Name: Rollno:

CSE340: Theory of Computation (Final Exam)

16th November, 2015

Total Number of Pages: 7 Total Points 100

Instructions

1. Read these instructions carefully.
2. Write you name and roll number on all the pages of the answer book.
3. Cheating or resorting to unfair means will be severely penalized.
4. Do not exchange question books or change the seat after obtaining question paper.
5. Using pens (blue/black ink) and not pencils. Do not use red pens for answering.

Question 1. (25 points) Recall the following classes of languages numbered $1 - 8$

(1) Regular (2) Context-free (3) P **(4)** NP **(5)** EXP **(6) Decidable (7) Turing recognizable (8) All languages**

For each of the following languages given below, specify the lowest-numbered class to which the language surely belongs. Give a short description to justify your answer.

For example, for a context-free language L that is not regular, the right number is 2. Similarly, suppose a language L is NP-complete, although it could possibly be in P but since that is not known, therefore the right answer is 4.

(a) Complement of a Turing recognizable language.

> **Solution:** 8
> Need not be Turing recognizable. Example $\overline{A_{TM}}$

(b) $\overline{\mathsf{SAT}}$

> **Solution:** 5
> $\overline{\mathsf{SAT}}$ is in coNP since SAT is in NP. One can give an EXP algorithm by cycling through all truth assignments and checking that the input does not satisfy any one of them.

(c) $A_{TM} \cap \overline{H_{TM}}$

> **Solution:** 1
> Empty set hence regular.

(d) $L_1 = \{a^i b^j c^k \mid i + j = k\}$

Solution: 2
Easy to give a CFG.

(e) $L_2 = \{a^i b^j c^k \mid i \times j = k\}$

Solution: 3
Multiplication can be done in polynomial time.

(f) $L_3 = \{\langle \phi, \tau \rangle \mid \phi$ is a Boolean formula, τ is a truth assignment and $\phi(\tau) = 1\}$

Solution: 3
Evaluating a Boolean formula on a truth assignment can be done in linear time.

(g) $L_4 = \{\langle M \rangle \mid M$ is a TM and $L(M)$ is context-free$\}$

Solution: 8
Can be shown that $\overline{A_{TM}} \leq_m L_4$ (similar to REG_{TM}).

(h) A language in $\mathsf{NPSPACE}$

Solution: 5
$\mathsf{NSPACE}(f(n)) \subseteq \mathsf{TIME}(2^{O(f(n))})$.

(i) A language in $\mathsf{SPACE}(\log^2 n)$

Solution: 5
Similar to earlier part.

(j) $\mathsf{Clique}_{100} = \{\langle G, k \rangle \mid G$ is a graph with at most 100 vertices and G has a clique of size at least $k, k \leq 100\}$

Solution: 1
Finite language hence regular.

Question 2. (4 points) Let A be a regular language. Using closure properties only show that the language

$$L = \{xy \mid x \in A \text{ and } y \notin A\}$$

is also regular. Do not attempt to construct an automaton or a regular expression for L.

Solution: Since A is regular therefore $\overline{A}$ is also regular. By definition, $L = A \cdot \overline{A}$. Therefore L is regular.

Question 3. (10 points) Give a CFG for the following language

$$L = \{a^i b^j c^k d^l \mid i + k = j + l\}.$$

Give a short explanation of the variables that you use in your CFG.

Solution: CFG for L. S is the start variable.

$$\begin{array}{rcl} S & \longrightarrow & S_1 \mid S_2 \\ S_1 & \longrightarrow & aS_1d \mid T_1 \\ T_1 & \longrightarrow & UV \\ S_2 & \longrightarrow & UT_2V \\ T_2 & \longrightarrow & bT_2c \mid \epsilon \\ U & \longrightarrow & aUb \mid \epsilon \\ V & \longrightarrow & cVd \mid \epsilon \end{array}$$

We first divide into two cases, $i \geq j$ and $i \leq j$. If $i \geq j$, a string in L can be written as $a^n a^j b^j c^k d^k d^n$. On the other hand if $i \leq j$, a string in L can be written as $a^i b^i b^n c^n c^l d^l$. We use this observation to construct the CFG.

- S_1: Generates all strings in L such that $i \geq j$.
- T_1: Generates strings of the form $a^n b^n c^m d^m$.
- S_2: Generates all strings in L such that $i \leq j$.
- T_2: Generates strings of the form $b^n c^n$.
- U: Generates strings of the form $a^n b^n$.
- V: Generates strings of the form $c^n d^n$.

Question 4. (10 points) Let

$$L = \{\langle D \rangle \mid D \text{ is a DFA and } D \text{ accepts } \mathrm{rev}(w) \text{ if and only if } D \text{ accepts } w\}.$$

Show that L is decidable.

Solution: We have seen in class that if a language L is regular then $\mathrm{rev}(L)$ is also regular. We use this fact in our algorithm.

Algorithm

Input: A DFA D.

1. Construct a DFA D' that accepts the reverse language of D. That is, $L(D') = \mathrm{rev}(L(D))$.

2. Using the algorithm for EQ_{DFA} check if $L(D) = L(D')$ and accept if and only if this algorithm accepts.

Observe that

$$\langle D \rangle \in L \iff \forall w, (w \in L(D) \iff \text{rev}(w) \in L(D)) \iff L(D) = \text{rev}(L(D)).$$

Question 5. (6 points) Show that the following language is undecidable

$$L = \{\langle M_1, M_2 \rangle \mid M_1, M_2 \text{ are two TMs and } L(M_1) = \overline{L(M_2)}\}.$$

Solution: Consider the undecidable language

$$E_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \emptyset\}.$$

Claim 1. $E_{TM} \leq_m L$.

We will construct a computable function f that takes as input $\langle M \rangle$ and produces an output $\langle M_1, M_2 \rangle$ such that $L(M) = \emptyset \iff L(M_1) = \overline{L(M_2)}$.

The reduction function f

Input: $\langle M \rangle$

1. Set $M_1 := M$.
2. Set M_2 to be a TM that accepts all strings ($L(M_2) = \Sigma^*$).

Output: $\langle M_1, M_2 \rangle$

Note that $\overline{L(M_2)} = \emptyset$.

Proof of correctness

Now,

$$L(M) = \emptyset \iff L(M_1) = \emptyset \iff L(M_1) = \overline{L(M_2)}$$

Therefore, $E_{TM} \leq_m L$. This proves that L is undecidable.

Question 6. A state q in a Turing machine is said to be a *useless state* if the Turing machine does not enter q on any input. Consider the language

$$L = \{\langle M, q \rangle \mid M \text{ is a TM and } q \text{ is a useless state in } M\}.$$

(a) (10 points) Show that L is undecidable.

Solution: Consider the undecidable language

$$E_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \emptyset\}.$$

Claim 2. $E_{TM} \leq_m L$.

We will construct a computable function f that takes as input $\langle M \rangle$ and produces an output $\langle M', q \rangle$ such that $L(M) = \emptyset \iff q$ is a useless state in M'.

The reduction function f

Input: $\langle M \rangle$

1. Set $M' := M$.
2. Set q to be the accept state of M'.

Output: $\langle M', q \rangle$

Proof of correctness

Now,

$$L(M) = \emptyset \iff L(M') = \emptyset \iff \text{the accept state of } M' \text{ is a useless state}$$

Therefore, $E_{TM} \leq_m L$. This proves that L is undecidable.

(b) (5 points) Is L Turing recognizable, co-Turing recognizable or neither? Prove your answer.

Solution: L is co-Turing recognizable.

We will show that the language

$$\overline{L} = \{\langle M, q \rangle \mid M \text{ is a TM and } q \text{ is not a useless state in } M\}$$

is Turing recognizable.

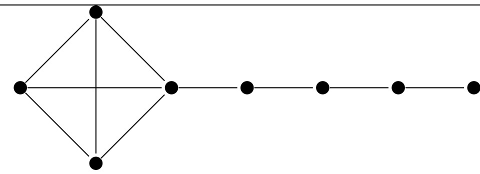
Algorithm

1. Iterate through all strings x and check if M enters the state q on input x. If so, then ACCEPT.

Since the computation can go on forever on an input without entering the state q, we use the diagonal method of traversing through all strings.

Now if M enters q on some input x then it must do so in some finite number of steps. Hence our algorithm would always accept in such cases.

Question 7. (15 points) A *kite* is a graph consisting of an even number of vertices, say $2k$, in which k of the vertices form a clique and the remaining k vertices are connected in a "tail" that consists of a path joined to one of the vertices of the clique.



The above figure shows a kite of size 8. We define

$$\mathsf{KITE} = \{\langle G, k\rangle \mid G \text{ has a kite of size } 2k \text{ as a subgraph}, k \geq 3\}.$$

Show that KITE is NP-complete.

Solution:

1. $\mathsf{KITE} \in \mathsf{NP}$.

   Certificate

   A sequence of vertices $v_1, \ldots, v_m$.

   Verifier's Algorithm

   Input: $\langle G, k, v_1, \ldots, v_m\rangle$

   (a) Check if $m = 2k$. If not then REJECT.
   (b) Check if $v_1, \ldots, v_{m/2}$ forms a clique. If not then REJECT.
   (c) Check if $v_{m/2+1}, \ldots, v_m$ forms a path. If so then ACCEPT else REJECT.

2. Choosing a suitable NP-complete problem. We will show that

$$\mathsf{Clique} \leq_p \mathsf{KITE}.$$

3. The reduction. Let $\langle G = (V, E), k\rangle$ be an instance of Clique. We will construct an instance of KITE, $\langle G' = (V', E'), k'\rangle$ by adding a path of length k to every vertex in G. Formally,

$$\begin{aligned} V' &= V \bigcup_{x \in V} \{v_{x_i} \mid 1 \leq i \leq k\} \\ E' &= E \bigcup_{x \in V} \left(\{(x, v_{x_1})\} \cup \{(v_{x_i}, v_{x_{i+1}}) \mid 1 \leq i \leq k-1\}\right) \end{aligned}$$

   Set $k' = k$.

4. The construction of G' can be achieved in linear time from the graph G.

5. Proof of correctness.

   If G has a clique of size k then the same set of vertices forms a clique in G'. Now since there is a path of length k attached to every vertex in G' therefore there is a kite of size $2k$ in G'.

   If G' has a kite of size $2k$, then there is a clique of size k in G as the additional vertices in G' cannot form a clique.

Question 8. (15 points) Let

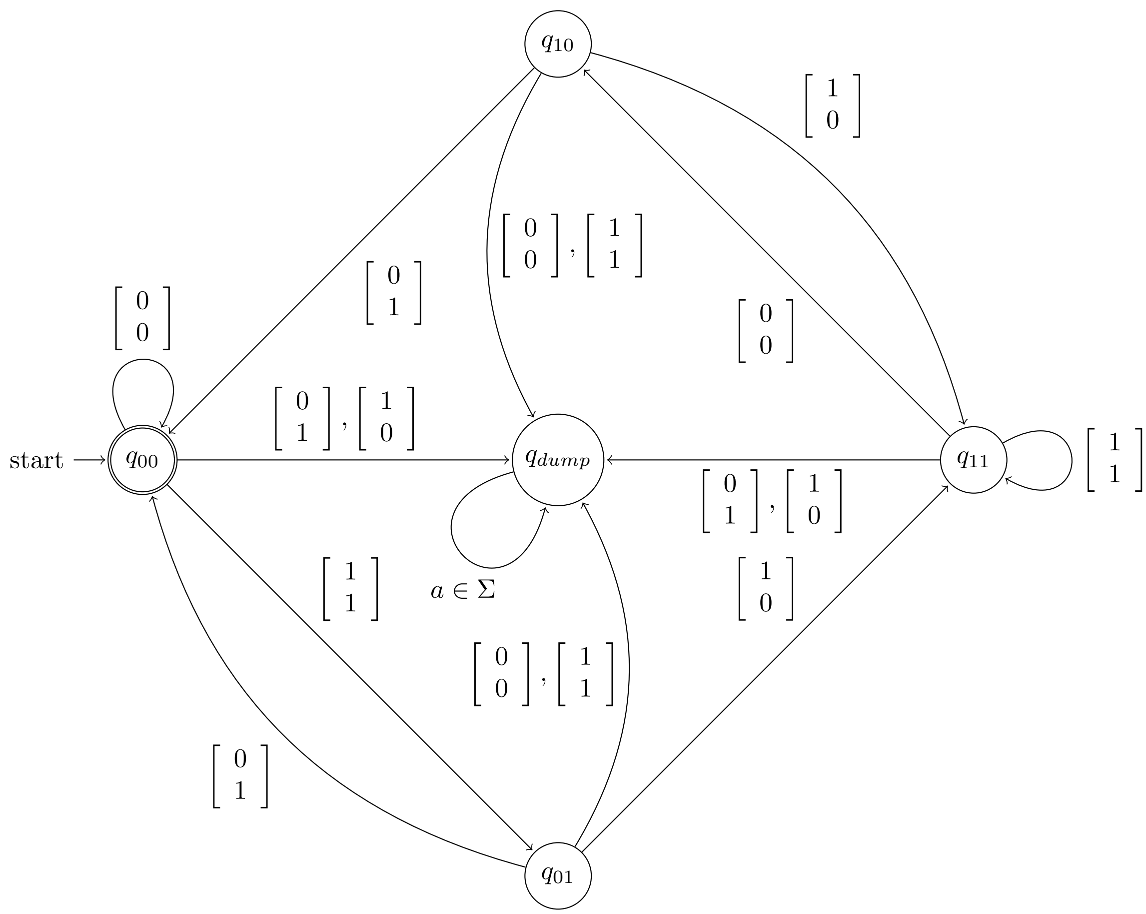
$$\Sigma = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}.$$

A string over Σ gives two rows of 0's and 1's. By considering each row as a binary number define the language

$$L = \{w \in \Sigma^* \mid \text{the bottom row of } w \text{ is three times the top row}\}.$$

Construct a DFA for $\text{rev}(L)$ having at most 5 states.

Solution: Observe that given a number n in binary, $2n$ is n concatenated with a 0 at the end and $3n = n + 2n$. We use this fact to construct a DFA that remembers the sum bit and the carry bit in each step.



DFA for $\text{rev}(L)$