Gebze Technical University
Department of Computer Engineering

CSE 341 – Homework 3 Report
(PARSER)

DİLARA KARAKAŞ

171044010

In this problem, the first thing we need to look at is whether an error comes to us from the lexer of the given code. We check the given input token and see if there is an error in it. If there is an error, it directly ends the program. If not, it resolves the tokenized input with the parser tree.

ReadMe for Yacc

The program is run by entering the following 4 commands one by one in order:

```
# yacc -y -d gpp_interpreter.y
# flex gpp_lexer.l
# gcc gpp_lib.c lex.yy.c y.tab.c -o runInterpreter -g -lm
# ./runInterpreter
```

NOTE: While the code is being compiled, it may give warning (s), it does not prevent it from running.

NOTE: When Syntax gives an error, the program stops running.

(A makefile was not requested in the descriptions in PDF and moodle. A makefile was not made because the score was broken due to the presence of unwanted files in previous assignments.)

TEST FOR YACC

Deneme.txt

deneme.txt

```
(+ 2 3)
(+ 1 (* 5 3))
(and (or true false) (and true true))
(and (or true false) (and true false))
(and (or (and true false) (or true false)) true)
(list 1 2 3)
(set a (list 1 2))
(set b 1)
(append (list 10 11) (list 12 15))
(if (and true false) (+ 5 3) (- 5 3))
(if (and true true) (+ 5 3) (- 5 3))
(not (and (or (and true false) (or true false)) true))
```

output.txt

```
5
16
true
false
true
( 1 2 3 )
( 1 2 )
1
( 10 11 12 15 )
2
8
0
```

```
Dilara-MacBook-Air:yacc_interpreter dilarakarakas$ ./runInterpreter
(set a 23)
```

output.txt

```
23
```

TEST FOR LISP

Deneme.txt

deneme.txt

```
(+ 2 3)
(+ 1 (* 5 3))
(and (or true false) (and true true))
(and (or true false) (and true false))
(and (or (and true false) (or true false)) true)
(list 1 2 3)
(set a (list 1 2))
(set b 1)
(append (list 10 11) (list 12 15))
(if (and true false) (+ 5 3) (- 5 3))
(if (and true true) (+ 5 3) (- 5 3))
(not (and (or (and true false) (or true false)) true))
```

output.txt

```
5
16
1
0
1
(1 2 3)
(1 2)
1
(10 11 12 15)
2
8
0
```

```
Dilara-MacBook-Air:lisp_interpreter dilarakarakas$ clisp gpp_interpreter.lisp
(set a 4)

Dilara-MacBook-Air:lisp_interpreter dilarakarakas$
```

output.txt

```
4
```

```
Dilara-MacBook-Air:lisp_interpreter dilarakarakas$ clisp gpp_interpreter.lisp
(list 1 2 3)
(+ 4 5 6)

Dilara-MacBook-Air:lisp_interpreter dilarakarakas$
```

output.txt

```
SYNTAX_ERROR
```

We observe an error in the last test. Because the + operator work with two inputs