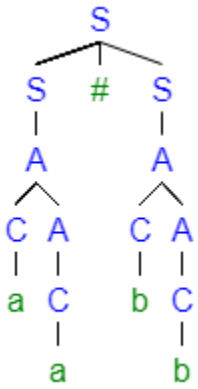
Kyle Frick
Principle of Programming Languages

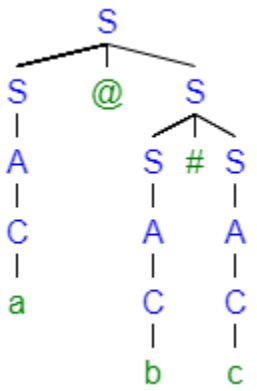
January 29th 2018
Homework 2

1. A. aa # bb
    a. Derivation:
        S
        S # S
        A # S
        C A # S
        a A # S
        a C # S
        a a # S
        a a # A
        a a # C A
        a a # b A
        a a # b C
        a a # b b
    b. Parse Tree



1. B. a @ b # c
    a. Derivation:
        S
        S@S
        A@S
        C@S
        a@S
        a@S#S
        a@A#S
        a@C#S
        a@b#S
        a@b#A
        a@b#C
        a@b#c

b. Parse Tree



1. C. ab
    a. Derivation
        S
        A
        CA
        aA
        aC
        ab
    b. Parse Tree

2. S → S @ B | B
   B → S # B | A
   A → C | C A
   C → a | b | c
3. A) a = 0,b; is in the L(G') because it is derivable.
    a. <Statement>
        <Assignment>
        <Var> = <Value>[,<Value>];
        a = <Value> [, <Value>];
        a = <Number> [, <Value>];
        a = 0 [, <Value>];
        a = 0 [, <Var>];
        a = 0 [, b];

a = 0,b; //the [] disappear since they are not terminal.

3. B) a = b,c,1; is not in the L(G') because the assignment of a var can only hold two values not three.
3. C) while(a){b =0; while(b) {}} is in the L(G') because it is derivable
    a. <Statement>
    <While>
    while(<Value>){ {<Statement>} }
    while(<Var>) { {<Statement>} }
    while(a) { {<Statement>} }
    while(a) { <Assignment> {<Statement>} }
    while(a) { <Var> = <Value> [, <Value>]; {<Statement>} }
    while(a) { b = <Value> [, <Value>]; {<Statement>} }
    while(a) { b = <Number> [, <Value>]; {<Statement>} }
    while(a) { b = 0; {<Statement>} } //ignoring the []
    while(a) { b = 0; <While> {<Statement>} }
    while(a) { b = 0; while(<Value>) { {<Statement>} } {<Statement>} }
    while(a) { b = 0; while(<Var>) { {<Statement>} } {<Statement>} }
    while(a) { b = 0; while(b) { {<Statement>} } {<Statement>} }
    while(a) { b = 0; while(b) { } {<Statement>} } //repeating section zero times
    while(a) { b = 0; while(b) {} } //end repetition
3. D) a=1; while(a) {a=1; while(a=0;} is not in the L(G') because the Assignment statement can't also be followed by a while statement unless within a while statement as seen above.
3. E) Terminals are in Red
    <Statement> → <If>
    <If> → if(<Value>){ {<Statement>} } else { {<Statement>} }
4. <syntax> ::= <rule> | <rule> <syntax>
    <rule>::= "<" <rule-name> ">" "::=" <expression>
    <expression>::= <list> | <list> "|" <expression>
    <list>::= <term> | <term> <list>
    <term>::= <literal> | "<" <rule-name> ">"
    <literal>::= '"' <text> '"' | "'" <text> "'"
    <text>::= "" | <character> <text>
    <character>::= <letter> | <digit> | <symbol>
    <letter>::= a…..z,A….Z
    <digit>::= 0…9
    <symbol>::= |…~
    <character>::= <character> | "'"
    <rule-name>::= <letter> | <rule-name>