Chris Fenton
CNC - Formal Languages
Homework 3

**1. Give a recursive definition of the length of a string over Σ. Use the primitive operation from the definition of string.**

Basis: If string w = $\lambda$, then length(w) = 0
Recursive: 1 + length(v), where w = uv

**5. Let L be the set of strings over {a, b} generated by the recursive definition**

i) Basis: b $\in$ L
ii) Recursive step: if u is in L then ub $\in$ L, uab $\in$ L, and uba $\in$ L, and bua $\in$ L
iii) Closure: a string v is in L only if it can be obtained from the basis by a finite number of iterations of the recursive step

**a) List the elements in the sets $L_0$, $L_1$, $L_2$.**

$L_0$ = {b}, $L_1$ = {b, bb, bab, bba}, $L_2$ = {bb, bbb, babb, bab, babab, bbaab, bba, bbba, babba, bbaba, bbbaa}

**b) Is the string “bbaaba” in L? If so, trace how it is produced. If not, explain why not.**

The string “bbaaba” is not in L. You need 2 applications of a rule that adds “a” to right side of a string to get the “aa” in the string. Since both “uab” and “bua” applied to “b” leave you with one more “b” than “a”, you cannot produce the substring “bbaa”.

**c) Is the string “bbaaaabb” in L? If so, trace how it is produced. If not, explain why not.**

The string “bbaaaabb” is not in L for the same reason as the previous subproblem--every time you add an “a” to the right side of a b, you have more “b”s then “a”s.

**6. Give a recursive definition of the set of strings over {a, b} that contain at least one b and have an even number of a’s before the first b.**

Let Σ = {a, b}, L = $\Sigma^*$

i) Basis: b $\in$ L
ii) Recursive step: if u is in L then ub $\in$ L, ua $\in$ L, and aau $\in$ L
iii) Closure: a string w is in L only if it can be obtained from the basis by a finite number of iterations of the recursive step

Example: $L_0$ = {b}, $L_1$ = {b, bb, ba, aab}, $L_2$ = {bb, bbb, bab, aabb, ba, bba, baa, aaba, aab, aaaab}

10.

12.

13.

**14. The set of strings over {a, b, c} in which all the a’s precede the b’s, which in turn precede the c’s. It is possible that there are no a’s, b’s, or c’s. (Give a regular expression)**

R = a*b*c*

**23. The set of strings over {a, b, c} that begin with a, contain exactly two b’s, and end with cc.**

R = a (a | c)* bb (a | c)* cc