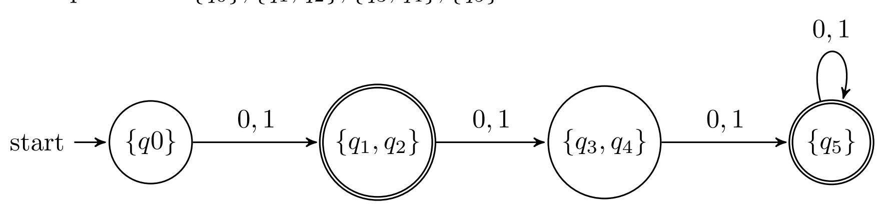
Roxanne Dewing
[redacted]
CSC 320 A01
Dr. Kapron
June 13, 2018

Homework 2–CSC 320 Summer 2018

Due by conneX submission at 11:55pm on Wednesday June 13

1. Use the state partiationing algorithm presented in class find the minimal automaton

   (a) Final partition: $\{q_0\}, \{q_1, q_2\}, \{q_3, q_4\}, \{q_5\}$



   (b) $L = \{w \in \{0, 1\}^* | \; |w| = 1$ or $|w| >= 3\}$

2. Prove the each of the following languages are not regular. You may use the pumping lemma, or closure properties of the regular langauges.

   (a) Suppose L is regular, let p be the pumping length
   Consider the string $w = 0^p 1^p 0^p$
   Clearly, $|w| >= p$, So, by the pumping lemma, there must be some choice of x,y,z satisfying the conditions of the pumping lemma
   Condition 1 of the pumping lemma states that $|xy| <= p$
   Therefor our choice of the x and y partitions of w is limited to:
   $x = \epsilon, y = 0^{p-v}$ where $0 <= v < p$ or
   $x = 0^{p-v}$ and $y = 0^{v-t}$ where $v < p$ and $t < v$
   Consider if we pump y zero times in both cases (i=0)
   In the first case, since x is empty and the non empty y partition can have up to p zeros, if pumped zero times, will guarentee the resulting string having less then p 0's in zeros leading up to the first one, which is less than the section of zeros after the last 1.
   Similarily in the second case, if we pump y zero times, the resulting string will have less than p zeros in its first section. Therefore the resulting string will not be in the language and the language is not regular by the pumping lemma

   (b) Suppose L is regular, let p be the pumping length
   Consider the string $w = 0^p 1^{2p}$
   Clearly, $|w| >= p$, So, by the pumping lemma, there must be some choice of x,y,z satisfying the conditions of the pumping lemma
   Condition 1 of the pumping lemma states that $|xy| <= p$
   Therefor our choice of the x and y partitions of w is limited to:
   $x = \epsilon, y = 0^{p-v}$ where $v < p$ or
   $x = 0^{p-t}$ and $y = 0^{t-v}$ where $t <= p$ and $v < t$
   in the first case, consider if i = $2p/(p - v)$, that means there would be $(p - v) * (2p/(p - v)) = 2p$ zeros in the resulting string, which is equal to the number of ones, therefor the string is not in the language.

in the second case, consider $i = ((p+t)/(t-v))$ that means there would be $(p-t)+(t-v)*((p+t)/(t-v)) = 2p$ zeros in the resulting string. then we will end with the string $0^{2p}1^{2p}$, which is not in the language.
Therefore the language is not regular by the pumping lemma

(c) Suppose L is regular, let p be the pumping length
Consider the string $w = 0^p110^p1$
Clearly, $|w| >= p$, So, by the pumping lemma, there must be some choice of x,y,z satisfying the conditions of the pumping lemma
This string can be broken up into 3 pieces w_1tw_1, where $w_1 = 0^p1$ and t = 1
Condition 1 of the pumping lemma states that $|xy| <= p$
Therefor our choice of the x and y partitions of w is limited to:
$x = \epsilon$ and $y = 0^{p-v}$ where $v < p$ or
or $x = 0^{p-t}$ and $y = 0^{t-v}$ where $0 < t < p$ and $v < x$
However in both cases, we can pump y, making a resulting string have more zeros in the first w component, and there will be no way to partition the resulting string into w_1tw_1

3. Give CFGs for the following languages over $\sigma = \{0, 1\}$

(a) S $\rightarrow \epsilon$
S $\rightarrow 1S1$
S $\rightarrow 0S0$
S $\rightarrow 1|0$

(b) S $\rightarrow \epsilon|S10|S01$
S $\rightarrow 1S0|0S1$

(c) S $\rightarrow 0S1|\epsilon$

4. Give a CFG that generates the language $A = \{a^ib^jc^k | i = j \text{ or } j = k \text{ where } i, j, k >= 0\}$

We can do this by considering the problem in 2 pieces and then combining by adding a new start variable.
First consider the language $\{a^ib^jc^k | i = j \text{ and } i, j, k >= 0\}$
This can be represented by the CFG:
B $\rightarrow AC$
C $\rightarrow \epsilon|Cc$
A $\rightarrow \epsilon|aAb$

Now consider the CFG representing the language $\{a^ib^jc^k | j = k \text{ and } i, j, k >= 0\}$
This can be represtented by:
D $\rightarrow EF$
E $\rightarrow \epsilon|Ea$
F $\rightarrow \epsilon|bFc$

Now we can add a new start symbol to combine these two languages, since they use an or statement and make the language $A = \{a^ib^jc^k | i = j \text{ or } j = k \text{ where } i, j, k >= 0\}$

$S_0 \rightarrow B$
$S_0 \rightarrow D$
C $\rightarrow \epsilon | Cc$
A $\rightarrow \epsilon | aAb$
B $\rightarrow AC$
E $\rightarrow \epsilon | Ea$
F $\rightarrow \epsilon | bFc$
D $\rightarrow EF$

5. Conversion into CNF

   $S_0 \rightarrow EB|TA|GJ|num$
   E $\rightarrow EB|TA|GJ|num$
   T $\rightarrow TA|GJ|num$
   F $\rightarrow GJ|num$
   A $\rightarrow CF$
   B $\rightarrow DT$
   J $\rightarrow EH$
   C $\rightarrow *$
   D $\rightarrow +$
   G $\rightarrow$ (
   H $\rightarrow$)

6. CYK algorithm

   Since the start symbol appears in the far right corner, running the CYK algorithm confirms that the string is in the language

(	id	+	num	)	*	num
L	-	-	-	E	-	E
	E	-	E	D	-	-
		P	B	-	-	-
			E	D	-	-
				R	-	-
					M	A
						E

7. Is every grammar in CNF unambiguous? If your answer is "yes", provide a proof. If your answer is "no", provide a counterexample.

   (a) No: counter example is an inheritdly ambigous grammer- since each context free grammer has a chomsky normal form, a CNF of an inheritdly ambiguous grammer would not remove abiguity. An example of this is $A = \{a^i b^j c^k | i = j \text{ or } j = k \text{ where } i, j, k >= 0\}$