Instructions

- Classroom Problems C6.1–C6.3 will be discussed and solved at the tutorial session on Wed 26 Feb, 14–16, Room T4 (A238). No credit is given for these problems.
- Homework Problems H6.1–H6.3 you should solve on your own, and submit your solutions via the MyCourses interface by the deadline of Tue 3 Mar, 23:59, These problems will be individually graded on a scale of 0–2 points per problem.
- In preparing your solutions to the Homework Problems:
  1. Justify your solutions, be precise, and provide sufficient detail so that it is easy to follow your reasoning.
  2. Submit your solutions as an easily readable, single pdf file, which is either typeset or written in full sentences and clean handwriting.
  3. **[Code of Conduct]** You can discuss the problems with your colleagues and the course's teaching staff, but you must write the presentations of your solutions *independently* and *individually*, without any notes from such discussions.

Classroom Problems

C6.1 Let $L_1, L_2 \in \mathbf{NP} \cap \mathbf{coNP}$. Prove that the language

$$L_1 \oplus L_2 = \{x \in \{0,1\}^* : \ x \text{ is in exactly one of } L_1 \text{ and } L_2\}$$

is in $\mathbf{NP} \cap \mathbf{coNP}$.

C6.2 A language A is *Turing-reducible* to language B, denoted

$$A \leq_T B,$$

if there exists an oracle Turing machine $M^?$ such that M^B decides A. This notion can be complexity-bounded by requiring the reduction machine to be e.g. logspace- or polynomial-time bounded, in which case the corresponding reductions are denoted by $A \leq_T^L B$ and $A \leq_T^P B$, respectively.[1]

For the purposes of this exercise, denote the "usual" logspace- and polynomial-time bounded reductions by $A \leq_m^L B$ and $A \leq_m^P$, respectively.[2]

[1] The polynomial-time bounded Turing reduction is also called the "Cook reduction" in the literature. Note that in OTM's the query string is treated as an output string, i.e. its length is not included in the space cost of a computation.

[2] In the literature, these are called "many-one" reductions, hence the subscript m. The polynomial-time bounded many-one reduction $A \leq_m^P B$ is also known as the "Karp reduction".

Prove the following facts:

(i) If $A \leq_m^P B$, then $A \leq_T^P B$.

(ii) If $A \leq_T^P B$ and $B \leq_T^P C$, then $A \leq_T^P C$.

(iii) For any language A, $A \leq_T^P \bar{A}$.

(iv) There exists a language B for which $B \leq_m^P \bar{B}$ does not hold.

C6.3 Prove that if $\mathbf{P} = \mathbf{NP}$, then $\mathbf{P} = \mathbf{PH}$.

(*Hint:* Start by proving that if $\mathbf{P} = \mathbf{NP}$, then $\mathbf{P} = \Sigma_2^p$. Generalise the idea to prove that if $\mathbf{P} = \Sigma_k^p$, then $\mathbf{P} = \Sigma_{k+1}^p$, and apply induction.)

Homework Problems

H6.1 Prove that $\mathbf{PH} \subseteq \mathbf{PSPACE}$. (*Hint:* Induction.) (2 points)

H6.2 A *(binary) linear code* of length n is defined by a *parity check matrix* $H \in \{0,1\}^{m \times n}$, and a sequence $c \in \{0,1\}^n$ is a *codeword* if it satisfies all the m parity checks, precisely speaking if $Hc = 0 \pmod 2$. The *covering radius* ρ of a code is the largest distance that any $x \in \{0,1\}^n$ is from some codeword, i.e. $\rho = \max_{x \in \{0,1\}^n} \min_{c \in C} d(x,c)$, where d denotes the Hamming distance and C the set of all codewords.

Show that the following problem belongs to the class Π_2^p:

> COVERING RADIUS
> INSTANCE: A parity check matrix $H \in \{0,1\}^{m \times n}$ and an integer R.
> QUESTION: Is the covering radius of the code defined by H at most R?

(*Note:* The problem is in fact Π_2^p-complete (A. McLoughlin 1984).) (2 points)

H6.3

(a) Show that a language L belongs to class Σ_2^p of the polynomial time hierarchy, if and only if there is a polynomially balanced, polynomially decidable ternary relation R such that for all x:

$$x \in L \quad \equiv \quad \exists y \, \forall z \; R(x,y,z).$$

(*Hint:* Keep in mind that $\Sigma_2^p = \mathbf{NP}^{\mathbf{NP}} = \mathbf{NP}^{SAT}$. The "only if" direction is probably the more difficult one. In that direction use auxiliary variables to represent the nondeterministic choices an **NP** oracle Turing machine makes and the query strings it generates.)

(b) Based on the previous characterisation, show that the language QSAT_2 is Σ_2^p-complete, where:

$$\phi(Y, Z) \in \mathrm{QSAT}_2 \quad \equiv \quad \exists Y \, \forall Z \, \phi(Y, Z) \text{ is true.}$$

(Here $\phi(Y, Z)$ is an abbreviation for a Boolean formula ϕ whose variables are partitioned into two groups $Y = \{y_1, \ldots, y_r\}$ and $Z = \{z_1, \ldots, z_s\}$.)
(2 points)