Solution

Computational Complexity – Homework 6

Discussed on 30.05.2016.

Exercise 6.1

You have seen that 2SAT is in NL. Show that 2SAT is also NL-hard.

**Solution:** Since \textit{REACHABILITY} is NL-hard and we know that NL is closed under complement, it suffices to show that there exists a logspace reduction from \textit{REACHABILITY} to 2SAT. Suppose that we are given a graph \( G = (V, E) \), an initial vertex \( v_0 \) and a target vertex \( v_f \). From this we assign a variable \( x_v \) to each node in \( V \) and then construct \( \phi_G := \bigwedge_{(v_1, v_2) \in E} (x_{v_1} \rightarrow x_{v_2}) \) (where \( x_{v_1} \rightarrow x_{v_2} \) is \( \neg x_{v_1} \vee x_{v_2} \)). Finally we take the result of the reduction to be \( \psi_G := x_{v_0} \wedge x_{v_f} \wedge \phi_G \).

\( \psi_G \) is a 2SAT instance and can be constructed in logspace (in the size of the reachability problem instance). Indeed the construction can be carried out in constant space: we can reuse the node IDs as variable IDs and in particular \( \phi_G \) is just a rewriting of \( E \) (copying node IDs from a pairs \((v_1, v_2)\) and adding the appropriate Boolean operators.

It just remains to check that \( v_f \) is NOT reachable from \( v_0 \) iff \( \psi_G \) is SAT. For this it suffices to show that (i) if a valuation satisfies \( x_{v_0} \wedge \phi_G \) it must set \( x_v \) to true for all \( v \) reachable from \( v_0 \), and (ii) if a node \( v \) is unreachable from \( v_0 \), then there exists a valuation satisfying \( x_{v_0} \wedge \phi_G \) that sets \( x_v \) to false for every unreachable node \( v \).

To prove (i) argue by induction on the number of steps to reach \( v \) from \( v_0 \). To prove (ii) take the valuation that sets \( x_v \) to true if \( v \) is reachable and false otherwise. Assume for contradiction that this is not a satisfying valuation. Since \( v_0 \) is trivially reachable it follows that there is a clause \( x_{v_1} \rightarrow x_{v_2} \) in \( \phi_G \) such that \( x_{v_1} \) is set to true but \( x_{v_2} \) is set to false. But if this clause exists, \( (v_1, v_2) \in E \) and by the definition of valuation \( v_1 \) is reachable whilst \( v_2 \) is not, which is a contradiction.

Exercise 6.2

Show that deciding the inequivalence of context-free grammars over one-letter terminal alphabet is \( \Sigma^p_2 \)-hard. You can make use of \( \Sigma^p_2 \)-hardness of integer expression inequivalence.

What does it imply for the equivalence problem?

Exercise 6.3

Under the assumption that 3SAT \( \leq_p \overline{3\text{SAT}} \) show that \( \text{NP} = \text{PH} \).
Solution: If $3\text{Sat} \leq_p \overline{3\text{Sat}}$, then $\mathbf{NP} = \mathbf{coNP}$, i.e., $\Sigma^p_1 = \Pi^p_1$. Consider now any $L \in \Sigma^p_2$. We have

$$x \in L \text{ iff } \exists u \in \{0,1\}^{p(|x|)} \forall v \in \{0,1\}^{q(|x|)} : M(x,u,v) = 1.$$ 

The language

$$L_1 \{ (x,u) \mid \forall v : M(x,u,v) = 1 \}$$

is then in $\mathbf{coNP}$ and, thus, in $\mathbf{NP}$, i.e., we find a TM $M'$ and a polynomial $r$, s.t.,

$$(x,u) \in L_1 \text{ iff } \exists v \in \{0,1\}^{r(|x|)+|u|} : M'(x,u,v) = 1.$$ 

As $|u| = p(|x|)$, we may assume that $|v| = r(|x|)$ by adjusting $r$. Hence,

$$x \in L \text{ iff } \exists uv \in \{0,1\}^{p(|x|)+r(|x|)} : M'(x,uv) = 1,$$

i.e., $L \in \mathbf{NP}$.

So, $\Sigma^p_2 \subseteq \mathbf{NP} = \mathbf{coNP}$. Similarly, $\Pi^p_2 \subseteq \mathbf{NP} = \mathbf{coNP}$.

Using induction, one now shows that $\mathbf{NP} = \mathbf{PH}$.

**Exercise 6.4**

Apart from the certificate definition and the alternative bounded alternating Turing machine characterization, there is one more standard characterization of the polynomial hierarchy via oracles.

For a language $L$, an oracle machine $M^L$ is a Turing machine which can moreover do the following kind of computation steps. It can write down a word $w$ on a special tape and ask whether $w \in L$ and it immediately receives the correct answer. One can also talk about this machine even when the oracle is not specified, then we write $M^\varnothing$.

**Example:** In Exercise 3.4 (a), you have constructed an example of $M^{\text{SAT}}$ where $M^\varnothing$ is a polynomial time TM.

- Prove or disprove: for every $M^\varnothing$, if $A \subseteq B$ then $\mathcal{L}(M^A) \subseteq \mathcal{L}(M^B)$.
- Prove or disprove: if $A \subseteq B$ then $\mathbf{P}^A \subseteq \mathbf{P}^B$ (as classes).

The polynomial hierarchy can be defined inductively setting $\Sigma^p_0 = \Pi^p_0 = \mathbf{P}$ and

$$\Sigma^p_{i+1} = \mathbf{NP}^{\Sigma^p_i},$$

$$\Pi^p_{i+1} = \mathbf{coNP}^{\Sigma^p_i},$$

where $A^B$ is the set of decision problems solvable by a Turing machine in class $A$ with an oracle for some complete problem in class $B$.

- Show this yields the same hierarchy as the original definition.

One can also define $\Delta^p_{i+1} = \mathbf{P}^{\Sigma^p_i}$ and show that $\Delta^p_{i+1} \subseteq \Sigma^p_{i+1} \cap \Pi^p_{i+1}$ and it contains all languages expressible as Boolean combinations (unions, intersections, complements) of languages of $\Sigma^p_i$ and $\Pi^p_i$.

- What is the relationship of these classes to $\mathbf{DP} = \{ L \mid \exists M,N \in \mathbf{NP} : L = M \setminus N \}$?