First Problem Set Assignment: Backus-Naur Form (BNF)

Abstract

The first problem set is intended to demonstrate Backus-Naur Form and its use in demonstrating the syntax of a programming language. Defining the grammar of a language demonstrates the ways in which a language is structured, any 'legal' combination of expressions that can be constructed in the confines of the language. Designing a parse tree allows for a visual representation of the syntax of the language, tracing the process from node to node to construct an expression in the language. Tasks 2 through 11 allowed for practice in constructing grammars and parse trees for predefined languages.

Task 1: BNF?

Backus-Naur Form is a way to logically reason the syntax of a programming language. It is a way to demonstrate how a language is interpreted by showing the permitted structures and expressions that are valid for that specific language. BNF is significant to programming because it demonstrates the structure and syntax of a language, shaping the functionality and limitations of a programming language.

Task 2

<string> ::= <+string> | <-string> | <empty>

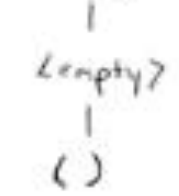
<+string> ::= +<+string> | + <string> | + <empty>

<-string> ::= -<-string> | -<string> | -<empty>

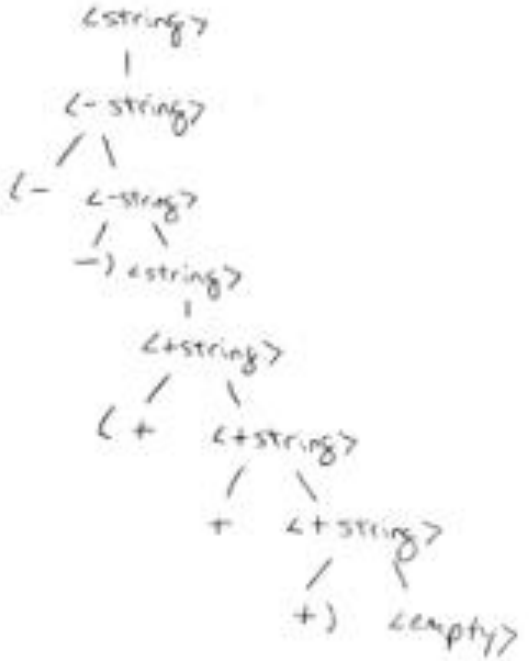
<empty> ::= ()<string> | ()

Task 3

1. ()



2. (--)(+++)



Task 4

<string> ::= <1> | <2> | <3> | <0>

<1> ::= 1<string> | 1<empty>

<2> ::= 2<string> | 2<empty>

<3> ::= 3<string> | 3<empty>

<0> ::= 0<string> | 0<empty>

<empty> ::= ' '

Task 5

0

<string>
|
<0>
/\
0 <empty>
|
' '

32100

<string>
|
<3>
/ \
3 <string>
|
<2>
/ \
2 <string>
|
<1>
/ \
1 <string>
|
<0>
/ \
0 <string>
|
<0>
/ \
0 <empty>
|
/ \

Task 6

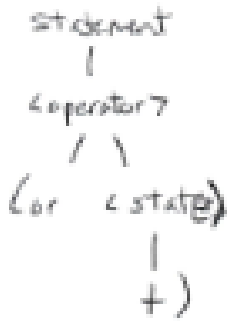
<statement> ::= <operator> | <state>

<operator> ::= (and <state>) | (or <state>) | (not <state>) |
(and <operator>) | (or <operator>) | (not <operator>)

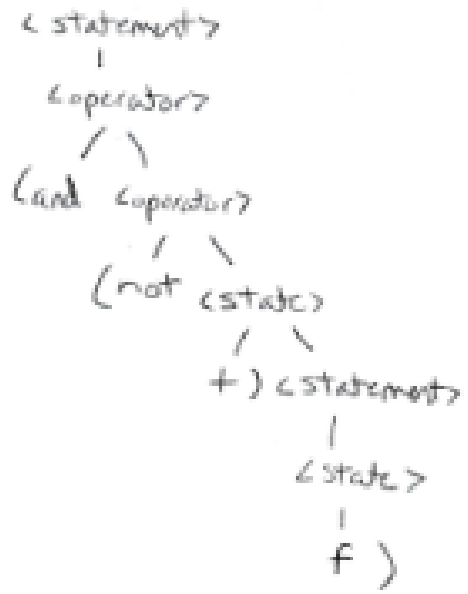
<state> ::= + | f | t <statement> | f <statement>

Task 7

(or +)



(and (not +) f)



Task 8

<Number> ::= <3digits> | <2digits> | <1digit>

<1digit> ::= one | two | three | four | five | six | seven | eight | nine | <teens>

<teens> ::= eleven | twelve | thirteen | fourteen | fifteen | sixteen | seventeen | eighteen | nineteen | ten

<2digits> ::= <tens><1digit> | <tens>

<tens> ::= twenty | thirty | forty | fifty | sixty | seventy | eighty | ninety

<3digits> ::= <1digit>hundred<2digits> | <1digit>hundred<empty>

Task 9

eighteen

<Number>
|
<1digit>
|
<teens>
|
eighteen

five hundred

<Number>
|
<3digits>
/ | \
<1digit> hundred <empty>
|
five

seven hundred fifty four

<Number>
|
<3digits>
/ | \
<1digit> hundred <2digits>
|
seven
/ \
<tens> <1digit>
| |
fifty four

Task 10

<Colorfun> ::= <describe> | <add> | <show> <colors>

<describe> ::= <colors> | <C#>

<C#> ::= <C1> | <C2> | <C3> | <C4> | <C28>

<show> ::= <colors> | <C#>

<add> ::= <N><N><N><colors> | <N><N><N><N> <color> |

<color> <C#> | <N><N><N> <C#>

<N> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 255

<Colors> ::= light red | red | purple

Task 11