Wentworth Institute of Technology

COMP3350 Programming Languages

Homework 3 - 20 points

1. Here are some productions that can generate simple expressions of the form

(a + a) \* a

Verify that you can generate the expression (a + a) \* a in this language: [5]

<EXPR> -> <EXPR> + <TERM> | <TERM>

<TERM> -> <TERM> \* <FACTOR> | <FACTOR>

<FACTOR> -> (<EXPR>) | a

<EXPR> -> <TERM>

-> <TERM> \* <FACTOR>

-> <FACTOR> \* <FACTOR>

-> (<EXPR>) \* <FACTOR>

-> (<EXPR> + <TERM>) \* <FACTOR>

-> (<TERM> + <TERM>) \* <FACTOR>

-> (<FACTOR> + <TERM>) \* <FACTOR>

-> (a + <TERM>) \* <FACTOR>

-> (a + <FACTOR>) \* <FACTOR>

-> (a + a) \* <FACTOR>

<EXPR> -> (a + a) \* a

1. Given the productions for infix expressions [5]

E → E + T | E − T | T

T → T ∗ F | T / F | F

F → (E) | identiﬁer | number

Construct a rightmost derivation for the expression (5 \* x) + y

E -> E + T

E -> E + F

E -> E + Y

E -> F +Y

E -> (E) + Y

E -> (T) + Y

E -> (T \* F) + Y

E -> (T \* X) + Y

E -> (F \* X) + Y

E -> (5 \* X + Y)

1. Show that the following grammar is ambiguous [5]

<assign> -> <id> = <expr>

<id> -> A | B | C

<expr> -> <expr> + <expr>

**<expr> \* <expr>**

| (<expr>)

| <id>

<assign> -> <id>=<expr>

-> A = <expr>

->A = <expr>+<expr>

->A=<expr>+<expr>

->A=<expr>+<id>

->A=B+<expr>

->A=B+<id>

->A=B+C\*<expr>

->A=B+C\*<id>

->A=B+C\*A

Use the sentence A = B + C \*A to demonstrate the ambiguity.

1. Consider the following grammar: [5]

S -> aScB | A | b

A -> cA | c

B -> d | A

Which of the following sentences are in the language generated by this

grammar.

1. **Abcd**
2. ~~accbcc~~