Complexity Theory

Jan Křetínský

Chair for Foundations of Software Reliability and Theoretical Computer Science
Technical University of Munich

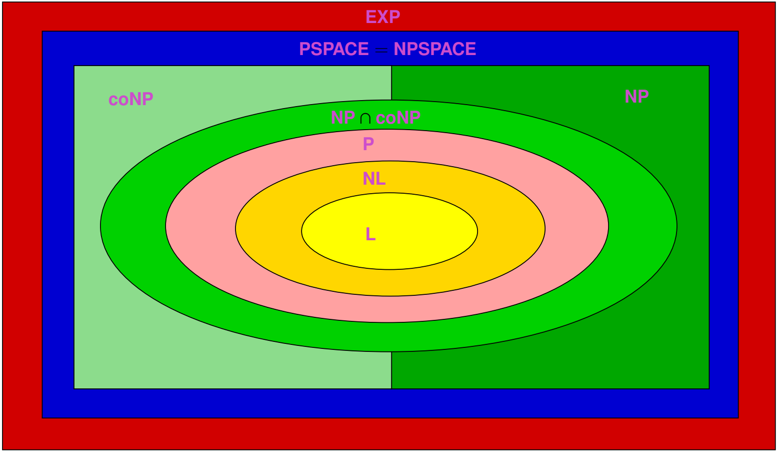
Summer 2016

Based on slides by Jörg Kreiker

Lecture 4

NP-completeness

Recap: relations between classes



Agenda

- efficiently checkable certificates
- reductions, hardness, completeness
- Cook-Levin: 3SAT is **NP**-complete

NP: efficiently checkable certificates

NP computable with NDTM in polynomial time.

NP: efficiently checkable certificates

NP computable with NDTM in polynomial time.

Theorem (Certificates)

*For every $L \subseteq \{0,1\}^*$ holds: $L \in$ **NP** if and only if there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a polynomial-time TM M such that for every $x \in \{0,1\}^*$*

$$x \in L \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)}.\ M(x,u) = 1$$

NP: efficiently checkable certificates

NP computable with NDTM in polynomial time.

Theorem (Certificates)

*For every $L \subseteq \{0,1\}^*$ holds: $L \in$ **NP** if and only if there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a polynomial-time TM M such that for every $x \in \{0,1\}^*$*

$$x \in L \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)}.\ M(x,u) = 1$$

- M is called verifier
- u is called certificate

NP: efficiently checkable certificates

NP computable with NDTM in polynomial time.

Theorem (Certificates)

*For every $L \subseteq \{0,1\}^*$ holds: $L \in$ **NP** if and only if there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a polynomial-time TM M such that for every $x \in \{0,1\}^*$*

$$x \in L \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)}.\ M(x,u) = 1$$

- M is called verifier
- u is called certificate

Proof:

$\Rightarrow$ certificate is sequence of choices

$\Leftarrow$ NDTM guesses certificate

Examples

- Indset: certificate is set of nodes, size of certificate for k nodes in graph with n nodes $\mathcal{O}(k \log n)$
- $0/1-$ILP: given a list of m linear inequalities with rational coefficients over variables $x_1, \ldots, x_k$; find out if there is an assignment of 0s and 1s to x_i satisfying all inequalities; certificate is assignment.
- Iso: given two $n \times n$ adjacency matrices; do they define isomorphic graphs; certificate is a permutation $\pi : [n] \to [n]$

Agenda

- efficiently checkable certificates ✓
- reductions, hardness, completeness
- Cook-Levin: 3SAT is **NP**-complete

Reductions – reminder

IF
- there is an efficient procedure for problem A and
- and an efficient procedure for B using the procedure for A

THEN B cannot be radically harder than A

notation: $B \leq A$

We have seen (at least) two reductions.

- 3−Coloring was reduced to Indset
- the diagonalized, undecidable language reduced to Halt
- reduction does not make anything *smaller*

Reductions – definition

Definition (Karp reduction)

Let $L, L' \subseteq \{0,1\}^*$ be languages. L is polynomial-time Karp reducible to L' iff there exists a polynomial-time computable funtion $f : \{0,1\}^* \to \{0,1\}^*$ such that for all $x \in \{0,1\}^*$

$$x \in L \Leftrightarrow f(x) \in L'$$

We write $L \leq_p L'$.

Reductions – definition

Definition (Karp reduction)

Let $L, L' \subseteq \{0,1\}^*$ be languages. L is polynomial-time Karp reducible to L' iff there exists a polynomial-time computable funtion $f : \{0,1\}^* \to \{0,1\}^*$ such that for all $x \in \{0,1\}^*$

$$x \in L \Leftrightarrow f(x) \in L'$$

We write $L \leq_p L'$.

Note: $\leq_p$ is a transitive relation on languages (because the composition of polynomials is a polynomial).

Hardness and Completeness

Definition (NP-hardness and -completness)

Let $L \subseteq \{0,1\}^*$ be a language.

- L is **NP**-hard if $L' \leq_p L$ for every $L' \in$ **NP**
- L is **NP**-complete if L is **NP**-hard and $L \in$ **NP**.

Hardness and Completeness

Definition (NP-hardness and -completness)

Let $L \subseteq \{0,1\}^*$ be a language.

- L is **NP**-hard if $L' \leq_p L$ for every $L' \in$ **NP**
- L is **NP**-complete if L is **NP**-hard and $L \in$ **NP**.

Examples of **NP**-hard languages: Indset, Halt_k, Halt

Hardness and Completeness

Definition (NP-hardness and -completness)

Let $L \subseteq \{0,1\}^*$ be a language.

- L is **NP**-hard if $L' \leq_p L$ for every $L' \in$ **NP**
- L is **NP**-complete if L is **NP**-hard and $L \in$ **NP**.

Examples of **NP**-hard languages: Indset, Halt_k, Halt

Observation

- L **NP**-hard and $L \in$ **P** implies **P** $=$ **NP**
- L **NP**-complete implies $L \in$ **P** iff **P** $=$ **NP**

Do NP-complete languages exist?

- upcoming result independently discovered by Cook (1971) and Levin (1973)
- uses notion of satisfiable Boolean formulas
- Boolean formula φ over variables $X = \{x_1, \ldots, x_k\}$ defined by

$$\varphi ::= x \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

- write $\overline{x}$ instead of $\neg x$, x and $\overline{x}$ literals u
- assume formulas are in CNF:

$$\varphi = \bigwedge_i \bigvee_j u_{i_j}$$

- disjunctions $\bigvee_j u_{i_j}$ called clauses
- formula is in k-CNF if the no clause has more than k literals

Cook-Levin Theorem

- φ is satisfiable iff there exists an assignments $a : X \to \{0, 1\}$ making φ true
- 3SAT $= \{\varphi \mid \varphi$ in 3-CNF and satisfiable$\}$

Theorem

3SAT *is* **NP***-complete.*

Proof agenda

1. SAT is **NP**-complete (without restriction to clauses of size three)
   1.1 SAT, 3SAT $\in$ **NP**
   1.2 for every $L \in$ **NP** $L \leq_p$ SAT
2. Show that SAT $\leq_p$ 3SAT

What have we learnt?

- **NP** is polynomial certificates
- Karp reductions, hardness, completeness
- Cook-Levin: reduce any language in **NP** to 3SAT
- up next: more **NP**-complete problems, **P** vs. **NP**, tool demos