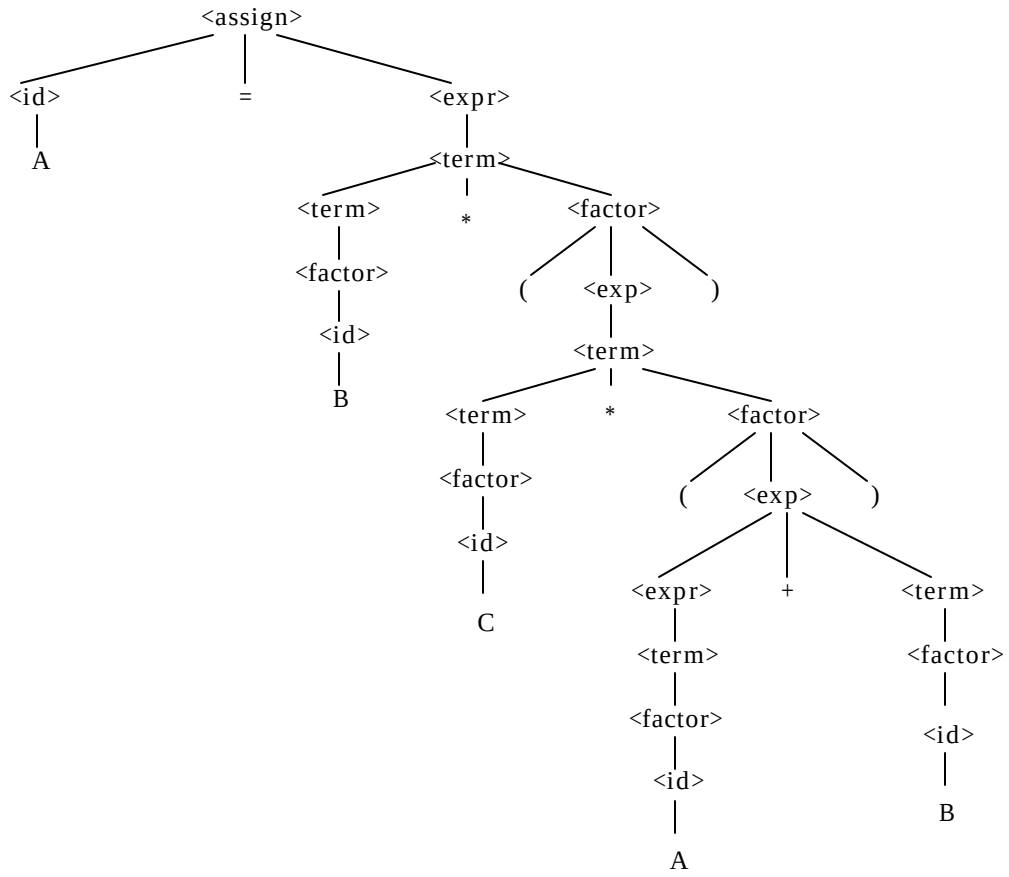
CMPT 383
Quiz #3
September 29, 2005

1) Using grammar 1, show the parse tree, the abstract syntax tree and the leftmost derivation for

`A = B*(C*(A+B))`

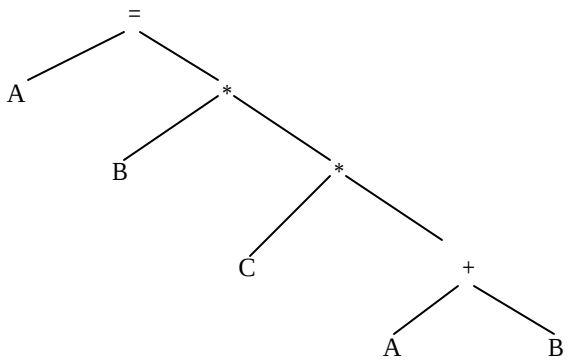
Parse Tree (3 marks)



Leftmost Derivation (2 marks)

<assign> ⇒ <id> = <expr> ⇒ A = <expr> ⇒ A = <term> ⇒ A = <term> * <factor>
⇒ A = <factor> * <factor> ⇒ A = <id> * <factor> ⇒ A = B * <factor>
⇒ A = B * (<expr>) ⇒ A = B * (<term>) ⇒ A = B * (<term> * <factor>)
⇒ A = B * (<factor> * <factor>) ⇒ A = B * (<id> * <factor>)
⇒ A = B * (C * <factor>) ⇒ A = B * (C * (<expr>))
⇒ A = B * (C * (<expr> + <term>)) ⇒ A = B * (C * (<term> + <term>))
⇒ A = B * (C * (<factor> + <term>)) ⇒ A = B * (C * (<id> + <term>))
⇒ A = B * (C * (A + <term>)) ⇒ A = B * (C * (A + <factor>))
⇒ A = B * (C * (A + <id>)) ⇒ A = B * (C * (A + B))

Abstract Syntax Tree (2 marks)



Grammar 1

<assign> ::= <id> = <expr>
<id> ::= A | B | C
<expr> ::= <expr> + <term> | <term>
<term> ::= <term> * <factor> | <factor>
<factor> ::= (<expr>) | <id>

2) Describe, in English, the language defined by grammar 2. **(2 marks)**

Grammar 2 produces strings formed by one or more letter *a* followed by one or more letter *b* followed by one or more letter *c*

Grammar 2

<S> ::= <A> <B> <C>
<A> ::= a <A> | a
<B> ::= b <B> | b
<C> ::= c <C> | c

3) Which of the following sentences are in the language generated by grammar 3? **(3 marks)**

a) abcd

Yes.

<S> ⇒ a <S> c <B> ⇒ a b c <B> ⇒ a b c d

b) acccbd

No.

<S> ⇒ a <S> c <B> ⇒ a <A> c <B> ⇒ a c <A> c <B> ⇒ a c c c <B>

No derivation of *bd* from *B*

c) acccbcc

No.

<S> ⇒ a <S> c <B> ⇒ a <A> c <B> ⇒ a c <A> c <B> ⇒ a c c c <B>

No derivation of *bcc* from *B*

d) acd

No.

<S> ⇒ a <S> c <B>

Every string that begins with *a* must have at least 4 characters.

e) accc

Yes.

<S> ⇒ a <S> c <B> ⇒ a <A> c <B> ⇒ a c c <B> ⇒ a c c <A> ⇒ a c c c

Grammar 3

<S> ::= a <S> c <B> | <A> | b

<A> ::= c <A> | c

<B> ::= d | <A>

4) Convert the following EBNF to BNF:

a) <S> ::= <A> { b <A> } **(2 marks)**

<S> ::= <A> <B> | <A>

<B> ::= b <A> <B> | b <A>

b) <A> ::= a [b] <A> **(1 mark)**

<A> ::= a b <A> | a <A>