5.1 Show that EQ_{CFG} is undecidable

Let's recall that ALL_{CFG} is undecidable. We will prove by contradiction with assumption that EQ_{CFG} is decidable. We can build a decider M for ALL_{CFG}. Indeed, M = on input $\langle G \rangle$, where G is a context-free grammar.

1. Create a context-free grammar H such that H accept all strings.
2. If $\langle G, H \rangle \in EQ_{CFG}$, ACCEPT
3. Else REJECT

Machine M is a valid decider for ALL_{CFG} since in step 1, we can easily pick a CFG that accept every string in finite time. Moreover, EQ_{CFG} is decidable so step 2 and 3 will terminate in finite time.
Hence, M is decider for ALL_{CFG} (Contradiction).
Therefore, EQ_{CFG} is undecidable.

5.2 Show that EQ_{CFG} is co-Turing-recognizable

We can design a recognizer for it. M: on input $\langle G_1, G_2 \rangle$ is two CFGs.

1. Repeat the following for i = 1, 2, 3,...
2. with string s_i, if $\langle G_1, s_i \rangle \in A_{CFG} \neq \langle G_2, s_i \rangle \in A_{CFG}$, ACCEPT

Since A_{CFG} is decidable and the set $\sum^*$ is countable, the machine M always terminates and returns ACCEPT if $\langle G_1, G_2 \rangle \in \overline{EQ}_{CFG}$.

5.9

$T = \{\langle M \rangle | \text{M is a TM that accepts } w^R \text{ whenever it accepts } w\}$
Suppose that T is decidable
Firstly, we design the turing machine A such that:

$$L(A) = \begin{cases} \{01, 10\}, \text{if M accept } w \\ \{001\}, \text{otherwise} \end{cases} \tag{1}$$

A: on input string x:

1. if $x \notin \{01, 10, 001\}$ REJECT.
2. if $x \in \{01, 10\}$
   (a) if M accept w ACCEPT
   (b) else REJECT

   if $x = 001$:

(a) if M accept w REJECT

(b) else ACCEPT

So, we can build a decidable for A_{TM}:
R: on input $\langle M, w\rangle$:

1. Run T on $\langle A\rangle$
2. If it accept, ACCEPT
3. Else REJECT

Since T is decidable so R is decider for A_{TM}. However, A_{TM} is undecidable (Contradiction). Therefore, T is undecidable

5.22

($\Rightarrow$) If $A \leq_m A_{TM}$, then A is Turing-recognizable because ATM is Turing recognizable. ($\Leftarrow$) If A is Turing-recognizable, then there exists some TM R that recognizes A. That is, R would receive an input w and accept if w is in A (otherwise R does not accept).
To show that $A \leq_m A_{TM}$, we design a function f that does the following: on input w, writes $\langle R, w\rangle$ on the tape and halts.
It is easy to check that w is in A if and only if $f(w) = \langle R, w\rangle \in A_{TM}$.
Thus, we get a mapping reduction of A to ATM.

5.23

Show that A is decidable iff $A \leq_m 0^*1^*$.
Firstly, we will show that 0^*1^* is decidable by designing a decider for it.
M = on input string x.

1. Design a DFA D that accept 0^*1^*
2. If $\langle D, x\rangle \in A_{DFA}$, ACCEPT.
3. Else REJECT.

($\Rightarrow$) If $A \leq_m 0^*1^*$, then A is decidable because 0^*1^* is decidable.
($\Leftarrow$) If A is decidable, then there exists some TM R that decides A. That is, R would receive an input w and accept if w is in A, reject if w is not in A.
To show $A \leq_m 0^*1^*$, we design a mapping function f that does the following:
On input w:

1. Runs R on w
2. If R accepts, outputs 01.

3. Otherwise, outputs 10.

Hence, we can see that $w \in A \iff f(w) \in 0^*1^*$
Therefore we obtain a mapping reduction of A to 0^*1^*.

7.9

Let $G = (V, E)$ be a graph with a set V of vertices and a set E of edges. We enumerate all triples (u, v, w) with vertices $u, v, w \in V$ and then check whether or not all three edges (u, v), (v, w) and (u, w) exist in E. Enumeration of all triples requires $O(|V|^3)$ time. Checking whether or not all three edges belong to E takes $O(|E|)$ time. Thus, the overall time is $O(|V|^3|E|)$, which is polynomial in the length of the input $\langle G \rangle$. Therefore, $TRIANGLE \in P$

7.11a

Algorithm to check DFA equivalent:
On input $\langle G_1, G_2 \rangle$ are two DFA:

1. Since $G_1 = G_2$, the initial states of them must be equal ($q_0^1 = q_0^2$)
2. Spread the equivalent by transition: if $q_i^1 = q_i^2$ then $q_j^1 = \sigma(q_i^1, a) = \sigma(q_i^2, a) = q_j^2$
3. If there is a conflict in any steps, REJECT.
4. After end of loop, ACCEPT.

The algorithm above need O(N * M) with N is the number of state and M is size of alphabet $\sum$.
Therefore EQ_{DFA} is in P.

7.12

We aim tøshow that the language ISO can be verified in polynomial time. Let the input x be two graphs G and H and let the certificate y be the indices $\{i_1, i_2, ..., i_n\}$. An algorithm A(x, y) verifies ISO by executing the following steps:

- Check if the certificate y is a permutation of $\{1, 2, ..., n\}$. If no, REJECT; else continue.
- Permute the vertices of G as given by the given permutation. Verify that the permuted G is identical to H.

Step 1 takes at most $O(V^2)$ time and step 2 runs in $O(V + E)$ time, therefore the verification algorithm A runs in $O(V^2)$ time.