CSE 105:Introduction to the Theory of Computation, Fall 1999

Homework 2

For many of the problems there are multiple possible solutions. The answers presented below outline one such solution for each problem. In some of the diagrams, E, represents an empty transition.

PROBLEM 1 a) $S->S1S1S0|S1S0S1|S0S1S1|$

PROBLEM 1 B) $R->A|b|(R\cup R)|(RoR)|(R*)||$

PROBLEM 1 C)

$S->0A|B|\ A->1A|\ B->10B|$

PROBLEM 2

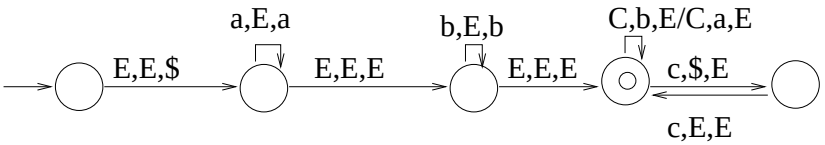

Fig. 1. Solution for 2.

informally: Push all the a's, Push all the b's, when a c is read, if a or b is on the stack, pop, if $ is on the stack, move to non-accepting state. If more c's are read, move to accepting state and stay even if additional c's are still on the stack.

See figure 1.

PROBLEM 3 a) $A=\{a^mb^nc^n|m,n\geq 0\}$. A is context free as can be seen in the following CFG.

- $S->AD$
- $D->bDc|$
- $A->aA|$

$B=\{a^nb^nc^m|m,n\geq 0\}$. B is context free as can be seen in the following CFG.

- $S->DC$
- $C->cC|$
- $D->aDb|$

Strings in A have the same number of b's and c's, while string in B have the same number of a's and b's. Thus, it is cleat that the union of A and B will have the same number of a's, b's, and c's $A\cap B=\{a^nb^nc^n|n\geq 0\}$. As $A\cap B$ is the same as language C in example 2.20, and language C in example 2.20 has been shoen to be non-context free, $A\cap B$ is non-context free. Thus, not all unions between two context free languages are context free, therefore context free languages are not closed under intersection.

PROBLEM 3 b) Assume that CFL's are closed under complementation. Thus, if A is CFL, so is !A. Therefore, $A\cap B=!(!A\cup !B)$ is also CFL. Thus, contradict the result of Problem 3, however. Therefore, one of the assumptions is wrong. The only assumption is that CFL's are closed under complementation. Therefore, CFL's are not closed under complementation.

PROBLEM 4

step 1. Add a new state: S-¿E

step 2. Remove rules: there are none

step 3. Remove unit rules:

a) Remove $T->F: S->E, E->E+T|TT->T*F|(E)|a, F->(E)|a$

b) Remove $E->T: S->E, E->E+T|T*F|(E)|a, T->T*F|(E)|a, F->(E)|a$

c) Remove $S->E: S->E+T|T*F|(E)|a, E->E+T|T*F|(E)|a, T->T*F|(E)|a, F->(E)|a$

step 4. Transform into proper form(fix $F->(E)|a, T->T*F|(E)|a, E-E+T|T*F|(E)|a, Srule$)

Final answer:

1. $S->EA_6|TA_4|A_1A_2|a$
2. $E->EA_6|TA_4|A_1A_2|a$
3. $T->TA_4|A_1A_2|a$
4. $F->A_1A_2|a$
5. $A_1->($
6. $A_2->EA_3$
7. $A_3->)$
8. $A_4->A_5F$
9. $A_5->*$
10. $A_6->A_7T$
11. $A_7->+$

PROBLEM 5

Consider the string s = c sub c sub c. As shown in figure 2, there are multiple distinct parse trees. Therefore, the grammer is ambigous.

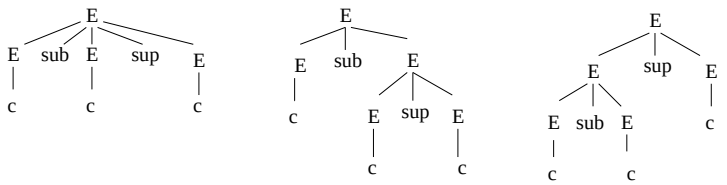

Fig. 2. Solution for 5.

PROBLEM 6

Proof by contradiction.

Assume $L = \{www|w \in \{a,b\}*\}$ is CFL. Since L is CFL, we can apply the pumping lemma.

Let $s = a^pb^pa^pb^pa^pb^p$ be the target string, where p is the pumping length. Since $|s| \geq p$ and $s \in L$, we know that s can be divded into $uvxyz$ such that

1. $|vxy| \leq p$
2. $|vy| > 0$
3. for all $i \geq 0$, $uv^ixy^iz \in L$

By (1) and (2) we know that $vy = a^jb^k$ or b^ja^k for $0 < j + k \leq p$.

Case 1: $vy = a^jb^k$

If vxy lies in the first third of the string, when we pump it we get $uv^0x^0z = a^{p-j}b^{p-k}a^pb^pa^pb^p$ which is not in L, since $0 < j + k$. It can be seen that if vxy lies in the second or third portion of the string, we get strings that also are not in L using the identical method.

Case 2: $vy = b^ja^k$

If vxy crosses the first and second thirds of the string, pumping can result in the string $uv^0xy^0z = a^pb^{p-j}a^{p-k}b^pa^pb^p$ which is not in L, since $0 < j + k$. It can be seen that if vxy crossed the second and third segment the strings are also not in L, using the same analysis.

As case 1 and 2 encompass all possible conditions, for all cases pumping can result in strings not contained in L. Therefore, L is not a CFL. This is a constradiction, indicating that one of our assumptions is incorrect. As our only assumption is that L is a CFL, L must not be a CFL.