Completeness

CS254

Chris Pollett

Oct 23, 2006.

Outline

- Polynomially Verifiable
- Complete problems for P and NP

Polynomially Verifiable Languages

- NP is sometimes called the class of languages which are polynomial time verifiable.
- Call a relation $R \subseteq \Sigma^* x \Sigma^*$ *polynomial decidable* if there a DTM which decides the language {<x,y> | (x,y) is in R}. We say R is *polynomially balanced* if (x,y) is in R implies $|y| \leq |x|^k$ for some $k \geq 1$.
- The next proposition shows what polynomial time verifiable means

Prop. Let L be a language. L is in NP iff there is a polynomially decidable and polynomially balanced (by $|x|^k$ for some k) relation R, such that

$L=\{x| \exists y, y \leq |x|^k \text{ and } (x,y) \text{ is in } R\}$

- So given x, if we had in $|x|^k$ proof string y we could verify in polynomial time whether x was in L.

Proof. Any L of the form $\{x| \exists y, y \leq |x|^k$ and (x,y) is in R$\}$ can be decided in NP by a machine which first nondeterministically guesses y and then runs R on (x,y). On the other hand, if L is NP via M, some NDTM, then we can let R be the p-time DTM which acts like M except when M needs to do its ith nondeterministic move, R instead consults the ith square of y and uses this value to say which possible next transition to follow.

Variations on SAT

- k-SAT is the variant of SAT where each clause has at most k literal.

Prop. 3SAT is NP-complete.

Proof. Notice our reduction of CIRCUIT-SAT to SAT is actually a reduction to 3SAT.

Prop. 3SAT remains NP-complete for expressions in which each variable appears at most three times and each literal at most twice.

Proof. Suppose a variable *x* appears *k* times in a 3SAT instance. We would replace this variable with k variables $x_1,\ldots,x_k$ and add the clause: $(\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \ldots \wedge (\neg x_k \vee x_1)$

2SAT is in P

Given a 2SAT instance I we can build a graph G(I) as follows:

- the vertices of V are the variables of I and there negations.
- there is an edge (a,b) in the graph iff there is a clause (¬a V b) in I. These edges can be viewed as capturing logical implication

Thm One can show I is unsatisfiable iff there is a variable x such that there are paths from from x to ¬x and from ¬x to x in G(I).

Proof. Suppose such a path exists then assigning x true and following the path of implications gives true=>false. Similarly, if one assigned x false.

On the other hand if there is no such path, we could pick a node a that has not been assigned and such that there is no path from a to ¬a, and assign it true. We also assign true all nodes reachable from a and assign false the negations of these nodes. Then we repeat.

This proves 2SAT is in P since reachability is in P-time.

2SAT is in NL

Recall NL is closed under complement. So it suffices to recognize unsatisfiable expression in NL. In NL,we guess a variable x and a sequence of successive pairs of vertices along a path from x to $\neg x$ and back.

MAX2SAT is NP-complete

- MAX2SAT is the problem give a 2SAT instance I, and an integer k: Is there an assignment which makes at least k clauses true?

Thm. MAX2SAT is NP-complete.

Proof. Consider the ten clauses:

$(x)(y)(z)(w)$

$(\neg x \vee \neg y)\ (\neg y \vee \neg z)\ (\neg z \vee \neg x)$

$(x \vee \neg w)\ (y \vee \neg w)\ (z \vee \neg w)$

There is no way to satisfy all these clauses. Notice if a truth assignment satisfies $(x \vee y \vee z)$ then we can satisfy 7 of these clauses. For all other truth assignments we can satisfy at most 6. So we can use this to reduce a 3SAT instance of m clauses to a MAX2SAT instance with k=7m.