Problem 1. Turing Machines (20 points)

Let $\Sigma = \{0, 1\}$. Construct the state–transition diagram of a deterministic Turing machine that decides the following language:

$$L = \{w | w \in \Sigma^* \text{ and } w \text{ contains at least two 0s and contains no two or more consecutive 0s} \}$$

Make ALL transitions explicit in the diagram.

Solution:
Problem 2. Recursive Functions (20 points)

Consider the function:

\[
\text{EVEN}(n) := \begin{cases} 
1, & \text{If } n \text{ is an even number, i.e. } n = 0, 2, 4, 6, 8, \ldots \\
0, & \text{Otherwise}
\end{cases}
\]

Show that the function \text{EVEN} is primitive recursive.

Any functions used in your solution and that were not shown to be primitive recursive in class or one of the handouts must be shown to be primitive recursive as part of your solution.

**Hints:** Define \text{EVEN} using the following template:

\[
\begin{align*}
\text{EVEN}(0) &= \\
\text{EVEN}(k + 1) &=
\end{align*}
\]

**Solution:**

\[
\begin{align*}
\text{EVEN}(0) &= 1 \\
\text{EVEN}(k + 1) &= \text{MINUS}(1, \text{EVEN}(k)) = “1 - \text{EVEN}(k)”.
\end{align*}
\]
Problem 3. Decidability (20 points)

Let \( \Sigma = \{a, b, c\} \). Show that the following language is decidable:

\[
L = \{a^n b^n c^n | n \geq 0\}
\]

where \( a^n \) denotes a string containing \( n \) consecutive occurrences of the letter “a”.

**Solution:** Let’s construct a TM \( M \) that decides \( L \):

\( M = \) “on input string \( w \):

1. If \( w \) is the empty word, then accept \( w \).

2. Otherwise, scan the input to make sure that \( w \) consists of a sequence of one or more “a”s followed by a sequence of one or more “b”s followed by a sequence of one or more “c”s.

3. REPEAT
   
   Locate the first “a” on the tape and cross it out.
   
   Locate the first “b” on the tape and cross it out. If no “b” is found, reject \( w \).
   
   Locate the first “c” on the tape and cross it out. If no “c” is found, reject \( w \).

   UNTIL no more “a”s are found on the tape.

4. Scan the tape to check if any “b”s or “c”s remain on the tape.
   
   If so, reject \( w \).
   
   Otherwise, accept \( w \).”

This machine \( M \) stops on all inputs; it accepts an input \( w \) iff \( w \) is of the form \( a^n b^n c^n \) for some \( n \geq 0 \); and rejects input \( w \) otherwise.
Problem 4. Undecidability (20 points)

Let \( \Sigma \) be an alphabet. Show that

\[
\text{FINITE}_{TM} = \{ \langle M \rangle | M \text{ is a Turing machine and } M \text{ accepts only finitely MANY words} \}
\]

is undecidable.

**Hint:** We already showed that \( A_{TM} \leq_m \text{INFINITE}_{TM} \), where

- \( \text{INFINITE}_{TM} = \{ \langle M \rangle | M \text{ is a Turing machine and } M \text{ accepts infinitely many words} \} \)
- \( A_{TM} = \{ < M, w > | M \text{ is a TM, } w \text{ is a word, and } M \text{ accepts } w \} \).

**Solution:**

Since \( A_{TM} \) is undecidable and we proved already that \( A_{TM} \leq_m \text{INFINITE}_{TM} \), then we know that \( \text{INFINITE}_{TM} \) is undecidable. Note that \( \text{FINITE}_{TM} \) is the complement of \( \text{INFINITE}_{TM} \). Hence, \( \text{FINITE}_{TM} \) has to be undecidable, because otherwise \( \text{INFINITE}_{TM} \) would be decidable.

(Note: Formally speaking \( \text{INFINITE}_{TM} \) is not equal to \( \Sigma^* - \text{FINITE}_{TM} \). Nevertheless you can express \( \text{INFINITE}_{TM} \) in terms of \( \text{FINITE}_{TM} \) as follows:

\[
\text{INFINITE}_{TM} = \{ \langle M \rangle | M \text{ is a TM} \} \cap [\Sigma^* - \text{FINITE}_{TM}]
\]

Since \( \{ \langle M \rangle | M \text{ is a TM} \} \) is decidable and decidable languages are closed under intersection and complementation, if \( \text{FINITE}_{TM} \) were decidable so would \( \text{INFINITE}_{TM} \).)


Problem 5. N, NP, NP–completeness (20 points + 10 extra points)

Let $G = (V, E)$ be an undirected graph. We say that $G$ contains a $k$–independent set if there is a subset $I$ of $V$ containing $k$ nodes such that whenever $a$ and $b$ belong to $I$ then there is NO edge between $a$ and $b$ in $E$.

Consider the language:

$$\text{INDEPENDENT} = \{ \langle G, k \rangle | G \text{ is an undirected graph and } G \text{ contains a } k\text{–independent set} \}.$$ 

5.1. (20 points) Show that INDEPENDENT is in NP.

Solution: The following nondeterministic TM constructs a $k$-independent set for $G$ if one exists:

$M_{\text{ind}} = "$on input $\langle G, k \rangle$:

1. Nondeterministically construct a set $I$ containing $k$ nodes from $V$.
2. For every pair of nodes $a$ and $b$ in $I$:
   (a) Check that there is no edge $(a, b)$ in $E$.
   (b) If there is such edge in $E$, reject $\langle G, k \rangle$.
3. If the checking on step 2 succeeded, then accept $\langle G, k \rangle.$"

input: $\langle G, k \rangle$, where $G = (V, E)$

size of the input: $n = |V|

number of steps for input of size $n$:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>number of steps the instruction takes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>2</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>3</td>
<td>$O(1)$</td>
</tr>
</tbody>
</table>

$k = O(n)$ and so there are at most $n^2$ pairs of nodes from $I$ to check

So, $M_{\text{ind}}$ runs in $O(n^2)$ and so polynomial time on the size of the input.

5.2. (10 extra points) Show that INDEPENDENT is NP–complete.

Hint: Show that CLIQUE $\leq_P$ INDEPENDENT, where

$$\text{CLIQUE} = \{ \langle G, k \rangle | G \text{ is an undirected graph and } G \text{ contains a } k\text{–clique} \}.$$ 

Solution: Consider the following reduction from CLIQUE to INDEPENDENT:

$$\text{CLIQUE} \leq_P \text{INDEPENDENT}$$

$\langle G, k \rangle$ belongs to CLIQUE iff $f(\langle G, k \rangle) = \langle \bar{G}, k \rangle$ belongs to INDEPENDENT

where $\bar{G} = (V, E)$ is constructed from the graph $G = (V, E)$ as follows: $\bar{G}$ contains the same nodes as $G$ does; and there is an edge in $\bar{G}$ between nodes $a$ and $b$ if and only if there in no such edge in $G$. In other words, $\bar{E}$ is the complement of $E$. (OVER)
Note that $\mathcal{G}$ can be constructed from $G$ in at most $O(n^2)$, where $n = |V|$, since there are at most $n^2$ edges to include in $\mathcal{G}$. Hence the function $f$ above can be computed by a polynomial time TM.

Let's prove that $G$ has a $k$-clique $V'$ iff $V'$ is a $k$-independent set in $\mathcal{G}$:

- $G$ has a $k$-clique $V'$ if and only if for each pair of nodes $a$ and $b$ in $V'$, the edge $(a, b)$ belongs to $E$
- $G$ has a $k$-clique $V'$ if and only if for each pair of nodes $a$ and $b$ in $V'$, the edge $(a, b)$ does not belong to $\mathcal{E}$
- $G$ has a $k$-clique $V'$ if and only if $V'$ is a $k$-independent set for $\mathcal{G}$. 