CSCU9Y4 Spring 2018

Programming Language Paradigms

Tutorial 18/6

Background reading: Chapter 3 of Introduction to Programming Languages. Supplementary notes 6.

Alternatively, Chapter 12 of Comparative Programming Languages, Chapter 3 of Sebesta, Concepts of Programming Languages, or Chapter 3.3 of Pratt and Zelkowitz. Or google "Backus-Naur form", "programming language semantics" etc.

1. Write some sentences which can be generated by the following grammar:

```
<sentence>      ::=   <noun phrase> <verb phrase>
<noun phrase>::=      <proper name>
                      | <article> <noun>
                      | <article> <adjective> <noun>
<verb phrase> ::=     <i_verb>
                      | <t_verb> <noun phrase>
<noun>          ::=   boy | girl
<i_verb>        ::=   smiled
<t_verb>        ::=   liked | annoyed
<proper name>::=      John | Janet
<adjective>     ::=   little | clever
<article>       ::=   a | the
```

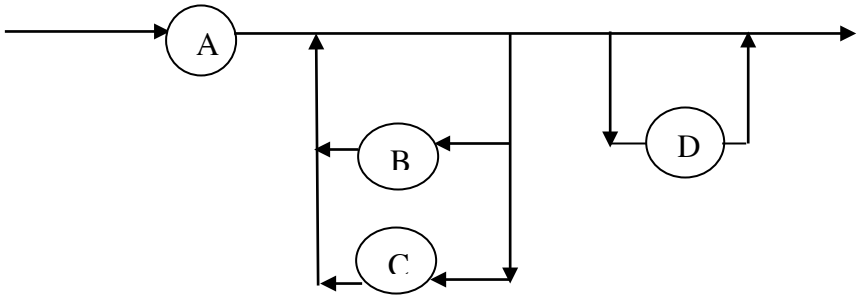
2. By generating two different derivation trees for the expression x * y + z, show that the BNF grammar

```
<exp>  ::=    <ident>
              | <exp> + <exp>
              | <exp> * <exp>
<ident> ::=   x | y | z
```

is ambiguous. How could you have known without doing the derivation that the grammar was ambiguous? What changes might be made to the grammar to remove the ambiguity discovered?

3. List all valid letter strings of one, two or three characters that can be derived from the syntax diagram overleaf:

PTO



Write down the corresponding EBNF production rule for this diagram (or alternatively, work out the EBNF from the set of expressions you've obtained).

4.

(a) Using the BNF grammar below, and assuming that <rule1> is the start symbol of the language, calculate the derivation of the following expressions to deduce if they are legal or illegal:

(i) Z
(ii) ZZY
(iii) ZXY
(iv) ZXYYY

<rule1> ::= <rule1> Y | <rule2>
<rule2> ::= Z <rule3> | Z
<rule3> ::= X

(b) Convert the BNF grammar into EBNF.

(c) Draw the corresponding syntax diagrams for each EBNF production rule.

5. The Java Language Specification uses a mixture of formal and informal techniques to describe the components of the language. An alternative description might have been to use a completely formal approach (or indeed, a completely informal approach). What are the advantages and disadvantages of these?