COMS 3261, Computer Science Theory (Fall 2023): Assignment 3
Due on Gradescope 11:59pm, Monday Nov 20, 2023

Instructions

- The total number of points is 55 and there are two pages. Submit your solutions in pdf format. Late homeworks will **not** be accepted.
- You can discuss with TAs, the prof, and other students, but please acknowledge them at the beginning of each problem. All solutions must be written in your own words.
- You should be able to solve questions 1-4a already; the material required to solve questions 4b and 5 will be covered Nov 8 and Nov 13.

Problems

1. (10 points) Give a **formal description** of a one or two tape input-output Turing machine that takes a string $w \in \{0,1\}^*$ as input and halts with w^R (the reverse of the string w) on the first tape. For example, if the tape initially contains the input "11001" (followed by blanks), then after halting the tape should be contain '10011' (followed by blanks). Include a brief high level description of your TM.

   High level description:

   (a) First we replace the leftmost symbol with 'a' if it is '0' and 'b' if it is '1', to indicate the left end of the tape. Then, we copy the input string, which is on tape 1, to tape 2 from left to right. Once we hit a blank, we move to step two.

   (b) Move tape head 2 once to the left, and move tape head 1 to the left until we reach the symbol 'a' or 'b'. Write the value pointed to by head 2 (currently at end of the input string) to the cell pointed to by head 1. Then move head 2 one cell to left and head 1 one cell to the right, and loop.

   (c) Finally, once head 2 points to 'a' we write 0 on the cell pointed to by head 1 and halt; otherwise if head 2 points to 'b' then we write 1 on the cell pointed to by head 1 and halt. At this point the reversed string is written on tape 1.

   Formal definition:

   We define the input-output Turing Machine M as follows:

   $$M = (Q, \Gamma, \Sigma, \delta, q_0, F)$$

   where:

   - $Q = \{q_0, q_{\text{mark}}, q_{\text{move}}, q_{\text{reverse}}, q_f\}$ is the set of states.
   - $\Gamma = \{0, 1, a, b, \sqcup\}$ is the tape alphabet.
   - $\Sigma = \{0, 1\}$ is the set of input symbols.

- δ is the transition function, defined as:

$$\begin{aligned}
\delta(q_0,(0,\sqcup)) &= (q_{\text{mark}},(a,a),(R,R))\\
\delta(q_0,(1,\sqcup)) &= (q_{\text{mark}},(b,b),(R,R))\\
\delta(q_{\text{mark}},(0,x)) &= (q_{\text{mark}},(0,0),(R,R)) \quad \text{for any } x\in\{0,1\}\\
\delta(q_{\text{mark}},(1,x)) &= (q_{\text{mark}},(1,1),(R,R)) \quad \text{for any } x\in\{0,1\}\\
\delta(q_{\text{mark}},(\sqcup,x)) &= (q_{\text{move}},(\sqcup,x),(L,L)) \quad \text{for any } x\in\{0,1\}\\
\delta(q_{\text{move}},(x,y)) &= (q_{\text{move}},(x,y),(L,S)) \quad \text{for any } x,y\in\{0,1\}\\
\delta(q_{\text{move}},(a,y)) &= (q_{\text{reverse}},(a,y),(S,S)) \quad \text{for any } y\in\{0,1\}\\
\delta(q_{\text{move}},(b,y)) &= (q_{\text{reverse}},(b,y),(S,S)) \quad \text{for any } y\in\{0,1\}\\
\delta(q_{\text{reverse}},(x,y)) &= (q_{\text{reverse}},(y,x),(R,L)) \quad \text{for any } x,y\in\{0,1\}\\
\delta(q_{\text{reverse}},(x,a)) &= (q_f,(0,0),(S,S)) \quad \text{for any } x\\
\delta(q_{\text{reverse}},(x,b)) &= (q_f,(1,1),(S,S)) \quad \text{for any } x
\end{aligned}$$

- q_0 is the initial state.
- $F=\{q_f\}$ is the set of final states.

2. (10 points) Prove that for every infinite set S, the following are equivalent:

   (i) There exists a function $g:\mathbb{N}\to S$ that is onto (i.e., g is surjective).

   (ii) There exists a function $f:S\to\mathbb{N}$ that is one-to-one (e.g., f is injective).

   Let S be an infinite set. We first prove (i) implies (ii). Let $g:\mathbb{N}\to S$ be an onto function. We want to show that there exists a 1-to-1 function $f:S\to\mathbb{N}$. For each $s\in S$, let $f(s)$ be equal to the minimum natural number $x\in\mathbb{N}$ such that $g(x)=S$. First note that f is well-defined since g is onto – that is, for every s g maps at least one element x in $\mathbb{N}$ to s. Also f is one-to-one since g is a function – that is, since g maps each $x\in\mathbb{N}$ to exactly one $s\in S$, it follows that for each $x\in\mathbb{N}$, there is at most one $s\in S$ such that $f(s)=x$.

   To prove (ii) implies (i), we want to prove that if $f:S\to\mathbb{N}$ is one-to-one, then there exists $g:\mathbb{N}\to S$ that is onto. For each $x\in\mathbb{N}$, we define $g(x)$ to be $f^{-1}(x)$ if f maps something to x, and otherwise let $g(x)=s_0$ where s_0 is some fixed element in S. (Note that S is infinite so s_0 exists.) First, g is well defined: if x is in the range of f then there is a unique inverse (since f is one-to-one) and if x is not in the range of f then we map x to s_0. Secondly, g is onto: since f is a function, f maps every $s\in S$ to exactly one natural number and therefore for every $s\in S$, g maps some natural number to s.

3. (10 points) Consider the language L_{DFA} which accepts an input w if w is an encoding of a DFA A, where A accepts at least one input $x\in\{0,1\}^*$.

   a. Prove that L_{DFA} is decidable by giving a **high-level description** of a TM that always halts and that accepts exactly the strings in L_{DFA}.

   We give a high level description of an algorithm, A for deciding L_{DFA}. The input to A is the encoding of a DFA, $M=(Q,\Sigma,q_0,F,\delta)$. The basic idea is to see if there is a path from q_0 to some final state in F in the state transition graph associated with M.

– First decode the encoding of M in order to create the state transition graph associated with M. The vertices correspond to the states $q_i \in Q$. For every state $q_i \in Q$ there are $|\Sigma|$ edges, one for each symbol $a \in \Sigma$.
– Initially Visited $= \{q_0\}$.
– Repeat for $|Q|$ steps: For each $q_i \in$ Visited, for all vertices q_j that are neighbors of q_i, add q_j to Visited (if it is not already in the set).
– If Visited contains some $q_i \in F$, halt and accept; otherwise halt and reject.

Proof of correctness. First our algorithm terminate on every input. Second we want to show that M accepts at least one input if and only if A accepts M. The main idea is that if there is a directed path from q_0 to some final state $q \in F$, then the string corresponding to this path will be accepted by M. Secondly, it suffices to consider paths of length at most $|Q|$ since there are only $|Q|$ states, so there is a path from q_0 to some state $q \in F$ if and only if there is a path of length at most $|Q|$ from q_0 to q. Therefore, if there is no path of length at most $|Q|$ from q_0 to an accept state, then M does not accept any inputs.

b. In one or two sentences, explain what goes wrong with your algorithm if you were to apply similar ideas to try to prove that L_{TM} is decidable (where L_{TM} accepts the set of encoding of TMs that accept at least one input).

Since TMs have unlimited memory, the set of possible distinct configurations that a TM can be in on an input can be infinite. For example, consider a TM that on input x computes the number π. Since π is irrational it consists of an infinite sequence of decimals, so the contents of the tape at each time step is distinct, and the TM runs forever. Therefore it reaches an infinite sequene of configurations. DFAs on the other hand cannot write, so the set of configurations is just the set of possible states, which is finite. For this reason it is possible to figure out in a finite amount of time whether a final configuration is reachable for a given DFA, but this argument can't work for a TM since its set of configurations can be potentially infinite.

4. (15 points) Let L_{pair} be the set of all encodings of Turing machines $<M>$ such that there is a pair of consecutive binary numbers that are both accepted by M. (For example, if a Turing machine M accepts both 100 and 101 then $<M>$ is in L_{pair}.)

a. Prove that L_{pair} is recognizable (r.e.) by giving a **high-level description** of a Turing Machine that accepts L_{pair}.

This is similar to the description of a TM accepting Nonempty which contains the strings $< M >$ such that $\mathcal{L}(M)$ is nonempty. For that question we gave the following algorithm, A for recognizing Nonempty on input $< M >$: For $i = 1, 2, \ldots$: Simulate M on each of the first i inputs, $w_1, \ldots, w_i$ for i steps each. If any of these simulations halts and accepts, the A halts and accepts M.
For this question we give a modification, A' of the above algorithm on input $< M >$: For $i = 1, 2, \ldots$: Simulate M on each of the first i inputs, $w_1, \ldots, w_i$ for i steps each. If for some $j < i$ M accepts both w_j and w_{j+1}, then halt and accept.
The argument that A' accepts exactly L_{pair} is similar to that given in the lecture notes for A (and in the book). (Details omitted here.)

b. Prove that L_{pair} is not decidable (not recursive).

The mapping reduction given in question 5 below also works here to show that L_{pair} is not recursive.

5. (10 points) Let $L_{\text{Prime}} = \{<M> \mid \text{the number of strings accepted by } M \text{ is prime}\}$. Classify this language as either (i) decidable, (ii) recognizable but not decidable, or (iii) not recognizable. Prove your answer. You may give a **high-level description** of any TM programs used in your proof. (You should not use Rice's theorem.)

We will give a mapping reduction, f, from A_{TM} to $\overline{L_{\text{Prime}}}$. Note that this is also a mapping reduction from $\overline{A_{TM}}$ to L_{Prime}, and therefore since $\overline{A_{TM}}$ is not r.e., this implies that L_{Prime} is also not r.e.

f will map $< M, w >$ to $< M' >$ where M' is the following Turing Machine. M' on input x: If $x \in \{0, 00, 000\}$ then immediately halt and accept x. Otherwise M' simulates M on input w. If M halts and accepts w, then M' halts and accepts x. Otherwise halt and reject.

f is computable (since given $< M, x >$, the encoding of M' can be computed). Next we will argue that $< M, w > \in A_{TM}$ if and only if $f(< M, w >) \in \overline{L_{\text{Prime}}}$. First let $< M, w > \in A_{TM}$. Then M' will accept any input, and therefore $|\mathcal{L}(M')| = \infty$ which is not a prime number. Now assume that $< M, w > \notin A_{TM}$. Then M' will only accept strings $0, 00, 000$, so $|\mathcal{L}(M')| = 3$ which is prime.