1. Define:

\[ A_{DFA} = \{ < M, w > \mid M \text{ is a DFA, } w \text{ is a string and } w \in L(M) \} \]

\[ EQ_{DFA} = \{ < M, N > \mid M \text{ and } N \text{ are DFAs, and } L(M) = L(N) \} \]

\[ INCL_{DFA} = \{ < M, N > \mid M \text{ and } N \text{ are DFAs, and } L(M) \subseteq L(N) \} \]

Figure 1: (i) DFA A (on left); and (ii) DFA B (on right).

Answer the following questions and give reasons for your answers.

(a) Is \( < A, aab > \in A_{DFA} \)? No. \( A \) does not accept \( aab \)

(b) Is \( < A, abb > \in A_{DFA} \)? Yes. \( A \) accepts \( abb \)

(c) Is \( < B, bbaabbaab > \in A_{DFA} \)? No. \( B \) does not accept \( bbaabbaab \)

(d) Is \( < A, B > \in EQ_{DFA} \)? No. \( L(A) \neq L(B) \) (\( ab \notin L(B) - L(A) \))

(e) Is \( < A, A > \in EQ_{DFA} \)? Yes. \( L(A) = L(A) \)

(f) Is \( < B, A > \in EQ_{DFA} \)? No. \( L(B) \neq L(A) \)

(g) Is \( < A, B > \in INCL_{DFA} \)? No. \( abab \notin L(A) - L(B) \)

(h) Is \( < B, A > \in INCL_{DFA} \)? No. \( ab \in L(B) - L(A) \)

(i) Is \( < A, A > \in INCL_{DFA} \)? Yes. \( L(A) \subseteq L(A) \)
2. (a) Consider the language

\[ C = \{ < M, w > \mid M \text{ is a DFA, and some string of } L(M) \text{ contains string } w \text{ as a substring} \} \]

Is the language \( C \) decidable or undecidable? Prove your answer.

\[ \text{C is decidable. \text{Decider for } C:} \]

- On input \( < M, w > \):
  1. Construct DFA \( B \) for \( \Sigma^* w \Sigma^* \)
  2. Construct DFA \( D \) for \( L(M) \cap L(B) \) // reg. \\
     \( \text{Languages closed under intersection} \)
  3. Decide whether \( L(D) = \emptyset. \)
     If yes, reject. If no, accept."

(b) Consider the language

\[ D = \{ < M > \mid M \text{ is a Turing machine and there exists a DFA } A \text{ such that } L(M) \subseteq L(A) \} \]

Is the language \( D \) decidable or undecidable? Prove your answer.

\[ \text{D is decidable. If } \Sigma \text{ is the input alphabet of } M, \]
\[ \Sigma^* \text{ is a regular language, and } L(M) \subseteq \Sigma^* \]

Hence

\[ D = \{ < M > \mid M \text{ is a Turing machine} \} \]

A decider for \( D \) needs only to check whether the input is a correct encoding of a Turing machine.
3. (a) Does the following instance of the Post Correspondence Problem have a match (that is, a solution). Justify your answer.

\[
\{ \begin{bmatrix} 1001 \\ 01 \end{bmatrix}, \begin{bmatrix} 101 \\ 1001 \end{bmatrix}, \begin{bmatrix} 1001 \\ 10001 \end{bmatrix}, \begin{bmatrix} 01 \\ 0110 \end{bmatrix} \} 
\]

Match: \[
\begin{bmatrix} 01 \\ 0110 \end{bmatrix} \begin{bmatrix} 1001 \\ 01 \end{bmatrix}
\]

(b) Let \( \Sigma = \{a, b\} \). We denote \( \text{finite}(\Sigma) = \{ L \subseteq \Sigma^* \mid L \text{ is finite} \} \). The elements of the set \( \text{finite}(\Sigma) \) are the finite languages over \( \Sigma \).

Show that the set \( \text{finite}(\Sigma) \) is countable.

We list elements of \( \text{finite}(\Sigma) \) in stages.

Stage 0: list \( \emptyset \)

For \( i \geq 1 \): list all sets of size at most \( i \) consisting of strings of length at most \( i \). Exclude sets that were listed on some previous stage.

An arbitrary finite set \( A \in \text{finite}(\Sigma) \) is listed in stage \( k \) where \( k \) is the maximum of the cardinality of \( A \) and the length of the longest string of \( A \).
4. Give a complete construction, that is, a state transition diagram of a single-tape deterministic Turing machine $M$ that decides the language

$$\{ a^i b^k \mid 0 \leq i \leq k \}$$

- Give also the sequence of configurations that your Turing machine $M$ enters when started on input string $abb$. 

Sequence of configurations:

1. $a b b$
2. $u 2 b b$
3. $u b 2 b$
4. $u b b 2 u$
5. $u b 3 b u$
6. $u b 4 b u$
7. $u 4 b u$
8. $u 6 b u$
9. $u 1 b 5 u$
10. $u 6 5 u$
11. $u 6 5 6 u$
12. $u 6 6 5 u$
13. $u 6 6 6 5 u$
14. $u 6 6 6 6 5 u$
15. $u 6 6 6 6 6 5 u$

ALL UNDEFINED TRANSITIONS GO TO THE REJECT STATE.
5. Given an example of a non-context-free language $A$ such that $A$ is in the class $L (= \text{SPACE}(\log n))$. You should briefly explain how a logarithmic space deterministic TM decides your language $A$. You do not need to prove that $A$ is non-context-free.

\[ A = \{ a^ib^ic^i \mid i \geq 0 \} \]

Deterministic TM $M$ for $A$:

1. First scan the input and check that it is in $a^*b^*c^*$
2. Return the tape head to left end.
3. Count the number of $a$'s and store the result in binary on the work tape.

   Similarly, count the number of $b$'s and $c$'s.

4. When $M$ has the three counts stored on the work tape, by zig-zagging check that they are all equal.

$M$ needs to store 3 binary numbers, each at most logarithmic length.

6. Let $B$ be a Turing-recognizable language and assume that $B \leq_m \overline{B}$. ($\overline{B}$ is the complement of $B$.)

Prove that $B$ is decidable.

$B \leq_m \overline{B}$ implies $\overline{B} \leq_m B$ (use the same mapping reduction)

Since $B$ is Turing recognizable, so is $\overline{B}$ (Th. 5.28)

Since $B$ and $\overline{B}$ are Turing recognizable, $B$ is decidable (Th. 4.22)
7. What is the relationship (equal "="; strict inclusion "⊂" or "⊃"; inclusion that is not known to be strict "⩽" or "⩾") between the following pairs of complexity classes. **Justify** your answers.

(a) \(\text{TIME}(n^3 \cdot \log n + n \cdot (\log n)^3)\) and \(\text{TIME}(n^3 + n^2 \cdot (\log n)^5)\)

\[ t_1(n) = n^3 + n^2 \cdot (\log n)^5, \quad t_2(n) = n^3 \log n + n \cdot (\log n)^2 \]

\[ t_1(n) = \Omega(t_2(n)) \quad \text{and time-hierarchy theorem does not apply.} \]

(b) \(\text{SPACE}(2^{n^2})\) and \(\text{SPACE}(2^{n^2 + n})\)

\[ 2^{n^2} \subset \text{SPACE}(2^{n^2 + n}) \quad \text{and strictness follow from space hierarchy theorem.} \]

(c) \(\text{TIME}(2^{n^2})\) and \(\text{TIME}(n^3 \cdot 2^n)\)

\[ n^3 \cdot 2^n = \Omega\left(\frac{2^{n^2}}{n^2}\right) \quad \text{and use time hierarchy theorem} \]

(d) \(\text{NSPACE}(n^2 \cdot (\log n)^3)\) and \(\text{SPACE}(n^5)\)

\[ \text{NSPACE}(n^2 (\log n)^3) \subset \text{SPACE}(n^4 (\log n)^6) \subset \text{SPACE}(n^5) \]

\[ \text{Switch} \quad \text{space hierarchy} \]

(e) \(\text{NTIME}(n^2 \cdot \log n)\) and \(\text{SPACE}(n^5)\)

\[ \text{NTIME}(n^2 \cdot \log n) \subset \text{NSPACE}(n^2 \log n) \subset \text{SPACE}(n^4 (\log n)^9) \]

\[ \subset \text{SPACE}(n^5) \quad \text{Switch} \quad \text{space hierarchy} \]
8. Define \( E_{CFG} = \{ <G> \mid G \text{ is a context-free grammar and } L(G) = \emptyset \} \).

Answer the following questions and \textit{justify} your answers.

(a) Does there exist a \textbf{decidable} language \( A \) such that \( A \leq_m E_{CFG} \)?

\textbf{Yes.} \( E_{CFG} \) \textit{is decidable and we can choose}
\[ A = E_{CFG}. \]

(b) Does there exist a \textbf{decidable} language \( B \) such that \( E_{CFG} \leq_m B \)?

\textbf{Yes.} \textit{Choose} \( B = E_{CFG} \).

(c) Does there exist an \textbf{undecidable} language \( C \) such that \( C \leq_m E_{CFG} \)?

\textbf{No.} \textit{Since} \( E_{CFG} \) \textit{is decidable,} \( C \leq_m E_{CFG} \) \textit{would imply that} \( C \) \textit{is decidable.}

(d) Does there exist an \textbf{undecidable} language \( D \) such that \( E_{CFG} \leq_m D \)?

\textbf{Yes.} \textit{Let} \( M_0 \) \textit{be a TM such that} \( a \in L(M_0), b \notin L(M_0). \)
\textbf{We define} \( f \) \textbf{by setting:}
\[ f(w) = \left\{ \begin{array}{ll}
\langle M_0, a \rangle & \text{if } w \text{ is an encoding of } G \text{ and } L(G) = \emptyset \\
\langle M_0, b \rangle & \text{if } \langle M_0, a \rangle \text{ is a grammar, } L(G) \neq \emptyset \\
\langle M_0, b \rangle & \text{if } w \text{ does not encode a grammar.}
\end{array} \right. \]
\( f \) \textit{is computable because} \( E_{CFG} \) \textit{is decidable.}
\( E_{CFG} \leq_m \text{ATM via function} f. \)
9. We define

\[ P_{TM} = \{ <M, w> \mid M \text{ is a Turing machine and } w \text{ is a string and } M \text{ accepts } w^k \text{ for all positive integers } k \}. \]

**Without using** Rice's theorem show that \( P_{TM} \) is undecidable.

We reduce \( A_{TM} \) to \( P_{TM} \).

Suppose \( TM \ Q \) decides \( P_{TM} \).

Construct following
decider for \( A_{TM} \):

\[ D = " \text{On input } <M, w> \text{ where } M \text{ is a } TM, w \text{ is a string} \]

1. Construct \( TM \ X \) as follows:

\[ X = " \text{On input } y \]

1. Run \( M \) on \( w \).

2. If \( M \) accepts, accept.
   
   If \( M \) rejects, reject."

2. Run decider \( Q \) on input \( <X, w> \)

3. If \( Q \) accepts, accept.
   
   If \( Q \) rejects, reject."

\( D \) decides \( A_{TM} \) correctly because:

- \( M \) accepts \( w \) implies \( X \) accepts \( w^k \) for all \( k \)
- \( M \) does not accept \( w \) implies \( L(X) = \emptyset \)
10. What is the relationship between the following pairs of classes of languages. In each case insert the correct symbol on the dotted line between each pair: equal ("="); strict inclusion in one direction ("\(\text{STRICT} \subset\)" or "\(\supset\text{STRICT}\)"); inclusion that is not known to be strict ("\(\subseteq\)" or "\(\supseteq\)"); or incomparable/not known ("OTHER"). The last option includes cases where the classes are incomparable or no inclusion relation is known.

- In order to avoid ambiguity between strict and non-strict inclusions, please write "\(\text{STRICT} \subset\)" or "\(\supset\text{STRICT}\)" for the strict inclusions.

In addition to standard notations for complexity classes from our textbook, we denote:

REGULAR = the class of regular languages

LBA = the class of languages decided by deterministic linear bounded automata

You do not need to explain the answers.

(a) NP .................. PSPACE
\(\subseteq\)

(b) REGULAR \(\text{STRICT} \subset\) NL

(c) L \(\subseteq\) NP

(d) P \(\subseteq\) PSPACE

(e) LBA \(\supset\text{STRICT}\) REGULAR

(f) LBA \(\subseteq\) SPACE(n)

(g) LBA \(\supset\text{STRICT}\) NL

(h) coNP \(\text{OTHER}\) NP

(i) coNL \(=\) NL

(j) coNL \(\subseteq\) NP
11. Here \( \leq_P \) denotes the polynomial time reducibility relation. Note: To show that a particular implication does not hold, you need to give a counter-example.

(a) If \( A \leq_P B \) and \( B \) is a regular language, does that imply that \( A \) is in \( P \)?
Justify your answer: why or why not?

Yes. \( B \) is regular implies that \( B \in P \) and claim follows from Th. 7.31

(b) If \( A \leq_P B \) and \( B \) is a regular language, does that imply that \( A \) is regular?
Justify your answer: why or why not?

\[ \text{No. Let } A = \{ a^i b^i \mid i \geq 0 \} \text{ and define } \]
\[ f(w) = \begin{cases} a & \text{if } w = a^i b^i, i \geq 0 \\ a a & \text{otherwise.} \end{cases} \]
\( f \) is computable in pol. time and \( A \leq_P B \) via \( f \).

(c) If \( A \leq_P B \) and \( A \) is a regular language, does that imply that \( B \) is context-free?
Justify your answer: why or why not?

\[ \text{No. Choose } A = \{ a^i \} \text{, } B = \text{ATM and let } M_0 \text{ be a TM} \]
that decides \( \{ a^i \} \). Define \( f(a) = \langle M_0, a \rangle \) and
\[ f(w) = \langle M_0, a a \rangle \text{ otherwise. } f \text{ can be computed in pol.time} \]
and \( \{ a^i \} \leq_P \text{ATM via function } f \).

(d) If \( A \leq_P B \) and \( A \) is a regular language, does that imply that \( B \) is decidable?
Justify your answer: why or why not?

\[ \text{No. Use the same example as above.} \]