Problem 1. [20 points]

Define following terms in your own words:

(a) Uncountable set

> **Solution:** A set that cannot be bijectively mapped to $\mathbb{N}$.

(b) Function $f(n)$ is in $o(g(n)))$ [little-o]

> **Solution:** $f(n) < c.g(n), \quad \forall c > 0,\ \exists n_0$ such that $n \geqslant n_0$

(c) Class NP

> **Solution:** The set of languages that can be decided in
>
> $$\mathrm{NTIME}(n^k), n \in \mathbb{N}, k \geqslant 0.$$

(d) Halting problem

> **Solution:** Deciding the language of Turing Machine and string pairs:
>
> $$\{\langle \mathsf{TM}, w\rangle : \text{TM halts on input } w\}.$$

(e) Turing recognizability

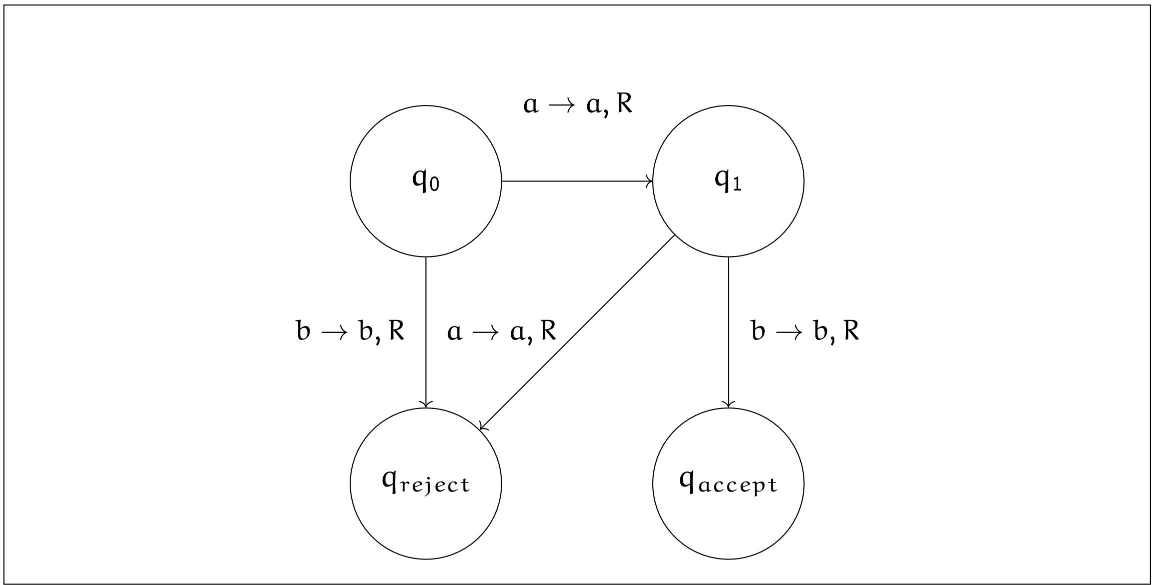
> **Solution:** A property of a language whose 'yes' instances (strings) can be known through a Turing Machine.

Problem 2. [10 points]

Design a Turing machine that accepts the following language:

$$L = \{ab(a \cup b)^*\}.$$

> **Solution:**
>
> $\Gamma = \{a, b, \sqcup\}$



Problem 3. [10 points]

Describe the algorithm for a three-tape Turing machine that computes the function $f(x) = x^2$. One of the tapes should have the value x^2 at the end.

Solution: On input w:

1. Print w in unary on Tape1 and Tape2.
2. Cross-out the left-most 1 on Tape1:
   (a) Append the contents of Tape2 (w in unary) to the contents of Tape3.
3. Repeat step 2 until all 1's are crossed out on Tape1.
4. Tape3 contains w^2 in unary.

Problem 4. [20 points]

Let $\Sigma = \{0, 1\}$. Consider the following eight classes of languages over Σ:

1. **ALL** $= \mathcal{P}(\Sigma^*)$
2. **TR** = Turing-recognizable
3. **TD** = Turing-decidable
4. **NP**
5. **P**
6. **CF** = Context-free
7. **REG** = Regular
8. **FIN** = Finite

[Here $\mathcal{P}$ represents the power-set.]

Each class is a superclass of the next one, and all inclusions except $\mathbf{P} \subseteq \mathbf{NP}$ are known to be proper. Situate each of the following languages as low as you can in the hierarchy (e.g., if a language is in $\mathbf{P}$ but not context-free, the answer is $\mathbf{P}$).

(a) $\{0^n 1^n 0^n : n \geqslant 2\}$

> **Solution:** 5. **P**

(b) $\{0^n 1^n : n \geqslant 2\}$

> **Solution:** 6. **CF**

(c) $\{0^n 1^n : n \geqslant 0 \text{ and } n \neq 2\}$

> **Solution:** 6. **CF**

(d) Σ^* without ϵ

> **Solution:** 7. **REG**

(e) $\{\langle M, w\rangle : M \text{ is a TM that accepts } w\}$

> **Solution:** 2. **TR**

Problem 5. [20 points]

Each of the following languages below are one of three types:

- DEC: Turing-decidable
- REC: Turing-recognizable (but not decidable)
- N-REC: Not Turing-recognizable

For each of the following languages, specify which type it is. Also, follow these instructions:

- If a language L is of type DEC, give a description of a Turing machine that decides L.
- If a language L is of type REC, give a prove that L is not Turing-decidable.
- If a language L is of type N-REC, give a proof that L is not Turing-recognizable.

(a) $EQ_{\text{DFA}} = \{\langle M_1, M_2\rangle : M_1, M_2 \text{ are DFAs with } L(M_1) = L(M_2)\}$.

Solution: EQ_{DFA} is DEC :
M_1 and M_2 are equal *iff* $\overline{M_1} \cap M_2 = \varnothing$
Therefore, we can use the decider for E_{DFA} on $\overline{M_1} \cap M_2$ to decide EQ_{DFA}
$\text{TM}_{\text{Decider}}$:
On input $\langle M_1, M_2 \rangle$:

1. Use M_1 and M_2 to create $M_3 = \overline{M_1} \cap M_2$.
2. Run M_3 on the decider D for E_{DFA}.
   (a) If D accepts, *accept*.
   (b) If D rejects, *reject*.

(b) $\overline{A_{\text{TM}}}$ where $A_{\text{TM}} = \{\langle M, w \rangle : M \text{ is a TM that accepts } w\}$.

Solution: $\overline{A_{\text{TM}}}$ is N-REC :
If a language is both recognizable and co-recognizable, then it decidable. We know that A_{TM} is recognizable. If A_{TM} were co-recognisable, then it would be decidable. But we know that A_{TM} is not decidable. Therefore, $\overline{A_{\text{TM}}}$ must be N-REC.

Problem 6. [10 points]

Show that $ISO \in \mathbf{NP}$, where ISO is defined as:

$$ISO = \{\langle G_1, G_2 \rangle : G_1 \text{ and } G_2 \text{ are isomorphic graphs}\}.$$

[Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are *isomorphic* if there exists a one-to-one correspondence f between V_1 and V_2 with the property that a and b are adjacent in G_1 if and only if $f(a)$ and $f(b)$ are adjacent in G_2, for all a and b in V_1.]

Solution: Here's a verifier for ISO:

V_{ISO} :

On input $(\langle G_1, G_2 \rangle, c)$ where c is the set of vertex pairs (a, b) where $a \in G_1$ and $b \in G_2$:

1. Compute $PAIRS \leftarrow c \times c$.
2. For each element $\{(a_1, b_1), (a_2, b_2)\}$ in $PAIRS$:
   (a) If (a_1, a_2) are adjacent, but (b_1, b_2) are not, *reject*.
   (b) If (a_1, a_2) are not adjacent, but (b_1, b_2) are adjacent, *reject*.

3. *Accept*

Steps 1 and 2 have runtime in $O(n^2)$ where n is the number of nodes in either graph. Therefore, V has runtime in $O(n^2)$ and $V_{\text{ISO}} \in \mathbf{P}$.