

Homework 3

Oct 28th, 2008

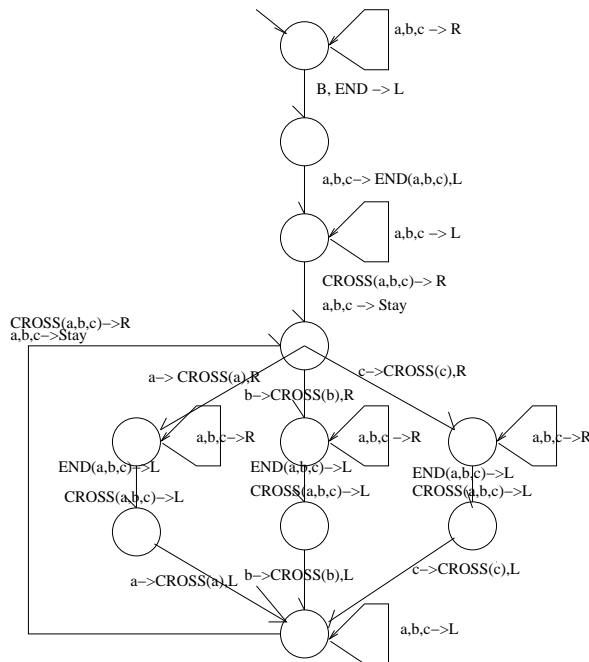
Problem 1: Give pseudocode for the Turing machine that accepts the following language: $L = \{xx^rw \mid x \in \Sigma^+, w \in \Sigma^+\}$. Here $\Sigma = \{a, b, c\}$.

Then draw the diagram of the Turing machine.

Solution 1: The pseudocode is as following:

Algorithm 1 TM_1

- 1: Mark the last unmarked character on the tape as *END*
 - 2: **if** There is no unmarked characters **then**
 - 3: Reject w
 - 4: Compare the first character without any mark with the last character before a marked character
 - 5: **if** Matching **then**
 - 6: Marking the two characters as *CROSS*
 - 7: Go to (2)
 - 8: **if** There is no unmarked characters **then**
 - 9: Accept w
 - 10: **else**
 - 11: Clear all the *CROSS* marks.
 - 12: Go to (1)
-



Problem 2: Without directly using Rice Theorem, prove that the following language is not decidable.
 $L = \{\langle M \rangle \mid M \text{ is a Turing machine and the size of } L(M) \text{ can be divided by } 3\}$.

Proof: Assume that there exists a TM R decides L . Now we construct a TM S decides $A_{TM} = \{\langle M, w \rangle \mid$

M accepts w . $S = \text{input } \langle M, w \rangle$

1. construct a TM M_2
input: a TM $\langle T \rangle$
if size of $L(T)$ can not be divided by 3, accept.
else run M on w , if M accepts w , accept
2. Run R on $\langle M_2 \rangle$ if R accepts it, then accept, else reject it.

Clearly, S is a decider for A_{TM} . However, A_{TM} is not recursive as we have already know. Thus there is a contradiction. So L is not decidable. ■

Problem 3: Which of the following languages is undecidable? Give proofs of your answer. You can use Rice Theorem whenever you can to say a language is not decidable.

1. $L = \{ \langle M \rangle \mid M \text{ is a Turing machine and the size of } L(M) \text{ can be divided by } 3 \}$.
2. $L = \{ \langle M \rangle \mid M \text{ is a Turing machine and } M \text{ has at most } 75 \text{ states} \}$.
3. $L = \{ \langle M_1, M_2 \rangle \mid M_1 \text{ and } M_2 \text{ are Turing machines and } L(M_1) \cap L(M_2) = \emptyset \}$.
4. $L = \{ \langle M \rangle \mid M \text{ is a Turing machine and } L(M) \text{ is recursive} \}$.

Solution 3:

1. There are three languages can not be decided. They are:

- (a) $L = \{ \langle M \rangle \mid M \text{ is a Turing machine and the size of } L(M) \text{ can be divided by } 3 \}$
- (b) $L = \{ \langle M_1, M_2 \rangle \mid M_1 \text{ and } M_2 \text{ are Turing machines and } L(M_1) \cap L(M_2) = \emptyset \}$.
- (c) $L = \{ \langle M \rangle \mid M \text{ is a Turing machine and } L(M) \text{ is recursive} \}$.

Proof: We use *Rice Theorem* to proof these three languages are not decidable. By *Rice Theorem*, to prove language is not decidable, we only need to prove that the property of that language is non-trivial. For that the property of all these three languages are not trivial, thus they are all undecidable. ■

2. There is only one language can be decided. It is:

- (a) $L = \{ \langle M \rangle \mid M \text{ is a Turing machine and } M \text{ has at most } 75 \text{ states} \}$.

Proof: For that we only need to check that the size of Q is 75 or not, which can be done at most 75 steps. So we can decide this language. ■

Problem 4: Prove that Turing machines as a generator is equivalent to Turing machines as acceptor. Here a TM G as a generator produces a language $L(G)$, if it will print all strings of $L(G)$ on the tape (here the printer will not go backward when printing).

In other words, you have to prove

1. Given a generator G , there is always a TM M such that $L(M) = L(G)$, and
2. Given a TM M , there is always a generator G , such that $L(M) = L(G)$.

Proof:

1. In this part, we proof that given a generator G , there is always a TM M such that $L(M) = L(G)$.

Given a generator G , we construct the TM M in the following way:

TM $M = \text{input } w$:

- (a) Lineary compare w with all the strings generated by G .
- (b) If matching, then accept.

(c) If not matching, keep searching.

By this algorithm, we can construct the TM M .

2. In this part, we prove that given a TM M , there is always a generator G , such that $L(M) = L(G)$.

We construct the generator G as stated below:

Generator G = input anything will be OK,

(a) When the TM M accepts a string w , G prints it in its printer.

By this method, we can guarantee that $\forall w \mid w$ belongs to M , w will be generated by G .

