CSC173 Project2 Parser

Fall 2023

No partner

Building instruction:

gcc -std=c99 -pedantic -Wall -Werror -o Parser *.c

Running instruction:

If you are using windows, double click on the “Parser.exe” file.

If you are using linux environment, type in “./Parser”.

On page2 is the parsing function for recursive-descendent parsing and parsing table for table-driven parsing.

On page3 is the symbol representation for parsing tree.

Recursive-Descendent:

Grammar:

Expr → Atom | List
Atom → Word | Number
Word → Letter WordT
WordT → Word | ε
Letter → a | · · · | z
Number → Digit NumberT
NumberT → Number | ε
Digit → 0 | · · · | 9
List → (Elements)
Elements → ε | Expr Rest
Rest → ε | Space Expr Rest

Table-Driven:

	a-z	0-9	(	)	(' ')	\0
E	2	2	1			
A	3	4				
W	4					
W	6			7	7	7
L	8					
N		9				
n		10		11	11	11
D		12				
L			13			
e	14	14	14			
R				17	16	
(			18			
)				19		
s					20	

Initial	Stack push(right to left)
(1) Expr	→ List
(2) Expr	→ Atom
(3) Atom	→ Word
(4) Atom	→ Number
(5) Word	→ Letter WordT
(6) WordT	→ Word
(7) WordT	→ ε
(8) Letter	→ l
(9) Number	→ Digit NumberT
(10) NumberT	→ Number
(11) NumberT	→ ε
(12) Digit	→ d
(13) List	→ (Elements)
(14) Elements	→ Expr Rest
(15) Elements	→ ε
(16) Rest	→ Space Expr Rest
(17) Rest	→ ε
(18) (	
(19))	
(20) ' '	

Parsing tree elements:

The symbols followed by “l” are letters (a-z), not other elements. The symbols followed by “d” are digits (0-9). All other small case letters are nodes.

Expr	E
Atom	A
Word	W
WordT	w
Letter	l
Number	N
NumberT	n
Digit	d
List	L
Elements	e
Rest	R
(	(
)	)
' '	s
ε	&