Homework #1

Solutions

Please email me if you see any errors or have any questions.

Each question is worth 10 points.

#1. Let Given the alphabet $\Sigma = \{a,b\}$, and the languages over Σ: $L_1 = \{aaa\}^*$, $L_2 = \{a,b\}\{a,b\}\{a,b\}\{a,b\}$ and $L_3 = L_2^*$, describe the strings in

a) L_2
b) L_3
c) $L_1 \cap L_3$

a) L_2 = {aaaa,aaab,aaba,aabb,abaa,abab,abba,abbb,baaa,baab,baba,babb,bbaa,bbab, bbba,bbbb}

b) $L_3 = \{w \varepsilon \{a,b\}^* : |w| = 4*n, n \geq 0\}$

c) $L_1 \cap L_3 = \{a^n \mid n = 12k, k \geq 0\}$

#2. Give regular expressions for the following:

a) The set of strings over {a,b,c} where all the a's precede all the b's which precede all the c's (there may be no a's, b's or c's)

a*b*c*

b) The set of strings over {0,1} which contain the substring *00* and the substring *11*.

((0U1)*00 (0U1)*11 (0U1)*) U ((0U1)*11 (0U1)*00 (0U1)*)

c) The set of strings over {a,b} which do not contain the substring *ab*.

b* a*

#3. a) Let G be the grammar:

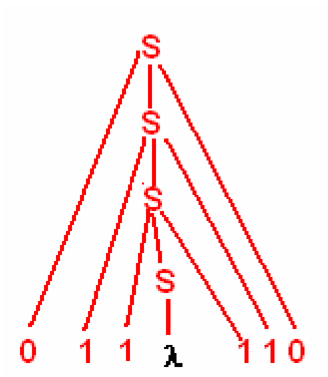
$S \rightarrow 0 \mid 1 \mid 0 S 0 \mid 1 S 1 \mid \lambda \mid 0 0 \mid 1 1$

a) Show a leftmost derivation of *0 1 1 1 1 0*

Note: any derivation will be leftmost (or rightmost!)

$S \rightarrow 0 S 0 \rightarrow 0 1 S 1 0 \rightarrow 0 1 1 S 1 1 0 \rightarrow 0 1 1 1 1 0$

b) Create a parse tree for *0 1 1 1 1 0*



c) Show that this grammar is ambiguous

Two leftmost derivations for 00 (among others):

1. **S → 0 0**
2. **S → 0 S 0 → 0 0 (using S→ λ)**

d) Describe L(G) using set notation

L(G) = {w ε {0,1}*| w = w^R}

b) Construct grammars to generate the languages of #2

a)
S → aS | B
B → bB | C
C → cC | λ

b)
S → 0 S | 1 S | 0 0 A |1 1 B
A → 0 A | 1 A | 1 1 C
B → 0 B | 1 B | 0 0 C
C → 0 C | 1 C | λ

c)
S → b A | A | λ
A → a A | λ

#4. Explain briefly and clearly why (how) all finite alphabets can be replaced with a two symbol alphabet. Do this in general (for any length alphabet) and then show your method for the alphabet {a,b,c} and the string *b b c a*.

Given the alphabet {a_1,a_2, a_3, …, a_n} and the two symbol alphabet { b_1,b_2,}.
Represent the symbols as follows:

$a_1 = b_1$
$a_2 = b_1\ b_1$
$a_3 = b_1\ b_1\ b_1$
…
$a_n = b_1\ b_1\ b_1 \ldots b_1$ (n b_1 's)
Use b_2 as a separator between symbols. So if the string is $a_3\ a_1\ a_2$, it can now be represented by $b_1\ b_1\ b_1\ b_2\ b_1\ b_1\ b_2\ b_1$

So for {a,b,c} using the two symbol alphabet {a,b},

a = a
b = a a
c = a a a
and *b b c a* is *a a b a a b a a a b a*

#5. For the CFG G defined by

S → 0 S | S 1 | 0 | 1
prove by induction on the depth of the parse tree that no string in the language has *10* as a substring.

<u>Basis</u> depth(tree) = 1. The only 2 trees are

and neither of them are *1 0.*

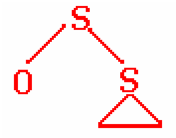
<u>Induction Hypothesis</u>

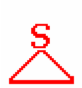
Assume for trees of depth k, $k \geq 1$ that none contain *1 0* as a substring.

<u>Induction Step</u>

Consider trees of depth $k + 1$:

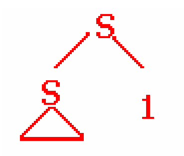
<u>Case 1</u>

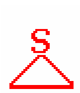


where 

is a parse tree of depth k and will not generate *1 0* as a substring by the induction hypothesis. Prefixing this string with 0 still will not generate a string with a substring 1 0

<u>Case 2</u>



where 

is a parse tree of depth k and will not generate *1 0* as a substring by the induction hypothesis. Suffixing this string with 1 still will not generate a string with a substring 1 0.