Nick G. Toth
Nov. 13, 2020
CIS 420
Assignment V

Exercise 1 (Sipser 3e - Exercise 2.30 a, d)

$a)$ Use the Pumping Lemma to show that the language $L := \{0^n1^n0^n1^n : n \geq 0\}$ is not context free.

Proof:

Suppose L is a context free language. Then by the Pumping Lemma for context free languages, there is a number p with the property that for any string $s \in L$ of length at least p, we can write $s = uvxyz$ satisfying

$i)$ For each $i \geq 0$, $uv^ixy^iz \in L$,
$ii)$ $|vy| > 0$,
$iii)$ $|vxy| \leq p$.

Set $s = 0^p1^p0^p1^p$ and observe that $|s| > p$, so s can be pumped.
Now let $uvxyz = s$ as specified by the Pumping Lemma.

Since $|vxy| \leq p$, we know that $|uz| \geq 3p$.

By the pigeon hole principle, either the first $0^p \subseteq u$ or the second $1^p \subseteq z$.

If the first $0^p \subseteq u$, then v^2xy^2 will begin with 1, or $0^p1 \subseteq u$, and v^2xy^2z will contain a string of $0s$ or $1s$ with length greater than p. But uv^2xy^2z begins with 0^p1, so it cannot be an element of L.

If the second $1^p \subseteq z$, then v^2xy^2 will end with 0, or $01^p \subseteq z$, and uv^2xy^2 will contain a string of $0s$ or $1s$ with length greater than p. But uv^2xy^2z ends with 01^p, so it cannot be an element of L.

Therefore L cannot be a context free language.

∎

$d)$ Use the Pumping Lemma to show that the following language, L, is not context free.

$$\{t_1\#t_2\#\ldots\#t_k : k \geq 2,\ t_i \in \{a,b\}^* \text{ for all i, and } t_m = t_n \text{ for some } m \neq n\}$$

Proof:

Suppose L is a context free language. Then by the Pumping Lemma for context free, languages, there is a number p with the property that for any string $s \in L$ of length at least p, we can write $s = uvxyz$ satisfying

$i)$ For each $i \geq 0$, $uv^ixy^iz \in L$,
$ii)$ $|vy| > 0$,
$iii)$ $|vxy| \leq p$.

So take $s = t_1\#t_2 = t\#t = a^pb^p\#a^pb^p$ and observe that $|s| > p$.
s can be pumped, so let $uvxyz = s$ as specified by the Pumping Lemma.

Note that condition ii tells us that v or y must be non-empty.
Also note that

Case $1)$ Suppose # is a substring of vxy.

If $x = \#$, then $v = t$ or $y = t$, so pumping up to obtain ut^2xy^2z or uv^2xt^2z, and observe that either of these cases will give us a string $t_1\#t_2$ where $t_1 \neq t_2$, so uv^2xy^2z cannot be an element of L.

Otherwise # is a substring of v or y, so clearly uv^0xy^0z will not contain #, and thus uv^0xy^0z cannot be an element of L.

Case $2)$ Suppose vxy is a substring of t.

We know v or y is non empty, so clearly v^2xy^2 contains at least one more a or b than t. But then u or z must contain the remaining string, so uv^2xy^2z will have more as or bs on one side or the other of #. So uv^2xy^2z will not satisfy the palindrome property, and thus $uv^2xy^2z \notin L$.

Since $uv^2xy^2z \notin L$ for each of these cases, L cannot be context free.

∎

Exercise 2 (Sipser 3e - Exercise 2.32)

Let $\Sigma = \{1,2,3,4\}$ and use the Pumping Lemma to show that the language C containing all strings in Σ^* such that the number of 1s equals the number of 2s, and the number of 3s equals the number of 4s is not context free.

Proof:

Suppose C is a context free language. Then by the Pumping Lemma for context free languages, there is a number p with the property that for any string $s \in C$ of length at least p, we can write $s = uvxyz$ satisfying

$i)$ For each $i \geq 0$, $uv^ixy^iz \in C$,
$ii)$ $|vy| > 0$,
$iii)$ $|vxy| \leq p$.

Let $s = 1^p2^p3^p4^p$ and observe that $|s| > p$.
s can be pumped, so let $uvxyz = s$ as specified by the Pumping Lemma.

Observe that vxy can't contain both 1s and 3s or both 2s and 4s since $|vxy| \leq p$, and that 1^p is contained in u or 4^p is contained in z.

If 1^p is contained in u, then $u'v^2xy^2z$ must contain more than p 2s, where u' is obtained by replacing every occurrence of 1 in u with the empty string. So uv^2xy^2z cannot be in C.

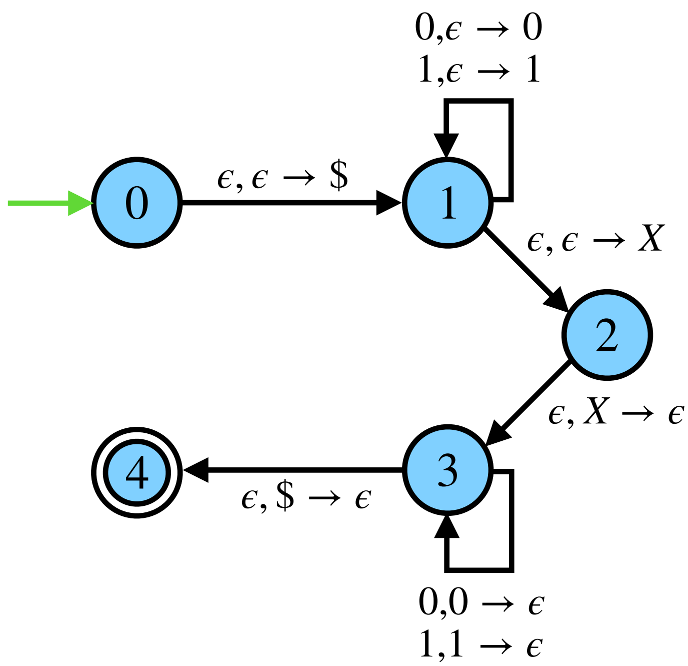
If 4^p is contained in z, then uv^2xy^2z' must contain more than p 3s, where z' is obtained by replacing every occurrence of 4 in z with the empty string. So uv^2xy^2z cannot be in C.

So uv^2xy^2z cannot be an element of C in general, and thus C is not a context free language.

∎

Exercise 3

Convert the following PDA into a CFG.
Note: I didn't save quite enough time to finish this exercise, sadly; it was more challenging than I anticipated. Why did I spend extra time to create the PDA that you already gave us? That's an excellent question.



Step 1
Not complete

$A_{13} \rightarrow 0A_{13}0$
$A_{13} \rightarrow 1A_{13}1$

Step 2
Not complete
$A_{04} \rightarrow A_{13}$
$A_{13} \rightarrow A_{22}$
$A_{11} \rightarrow A_{11}$
$A_{33} \rightarrow A_{33}$
$A_{01} \rightarrow A_{0x}A_{y1}$
$A_{02} \rightarrow A_{0x}A_{y2}$
$A_{03} \rightarrow A_{0x}A_{y3}$
$A_{04} \rightarrow A_{0x}A_{y4}$

$A_{12} \to A_{1x}A_{y2}$
$A_{13} \to A_{1x}A_{y3}$
$A_{14} \to A_{1x}A_{y4}$
$A_{23} \to A_{2x}A_{y3}$
$A_{24} \to A_{2x}A_{y4}$
$A_{34} \to A_{3x}A_{y4}$

Step 3: $A_{00} \to \epsilon, A_{11} \to \epsilon, A_{22} \to \epsilon, A_{33} \to \epsilon, A_{44} \to \epsilon.$

Exercise 4 (Sipser 3e - Exercise 3.2 b, c)

This exercise concerns TM M_1, whose description and state diagram appear in Example 3.9. In each of the parts, give the sequence of configurations that M_1 enters when started on the indicated input string.

$b)$ 1#1.

Start configuration: $q_1 1\#1$
Configuration 2: $xq_3\#1$
Configuration 3: $x\#q_5 1$
Configuration 4: $xq_6\#x$
Configuration 5: $q_7 x\#x$
Configuration 6: $xq_1\#x$
Configuration 7: $x\#q_8 x$
Configuration 8: $x\#xq_8$
Accepting Configuration: $x\#xq_{accept}$

$c)$ 1##1.

Start configuration: $q_1 1\#\#1$
Configuration 2: $xq_3\#\#1$
Configuration 3: $x\#q_5\#1$
Rejecting Configuration: $x\#q_{reject}\#1.$

Exercise 5 (Sipser 3e - Exercise 3.8 b, c)

Give implementation-level descriptions of Turing machines that decide the following languages over the alphabet {0,1}.

b) {w | w contains twice as many 0s as 1s }

1. Scan the tape from the head, and mark the first unmarked 1. If there are no unmarked 1s, go to step 4.

2. Scan the tape from the head and mark the first unmarked 0. Reject if there are no 0s.

3. Scan the tape from the head and mark the first unmarked 0. Reject if there are no 0s. Otherwise go to step 1.

4. Scan the tape from the head and mark the first unmarked 0. Accept if there are no unmarked 0s. Reject otherwise.

c) {w | w does not contain twice as many 0s as 1s}

1. Scan the tape from the beginning and mark the first unmarked 1. If there are no unmarked 1s, then go to step 4.

2. Scan from the beginning and mark the first unmarked 0. Accept if there are no unmarked 0s.

3. Scan and mark the first unmarked 0. Accept if there are no unmarked 0s. Otherwise go to step 1.

4. Scan from the beginning and mark the first unmarked 0. Accept if an unmarked 0 is found. Reject otherwise.