Homework 5

Rusu Wu
11694764

1. As we count for $\#a$, $\#b$ and $\#c$, we could define $L' = ((abc)^{x_1}(ab)^{x_2})^{x3}$, in this case, $\#L$ can be expressed as:

$\exists x_1, x_2, x_3.(\#a = (x_1 + x_2) * x_3 \wedge \#b = (x_1 + x_2)x_3 \wedge \#c = x_1 * x_3 \wedge x_1 \geq 0 \wedge x_2 \geq 0 \wedge x_3 \geq 0)$

Let $n_1 = x_1 * x_3, n_2 = x_2 * x_3$, in this case, we could get $L = L' = (abc)^{n_1}(ab)^{n_2}$. Then we have:

$\exists n_1, n_2.(\#a = n_1 + n_2 \wedge \#b = n_1 + n_2 \wedge \#c = n_1 \wedge n_1 \geq 0 \wedge n_2 \geq 0)$

2. Let M be the given PDA M, and we contract a new PDA M' to simulate M as followes.

On input word w that M runs on, M' simulate M faithfully. However, whenever M does "push a", M' will do the same and additionally M' reads an input symbol "push a" from its own input tape. Whenever M does "push b", M' will do the same and additionally M' reads an input symbol "push b" from its own input tape. Similary, whenever M does "pop a", M' will do the same and additionally M' reads an input symbol "pop a" from its own input tape. Whenever M does "pop b", M' will do the same and additionally M' reads an input symbol "pop b" from its own input tape. Finally, when M says yes, M' says yes.

$L_1 = L(M')$

$L_2 = \{w' : 3 * \#(pushb)) - 2 * \#(pusha) \geq 0 \wedge 3 * ((3 * \#(pushb) - 2 * \#(pusha)) \geq \#(pusha) + \#(pushb) - \#(popa) - \#(popb)\}$

$L_1 \cap L_2$ is semilinear, so the result follows. So, it is decidable whether M is stable.

3. A sequence is fair only when there are infinited number of prefixes equaling on ativities a, b,c. In order to prove it is decidable, we can defined a Presburger formula $P = (p, \#_a, \#_b, \#_c, q, \#'_a, \#'_b, \#'_c)$ to represent a transition from one state to another in FA. p and q are the transitions of states and $\#_a, \#_b, \#_c, \#'_a, \#'_b, \#'_c$ are the number of activities of a,b,c.

Hence, we could design an Alg for any run of $P = (p, \#_a, \#_b, \#_c, q, \#'_a, \#'_b, \#'_c)$ from state p to q, $\#_a = \#_b = \#_c$ and aslo $\#'_a = \#'_b = \#'_c$; such that, we could say a finite state process is decidable.

Define L_1 to be all sequence of a, b, c in all runs of a finite state process

$L_2 = \{\#_a = \#_b = \#_c \wedge \#'_a = \#'_b = \#'_c \wedge P \wedge \#a' \geq \#a \wedge \#b' \geq \#b \wedge \#c' \geq \#c\}$, P is the state transition(p to q) of FA or the chain of the Presburger formula.

$L_1 \cap L_2$ is semilinear, and the result follows.

4. All the sequences of $\#a\#b$ and $\#c$ can be defined as Presburger formular in the run from star to the end. Considering t_a, t_b, t_c as the total time of the run. we could have a fomular as follow,

$\exists \#a, \#b\#c, t_a, t_b, t_c.(\#a = \#b \wedge \#b = \#c \wedge |t_a - t_b| \leq 100 \wedge |t_b - t_c| \leq 100 \wedge |t_a - t_c| \leq 100 \wedge \#a < t_a < 2 * \#a \wedge 2 * \#b < t_b < 3 * \#b \wedge 3 * \#c < t_c < 4 * \#c \wedge \#a > 0 \wedge \#b > 0 \wedge \#c > 0)$

5. We have a Presburger formula $P(x_1, ..., x_k)$ and a PDA M.

Since α is a prefix of some words accepted by M, we contract another PDA M' to find the prefix, and run it as follow:

Both M and M' read on an input w; when M read a symble from w, M' will follow and read a symble. After some symbles have been read, M' would guess it is the end of the prefix and stop, while M' keep reading w, and finally finish reading. When M finish reading, M says yes as well as M'. In this case, $L_1 = L(M')$ is the prefix of the language that be accepted by M, so it is comutative .

Also, define $L_2 = (\neg P)$

$L_1 \cap L_2$ is known as semilinear, so it is decidable.

6. It is decidable that configuration (p, v) reach configuration (q, u). Also, for given v and u, whether Presburger formulas P(v) and Q(u) are both ture is deciable.

Hence, we could first find the solution set of the two Presburger formular and then determine whether some of the solutions are also true for VASS from state p to q. Such that we can determine the existent of configuration (p, v) and (q, u).

7. In order to show ADTAs are universal, we can use it to simulate a two counter machine as follow;

PDTAs	:	Two-counter machines
s, x_1++, s'	:	$x++$
s, x_2++, s'	:	$x--$
s, x_3++, s'	:	$y++$
s, x_4++, s'	:	$y--$
$s, x_1 - x_2 = 0, s'$	:	test $x == 0$?
$s, x_3 - x_4 = 0, s'$	:	test $y == 0$?

8. This trick is about how to simulate $x--$ and $y--$ with the resets of x_1 to x_k.