Κατασκευή Λεκτικού-Συντακτικού Αναλυτή για την γλώσσα mini-Python

Εργαστήριο Μεταγλωττιστών Χειμερινό εξάμηνο 2017-18

Ομάδα Ε1-10

Ίρις-Αικατερίνη Διαμαντή - 040038

Ευάγγελος Χουχούμης - 050129

Ε1-10

ΤΕΙ Αθήνας - Τμήμα Μηχανικών Πληροφορικής - 2017-18

**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 - Tο 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"|digit-"0",{digit});**

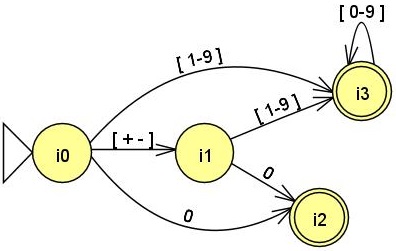
Αποδεκτοί ακέραιοι:0, -0, +0, -23871, +2132, 288200

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

Πεπερασμένο Προσδιοριστικό Αυτόματο (ΠΠΑ) αναγνώρισης ακέραιων:

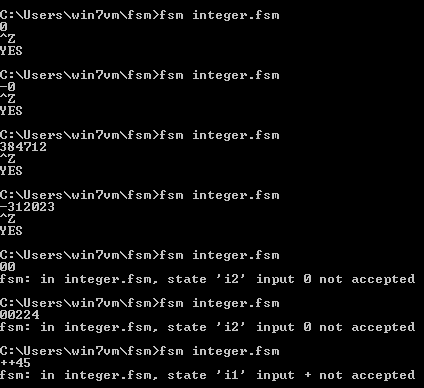
*Διάγραμμα Μετάβασης (ΔΜ): Πίνακας Μετάβασης (ΠΜ):*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 | [+-] | [1-9] | exit |
| Ι0 | Ι2 | Ι1 | Ι3 |  |
| Ι1 | Ι2 |  | Ι3 |  |
| Ι2 |  |  |  | OK |
| I3 | I3 |  | I3 | ΟΚ |



*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):

*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 (\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:*  **FLOAT =[+-]?([0-9]+[.][0-9]\*|[.][0-9]+)([eE][+-]?[0-9]+)?**

*Εναλλακτική σημειογραφία eBNF:*

**FLOAT =**

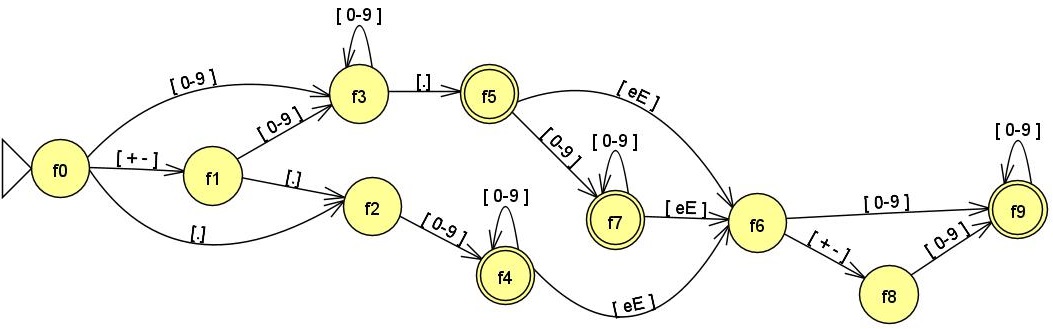
**["+"|"-"](digit,{digit},".",{digit}|".",digit,{digit})[("e"|"E"),("+"|"-"),digit,{digit})];**

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

Μη αποδεκτόι 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):

*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:

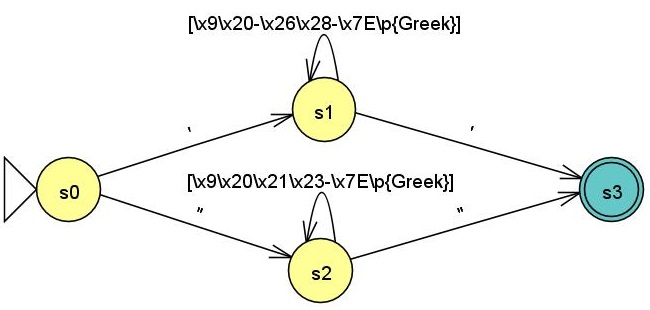
"Κώστας ", "Hello there!", 'Goodbye "Argentina" !', "Your number is '34375'."

Μη αποδεκτές συμβολοσειρές:

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

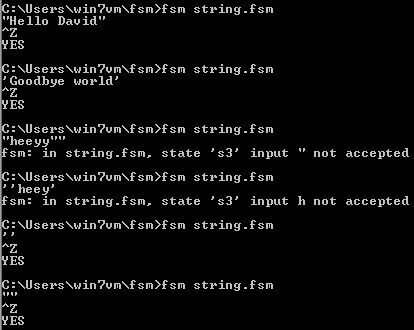
*ΠΠΑ αναγνώρισης συμβολοσειρών:*

*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):

*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.

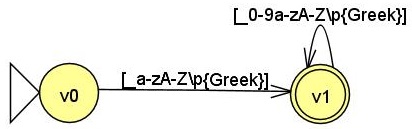
*Κανονική έκφραση:* **VAR =[\_a-zA-Z\p{Greek}][\_0-9a-zA-Z\p{Greek}]\***

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

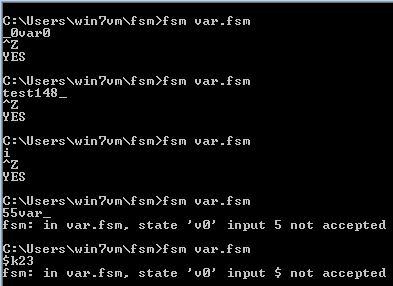
Αποδεκτά ονόματα μεταβλητών: X, αβ, sdfg, A23fv, \_var1, \_ονομα55, \_3847\_

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):

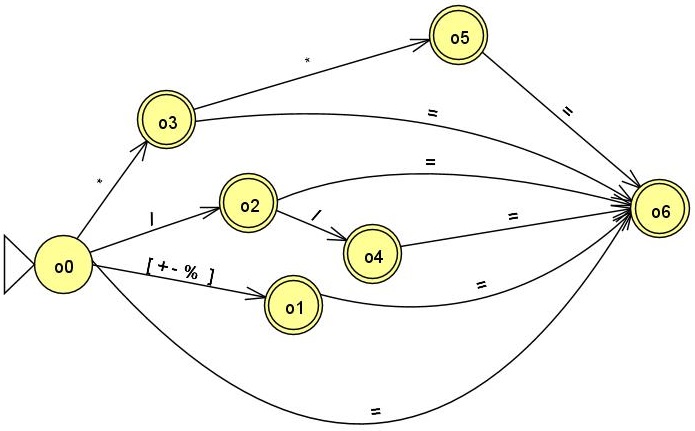
**1.4 - Τελεστές**

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

*Σημειογραφία eBNF:* **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:*

O0: \\* -> O3

/ -> O2

+ \- % -> O1

= -> O6

O1: = -> O6

\n -> GOOD

O2: / -> O4

\n -> GOOD

O3: \\* -> O5

= -> O6

\n -> GOOD

O4: = -> O6

\n -> GOOD

O5: = -> O6

\n -> GOOD

O6: \n -> GOOD

GOOD(OK):

**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

oooooong comment"""

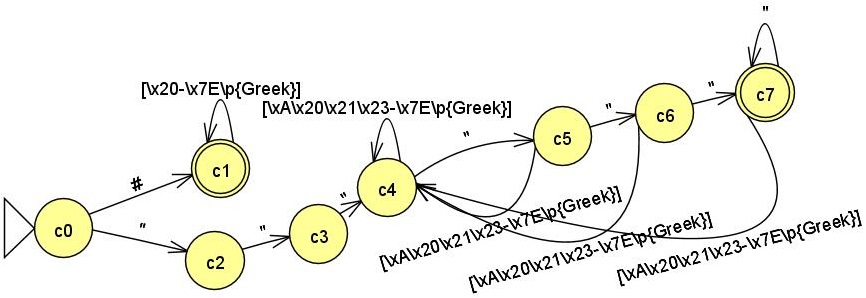
*Μη αποδεκτά σχόλια:* # This is a looooo

ooooong comment

"" This is a comment"""

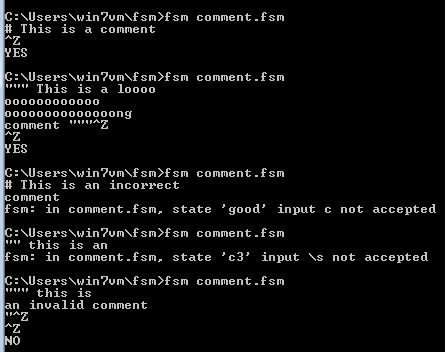
*ΠΠΑ αναγνώρισης σχολίων:*

*Διάγραμμα Μετάβασης (ΔΜ):*

**

*Πίνακας Μετάβασης (ΠΜ):*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | # | " | [\x20-\x7E\p{Greek}] | [\xA\x20\x21\x23-\x7E\p{Greek}] | 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

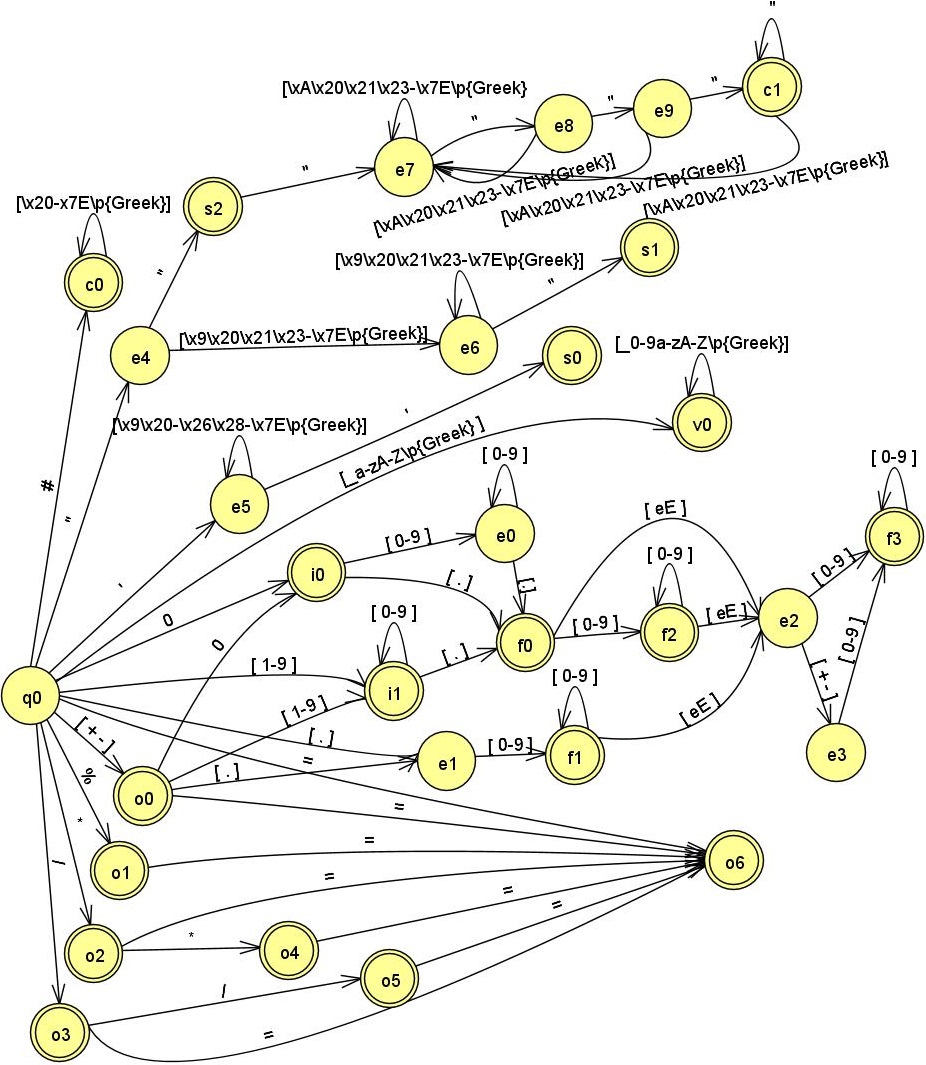
GOOD(OK):

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

