

Μεταγλωττιστές 2019

Προγραμματιστική Εργασία #2

Ονοματεπώνυμο: Γιώργος Χαλαβαζής

ΑΜ: Π2016056

### Κανόνες γραμματικής

Grammar	
Stmt_list →	Stmt Stmt_list   .
Stmt →	id equal Expression   print Expression.
Expression →	Term Term_Tail.
Term_Tail →	xor Term Term_Tail   .
Term →	Factor Factor_Tail.
Factor_Tail →	or Factor Factor_Tail   .
Factor →	Atom Atom_Tail.
Atom_Tail →	and Atom Atom_Tail   .
Atom →	leftpar Expression rightpar   id   number.

Στην παραπάνω εικόνα παρουσιάζεται το σετ κανόνων γραμματικής που χρησιμοποιήθηκε για την υλοποίηση της εργασίας. Η γραμματική αυτή δέχεται συμβολοσειρές από bits και με τις λογικές πράξεις and, or και xor της εκτυπώνει αναλόγως. Επίσης για την γραμματική χρησιμοποιήθηκε αναδρομική κατάβαση οι λογικές πράξεις μπήκαν με την σειρά που αναφέρθηκε προηγουμένως. Επειδή σε αυτήν την γραμματική όσο πιο κάτω είναι ένας τελεστής τόσο πιο πολύ προτεραιότητα έχει. Για αυτό το and μπήκε στην factor\_Tail το or στην Factor\_Tail και το xor στην Atom\_Tail.

## Αποτελέσματα ελέγχου LL(1)

- All nonterminals are reachable and realizable.
- The nullable nonterminals are: Stmt\_list Term\_Tail Factor\_Tail Atom\_Tail.
- The endable nonterminals are: Atom\_Tail Atom Factor\_Tail Factor Term\_Tail Term Expression Stmt\_list Stmt.
- No cycles.

nonterminal	first set	follow set	nullable	endable
Stmt_list	id print	$\emptyset$	yes	yes
Stmt	id print	id print	no	yes
Term_Tail	xor	rightpar id print	yes	yes
Term	leftpar id number	rightpar xor id print	no	yes
Factor_Tail	or	rightpar xor id print	yes	yes
Factor	leftpar id number	rightpar or xor id print	no	yes
Atom_Tail	and	rightpar or xor id print	yes	yes
Atom	leftpar id number	rightpar and or xor id print	no	yes
Expression	leftpar id number	rightpar id print	no	yes

The grammar is LL(1).

Η γραμματική όπως φαίνεται παραπάνω είναι LL(1). Το διάβασμα γίνεται από τα αριστερά στα δεξιά και υπάρχει μονοπάτι για όλα τα μη τερματικά σύμβολα. Επίσης συμφωνεί με την μέθοδο την αναδρομικής κατάβασης διότι δεν υπάρχουν κυκλικές μεταβάσεις.

## Πίνακες με τα First και Follow Sets

nonterminal	first set	follow set
Stmt_list	id print	$\emptyset$
Stmt	id print	id print
Term_Tail	xor	rightpar id print
Term	leftpar id number	rightpar xor id print
Factor_Tail	or	rightpar xor id print
Factor	leftpar id number	rightpar or xor id print
Atom_Tail	and	rightpar or xor id print
Atom	leftpar id number	rightpar and or xor id print
Expression	leftpar id number	rightpar id print

Παραπάνω φαίνονται τα First και Follow sets. Κάποιες διευκρινήσεις: Το id είναι κάποιο γράμμα που βάζουμε στην γραμματική και το print είναι η λέξη print. Το σβησμένο σύμβολο είναι το ε και το leftpar και rightpar είναι η αριστερή και δεξιά παρένθεση αντίστοιχα. Με αυτά τα sets είναι

σε θέση ο συντακτικός αναλυτής να εφαρμόσει των σωστό κανόνα στην σωστή θέση.

### Αποτελέσματα εξόδου για έγκυρες και άκυρες μορφές εισόδου

#### Έγκυρα αποτελέσματα

##### Είσοδος

```
a = 1010101
b = 010 ^ 101
c = 01010101 & 10111011
print (01010 & 10101 | 01011 ^ 00110)
print a
print 1 | 0 ^ 1
print c
print b
```

---

##### Έξοδος

```
george2705@samsung-pc:~$ cd /
george2705@samsung-pc:/$ cd mnt
george2705@samsung-pc:/mnt$ cd c
george2705@samsung-pc:/mnt/c$ cd users
george2705@samsung-pc:/mnt/c/users$ cd xalavazis
george2705@samsung-pc:/mnt/c/users/xalavazis$ cd Desktop
george2705@samsung-pc:/mnt/c/users/xalavazis/Desktop$ python3 runner.py
1101
1010101
0
10001
111
george2705@samsung-pc:/mnt/c/users/xalavazis/Desktop$
```

---

#### Άκυρα αποτελέσματα

##### Είσοδος

```
a = 0110101 & 1011
b = 011 ^ 111
c = 192 | 011
```

---

##### Έξοδος

```
george2705@samsung-pc:/mnt/c/users/xalavazis/Desktop$ python3 runner.py
Traceback (most recent call last):
  File "runner.py", line 130, in <module>
    parser.parse(fp)
  File "runner.py", line 46, in parse
    self.stmt_list()
  File "runner.py", line 51, in stmt_list
    self.stmt_list()
  File "runner.py", line 51, in stmt_list
    self.stmt_list()
  File "runner.py", line 50, in stmt_list
    self.stmt()
  File "runner.py", line 61, in stmt
    e=self.expr()
  File "runner.py", line 71, in expr
    t=self.term()
  File "runner.py", line 84, in term
    t=self.factor()
  File "runner.py", line 98, in factor
    t = self.atom()
  File "runner.py", line 123, in atom
    self.match('BIT_TOKEN')
  File "runner.py", line 40, in match
    self.la,self.text=self.next_token()
  File "runner.py", line 36, in next_token
    return self.scanner.read()
  File "/home/george2705/.local/lib/python3.6/site-packages/plex/scanners.py", line 94, in read
    self.text, action = self.scan_a_token()
  File "/home/george2705/.local/lib/python3.6/site-packages/plex/scanners.py", line 138, in scan_a_token
    raise errors.UnrecognizedInput(self, self.state_name)
plex.errors.UnrecognizedInput: '', line 3, char 5: Token not recognised in state ''
```

## Πηγές

Δεν χρησιμοποιήθηκαν κάποιες πηγές για την δημιουργία αυτής της εργασίας.