COMP 615 (Fall 2021): HW 4

Name: Hossein Alishah
Student ID: 202048619

Section: Mon (7:00 pm – 9:45 pm)

Problem 1. (10 points)

Create an CFG that accepts $\{a^i b^j c^k d^l : i > 2j \ \& \ k > 3\ell\}$

Answer:

$S \rightarrow aXcY$
$X \rightarrow AAXB \mid \lambda$
$A \rightarrow aA \mid a$
$B \rightarrow b$
$Y \rightarrow CCCYD \mid \lambda$
$C \rightarrow cC \mid c$
$D \rightarrow d$

Problem 2. (10 points)

What language is generated by the CFG below.

$$\begin{aligned} S &\rightarrow aB \mid aS \mid bA \mid \lambda \\ A &\rightarrow aS \mid bAA \\ B &\rightarrow aBB \mid bS \end{aligned}$$

Answer:

L = { w ∈ {a, b}*: #a ≥ #b}

Or

L = { w ∈ {a, b}*: number of a's is greater than or equal number of b's}

I have tried some strings to get the answer:

S ⟹ aB ⟹ aaBB ⟹ aabSbS ⟹ aabbS ⟹ aabb (#a = #b)

S ⟹ bA ⟹ bbAA ⟹ bbaSaS ⟹ bbaaS ⟹ bbaaaS ⟹ bbaaa (#a ≥ #b)

S ⟹ aS ⟹ aaB⟹ aabS ⟹ aabsS ⟹ aabs (#a ≥ #b)

So, it is true to say number of a's are equal or greater than number of b's. if we didn't have the production S → aS, we could say #a = #b, bur this production make this language to accept all strings which #a ≥ #b.

Problem 3. (10 points)

Modify the grammar below to remove all λ-productions without changing the language generated.

$$
\begin{aligned}
S &\rightarrow SAS \mid CaA \mid b \\
A &\rightarrow BaB \mid CC \mid a \\
B &\rightarrow AaC \mid BA \mid b \\
C &\rightarrow aSa \mid SaACB \mid a \mid \cancel{\lambda}
\end{aligned}
$$

Answer:

First, find the nullable variable, λ-productions:

$$C \rightarrow \lambda$$

$$A \rightarrow CC \Rightarrow A \rightarrow \lambda$$

then, remove λ and make new productions:

$$
\begin{aligned}
&S \rightarrow SS \mid aA \mid Ca \mid a \\
&A \rightarrow C \\
&B \rightarrow Aa \mid aC \mid B \mid a \\
&C \rightarrow SaCB \mid SaAB \mid SaB
\end{aligned}
$$

Problem 4. (10 points)

Modify the grammar below to remove all unit-productions without changing the language generated.

$$
\begin{aligned}
S &\rightarrow aB \mid aS \mid bA \mid cA \mid cB \mid c \\
A &\rightarrow aS \mid bAA \mid \cancel{C} \\
B &\rightarrow aBB \mid bS \mid \cancel{C} \\
C &\rightarrow cS \mid aBC \mid aAC \mid \cancel{S}
\end{aligned}
$$

Answer:

First, find all direct and indirect unit-productions:

Directs:

$$
\begin{aligned}
&A \rightarrow C \\
&B \rightarrow C \\
&C \rightarrow S
\end{aligned}
$$

Indirect:

$$
\begin{aligned}
&A \rightarrow S \\
&B \rightarrow S
\end{aligned}
$$

Second, remove all direct unit productions.

Finally, add productions that allows for lost opportunities:

A → cS | aBC | aAC | aB | bA | cA | cB | c
B → cS | aBC | aAC | aB | aS | bA | cA | cB | c
C → aB | aS | bA | cA | cB | c

Problem 5. (10 points)

Convert the grammar below into Chomsky Normal Form (ie give a grammar in CNF that still generates the same language).

$$
\begin{aligned}
S &\rightarrow SAS \mid CaA \mid b \\
A &\rightarrow BaB \mid CC \mid a \\
B &\rightarrow AaC \mid BA \mid b \\
C &\rightarrow aSa \mid SaACB \mid a
\end{aligned}
$$

DEFINITION 6.4

A context-free grammar is in Chomsky normal form if all productions are of the form

$$A \rightarrow BC$$

or

$$A \rightarrow a,$$

where A, B, C are in V, and a is in T.

Answer:

First, we must make sure out grammar doesn't have any λ-productions, next check for unit-production. This grammar doesn't have λ-productions and unit-productions.

CNF:

S → SE | CJ | b
E → AS
J → DA
D → a
A → BI | CC | a
I → DB |
B → AH | BA | b
H → DC
C → DG | SF | a

$G \rightarrow SD$
$F \rightarrow DK$
$K \rightarrow AL$
$L \rightarrow CB$

Problem 6. (10 points)

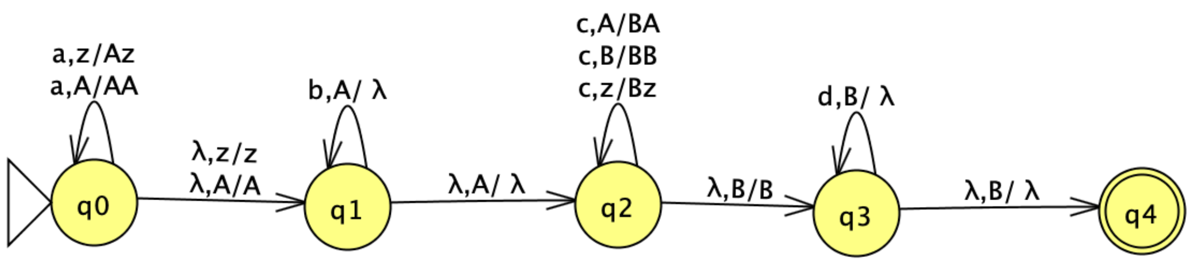
Create an NPDA that accepts $\{a^i b^j c^k d^l : i > j \ \& \ k > \ell\}$.

Answer:

Context Free Grammar is:

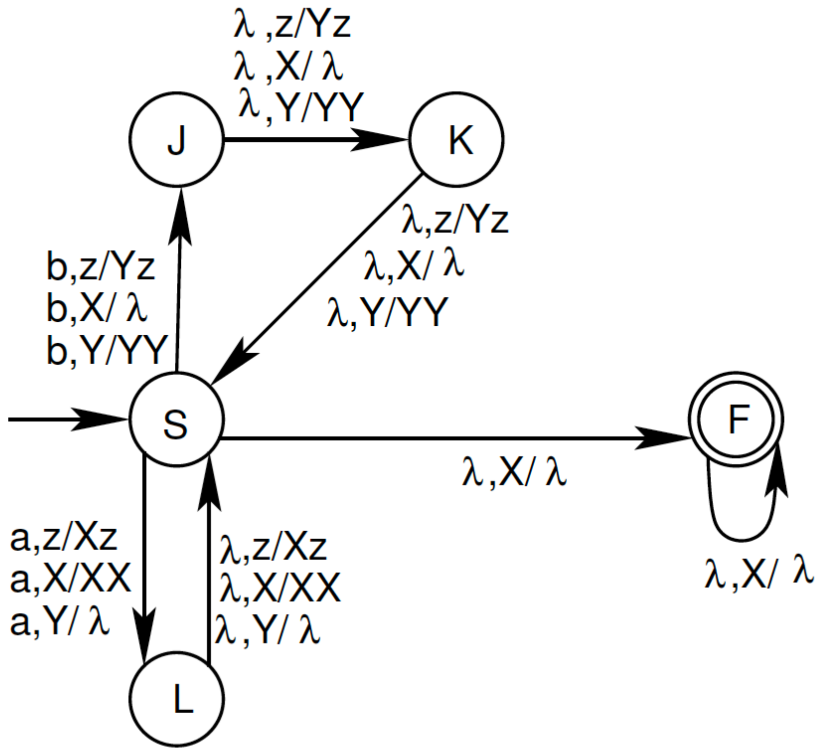
$S \rightarrow aXcY$
$X \rightarrow AXB \mid \lambda$
$A \rightarrow aA \mid a$
$B \rightarrow b$
$Y \rightarrow CYD \mid \lambda$
$C \rightarrow cC \mid c$
$D \rightarrow d$

NPDA:



Problem 7. (10 points)

What language is accepted by the NPDA below.



Answer:

L = {w ∈ {a, b}*: 2×(#a) > 3×(#b)}

For more clarification, this NPDA just accepts those strings have 2×(#a) > 3×(#b)}, because when the machine read one b, then it pushes three Y's to stack or erase two X's from the stack, but it just erase two Y's from the stack when it read each a. this NPDA accept when read all strings and there is at least one X on the top of the stack.

Problem 8. (10 points)

Use the pumping lemma to prove that L = {$a^ib^jc^k$: i = j . k} is not context free.

Answer:

Proof:

Assume L is CF.

The Pumping Lemma (*PL*) applies and there is $M>0$ given by *P.L*. for all $w \in L$, $|w| \geq M$

Pick $w = \boldsymbol{a^{M^\wedge 2} b^M c^M}$

Let $w = uvxyz$ for some *u, v, x, y*, and *z* with $|vy| \neq 0$ and $|vxy| \leq M$.

Notice: *vy* must contain some type of letter (a's or b's or c's)

Notice: *vy* cannot contain all three types of letters.

Let's consider $uv^2xy^2z \in L$ (uv^ixy^iz) because conclusion of pumping lemma:

Case 1: v & y are only in a's: if we pump down to i=0, we have more b's or c's than a's, so w $\notin L$

Case 2: v & y are only in b's: if we pump up to i=2, we have more b's than a's or c's, so w $\notin L$

Case 3: v & y are only in c's: if we pump up to i=2, we have more c's than a's or b's, so w $\notin L$

Case 4: v & y are only in a's and b's: if we pump down to i=0, we have more c's than a's or b's, so w $\notin L$

Case 5: v & y are only in b's and c's: if we pump up to i=2, we have more b's or c's than a's, so w $\notin L$

Which means that $uv^2xy^2z \notin L$. *So*, L is not *CFL*.

Problem 9. (10 points)

Use the Pumping Lemma for CFL to show that
L = {w ∈ {a, b, c}*: #a · #b = #c} is not context free.

Answer:

Proof:

Assume L is CF.

The Pumping Lemma (*PL*) applies and there is $M>0$ given by *P.L.* for all $w \in L$, $|w| \geq M$

Pick $w=$$\boldsymbol{c^{M\text{^}2}a^Mb^M}$ ***(order of a's or b's or c's doesn't matter)***

This **#*a* · #*b* = #*c*** is means that we have more #c than b's or c's.

Let $w = uvxyz$ for some u, v, x, y, and z with $|vy| \neq 0$ and $|vxy| \leq M$.

Notice: vy must contain some type of letter (a's or b's or c's)

Notice: vy cannot contain all three types of letters.

Let's consider $uv^2xy^2z \in L$ (uv^ixy^iz) because conclusion of pumping lemma:

Case 1: v & y are only in c's: if we pump down to i=0, we have more b's or a's than c's, so w $\notin L$

Case 2: v & y are only in b's or only in a's: if we pump up to i=2, we have more b's or a's than c's, so w $\notin L$
Case 3: v & y are only in a's and c's: if we pump down to i=0, we have more b's than c's, so w $\notin L$
Which means that $uv^2xy^2z \notin L$. *So,* L is not *CFL*.

Problem 10. (10 points)

Given any 2 languages L_1 and L_2 define *INTERLACE*(L_1, L_2) as the set of strings that can be created by taking any string $x = x[1]x[2] : : : x[k] \in L_1$ and $y = y[1]y[2] : : : y[\ell] \in L_2$ and performing the following process

```
int i=0, j=0;
String str="";
while (i<k || j<l) {
   if (i==k)
      str += y[++j];
   else if (j==l)
      str += x[++i];
   else {
      str += x[++j];
      str += y[++i];
   }
}
```

Prove: if L_1 and L_2 are context free then *INTERLACE*(L_1, L_2) is also context free.