Μεταγλωτιστές 2019

Προγραμματιστική Εργασία #2

Ονοματεπώνυμο : ΕΥΑΓΓΕΛΟΣ ΚΕΡΑΜΙΔΑΣ

ΑΜ : Π2016095

1) **Κανόνες της γραμματικής**

Grammar

```
Stmt_list →   Stmt Stmt_list
            | .
Stmt →        identifier 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
            | .
Atom →        ( Expr )
            | identifier
            | bit_token.
```

Some sentences generated by this grammar: {ε, print bit_token, print identifier, print (bit_token), print (identifier), identifier equal bit_token, identifier equal identifier, identifier equal (bit_token), identifier equal (identifier), print identifier print identifier, identifier equal bit_token print bit_token, identifier equal bit_token print identifier, identifier equal identifier print bit_token, print identifier identifier equal bit_token, identifier equal identifier print identifier, print identifier identifier equal identifier, identifier equal bit_token identifier equal bit_token, identifier equal bit_token identifier equal identifier, identifier equal identifier identifier equal bit_token, identifier equal identifier identifier equal identifier}

- You have unrealizable nonterminals in your grammar. They are: XOR OR AND
- 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.

2) **Αποτελέσματα ελέγχου για LL(1) συμβατότητα**

Και

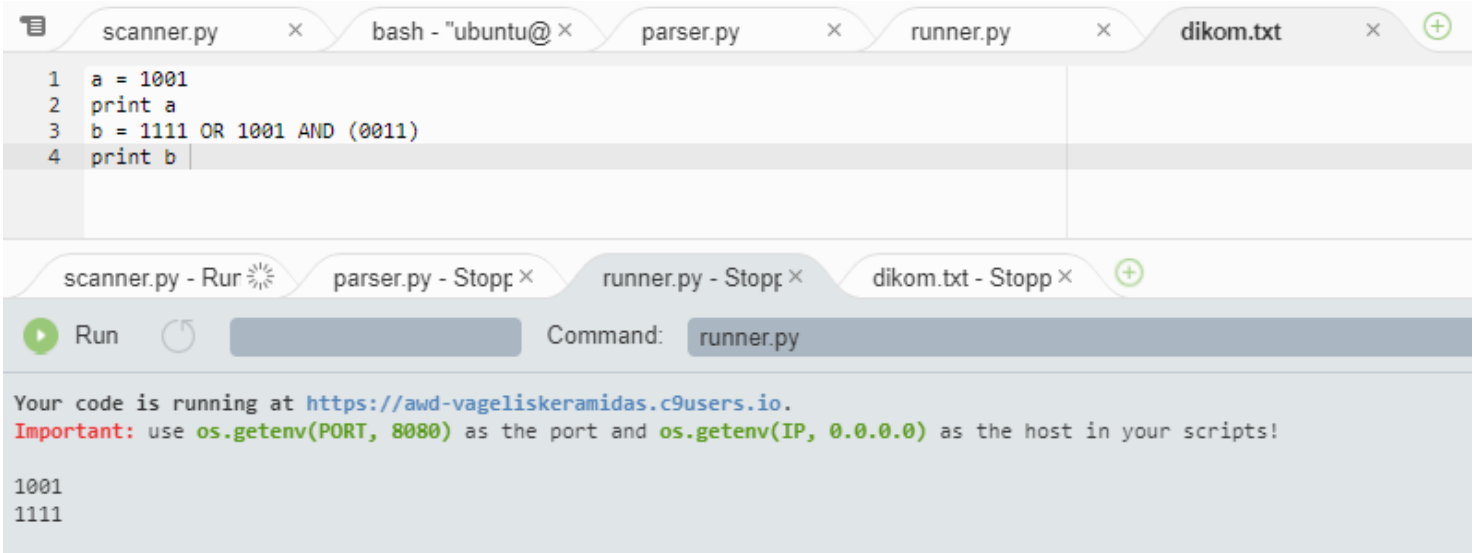
Πίνακας με τα FIRST και FOLLOW sets για όλα τα μη τερματικά σύμβολα

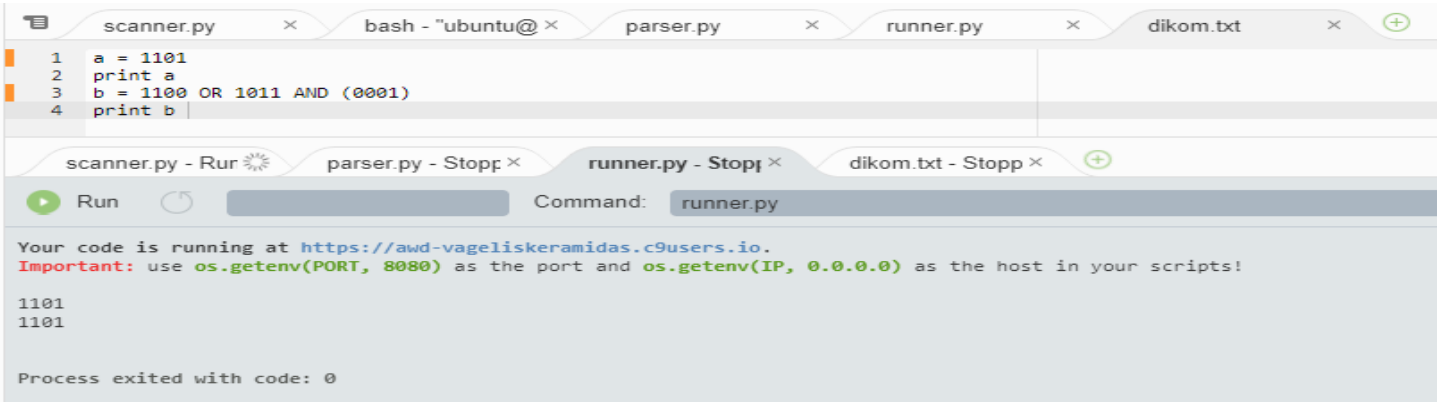
nonterminal	first set	follow set	nullable	endable
Stmt_list	identifier print	∅	yes	yes
Stmt	identifier print	identifier print	no	yes
XOR	∅	∅	no	no
Term_tail	∅	) identifier print	yes	yes
Term	(identifier bit_token	) identifier print	no	yes
OR	∅	∅	no	no
Factor_tail	∅	) identifier print	yes	yes
Factor	(identifier bit_token	) identifier print	no	yes
AND	∅	∅	no	no
Atom_tail	∅	) identifier print	yes	yes
Atom	(identifier bit_token	) identifier print	no	yes
Expr	(identifier bit_token	) identifier print	no	yes

The grammar is LL(1).

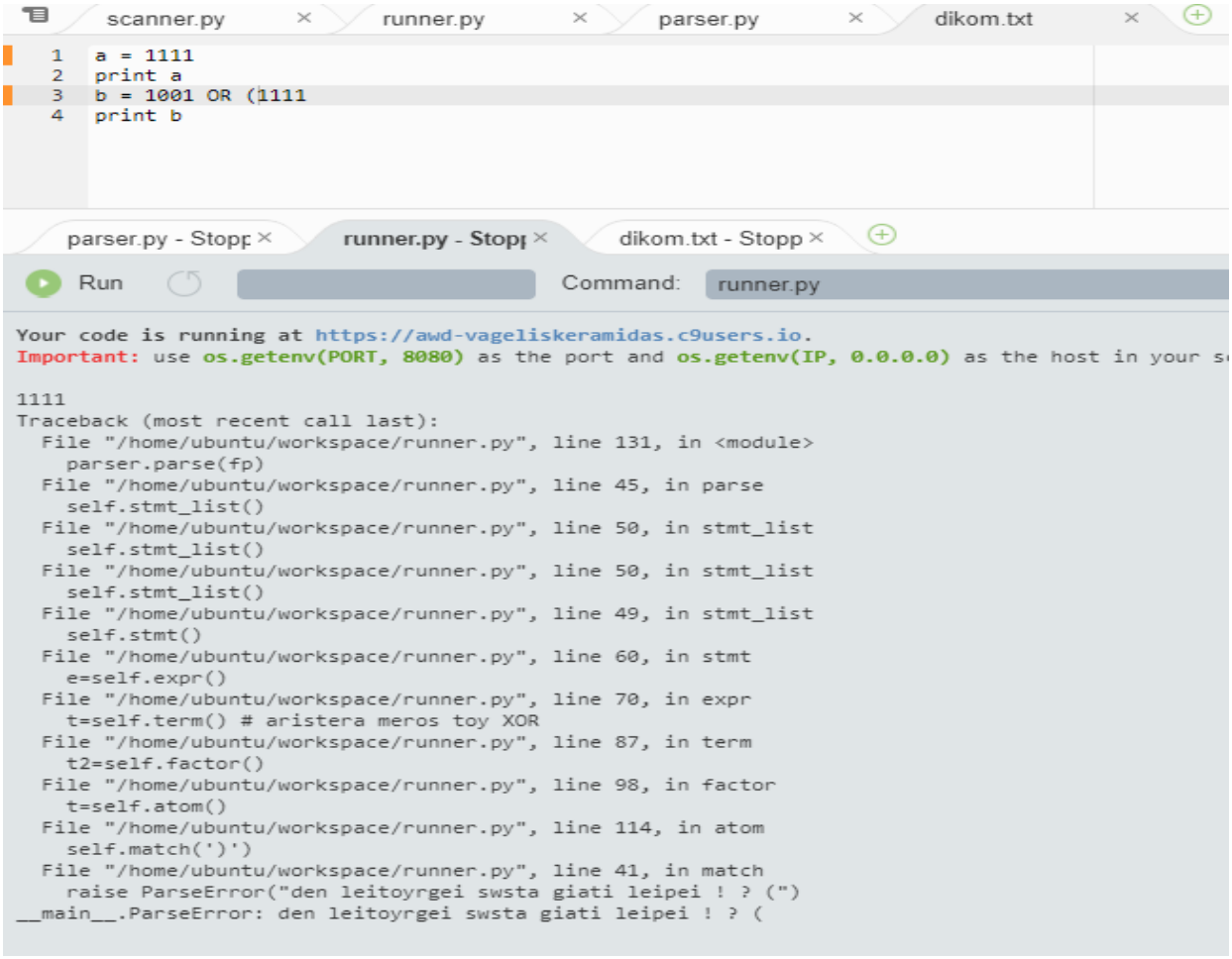
4) **Αποτελέσματα εξόδου για έγκυρες και άκυρος μορφές εισόδου**

Έγκυρες μορφές εισοδου :





Άκυρες μορφές εισόδου :



5) **Πηγές που χρησιμοποιήθηκαν για την πρόοδο της εργασιας**

- **Google.gr**
- **Youtube.com**
- **C9.io**
- **http://www.csd.uwo.ca/~moreno/CS447/Lectures/Syntax.html/node14.html**
-