

Μεταγλωτιστές 2018

Προγραμματιστική Εργασία #2

Ονοματεπώνυμο : Ορέστης Καπλάνης

ΑΜ:Π2013139

Γραμματική :

Stmt_list -> Stmt Stmt_list | ε.

Stmt -> id = Expr | print Expr.

Expr -> Term Term_tail.

Term_tail -> First_priority Term Term_tail | ε.

Term -> Factor Factor_tail.

Factor_tail -> Second_priority Factor Factor_tail | ε.

Factor -> (Expr) | id | boolean.

First_priority -> not | ε.

Second_priority -> and | or .

FIRST sets

Stmt_list	id,print, ϵ
Stmt	id print
Expr	(,id,boolean
Term_tail	not, ϵ
Term	(,id,boolean
Factor_tail	ϵ ,and or
Factor	(,id,boolean
First_priority	ϵ ,not
Second_priority	and,or

FOLLOW sets

Stmt_list	\emptyset
Stmt	\emptyset
Expr)
Term_tail)
Term	not
Factor_tail	not
Factor	and ,or
First_priority)
Second_priority)

Η γραμματική είναι LL(1)

Factor_tail → Second_priority Factor Factor_tail
| ε.
Factor → (Expr)
| id
| boolean.
First_priority → not
| ε.
Second_priority → and
| or .

Line 7: Ignoring bad character & .

Some sentences generated by this grammar: {#949;, id id #949; #949; #949;., print id #949; #949; #949;., id (Expr) #949; #949; #949;., id boolean #949; #949; #949;., id id or id #949; #949; #949;., id id and id #949; #949; #949;., print (Expr) #949; #949; #949;., print boolean #949; #949; #949;., id (Expr) or id #949; #949; #949;., id id or (Expr) #949; #949; #949;., id id and (Expr) #949; #949; #949;., id (Expr) and id #949; #949; #949;., id id or boolean #949; #949; #949;., id id and boolean #949; #949; #949;., id (Expr) or (Expr) #949; #949; #949;., id (Expr) or boolean #949; #949; #949;., id (Expr) and (Expr) #949; #949; #949;., id (Expr) and boolean #949; #949; #949;., id boolean and (Expr) #949; #949; #949;.

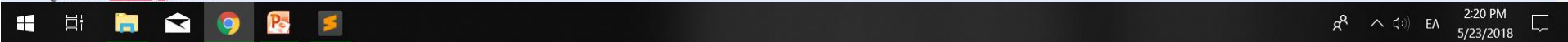
- All nonterminals are reachable and realizable.
- There are no nullable nonterminals.
- The endable nonterminals are: Stmt_list.
- No cycles.

nonterminal	first set	follow set	nullable	endable
Stmt_list	#949; id print	∅	no	yes
Stmt	id print	#949; id print	no	no
Expr	(Expr) id boolean	#949; id print	no	no
Term_tail	not #949;.	#949; id print	no	no
Term	(Expr) id boolean	not #949;.	no	no
Factor_tail	#949; and or	not #949;.	no	no
Factor	(Expr) id boolean	#949; and or	no	no
First_priority	not #949;.	(Expr) id boolean	no	no
Second_priority	and or	(Expr) id boolean	no	no

The grammar is not LL(1) because:

- Term_tail has a first set conflict.

- attempt to transform the grammar (to LL(1))
- generate LL(1) parsing table
- generate LR(0)/SLR(1) automaton
- generate LALR(1) automaton



Ο parser αναγνωρίζει boolean και logical τιμές καθώς επίσης και identifiers.

```
parser.py - C:\Users\oresp\Desktop\Μαθηματα\METAGLWTISTES-PYTHON\parser.py (3.6.5rc1)
File Edit Format Run Options Window Help

    else:
        raise ParseError("Expected identifier or print keyword")

def identifier(self):
    if self.la == "Identifier":
        print("found Identifier")
        self.match(self.la)
        self.logical()
    elif self.la == "Got a TRUE" or self.la == "Got a FALSE":
        print("found boolean Value")
        self.match(self.la)
        self.logical()
    else:
        raise ParseError("Expected identifier or boolean")

def equalSign(self):
    if self.la == "equalSign":
        print("found Equal_Sign")
        self.match(self.la)
        self.identifier()
    else:
        print(self.la)
        raise ParseError("Expected '=' ")

def logical(self):
    if self.la == "Got a NOT" or (self.la == "Got an OR" or self.la == "Got an A
        print("found logical")
        self.match(self.la)
        self.identifier()
    else:
        raise ParseError("Expected not, and, or")

# the main part of prog

# create the parser object
parser = MyParser()

# open file for parsing
with open("recursive.txt", "r") as fp:

    # parse file
    try:
        parser.parse(fp)
    except errors.PlexError:
```

Ln: 160 Col: 19

```
Python 3.6.5rc1 Shell
File Edit Shell Debug Options Window Help

Python 3.6.5rc1 (v3.6.5rc1:f03c5148cf, Mar 14 2018, 03:12:11) [MSC v.1913 64 bit (AMD64)] on
win32
Type "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\oresp\Desktop\Μαθηματα\METAGLWTISTES-PYTHON\parser.py ==
Got identifier
Parser Error: Expected '=' at line 1 char 4
>>>
== RESTART: C:\Users\oresp\Desktop\Μαθηματα\METAGLWTISTES-PYTHON\parser.py ==
Got identifier
Parser Error: Expected '=' at line 1 char 4
>>>
== RESTART: C:\Users\oresp\Desktop\Μαθηματα\METAGLWTISTES-PYTHON\parser.py ==
Got identifier
Got an EQUAL_SIGN
Parser Error: Expected '=' at line 1 char 4
>>>
== RESTART: C:\Users\oresp\Desktop\Μαθηματα\METAGLWTISTES-PYTHON\parser.py ==
Got identifier
Got an EQUAL_SIGN
Parser Error: Expected '=' at line 1 char 5
>>>
== RESTART: C:\Users\oresp\Desktop\Μαθηματα\METAGLWTISTES-PYTHON\parser.py ==
Got identifier
found Equal_Sign
found Identifier
Parser Error: Expected not, and, or at line 1 char 8
>>>
== RESTART: C:\Users\oresp\Desktop\Μαθηματα\METAGLWTISTES-PYTHON\parser.py ==
Got identifier
found Equal_Sign
found Identifier
found logical
found boolean Value
found logical
Parser Error: Expected identifier or boolean at line 1 char 13
>>>
== RESTART: C:\Users\oresp\Desktop\Μαθηματα\METAGLWTISTES-PYTHON\parser.py ==
Got identifier
Got an EQUAL_SIGN
Parser Error: Expected '=' at line 1 char 5
>>>
```

Ln: 31 Col: 16