Final

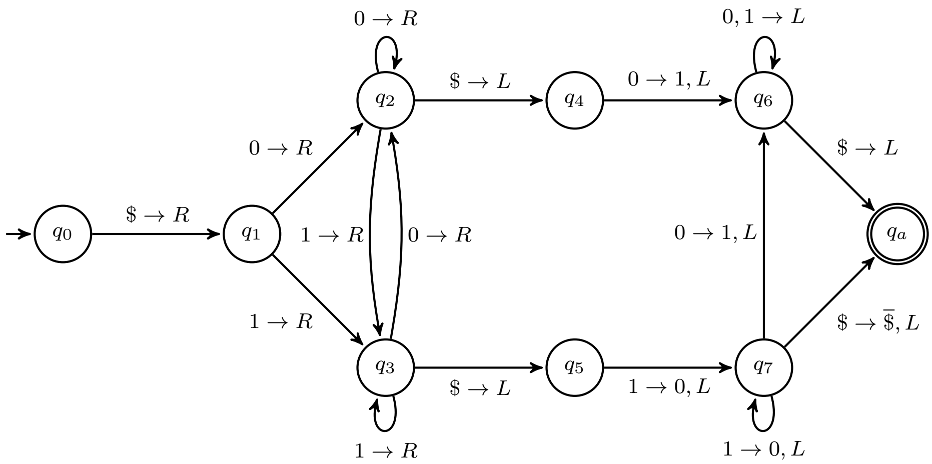
Note

This is a closed-book exam. Each problem accounts for 10 points, unless otherwise marked.

Problems

1. Prove *by induction* that, if G is a CFG in Chomsky normal form, then for any string $w \in L(G)$ of length $n \geq 1$, exactly $2n-1$ steps are required for any derivation of w.

2. Below is a formal description of a (single-tape) Turing machine that computes a function from $\{0, 1, \$, \overline{\$}\}^*$ to $\{0, 1, \$, \overline{\$}\}^*$. The machine is meant to be deterministic; however, for brevity we have omitted transitions that go to a rejecting and terminating state. Explain in words what exactly the machine computes. You only need to consider the cases where the computation successfully terminates at state q_a.



3. Give a formal description of a (single-tape deterministic) Turing machine that decides the language $\{\#1^m\#1^n\#1^{m+n}\# \mid m, n \geq 1\}$.

4. Let $PAL_{\text{DFA}} = \{\langle M \rangle \mid M \text{ is a DFA that accepts some palindrome}\}$. Show that PAL_{DFA} is decidable. (Hint: theorems about CFLs are helpful here.)

5. Let $E_{\text{TM}} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \emptyset\}$. Show that $\overline{E_{\text{TM}}}$, the complement of E_{TM}, is Turing-recognizable.

6. Let $AMBIG_{\text{CFG}} = \{\langle G \rangle \mid G \text{ is an ambiguous CFG}\}$. Show that $AMBIG_{\text{CFG}}$ is undecidable. (Hint: use a reduction from PCP. Given an instance

$$P = \left\{ \left[\frac{t_1}{b_1}\right], \left[\frac{t_2}{b_2}\right], \ldots, \left[\frac{t_k}{b_k}\right] \right\}$$

of PCP, construct a CFG G with the rules:

$$\begin{array}{lcl} S & \to & T \mid B \\ T & \to & t_1Ta_1 \mid \cdots \mid t_kTa_k \mid t_1a_1 \mid \cdots \mid t_ka_k \\ B & \to & b_1Ba_1 \mid \cdots \mid b_kBa_k \mid b_1a_1 \mid \cdots \mid b_ka_k, \end{array}$$

where $a_1, \ldots, a_k$ are new terminal symbols. Prove that this reduction works.)

7. A variable A in CFG G is said to be *necessary* if it appears in every derivation of some string $w \in L(G)$. Let $NECESSARY_{\text{CFG}} = \{\langle G, A \rangle \mid A \text{ is a necessary variable in } G\}$. Prove that $NECESSARY_{\text{CFG}}$ is undecidable.

8. In the proof of the Cook-Levin theorem, which states that *SAT* is NP-complete, we used 2×3 windows of cells to formulate the constraint that the configuration of each row (except the first one) in the $n^k \times n^k$ tableau legally follows the configuration of the preceding row. Suppose the Turing machine being reduced is that in Problem 2 (though it is a function computer rather than a language decider). For each of the following 2×3 windows, indicate whether it is legal or not and briefly explain the reason.

0	0	\$
q_2	0	\$

0	q_3	\$
q_5	0	\$

q_3	\$	0
1	\$	0

\$	1	0
\$	1	q_7

0	1	0
1	1	0

q_7	\$	0
q_a	\$	0

9. In the proof that the *3SAT* problem is polynomially reducible to the *CLIQUE* problem, we convert an arbitrary Boolean expression in 3CNF (input of the *3SAT* problem) to a graph and an integer (input of the *CLIQUE* problem).

   (a) Please illustrate the conversion by drawing the graph and giving the integer that will be obtained from the following boolean expression:

   $$(w + \overline{x} + \overline{y}) \cdot (\overline{w} + y + z) \cdot (\overline{x} + \overline{y} + z) \cdot (x + \overline{y} + \overline{z}).$$

   (b) The original Boolean expression is satisfiable. As a demonstration of why the reduction is correct, please use the obtained result to argue that it is indeed the case.

10. Let $DOUBLE\_SAT = \{\langle \phi \rangle \mid \phi \text{ is a Boolean formula with at least two satisfying assignments}\}$. Prove that $DOUBLE\_SAT$ is NP-complete.

Appendix

- A context-free grammar is in **Chomsky normal form** if every rule is of the form

$$\begin{aligned} A &\rightarrow BC \quad \text{or} \\ A &\rightarrow a \end{aligned}$$

  where a is any terminal and A, B, and C are any variables—except that B and C may not be the start variable. In addition,

$$S \rightarrow \varepsilon$$

  is permitted if S is the start variable.

- $E_{\text{CFG}} = \{\langle G \rangle \mid G \text{ is a CFG and } L(G) = \emptyset\}$ is decidable.

- $ALL_{\text{CFG}} = \{\langle G \rangle \mid G \text{ is a CFG and } L(G) = \Sigma^*\}$ is undecidable.

- Language A is **mapping reducible** (many-one reducible) to language B, written $A \leq_m B$, if there is a computable function $f : \Sigma^* \longrightarrow \Sigma^*$, where for every w,

$$w \in A \Longleftrightarrow f(w) \in B.$$

- If $A \leq_m B$ and A is undecidable, then B is undecidable.

- Language A is **polynomial time mapping reducible** (polynomially reducible) to language B, written $A \leq_p B$, if there is a *polynomial time* computable function $f : \Sigma^* \longrightarrow \Sigma^*$, where for every w,

$$w \in A \Longleftrightarrow f(w) \in B.$$

- $SAT = \{\langle \phi \rangle \mid \phi \text{ is a satisfiable Boolean formula}\}$. SAT is NP-complete (the Cook-Levin theorem).

- $3SAT = \{\langle \phi \rangle \mid \phi \text{ is a satisfiable 3CNF-formula}\}$. (A 3CNF-formula is a CNF-formula where all the clauses have three literals.) $3SAT$ is NP-complete.

- $CLIQUE = \{\langle G, k \rangle \mid G \text{ is an undirected graph with a } k\text{-clique}\}$. (A *clique* in an undirected graph is a subgraph, wherein every two nodes are connected by an edge, and a *k-clique* is a clique that contains k nodes.) $CLIQUE$ is NP-complete.