Introduction:

We've recently seen a few things about grammars, parsing, derivation and how to decide and design different elements of languages. Demonstrate your knowledge and facility with these mechanisms below.

Methodology:

Submit a PDF with your answers to the below on Sakai on or before 2355 23 September.

Note: in all of the below capital letters are non-terminals, non-grammar symbols are terminals.

0. Given the following grammar, G:

S → A | S # S | S @ S
A → C | C A
C → a | b | c

For each production below, state whether it is:

In the language G decides
- if so, prove it with a derviation and parse tree
- if not, explain why

If it is ambiguous in G
- if so, prove it is with another valid parse tree in G

Productions:

I. aa ## bb
II. a @ b # c
III. ab

1. Rewrite grammar G so that:

- no string in it is ambiguous
- # has higher precedence than @
- # and @ are both left-associative
  (i.e. a # b # c should mean (a # b) # c)

2. Given grammar H below:

(in Extended BNF, terminals in bold and underlined):

<Statement> → <Assignment>
<Statement> → <While>
<Assignment> → <Var> **=** <Value> [**,** <Value>] **;**
<While> → **while(**<Value>**)** **{** {<Statement>} **}**
<Value> → <Var> | <Number>
<Var> → **a** | **b** | **c**
<Number> → **0** | **1**

For each production below:

- determine if it is in the language H
- explain why or why not

Productions:

I. a = 0,b;
II. a = b,c,1;
III. while(a){b = 0; while (b) { }}
IV. a=1; while(a) {a = 1; while (a=0;}

3. Write a grammar for BNF in BNF

(not EBNF, e.g. no curly braces)

Evaluation:

All on time submissions will be graded out of 100 points, 25 points per question.