



**Πανεπιστήμιο Δυτικής Αττικής  
Σχολή Μηχανικών  
Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών**

**Μεταγλωττιστές**

**1<sup>ο</sup> Μέρος Εργασίας (Α2)**

**Τμήμα Α3  
Ομάδα (4)**

**ΘΩΜΑΣ ΝΙΚΟΛΑΟΣ (ΠΑΔΑ-21390068-6<sup>ο</sup> εξάμηνο)  
ΠΛΑΓΟΥ ΑΙΚΑΤΕΡΙΝΗ (ΠΑΔΑ-20390191-8<sup>ο</sup> εξάμηνο)  
ΜΠΗΛΙΩΝΗ ΠΑΡΑΣΚΕΥΗ (ΠΑΔΑ-20390286-8<sup>ο</sup> εξάμηνο)  
ΠΟΥΛΙΑΝΑΣ ΤΗΛΕΜΑΧΟΣ (ΠΑΔΑ-21390304-6<sup>ο</sup> εξάμηνο)  
ΜΟΥΤΖΟΥΡΗΣ ΑΘΑΝΑΣΙΟΣ (ΠΑΔΑ-21390137-6<sup>ο</sup> εξάμηνο)**

**ΑΘΗΝΑ**

# Περιεχόμενα

ΜΕΡΟΣ Α-2: Κωδικοποίηση αυτομάτων πεπερασμένων καταστάσεων μέσω FSM .....	3
Παρουσίαση στόχου εργασίας .....	3
1. Λεκτικές μονάδες (tokens) – Επιμέρους.....	4
1.1 Αναγνωριστικά (ονόματα) / identifiers (names) .....	4
Ανάλυση της κανονικής έκφρασης .....	4
Σχολιασμός παραδειγμάτων.....	5
1.2 Λεκτικά κυριολεκτικά (strings).....	6
Ανάλυση της κανονικής έκφρασης .....	7
Σχολιασμός παραδειγμάτων.....	7
1.3 Αριθμητικά κυριολεκτικά (numbers) .....	9
Ανάλυση της κανονικής έκφρασης .....	10
Σχολιασμός παραδειγμάτων.....	10
1.4 Ακέραιοι (integers).....	12
Ανάλυση της κανονικής έκφρασης .....	13
Σχολιασμός παραδειγμάτων.....	13
1.5 Αριθμοί κινούμενης υποδιαστολής (float) .....	15
Ανάλυση της κανονικής έκφρασης .....	16
Σχολιασμός παραδειγμάτων.....	16
2. Λεκτικές μονάδες (tokens) – Ενιαίο.....	18
Σχολιασμός παραδειγμάτων.....	21
Αναφορές / Δυσκολίες:.....	23
Ανάθεση αρμοδιοτήτων – ρόλων:.....	25

## **ΜΕΡΟΣ Α-2: Κωδικοποίηση αυτομάτων πεπερασμένων καταστάσεων μέσω FSM**

### **Παρουσίαση στόχου εργασίας**

Οι πεπερασμένες καταστάσεις των λεκτικών μονάδων, περιλαμβάνουν τα αναγνωριστικά, τα λεκτικά κυριολεκτικά, τα αριθμητικά κυριολεκτικά (τους ακεραίους και τους αριθμούς κινούμενης υποδιαστολής). Για την κωδικοποίηση αυτών είναι απαραίτητος ο σχεδιασμός του αυτόματου αναγνώρισης, το οποίο περιλαμβάνει τους κανόνες για την αναγνώριση των λεκτικών μονάδων. Επιπλέον, παρουσιάζεται ο Πίνακας Μεταβάσεων (ΠΜ) τόσο των επιμέρους καταστάσεων όσο και του ενιαίου αυτόματου που προκύπτει. Για την δημιουργία των κανονικών εκφράσεων χρησιμοποιείται το εργαλείο [REGEXPAL](#), το οποίο είναι μια αναγνωριστική μηχανή που ελέγχει την ορθότητα αυτών. Για τις πεπερασμένες καταστάσεις χρησιμοποιείται το εργαλείο FSM. Η διαδικασία της ορθότητας των πεπερασμένων καταστάσεων περιλαμβάνει την αντιστοίχιση όλων των επιτρεπτών εισόδων σε μια τελική κατάσταση 'GOOD' και όλων των μη επιτρεπτών εισόδων σε μία τελική κατάσταση 'BAD'.

Σκοπός της παρούσας εργασίας είναι η σχεδίαση, η υλοποίηση και η τεκμηρίωση ενός αυτόματου αναγνώρισης λεκτικών μονάδων. Πιο συγκεκριμένα, επιτυγχάνεται η δημιουργία ενός αυτόματου που αναγνωρίζει διάφορες λεκτικές μονάδες με βάση τους κανόνες που ορίζονται στη γραμματική της γλώσσας UNI-C. Με βάση αυτών των αυτομάτων, παρουσιάζονται οι Πίνακες Μεταβάσεων (ΠΜ) και ο τρόπος λειτουργίας τους. Τέλος, γίνεται έλεγχος των αποτελεσμάτων του αυτόματου καθώς και αναλυτικός σχολιασμός αυτών των αποτελεσμάτων σχετικά με την ορθότητά τους.

# 1. Λεκτικές μονάδες (tokens) – Επιμέρους

## 1.1 Αναγνωριστικά (ονόματα) / identifiers (names)

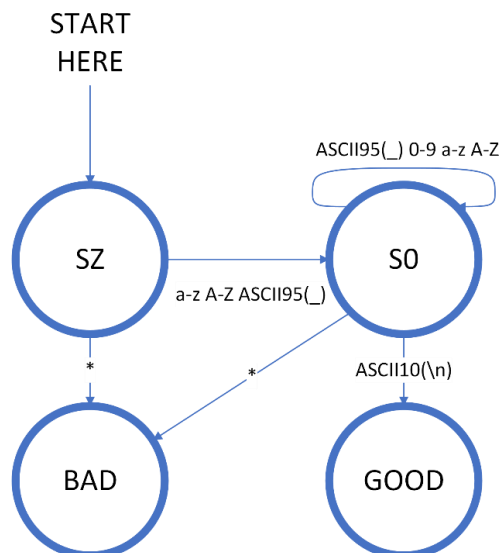
### Κανονική Έκφραση:

- $[a-zA-Z]+((\_+)(\d+)([a-zA-Z]+)^*)|\_+((\_+)(\d+)([a-zA-Z]+)^*)^*$

START=SZ		
SZ:	a-z A-Z _	-> S0
	*	-> BAD
S0:	_ 0-9 a-z A-Z	-> S0
	\n	-> GOOD
	*	-> BAD
GOOD(OK):		

Εικόνα 1: Κώδικας FSM και σχέδιο καταστάσεων για τα αναγνωριστικά.

Name	-> GOOD
Name123	-> GOOD
Name_123	-> GOOD
_name123	-> GOOD
Name_1_END	-> GOOD
_name	-> GOOD
_123	-> GOOD
___	-> GOOD
name@email	-> BAD
name_123!23	-> BAD
name.123	-> BAD
123	-> BAD
Name 123	-> BAD
name\123	-> BAD
123_name	-> BAD



State	a-z A-Z	ASCII(_)	[0-9]	*	ASCII10(/n)
SZ	S0	S0	∅	BAD	∅
S0	S0	S0	S0	BAD	GOOD

Πίνακας 1: Παραδείγματα χρήσης και πίνακας μεταβάσεων αναγνωριστικών.

## Ανάλυση της κανονικής έκφρασης

$[a-zA-Z]+((\_+)(\d+)([a-zA-Z]+)^*)|\_+((\_+)(\d+)([a-zA-Z]+)^*)^*$

$[a-zA-Z]^+$  : οποιοδήποτε γράμμα κεφαλαίο ή μικρό μια ή περισσότερες φορές

$(\_+)^*$  : χαρακτήρας κάτω παύλα \_ μια ή περισσότερες φορές, που μπορεί να επαναλαμβάνεται καμία ή περισσότερες φορές

$(\d+)^*$  : κανέναν ή πολλοί ακέραιοι αριθμοί με ένα ή περισσότερα ψηφία

$([a-zA-Z]+)^*$  : οποιοδήποτε γράμμα κεφαλαίο ή πεζό μια ή περισσότερες φορές, που μπορεί να επαναλαμβάνεται καμία ή περισσότερες φορές

$((\_+)(\d+)([a-zA-Z]+)^*)^*$  : όλα μέσα στην παρένθεση μπορούν να επαναλαμβάνονται καμία ή περισσότερες φορές

| : λογικός τελεστής ή

Δεν αναγράφεται ανάλυση σε επαναλαμβανόμενες εκφράσεις.

## Σχολιασμός παραδειγμάτων

Δίπλα από τα παραδείγματα εμφανίζεται το αποτέλεσμα που επεστράφη σε δοκιμές με την εκτέλεση του fsm κώδικα, όπως και η επεξήγηση του αποτελέσματος.

- **Name** → **GOOD** [YES]  
Αποδεκτό επειδή αρχίζει με γράμμα και περιέχει μόνο γράμματα.
- **Name123** → **GOOD** [YES]  
Αποδεκτό επειδή αρχίζει με γράμμα και περιέχει γράμματα και αριθμούς.
- **Name\_123** → **GOOD** [YES]  
Αποδεκτό επειδή αρχίζει με γράμμα και περιέχει γράμματα, αριθμούς και κάτω παύλα.
- **\_name123** → **GOOD** [YES]  
Αποδεκτό επειδή αρχίζει με κάτω παύλα και περιέχει γράμματα, αριθμούς και κάτω παύλα.
- **Name\_1\_END** → **GOOD** [YES]  
Αποδεκτό επειδή αρχίζει με γράμμα και περιέχει γράμματα, αριθμούς και κάτω παύλες.
- **\_name** → **GOOD** [YES]  
Αποδεκτό επειδή αρχίζει με κάτω παύλα και περιέχει γράμματα και κάτω παύλα.
- **\_123** → **GOOD** [YES]  
Αποδεκτό επειδή αρχίζει με κάτω παύλα και περιέχει αριθμούς και κάτω παύλα.
- **\_\_\_\_\_** → **GOOD** [YES]  
Αποδεκτό επειδή αρχίζει με κάτω παύλα και περιέχει κάτω παύλες.
- **name@email** → **BAD** [fsm: in identifiers.fsm, state 'bad' input e not accepted]  
Μη αποδεκτό επειδή περιέχει @ (μη αποδεκτός χαρακτήρας για τα αναγνωριστικά).
- **name\_123!23** → **BAD** [fsm: in identifiers.fsm, state 'bad' input 2 not accepted]  
Μη αποδεκτό επειδή περιέχει ! (μη αποδεκτός χαρακτήρας για τα αναγνωριστικά).
- **name.123** → **BAD** [fsm: in identifiers.fsm, state 'bad' input 1 not accepted]  
Μη αποδεκτό επειδή περιέχει . (μη αποδεκτός χαρακτήρας για τα αναγνωριστικά).
- **123** → **BAD** [fsm: in identifiers.fsm, state 'bad' input 2 not accepted]  
Μη αποδεκτό επειδή αρχίζει με αριθμό.
- **Name 123** → **BAD** [fsm: in identifiers.fsm, state 'bad' input 1 not accepted]  
Μη αποδεκτό επειδή περιέχει τον κενό χαρακτήρα (space) (μη αποδεκτός για τα αναγνωριστικά).
- **name\123** → **BAD** [fsm: in identifiers.fsm, state 'bad' input 1 not accepted]  
Μη αποδεκτό επειδή περιέχει \ (μη αποδεκτός χαρακτήρας για τα αναγνωριστικά).
- **123\_name** → **BAD** [fsm: in identifiers.fsm, state 'bad' input 2 not accepted]  
Μη αποδεκτό επειδή αρχίζει με αριθμό.

## 1.2 Λεκτικά κυριολεκτικά (strings)

### Κανονική Έκφραση:

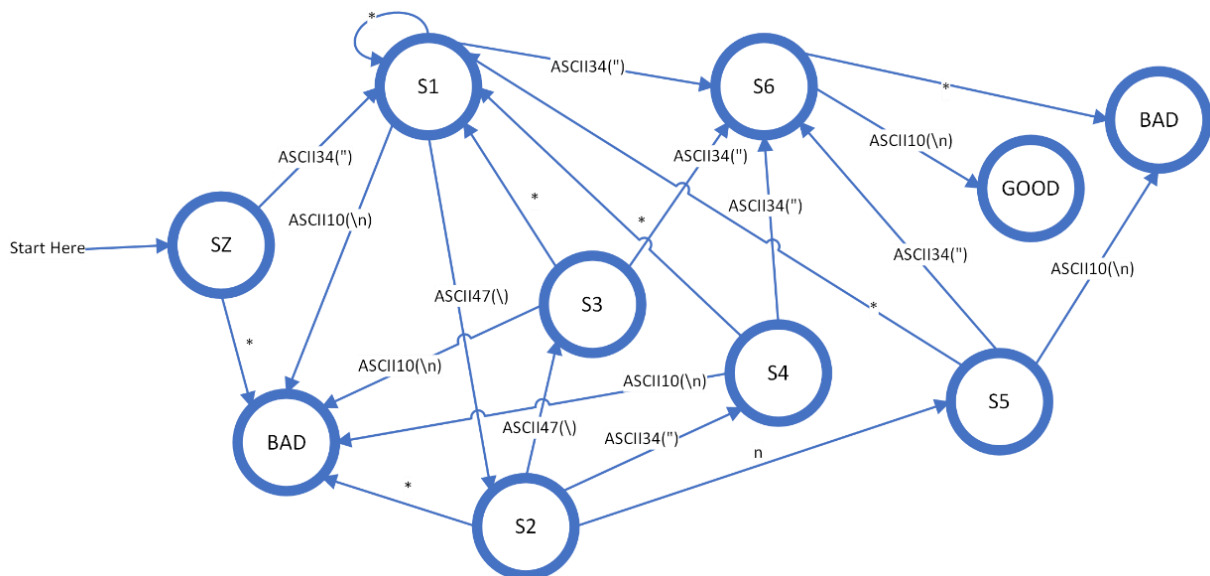
- $^{"}([^\backslash n\backslash ]*(\backslash ")*(\backslash n)*(\backslash \backslash )*)*)\$$

START=SZ		
SZ:	"	-> S1
	*	-> BAD
S1:	\	-> S2
	\n	-> BAD
	"	-> S6
	*	-> S1
S2:	\	-> S3
	"	-> S4
	n	-> S5
	*	-> BAD
S3:	"	-> S6
	\n	-> BAD
	*	-> S1
S4:	"	-> S6
	\n	-> BAD
	*	-> S1
S5:	"	-> S6
	\n	-> BAD
	*	-> S1
S6:	\n	-> GOOD

State	"	\	\n	n	*
SZ	S1	∅	∅	∅	BAD
S1	S6	S2	BAD	∅	S1
S2	S4	S3	∅	S5	BAD
S3	S6	∅	BAD	∅	S1
S4	S6	∅	BAD	∅	S1
S5	S6	∅	BAD	∅	S1
S6	∅	∅	GOOD	∅	BAD

"Test" -> GOOD  
 "Example using \"Quotes!\"" -> GOOD  
 "Newline \n" -> GOOD  
 " " -> GOOD  
 "example with tab and different characters %\$#^&\*"-> GOOD  
 "TESTING escape sequences \\ \"example\" \n" -> GOOD  
 "" -> GOOD  
 "new  
line" -> BAD  
 "This is an example with \" -> BAD  
 "Wrong input -> BAD  
 Example using no quotes -> BAD  
 "Trying two strings in one line""This is another string"-> BAD  
 "something" something "something" -> BAD  
 "Using space character after closing the string" -> BAD  
 Wrong input" -> BAD

Πίνακας 2: Κώδικας FSM, παραδείγματα χρήσης και πίνακας μεταβάσεων λεκτικών κυριολεκτικών.



Εικόνα 2: Σχέδιο καταστάσεων για τα λεκτικά κυριολεκτικά.

## Ανάλυση της κανονικής έκφρασης

`^("[^"\\n\\]*(\\")*(\\n)*(\\\\\\)*")*$`

`^` : αρχή γραμμής - string

`"` : χαρακτήρας διπλή απόστροφος "

`[^"\\n\\]*` : οποιοσδήποτε χαρακτήρας δεν είναι διπλή απόστροφος " , newline `\n` (αλλαγή γραμμής) και backslash `\` καμία ή περισσότερες φορές. Το backslash `\` είναι ιδικός χαρακτήρας συνεπώς χρησιμοποιούμε τη σειρά διαφυγής `\\`.

`(\\")*` : χαρακτήρες `\"` καμία ή περισσότερες φορές, χρησιμοποιώντας τη σειρά διαφυγής `\\`.

`(\\n)*` : χαρακτήρες `\n` καμία ή περισσότερες φορές, χρησιμοποιώντας τη σειρά διαφυγής `\\`.

`(\\\\\\)*` : χαρακτήρες `\\` καμία ή περισσότερες φορές, χρησιμοποιώντας τη σειρά διαφυγής `\\` δύο φορές.

`([^\n\\"]*(\\")*(\\n)*(\\\\\\)*)*` : όλα μέσα στην παρένθεση μπορούν να επαναλαμβάνονται καμία ή περισσότερες φορές

`$` : τέλος γραμμής

Δεν αναγράφεται ανάλυση σε επαναλαμβανόμενες εκφράσεις.

## Σχολιασμός παραδειγμάτων

Δίπλα από τα παραδείγματα εμφανίζεται το αποτέλεσμα που επεστράφη σε δοκιμές με την εκτέλεση του fsm κώδικα, όπως και η επεξήγηση του αποτελέσματος.

- "Test" → GOOD [YES]

Αποδεκτό διότι περικλείονται απο διπλές αποστρόφους `"` χαρακτήρες εκτός του backslash `\`, της νέας γραμμής `\n` και της διπλής αποστρόφου `"`.

- "Example using \"Quotes!\"" → GOOD [YES]

Αποδεκτό διότι περικλείονται απο διπλές αποστρόφους `"` χαρακτήρες εκτός του backslash `\`, της νέας γραμμής `\n` και της διπλής αποστρόφου `"`. Η χρήση του backslash μαζί με τη διπλή απόστροφο `\` είναι αποδεκτή εφόσον αναγνωρίζεται ως σειρά διαφυγής.

- "Newline \n" → GOOD [YES]

Αποδεκτό διότι περικλείονται απο διπλές αποστρόφους `"` χαρακτήρες εκτός του backslash `\`, της νέας γραμμής `\n` και της διπλής αποστρόφου `"`. Η χρήση του `\n` παρόλο που συμβολίζει την αλλαγή γραμμής (newline) είναι αποδεκτή διότι αναγνωρίζεται ως σειρά διαφυγής και δεν εκτελείται ως αλλαγή γραμμής.

- " " → GOOD [YES]

Αποδεκτό διότι περικλείεται απο διπλές αποστρόφους `"` μόνο ο χαρακτήρας του κενού.

- "example with tab and different characters %\$#^&\*" → GOOD [YES]

Αποδεκτό διότι περικλείονται απο διπλές αποστρόφους `"` χαρακτήρες εκτός του backslash `\`, της νέας γραμμής `\n` και της διπλής αποστρόφου `"`.

- "TESTING escape sequences \\ \"example\" \n" → GOOD [YES]

Αποδεκτό διότι περικλείονται απο διπλές αποστρόφους `"` χαρακτήρες εκτός του backslash `\`, της νέας γραμμής `\n` και της διπλής αποστρόφου `"`. Εδώ χρησιμοποιούνται όλοι οι συνδυασμοί σειρών διαφυγής, με τον μόνο που δεν έχει προαναφερθεί να είναι το διπλό backslash `\\`.

- "" → GOOD [YES]

Αποδεκτό διότι είναι συντακτικά ορθή η χρήση των διπλών αποστρόφων " " παρόλο που δεν περικλείουν κάποιο χαρακτήρα.

- "This is an example with \" → BAD [NO]

Μη αποδεκτό γιατί παρόλο που υπάρχουν διπλοί απόστροφους " στην αρχή της φράσης, δεν υπάρχουν στο τέλος. Η χρήση του backslash μαζί με τη διπλή απόστροφο \" αναγνωρίζεται ως σειρά διαφυγής (escape sequence) καθιστώντας τη διπλή απόστροφο μη λειτουργική.

- "Wrong input → BAD [NO]

Μη αποδεκτό γιατί παρόλο που υπάρχει διπλή απόστροφος " στην αρχή της φράσης, δεν υπάρχουν στο τέλος.

- Example using no quotes → BAD [fsm: in string.fsm, state 'bad' input x not accepted]

Μη αποδεκτό διότι δεν περικλείονται σε διπλές αποστρόφους " " οι χαρακτήρες.

- "new  
line" → BAD [fsm: in string.fsm, state 'bad' input l not accepted]

Μη αποδεκτό διότι εκτελείται αλλαγή γραμμής (\n) πριν λήξει η συμβολοσειρά.

- "Trying two strings in one line""This is another string" → BAD

[fsm: in string.fsm, state 'bad' input T not accepted]

Μη αποδεκτό διότι χρησιμοποιούνται πάνω από δύο φορές (μία στην αρχή και μία στο τέλος) οι διπλές αποστρόφοι " ".

- "something" something "something" → BAD

[fsm: in string.fsm, state 'good' input s not accepted]

Μη αποδεκτό διότι υπάρχουν κι άλλοι χαρακτήρες μετά τη λήξη των διπλών αποστρόφων " " που περικλείουν τη συμβολοσειρά.

- "Using space character after closing the string" → BAD

[fsm: in string.fsm, state 'bad' input \n not accepted]

Μη αποδεκτό διότι υπάρχει ο κενός χαρακτήρας μετά τη λήξη των διπλών αποστρόφων " " που περικλείει τη συμβολοσειρά.

- Wrong input" → BAD [fsm: in string.fsm, state 'bad' input r not accepted]

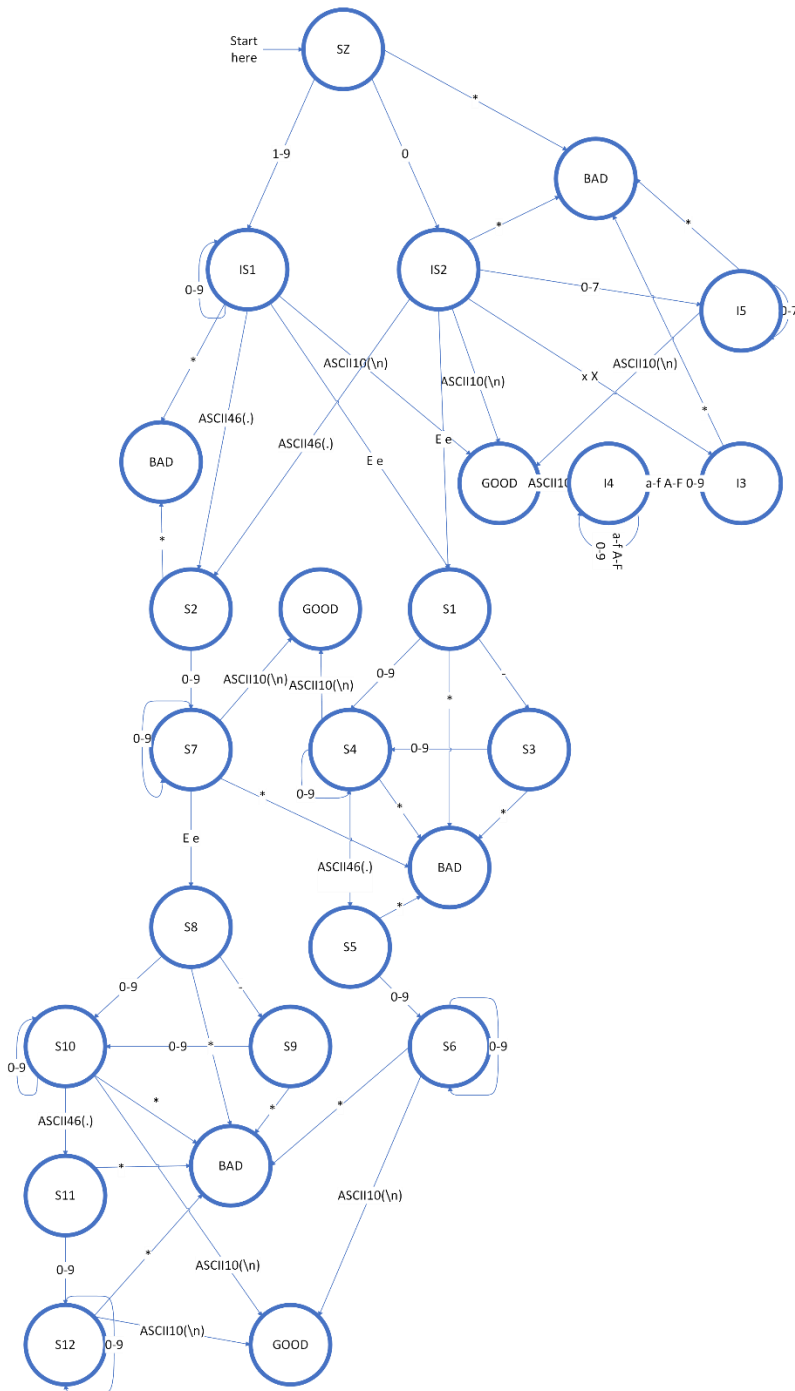
Μη αποδεκτό γιατί παρόλο που υπάρχει διπλή απόστροφος " στο τέλος της φράσης, δεν υπάρχουν στην αρχή.



## 1.3 Αριθμητικά κυριολεκτικά (numbers)

### Κανονική Έκφραση:

- $(\backslash d+[eE]-?\backslash d+(\backslash.\backslash d+)?|\backslash d+\backslash.\backslash d+([eE]-?\backslash d+(\backslash.\backslash d+)?))|([1-9](\backslash d+)?|0[xX][a-fA-F\backslash d]+|0[0-7]*)$



Εικόνα 3: Κώδικας FSM και σχέδιο καταστάσεων για τα αριθμητικά κυριολεκτικά.

START=SZ	
SZ:	1-9 -> IS1
	0 -> IS2
	* -> BAD
IS1:	0-9 -> IS1
	E e -> S1
	. -> S2
	\n -> GOOD
	* -> BAD
IS2: x X	-> I3
	E e -> S1
	. -> S2
	0-7 -> I5
	\n -> GOOD
	* -> BAD
I3: a-f A-F 0-9	-> I4
	* -> BAD
I4: a-f A-F 0-9	-> I4
	* -> BAD
	\n -> GOOD
I5:	0-7 -> I5
	\n -> GOOD
	* -> BAD
S1:	- -> S3
	0-9 -> S4
	* -> BAD
S2:	0-9 -> S7
	* -> BAD
S3:	0-9 -> S4
	* -> BAD
S4:	0-9 -> S4
	. -> S5
	\n -> GOOD
	* -> BAD
S5:	0-9 -> S6
	* -> BAD
S6:	0-9 -> S6
	\n -> GOOD
	* -> BAD
S7:	0-9 -> S7
	E e -> S8
	\n -> GOOD
	* -> BAD
S8:	- -> S9
	0-9 -> S10
	* -> BAD
S9:	0-9 -> S10
	* -> BAD
S10:	0-9 -> S10
	. -> S11
	\n -> GOOD
	* -> BAD
S11:	0-9 -> S12
	* -> BAD
S12:	0-9 -> S12
	\n -> GOOD
	* -> BAD
GOOD(OK):	

0	-> GOOD
0x4F	-> GOOD
0147	-> GOOD
0XFF42	-> GOOD
029123	-> BAD
0xG123	-> BAD
055X3	-> BAD
0x	-> BAD
3.14	-> GOOD
1e100	-> GOOD
14e1.2	-> GOOD
0e-1.2	-> GOOD
e22	-> BAD
-22	-> BAD
1.10.10	-> BAD
0e	-> BAD

State	[1-9]	0	[0-9]	eE	.	xX	[0-7]	a-f A-F	(-)	ASCII10(/n)	*
SZ	IS1	IS2	∅	∅	∅	∅	∅	∅	∅	∅	BAD
IS1	∅	∅	IS1	S1	S2	∅	∅	∅	∅	GOOD	BAD
IS2	∅	∅	∅	S1	S2	I3	I5	∅	∅	GOOD	BAD
I3	∅	∅	I4	∅	∅	∅	∅	I4	∅	∅	BAD
I4	∅	∅	I4	∅	∅	∅	∅	I4	∅	GOOD	BAD
I5	∅	∅	∅	∅	∅	∅	I5	∅	∅	GOOD	BAD
S1	∅	∅	S4	∅	∅	∅	∅	∅	S3	∅	BAD
S2	∅	∅	S7	∅	∅	∅	∅	∅	∅	∅	BAD
S3	∅	∅	S4	∅	∅	∅	∅	∅	∅	∅	BAD
S4	∅	∅	S4	∅	S5	∅	∅	∅	∅	GOOD	BAD
S5	∅	∅	S6	∅	∅	∅	∅	∅	∅	GOOD	BAD
S6	∅	∅	S6	∅	∅	∅	∅	∅	∅	GOOD	BAD
S7	∅	∅	S7	S8	∅	∅	∅	∅	∅	GOOD	BAD
S8	∅	∅	S10	∅	∅	∅	∅	∅	S9	∅	BAD
S9	∅	∅	S10	∅	∅	∅	∅	∅	∅	∅	BAD
S10	∅	∅	S10	∅	S11	∅	∅	∅	∅	GOOD	BAD
S11	∅	∅	S12	∅	∅	∅	∅	∅	∅	∅	BAD
S12	∅	∅	S12	∅	∅	∅	∅	∅	∅	GOOD	BAD

Πίνακας 1: Παραδείγματα χρήσης και πίνακας μεταβάσεων αριθμητικών κυριολεκτικών.

## Ανάλυση της κανονικής έκφρασης

$(\text{id} + [\text{eE}] - ? \text{id} + (\backslash \text{id} +) ? \text{id} + \backslash \text{id} + ([\text{eE}] - ? \text{id} + (\backslash \text{id} +) ?) ?) ([1-9] \backslash \text{id} +) ? | 0 [\text{xX}] [\text{a-fA-F} \backslash \text{id}] + | 0 [0-7]^*$

Η κανονική έκφραση δημιουργείται από τη συνένωση, με τον λογικό τελεστή ή |, των εκφράσεων ακεραίων και αριθμών κινούμενης υποδιαστολής. Η ανάλυση αυτών πραγματοποιείται παρακάτω.

## Σχολιασμός παραδειγμάτων

Δίπλα από τα παραδείγματα εμφανίζεται το αποτέλεσμα που επεστράφη σε δοκιμές με την εκτέλεση του fsm κώδικα, όπως και η επεξήγηση του αποτελέσματος.

- 0 → GOOD [YES]

Αποδεκτό διότι είναι οκταδικός ακέραιος που ξεκινάει με 0 και βρίσκεται μέσα στα όρια ενός οκταδικού αριθμού (0-7).

- 0x4F → GOOD [YES]

Αποδεκτό διότι είναι δεκαεξαδικός ακέραιος που ξεκινάει με 0x ή 0X και βρίσκεται μέσα στα όρια ενός δεκαεξαδικού αριθμού (0-9, A-F).

- 0147 → GOOD [YES]

Αποδεκτό διότι είναι οκταδικός ακέραιος που ξεκινάει με 0 και βρίσκεται μέσα στα όρια ενός οκταδικού αριθμού (0-7).

- 0XFF42 → GOOD [YES]

Αποδεκτό διότι είναι δεκαεξαδικός ακέραιος που ξεκινάει με 0x ή 0X και βρίσκεται μέσα στα όρια ενός δεκαεξαδικού αριθμού (0-9, A-F).

029123 → **BAD** [fsm: in integer.fsm, state 'bad' input 1 not accepted]

Μη αποδεκτό διότι ξεπερνάει τα όρια ενός οκταδικού αριθμού (0-7).

- 0xG123 → **BAD** [fsm: in integer.fsm, state 'bad' input 1 not accepted]

Μη αποδεκτό διότι περιέχει το γράμμα G το οποίο ξεπερνάει τα όρια ενός δεκαεξαδικού αριθμού (0-9, A-F).

- 055X3 → **BAD** [fsm: in integer.fsm, state 'bad' input 3 not accepted]

Μη αποδεκτό διότι ξεπερνάει τα όρια ενός οκταδικού αριθμού (0-7).

- 0x → **BAD** [NO]

Μη αποδεκτό διότι είναι δεκαεξαδικός ακέραιος που ξεκινάει από 0x ή 0X αλλά δεν ακολουθείτε από κάποιον άλλο αριθμό (0-9,A-F).

- 3.14 → **GOOD** [YES]

Αποδεκτό διότι περιέχει αριθμούς κινητής υποδιαστολής.

- 1e100 → **GOOD** [YES]

Αποδεκτό διότι περιέχει ύψωση σε δύναμη.

- 14e1.2 → **GOOD** [YES]

Αποδεκτό διότι περιέχει ύψωση σε δύναμη και αριθμούς κινούμενης υποδιαστολής.

- 0e-1.2 → **GOOD** [YES]

Αποδεκτό διότι περιέχει ύψωση σε δύναμη και αριθμούς κινητής υποδιαστολής.

- e22 → **BAD** [fsm: in float.fsm, state 'bad' input 2 not accepted]

Μη αποδεκτό διότι ξεκινάει με ύψωση σε δύναμη (e).

- -22 → **BAD** [fsm: in float.fsm, state 'bad' input 2 not accepted]

Μη αποδεκτό διότι είναι προσημασμένος ακέραιος.

- 1.10.10 → **BAD** [fsm: in float.fsm, state 'bad' input 1 not accepted]

Μη αποδεκτό διότι περιέχει τρεις τελείες.

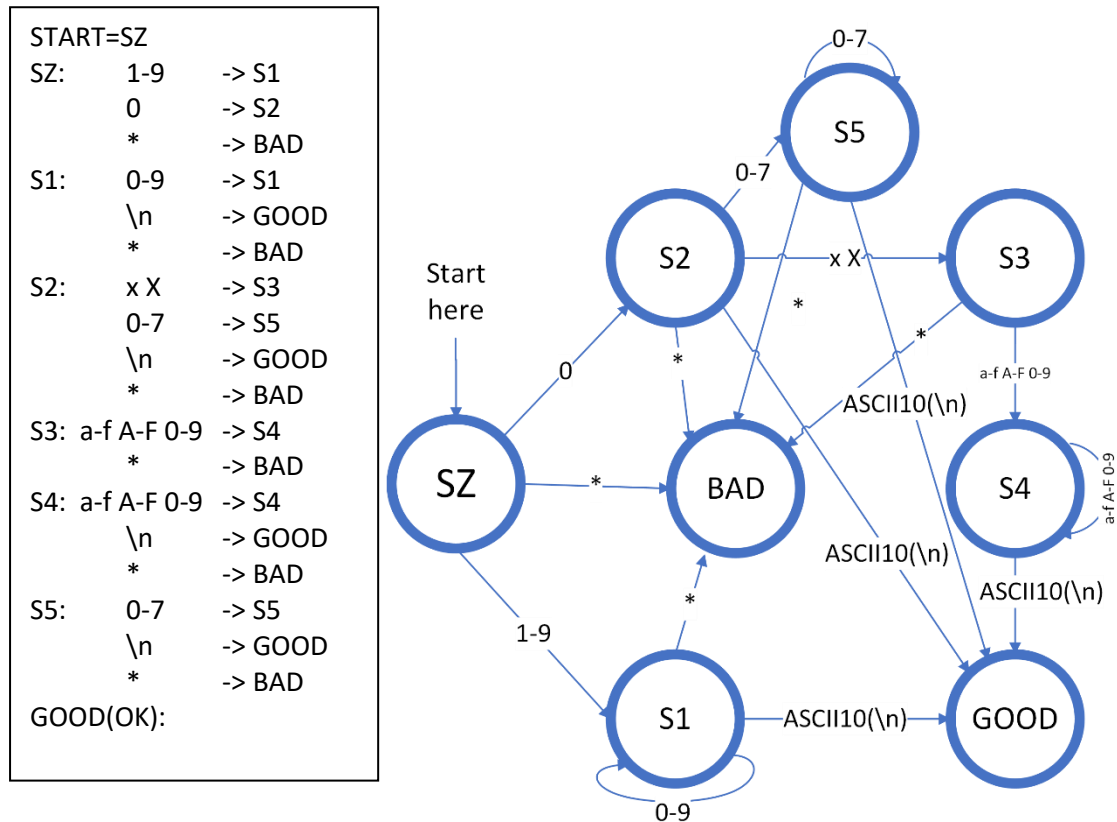
- 0e → **BAD** [NO]

Μη αποδεκτό διότι περιέχει ύψωση σε δύναμη η οποία δεν ακολουθείτε από κάποιον αριθμό.

## 1.4 Ακέραιοι (integers)

### Κανονική Έκφραση:

- $[1-9](\backslash d+)?|0[xX][a-fA-F\backslash d]+|0[0-7]^*$



Εικόνα 4: Κώδικας FSM και σχέδιο καταστάσεων για τους ακέραιους αριθμούς.

0	-> GOOD
3	-> GOOD
214748	-> GOOD
0x4F	-> GOOD
0xFF42	-> GOOD
063	-> GOOD
0147	-> GOOD
00	-> GOOD
029123	-> BAD
0xG123	-> BAD
1.2	-> BAD
-3	-> BAD
0XX15	-> BAD
055X3	-> BAD
0x	-> BAD

State	0	[1-9]	[0-9]	[0-7]	x X	a-f A-F 0-9	ASCII10(\n)	*
SZ	S2	S1	∅	∅	∅	∅	∅	BAD
S1	∅	∅	S1	∅	∅	∅	GOOD	BAD
S2	∅	∅	∅	S5	S3	∅	GOOD	BAD
S3	∅	∅	∅	∅	∅	S4	∅	BAD
S4	∅	∅	∅	∅	∅	S4	GOOD	BAD
S5	∅	∅	∅	S5	∅	∅	GOOD	BAD

Πίνακας 2: Παραδείγματα χρήσης και πίνακας μεταβάσεων ακεραίων.

## Ανάλυση της κανονικής έκφρασης

**[1-9](\d+)?|0[xX][a-fA-F\d]+|0[0-7]\***

**[1-9]** : ένα αριθμητικό ψηφίο από το 1 έως το 9

**(\d+)?** : κανένας ή ένας ακέραιος αριθμός με ένα ή περισσότερα ψηφία

**|** : λογικός τελεστής ή

**0** : αριθμός 0

**[xX]** : γράμματα x ή X

**[a-fA-F\d]+** : γράμμα a, b, c, d ή f κεφαλαίο ή πεζό μια ή περισσότερες φορές ή ακέραιος αριθμός με ένα ή περισσότερα ψηφία

**[0-7]\*** : κανένας ή πολλοί ακέραιοι αριθμοί με ψηφία από το 0 έως το 7

Δεν αναγράφεται ανάλυση σε επαναλαμβανόμενες εκφράσεις.

## Σχολιασμός παραδειγμάτων.

Δίπλα από τα παραδείγματα εμφανίζεται το αποτέλεσμα που επεστράφη σε δοκιμές με την εκτέλεση του fsm κώδικα, όπως και η επεξήγηση του αποτελέσματος.

- **0** → **GOOD** [YES]  
Αποδεκτό διότι είναι οκταδικός ακέραιος που ξεκινάει με 0 και βρίσκεται μέσα στα όρια ενός οκταδικού αριθμού (0-7).
- **3** → **GOOD** [YES]  
Αποδεκτό διότι είναι δεκαδικός ακέραιος που ξεκινάει με μη μηδενικό ψηφίο (1-9).
- **214748** → **GOOD** [YES]  
Αποδεκτό διότι είναι δεκαδικός ακέραιος που ξεκινάει με μη μηδενικό ψηφίο (1-9).
- **0x4F** → **GOOD** [YES]  
Αποδεκτό διότι είναι δεκαεξαδικός ακέραιος που ξεκινάει με 0x ή 0X και βρίσκεται μέσα στα όρια ενός δεκαεξαδικού αριθμού (0-9, A-F).
- **0XFF42** → **GOOD** [YES]  
Αποδεκτό διότι είναι δεκαεξαδικός ακέραιος που ξεκινάει με 0x ή 0X και βρίσκεται μέσα στα όρια ενός δεκαεξαδικού αριθμού (0-9, A-F).
- **063** → **GOOD** [YES]  
Αποδεκτό διότι είναι οκταδικός ακέραιος που ξεκινάει με 0 και βρίσκεται μέσα στα όρια ενός οκταδικού αριθμού (0-7).
- **0147** → **GOOD** [YES]  
Αποδεκτό διότι είναι οκταδικός ακέραιος που ξεκινάει με 0 και βρίσκεται μέσα στα όρια ενός οκταδικού αριθμού (0-7).
- **00** → **GOOD** [YES]  
Αποδεκτό διότι είναι οκταδικός ακέραιος που ξεκινάει με 0 και βρίσκεται μέσα στα όρια ενός οκταδικού αριθμού (0-7).
- **029123** → **BAD** [fsm: in integer.fsm, state 'bad' input 1 not accepted]  
Μη αποδεκτό διότι είναι οκταδικός ακέραιος και δεν βρίσκεται μέσα στα όρια ενός οκταδικού

αριθμού (0-7).

- **0xG123** → **BAD** [fsm: in integer.fsm, state 'bad' input 1 not accepted]

Μη αποδεκτό διότι είναι δεκαεξαδικός ακέραιος και δεν βρίσκεται μέσα στα όρια ενός δεκαεξαδικού αριθμού (0-9, A-F).

- **1.2** → **BAD** [fsm: in integer.fsm, state 'bad' input 2 not accepted]

Μη αποδεκτό διότι είναι αριθμός κινούμενης υποδιαστολής.

- **-3** → **BAD** [fsm: in integer.fsm, state 'bad' input 3 not accepted]

Μη αποδεκτό διότι είναι προσημασμένος ακέραιος.

- **0XX15** → **BAD** [fsm: in integer.fsm, state 'bad' input 1 not accepted]

Μη αποδεκτό διότι είναι δεκαεξαδικός ακέραιος και δεν βρίσκεται μέσα στα όρια ενός δεκαεξαδικού αριθμού (0-9, A-F).

- **055X3** → **BAD** [fsm: in integer.fsm, state 'bad' input 3 not accepted]

Μη αποδεκτό διότι είναι δεκαεξαδικός ακέραιος και δεν βρίσκεται μέσα στα όρια ενός δεκαεξαδικού αριθμού (0-9, A-F).

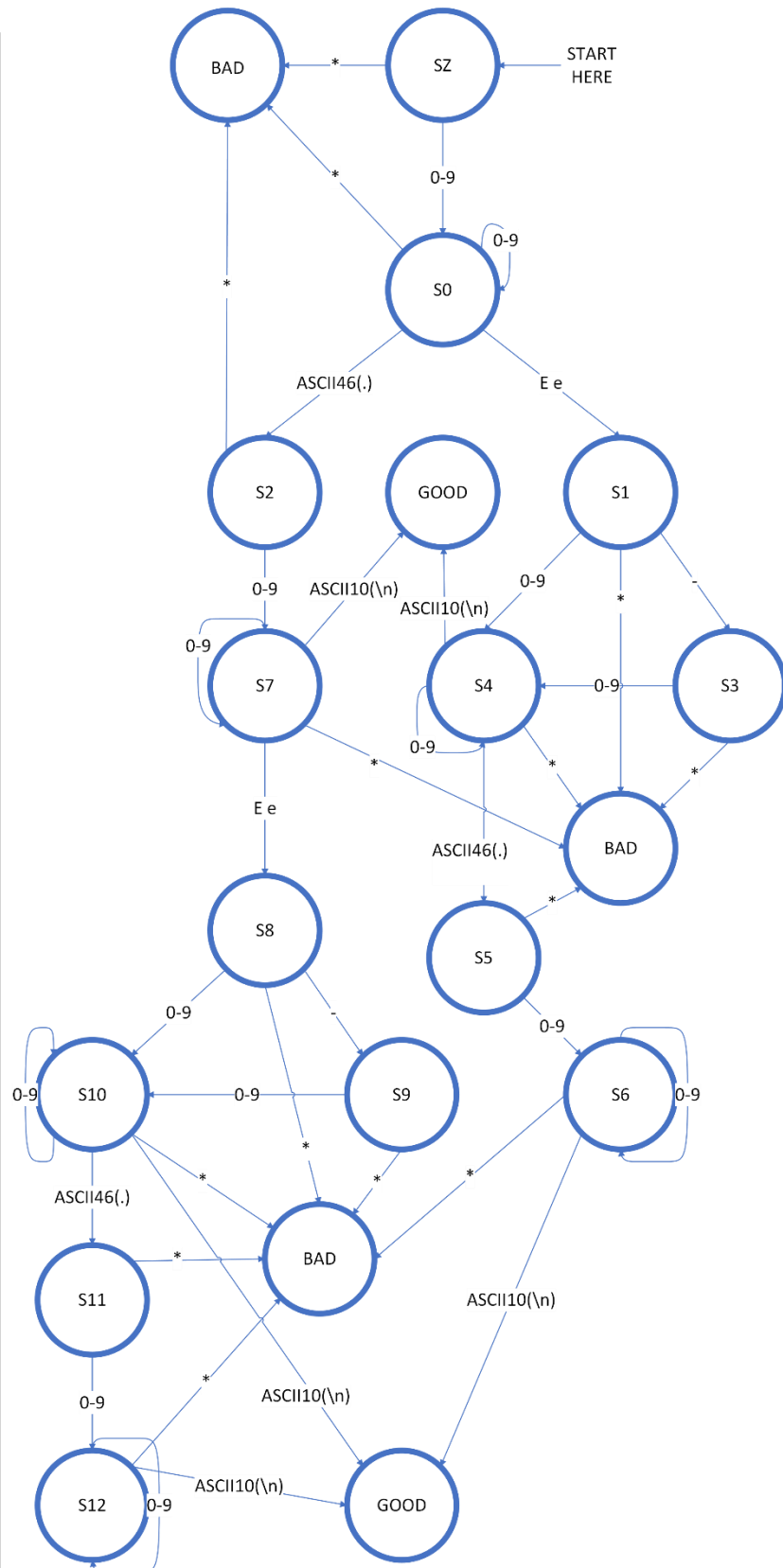
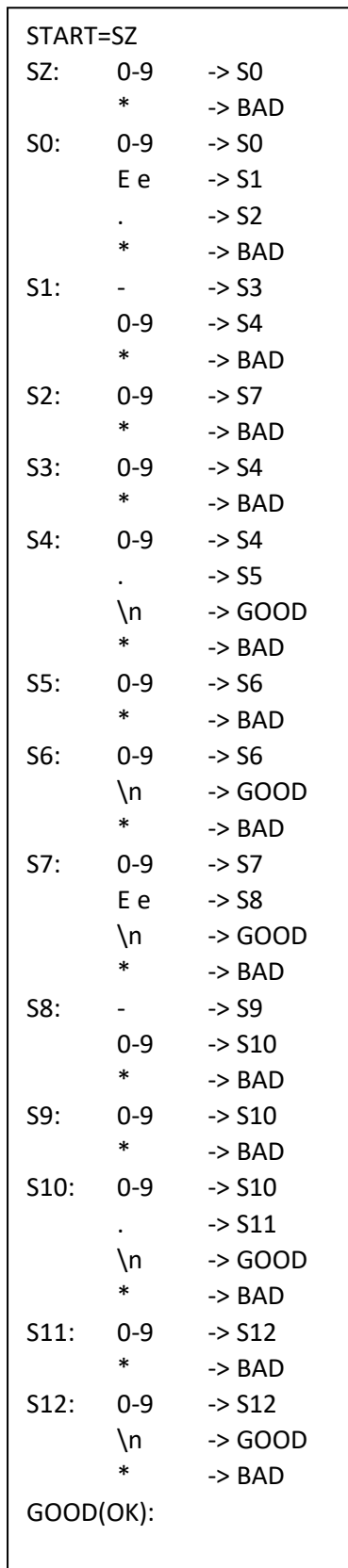
- **0x** → **BAD** [NO]

Μη αποδεκτό διότι είναι δεκαεξαδικός ακέραιος που ξεκινάει από 0x ή 0X αλλά δεν ακολουθείτε από κάποιον άλλο αριθμό (0-9,A-F).

## 1.5 Αριθμοί κινούμενης υποδιαστολής (float)

### Κανονική Έκφραση:

- `\d+[eE]-?\d+(\.\d+)?|\d+\.\d+([eE]-?\d+(\.\d+)?)?`



Εικόνα 5: Κώδικας FSM και σχέδιο καταστάσεων για τους αριθμούς κινούμενης υποδιαστολής.

3.14	-> GOOD
10.0	-> GOOD
0.001	-> GOOD
1e100	-> GOOD
14e1.2	-> GOOD
0e-1.2	-> GOOD
0e0	-> GOOD
3.1e1	-> GOOD
.10	-> BAD
e22	-> BAD
-22	-> BAD
3-	-> BAD
1.10.10	-> BAD
0e	-> BAD
0	-> BAD

State	[1-9]	0	[0-9]	eE	.	(-)	ASCII10(/n)	*
SZ	∅	S0	∅	∅	∅	∅	∅	BAD
S0	S1	S2	S0	S1	S2	∅	∅	BAD
S1	∅	∅	S4	∅	∅	S3	∅	BAD
S2	∅	∅	S7	∅	∅	∅	∅	BAD
S3	∅	∅	S4	∅	∅	∅	∅	BAD
S4	∅	∅	S4	∅	S5	∅	GOOD	BAD
S5	∅	∅	S6	∅	∅	∅	GOOD	BAD
S6	∅	∅	S6	∅	∅	∅	GOOD	BAD
S7	∅	∅	S7	S8	∅	∅	GOOD	BAD
S8	∅	∅	S10	∅	∅	S9	∅	BAD
S9	∅	∅	S10	∅	∅	∅	∅	BAD
S10	∅	∅	S10	∅	S11	∅	GOOD	BAD
S11	∅	∅	S12	∅	∅	∅	∅	BAD
S12	∅	∅	S12	∅	∅	∅	GOOD	BAD

Πίνακας 5: Παραδείγματα χρήσης και πίνακας μεταβάσεων αριθμών κινούμενης υποδιαστολής.

## Ανάλυση της κανονικής έκφρασης

$\backslash d+[eE]-?\backslash d+(\backslash.\backslash d+)?|\backslash d+\backslash.\backslash d+([eE]-?\backslash d+(\backslash.\backslash d+)?)?$

$\backslash d+$  : ακέραιος αριθμός με ένα ή περισσότερα ψηφία

$[eE]$  : γράμματα e ή E

$-?$  : χαρακτήρας παύλα (μείον) -, καμία ή μία φορά

$(\backslash.\backslash d+)?$  : κανένας ή ένας χαρακτήρας τελεία (υποδιαστολή) . ακολουθούμενος από ακέραιο αριθμό με ένα ή περισσότερα ψηφία. Η τελεία . είναι ιδικός χαρακτήρας συνεπώς χρησιμοποιούμε τη σειρά διαφυγής \. .

| : λογικός τελεστής ή

$\backslash d+\backslash.\backslash d+$  : ακέραιος αριθμός με ένα ή περισσότερα ψηφία και ο χαρακτήρας τελεία (υποδιαστολή) . (χρησιμοποιώντας τη σειρά διαφυγής \. ), ακολουθούμενος από ακέραιο αριθμό με ένα ή περισσότερα ψηφία

$(\backslash d+[eE]-?\backslash d+(\backslash.\backslash d+)?|\backslash d+\backslash.\backslash d+([eE]-?\backslash d+(\backslash.\backslash d+)?)?$  : όλα μέσα στην παρένθεση μπορούν να επαναλαμβάνονται καμία ή μια φορά

Δεν αναγράφεται ανάλυση σε επαναλαμβανόμενες εκφράσεις.

## Σχολιασμός παραδειγμάτων

Δίπλα από τα παραδείγματα εμφανίζεται το αποτέλεσμα που επεστράφη σε δοκιμές με την εκτέλεση του fsm κώδικα, όπως και η επεξήγηση του αποτελέσματος.

- 3.14 → GOOD [YES]  
Αποδεκτό διότι περιέχει αριθμούς κινούμενης υποδιαστολής.



- 10.0 → GOOD [YES]  
Αποδεκτό διότι περιέχει αριθμούς κινούμενης υποδιαστολής.
- 0.001 → GOOD [YES]  
Αποδεκτό διότι περιέχει αριθμούς κινούμενης υποδιαστολής.
- 1e100 → GOOD [YES]  
Αποδεκτό διότι περιέχει ύψωση σε δύναμη.
- 14e1.2 → GOOD [YES]  
Αποδεκτό διότι περιέχει ύψωση σε δύναμη και αριθμούς κινούμενης υποδιαστολής.
- 0e-1.2 → GOOD [YES]  
Αποδεκτό διότι περιέχει ύψωση σε δύναμη και αριθμούς κινούμενης υποδιαστολής.
- 0e0 → GOOD [YES]  
Αποδεκτό διότι περιέχει ύψωση σε δύναμη.
- 3.1e1 → GOOD [YES]  
Αποδεκτό διότι περιέχει ακραίους, ύψωση σε δύναμη και αριθμούς κινητής υποδιαστολής.
- .10 → BAD [fsm: in float.fsm, state 'bad' input 1 not accepted]  
Μη αποδεκτό διότι ξεκινάει με τελεία.
- e22 → BAD [fsm: in float.fsm, state 'bad' input 2 not accepted]  
Μη αποδεκτό διότι ξεκινάει με ύψωση σε δύναμη (e).
- -22 → BAD [fsm: in float.fsm, state 'bad' input 2 not accepted]  
Μη αποδεκτό διότι είναι προσημασμένος ακέραιος.
- 3- → BAD [fsm: in float.fsm, state 'bad' input \n not accepted]  
Μη αποδεκτό διότι περιέχει έναν ακέραιο και το σύμβολο μείον ( - ) και ενδιάμεσα τους δεν περιέχεται η ύψωση σε δύναμη. Επίσης μετά το μείον δεν υπάρχει κάποιος αριθμός (δεκαδικός ή ακέραιος).
- 1.10.10 → BAD [fsm: in float.fsm, state 'bad' input 1 not accepted]  
Μη αποδεκτό διότι περιέχει τρεις τελείες. (Κανονικά θα έπρεπε να τελειώνει στο 1.10)
- 0e → BAD [NO]  
Μη αποδεκτό διότι περιέχει ύψωση σε δύναμη η οποία δεν ακολουθείται από κάποιον αριθμό.
- 0. → BAD [NO]  
Μη αποδεκτό διότι περιέχει τελεία που δεν ακολουθείται από κάποιον αριθμό.

## 2. Λεκτικές μονάδες (tokens) – Ενιαίο

START=SZ			N11:	0-9	-> N11
SZ:	a-z A-Z _	-> I0		\n	-> GOOD
	1-9	-> N1		*	-> BAD
	0	-> N2	N12:	0-9	-> N12
	"	-> S1		E e	-> N13
	*	-> BAD		\n	-> GOOD
I0:	_ 0-9 a-z A-Z	-> I0		*	-> BAD
	\n	-> GOOD	N13:	-	-> N14
	*	-> BAD		0-9	-> N15
N1:	0-9	-> N1		*	-> BAD
	E e	-> N6	N14:	0-9	-> N15
	.	-> N7		*	-> BAD
	\n	-> GOOD	N15:	0-9	-> N15
	*	-> BAD		.	-> N16
N2:	x X	-> N3		\n	-> GOOD
	E e	-> N6		*	-> BAD
	.	-> N7	N16:	0-9	-> N17
	0-7	-> N5		*	-> BAD
	\n	-> GOOD	N17:	0-9	-> N17
	*	-> BAD		\n	-> GOOD
N3:	a-f A-F 0-9	-> N4		*	-> BAD
	*	-> BAD	S1:	\\	-> S2
N4:	a-f A-F 0-9	-> N4		\n	-> BAD
	\n	-> GOOD		"	-> S6
	*	-> BAD		*	-> S1
N5:	0-7	-> N5	S2:	\\	-> S3
	\n	-> GOOD		"	-> S4
	*	-> BAD		n	-> S5
N6:	-	-> N8		*	-> BAD
	0-9	-> N9	S3:	"	-> S6
	*	-> BAD		\n	-> BAD
N7:	0-9	-> N12		*	-> S1
	*	-> BAD	S4:	"	-> S6
N8:	0-9	-> N9		\n	-> BAD
	*	-> BAD		*	-> S1
N9:	0-9	-> N9	S5:	"	-> S6
	.	-> N10		\n	-> BAD
	\n	-> GOOD		*	-> S1
	*	-> BAD	S6:	\n	-> GOOD
N10:	0-9	-> N11		*	-> BAD
	*	-> BAD	GOOD(OK):		

Figure 3: Κώδικας FSM για το ενιαίο (tokens).



State	a-z A-Z _	[1-9]	0	"	*	_ 0-9 a-z A-Z	ASCII10(/n)	[0-9]	Ee	.	xX	[0-7]	a-f A-F 0-9	(-)	\	n
SZ	I0	N1	N2	S1	BAD	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅
I0	∅	∅	∅	∅	BAD	I0	GOOD	∅	∅	∅	∅	∅	∅	∅	∅	∅
N1	∅	∅	∅	∅	BAD	∅	GOOD	N1	N6	N7	∅	∅	∅	∅	∅	∅
N2	∅	∅	∅	∅	BAD	∅	GOOD	∅	N6	N7	N3	N5	∅	∅	∅	∅
N3	∅	∅	∅	∅	BAD	∅	∅	∅	∅	∅	∅	∅	N4	∅	∅	∅
N4	∅	∅	∅	∅	BAD	∅	GOOD	∅	∅	∅	∅	∅	N4	∅	∅	∅
N5	∅	∅	∅	∅	BAD	∅	GOOD	∅	∅	∅	∅	N5	∅	∅	∅	∅
N6	∅	∅	∅	∅	BAD	∅	∅	N9	∅	∅	∅	∅	∅	N8	∅	∅
N7	∅	∅	∅	∅	BAD	∅	∅	∅	∅	∅	∅	N12	∅	∅	∅	∅
N8	∅	∅	∅	∅	BAD	∅	∅	N9	∅	∅	∅	∅	∅	∅	∅	∅
N9	∅	∅	∅	∅	BAD	∅	GOOD	N9	∅	N10	∅	∅	∅	∅	∅	∅
N10	∅	∅	∅	∅	BAD	∅	∅	N11	∅	∅	∅	∅	∅	∅	∅	∅
N11	∅	∅	∅	∅	BAD	∅	GOOD	N11	∅	∅	∅	∅	∅	∅	∅	∅
N12	∅	∅	∅	∅	BAD	∅	GOOD	N12	N13	∅	∅	∅	∅	∅	∅	∅
N13	∅	∅	∅	∅	BAD	∅	∅	N15	∅	∅	∅	∅	∅	N14	∅	∅
N14	∅	∅	∅	∅	BAD	∅	∅	N15	∅	∅	∅	∅	∅	∅	∅	∅
N15	∅	∅	∅	∅	BAD	∅	GOOD	N15	∅	N16	∅	∅	∅	∅	∅	∅
N16	∅	∅	∅	∅	BAD	∅	∅	N17	∅	∅	∅	∅	∅	∅	∅	∅
N17	∅	∅	∅	∅	BAD	∅	GOOD	N17	∅	∅	∅	∅	∅	∅	∅	∅
S1	∅	∅	∅	S6	S1	∅	BAD	∅	∅	∅	∅	∅	∅	∅	S2	∅
S2	∅	∅	∅	S4	∅	∅	BAD	∅	∅	∅	∅	∅	∅	∅	S3	S5
S3	∅	∅	∅	S6	S1	∅	BAD	∅	∅	∅	∅	∅	∅	∅	∅	∅
S4	∅	∅	∅	S6	S1	∅	BAD	∅	∅	∅	∅	∅	∅	∅	∅	∅
S5	∅	∅	∅	S6	S1	∅	BAD	∅	∅	∅	∅	∅	∅	∅	∅	∅
S6	∅	∅	∅	∅	BAD	∅	GOOD	∅	∅	∅	∅	∅	∅	∅	∅	∅

Πίνακας 6: Πίνακας μεταβάσεων λεκτικών μονάδων (ενιαίο).

## Σχολιασμός παραδειγμάτων

- **Name123** → **GOOD** [YES]  
Αποδεκτό επειδή αρχίζει με γράμμα και περιέχει γράμματα και αριθμούς.
- **\_name123** → **GOOD** [YES]  
Αποδεκτό επειδή αρχίζει με κάτω παύλα και περιέχει γράμματα, αριθμούς και κάτω παύλα.
- **\_123** → **GOOD** [YES]  
Αποδεκτό επειδή αρχίζει με κάτω παύλα και περιέχει αριθμούς και κάτω παύλα.
- **name.123** → **BAD** [fsm: in identifiers.fsm, state 'bad' input 1 not accepted]  
Μη αποδεκτό επειδή περιέχει . (μη αποδεκτός χαρακτήρας για τα αναγνωριστικά).
- **Name 123** → **BAD** [fsm: in identifiers.fsm, state 'bad' input 1 not accepted]  
Μη αποδεκτό επειδή περιέχει τον κενό χαρακτήρα (space) (μη αποδεκτός για τα αναγνωριστικά).
- **123\_name** → **BAD** [fsm: in identifiers.fsm, state 'bad' input 2 not accepted]  
Μη αποδεκτό επειδή αρχίζει με αριθμό.
- **“Test”** → **GOOD** [YES]  
  
Αποδεκτό διότι περικλείονται απο διπλές αποστρόφους “ ” χαρακτήρες εκτός του backslash \, της νέας γραμμής \n και της διπλής αποστρόφου “.
- **“Newline \n”** → **GOOD** [YES]  
  
Αποδεκτό διότι περικλείονται απο διπλές αποστρόφους “ ” χαρακτήρες εκτός του backslash \, της νέας γραμμής \n και της διπλής αποστρόφου “. Η χρήση του \n παρόλο που συμβολίζει την αλλαγή γραμμής (newline) είναι αποδεκτή διότι αναγνωρίζεται ως σειρά διαφυγής και δεν εκτελείται ως αλλαγή γραμμής.
- **“”** → **GOOD** [YES]  
Αποδεκτό διότι είναι συντακτικά ορθή η χρήση των διπλών αποστρόφων “ ” παρόλο που δεν περικλείουν κάποιο χαρακτήρα.
- **“new line”** → **BAD** [fsm: in string.fsm, state 'bad' input l not accepted]  
Μη αποδεκτό διότι εκτελείται αλλαγή γραμμής (\n) πριν λήξει η συμβολοσειρά.
- **“Wrong input** → **BAD** [NO]  
Μη αποδεκτό γιατί παρόλο που υπάρχει διπλή απόστροφος ” στην αρχή της φράσης, δεν υπάρχουν στο τέλος.
- **Wrong input”** → **BAD** [fsm: in string.fsm, state 'bad' input r not accepted]  
Μη αποδεκτό γιατί παρόλο που υπάρχει διπλή απόστροφος ” στο τέλος της φράσης, δεν υπάρχουν στην αρχή.

- 0 → **GOOD** [YES]  
Αποδεκτό διότι είναι οκταδικός ακέραιος που ξεκινάει με 0 και βρίσκεται μέσα στα όρια ενός οκταδικού αριθμού (0-7).
- 0XFF42 → **GOOD** [YES]  
Αποδεκτό διότι είναι δεκαεξαδικός ακέραιος που ξεκινάει με 0x ή 0X και βρίσκεται μέσα στα όρια ενός δεκαεξαδικού αριθμού (0-9, A-F).
- 029123 → **BAD** [fsm: in numbers.fsm, state 'bad' input 1 not accepted]  
Μη αποδεκτό διότι ξεπερνάει τα όρια ενός οκταδικού αριθμού (0-7).
- 0x → **BAD** [NO]  
Μη αποδεκτό διότι είναι δεκαεξαδικός ακέραιος που ξεκινάει από 0x ή 0X αλλά δεν ακολουθείτε από κάποιον άλλο αριθμό (0-9,A-F).
- 3.14 → **GOOD** [YES]  
Αποδεκτό διότι περιέχει αριθμούς κινούμενης υποδιαστολής.
- 0e-1.2 → **GOOD** [YES]  
Αποδεκτό διότι περιέχει ύψωση σε δύναμη και αριθμούς κινούμενης υποδιαστολής.
- 0e → **BAD** [NO]  
Μη αποδεκτό διότι περιέχει ύψωση σε δύναμη η οποία δεν ακολουθείτε από κάποιον αριθμό.
- 0. → **BAD** [NO]  
Μη αποδεκτό διότι περιέχει τελεία που δεν ακολουθείτε από κάποιον αριθμό.

## Αναφορές / Δυσκολίες:

### Κανονικές εκφράσεις:

**Ακέραιοι:** Δεν αντιμετωπίσαμε ιδιαίτερη δυσκολία στην συγγραφή της κανονικής έκφρασης για τους ακεραίους. Μετά την εύρεση της αρχικής έκφρασης, με παραδείγματα καταλάβαμε πως ήταν ελλιπής και δεχόταν σωστά παραδείγματα που ήταν λανθασμένα, όπως το (0x). Η διόρθωση της ήταν αρκετά εύκολη αλλάζοντας το (\*) με (+).

**Αριθμοί κινούμενης υποδιαστολής:** Η αρχική δυσκολία που αντιμετωπίσαμε, ήταν ότι δεχόταν σωστές, στις κανονικές εκφράσεις το “.e” και το “e.”. Δηλαδή μπορούσε να ξεκινάει με τελεία ή με ύψωση σε δύναμη. Αυτό διορθώθηκε με το (\d+), που σημαίνει ότι πρέπει να υπάρχει τουλάχιστον ένα digit ή περισσότερα στην αρχή. Στην συνέχεια αντιμετωπίσαμε το πρόβλημα των ακεραίων που βρέθηκε μέσω παραδειγμάτων στο regex (π.χ.: 1, 3, 3., κ.λ.π.), επειδή πρέπει οπωσδήποτε να είναι float αριθμοί και όχι ακέραιοι φτιάξαμε τον κώδικα με τέτοιο τρόπο έτσι ώστε να ξεκινάει πάντα με digit και (e, E) or (“|”) digit και “.”(κόμμα). Με αυτήν την αλλαγή όμως εμφανίστηκε και άλλο πρόβλημα. Δεχόταν σωστά, τα παραδείγματα: (0e), (3.),..., κ.λ.π. Αυτό το διορθώσαμε με το να υπάρχει μετά από τα digit (e, E) or (“|”) digit “.”(κόμμα) και έστω ένα ή περισσότερα digit.

**Αναγνωριστικά:** Μοναδική δυσκολία που αντιμετωπίσαμε για τα αναγνωριστικά, ήταν πως μετά από την ( ) ή το γράμμα, θα επαναλαμβάνεται συνεχώς το ίδιο μοτίβο. Επιλύσαμε την δυσκολία αυτή, αναζητώντας περισσότερες πληροφορίες για τις κανονικές εκφράσεις και δοκιμάζοντας πολλές διαφορετικές περιπτώσεις.

**Λεκτικά κυριολεκτικά:** Αρχική δυσκολία για τα λεκτικά κυριολεκτικά, ήταν η κατανόηση των πολλαπλών (\) που χρειάστηκαν για την δημιουργία της έκφρασης. Διότι καθ’ όλη την έκφραση , είναι απαραίτητη η ακύρωση των λειτουργιών κάποιων ειδικών χαρακτήρων. Επίσης, κάτι που δεν ξεκαθαρίσαμε είναι αν εν τέλει, χρειάζονται στην αρχή ο χαρακτήρας (^) και στο τέλος της έκφρασης ο χαρακτήρας (\$) .

**Αριθμητικά κυριολεκτικά:** Στην δημιουργία της κανονικής έκφρασης των αριθμητικών κυριολεκτικών, δεν αντιμετωπίσαμε δυσκολίες, καθώς τις είχαμε επιλύσει στους ακεραίους και στους αριθμούς κινούμενης υποδιαστολής. Στην συνέχεια, απλά τα ενώσαμε σε μία κανονική έκφραση και δημιουργήθηκαν τα αριθμητικά κυριολεκτικά.

### Σχέδια αυτόματων πεπερασμένων:

Στα σχέδια αυτόματων πεπερασμένων δεν αντιμετωπίσαμε ιδιαίτερη δυσκολία, απλά η συγγραφή τους ήθελε ιδιαίτερη προσοχή, καθώς ήταν πολύ εύκολο να ξεχάσουμε κάτι ή ακόμα και να μην το συμπεριλάβουμε. Στην συνέχεια επιλέξαμε να χρησιμοποιήσουμε ASCII για: “.”, “\_”, “/n” κ.λπ., διότι είναι πιο εύκολο να τα διακρίνουμε. Τέλος, βάλλαμε πολλαπλές καταστάσεις με ονομασία “BAD” και “GOOD” για την διευκόλυνσή μας, καθώς και για την συνολική εμφάνιση του σχεδίου. Η δυνατότητα απλοποίησης ορισμένων σχεδίων λιγοστεύοντας τις καταστάσεις υπήρχε, αλλά επιλέξαμε καλύτερα να είναι κατανοητά και λιγότερο “όμορφα”.

**Ενιαίο:** Η δυσκολία στο ενιαίο αυτόματο, ήταν η ένωση όλων των επιμέρους σχεδίων και η μετονομασία των καταστάσεων σε κατάλληλα ονόματα. Ο κώδικας FSM ήταν αρκετά απλός, μιας και είχαμε κάθε κώδικα από τα επιμέρους αυτόματα.

## **Λειτουργία:**

Κάθε κώδικας λειτουργεί κανονικά χωρίς κάποιο σφάλμα. Επιπλέον σε κάθε ένα από αυτούς, μετά από την εξαντλητική χρήση τους λάβαμε τα επιθυμητά αποτελέσματα. Ως ομάδα, δεν βρήκαμε κάποια έλλειψη στην εργασία μας.

## **Για την εκτέλεση των προγραμμάτων:**

Για να τρέξουμε το οποιοδήποτε πρόγραμμα χρησιμοποιήσαμε το command prompt και την εντολή:

- fsm.exe “όνομα\_αρχείου”.fsm

## **Προγράμματα για την εκπόνηση της εργασίας:**

Χρησιμοποιήσαμε διάφορα προγράμματα για την συγκεκριμένη εργασία που αναφέρονται παρακάτω:

- fsm.exe
- <https://www.regexpal.com/>
- [Microsoft Visio](#)
- [Microsoft Word](#)

## **Πηγές:**

E-class:

- [Εργασία εξαμήνου](#)
- [Εργαλείο FSM](#)
- [Γλώσσα Uni-C](#)



## Ανάθεση αρμοδιοτήτων – ρόλων:

Παρουσίαση στόχου εργασίας: ΜΟΥΤΖΟΥΡΗΣ ΑΘΑΝΑΣΙΟΣ

### 1. Λεκτικές μονάδες (tokens) – Επιμέρους

#### 1.1 Αναγνωριστικά (ονόματα) / identifiers (names)

Κανονική Έκφραση: ΘΩΜΑΣ ΝΙΚΟΛΑΟΣ, ΠΟΥΛΙΑΝΑΣ ΤΗΛΕΜΑΧΟΣ

Σχήμα & Κώδικας FSM: ΠΛΑΓΟΥ ΑΙΚΑΤΕΡΙΝΗ

Πίνακας Μετάβασης: ΠΛΑΓΟΥ ΑΙΚΑΤΕΡΙΝΗ

Σχολιασμός (εντός) κώδικα FSM: ΜΠΗΛΙΩΝΗ ΠΑΡΑΣΚΕΥΗ

Ανάλυση της κανονικής έκφρασης: ΜΠΗΛΙΩΝΗ ΠΑΡΑΣΚΕΥΗ

Σχολιασμός παραδειγμάτων: ΜΟΥΤΖΟΥΡΗΣ ΑΘΑΝΑΣΙΟΣ

#### 1.2 Λεκτικά κυριολεκτικά (strings)

Κανονική Έκφραση: ΜΠΗΛΙΩΝΗ ΠΑΡΑΣΚΕΥΗ

Σχήμα & Κώδικας FSM: ΜΠΗΛΙΩΝΗ ΠΑΡΑΣΚΕΥΗ

Πίνακας Μετάβασης: ΜΠΗΛΙΩΝΗ ΠΑΡΑΣΚΕΥΗ

Σχολιασμός (εντός) κώδικα FSM: ΜΠΗΛΙΩΝΗ ΠΑΡΑΣΚΕΥΗ

Ανάλυση της κανονικής έκφρασης: ΜΠΗΛΙΩΝΗ ΠΑΡΑΣΚΕΥΗ

Σχολιασμός παραδειγμάτων: ΜΟΥΤΖΟΥΡΗΣ ΑΘΑΝΑΣΙΟΣ, ΜΠΗΛΙΩΝΗ ΠΑΡΑΣΚΕΥΗ

#### 1.3 Αριθμητικά κυριολεκτικά (numbers)

Κανονική Έκφραση: ΘΩΜΑΣ ΝΙΚΟΛΑΟΣ, ΠΟΥΛΙΑΝΑΣ ΤΗΛΕΜΑΧΟΣ

Σχήμα & Κώδικας FSM: ΘΩΜΑΣ ΝΙΚΟΛΑΟΣ, ΠΟΥΛΙΑΝΑΣ ΤΗΛΕΜΑΧΟΣ

Πίνακας Μετάβασης: ΘΩΜΑΣ ΝΙΚΟΛΑΟΣ

Σχολιασμός (εντός) κώδικα FSM: ΜΠΗΛΙΩΝΗ ΠΑΡΑΣΚΕΥΗ

Ανάλυση της κανονικής έκφρασης: ΜΠΗΛΙΩΝΗ ΠΑΡΑΣΚΕΥΗ

Σχολιασμός παραδειγμάτων: ΜΟΥΤΖΟΥΡΗΣ ΑΘΑΝΑΣΙΟΣ

#### 1.4 Ακέραιοι (integers)

Κανονική Έκφραση: ΘΩΜΑΣ ΝΙΚΟΛΑΟΣ

Σχήμα & Κώδικας FSM: ΘΩΜΑΣ ΝΙΚΟΛΑΟΣ

Πίνακας Μετάβασης: ΘΩΜΑΣ ΝΙΚΟΛΑΟΣ

Σχολιασμός (εντός) κώδικα FSM: ΜΠΗΛΙΩΝΗ ΠΑΡΑΣΚΕΥΗ

Ανάλυση της κανονικής έκφρασης: ΜΠΗΛΙΩΝΗ ΠΑΡΑΣΚΕΥΗ

Σχολιασμός παραδειγμάτων: ΜΟΥΤΖΟΥΡΗΣ ΑΘΑΝΑΣΙΟΣ

#### 1.5 Αριθμοί κινούμενης υποδιαστολής (float)

Κανονική Έκφραση: ΠΟΥΛΙΑΝΑΣ ΤΗΛΕΜΑΧΟΣ

Σχήμα & Κώδικας FSM: ΠΟΥΛΙΑΝΑΣ ΤΗΛΕΜΑΧΟΣ

Πίνακας Μετάβασης: ΠΟΥΛΙΑΝΑΣ ΤΗΛΕΜΑΧΟΣ

Σχολιασμός (εντός) κώδικα FSM: ΜΠΗΛΙΩΝΗ ΠΑΡΑΣΚΕΥΗ

Ανάλυση της κανονικής έκφρασης: ΜΠΗΛΙΩΝΗ ΠΑΡΑΣΚΕΥΗ

Σχολιασμός παραδειγμάτων: ΜΟΥΤΖΟΥΡΗΣ ΑΘΑΝΑΣΙΟΣ

### 2. Λεκτικές μονάδες (tokens) – Ενιαίο

Σχήμα & Κώδικας FSM: ΘΩΜΑΣ ΝΙΚΟΛΑΟΣ, ΠΛΑΓΟΥ ΑΙΚΑΤΕΡΙΝΗ

Πίνακας Μετάβασης: ΠΛΑΓΟΥ ΑΙΚΑΤΕΡΙΝΗ

Σχολιασμός (εντός) κώδικα FSM: ΜΠΗΛΙΩΝΗ ΠΑΡΑΣΚΕΥΗ

Σχολιασμός παραδειγμάτων: ΜΟΥΤΖΟΥΡΗΣ ΑΘΑΝΑΣΙΟΣ

Αναφορές / Δυσκολίες: ΘΩΜΑΣ ΝΙΚΟΛΑΟΣ, ΠΟΥΛΙΑΝΑΣ ΤΗΛΕΜΑΧΟΣ

Επιμέλεια εγγράφου τεκμηρίωσης: ΘΩΜΑΣ ΝΙΚΟΛΑΟΣ, ΠΛΑΓΟΥ ΑΙΚΑΤΕΡΙΝΗ