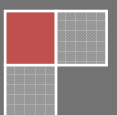


E1-10

Κατασκευή Λεκτικού- Συντακτικού Αναλυτή για την γλώσσα mini-Python

Εργαστήριο Μεταγλωττιστών
Χειμερινό εξάμηνο 2017-18

Ομάδα E1-10
Ίρις-Αικατερίνη Διαμαντή - 040038
Ευάγγελος Χουχούμης - 050129



Περιεχόμενα

0 - Το αλφάβητο της γλώσσας mini-Python	3
1 - Λεκτικός Αναλυτής	4
1.1 - Αριθμοί	4
1.2 - Συμβολοσειρές	8
1.3 - Μεταβλητές	10
1.4 - Τελεστές	11
1.5 - Σχόλια	13
1.6 - Ολοκληρωμένο ΠΠΑ	15
1.7 - Πηγαίος Κώδικας ΛΑ	

0 - Το αλφάβητο της γλώσσας mini-Python

Η γλώσσα mini-Python είναι μια ψευδογλώσσα βασισμένη στη γνωστή γλώσσα προγραμματισμού Python 3. Το αλφάβητό της αποτελείται από όλον τον ASCII κώδικα εκτός των χαρακτήρων \$, ? τα οποία όμως επιτρέπεται να εμφανίζονται μέσα σε κυριολεκτικές συμβολοσειρές (strings). Η γλώσσα κάνει διαχωρισμό μεταξύ πεζών και κεφαλαίων (case sensitive language).

Σύνολο χαρακτήρων αλφαβήτου (σε Extended BNF):

Γράμματα:

```
lower ::= a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z
upper ::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
```

Ψηφία:

```
digit ::= 0|1|2|3|4|5|6|7|8|9
```

Ειδικοί χαρακτήρες:

```
lower ::= a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z
upper ::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
digit ::= 0|1|2|3|4|5|6|7|8|9
ordinary ::= _|(|)|[|]|{|}|+|-|*|/|%|!|&|'|~|^|<|=|>|,|.|:|;|$|?|#
special ::= '|"|\
```

Διαχωριστές:

```
white_spaces ::= \s|\t|\n (spaces, tabs, newline)
```

1 - Λεκτικός Αναλυτής

Ένα πηγαίο πρόγραμμα mini-Python διαβάζεται από έναν Λεκτικό Αναλυτή - ΛΑ (lexical analyzer) ως μια μεγάλη συμβολοσειρά εισόδου. Στη συνέχεια, ο ΛΑ διαχωρίζει ένα-ένα τα λεξήματα που περιέχονται μέσα στη συμβολοσειρά εισόδου κάθε φορά που του στέλνει ένα αίτημα ο Συντακτικός Αναλυτής - ΣΑ (parser) και στη συνέχεια τα αναγνωρίζει ως λεκτικές μονάδες. Η διαδικασία διαχωρισμού καλείται και scanning και ο ΛΑ αντίστοιχα scanner.

1.1 - Αριθμοί

Οι αριθμοί στην mini-Python είναι δύο τύπων: οι ακέραιοι (integers) και οι αριθμοί κινητής υποδιαστολής (floating point). Όλοι οι αριθμοί μπορεί να είναι προσημασμένοι ή μη.

Ακέραιοι αριθμοί

Ένας ακέραιος αποτελείται από ένα ή περισσότερα ψηφία 0-9. Ένας ακέραιος με μήκος 2 ή μεγαλύτερο δεν μπορεί να αρχίζει από 0.

Κανονική έκφραση: **INTEGER** = $[+-]? (0 | [1-9] [0-9]^*)$

Η παραπάνω κανονική έκφραση είναι μια έγκυρη εκδοχή σημειογραφίας eBNF. Εναλλακτικά θα μπορούσαμε να την διατυπώσουμε ως εξής:

INTEGER = $["+" | "-"], ("0" | \text{digit} - "0", \{\text{digit}\})$;

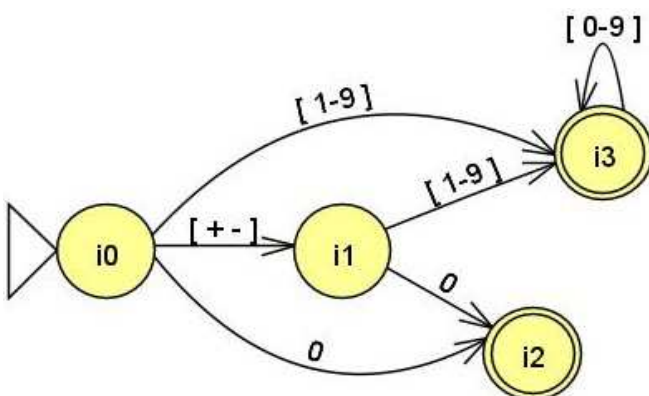
Αποδεκτοί ακέραιοι: 0, -0, +0, -23871, +2132, 288200

Μη αποδεκτοί ακέραιοι: 00, 0023, --324, +-87, ++0

Πεπερασμένο Προσδιοριστικό Αυτόματο (ΠΠΑ) αναγνώρισης ακέραιων:

Διάγραμμα Μετάβασης (ΔΜ):

Πίνακας Μετάβασης (ΠΜ):



	0	[+-]	[1-9]	exit
I0	I2	I1	I3	
I1	I2		I3	
I2				OK
I3	I3		I3	OK

Αρχείο εισόδου fsm:

```

I0:  0      -> I2
      + \-   -> I1
      1-9    -> I3
I1:  0      -> I2
      1-9    -> I3
I2:  \n     -> GOOD
I3:  0-9    -> I3
      \n     -> GOOD
GOOD (OK) :

```

```

C:\Users\win7vm\fsm>fsm integer.fsm
0
^Z
YES

C:\Users\win7vm\fsm>fsm integer.fsm
-0
^Z
YES

C:\Users\win7vm\fsm>fsm integer.fsm
384712
^Z
YES

C:\Users\win7vm\fsm>fsm integer.fsm
-312023
^Z
YES

C:\Users\win7vm\fsm>fsm integer.fsm
00
fsm: in integer.fsm, state 'i2' input 0 not accepted

C:\Users\win7vm\fsm>fsm integer.fsm
00224
fsm: in integer.fsm, state 'i2' input 0 not accepted

C:\Users\win7vm\fsm>fsm integer.fsm
++45
fsm: in integer.fsm, state 'i1' input + not accepted

```

Σημειώσεις:

Σε μια αριθμητική έκφραση όπως: $-5+-4$ ο υπεύθυνος για να αναγνωρίσει ότι το + είναι ο τελεστής και όχι ότι το 4 έχει δύο πρόσημα και άρα είναι λάθος, είναι ο συντακτικός αναλυτής.

Στο αρχείο fsm χρησιμοποιούμε τον χαρακτήρα καινούριας γραμμής ($\backslash n$) ως μετάβαση στην σωστή έξοδο. Η λανθασμένη έξοδος προκύπτει από την απουσία σωστής εξόδου, άρα δεν χρησιμοποιούμε καταστάσεις bad. Η μόνη εξαίρεση, όπως θα δούμε παρακάτω είναι στην περίπτωση σχολίων με τριπλά " .

Αριθμοί κινητής υποδιαστολής

Ένας αριθμός κινητής υποδιαστολής αποτελείται από ένα ακέραιο μέρος (ακέραιος αριθμός) και ένα δεκαδικό μέρος (ακολουθία από ψηφία 0-9) που διαχωρίζονται από μια τελεία. Ένα από τα δύο μέρη μπορεί να λείπει, αλλά όχι και τα δύο μαζί. Επιτρέπεται η χρήση του συμβόλου *e* που δηλώνει δύναμη του 10 και αρμόδιος να το γνωρίζει και να το διαχειριστεί είναι ο σημασιολογικός αναλυτής.

Κανονική έκφραση: $\text{FLOAT} = [+ -] ? ([0-9] + [.] [0-9] * | [.] [0-9] +) ([eE] [+ -] ? [0-9] +) ?$

Εναλλακτική σημειογραφία eBNF:

FLOAT =

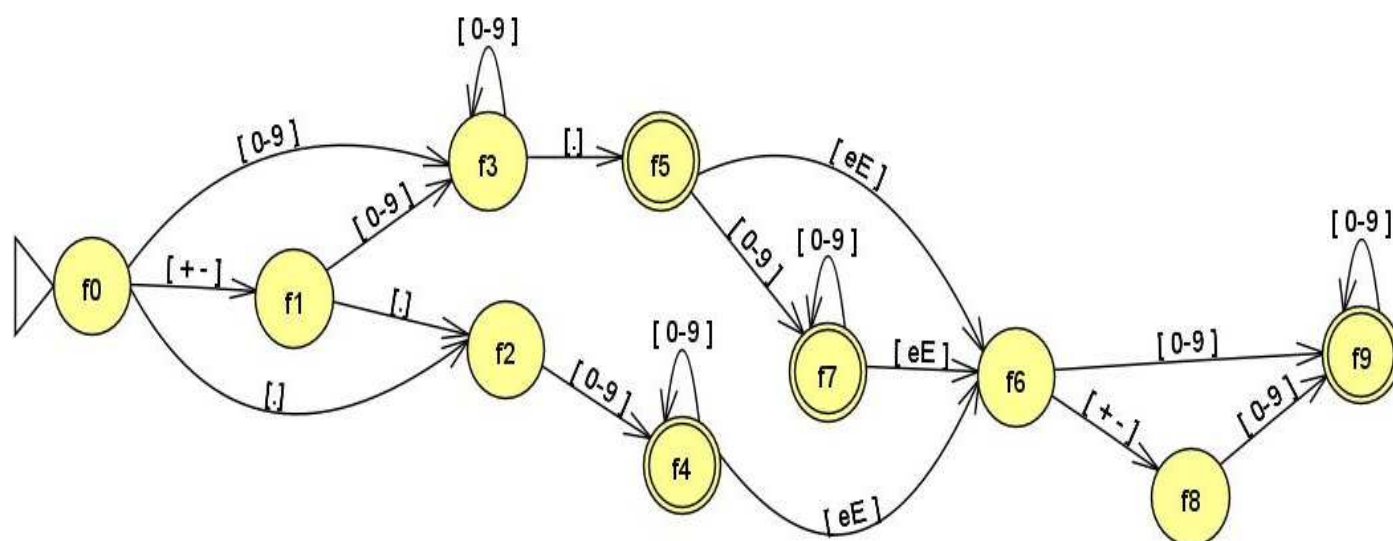
`["+" | "-"] (digit , { digit } , "." , { digit } | "." , digit , { digit }) [("e" | "E") , ("+" | "-") , digit , { digit }] ;`

Αποδεκτοί αριθμοί κινητής υποδιαστολής: -0., +.092, 000.2300, 1.e-8, .0E+2, 23.66

Μη αποδεκτοί αριθμοί κινητής υποδιαστολής: ., .e5, +4e-3, -e-2

ΠΠΑ αναγνώρισης αριθμών κινητής υποδιαστολής:

Διάγραμμα Μετάβασης (ΔΜ):



Πίνακας Μετάβασης (ΠΜ):

	[0-9]	[+ -]	[.]	[eE]	exit
F0	F3	F1	F2		
F1	F3		F2		
F2	F4				
F3	F3		F5		
F4	F4			F6	OK
F5	F7			F6	OK
F6	F9	F8			
F7	F7			F6	OK
F8	F9				
F9	F9				OK

Αρχείο εισόδου fsm:

```

F0:  0-9  -> F3
      + \- -> F1
      .    -> F2
F1:  0-9  -> F3
      .    -> F2
F2:  0-9  -> F4
F3:  0-9  -> F3
      .    -> F5
F4:  0-9  -> F4
      e E  -> F6
      \n   -> GOOD
F5:  e E  -> F6
      0-9  -> F7
      \n   -> GOOD
F6:  0-9  -> F9
      + \- -> F8
F7:  0-9  -> F7
      e E  -> F6
      \n   -> GOOD
F8:  0-9  -> F9
F9:  0-9  -> F9
      \n   -> GOOD
GOOD (OK) :

```

```

C:\Users\win7vm\fsm>fsm float.fsm
0.00
^Z
YES
C:\Users\win7vm\fsm>fsm float.fsm
-.33000e+4
^Z
YES
C:\Users\win7vm\fsm>fsm float.fsm
+8.
^Z
YES
C:\Users\win7vm\fsm>fsm float.fsm
34.E-33
^Z
YES
C:\Users\win7vm\fsm>fsm float.fsm
000000.12e86
^Z
YES
C:\Users\win7vm\fsm>fsm float.fsm
-.e33
fsm: in float.fsm, state 'f2' input e not accepted
C:\Users\win7vm\fsm>fsm float.fsm
0e44
fsm: in float.fsm, state 'f3' input e not accepted
C:\Users\win7vm\fsm>fsm float.fsm
234e-5
fsm: in float.fsm, state 'f3' input e not accepted

```

Σημειώσεις:

Τα περιττά μηδενικά μπροστά και πίσω από τον αριθμό αναγνωρίζονται από τον σημασιολογικό αναλυτή.

1.2 - Συμβολοσειρές

Μια συμβολοσειρά είναι μια ακολουθία από χαρακτήρες μέσα από το αλφάβητο της γλώσσας, που περικλείονται μεταξύ μονών ή διπλών εισαγωγικών. Υπάρχει δυνατότητα χρήσης ελληνικών χαρακτήρων.

Κανονική έκφραση:

STRING = "[\x9\x20\x21\x23-\x7E\p{Greek}]*" | '[\x9\x20-\x26\x28-\x7E\p{Greek}]*'

Σημειογραφία eBNF:

STRING =
("'", {"\t" | "\s" | upper | lower | digit | ordinary}, "'") | '"', {"\t" | "\s" | upper | lower | digit | ordinary}, '"');

Αποδεκτές συμβολοσειρές:

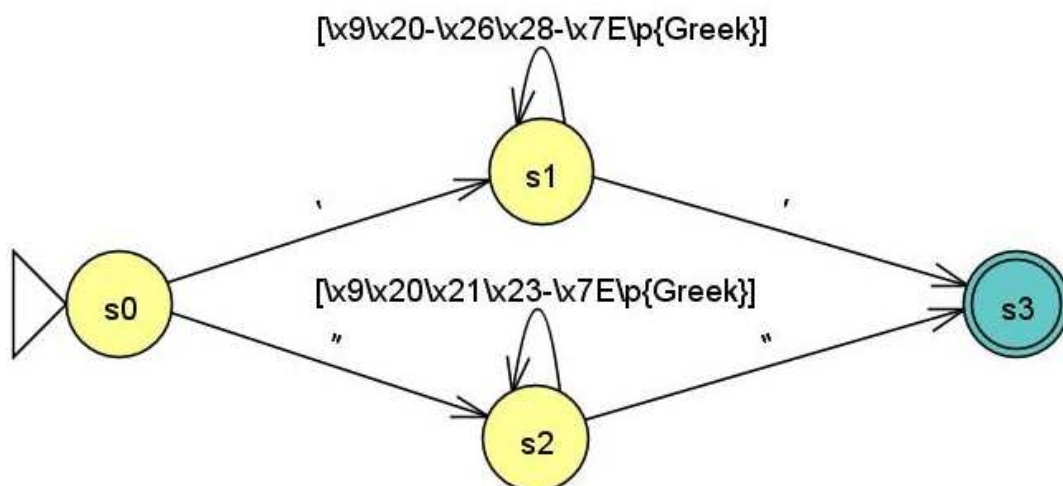
"Κώστας", "Hello there!", 'Goodbye "Argentina" !', "Your number is '34375'."

Μη αποδεκτές συμβολοσειρές:

"Κώστας , Hello there!", 'Goodbye 'Argentina' !', "Your number is "34375"."

ΠΠΑ αναγνώρισης συμβολοσειρών:

Διάγραμμα Μετάβασης (ΔΜ):



Πίνακας Μετάβασης (ΠΜ):

	'	"	[\x9\x20-\x26\x28-\x7E\p{Greek}]	[\x9\x20\x21\x23-\x7E\p{Greek}]	exit
S0	S1	S2			
S1	S3		S1		
S2		S3		S2	
S3					OK

Αρχείο εισόδου fsm:

```

S0:  '      -> S1
     "      -> S2
S1:  a-z A-Z 0-9 \s -> S1
     '      -> S3
S2:  a-z A-Z 0-9 \s -> S2
     "      -> S3
S3:  \n      -> GOOD
GOOD (OK) :

```

```

C:\Users\win7vm\fsm>fsm string.fsm
"Hello David"
^Z
YES

C:\Users\win7vm\fsm>fsm string.fsm
'Goodbye world'
^Z
YES

C:\Users\win7vm\fsm>fsm string.fsm
"heeyy""
fsm: in string.fsm, state 's3' input " not accepted

C:\Users\win7vm\fsm>fsm string.fsm
''heey'
fsm: in string.fsm, state 's3' input h not accepted

C:\Users\win7vm\fsm>fsm string.fsm
''
^Z
YES

C:\Users\win7vm\fsm>fsm string.fsm
""
^Z
YES

```

Σημειώσεις:

Στην κανονική έκφραση χρησιμοποιούμε τους δεκαεξαδικούς κώδικες του πίνακα ASCII για να δηλώσουμε ποιά σύμβολα θέλουμε και ποιά όχι. Στην περίπτωση των διπλών εισαγωγικών το αυτόματο δέχεται τα σύμβολα από το x20 ως το x7E, εκτός από το x22 που είναι το σύμβολο των διπλών εισαγωγικών ("). Στην περίπτωση των μονών εισαγωγικών γίνεται το αντίστοιχο - δεν δέχεται το σύμβολο x27 που είναι το σύμβολο '. Και στις δύο περιπτώσεις συμπεριλάβαμε και το x9 που είναι το TAB. Όλα τα παραπάνω καταλήγουν στο να μην γίνονται αποδεκτές συμβολοσειρές που εμπεριέχουν το είδος των εισαγωγικών που χρησιμοποιούν για να αυτοπροσδιοριστούν. Εκτός αν χρησιμοποιηθεί χαρακτήρας διαφυγής \, όμως η αναγνώριση αυτή είναι καθήκον του συντακτικού αναλυτή.

Η χρήση του \p{Greek} γίνεται για οικονομία χώρου και για ευκολία στην ανάγνωση. Κανονικά χρησιμοποιούμε τους ελληνικούς χαρακτήρες του κώδικα UTF-8: [\x{0386}–\x{03CE}]

Τέλος το αρχείο fsm είναι απλοποιημένο για τους ίδιους λόγους.

1.3 - Μεταβλητές

Οι μεταβλητές αποτελούν συμβολικά ονόματα θέσεων μνήμης. Το όνομα μιας μεταβλητής μπορεί να περιλαμβάνει λατινικούς χαρακτήρες a-z, A-Z, αριθμούς 0-9 και κάτω παύλα (_). Το όνομα μιας μεταβλητής δεν μπορεί να αρχίζει με αριθμητικό ψηφίο. Στο όνομα ενός αναγνωριστικού διακρίνονται τα πεζά από τα κεφαλαία (case-sensitive). Τέλος μπορεί να περιλαμβάνει ελληνικούς χαρακτήρες.

Κανονική έκφραση: **VAR** = [_a-zA-Z\p{Greek}] [_0-9a-zA-Z\p{Greek}] *

Σημειογραφία eBNF: **VAR** = ("_" | lower | upper) , { "_" | digit | lower | upper } ;

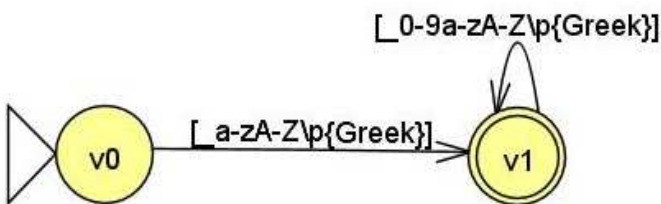
Αποδεκτά ονόματα μεταβλητών: X, αβ, sdfg, A23fv, _var1, _ονομα55, _3847_

Μη αποδεκτά ονόματα μεταβλητών: 3ονομα, 94var1, 0_var, \$par, #_k8, *w33

ΠΠΑ αναγνώρισης μεταβλητών:

Διάγραμμα Μετάβασης (ΔΜ):

Πίνακας Μετάβασης (ΠΜ):



	[_a-zA-Z\p{Greek}]	[_0-9a-zA-Z\p{Greek}]	exit
V0	V1		
V1		V1	OK

Αρχείο εισόδου fsm:

```

V0:  _ a-z A-Z  -> V1
V1:  _ a-z A-Z 0-9  -> V1
      \n      -> GOOD
GOOD (OK) :
```

```

C:\Users\win7um\fsm>fsm var.fsm
0var0
^Z
YES

C:\Users\win7um\fsm>fsm var.fsm
test148_
^Z
YES

C:\Users\win7um\fsm>fsm var.fsm
i
^Z
YES

C:\Users\win7um\fsm>fsm var.fsm
55var_
fsm: in var.fsm, state 'v0' input 5 not accepted

C:\Users\win7um\fsm>fsm var.fsm
$k23
fsm: in var.fsm, state 'v0' input $ not accepted
```

1.4 - Τελεστές

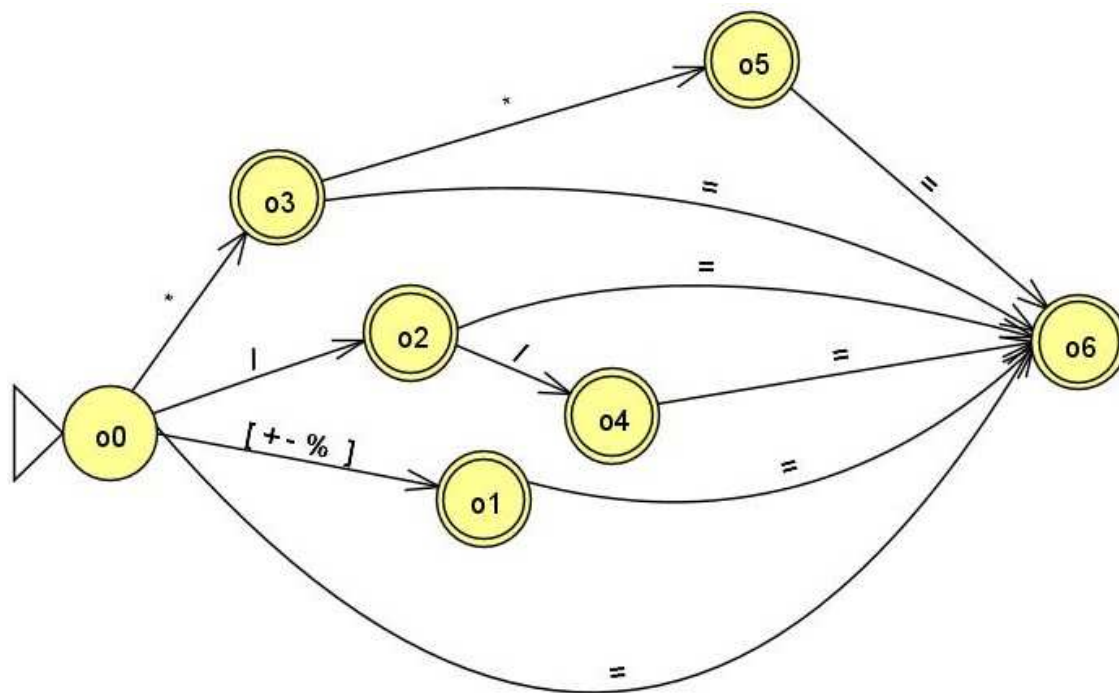
Οι αριθμητικοί τελεστές χρησιμοποιούνται για πράξεις μεταξύ δύο αριθμών (τελεστέων) και είναι οι ακόλουθοι: +, -, *, /, %, //, **. Στις συμβολοσειρές μπορεί να εφαρμοστεί η πράξη της συνένωσης με τον τελεστή +. Οι τελεστές εκχώρησης είναι: =, +=, -=, *=, /=, % =, //=, **=.

Κανονική έκφραση: $\text{OPERATOR} = \backslash = | ([+\backslash-\backslash\%] | \backslash * \{1, 2\} | \backslash / \{1, 2\}) \backslash = ?$

Σημειογραφία eBNF: $\text{OPERATOR} = "=" | ("+" | "-" | "\%" | "**", ["*"] | "/" , ["/"]) , ["="]$

ΠΠΑ αναγνώρισης τελεστών:

Διάγραμμα Μετάβασης (ΔΜ):



Πίνακας Μετάβασης (ΠΜ):

	*	/	[+-%]	=	exit
o0	o3	o2	o1	o6	
o1				o6	OK
o2		o4		o6	OK
o3	o5			o6	OK
o4				o6	OK
o5				o6	OK
o6					Ok

Αρχείου εισόδου fsm:

```

00:  \*    -> 03
      /     -> 02
      + \- %    -> 01
      =     -> 06
01:  =     -> 06
      \n    -> GOOD
02:  /     -> 04
      \n    -> GOOD
03:  \*    -> 05
      =     -> 06
      \n    -> GOOD
04:  =     -> 06
      \n    -> GOOD
05:  =     -> 06
      \n    -> GOOD
06:  \n    -> GOOD
GOOD (OK) :

```

```

C:\Users\win7vm\fsm>fsm operator.fsm
+
^Z
YES
C:\Users\win7vm\fsm>fsm operator.fsm
**
^Z
YES
C:\Users\win7vm\fsm>fsm operator.fsm
//=
^Z
YES
C:\Users\win7vm\fsm>fsm operator.fsm
%=
^Z
YES
C:\Users\win7vm\fsm>fsm operator.fsm
--
fsm: in operator.fsm, state 'o1' input - not accepted
C:\Users\win7vm\fsm>fsm operator.fsm
%%
fsm: in operator.fsm, state 'o1' input % not accepted
C:\Users\win7vm\fsm>fsm operator.fsm
+*
fsm: in operator.fsm, state 'o1' input * not accepted
C:\Users\win7vm\fsm>fsm operator.fsm
++
fsm: in operator.fsm, state 'o1' input + not accepted

```

1.5 - Σχόλια

Τα σχόλια αρχίζουν με το σύμβολο # και συνεχίζουν μέχρι το τέλος της γραμμής. Αν επιθυμούμε σχόλια που να επεκτείνονται σε πολλές γραμμές θα πρέπει να τα περικλείσουμε μεταξύ δύο τριάδων διπλών εισαγωγικών """. Υπάρχει δυνατότητα χρήσης ελληνικών χαρακτήρων.

Κανονική έκφραση:

COMMENT = `#[\x20-\x7E\p{Greek}]*|\{"{3}[\xA\x20-\x7E\p{Greek}]*\{"{3}`

Σημειογραφία eBNF: **COMMENT** =

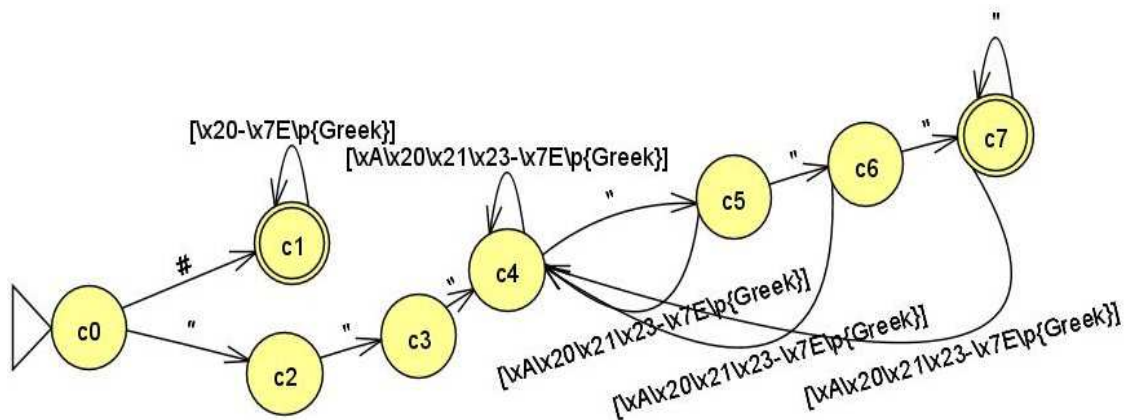
`"#", {lower|upper|digit|ordinary|special|"\"s"|"\"t"}|'','','',{lower|upper|digit|ordinary|special|"\"s"|"\"t"|"\"n"}, '','','';`

Αποδεκτά σχόλια: # This is a comment, ""This is a loooo
ooooooooong comment""

Μη αποδεκτά σχόλια: # This is a looooo
ooooooooong comment
"" This is a comment""

ΠΠΑ αναγνώρισης σχολίων:

Διάγραμμα Μετάβασης (ΔΜ):



Πίνακας Μετάβασης (ΠΜ):

	#	"	<code>[\x20-\x7E\p{Greek}]</code>	<code>[\xA\x20\x21\x23-\x7E\p{Greek}]</code>	exit
c0	c1	c2			
c1			c1		OK
c2		c3			
c3		c4			
c4		c5		c4	
c5		c6		c4	
c6		c7		c4	
c7		c7		c4	OK

Αρχείο εισόδου fsm:

```

C0:  #      -> C1
     "      -> C2
C1:  a-z A-Z 0-9 \s -> C1
     \n     -> GOOD
C2:  "      -> C3
C3:  "      -> C4
C4:  a-z A-Z 0-9 \s \n -> C4
     "      -> C5
C5:  a-z A-Z 0-9 \s \n -> C4
     "      -> C6
C6:  a-z A-Z 0-9 \s \n -> C4
     "      -> C7
C7:  a-z A-Z 0-9 \s \n -> C4
     "      -> C7
     EOF    -> GOOD
GOOD (OK) :

```

```

C:\Users\win7vm\fsm>fsm comment.fsm
# This is a comment
^Z
YES

C:\Users\win7vm\fsm>fsm comment.fsm
"" This is a loooo
ooooooooooooooooo
ooooooooooooooooooooong
comment ""^Z
^Z
YES

C:\Users\win7vm\fsm>fsm comment.fsm
# This is an incorrect
comment
fsm: in comment.fsm, state 'good' input c not accepted

C:\Users\win7vm\fsm>fsm comment.fsm
"" this is an
fsm: in comment.fsm, state 'c3' input \s not accepted

C:\Users\win7vm\fsm>fsm comment.fsm
"" this is
an invalid comment
""^Z
^Z
NO

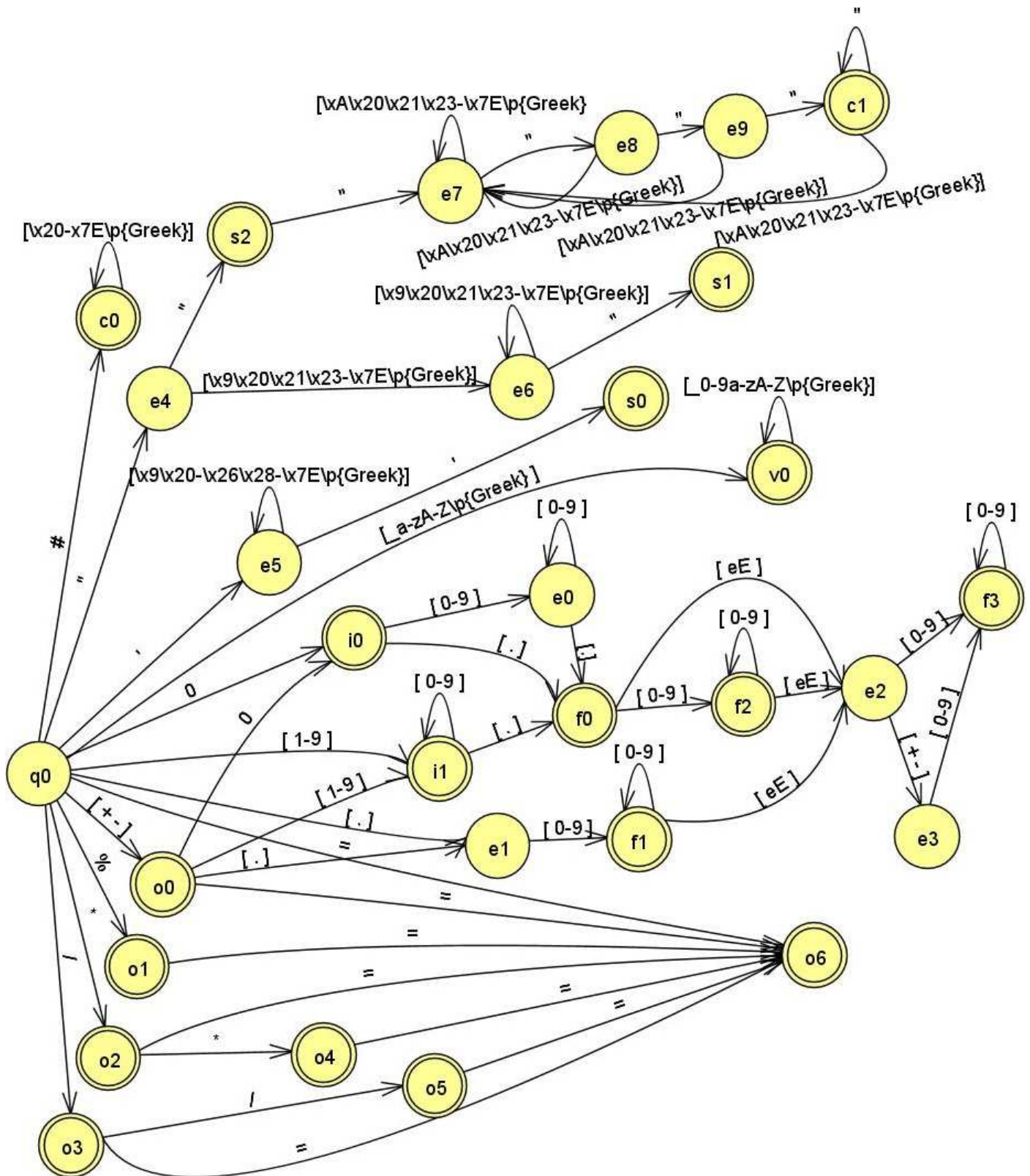
```

Σημειώσεις:

Στην κανονική έκφραση έχουμε χρησιμοποιήσει πάλι τον κώδικα ASCII για καθορισμό των επιτρεπόμενων συμβόλων. Στην περίπτωση των σχολίων που αρχίζουν με # δεν συμπεριλαμβάνουμε τον χαρακτήρα νέας γραμμής xA, ενώ στη άλλη περίπτωση δεν συμπεριλαμβάνεται ο χαρακτήρας " (x22). Για να τερματίσουμε σωστά το fsm στην περίπτωση των τριπλών εισαγωγικών "", πρέπει **πρώτα** να πατήσουμε EOF, δηλαδή ^Z **και μετά** να πατήσουμε χαρακτήρα νέας γραμμής (\n).

1.6 - Ολοκληρωμένο ΠΠΑ αναγνώρισης της γλώσσας

Διάγραμμα Μετάβασης (ΔΜ):



Πίνακας Μετάβασης (ΠΜ):

	#	"	'	[_a-zA-Z]	0	[1-9]	[.]	=	[+ -]	%	*	/	[e E]	\s	\n	exit
Q0	C0	E4	E5	V0	I0	I1	E1	O6	O0	O1	O2	O3				
O0					I0	I1	E1	O6								OK
O1								O6								OK
O2								O6								OK
O3								O6				O5				OK
O4								O6								OK
O5								O6								OK
O6																OK
I0					E0	E0	F0									OK
I1					I1	I1	F0									OK
F0					F2	F2							E2			OK
F1					F1	F1							E2			OK
F2					F2	F2							E2			OK
F3					F3	F3										OK
V0				V0	V0	V0										OK
S0																OK
S1																OK
S2		E7														OK
C0				C0	C0	C0								C0		OK
C1		C1		E7	E7	E7								E7	E7	OK
E0					E0	E0	F0									
E1					F1	F1										
E2					F3	F3			E3							
E3					F3	F3										
E4		S2		E6	E6	E6								E6		
E5			S0	E5	E5	E5								E5		
E6		S1		E6	E6	E6								E6		
E7		E8		E7	E7	E7								E7	E7	
E8		E9		E7	E7	E7								E7	E7	
E9		C1		E7	E7	E7								E7	E7	

Σημειώσεις:

Η αρχική κατάσταση είναι η Q0. Στις επόμενες καταστάσεις τα ονόματα δηλώνουν τον τύπο του αλφαριθμητικού, εάν υπάρχει δυνατότητα εξόδου, αλλιώς έχουν το όνομα e (error) που δηλώνει ότι δεν μπορεί να ολοκληρώσει εκεί.

πχ. το V0 σημαίνει ότι ολοκληρώνει με τύπο variable, το F2 με τύπο float κλπ.

Όπως βλέπουμε και στον ΠΜ οι έξοδοι επιτρέπονται σε όλες τις καταστάσεις πλην των e (error).

Αρχείου εισόδου fsm:

Q0: # -> C0	C0: 0-9 a-z A-Z \s -> C0
" -> E4	\n -> GOOD
' -> E5	C1: 0-9 a-z A-Z \s \n -> E7
_ a-z A-Z -> V0	" -> C1
0 -> I0	EOF -> GOOD
1-9 -> I1	E0: 0-9 -> E0
. -> E1	. -> F0
= -> O6	E1: 0-9 -> F1
+ \- -> O0	E2: 0-9 -> F3
% -> O1	+ \- -> E3
* -> O2	E3: 0-9 -> F3
/ -> O3	E4: " -> S2
O0: 0 -> I0	0-9 a-z A-Z \s -> E6
1-9 -> I1	E5: 0-9 a-z A-Z \s -> E5
. -> E1	' -> S0
= -> O6	E6: 0-9 a-z A-Z \s -> E6
\n -> GOOD	" -> S1
O1: = -> O6	E7: 0-9 a-z A-Z \s \n -> E7
\n -> GOOD	" -> E8
O2: = -> O6	E8: 0-9 a-z A-Z \s \n -> E7
* -> O4	" -> E9
\n -> GOOD	E9: 0-9 a-z A-Z \s \n -> E7
O3: / -> O5	" -> C1
= -> O6	GOOD (OK) :
\n -> GOOD	
O4: = -> O6	
\n -> GOOD	
O5: = -> O6	
\n -> GOOD	
O6: \n -> GOOD	
I0: 0-9 -> E0	
. -> F0	
\n -> GOOD	
I1: 0-9 -> I1	
. -> F0	
\n -> GOOD	
F0: e E -> E2	
0-9 -> F2	
\n -> GOOD	
F1: 0-9 -> F1	
e E -> E2	
\n -> GOOD	
F2: 0-9 -> F2	
e E -> E2	
\n -> GOOD	
F3: 0-9 -> F3	
\n -> GOOD	
V0: _ 0-9 a-z A-Z -> V0	
\n -> GOOD	
S0: \n -> GOOD	
S1: \n -> GOOD	
S2: " -> E7	
\n -> GOOD	

```

C:\Users\win7um\fsm>fsm ppa.fsm
34786
^Z
YES

C:\Users\win7um\fsm>fsm ppa.fsm
+24.0990
^Z
YES

C:\Users\win7um\fsm>fsm ppa.fsm
-.33e+54
^Z
YES

C:\Users\win7um\fsm>fsm ppa.fsm
00
fsm: in ppa.fsm, state 'e0' input \n not accepted

C:\Users\win7um\fsm>fsm ppa.fsm
-0
^Z
YES

C:\Users\win7um\fsm>fsm ppa.fsm
"Hello there"
^Z
YES

C:\Users\win7um\fsm>fsm ppa.fsm
'Hello to you'
^Z
YES

C:\Users\win7um\fsm>fsm ppa.fsm
#this is a comment
^Z
YES

C:\Users\win7um\fsm>fsm ppa.fsm
""""this is a looooo
ooooooooong comment""""^Z
^Z
YES

C:\Users\win7um\fsm>fsm ppa.fsm
//
^Z
YES

C:\Users\win7um\fsm>fsm ppa.fsm
%
^Z
YES

C:\Users\win7um\fsm>fsm ppa.fsm
+=
^Z
YES

C:\Users\win7um\fsm>fsm ppa.fsm
_var11
^Z
YES

C:\Users\win7um\fsm>fsm ppa.fsm
04_var
fsm: in ppa.fsm, state 'e0' input _ not accepted

C:\Users\win7um\fsm>fsm ppa.fsm
.e44
fsm: in ppa.fsm, state 'e1' input e not accepted

C:\Users\win7um\fsm>fsm ppa.fsm
==
fsm: in ppa.fsm, state 'o6' input = not accepted

```