Question 1 (7 × 2 = 14 marks)

For each of the foll[illegible]nswer whether they are TRUE, FALSE, or OPEN (answer not curre[illegible]ers should be justified.

(a) The following [illegible]ble:

$$\{\langle M, N\rangle : M \text{ is a DFA and } N \text{ is a PDA and } L(M) \subseteq L(N)\}$$

(b) There exists a recursively enumerable language whose complement is regular.

(c) Any countable union of recursively enumerable languages is recursively enumerable.

(d) If **P** = **NP** then all languages in **P** are **NP**-complete.

(e) **NL** = **AP**

(f) **NP** ⊆ **TIME**$(2^{O(n)})$

(g) If **P** = **PSPACE** then **NP** = **BPP**

Solution

(a) FALSE: Taking M to be a DFA that accepts Σ^* gives a reduction from the universality of CFLs.

(b) TRUE: Any regular language is recursively-enumerable and its complement is regular.

(c) FALSE: All languages are countable

(d) FALSE: ∅ is not **NP**-complete

(e) FALSE: Space-hierarchy theorem

(f) OPEN: If **P** = **NP** then it would be true. If **NP** = **EXPTIME** then it would be false as then **TIME**$(2^{O(n)}) \subseteq$ **NP**, but **TIME**$(2^{O(n)})$ is not closed under polynomial-time many-one reductions so they cannot be equal.

(g) TRUE: If **P** = **PSPACE** then the polynomial-time hierarchy collapses and **P** = **NP** = **BPP** = **PSPACE**.

Question 2 (3 × 5 = 15 marks)

Are the following languages regular, context-free but not regular, or not context-free? Justify your answer.

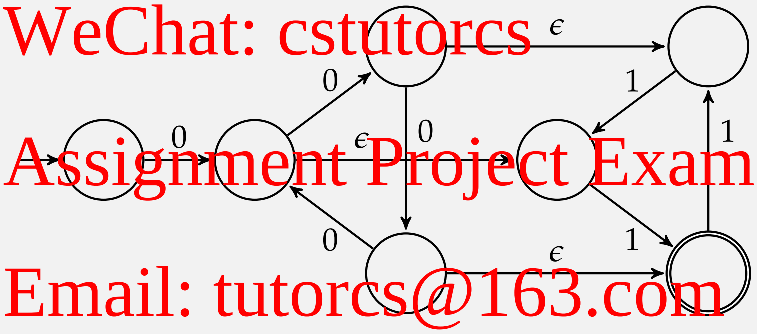
(a) $\{0^n1^m : n,m \geq 1 \text{ and } n \equiv m \pmod 3\}$

(b) $\{0^n1^m : n,m \geq$ [illegible] $m)\}$

(c) $\{0^n1^m : n,m \geq$ [illegible] $3\}$

Solution

(a) The language is regular. Here is an NFA that accepts the language:



Intuitively, the automaton "remembers" $n \pmod 3$ and then checks to make sure $m \pmod 3$ is equal to $n \pmod 3$.

Alternative proof

The language can be written as the union $L_1 \cup L_2 \cup L_3$ where:

$$\begin{aligned}
L_1 &= \{0^{3k+1}1^{3j+1} : k,j \in \mathbb{N}\} &= L\big(0(000)^*1(111)^*\big)\\
L_2 &= \{0^{3k+2}1^{3j+2} : k,j \in \mathbb{N}\} &= L\big(00(000)^*11(111)^*\big)\\
L_3 &= \{0^{3k+3}1^{3j+3} : k,j \in \mathbb{N}\} &= L\big(000(000)^*111(111)^*\big)
\end{aligned}$$

From this we can see that the language is matched by the regular expression:

$$0(000)^*1(111)^* \cup 00(000)^*11(111)^* \cup 000(000)^*111(111)^*$$

(b) This language is not context-free. We show this with the pumping lemma for CFLs.

Assume $L = \{0^n1^m : n,m \geq 1 \text{ and } n \equiv 3 \pmod m\}$ is context-free, and satisfies the pumping lemma for context-free languages with pumping length p.

We will make use of the following result:

Lemma

Let $p \in \mathbb{N}$ with $p \geq 1$, and let $k = 2p+1$. Then for all a, b with $1 \leq a \leq p$ and $0 \leq b \leq p$ we have

$$(k-b) \nmid (k! - a)$$

Proof of [illegible]

Assume [illegible] s $(k-b)|k!$, this statement is equivalent to saying [illegible] p and $b \leq p$, we have $0 < a < k-b$, so $(k-b) \nmid a$ – contradiction.

Let $k = 2p+1$ and consider the word $w = 0^{k!+3}1^k \in L$. By the Pumping Lemma for CFLs, there exists u, v, x, y, z such that:

0. $w = uvxyz$
1. $|vxy| \leq p$
2. $|vy| > 0$
3. $uv^ixy^iz \in L$ for all $i \in \mathbb{N}$.

Suppose u, v, x, y, z satisfy conditions 0, 1 and 2 of the Pumping Lemma. We consider 3 cases:

Case 1: v or y of the form 0^a1^b where $a, b > 0$.
In this case uv^2xy^2z will not be of the form 0^n1^m and therefore not in L.

Case 2: uxy is of the form $0^{k!+3}1^m$ where $m \leq k$.
In this case, v and y are of the form $v = 1^a$ and $y = 1^b$ where at least one of a, b is greater than 0. Now, for $i = k!+3$, we have $uv^ixy^iz = 0^n1^m$ where

- $n = k!+3$, and
- $m > k!+3$

But then $n \equiv k!+3 \not\equiv 3 \pmod{m}$, so $uv^ixy^iz \notin L$.

Case 3: uxy is of the form 0^n1^m where $n < k!+3$ and $m \leq k$.
In this case, let $a = k!+3-n > 0$ and $b = k-m \geq 0$. From condition 1 of the Pumping Lemma, we have $a, b \leq p$. If $n \equiv 3 \pmod{m}$, then $m|n-3$. That is, $(k-b)|(k!-a)$. From the above Lemma we have a contradiction, so $uxy \notin L$.

In all cases, we see that it is not possible to break w into $uvxyz$ such that the conditions of the Pumping Lemma are satisfied. Therefore L is not context-free.

(c) This language is context-free, but not regular. To see that it is context-free, we observe that it can be written as

$$\{0^n1^m \ldots 3n + x \text{ where } x \in \{0,1,2\}\}.$$

It follows ... guage is generated by the grammar $(\{S,T\},\{0,$ ... is the set of rules:

$$\begin{aligned} &\to 0S111 \mid 0T111 \\ &\to \epsilon \mid 0 \mid 00 \end{aligned}$$

To show the language is not regular, we will use the Myhill-Nerode theorem. For each $i, j \in \mathbb{N}$ let:

- $u_i = 0^i$
- $w_{ij} = 1^{3i}$

We see that if $i \neq j$ then $u_i w_{ij} = 0^i 1^{3i}$ is in the language, but $u_j w_{ij} = 0^j 1^{3i}$ is not in the language (as $3i \text{ div } 3 = i \neq j$). Thus the syntactic equivalence relation of the language has infinite index, and by the Myhill-Nerode theorem, the language is not regular.

Question 3 (14 marks)

Show that the following language is not in **coRE**:

Input: A Turing Machine M

Question: Does M halt in the reject state on input ϵ?

Solution

The language is accepted by the following Turing Machine. On input $\langle M \rangle$:

1. Simulate M on input ϵ.
2. If M rejects then accept. If M accepts then reject.

Hence the language is recursively enumerable.

To show that it is undecidable, we show a reduction from the acceptance problem for Turing Machines:

$$\{\langle M, x \rangle : M \text{ accepts } x\}.$$

Given a pair $\langle M, x \rangle$ we construct a Turing Machine M_x as follows:

1. On any input y, simulate M with input x.
2. If M accepts x then reject. If M rejects x then accept.

Clearly M_x rejects x if and only if, M accepts x. Thus this is a reduction from the acceptance [illegible] problem. As we know the acceptance problem for Turing Mac[illegible]ble, it therefore follows that the our language is also undecida[illegible]t decidable.

Question 4 (15 marks)

Show that the following problem is **NP**-complete:

Input: A finite collection C of finite sets, and a number $k \leq |C|$.

Question: Does C contain k sets $S_1, \ldots, S_k$ such that $S_i \cap S_j = \emptyset$ whenever $i \neq j$?

Solution

To show the problem is **NP**-complete, we must show that it is in **NP**, and that it is **NP**-hard.

To show the problem is in **NP**, consider the following algorithm:

On input $C = \{T_1, T_2, \ldots, T_n\}$ and $k \leq n$:

- Non-deterministically guess $C' \subseteq C$
- Check if $|C'| = k$. If not then reject.
- Check if $S \cap T = \emptyset$ for all $S, T \in C'$. If not, then reject
- Accept

This algorithm clearly decides the decision problem with a non-determinstic TM. It remains to show that it runs in polynomial time. If $m = \max |T_i|$ and $n = |C|$ then the running time for each of these steps is at most $O(n) + O(n) + O(n^2 m) = O(n^2 m)$. As the input has size at least $\Omega(m + n)$, the running time of the algorithm is polynomial in the size of the input. Therefore, the problem is in **NP**.

To show the problem is **NP**-hard, we give a reduction from Independent Set:

INDEPENDENT SET

Input: A graph $G = (V, E)$ and an integer k

Question: D[illegible] vertices $V' \subseteq V$ such that $|V'| \geq k$ and for all $u, v \in V'$, [illegible]

This was shown [illegible] tutorials.

Given an instan[illegible]DEPENDENT SET, define the following instance (C, k') of our decision problem.

- $C = \{S_v : v \in V\}$ where $S_v = \{e \in E : v \in e\} \cup \{v\}$
- $k' = k$

We can construct (C, k') from $((V, E), k)$ in time $O(|V||E|)$, so this is a polynomial-time mapping. Note that if $v \neq w$ then $S_v \neq S_w$. Also, note that $S_v \cap S_w = \emptyset$ if and only if v and w do not have an edge in common – i.e. if and only if $\{v, w\} \notin E$.

Suppose $((V, E), k)$ is a yes-instance which maps to (C, k'). Then there exists an independent set $V' \subseteq V$ of size at least k in (V, E). As a subset of an independent set is also an independent set, we can assume that V' is of size exactly k. Consider $C' = \{S_v : v \in V'\} \subseteq C$. As $|V'| = k$, we have $|C'| = k = k'$. As V' is an indpendent set, $\{v, w\} \notin E$ for all $v, w \in V'$, so $S_v \cap S_w = \emptyset$ for all $S_v, S_w \in C'$. Therefore (C, k') is a yes-instance of our problem.

Conversely, suppose $((V, E), k)$ maps to (C, k') and (C, k') is a yes-instance. Let $C' \subseteq C$ be the collection of k' sets such that $S_v \cap S_w = \emptyset$ whenever $v \neq w$. Let $V' \subseteq V$ be the set of $k' = k$ vertices such that $C' = \{S_v : v \in V'\}$. Then, from the definition of the S_v, it follows that $\{v, w\} \notin E$ for all $v, w \in V'$. Therefore V' is an independent set of size k. So $((V, E), k)$ is a yes-instance.

Therefore the mapping is a polynomial-time many-one reduction from INDEPENDENT SET to our problem; and so the problem is **NP**-hard.

Question 5 $(2 \times 7 = 14$ marks$)$

Let MIN be the set of formulas ϕ of propositional logic such that there does not exist a shorter formula ψ with $\phi \equiv \psi$. (Here $\phi \equiv \psi$ means that the formulas are logically equivalent, i.e. they take the same truth value for all assignments of truth value to the variables.)

(a) Show that MIN $\in$ **PSPACE**

(b) Explain why the following argument does not show that $\textsc{Min} \in \mathbf{coNP}$:

> If $\phi \notin \textsc{Min}$ then there exists a shorter equivalent formula ψ. A nondeterministic machine can verify that $\phi \notin \textsc{Min}$ by guessing that formula.

Solution

(a) In lectures [illegible] $\textsc{Min} \in \Pi_2\mathbf{P}$ and $\Pi_2\mathbf{P} \subseteq \mathbf{PSPACE}$. Therefore $\textsc{Min} \in \mathbf{PSP}$[illegible]

(b) The issue i[illegible]ed" formula ψ still needs to be shown to be equivalent to ϕ (the existence of a shorter formula does not establish that $\phi \notin \textsc{Min}$). In order to show the two formulas are equivalent, we must check if they agree on all truth assignments. This can be done non-deterministically – but with a *universal* non-deterministic step. Whether it is possible to do so with existential non-determinism is, of course, an open problem.

Question 6 (2 × 7 = 14 marks)

(a) Show that the complement of an **NP**-complete language is **coNP**-complete.

(b) Show that for any language A, the complement of a language in $\mathbf{NP}^A$ is in $\mathbf{coNP}^A$.

Solution

(a) Let L be an **NP**-complete language, and let L' denote its complement. Since $L \in \mathbf{NP}$, we have that $L' \in \mathbf{coNP}$.

To show that L' is **coNP**-hard. consider any language $K \in \mathbf{coNP}$. Because $K \in \mathbf{coNP}$, its complement, K' is in **NP**. Because L is **NP**-hard, there exists a polynomial-time many-one reduction, f, from K' to L. From the definition of a reduction, we observe that f is also a polynomial-time many-one reduction from K to L'. Therefore, $K \leq_P L'$ and so L' is **coNP**-hard.

(b) Let $L \in \mathbf{NP}^A$. Then there is a non-deterministic, oracle-for-A Turing Machine M that accepts L in polynomial time. Consider the following co-non-deterministic, oracle-for A Turing Machine M':

- M' has the same states as M except the accept and reject states are swapped.
- M' has the same transitions as M but non-determinism is resolved universally rather than existentially (note: this only affects acceptance).

Oracle queries are handled identically.

We observe the following:

$w \in$ ... ns of w in M' end in M''s accept state

... ns of w in M end in M's reject state

... n of w in M ends in M's accept state

... $L(M)$

Therefore L ... accepts $\overline{L}$ in polynomial time. Therefore M' is a polynomial-time, co-non-determinstic TM that accepts $\overline{L}$. So $\overline{L} \in \mathbf{coNP}^A$.

Question 7 (2 × 7 = 14 marks)

(a) Show that if $L_1, L_2 \in \mathbf{BPP}$ then $L_1 \setminus L_2 \in \mathbf{BPP}$.

(b) Show that if $L_1, L_2 \in \mathbf{ZPP}$ then $L_1 \cup L_2 \in \mathbf{ZPP}$.

Solution

(a) If $L_1, L_2 \in \mathbf{BPP}$, let M_1 and M_2 be bounded-error, polynomial-time, probabilistic TMs that probabilistically accept L_1 and L_2 (respectively). By the Amplification Lemma, we can assume that M_1 and M_2 have bounded error $\frac{1}{6}$. That is,

- If $w \in L_1$ then $P(M_1 \text{ accepts } w) \geq \frac{5}{6}$, and
- If $w \notin L_1$ then $P(M_1 \text{ accepts } w) < \frac{1}{6}$,

and similarly for M_2.

Consider the following probabilistic Turing Machine M:

On input w:

- Run M_1 on w
  - If M_1 rejects then REJECT
- Run M_2 on w
  - If M_2 accepts then REJECT
  - Else ACCEPT

As M_1 and M_2 run in polynomial time, M also runs in polynomial time. Also:

- If $w \in L_1 \setminus L_2$ then:

$$P(M \text{ accepts } w) \geq P(M_1 \text{ accepts } w \text{ and } M_2 \text{ rejects } w) \geq \frac{5}{6} \cdot \frac{5}{6} > \frac{2}{3}.$$

- If $w \notin L_1 \setminus L_2$ then either:
  - $w \notin L_1$, in which case:

$$P(M \text{ rejects } w) \geq P(M_1 \text{ rejects } w) \geq \frac{5}{6} > \frac{2}{3}$$

  - $w \in L_1 \cap L_2$, in which case:

$$P(M \text{ rejects } w) \geq P(M_1 \text{ accepts } w \text{ and } M_2 \text{ accepts } w) \geq \frac{5}{6} \cdot \frac{5}{6} > \frac{2}{3}$$

  In either case $P(M \text{ accepts } w) = 1 - P(M \text{ rejects } w) < \frac{1}{3}$.

So M probabilistically accepts $L_1 \setminus L_2$ with bounded error. Hence $L_1 \setminus L_2 \in$ **BPP**.

(b) If $L_1, L_2 \in$ **ZPP** then $L_1, L_2 \in$ **RP** and $L_1, L_2 \in$ **coRP**. For $i = 1, 2$, let M_i, N_i be the probabilistic polynomial time TMs where:

- If $w \in L_i$ then $P(M_i \text{ accepts } w) \geq \frac{1}{2}$ and $P(N_i \text{ accepts } w) = 1$, and
- If $w \notin L_i$ then $P(M_i \text{ accepts } w) = 0$ and $P(N_i \text{ accepts } w) < \frac{1}{2}$.

That is, M_i shows $L_i \in$ **RP** and N_i shows that $L_i \in$ **coRP**.

Consider the following probabilistic TMs:

M	N
On input w:	On input w:
• Run M_1 on w	• Run N_1 on w
• Run M_2 on w	• Run N_2 on w
• If either M_1 or M_2 accepts then ACCEPT	• If either N_1 or N_2 accepts then ACCEPT
• Else REJECT	• Else REJECT

As M_i and N_i run in polynomial time, it follows that M and N also do. Furthermore:

- If $w \in L_i$ then

  - $P(M \text{ accepts } w) \geq P(M_i \text{ accepts } w) \geq \frac{1}{2}$
  - $P($[illegible]$N_i \text{ accepts } w) = 1$

- Therefo[illegible]en

  - $P($[illegible]
  - $P($[illegible]

- On the [illegible] $w \notin L_1 \cup L_2$ then

  - $P(M \text{ accepts } w) = 0$ as neither M_1 nor M_2 will accept w
  - $P(N \text{ accepts } w) \leq P(N_1 \text{ accepts } w) \cdot P(N_2 \text{ accepts } w) \leq \frac{1}{2} \cdot \frac{1}{2} < \frac{1}{2}$

Thus M shows that $L_1 \cup L_2 \in$ **RP** and N shows that $L_1 \cup L_2 \in$ **coRP**. Therefore $L_1 \cup L_2 \in$ **ZPP**.

— END OF EXAMINATION —