COMP3721 Tutorial 9

CSE, HKUST

November 4, 2016

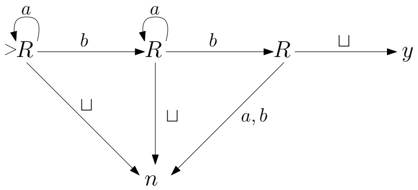
Problem 1

Q1. Give a Turing machine that decides the regular language a^*ba^*b.

Problem 1

Q1. Give a Turing machine that decides the regular language a^*ba^*b.

Solution:



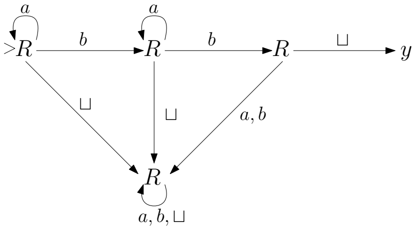
Problem 2

Q2. Give a Turing machine that semi-decides the regular language a^*ba^*b.

Problem 2

Q2. Give a Turing machine that semi-decides the regular language a^*ba^*b.

Solution:



Problem 3

Q3. Prove that a language L is recursive if and only if L and $\bar{L}$ are both recursively enumerable.

Problem 3

Q3. Prove that a language L is recursive if and only if L and $\bar{L}$ are both recursively enumerable.

Only-if part: since L is recursive, so is $\bar{L}$. Since L and $\bar{L}$ are recursive, they are also recursively enumerable.
If part: since L and $\bar{L}$ are recursively enumerable, there are standard Turing machines M_1 and M_2 semi-decide them respectively. Now we use M_1 and M_2 to construct a Turing machine M^* that decides L, which implies that L is recursive. Conceptually, given a string w, M^* passes w to both of M_1 and M_2, and run M_1 and M_2 in a parallel manner. If M_1 halts, then M^* halts and accepts w. If M_2 halts, then M^* halts and rejects w. Note that M^* always halts since exactly one of M_1 and M_2 will halt on any given string.

Problem 4

Q4. Let $L = \{0^k : k \text{ is a Fibonacci number}\}$. Prove that L is recursive.

Problem 4

Q4. Let $L = \{0^k : k \text{ is a Fibonacci number}\}$. Prove that L is recursive.

Idea:

1. We provide a Turing machine that decides L.
2. We use a more flexible multi-tape Turing machine rather than a standard one.
3. We first build Turing machines that achieve simple tasks (here we first build a TM which, given an integer i, generates the i^{th} Fibonacci number), and then combine these simple machines to get a machine that meets our requirement.