠BPP. It also gives our first derandomization result.

BPP $\subseteq$P/Poly

Theorem. BPP $\subseteq$ P/poly

Proof. Let L be in BPP decided by NTM N with a clear majority. We claim that L has a p-size circuit family (C_0 , C_1, …C_n).

C_n is based on a sequence of bit strings A_n=(a_1,…,a_m) where each a_i has length p(n), and where m=12(n+1). Each bit string represents a string of nondeterministic choice that N might have used. The idea is that C_n will simulate N on each of these 12(n+1) many paths and take the majority outcome. Since given the path we can use the tableau method to simulate N on inputs of length n, C_n will be poly-size in n. So it suffices to prove that there exists an A_n which has the desired properties…

Proof Cont'd

Call a_i *bad* if it leads C_n to a false positive or a false negative answer.

Claim. For all n>0 there is a set A_n of 12(n+1) bit strings such that for all x with |x|=n fewer than half of the choices in A_n are bad.

Proof. Consider a sequence A_n of bit strings of length p(n) obtained by m independent random samples. *What is the probability that for each x in $\{0,1\}^n$ more than half the choices are correct?*

Proof cont'd some more

- For each x of length n at most 1/4 of the computations are bad. So we expect at most (1/4)*m many bad ones in A_n. By Chernoff bounds the probability that the number of bad bit strings is (1/2)*m or more is at most $e^{-m/12} < 1/2^{n+1}$.
- This holds for each x of length n. Thus the probability that there is an x with no accepting sequence in A_n is at most the sum of the probabilities among all x of length n; and this gives $2^{n} * 1/2^{n+1} = 1/2$. So with probability at least 1/2 our random selection has the desired property.