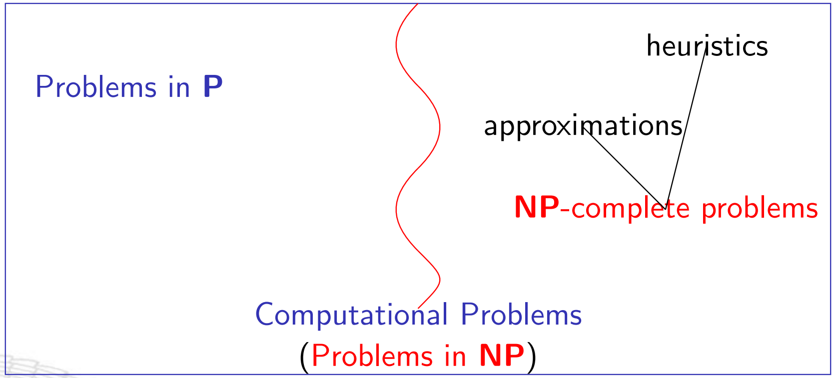
Algorithm Theory

Georg Moser Mircea Dan Hernest

Institute of Computer Science @ UIBK

Summer 2007

The two faces of complexity theory



➡ "There is nothing wrong with trying to prove **P** = **NP** by developing a **P**-algorithm for an **NP**-complete problem. However, without **NP**-completeness proof we wouldn't know"

Theorem
CIRCUIT SAT reduces to SAT

Proof
- ➡ given a circuit C
- ➡ define a Boolean expression $R(C)$ with C is satisfiable iff $R(C)$ is satisfiable

Definition construction

- ➡ variables of $R(C)$ contain all variables in C + for any gate g a variable g
- ➡ g a variable gate, corresponding to x add CNF of $(g \leftrightarrow x)$
- ➡ g a **true** gate add clause (g)
- ➡ g **false** gate add clause $(\neg g)$
- ➡ g a not gate add CNF of $(g \leftrightarrow \neg h)$

here h is the predecessor of g in C

- ➡ g an or gate add CNF of $(g \leftrightarrow (h \vee h'))$
- ➡ g an and gate add CNF of $(g \leftrightarrow (h \wedge h'))$
- ➡ g the output gate add (g)

here h, h' are the predecessors of g in C □

Definition 3SAT
- ➡ a 3CNF-formula is a CNF, where each clause has exactly 3 literals
- ➡ 3SAT $= \{\varphi \colon \varphi$ is a satisfiable 3CNF-formula$\}$

Theorem
3SAT is **NP**-complete

Proof
- ➡ reduction from CIRCUIT SAT produces clauses D with $|D| \leqslant 3$
- ➡ duplicate literals if necessary □

Variants of 3SAT

Definition $3\mathrm{SAT}_1$

$$3\mathrm{SAT}_1 = \left\{\varphi \mid \begin{array}{l}\varphi \text{ is a satisfiable 3CNF-formula and each clause contains} \\ \text{distinct variables}\end{array}\right\}$$

Theorem
$3\mathrm{SAT}_1$ is **NP**-complete

Definition $3\mathrm{SAT}_2$

$$3\mathrm{SAT}_2 = \left\{\varphi \mid \begin{array}{l}\varphi \text{ is a satisfiable CNF-formula, for each clause } C, |C| \leqslant 3 \\ \text{and each variable occurs at most 3 times and each literal} \\ \text{at most twice in the formula}\end{array}\right\}$$

Theorem
$3\mathrm{SAT}_2$ is **NP**-complete

Proof Idea
rewrite instances of 3SAT such that the restriction is fulfilled

Proof

- ➡ given $\varphi \in 3\mathrm{CNF}$

suppose the variable x occurs $k > 3$ times in φ

Definition construction

1. introduce k new variables $x_1, \dots, x_k$
2. replace distinct occurrences by distinct variables
3. add $(\neg x_1 \lor x_2) \land (\neg x_2 \lor x_3) \land \cdots \land (\neg x_k \lor x_1)$
4. number of occurrences of other variables is unchanged
5. number of occurrences of literals in the new clauses is $\leqslant 2$

repeat for all variables violating the restriction

□

Definition 2SAT

$$2\mathrm{SAT} = \{\varphi \colon \varphi \text{ is a satisfiable 2CNF-formula}\}$$

Theorem
2SAT is in **P**, more precisely in **NL**.

Definition $G(\varphi)$
φ be a 2CNF formula
define a graph $G(\varphi)$:

1. nodes of $G(\varphi)$ are the variables (and their negations) of φ
2. for any pair $(\alpha, \beta) \in G$:
   if exists clause $\neg\alpha \lor \beta$ in φ, add (α, β) to the set of edges

Fact
If (α, β) is an edge, then so is $(\neg\beta, \neg\alpha)$

Theorem
φ is unsatisfiable iff $\exists$ variable x such that $\exists$ paths from x to $\neg x$ and from $\neg x$ to x in $G(\varphi)$

Proof
if

- ➡ suppose there exists a path from x to $\neg x$ and from $\neg x$ to x; further there is a satisfying assignment T
- ➡ we assume $T(x) = \textbf{true}$
- ➡ (i) $T(x) = \textbf{true}$ (ii) $T(\neg x) = \textbf{false}$ (iii) $\exists$ path between x and $\neg x$
- ➡ $\exists$ an edge $(\alpha, \beta) \in G(\varphi)$: $T(\alpha) = \textbf{true}$, $T(\beta) = \textbf{false}$
- ➡ $\exists$ clause in φ: $\neg\alpha \lor \beta$ and $T(\neg\alpha \lor \beta) = \textbf{false}$
- ➡ contradiction

only-if

- ➡ suppose no such paths exists; we define assignment T

Definition

1. pick a node α, not yet assigned a truth value
2. such that there is no path from α to $\neg\alpha$
3. set α and all successors to **true**
4. set $\neg\alpha$ and all predecessors to **false**

- ➡ this is well-defined
- ➡ only β or $\neg\beta$ can be successor of α
- ➡ we cannot change existing truth-values

□

Theorem

2SAT is in **NL** $\subseteq$ **P**

Proof

NL is closed under complement, it suffices to recognise unsatisfiable expressions

Definition algorithm

1. guess a variable x
2. check the existence of a path from x to $\neg x$ (and back)
3. instance of REACHABILITY which is decidable by a NTM operating in log-space

□

Definition MAX2SAT

$$\text{MAX2SAT} = \left\{\varphi\colon \begin{array}{l}\varphi \text{ is a 2CNF-formula and at least } K \text{ clauses are}\\ \text{satisfiable.}\end{array}\right\}$$

Theorem

MAX2SAT is **NP**-complete

$$\begin{array}{ccc} (x) & (y) \quad (z) & (w) \\ (\neg x \vee \neg y) & (\neg y \vee \neg z) & (\neg z \vee \neg x) \\ (x \vee \neg w) & (y \vee \neg w) & (z \vee \neg w) \end{array}$$

if $T(x \vee y \vee z) = $ **true**
7 clauses can be satisfied; otherwise only 6

Proof

block

$$\begin{array}{ccc} (x) & (y) \quad (z) & (w) \\ (\neg x \vee \neg y) & (\neg y \vee \neg z) & (\neg z \vee \neg x) \\ (x \vee \neg w) & (y \vee \neg w) & (z \vee \neg w) \end{array}$$

MAX2SAT $\in$ **NP**

easy

reduction from 3SAT

- ➡ let $\varphi \in$ 3CNF
- ➡ let m denote the number of clauses in φ

Definition construction

1. for any clause $C_i = (\alpha \vee \beta \vee \gamma)$ in φ
   add the block with w_i fresh to $R(\varphi)$
2. set $K = 7m$

$\varphi \in$ 3SAT iff $R(\varphi) \in$ MAX2SAT

- suppose φ is satisfied

  for each block 7 clauses are satisfied

  thus $R(\varphi) \in$ MAX2SAT

- suppose in $R(\varphi)$ $7m$ clauses are satisfied

  hence in each block at most 7 clauses can be satisfied

  hence in any block $\alpha \vee \beta \vee \gamma$ is satisfied

  hence any clause in φ is satisfied

logspace reduction

easy to check that computation of R needs only log-space

$\square$

this form of approach to a reduction is called gadget construction;
the blocks form the gadget