**Homework 4**

**Programming Languages Principles and Implementation**

**Yevgeniy Shatrovskiy**

## Grammars

*Due on 11/13*

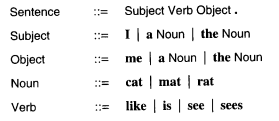
The work has to be done alone or in a group of 2 students.

A hard copy is required. Professional presentation is important.

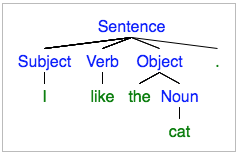
100 points

**Exercise 1:**

We consider the BNF grammar below:



1. Show that **I like the cat.** is recognized by this BNF grammar using a rightmost derivation and, then, a parse tree.



Sentence => Subject Verb Object.

=> Subject Verb the Noun.

=> Subject Verb the cat.

=> Subject like the cat.

=> I like the cat.

1. Provide an expression that is NOT recognized by the grammar.

I like cats.

Explanation: Object needs either “a” or “the” if not using “me”. Also, “Cats” is not a known noun.

**Exercise 2:**

We consider the following grammar:

EXPRESSION ::= NUMERAL | ( EXPRESSION OPERATOR EXPRESSION )

NUMERAL ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

OPERATORS ::= + | -

Show that (4 - (3 + 2)) is a legal EXPRESSION using a leftmost derivation, and,then, a parse tree.

EXPRESSION => (EXPRESSION OPERATOR EXPRESSION)

=> (NUMERAL OPERATOR EXPRESSION)

=> (4 OPERATOR EXPRESSION)

=> (4 - EXPRESSION)

=> (4 - (EXPRESSION OPERATOR EXPRESSION))

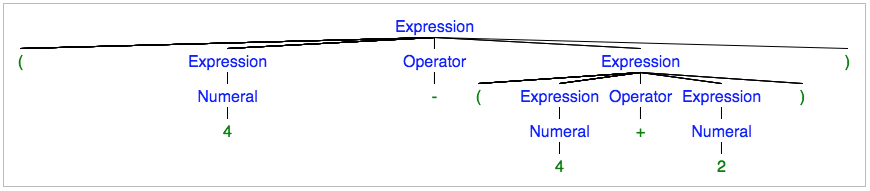
=> (4 - (NUMERAL OPERATOR EXPRESSION))

=> (4 - (3 OPERATOR EXPRESSION))

=> (4 - (3 + EXPRESSION))

=> (4 - (3 + NUMERAL))

=> (4 - (3 + 2))



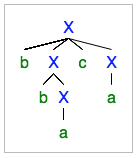
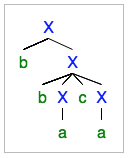
**Exercise 3:**

Show that the following grammar is ambiguous:

X -> a | bX | bXcX

where a,b,c are terminals.

There are two parse trees possible:



**Exercise 4:**

1. Design a BNF grammar that recognizes expressions of the form Ai where A is in {a,b,c} and i is a digit.

EXPRESSION ::= LETTER DIGIT

A ::= |a| b| c

i ::= 0|1|2|3|4|5|6|7|8|9

1. Design a BNF grammar that recognizes lists of the form A1, A2, A3, …, An. Use question a).

EXPRESSION ::= SET | EXPRESSION , SET

SET ::= LETTER DIGIT

A ::= |a| b| c

i ::= 0|1|2|3|4|5|6|7|8|9

**Exercise 5:**

1. Write a JAY program that computes the sum of the *n* first numbers with a loop.

void main () {

int n, counter, total;

n = 5 // Example Input

counter = 1;

total = 0;

while (counter <= n) {

total = total + counter;

counter = counter + 1;

}

}

1. Write a JAY program that assigns the minimum of two numbers in a variable called min.

void main () {

int min, firstNum, secondNum;

min = 0;

firstNum = 10;

secondNum = 12;

if (firstNum <= secondNum)

min = firstNum;

else

min = secondNum;

}

1. Provide 2 examples of lexical errors in JAY.
2. &

“&&” is recognized, but a single “&” is not.

1. |

“||” is recognized, but a single “|” is not.

1. Provide 2 examples of JAY programs with 2 different syntax errors.
2. firstVar += 4;

“+=” is not recognized as an assignment expression because the “+” would not be considered as part of the Identifier. In other languages, this would be interpreted as “firstVar = firstVar + 4;” , but in Jay, this would be a syntax error.

1. if () firstVar = firstVar + 2;

The if statement cannot have a blank Expression in the IfStatement part of the Statement. The “(“ and “)” would not be lexical errors since they are recognized, but the missing Expression would have an error.

1. Provide 2 examples of JAY programs with errors that are neither detected during the lexical analysis nor during the syntactic analysis.

1)

void main () {

// Divide by Zero

int min, first, second;

min = 5;

first = 0;

second = min / first;

}

2)

void main () {

// Adding Boolean Values

boolean min, first, second;

min = false;

first = true;

second = first + min;

}

Explanation:

1. Dividing by zero does not make sense and even though the syntax and lexical analysis will not see an error, this does not logically make sense and will produce unwanted results.
2. The concrete syntax of Jay allows the user to add two identifiers, but adding two boolean values is not possible.