Midterm II

Problem 1

Complete the following definitions:

(a) Let $A \subseteq \Sigma^*$ and $B \subseteq \Sigma^*$ be languages. We say that *there exists a polynomial-time mapping reduction from A to B*, and write

$$A \leq_p B,$$

if... (5 pts)

there exists a function $f : \Sigma^* \to \Sigma^*$ that can be computed in polynomial time such that

$$\forall x \in \Sigma^* : \; x \in A \iff f(x) \in B.$$

(b) Assume we have already defined the set NP as we did in class:

$$\mathsf{NP} = \bigcup_{c \in \mathbb{N}} \mathsf{NTIME}\left(n^c\right).$$

Let $L \subseteq \Sigma^*$ be a language. We say that L *is* NP*-complete* if... (5 pts)

$L \in \mathsf{NP}$ and for every $A \in \mathsf{NP}$ we have $A \leq_p L$.

Problem 2

Let T be a Turing machine and $w \in \{0,1\}^*$ be a string.

(a) We say that T *zeros* w if at some time during the execution of T on input w, the first $|w|$ cells on the tape are all 0.

Consider the language

$$L = \{\langle M, x\rangle \in \{0,1\}^* : \ M \text{ is a TM } \wedge \ x \in \{0,1\}^* \ \wedge \ M \text{ zeros } x\}.$$

Is L decidable? Prove your answer. (15 pts)

Claim: L is not decidable.

Proof. We show a reduction $H_{\mathsf{TM}} \leq_m L$, where

$$H_{\mathsf{TM}} = \{\langle A, y\rangle : \ A \text{ is a TM } \wedge \ A \text{ halts on input } y\}.$$

We saw in class that H_{TM} is not decidable, and so such a reduction entails that L is not decidable. The reduction function is

$$\langle A, y\rangle \ \mapsto \ \langle A', 1 \circ y\rangle,$$

where $\circ$ denotes concatenation and A' is the following TM. Upon receiving input $s = s_1 s_2 \dots s_n$, A' simulates the execution of A on input $s_2 \dots s_n$, while ignoring the contents of the first cell of the tape and keeping it unchanged. If the simulation halts, then A' sets the contents of the first n cells on the tape to 0, and then halts.

We observe that:

- If A halts on input y, then A' zeros $1 \circ y$.
- If A does not halt on input y, then A' will never finish its simulation, and so it will never alter the contents of the first cell on the tape, and the contents of that cell will remain 1 (because A' does not alter the first cell during the simulation). So A' does not zero $1 \circ y$.

Hence, $\langle A, y\rangle \in H_{\mathsf{TM}} \iff \langle A', 1 \circ y\rangle \in L$. Furthermore the function is clearly computable, and so our proof is complete. ■

(b) We say that T *likes* w if when executing T on input w, the read-write head of T never leaves the first $|w|$ cells of the tape.

Consider the language

$$L = \{\langle M, x\rangle \in \{0,1\}^* : \ M \text{ is a TM } \wedge \ x \in \{0,1\}^* \ \wedge \ M \text{ likes } x\}.$$

Is L decidable? Prove your answer. (15 pts)

Claim: L is decidable.

Proof. The collection of configurations of a Turing machine M on input x such that M likes x is finite. Indeed, the total number of configurations is at most

$$T = |x| \cdot |Q_M| \cdot |\Gamma|^{|x|}.$$

Therefore a decider for L can operate as follows. On input $\langle M, x\rangle$, simulate M on x for $T+1$ steps.

- If M left the first $|x|$ cells during this time, we can safely say that M does not like x.
- Else, if M remained within the first $|x|$ cells and halted during this time, we can safely say that M likes x.
- Otherwise, M remained within the first $|x|$ cells and did not halt during this time. From the pigeonhole principle, some configuration must have repeated twice during this time, which means that M is in a loop, and so it is safe to say that M likes x (it will never leave the first $|x|$ cells, since it has not left this area so far).

Problem 3

For any string $w \in \{0,1\}^*$, let $K(w)$ denote the Kolmogorov complexity of w.

(a) Show that for every $n \geq 1$, there exists a string $x \in \{0,1\}^n$ such that $K(x) \geq n$. (10 pts)

Fix $n \geq 1$, and denote

$$S = \{x \in \{0,1\}^* : \ K(x) < n\},$$
$$E = \{x \in \{0,1\}^* : |x| < n\}.$$

Observe that from the definition of Kolmogorov complexity, there exists a one-to-one mapping

$$S \to E,$$

because every string in S must have some encoding which is a string of length strictly less than n. The mapping is one-to-one because two different strings in S cannot have the same encoding (every encoding encodes at most one string).

Hence,

$$|S| \leq |E| = 2^0 + 2^1 + 2^2 + \cdots + 2^{n-1} = 2^n - 1.$$

However, $|\{0,1\}^n| = 2^n$, and so $|S| < |\{0,1\}^n|$.

Thus, it is not possible that $\{0,1\}^n \subseteq S$. Therefore $\exists x \in \{0,1\}^n : \ x \notin S$. In words, this means that there exists some string of length n with Kolmogorov complexity at least n, as desired.

(b) If $x, y \in \{0,1\}^*$ are strings, we write $\langle x, y\rangle$ to denote the encoding of these two strings given by

$$x_1 x_1 x_2 x_2 \dots x_t x_t \circ 01 \circ y,$$

where $\circ$ denotes concatenation and x_i is the i-th bit of x, i.e. $x = x_1 \dots x_t$.

Let

$$L = \{\langle x, y\rangle \in \{0,1\}^* : \ x, y \in \{0,1\}^* \ \wedge \ K(x) \geq K(y)\}$$

Prove that L is not decidable. (20 pts)

We use a variant of the Berry paradox. Assume towards a contradiction that L is decidable, and let D be a Turing machine that decides L. Consider the following TM B, which takes a natural number n as input:

```
B(⟨n⟩):
    x := 0^n
    for y ∈ {0,1}^n:
        simulate D on ⟨y, x⟩
        if D accepts:                    [this means that K(y) ≥ K(x)]
            x := y
    output x and halt
```

$B(\langle n\rangle)$ will output a string $x \in \{0,1\}^n$ with maximal Kolmogorov complexity.

Hence

$$n \leq \max_{w \in \{0,1\}^n} K(w) \leq K(x) \leq |\langle B, n\rangle| \leq \log(n) + c$$

where the first inequality follows from part (a), the second from the construction of B, the third from the definition of Kolmogorov complexity, and the fourth from the fact that encoding B (and the separator) takes some constant number of bits c.

For large enough values of n, this is a contradiction. ■

Problem 4

A *system of quadratic integer equations over n variables* is a set of equations

$$\begin{aligned} f_1(x_1, \ldots, x_n) &= 0 \\ &\vdots \\ f_t(x_1, \ldots, x_n) &= 0 \end{aligned}$$

where for all $1 \le k \le t$, the function $f_k : \mathbb{Z}^n \to \mathbb{Z}$ is a quadratic polynomial of the form

$$f_k(x_1, \ldots, x_n) = \sum_{1 \le i \le j \le n} a_{ij} x_i x_j + \sum_{1 \le i \le n} b_i x_i + c,$$

and the coefficients a_{ij}, b_i and c are integers for all i and j.

For example, the function $2x_1x_3 - 4x_2x_7 - 3x_4^2 + 5$ is a quadratic polynomial, but $x_1x_2x_3$ is not.

A *solution* of a system is a tuple of integers $(a_1, \ldots, a_n) \in \mathbb{Z}^n$ such that

$$\forall 1 \le k \le t : \; f_k(a_1, \ldots, a_n) = 0.$$

Note: If you get stuck on section (a), we recommend skipping to section (b).

(a) Let φ denote the logical clause $\varphi(x, y, z) = (x \vee y \vee \overline{z})$.

Observe that the set of solutions for the following system

$$x(1-x) = 0, \qquad y(1-y) = 0, \qquad z(1-z) = 0.$$

is precisely the set $\{0, 1\}^3$.

Write down two more equations that when added to the system, result in a system of quadratic integer equations over 4 variables x, y, z, w, with the following property:

For any $a_1, a_2, a_3 \in \mathbb{Z}$,

$$\begin{gathered} (a_1, a_2, a_3) \in \{0,1\}^3 \;\wedge\; \varphi(a_1, a_2, a_3) = 1 \\ \Updownarrow \\ \exists a_4 \in \mathbb{Z} : \; (a_1, a_2, a_3, a_4) \text{ is a solution of the system.} \end{gathered}$$

Prove that your answer is correct. (8 pts)

The two equations are

$$(1 - w) = (1 - x)(1 - y)$$

and

$$(1 - w)z = 0.$$

The first equation is satisfied iff $w = x \vee y$. The second is satisfied iff $w \vee \overline{z} = 1$. Together, they are satisfied iff $x \vee y \vee \overline{z}$, as desired.

(b) Let

$L = \{\langle S\rangle : S \text{ is a system of quadratic integer equations that has a solution } a_1, \ldots, a_n \in \{0,1\}\ \}$.

Prove that L is NP-complete.

You may rely on the result of section (a) regardless of whether you proved it or not. (22 pts)

First, $L \in \mathsf{NP}$. The witness is $a \in \{0,1\}^n$ such that $f_k(a) = 0$ for all k. Verifying that all these equalities holds can be done in polynomial time since arithmetic operations can be performed in polynomial time.

Note: There was a typo in the wording of the question as it appeared in the exam, which effected the membership of L in NP. For this reason, everyone received full credit for the membership part, regardless of whether they wrote anything about it. That typo is corrected in the present document.

Second, we show that 3-SAT $\leq_p L$. Given a 3CNF $\psi = \bigwedge_{i=1}^{k} C_i$, we need to construct in polynomial time a system of quadratic equations. Write every clause C_i as $C_i = \ell_{i,1} \vee \ell_{i,2} \vee \overline{\ell_{i,3}}$, for some literals $\ell_{i,j}$. Let w_i, for $i \in [k]$, be a new variable. Let $x_{i,1}, x_{i,2}, x_{i,3}$ be the 3 variables that appear in C_i. Let $f_i(x_{i,1}, x_{i,2}, x_{i,3}, w_i)$ and $g_i(x_{i,1}, x_{i,2}, x_{i,3}, w_i)$ be the two quadratic equations from part (a) that have a solution iff C_i is satisfied. Note that to generated f_i, g_i we may need to negate some variables when for example $\ell_{i,1}$ is not $x_{i,1}$.

By part (a), it follows that $\langle\psi\rangle \in$ 3-SAT iff $f_1, g_1, \ldots, f_k, g_k$ has a solution. The last piece to notice is that the construction of this list of polynomials can be done in polynomial time, given $\langle\psi\rangle$.