# PTC (Fall 2018) – Assignment 3

徐翔哲 161250170

2018 年 12 月 19 日

# Problem 1

Consider the (deterministic) Turing machine $M$ given by

$$M = (\{q_0, q_1, q_2\}, \{a, b\}, \{a, b, B\}, \delta, q_0, B, \{q_2\})$$

which has exactly four transitions defined in it, as described below.

1. $\delta(q_0, a) = (q_0, B, R)$
2. $\delta(q_0, b) = (q_1, B, R)$
3. $\delta(q_1, b) = (q_1, B, R)$
4. $\delta(q_1, B) = (q_2, B, R)$

Please answer the following questions:

a. Specify the execution trace of $M$ on the input string $abb$.

b. Provide a regular expression for the language of the Turing machine.

c. Suppose we added the transition $\delta(q_1, a) = (q_0, B, R)$ to the above machine, provide a regular expression for the language of the resulting Turing machine.

### a

$q_0aab \vdash q_0ab \vdash q_0b \vdash q_1B \vdash q_2B$

### b

$a^*b^+$

### c

$a^*b^+(a^+b^+)^*$

# Problem 2

Please design TM′s to decide following languages:

a. $L_1 = \{1^m \times 1^n = 1^{mn} \mid m, n \in \mathbb{N}^+\}$ (e.g. $11 \times 111 = 111111 \in L_1$, but $1 \times 1 = 11 \notin L_1$)

b. $L_2 = \{ww \mid w \in \{a, b\}^*\}$

**a**

First, let's define a TM $M_1$ which will convert $\times 1^n = 1^{n'}$ to $\times 1^n = 1^{n'-n}$ where $n' \geq n > 0$

$$M_1 = (\{p_0, p_R, p_a, p_{a'}, p_d, p_L, p_f, p_e\}, \{1, 0, \times, =\}, \{1, 0, \times, =, B, a\}, \delta, p_0, B, \{p_e\})$$

At the beginning , the head should at $\times$ Then we will enter $p_a$, which will change the first 1 to a when the head goes right.
$\delta(p_0, \times) = (p_a, \times, R)$
$\delta(p_a, a) = (p_a, a, R)$
$\delta(p_a, 1) = (p_R, a, R)$

The state $p_R$ moves the head to the first blank at the right of the input, and then switches to state $p_d$, which will delete a 1.
$\delta(p_R, X) = (p_R, X, R)$, where $X = \{=, 1\}$
$\delta(p_R, B) = (p_d, B, L)$
$\delta(p_d, 1) = (p_L, B, L)$
The state $p_L$ will moves to $\times$ and then enters $p'_a$
$\delta(p_L, X) = (p_L, X, L)$, where $X = \{1, =, a\}$
$\delta(p_L, \times) = (p_a, \times, R)$

Then the state $p_{a'}$ will act as $p_a$, change a 1 to a, then move to the right and delete a 1 ...... except that $p_{a'}$ will go to the state $p_f$ if it sees no 1 before the =, under which condition $p_a$ would halt without accepting.
$\delta(p_{a'}, a) = (p_{a'}, a, R)$
$\delta(p_{a'}, 1) = (p_R, a, R)$
$\delta(p_{a'}, =) = (p_f, =, L)$

The state $p_f$ will change all the a back to 1.
$\delta(p_f, a) = (p_f, 1, L)$
$\delta(p_f, \times) = (p_e, \times, R)$
Then we construct the TM $M$ to describe the language $L_1$

$$M = (M_1.states \cup \{q_0, q_R, q_{call}, q_L, q_1, q_e, q_{check}, q_f\}, \{1, 0, \times, =\}, \{1, 0, \times, =, B, a\}, M_1.\delta \cup \delta, q_0, B, \{q_f\})$$

The state $q_0$ will remove the first 1, and goes to state $q_R$
$\delta(q_0, 1) = (q_R, B, R)$
The state $q_R$ will move to the $\times$ and then enters $q_{call}$
$\delta(q_R, 1) = (q_R, 1, R)$
$\delta(q_R, \times) = (q_{call}, \times, L)$
The state $q_{call}$ will call the $M_1$
$\delta(q_call, 1) = (p_0, 1, R)$

When the M returns from $M_1$, state $q_L$ will move the head to the blank left to .

# Problem 3

A *useless state* in a Turing machine is one that is never entered on any input string. Consider the problem of determining whether a state in a Turing machine is useless. Formulate this problem as a language and show it is decidable or undecidable. (**Hint**: consider the language $E_{\text{TM}}$)

**Solution.**

# Problem 4

Show that the following questions are decidable:

a. The set $L$ of codes for TM's $M$ such that, when started with the blank tape will eventually write some nonblank symbol on its tape. (**Hint**: If $M$ has $m$ states, consider the first $m$ transitions that it makes)

b. The set $L$ of codes for TM's that never make a move left on any input.

c. The set $L$ of pairs $(M, w)$ such that TM $M$, started with input $w$, never scans any tape cell more than once.

**Proof.**

# Problem 5

If a pushdown automaton has $k$ stacks, we call it $k-$PDA. Clearly, $0-$PDA is NFA, $1-$PDA is PDA, and $1-$PDA is more powerful than $0-$PDA.

1. What is the difference between the express ability of 2-PDA and 1-PDA. Please clarify your argument. Prove the (un)equivalence.

2. How about 3-PDA and 2-PDA.

**Solution.**

# Problem 6

Suppose we have an encoding of context-free grammars using some finite alphabet. Consider the following two languages:

1. $L_1 = \big\{(G, A, B) \mid G$ is a (coded) CFG, $A$ and $B$ are (coded) varibles of $G$, and the sets of terminal strings derived from $A$ and $B$ are the same$\big\}$.

2. $L_2 = \big\{(G_1, G_2) \mid G_1$ and $G_2$ are (coded) CFG$'$s, and $L(G_1) = L(G_2)\big\}$.

Answer the following questions:

a. Show that $L_1$ is polynomial-time reducible to $L_2$.

b. Show that $L_2$ is polynomial-time reducible to $L_1$.

**Proof.**

# Problem 7

As classes of languages, $\mathcal{P}$ and $\mathcal{NP}$ each have certain closure properties. Prove or disprove that $\mathcal{P}$ and $\mathcal{NP}$ are closed under each of the following operations:

a. Union.

b. Concatenation.

c. Complementation.

**Proof.**