**ComS 331** **Spring 2025** **Name:** Mobina Amrollahi

**HW 13 Due: May $4^{th}$ 2025**

1. We know that the language $L = \{0^n 1^m : n, m \in \mathbb{N}, n = m\}$ is context-free, but not regular. Consider now the language $L' = \{0^n 1^m : n, m \in \mathbb{N}, n \bmod 3 = m \bmod 3\}$. Is it regular? Is it context-free?
   **Answer**
   This language is regular and can be expressed by the following regular expression:
$$(000)^*(111)^* + 0(000)^*1(111)^* + 00(000)^*11(111)^*$$

2. Prove or disprove that the language $L = \{a^n b^m a^n b^m : n, m \in \mathbb{N}\}$ is context-free.
   **Answer**
   We will prove by contradiction that this language is not context-free.

   *Proof.* Assume the contrary that $L$ is context-free. Let $p$ be the given pumping length. Now, consider the string $s = a^p b^p a^p b^p \in L$. $\mid s \mid \geq p$, and based on pumping lemma, $s$ may be divided into five pieces s $= uvxyz$ satisfying the conditions:

   (a) for each $i \geq 0$, $uv^i xy^i z \in L$,
   (b) $\mid vy \mid > 0$, and
   (c) $\mid vxy \mid \leq p$.

   The restriction $\mid vxy \mid \leq p$ guarantees that $v$ and $y$ can only span at most two consecutive runs out of the four runs. That is whether they span at:

   (a) the first run of $a$'s and the first run of $b$'s, and **not** the **second** run of $a$'s and the **second** run of $b$'s:
$$\underbrace{a^p b^p}_{vxy \subset} a^p b^p$$
   In this case, consider the string $uv^0xy^0z$. Since $vxy$ contains at least one $a$ or $b$, then either $\mid uv^0xy^0 \mid_a < p$ or $\mid uv^0xy^0 \mid_b < p$. However, the second run of $a$'s and the second run of $b$'s has length $p$, which mean the run of $a$'s and $b$'s don't match with each other anymore. Therefore, we arrive at a contradiction that $uv^0xy^0z \notin L$ and conclude that $L$ is not context-free.

   (b) the first run of $b$'s and the **second** run of $a$'s, but not the **first** run of $a$'s or the **second** run of $b$'s:
$$a^p \underbrace{b^p a^p}_{vxy \subset} b^p$$
   In this case, consider the string $uv^0xy^0z$. Since $vxy$ contains at least one $a$ or $b$, then either $\mid uv^0xy^0 \mid_a < p$ or $\mid uv^0xy^0 \mid_b < p$. However, the first run of $a$'s and the second run of $b$'s has length $p$, which mean the run of $a$'s and $b$'s don't match with each other anymore. Therefore, we arrive at a contradiction that $uv^0xy^0z \notin L$ and conclude that $L$ is not context-free.

(c) the second run of $a$'s and the second run of $b$'s, and **not** the **first** run of $a$'s and the **first** run of $b$'s:

$$a^p b^p \underbrace{a^p b^p}_{vxy \subset}$$

In this case, consider the string $uv^0xy^0z$. Since $vxy$ contains at least one $a$ or $b$, then either $\mid uv^0xy^0 \mid_a < p$ or $\mid uv^0xy^0 \mid_b < p$. However, the first run of $a$'s and the first run of $b$'s has length $p$, which mean the run of $a$'s and $b$'s don't match with each other anymore. Therefore, we arrive at a contradiction that $uv^0xy^0z \notin L$ and conclude that $L$ is not context-free.

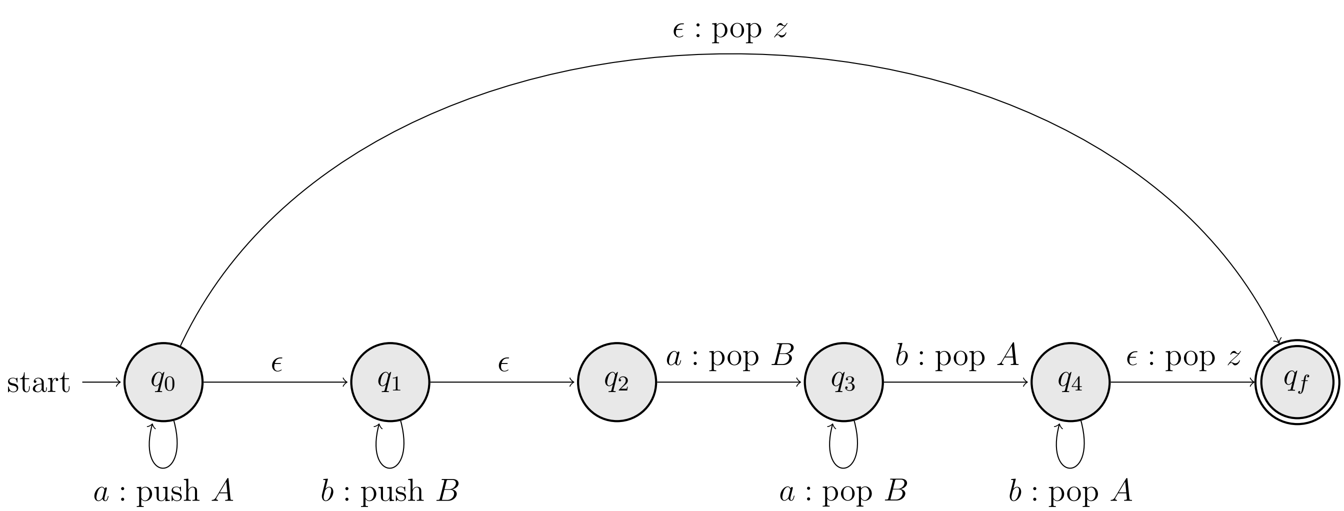
□

3. Prove or disprove that the language $L = \{a^n b^m a^m b^n : n, m \in \mathbb{N}\}$ is context-free.

**Answer**

Let $M$ be $(Q, \Sigma, \Gamma, \delta, q_0, z, F)$, where

$$\begin{aligned} Q &= \{q_0, q_1, q_2, q_f\}, \\ \Sigma &= \{a, b\}, \\ \Gamma &= \{A, z\}, \\ F &= \{q_f\}, \text{ and} \end{aligned}$$




We can also consider the grammar $G = (\{S, T\}, \{a, b\}, R, S)$ where the set of rules, $R$, is

$$\begin{aligned} S &\to aSb \mid T \\ T &\to bTa \mid \epsilon. \end{aligned}$$

4. Are the following languages Turing-decidable? Turing-acceptable but not Turing-decidable? Not even Turing-acceptable? For each answer, give an explanation of your reasoning (just as in class, $M$ is a generic deterministic Turing machine, $w$ a generic input string to it, and $\rho$ is an encoding function).

- $L_1 = \{\rho(M) : |L(M)| > 10\}$

  **Answer.** We show that the language $L_1 = \{\rho(M) \mid |L(M)| > 10\}$ is **undecidable** by applying

Rice's Theorem.

Rice's Theorem states that any non-trivial semantic property of the language recognized by a Turing machine is undecidable. A property is semantic if it depends only on the language $L(M)$, not the syntax of $M$, and it is *non-trivial* if it is true for some Turing machines and false for others.

The property "$|L(M)| > 10$" depends only on the language recognized by $M$, and not on the internal structure of $M$. It is also non-trivial because:

- There exists a Turing machine $M_1$ such that $L(M_1) = \Sigma^*$, so $|L(M_1)| > 10$,
- And there exists a Turing machine $M_2$ that accepts fewer than or equal to 10 strings.

Since this is a non-trivial property of $L(M)$, Rice's Theorem implies that $L_1$ is undecidable.

To show that $L_1$ is **Turing-acceptable**, simulate $M$ on all strings in $\Sigma^*$ using dovetailing. Keep a counter $n$, initialized to 0, and increment it each time $M$ accepts a string. If $n$ exceeds 10, accept $\rho(M)$.

- $L_2 = \{\rho(M) : |L(M)| \leq 10\}$
  **Answer.** Since $L_2 = \overline{L_1}$, and $L_1$ is only Turing-acceptable, then $L_2$ **is not even Turing-acceptable**.

- $L_3 = \{\rho(M)\rho(w) : M \searrow w \text{ in 10 steps or less}\}$
  **Answer.** We will show that $L_3$ is decidable by constructing the following Turing machine $S$:

  $S =$ On input $\rho(M)\rho(w)$, where $M$ is a Turing machine and $w$ is a string:

  Simulate $M$ on input $w$ for at most 10 configurations.

  If $M$ halts then **accept**.

  **rejects**

  Since $S$ only needs to simulate $M$ for a finite number of steps (at most 10), this procedure always halts. Therefore, $L_3$ **is decidable**.

- $L_4 = \{\rho(M)\rho(w) : M \searrow w \text{ in more than 10 steps}\}$
  **Answer.** We know that $K_0$, the problem of determining whether a Turing machine accepts a given input $w$, is undecidable:

$$K_0 = \{\rho(M)\rho(w) : M \searrow w\}.$$

We will prove by contradiction that if $L_4$ is decidable, then $K_0$ would also be decidable.

*Proof by Contradiction.* Assume there exists a Turing machine $R$ that decides $L_4$. We will construct a Turing machine $S$ that decides $K_0$ as follows:

$S =$ "On input $\rho(M)\rho(w)$, where $M$ is a Turing machine and $w$ is a string:

(a) Run $R$ on input $\rho(M)\rho(w)$.
   - If $R$ **accepts**, then **accepts**.
   - If $R$ **rejects**:
     * Simulate $M$ on input $w$.
     * If $M$ halts in at most ten steps, **accept**.
     * Otherwise **reject**.

On input $\rho(M)\rho(w)$, machine $S$ first queries $R$. If $R$ accepts, we conclude that $M$ accepts $w$ in more than 10 steps and thus accept. If $R$ rejects, it means $M$ either does not accept $w$ at all or does so in 10 steps or fewer. We then simulate $M$ on $w$ for at most 10 steps: if it accepts within that time, we accept; otherwise, we reject. This process allows $S$ to decide whether $M$ accepts $w$, which contradicts the known undecidability of $K_0$. Therefore, our assumption must be false, and $L_4$ is not decidable.
Note that $L_4 \subseteq K_0$, and since $K_0$ is Turing-acceptable, $L_4$ is also Turing-acceptable. To construct a Turing machine $R$ that accepts $L_4$, we can use a Turing machine $S$ that accepts $K_0$ and refine its acceptance condition as follows:

$R =$ "On input $\rho(M)\rho(w)$, where $M$ is a Turing machine and $w$ is a string:

(a) Run $S$ on $\rho(M)\rho(w)$.

(b) If $S$ accepts, simulate $M$ on $w$.

- If $M$ goes beyond ten configurations (step) while being simulated on $w$, **accept**.

Since $S$ accepts exactly those pairs where $M$ accepts $w$, and $R$ further filters to ensure the acceptance happens in more than 10 steps, $R$ semidecides $L_4$. Hence, $L_4$ **is Turing-acceptable**.

5. Use reduction to prove that the language

$$L = \{\rho(M_1)\rho(M_2) : L(M_1) \subseteq L(M_2)\}$$

is not decidable ($M_1$ and $M_2$ are Turing machines, of course).
**Answer.** We have already established the undecidability of $K_0$, the problem of determining whether a Turing machine accepts a given input $w$.

$$K_0 = \{\rho(M)\rho(w) : M \searrow w\}.$$

By proving by contradiction, we will show that if $L$ is decidable, then $K_0$ is also decidable.
*Prove by Contradiction.* Let $R$ be a TM that decides $L$. We'll construct TM $S$ to decide $K_0$ by working in the following manner:
$S =$ "On input $\rho(M_2)\rho(w)$, where $M_2$ is a TM and $w$ is a string:

1. Run $R$ on input $\rho(M_1)\rho(M_2)$, where $M_1$ is a TM that rejects everything and only accepts $w$
    1. If $R$ accepts, *accept*;
    2. If $R$ rejects, *reject*."

If $R$ accepts, then $\{w\} = L(M_1) \subseteq L(M_2)$, which implies $w \in L(M_2)$, so $S$ accepts. If $R$ rejects, then $\{w\} = L(M_1) \not\subset L(M_2)$ which implies $w \notin L(M_2)$ and $S$ rejects. Thus, $S$ decides $K_0$ using $R$, implying $K_0$ is decidable, contradicting the known undecidability of $K_0$. Therefore, $L$ is not decidable.