15-453 Formal Languages, Automata, and Computation

Practice Midterm II Solutions

Problem 1 (50 pts)

a. (10 pts) Carefully state Rice's Theorem.

Solution: Let $P : \{\langle M\rangle \mid M \text{ is a TM}\} \rightarrow \{true, false\}$ be a "property". If M_1 and M_2 are Turing machines and the following hold:

- $P(\langle M_1\rangle) = P(\langle M_2\rangle)$ iff $\mathcal{L}(M_1) = \mathcal{L}(M_2)$.
- $\exists x_0, x_1 \in \{\langle M\rangle \mid M \text{ is a TM}\}.P(x_0) = true$ and $P(x_1) = false$.

then P is undecidable.

For the remaining questions, indicate if the language is decidable or undecidable and prove your result in each case. Use Rice's Theorem whenever applicable. If you give a correct alternative proof, you will receive partial credit

b. (10 pts) $\{\langle M, w\rangle \mid M \text{ accepts } w \in \{0,1\}^* \text{ and every prefix of } w\}$

Solution: This is *undecidable*; we can reduce from A_{TM}. The reduction function takes $\langle M, w\rangle$ into $\langle N, w\rangle$ where N is a Turing machine that on input x, does the following:

1. if x is a strict prefix of w then accept
2. otherwise, run M on x.

This map is computable, and $\langle M, w\rangle \in A_{TM}$ iff

$$\langle N, w\rangle \in \{\langle M, w\rangle \mid M \text{ accepts } w \in \{0,1\}^* \text{ and every prefix of } w\}$$

c. (10 pts) $\{\langle M\rangle \mid M \text{ accepts all words } w \in \{0,1\}^* \text{ of even length}\}$

Solution: This is *undecidable* by Rice's theorem. Both conditions are satisfied:

- The first condition is obviously satisfied.
- For the second condition we can choose $x_0 = \langle M_0\rangle$ where M_0 accepts iff the input tape is empty, and $x_1 = \langle M_1\rangle$ where M_1 accepts iff the input tape contains only one letter.

d. (10 pts) $\{\langle M\rangle \mid M$ decides the H_{TM}, the halting problem for Turing machines$\}$

Solution: This language is *decidable.* In fact, this is the empty language, because no machine M can decide H_{TM}.

e. (10 pts) $\{\langle M, w\rangle \mid M$, starting from $q_0 w$, overwrites a blank with a different symbol$\}$

Solution: This language is *decidable.* Let us say that M has q states. If M writes something over a space, it has to do it on top of one of the first q space symbols after the end of the input word. Otherwise, the machine must have reached a state twice while transiting to the right, so it is stuck in an infinite loop, and it will never overwrite any space symbol.

The number of configurations with the head on the top of the first $|w| + q$ tape cells is finite, being bounded by $B = q(|w| + q)^{|\Sigma|+1}$.

An algorithm to decide this language is as follows:

(a) Run M on w and reject after B steps.

(b) If M overwrites a space symbol, accept.

Problem 2 (50 pts)

A co-nondeterministic Turing machine (Co-NTM) is a nondeterministic Turing machine which *accepts* an input w if all computations starting from the initial configuration $q_0 w$ reach the accepting state. It *rejects* w if all computations starting at $q_0 w$ terminate and it does not accept w.

A Co-NTM C *decides* a language L if it accepts every word $w \in L$ and rejects every $w \notin L$. We call a language L *co-decidable* if there is a co-NTM deciding L. In the questions below you may use the fact that Turing machines can simulate nondeterministic Turing machines.

a. (25 pts) Prove that if L is decidable then it is co-decidable.

   Solution: A Turing machine M that decides L is a particular case of a co-nondeterministic Turing machine, so L is co-decidable.

b. (25 pts) Prove that if L is co-decidable then it is decidable.

Solution: Let N be a Co-NTM that co-decides L. We can create a Turing machine M that decides L. It simulates N in a way described in class (by "iterative deepening" on the computation tree of N). When all of the computation paths of N are exhausted, then M accepts iff all of the paths lead to acceptance, and rejects otherwise. Now, since N is co-decider, all of its computations eventually finish, so M will be able to search through them all, and accept iff N accepts.

Problem 3 (30 pts)

A Co-NTM C *recognizes* a language L if it accepts all words $w \in L$ and does not accept words $w \notin L$.

What can you say about the set of languages recognized by a Co-NTM?

Solution: The set of Co-NTM recognizable languages is equal to the set of recognizable languages. Indeed, if a language is recognized by a TM M, then it is Co-NTM recognized by the same machine too. Conversely, if N is Co-NTM recognizer of L, then the machine M described in the solution of Problem 2.b is a recognizer for L.