Μεταγλωττιστές 2019

Προγραμματιστική Εργασία #2

Ονοματεπώνυμο: ΤΡΑΝΤΗ ΕΛΕΝΗ

ΑΜ: Π2016106

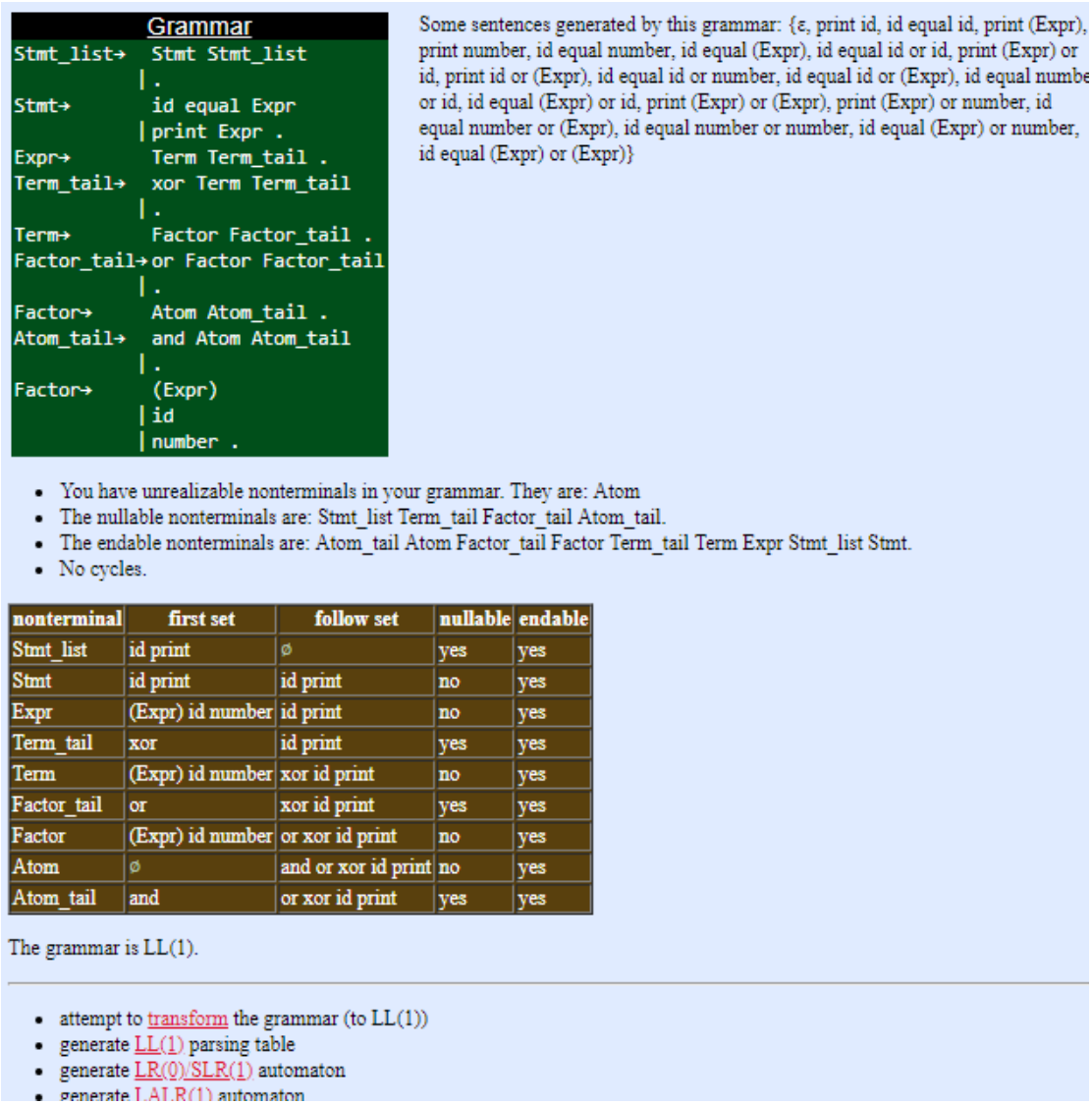
Κανόνες γραμματικής:

Grammar

```
Stmt_list→   Stmt Stmt_list
           | .
Stmt→        id equal Expr
           | print Expr .
Expr→        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
           | .
Factor→      (Expr)
           | id
           | number .
```

Η γραμματική της παραπάνω εικόνας είναι αυτή που χρησιμοποιήθηκε στην εργασία. Είναι αναδρομική κάτι που σημαίνει ότι υπάρχει σειρά προτεραιότητας: and, or, xor.

Αποτελέσματα ελέγχου LL(1) συμβατότητας:



Grammar

```
Stmt_list→   Stmt Stmt_list
           | .
Stmt→        id equal Expr
           | print Expr .
Expr→        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
           | .
Factor→      (Expr)
           | id
           | number .
```

Some sentences generated by this grammar: {ε, print id, id equal id, print (Expr), print number, id equal number, id equal (Expr), id equal id or id, print (Expr) or id, print id or (Expr), id equal id or number, id equal id or (Expr), id equal numbe or id, id equal (Expr) or id, print (Expr) or (Expr), print (Expr) or number, id equal number or (Expr), id equal number or number, id equal (Expr) or number, id equal (Expr) or (Expr)}

- You have unrealizable nonterminals in your grammar. They are: Atom
- 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 Expr Stmt_list Stmt.
- No cycles.

nonterminal	first set	follow set	nullable	endable
Stmt_list	id print	∅	yes	yes
Stmt	id print	id print	no	yes
Expr	(Expr) id number	id print	no	yes
Term_tail	xor	id print	yes	yes
Term	(Expr) id number	xor id print	no	yes
Factor_tail	or	xor id print	yes	yes
Factor	(Expr) id number	or xor id print	no	yes
Atom	∅	and or xor id print	no	yes
Atom_tail	and	or xor id print	yes	yes

The grammar is LL(1).

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

Η γραμματική είναι LL(1) όπως μας δείχνει και το αποτέλεσμα του παραπάνω εργαλείου.

Πίνακες με τα First και Follow sets:

nonterminal	first set	follow set	nullable	endable
Stmt_list	id print	∅	yes	yes
Stmt	id print	id print	no	yes
Expr	(Expr) id number	id print	no	yes
Term_tail	xor	id print	yes	yes
Term	(Expr) id number	xor id print	no	yes
Factor_tail	or	xor id print	yes	yes
Factor	(Expr) id number	or xor id print	no	yes
Atom	∅	and or xor id print	no	yes
Atom_tail	and	or xor id print	yes	yes

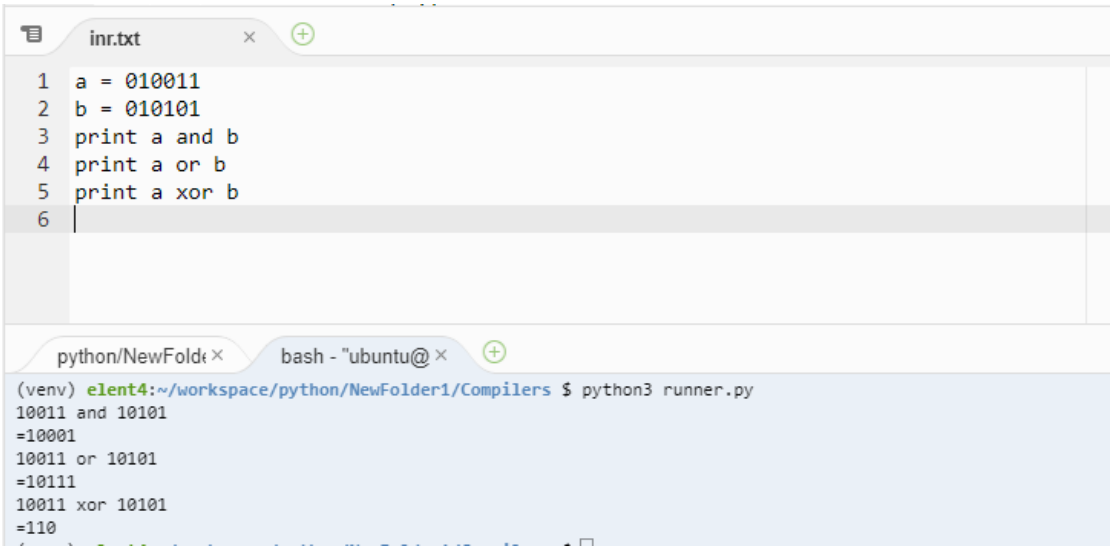
The grammar is LL(1).

Αποτελέσματα εξόδου:

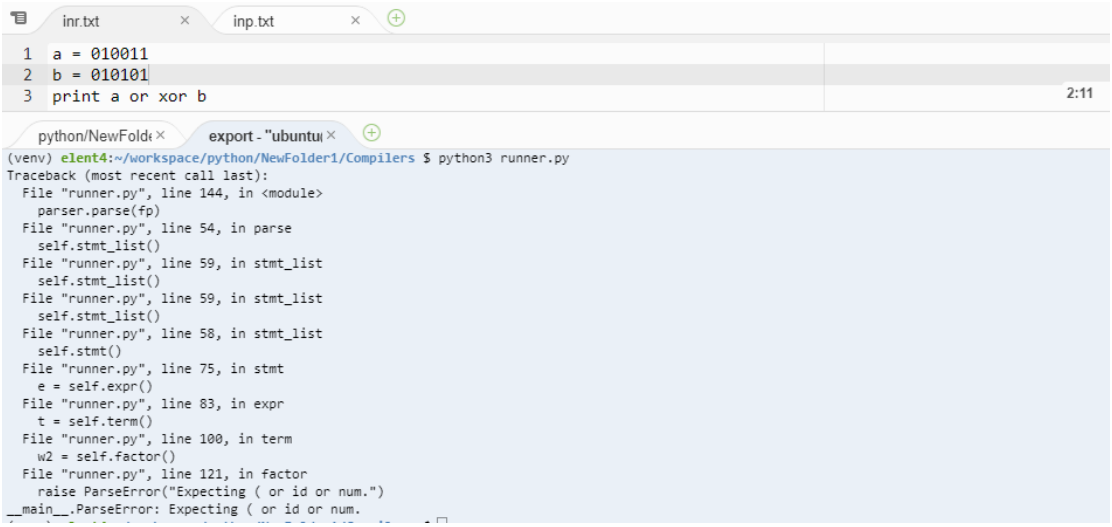
Έγκυρη είσοδος και έξοδος του parser.py.

Άκυρη είσοδος και έξοδος του parser.py.

Έγκυρη είσοδος και έξοδος του runner.py.



Άκυρη είσοδος και έξοδος του runner.py.



Πηγές:

Cloud9: https://ide.c9.io