COMS W3261
Computer Science Theory
Lecture 19
Satisfiability

Alexander Roth

2014 – 11 – 12

Outline

- Boolean expressions
- The satisfiability problem
- Normal forms for boolean expressions
- The problems CSAT and kSAT
- SAT is NP-complete: The Cook-Levin theorem

1 Boolean Expressions

- Boolean expressions are generated by the following CFG:

$$E \to E \vee T \mid T$$
$$T \to T \wedge F \mid F$$
$$F \to (E) \mid \neg \mid var$$

  var represents a variable whose value can be either 1 (for true) or 0 (for false). The operator $\wedge$ stands for logical AND, $\vee$ for logical OR, and $\neg$ for logical NOT. Note that the grammar gives the operator $\neg$ the highest precedence, then $\wedge$, then $\vee$.

Class Notes

Normal Forms for BE's

- Conjunction: AND $x \wedge y$ xy

- Disjunction: OR $x \vee y$ $x + y$
- Negation: NOT $\neg x$ $\overline{x}$
- Literal: variable or negated variable x, $\overline{x}$
- Clause: logical OR of one or more literals.

A BE is in Conjunctive Normal Form (CNF) if it is the logical AND of clauses, e.g., $(x + \overline{y})(\overline{x} + y) = (x \vee \neg y) \wedge (\neg x \vee y)$

2 The Satisfiability Problem

- A truth assignment for a boolean expression assigns either the value true (1) or the value false(0) to each of the variables in the expression.
- The value $E(T)$ of an expression E given a truth assignment T is the result of evaluating E with each variable x in E replaced by $T(x)$.
- A truth assignment T satisfies E if $E(T) = 1$.
- An expression E is *satisfiable* if there exists a truth assignment T that satisfies E.
- The *satisfiability problem* (SAT) is to determine whether a given boolean expression is satisfiable.
  - The value of $E = x \wedge \neg(y \vee z)$ given the truth assignment $T(x) = 1$, $T(y) = 0$, $T(z) = 0$ is 1. $[1 \wedge \neq (0 \vee 0) = 1]$. Thus, E is satisfiable.
  - The expression $E = x \wedge (\neq x \vee y) \wedge \neg y$ is not satisfiable because none of the four truth assignments to the variables x and y causes E to have the value 1.
- We shall shortly prove that SAT is NP-complete.

Class Notes

SAT Given a BE E, is E satisfiable?

Theorem 1. (Cook-Levin Theorem) SAT is NP-complete

1. SAT is in NP.
2. Every problem in NP can be reduced in polynomial time to SAT.

3 Normal Forms for Boolean Expressions

- In boolean expressions
  - Logical AND, as in $x \wedge y$, is often called conjunction and is sometimes written as a product, as in xy.
  - Logical OR, as in $x \vee y$, is often called disjunction and is sometimes written as a sum, as in $x + y$.
  - Logical NOT, as in $\neg x$, is often called negation and is sometimes written with an overbar, as in $\overline{x}$.
  - A literal is a variable or a negated variable; e.g., x and $\neg x$ are both literals.
  - A clause is the logical OR (disjunction) of one or more literals; e.g., $x \vee \neg y$ is a clause.
- A boolean expression is in conjunctive normal form (CNF) if it is the local AND (conjunction) of clauses; e.g., $(x \vee \neg y) \wedge (\neg x \vee z)$ is in CNF.
- A boolean expression is in k-CNF if it is the logical AND of clauses each one of which is the logical OR of exactly k distinct literals; e.g., $(w \vee \neg x \vee y) \wedge (x \vee \neg y \vee z)$ is in 3-CNF.
- Two boolean expressions are equivalent if they have the same result on any truth assignment to their variables.

4 The Problems CSAT and kSAT

- CSAT
  - Given a boolean expression E in k-CNF, is E satisfiable?
  - We can view CNF as the language $\{E \mid E$ is the representation of a satisfiable CNF boolean expressions $\}$.
  - CST is NP-complete.
- kSAT
  - Given a boolean expression E in k-CNF, is E satisfiable?
  - 1SAT and 2SAT are in P; kSAT is NP=complete for $k \geq 3$.

5 SAT is NP-complete: the Cook-Levin theorem

- SAT is in NP

- Given a boolean expression E of length n, a multitape nondeterministic Turing machine can guess a truth assignment T for E in $O(n)$ time.
- The NTM can then evaluate E using the truth assignment T in $O(n^2)$ time.
- If $E(T) = 1$, then the NTM accepts E.
- The NTM can be simulated by a single-tape deterministic TM in $O(n^4)$ time.

- If L is in NP, then there is a polynomial-time reduction of L to SAT.
- If a NTM M accepts an input w of length n in $p(n)$ time, then M has a sequence of moves such that

  1. α_0 is the initial ID of M with input w.
  2. $\alpha_0 \vdash \alpha_1 \vdash \ldots \vdash \alpha_k$ where $k \leq p(n)$.
  3. α_k is an ID with an accepting state.
  4. Each α_i consists only of nonblanks unless α_i ends in a state and a blank, and extends from the initial head position to the right.

- From M and w, we can construct a boolean expression $E_{M,w}$ that is satisfiable iff M accepts w within $p(n)$ moves. See HMU, pp. 400–446 for details.

Class Notes

$$E_{M,w} = U \wedge S \wedge S \wedge N \wedge F$$

where

- U is the unique symbol in each cell
- S starts off right
- N next move is right
- F finishes right.

Class Notes

Some Examples of NP-Complete Problems

- SAT
- CSAT