CSC 320 Assignment 2

Oliver Tonnesen
V00885732

February 17, 2019

1

Let $w = 1^{2p}$, where p is the pumping length of L. Let $w = xyz$ where $x = 1^i$, $y = 1^j$, and $z = 1^k$. To satisfy the conditions of the pumping lemma, we have $i + j \leq p$, and so $k = 2p - i - j \geq p$. The pumping lemma then asserts that $xy^l z \in L \; \forall l \geq 0$. So

$$1^i(1^j)^l 1^k \in L \; \forall l \geq 0.$$

So $|w| = i + jl + k$. Let $l = i + k$. Then

$$\begin{aligned}|w| &= i + j(i + k) + k \\ &= i + ij + jk + k \\ &= i(1 + j) + k(j + 1) \\ &= (i + k)(j + 1)\end{aligned}$$

Recall that $k \geq p$ and also note that $|y| = j > 0$, so both $i + k$ and $j + 1$ are larger than 1. Thus, $(i+k)(j+1)$ must be a composite number (in other words, $1^{(i+k)(j+1)} \notin L$) and so L is not regular.

2

Let $s = 1^{2p}$ where p is the pumping length of L. Let $s = uvwxy$ where $u = 1^i$, $v = 1^j$, $w = 1^k$, $x = 1^l$, and $y = 1^m$. To satisfy the conditions of the pumping lemma, we have $j + k + l \leq p$, and so $i + m = 2p - j - k - l \geq p$. The pumping lemma then asserts that $uv^n wx^n y \in L \; \forall n \geq 0$. So

$$1^i(1^j)^n 1^k (1^l)^n 1^m \in L \; \forall n \geq 0.$$

So $|s| = i + jn + k + ln + m$. Let $n = i + k + m$. Then

$$\begin{aligned}|s| &= i + k + m + n(j + l) \\ &= i + k + m + (i + k + m)(j + l) \\ &= (i + k + m)(j + l + 1)\end{aligned}$$

Recall that $i+m \geq p$ and also note that $|vx| = j+l > 0$, so both $i+k+m$ and $j+l+1$ are larger than 1. Thus, $(i+k+m)(j+l+1)$ must be a composite number (in other words, $1^{(i+k+m)(j+l+1)} \notin L$) and so L is not context-free.

3

The CFG for L:

$$\begin{aligned}(\{S\}, \{a, b, \varepsilon, \emptyset, (,), \cup, \circ, *\}, \{S \to a, S \to b, S \to \varepsilon, S \to \emptyset,\\ S \to (S \cup S), S \to (S \circ S), S \to (S^*)\}, S)\end{aligned}$$

Here are the language's productions in a more readable format:

$$\begin{aligned}&S \to a\\ &S \to b\\ &S \to \varepsilon\\ &S \to \emptyset\\ &S \to (S \cup S)\\ &S \to (S \circ S)\\ &S \to (S^*)\end{aligned}$$

4

4.a

G in Chomsky Normal Form:

$$\begin{aligned}(\{S_0, S, A, B, S_A, S_B\}, \{a, b, \varepsilon\}, \{S_0 \to \varepsilon, A \to a, B \to b, S_0 \to AS_A,\\ S \to AS_A, S_A \to SA, S_0 \to BS_B, S \to BS_B,\\ S_B \to SB, S_0 \to AA, S \to BB, S_0 \to BB, S \to AA\}, S_0)\end{aligned}$$

Again, the language's productions in a more readable format:

$$\begin{aligned}
S_0 &\to \varepsilon \\
A &\to a \\
B &\to b \\
S_0 &\to AS_A \\
S &\to AS_A \\
S_A &\to SA \\
S_0 &\to BS_B \\
S &\to BS_B \\
S_B &\to SB \\
S_0 &\to AA \\
S &\to BB \\
S_0 &\to BB \\
S &\to AA
\end{aligned}$$

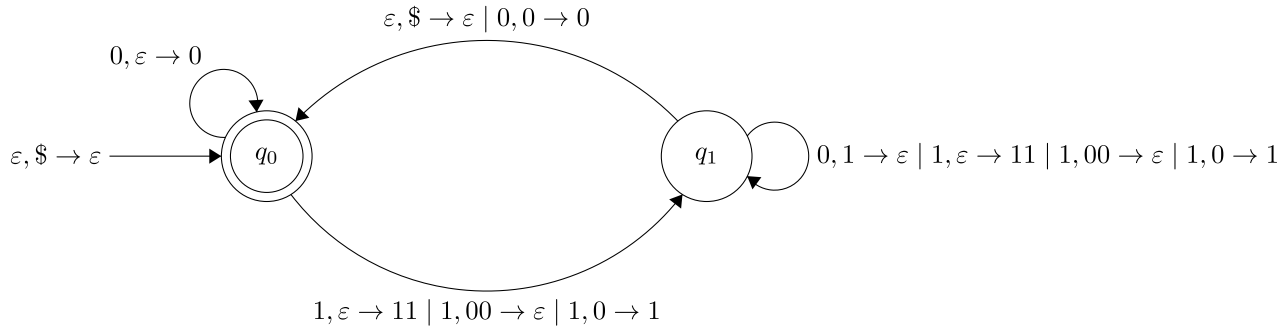
4.b

6	$\{S_0, S\}$					
5	$\emptyset$	$\{S_A\}$				
4	$\emptyset$	$\{S_0, S\}$	$\emptyset$			
3	$\emptyset$	$\{S_B\}$	$\{S_B\}$	$\{S_A\}$		
2	$\emptyset$	$\{S_0, S\}$	$\{S_0, S\}$	$\{S_0, S\}$	$\emptyset$	
1	$\{A\}$	$\{B\}$	$\{B\}$	$\{B\}$	$\{B\}$	$\{A\}$
	a	b	b	b	b	a

There is a start variable in the top rightmost cell of the table, so *abbbba* is in the language.

5

The general idea is to push two 1s onto the stack for each 1 seen, and to pop one 1 for each 0 seen. We accept only when there are no 1s left on the stack when we reach the end of the input.


$0, \varepsilon \to 0$
$\varepsilon, \$ \to \varepsilon \mid 0, 0 \to 0$
$\varepsilon, \$ \to \varepsilon$
q_0
q_1
$0, 1 \to \varepsilon \mid 1, \varepsilon \to 11 \mid 1, 00 \to \varepsilon \mid 1, 0 \to 1$
$1, \varepsilon \to 11 \mid 1, 00 \to \varepsilon \mid 1, 0 \to 1$

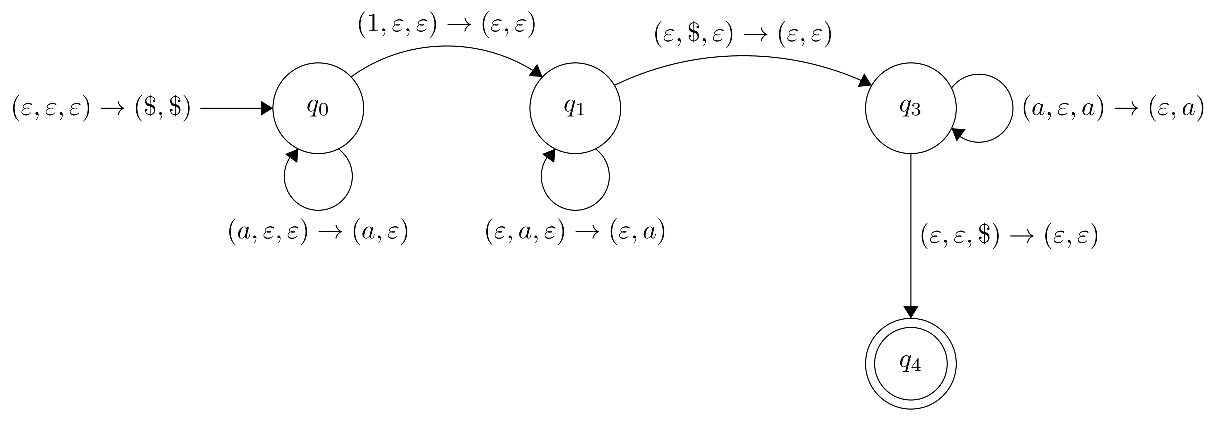
6

We read the first copy of w onto the first stack. At some point, we "guess" that we're in the middle of the string when we read a 1. We now pop all of w off of the first stack and onto the second stack. We now have a copy of w "backwards" (as in the last letter is on the bottom, and the first letter is on the top) on the second stack. We compare each of these to what remains of the string, and accept when we find the bottom of the second stack with no input left.

Here our transitions work as follows:

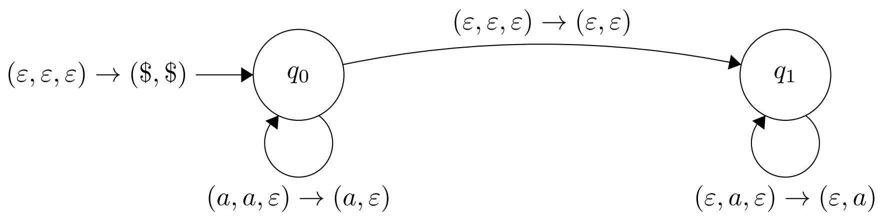
$$(\lambda, a, b) \to (\alpha, \beta)$$

reads λ from the input, pops a from stack 1 and pops b from stack 2, then pushes α onto stack 1 and pushes β onto stack 2.


$(\varepsilon, \varepsilon, \varepsilon) \to (\$, \$)$
q_0
$(1, \varepsilon, \varepsilon) \to (\varepsilon, \varepsilon)$
$(a, \varepsilon, \varepsilon) \to (a, \varepsilon)$
q_1
$(\varepsilon, a, \varepsilon) \to (\varepsilon, a)$
$(\varepsilon, \$, \varepsilon) \to (\varepsilon, \varepsilon)$
q_3
$(a, \varepsilon, a) \to (\varepsilon, a)$
$(\varepsilon, \varepsilon, \$) \to (\varepsilon, \varepsilon)$
q_4

7

7.a


$(\varepsilon, \varepsilon, \varepsilon) \to (\$, \$)$
q_0
$(\varepsilon, \varepsilon, \varepsilon) \to (\varepsilon, \varepsilon)$
q_1
$(a, a, \varepsilon) \to (a, \varepsilon)$
$(\varepsilon, a, \varepsilon) \to (\varepsilon, a)$

7.b

The double pushdown automaton would use the following transition:

$$(q, A, \varepsilon, A) \to \{(p, B, \varepsilon\}$$

First, we move from state q to p. Then (more importantly) we pop A from stack 2 (the portion of the "tape" to the right of the "tape head") and push B onto stack 1, simulating writing to the tape and moving one cell to the right.