



**Πανεπιστήμιο Δυτικής Αττικής
Σχολή Μηχανικών
Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών
ΕΡΓΑΣΤΗΡΙΟ ΜΕΤΑΓΛΩΤΙΣΤΕΣ ΕΡΓΑΣΙΑ Α2.**

ΗΜΕΡΟΜΗΝΙΑ ΠΑΡΑΔΟΣΗΣ:

Τρίτη 7 Μαΐου 2024 - 11:59 μ.μ.

ΟΜΑΔΑ ΕΡΓΑΣΤΗΡΙΟΥ:

Τμ.Α2(Τετ 14:00-16:00)

Υπευθύνος Ομάδας:

ΙΟΡΔΑΝΑΚΗΣ ΜΙΧΑΛΗΣ

ΟΜΑΔΑ:

5) ΓΕΩΡΓΙΑ ΠΑΠΑΘΑΝΑΣΙΟΥ (ΠΑΔΑ-21390279)

ΠΑΝΑΓΙΩΤ ΣΠΥΡΟΠΑΛΗΣ (ΠΑΔΑ-19390218)

ΦΙΛΙΠΠΟΣ ΠΑΠΑΓΕΩΡΓΙΟΥ(ΠΑΔΑ-21390174)

ΣΑΝΤΑΣ ΑΝΤΩΝΙΟΣ (ΠΑΔΑ-21390199)

Εισαγωγή

Στο εργαστηριακό μάθημα των Μεταγλωττιστών για το Εαρινό Εξάμηνο 2023-2024 θα χρησιμοποιηθεί η γραμματική της γλώσσας Uni-C. Η συγκεκριμένη γλώσσα αποτελεί υποσύνολο της γλώσσας προγραμματισμού C. Αρχικά, το συγκεκριμένο μέρος της Εργασίας(A2), αφορά τις Κανονικές Εκφράσεις που περιγράφουν τις Λεκτικές Μονάδες της Uni-C. Έπειτα, σχεδιάστηκαν τα Διαγράμματα Μετάβασης(Πεπερασμένα αυτόματα) αναγνώρισης των κανονικών εκφράσεων, που προκύπτουν από τις αναφερόμενες παραπάνω Κανονικές Εκφράσεις, με το πρόγραμμα Visio. Επίσης, για το Ενιαίο Διάγραμμα Μετάβασης που αποτελεί ένωση των Πεπερασμένων Αυτόματων. Στη συνέχεια, προσομοιώσαμε το Ενιαίο Αυτόματο σε ένα Γενικό Πίνακα Μεταβάσεων. Επιπροσθέτως, για την αναγνώριση των Λεκτικών Μονάδων της Uni-C, κωδικοποιήσαμε το Γενικό Πίνακα Μεταβάσεων με τη βοήθεια του FSM. Το FSM αποτελεί έναν μεταφραστή που δέχεται την περιγραφή ενός διαγράμματος(κώδικας) , επιβεβαιώνει ότι είναι ντετερμινιστικό και ως αποτέλεσμα απαντάει αν τα εισαγόμενα από τον προγραμματιστή δεδομένα αναγνωρίζονται από το Αυτόματο. Τέλος, εφαρμόστηκε ένας γενικός έλεγχος για όλα τα αρχεία FSM και παρατηρήθηκαν-σχολιάστηκαν τα αποτελέσματα τους

1. Κατανόηση του Αλφαβήτου

Το αλφαβήτο της γλώσσας σας περιλαμβάνει:

- Πεζοί και Κεφαλαίοι Λατινικοί χαρακτήρες: `a-z` και `A-Z`
- Αριθμητικά ψηφία: `0-9`
- Ειδικοί χαρακτήρες: `! " # % & ' () * + , - . / : ; < = > ? [\] ^ _ { | } ~`
- Whitespace χαρακτήρες: κενό (space), tab, νέα γραμμή, τέλος αρχείου (EOF)

Η γλώσσα είναι case-sensitive, πράγμα που σημαίνει ότι θα πρέπει οι κανονικές εκφράσεις και οι καταστάσεις των FSM να διαχωρίζουν ανάμεσα σε πεζά και κεφαλαία γράμματα.

2. Δημιουργία Κανονικών Εκφράσεων

Για την αναγνώριση διαφορετικών τύπων tokens στη γλώσσα σας, θα πρέπει να δημιουργήσουμε κανονικές εκφράσεις για κάθε είδος. Για παράδειγμα:

- Αριθμοί: `[0-9]+`
- Ταυτότητες (Identifiers): `[a-zA-Z][a-zA-Z0-9]*`
- Ειδικοί χαρακτήρες: κάθε χαρακτήρας μπορεί να αναπαρασταθεί ως ένας μόνος του token. Π.χ., για τον χαρακτήρα `+`, η κανονική έκφραση θα είναι `\[+`.
- Whitespace: (πολλαπλοί whitespace χαρακτήρες μπορεί να θεωρούνται ως ένα token για απλούστευση)

3. Χρήση του FSM

Για κάθε τύπο token, ένα FSM μπορεί να δημιουργήσουμε για να πραγματοποιεί τις απαραίτητες μεταβάσεις μεταξύ καταστάσεων ανάλογα με τους εισερχόμενους χαρακτήρες. Για παράδειγμα, ένα FSM για την αναγνώριση αριθμών θα ξεκινά από μια αρχική κατάσταση, θα περνά σε κατάσταση "αριθμός" όταν διαβάζει έναν ψηφίο, και θα παραμένει εκεί όσο συνεχίζει να διαβάζει ψηφία. Καθώς και τους αντίστοιχους πίνακες μετάβασης του καθε ένα.

1.3.1 Αναγνωριστικά (ονόματα)

ΚΑΝΟΝΙΚΗ ΕΚΦΡΑΣΗ:

$([A-Za-z][A-Za-z 0-9]+)$

1. $[A-Za-z]^*$:

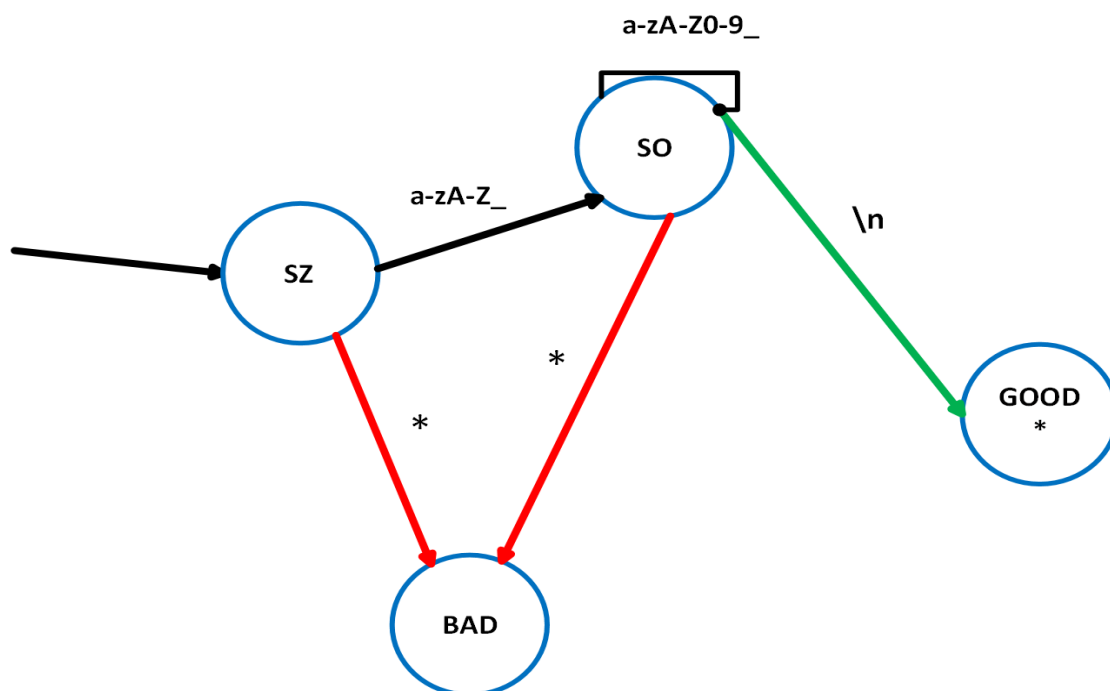
- Αυτό το τμήμα της κανονικής εκφράσεως αντιπροσωπεύει ένα χαρακτήρα που μπορεί να είναι οποιοδήποτε μεγάλο ή μικρό λατινικό γράμμα (A-Z, a-z) ή το υπογράμμιση ('_'). Αυτός ο χαρακτήρας είναι ο πρώτος χαρακτήρας της ταυτότητας και στην πολλές γλώσσες είναι απαραίτητο να μην είναι ψηφίο.

2. $[A-Za-z 0-9]^+$:

- Αυτό το τμήμα καθορίζει τους υπόλοιπους χαρακτήρες που μπορεί να περιέχει η ταυτότητα. Μπορεί να περιλαμβάνει λατινικά γράμματα μεγάλα ή μικρά (A-Z, a-z), ψηφία (0-9), και υπογραμμίσεις ('_'). Το '+' σημαίνει ότι αυτό το μοτίβο πρέπει να εμφανίζεται τουλάχιστον μία φορά, δηλαδή η ταυτότητα πρέπει να έχει τουλάχιστον δύο χαρακτήρες συνολικά.

STATE	A-Za-z_	A-Za-z0-9_	*	\n
SZ	SO	-	BAD	-
SO	-	SO	BAD	GOOD

ΔΙΑΓΡΑΜΜΑ ΜΕΤΑΒΑΣΗΣ



1.3.3 Λεκτικά κυριολεκτικά

Κανονική Εκφραση:

"([^\n\\|\\|\"\\n])*"

Αυτή η regex εξηγεί ως:

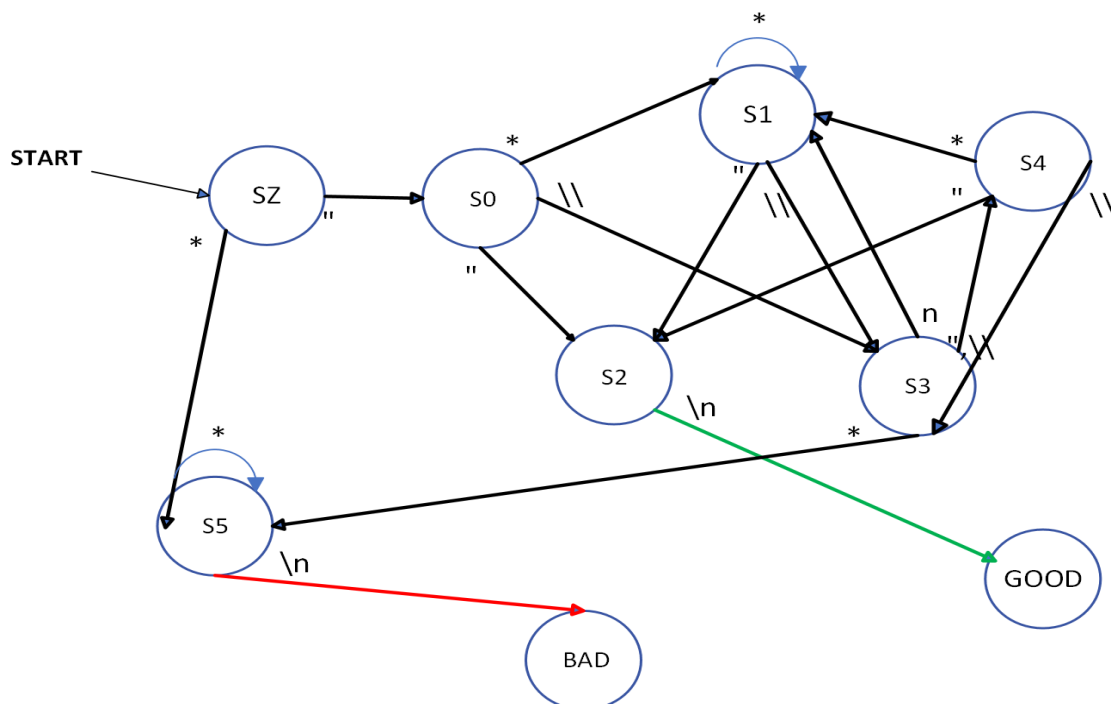
" : Ξεκινάει με ένα διπλό εισαγωγικό.

([^\n\\|\\|\"\\n]): Ταιριάζει με οποιονδήποτε χαρακτήρα εκτός από διπλό εισαγωγικό και backslash ([^\n\\|\\|\"\\n]), ή ταιριάζει με διαφυγόν διπλό εισαγωγικό, backslash ή νέα γραμμή (\\|\"\\n)). Επαναλάβετε αυτό το μοτίβο όσες φορές θέλετε.

" : Τελειώνει με ένα διπλό εισαγωγικό.

STATE	"	\\	n	*	\\n
SZ	S0	-	-	S5	-
S0	S2	S3	-	S1	-
S1	S2	S3	-	S1	-
S2				S5	GOOD
S3	S4	S4	S1	S5	
S4	S2	S3		S1	
S5				S5	BAD

ΔΙΑΓΡΑΜΜΑ ΜΕΤΑΒΑΣΗΣ:



1.3.4 Αριθμητικά κυριολεκτικά

Τα αριθμητικά κυριολεκτικά (numerical literals) περιέχουν ακραίους αριθμούς (integers) και αριθμούς κινητής υποδιαστολής (floating point numbers). Σημειώστε ότι τα αριθμητικά δεν περιλαμβάνουν πρόσημο: η έκφραση -1 στην ουσία είναι σύνθεση του τελεστή '-' και του αριθμητικού 1.

1.3.4.1 Ακέραιοι:

Κανονική Εκφραση:

-?[0-9]+|0[xX][0-9A-Fa-f]+|0[0-7]*

1. **`-?[0-9]+'`**

- **`-?'`** : Αυτό το σύμβολο (**`-?'`**) είναι προαιρετικό και σημαίνει ότι ο αριθμός μπορεί να είναι αρνητικός. Το **`?'`** υποδηλώνει ότι το προηγούμενο στοιχείο (εδώ το **`-?'`**) μπορεί να εμφανιστεί μία ή καθόλου φορές.

- **`[0-9]+'`** : Αντιπροσωπεύει μια ακολουθία από τουλάχιστον έναν ψηφίο (0 έως 9). Το **`+'`** σημαίνει "ένα ή περισσότερα" από το προηγούμενο στοιχείο.

2. **`0[xX][0-9A-Fa-f]+'`**

- **`0[xX]'`** : Ξεκινά με το ψηφίο **`0'`** ακολουθούμενο από το γράμμα **`x'`** ή **`X'`**, υποδηλώνοντας ότι ο ακόλουθος αριθμός είναι δεκαεξαδικός.

- **`[0-9A-Fa-f]+'`** : Ακολουθεί μία ακολουθία τουλάχιστον ενός ψηφίου που μπορεί να είναι από 0 έως 9 ή από A έως F (ή a έως f), αντιπροσωπεύοντας τους δεκαεξαδικούς χαρακτήρες.

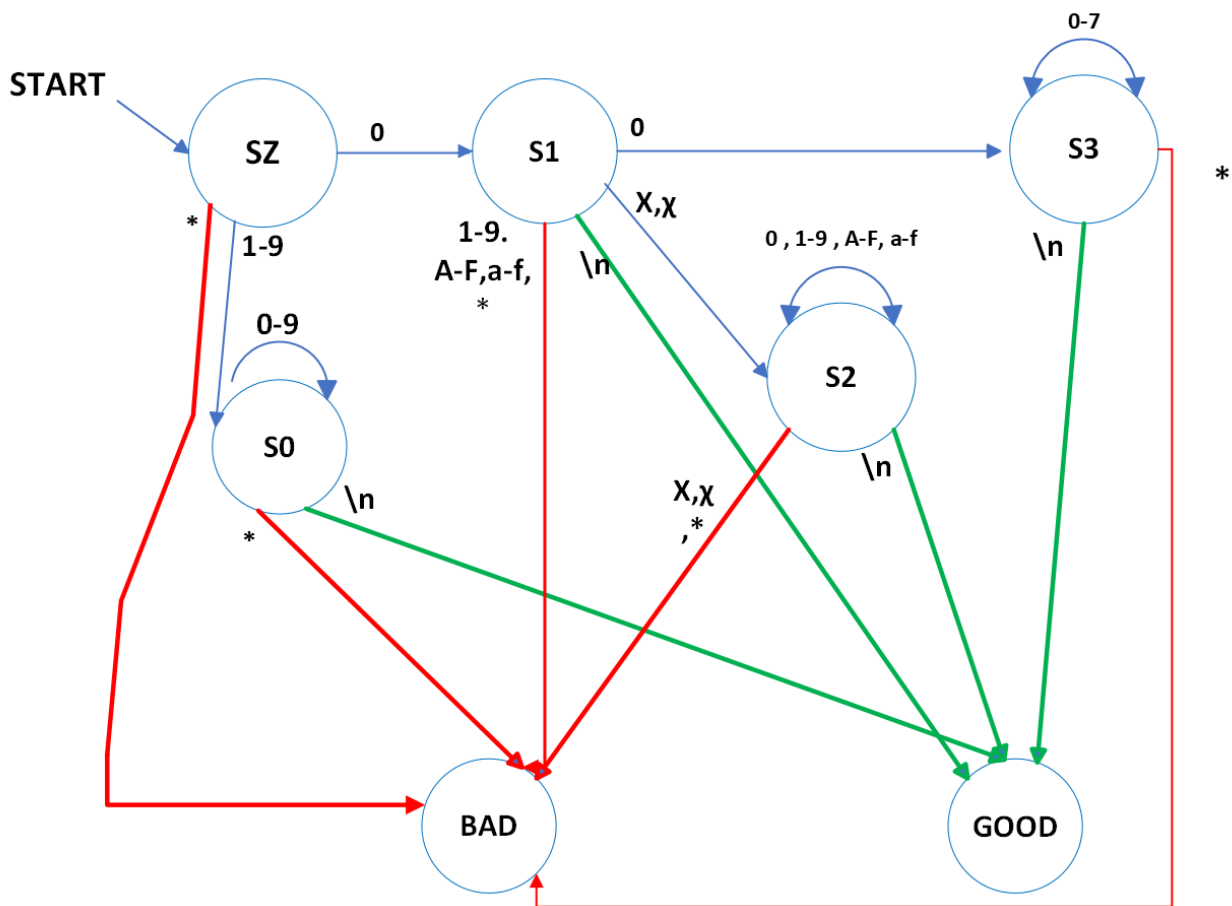
3. **`0[0-7]*'`**

- **`0'`** : Ξεκινά με το ψηφίο **`0'`**, υποδηλώνοντας ότι ο ακόλουθος αριθμός είναι οκταδικός.

- **`[0-7]*'`** : Ακολουθεί μία προαιρετική ακολουθία ψηφίων από 0 έως 7. Το **`*'`** σημαίνει "μηδέν ή περισσότερα" από το προηγούμενο στοιχείο.

STATE	0	1-9	X,x	A-F,a-f	0-7	*	\n
SZ	S1	S0	-	-	-	BAD	BAD
S0	S0	S0	-	-	-	BAD	GOOD
S1	S3	BAD	S2	BAD	S3	BAD	GOOD
S2	S2	S2	BAD	S2	S2	BAD	GOOD
S3	S3	BAD	BAD	BAD	S3	BAD	GOOD

ΔΙΑΓΡΑΜΜΑ ΜΕΤΑΒΑΣΗΣ:



1.3.4.2 Αριθμοί κινούμενης υποδιαστολής

Κανονική Εκφραση:

$[[+-]?([0-9]*[.])?[0-9]+([eE][+-]?[0-9]+)?$

1. $[[+-]?$ - Αυτό το τμήμα αντιπροσωπεύει ένα προαιρετικό σύμβολο πρόσημου ('+' ή '-'). Το '?' σημαίνει ότι το πρόσημο μπορεί να υπάρχει μία ή καθόλου φορές.

2. $([0-9]*[.])?$

- Προαιρετική ομάδα που αντιπροσωπεύει το δεκαδικό μέρος του αριθμού πριν από τη δεκαδική τελεία. Τα $[0-9]^*$ σημαίνει ότι μπορεί να υπάρχουν μηδέν ή περισσότερα ψηφία. Η δεκαδική τελεία $[.]$ είναι escape για να μην ερμηνευτεί ως ειδικός χαρακτήρας της κανονικής εκφράσεως. Το '?' στο τέλος της ομάδας κάνει όλο το σύνολο προαιρετικό.

3. $[0-9]^+$ - Αυτό το τμήμα απαιτεί τουλάχιστον ένα ψηφίο. Είναι απαραίτητο για να αποτελέσει έναν έγκυρο αριθμό, είτε πριν είτε μετά τη δεκαδική τελεία.

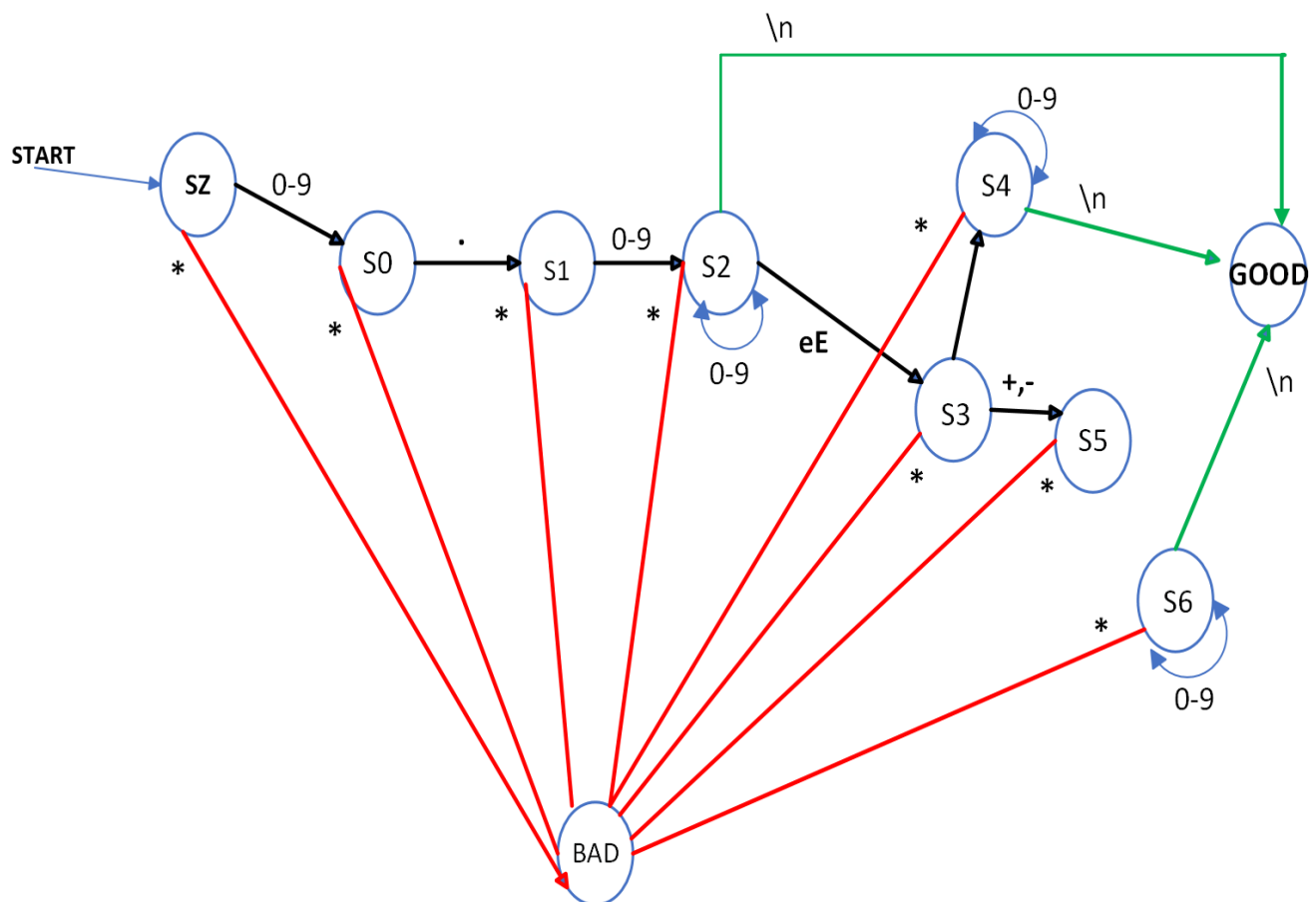
4. $([eE][+-]?[0-9]+)?$

- Προαιρετική ομάδα για επιστημονική σημειογραφία. Τα $[eE]$ σημαίνει ότι μπορεί να ακολουθείται από το γράμμα 'e' ή 'E' (που χρησιμοποιείται στην επιστημονική σημειογραφία για να

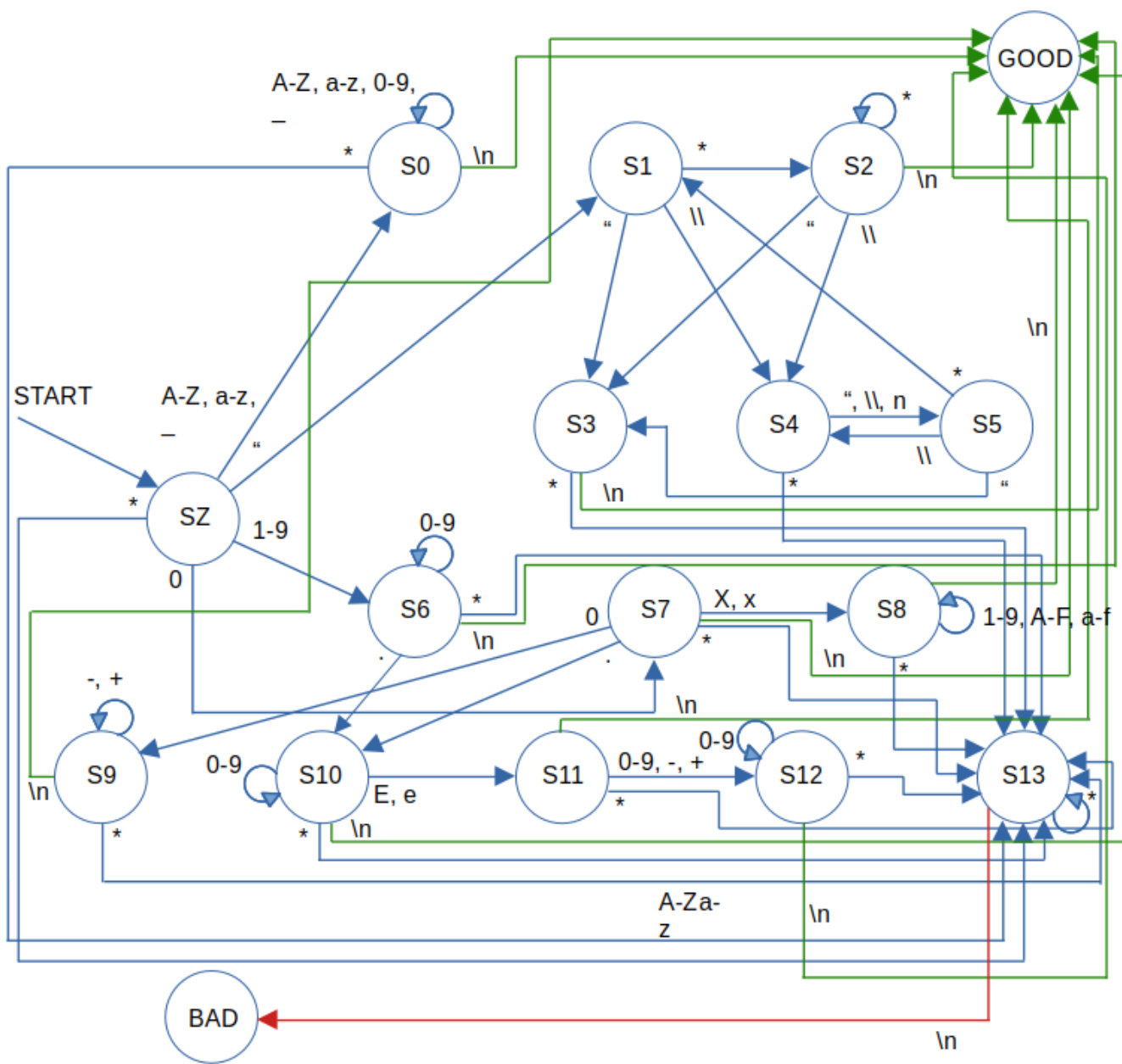
να υποδείξει την δύναμη του 10). Το `[+]?` επιτρέπει ένα προαιρετικό πρόσημο για τον εκθέτη, και το `[0-9]+` απαιτεί τουλάχιστον ένα ψηφίο για τον εκθέτη.

STATE	0-9	.	E e	+ -	*	\n
SZ	S0	-	-	-	BAD	-
S0	S0	S1	-	-	BAD	-
S1	S2	-	-	-	BAD	-
S2	S2	-	S3	-	BAD	GOOD
S3	S4	-	-	S5	BAD	-
S4	S4	-	-	-	BAD	GOOD
S5	S5	-	-	-	BAD	GOOD

ΔΙΑΓΡΑΜΜΑ ΜΕΤΑΒΑΣΗΣ:



Ενιαίο Διάγραμμα Μετάβασης:



Ενιαίος Πινάκας Μετάβασης:

STATE	a-z ,A- Z,_	"	0	1- 9	a-z,A- Z,0-9,_	' \\'	'n'	0-9	.	E,e	'X,x'	'1- 9,A- F,a- f	0- 7	- , +	*	\n
SZ	S0	S1	S7	S6	-	-	-	-	-		-	-	-	-	S13	-
S0	-	-	-	-	S0										S13	GOOD
S1	-	S3	-	-	-	S4	-	-	-		-	-	-	-	S2	-
S2	-	S3				S4									S2	GOOD
S3	-	-	-	-	-	-	-	-	-		-	-	-	-	S13	GOOD
S4	-	S5				S5	S5								S13	-
S5	-	S3	-	-	-	S4	-	-	-		-	-	-	-	S1	-
S6	-	-	-	-	-	-	-	S6	S10						S13	GOOD
S7	-	-	S9	-	-	-	-	-	S10		S8	-	-	-	S13	GOOD
S8	-	-	-	-	-	-	-	-	-		-	S8	-	-	S13	GOOD
S9	-	--	-	-	-	-	-	-	-		-	-	-	S9	S13	GOOD
S10	-	-	-	-	-	-	-	S10		S11	-	-	-	-	S13	GOOD
S11	-	-	-	-	-	-	-	S12	-	-	-	-	-	S12	S13	GOOD
S12	-	-	-	-	-	-	-	S12	-	-	-	-	-	-	S13	GOOD
S13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	S13	BAD

Έλεγχος εκτέλεσης των FSM:

Αναγνωριστικά: Αρχείο identifier.fsm

START=SZ

SZ: a-z A-Z _ -> S0

* -> BAD

S0: a-z A-Z 0-9 _ -> S0

* -> BAD

\n -> GOOD

GOOD(OK):

Ο κώδικας FSM ορίζει μια απλή μηχανή κατάστασης για την αναγνώριση έγκυρων αναγνωριστικών (identifiers) σε έναν προγραμματιστικό κώδικα:

- **Κατάσταση SZ:** Αρχική κατάσταση, αποδέχεται χαρακτήρες από 'a' έως 'z', 'A' έως 'Z', και '_'. Οδηγεί στην κατάσταση S0. Οποιοσδήποτε άλλος χαρακτήρας οδηγεί στην κατάσταση BAD.
- **Κατάσταση S0:** Δεχτείτε αλφαριθμητικούς χαρακτήρες και τον χαρακτήρα '_', παραμένοντας στην ίδια κατάσταση. Οποιοσδήποτε άλλος χαρακτήρας οδηγεί στην κατάσταση BAD, εκτός από τη νέα γραμμή '\n' που οδηγεί στην κατάσταση GOOD.
- **Κατάσταση GOOD (OK):** Σηματοδοτεί την επιτυχή αναγνώριση του αναγνωριστικού.

Έγκυρα:

1

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe  
identifier.fsm
```

```
antonis
```

```
^Z
```

```
YES
```

Το αποτέλεσμα YES που λαμβάνεται μετά την εισαγωγή του ονόματος antonis και του χαρακτήρα τέλους εισόδου (^Z) υποδηλώνει ότι το FSM αναγνώρισε επιτυχώς το όνομα antonis ως έγκυρο αναγνωριστικό. Ο χαρακτήρας ^Z στα Windows σηματοδοτεί το τέλος της εισόδου και είναι φυσιολογικό το πρόγραμμα να τερματίζει την είσοδο εδώ.

2

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe  
identifier.fsm
```

```
Antonis
```

```
^Z
```

```
YES
```

Από το αποτέλεσμα φαίνεται ότι το FSM αναγνωρίζει το Antonis ως ένα έγκυρο αναγνωριστικό. Αυτό δείχνει ότι το FSM διαχειρίζεται σωστά τα ονόματα που ξεκινούν με κεφαλαίο.

3

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe
identifier.fsm

antonis_
^Z
YES
```

Βάσει του αποτελέσματος φαίνεται ότι το FSM προσδιορίζει επιτυχώς την συμβολοσειρά antonis_ ως ένα έγκυρο αναγνωριστικό. Αυτό επιβεβαιώνει περαιτέρω την ικανότητα του FSM να αναγνωρίζει αναγνωριστικά που περιέχουν κάτω παύλες. Η χρήση του χαρακτήρα ^Z για τον τερματισμό της εισόδου λειτουργεί σωστά, δηλώνοντας ότι το πρόγραμμα επεξεργάζεται κατάλληλα την είσοδο μέχρι το τέλος και επιστρέφει το αναμενόμενο αποτέλεσμα.

4

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe
identifier.fsm

antonis7
^Z
YES
```

Το αποτέλεσμα επιβεβαιώνει ότι το FSM μπορεί να αναγνωρίσει αναγνωριστικά που περιλαμβάνουν ψηφία μετά από γράμματα. Αυτός ο τύπος αναγνωριστικού επιτρέπει την ένταξη αριθμών μετά από αρχικά γράμματα παρέχοντας ευελιξία στην ονομασία.

5

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe
identifier.fsm

antonis_santas_7
^Z
YES
```

Το αποτέλεσμα που παίρνουμε όπου το αναγνωριστικό antonis_santas_7 αναγνωρίζεται επιτυχώς ως έγκυρο, δείχνει ότι το FSM έχει καλά ορισμένη δυνατότητα αναγνώρισης περιπλοκότερων αναγνωριστικών που περιλαμβάνουν γράμματα, κάτω παύλες και αριθμούς.

6

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe  
identifier.fsm
```

```
_antonis
```

```
^Z
```

```
YES
```

Το αποτέλεσμα που πήραμε δηλαδή YES για την είσοδο _antonis, δείχνει ότι το FSM επιτρέπει τη χρήση κάτω παύλας (_) στην αρχή των αναγνωριστικών.

Άκυρα:

1

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe  
identifier.fsm
```

```
7antonis
```

```
fsm: in identifier.fsm, state 'bad' input a not accepted
```

Το αποτέλεσμα που πήραμε από την εκτέλεση του identifier.fsm με την είσοδο 7antonis δείχνει ότι το FSM αναγνωρίζει ως άκυρη μια είσοδο που ξεκινά με ψηφίο.

2

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe  
identifier.fsm
```

```
7_antonis
```

```
fsm: in identifier.fsm, state 'bad' input _ not accepted
```

Από το αποτέλεσμα που πήραμε με την είσοδο 7_antonis, φαίνεται ότι το FSM απορρίπτει αναγνωριστικά που ξεκινούν με ψηφίο ακολουθούμενο από κάτω παύλα (_).

3

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe
identifier.fsm

antonis santas

fsm: in identifier.fsm, state 'bad' input s not accepted
```

Το αποτέλεσμα που πήραμε για την είσοδο antonis santas δείχνει ότι το FSM δεν δέχεται τον χαρακτήρα κενού διαστήματος () μεταξύ δύο λέξεων.

4

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe
identifier.fsm

\antonis

fsm: in identifier.fsm, state 'bad' input a not accepted
```

Το μήνυμα σφάλματος που πήραμε κατά την εκτέλεση του identifier.fsm με την είσοδο \antonis δείχνει ότι το FSM απορρίπτει την είσοδο όταν περιλαμβάνει έναν χαρακτήρα escape (\) στην αρχή της.

5

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe
identifier.fsm

@antonis

fsm: in identifier.fsm, state 'bad' input a not accepted
```

Το αποτέλεσμα που πήραμε με την είσοδο @antonis δείχνει ότι το FSM απορρίπτει αναγνωριστικά που ξεκινούν με τον χαρακτήρα @. Αυτή η συμπεριφορά υποδηλώνει ότι το FSM έχει ρυθμιστεί να αποδέχεται μόνο συγκεκριμένους χαρακτήρες ως έγκυρο πρώτο χαρακτήρα σε ένα αναγνωριστικό, πιθανώς γράμματα και κάτω παύλες αποκλείοντας σύμβολα και άλλους ειδικούς χαρακτήρες.

Αριθμοί κινητής υποδιαστολής (αρχείο float.fsm)

```
START=SZ

SZ:  0-9 -> S0
      * -> BAD
```

S0: 0-9 -> S0
 . -> S1
 e,E -> S3
 * -> BAD
 \n -> BAD
 S1: 0-9 -> S2
 * -> BAD
 S2: 0-9 -> S2
 e,E -> S3
 \n -> GOOD
 * -> BAD
 S3: -,+ -> S4
 0-9 -> S5
 * -> BAD
 S4: 0-9 -> S5
 * -> BAD
 S5: 0-9 -> S5
 \n -> GOOD
 * -> BAD
 GOOD(OK):

1. Κατάσταση SZ (Αρχική Κατάσταση):

- Αναγνωρίζει τα ψηφία από 0-9 και μεταβαίνει στην κατάσταση S0. Επιλογές που δεν είναι ψηφία οδηγούν στην κατάσταση BAD δηλαδή σε σφάλμα. Αυτό είναι σημαντικό για την αποφυγή μη αριθμητικών εισόδων από την αρχή.

2. Κατάσταση S0 (Ακέραιοι Αριθμοί):

- Δέχεται συνεχόμενα ψηφία (0-9), επιτρέποντας την καταχώρηση πολυψηφίων αριθμών. Η μετάβαση στην S1 γίνεται με τον χαρακτήρα '.',

προετοιμάζοντας για δεκαδικά ψηφία. Η μετάβαση στην S3 γίνεται με τους χαρακτήρες 'e' ή 'E' για την εισαγωγή επιστημονικής σημειογραφίας.

3. Κατάσταση S1 (Έναρξη Δεκαδικού Αριθμού):

- Απαιτείται τουλάχιστον ένα ψηφίο μετά το δεκαδικό για να μεταβεί στην S2, όπου ο αριθμός θα αναγνωριστεί ως έγκυρος πραγματικός αριθμός.

4. Κατάσταση S2 (Δεκαδικός Αριθμός):

- Συνεχίζει να δέχεται ψηφία και μπορεί να αποδεχθεί έναν εκθέτη (e, E) μεταβαίνοντας στην S3. Ο αριθμός μπορεί να καταλήξει εδώ αν ακολουθείται από το τερματικό \n, και αναγνωρίζεται ως έγκυρος.

5. Κατάσταση S3 (Έναρξη Επιστημονικής Σημειογραφίας):

- Επιτρέπει την εισαγωγή προσήμου για τον εκθέτη (μετάβαση στην S4) ή ψηφίου (μετάβαση στην S5). Αυτό είναι κρίσιμο για τη σωστή αναγνώριση των εκθετών.

6. Κατάσταση S4 και S5 (Επεξεργασία Εκθέτη):

- Στην S4, πρέπει να ακολουθήσει ψηφίο (μετάβαση στην S5), ενώ η S5 δέχεται συνεχόμενα ψηφία ως μέρος του εκθέτη και ολοκληρώνει με \n ως έγκυρο.

Έγκυρα:

1

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe float.fsm
```

2.3

^Z

YES

Από το αποτέλεσμα που πήραμε για την είσοδο 2.3, φαίνεται ότι το FSM λειτουργεί σωστά για την αναγνώριση απλών πραγματικών αριθμών. Το FSM δέχεται αυτόν τον αριθμό ως έγκυρο, ακολουθώντας τις καθορισμένες καταστάσεις και μεταβάσεις που περιγράψαμε στον αρχικό κώδικα.

2

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe float.fsm
```

3.14

^Z

YES

Το αποτέλεσμα που πήραμε για την είσοδο 3.14 δείχνει ότι το FSM λειτουργεί σωστά και αναγνωρίζει έγκυρα τον πραγματικό αριθμό. Αυτό υποδηλώνει ότι οι καταστάσεις και

οι μεταβάσεις που έχουμε ορίσει είναι ορθές και αποτελεσματικές για την αναγνώριση των συμβατικών δεκαδικών μορφών.

3

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe float.fsm
```

```
1e100
```

```
^Z
```

```
YES
```

Το αποτέλεσμα που πήραμε για την είσοδο 1e100 δείχνει ότι το FSM αναγνωρίζει επιτυχώς την επιστημονική μορφή αριθμού (scientific notation). Η είσοδος 1e100 αναπαριστά τον αριθμό 1 πολλαπλασιασμένο με 10 στην εκατοστή δύναμη.

4

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe float.fsm
```

```
3.14e-10
```

```
^Z
```

```
YES
```

Το αποτέλεσμα που πήραμε για την είσοδο 3.14e-10 επιβεβαιώνει περαιτέρω την ικανότητα του FSM να αναγνωρίζει πολύπλοκες αριθμητικές εκφράσεις σε επιστημονική μορφή και συγκεκριμένα τη δυνατότητα να διαχειριστεί αρνητικές δυνάμεις του δέκα.

5

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe float.fsm
```

```
0e0
```

```
^Z
```

```
YES
```

Το αποτέλεσμα που πήραμε για την είσοδο 0e0 δείχνει ότι το FSM συνεχίζει να λειτουργεί αποτελεσματικά, αναγνωρίζοντας ακόμη και την απλούστερη μορφή επιστημονικής σημειογραφίας. Αυτός ο αριθμός, 0e0, αναπαριστά το μηδέν και είναι σημαντικό για το FSM να δέχεται και να αναγνωρίζει σωστά ακόμα και τις πιο βασικές εκφράσεις επιστημονικής μορφής.

6

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe float.fsm
```

0.0101

^Z

YES

Το αποτέλεσμα που πήραμε για την είσοδο 0.0101 δείχνει ότι το FSM αναγνωρίζει επιτυχώς την μορφή αυτού του πραγματικού αριθμού που περιλαμβάνει και προπορευόμενα μηδενικά μετά το δεκαδικό σημείο. Αυτό υποδεικνύει καλά καθορισμένη και εφαρμοσμένη λειτουργικότητα για την αναγνώριση αριθμών που μπορεί να έχουν χαμηλή απόλυτη αξία αλλά ακριβή δεκαδική αναπαράσταση.

7

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe float.fsm
```

1e+10

^Z

YES

Το αποτέλεσμα που πήραμε με την είσοδο 1e+10 δείχνει ότι το FSM διαχειρίζεται επιτυχώς την επιστημονική σημειογραφία με θετικούς εκθέτες. Αυτή η μορφή χρησιμοποιείται συχνά για να εκφράσει μεγάλους αριθμούς με έναν συντομευμένο τρόπο και η ικανότητα του FSM να την αναγνωρίζει επιβεβαιώνει την ευελιξία και την ορθότητα του συστήματος.

8

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe float.fsm
```

01.2

^Z

YES

Το αποτέλεσμα που πήραμε για την είσοδο 01.2 δείχνει ότι το FSM επιτρέπει αριθμούς με προπορευόμενα μηδενικά πριν από το δεκαδικό σημείο.

Άκυρα:

1

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe float.fsm
```

5

^Z

NO

Το αποτέλεσμα που πήραμε με την είσοδο 5 και την απάντηση "NO" επιβεβαιώνει τη σωστή λειτουργία του FSM σύμφωνα με την ρύθμισή του να μην αποδέχεται απλούς ακέραιους χωρίς δεκαδικά ή επιστημονική σημειογραφία ως έγκυρους πραγματικούς αριθμούς.

2

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe float.fsm
```

0.

^Z

NO

Το αποτέλεσμα NO για την είσοδο 0. δείχνει ότι το FSM δεν αναγνωρίζει την ακολουθία ως έγκυρη.

3

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe float.fsm
```

1e

^Z

NO

Η είσοδος 1e και το αποτέλεσμα NO επιβεβαιώνει ότι το FSM δεν θεωρεί αυτή την είσοδο έγκυρη. Αυτό συμβαίνει διότι ο κώδικας FSM απαιτεί επιπλέον είσοδο μετά τον χαρακτήρα e για να ολοκληρώσει την ανάλυση ενός επιστημονικού αριθμού (π.χ., 1e10 ή 1e-3).

4

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe float.fsm  
3.14e  
^Z  
NO
```

Η είσοδος 3.14e ακολουθούμενη από NO δείχνει ότι το FSM δεν θεωρεί την είσοδο αυτή έγκυρη. Αυτό συμβαίνει γιατί η σειρά χαρακτήρων μετά το e πρέπει να περιλαμβάνει είτε έναν αριθμό είτε ένα σύμβολο + ή -, ακολουθούμενο από έναν αριθμό για να συμπληρώσει την έκφραση του εκθετικού αριθμού.

5

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe float.fsm  
10.e10  
fsm: in float.fsm, state 'bad' input 1 not accepted
```

Η είσοδος 10.e10 που καταλήγει στο μήνυμα λάθους "fsm: in float.fsm, state 'bad' input 1 not accepted" υποδηλώνει ότι υπάρχει ένα θέμα με την διαχείριση των εισόδων μετά την τελεία (.) και πριν από το e.

6

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe float.fsm  
1e.10  
fsm: in float.fsm, state 'bad' input 1 not accepted
```

Το αποτέλεσμα όπου η είσοδος 1e.10 οδηγεί στο μήνυμα λάθους "fsm: in float.fsm, state 'bad' input 1 not accepted", δείχνει ότι υπάρχει πρόβλημα στη διαχείριση της εισόδου μετά τη χρήση του e.

7

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe float.fsm  
10e 10  
fsm: in float.fsm, state 'bad' input 1 not accepted
```

Το μήνυμα λάθους που πήραμε, "fsm: in float.fsm, state 'bad' input 1 not accepted" για την είσοδο 10e 10, υποδηλώνει ότι το FSM δεν διαχειρίζεται σωστά τους χαρακτήρες που ακολουθούν το e. Αυτό πιθανώς σημαίνει ότι ένας κενός χαρακτήρας (space) δεν είναι προβλεπόμενος ή επιτρεπτός στην είσοδο μετά από το e.

8

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe float.fsm  
.7
```

```
fsm: in float.fsm, state 'bad' input 7 not accepted
```

Η είσοδος .7 που οδηγεί στο μήνυμα λάθους "fsm: in float.fsm, state 'bad' input 7 not accepted" υποδηλώνει ότι το FSM δεν μπορεί να διαχειριστεί σωστά τις αριθμητικές τιμές που αρχίζουν με τελεία.

9

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe float.fsm  
test
```

```
fsm: in float.fsm, state 'bad' input e not accepted
```

Το αποτέλεσμα που παίρνουμε για την είσοδο test, δηλαδή το μήνυμα λάθους "fsm: in float.fsm, state 'bad' input e not accepted", υποδεικνύει ότι η είσοδος δεν αναγνωρίζεται σωστά από το FSM και οδηγεί σε ένα μη αποδεκτό χαρακτήρα e.

10

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe float.fsm  
0.5.7
```

```
fsm: in float.fsm, state 'bad' input 7 not accepted
```

Το αποτέλεσμα που παίρνουμε για την είσοδο 0.5.7, δηλαδή το μήνυμα λάθους "fsm: in float.fsm, state 'bad' input 7 not accepted", υποδεικνύει ότι η είσοδος δεν αναγνωρίζεται σωστά από το FSM και οδηγεί σε ένα μη αποδεκτό χαρακτήρα συγκεκριμένα το 7.

Ακέραιοι δεκαδικοί (αρχείο int.fsm)

START=SZ

SZ: 0 -> S1

1-9 -> S0

* -> BAD

S0: 0-9 -> S0

\n -> GOOD

* -> BAD

S1: 0 -> S3

X -> S2

x -> S2

\n -> GOOD

1-9,A-F,a-f,* -> BAD

S2: 1-9 -> S2

A-F -> S2

a-f -> S2

\n -> GOOD

X -> BAD

x -> BAD

* -> BAD

S3: 0-7 -> S3

\n -> GOOD

* -> BAD

GOOD(OK):

1.SZ (Αρχική Κατάσταση):

- 0 -> S1: Μεταβαίνει στην κατάσταση S1 αν λάβει τον χαρακτήρα '0'.
- 1-9 -> S0: Μεταβαίνει στην κατάσταση S0 αν λάβει έναν ψηφίο από 1 έως 9.
- -> BAD: Μεταβαίνει στην κατάσταση BAD για οποιοδήποτε άλλο χαρακτήρα.

2.S0 (Διαχείριση Συνεχών Ψηφίων):

- 0-9 -> S0: Μένει στην ίδια κατάσταση αν συνεχίσει να λαμβάνει ψηφία.
- \n -> GOOD: Ολοκληρώνει επιτυχώς την είσοδο αν λάβει τον χαρακτήρα νέας γραμμής.
- -> BAD: Οποιοσδήποτε άλλος χαρακτήρας οδηγεί σε κατάσταση BAD.

3.S1 (Έλεγχος για Ειδικές Καταστάσεις):

- 0 -> S3: Μεταβαίνει στην κατάσταση S3 αν λάβει έναν άλλο '0'.
- X, x -> S2: Μεταβαίνει στην κατάσταση S2 αν λάβει 'X' ή 'x'.
- \n -> GOOD: Μεταβαίνει σε GOOD αν λάβει νέα γραμμή.
- 1-9, A-F, a-f, * -> BAD: Οδηγεί σε BAD για αυτούς τους χαρακτήρες.

4.S2 (Διαχείριση Εξαδικών Αριθμών):

- 1-9, A-F, a-f -> S2: Διατηρεί την κατάσταση S2 με τη λήψη ψηφίων ή εξαδικών χαρακτήρων.
- \n -> GOOD: Μεταβαίνει σε GOOD με νέα γραμμή.
- X, x, * -> BAD: Μεταβαίνει σε BAD με αυτούς τους χαρακτήρες.

5.S3 (Διαχείριση Οκταδικών Αριθμών):

- 0-7 -> S3: Μένει στην S3 με τη λήψη οκταδικών ψηφίων.
- \n -> GOOD: Ολοκληρώνει επιτυχώς με νέα γραμμή.
- -> BAD: Μεταβαίνει σε BAD με άλλους χαρακτήρες.

Στη συνέχεια, παραθέτουμε μερικά έγκυρα παραδείγματα:

1

Για την είσοδο 0:

C:\Users\USER\Desktop\METAΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe int.fsm

0

^Z

YES

Το FSM δέχεται αυτόν τον αριθμό ως έγκυρο.

2

Για την είσοδο 9:

```
C:\Users\USER\Desktop\METAΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe int.fsm
```

9

 $^{\wedge}Z$

YES

Το FSM δέχεται αυτόν τον αριθμό ως έγκυρο.

3

Για την είσοδο 100:

```
C:\Users\USER\Desktop\METAΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe int.fsm
```

100

 Z

YES

Το FSM δέχεται αυτόν τον αριθμό ως έγκυρο.

4

Για την είσοδο 123456789:

```
C:\Users\USER\Desktop\METAΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe int.fsm
```

123456789

 $^{\wedge}Z$

YES

Το FSM δέχεται αυτόν τον αριθμό ως έγκυρο.

5

Για την είσοδο 55:

```
C:\Users\USER\Desktop\METAΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM> fsm.exe int.fsm
```

55

Z

YES

Το FSM δέχεται αυτόν τον αριθμό ως έγκυρο.

Παρακάτω παρουσιάζουμε μερικά άκυρα παραδείγματα:

1

Για την είσοδο 01:

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe int.fsm
```

```
01
```

```
fsm: in int.fsm, state 'bad' input \n not accepted
```

Υπήρξε πρόβλημα κατά την επεξεργασία της εισόδου στο FSM.

2

Για την είσοδο 1.2:

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe int.fsm
```

```
1.2
```

```
fsm: in int.fsm, state 'bad' input 2 not accepted
```

Ο FSM έχει μεταβεί σε μια κατάσταση 'bad' και δεν μπορεί να επεξεργαστεί τον χαρακτήρα '2' που παρέχεται μετά τον χαρακτήρα '.' στην είσοδο 1.2.

3

Για την είσοδο 10 0:

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe int.fsm
```

```
10 0
```

```
fsm: in int.fsm, state 'bad' input 0 not accepted
```

Στον FSM υπάρχει ένα σφάλμα στον ορισμό των εισόδων που μπορεί να δεχθεί μια συγκεκριμένη κατάσταση. Στη συγκεκριμένη περίπτωση ο FSM προσπαθεί να επεξεργαστεί την είσοδο '0' όταν βρίσκεται στην κατάσταση 'bad', αλλά αυτή η είσοδος δεν είναι αποδεκτή για αυτήν την κατάσταση.

4

Για την είσοδο 1n:

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe int.fsm
```

```
1n
```

```
fsm: in int.fsm, state 'bad' input \n not accepted
```

Η συγκεκριμένη είσοδος δεν είναι αποδεκτή όταν ο FSM βρίσκεται στην κατάσταση 'bad'.

5

Για την είσοδο 1j:

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe int.fsm
```

```
1j
```

```
fsm: in int.fsm, state 'bad' input \n not accepted
```

Υπήρξε πρόβλημα κατά την επεξεργασία της συγκεκριμένης εισόδου στο FSM.

6

Για την είσοδο test:

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe int.fsm
```

```
test
```

```
fsm: in int.fsm, state 'bad' input e not accepted
```

Το αποτέλεσμα που παίρνουμε για την είσοδο test, δηλαδή το μήνυμα λάθους " fsm: in int.fsm, state 'bad' input e not accepted ", υποδεικνύει ότι η είσοδος δεν αναγνωρίζεται σωστά από το FSM και οδηγεί σε ένα μη αποδεκτό χαρακτήρα e.

Συμβολοσειρές (αρχείο string.fsm)

```
START=SZ
```

```
SZ: " ->S0
```

```
    * ->S5
```

S0: * ->S1
 \\ ->S3
 " ->S2
 S1: * ->S1
 " ->S2
 \\ ->S3
 S2: \n ->GOOD
 * ->S5
 S3: * ->S5
 \\ ->S4
 " ->S4
 n ->S5
 S4: * ->S1
 \\ ->S3
 " ->S2
 S5: * ->S5
 \n ->BAD
 GOOD(OK):

- SZ (Ζώνη έναρξης):

- Αυτό είναι το αρχικό σημείο απόφασης όπου ξεκινά η συμβολοσειρά (" -> S0`) ή μεταβαίνει σε κατάσταση σφάλματος για οποιαδήποτε άλλη είσοδο (* -> S5`).

- S0 (Άνοιγμα συμβολοσειράς):

- Αυτή η κατάσταση χειρίζεται την αρχή της συμβολοσειράς, αμέσως μετά το αρχικό διπλό εισαγωγικό. Ελέγχει αμέσως αν ακολουθεί άλλο εισαγωγικό (που υποδηλώνει κενή συμβολοσειρά), μια backslash (που ξεκινά μια ακολουθία διαφυγής) ή οποιοσδήποτε άλλος χαρακτήρας (μέρος της συμβολοσειράς).

- S1 (Περιεχόμενο συμβολοσειράς):

- Αυτή είναι η κύρια κατάσταση για τη συσσώρευση περιεχομένου συμβολοσειράς. Δέχεται οποιοσδήποτε χαρακτήρες μέχρι να συναντήσει ένα διπλό εισαγωγικό (για το πιθανό τέλος της συμβολοσειράς) ή μια backslash (για το χειρισμό των escapes).

- S2 (Κλείσιμο συμβολοσειράς):

- Αυτή η κατάσταση εμφανίζεται για να χειριστεί την κατάσταση αμέσως μετά από ένα τελικό εισαγωγικό. Ένα έγκυρο τέλος της συμβολοσειράς σηματοδοτείται από

μια νέα γραμμή (`\n -> GOOD`), ενώ οποιοσδήποτε άλλος χαρακτήρας υποδηλώνει πρόβλημα ή απροσδόκητη συνέχεια (`* -> S5`).

- S3 (Έναρξη διαφυγής)

- : Ξεκινά τον χειρισμό ακολουθίας διαφυγής. Αυτή η κατάσταση μεταβαίνει με βάση τον χαρακτήρα που ακολουθεί την backslash, διαχειριζόμενη κοινές ακολουθίες διαφυγής ή μεταβαίνοντας σε κατάσταση σφάλματος εάν η ακολουθία είναι άκυρη.

- S4 (Συνέχιση διαφυγής):

- Διαχειρίζεται τη συνέχιση ή το κλείσιμο ακολουθιών διαφυγής. Επιστρέφει στον κανονικό χειρισμό συμβολοσειράς ή κλείνει τη συμβολοσειρά αν ακολουθείται από κατάλληλους χαρακτήρες.

- S5 (Κατάσταση σφάλματος):

- Αυτή η κατάσταση καταγράφει και κρατά τυχόν σφάλματα ανάλυσης, επιβεβαιώνοντας με μια νέα γραμμή.

Έγκυρα παραδείγματα:

1

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe string.fsm  
"test 1"  
^Z  
YES
```

Ο FSM λειτουργεί κανονικά για την συμβολοσειρά "test 1" που καταλήγει σε θετικό αποτέλεσμα (YES). Αυτό σημαίνει ότι ο FSM επιτυχώς αναγνωρίζει συμβολοσειρές που πλαισιώνονται από εισαγωγικά και δεν περιλαμβάνουν ειδικούς χαρακτήρες που θα τον οδηγούσαν σε κάποια από τις καταστάσεις BAD

2

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe string.fsm  
"test 3 \n\n"  
^Z  
YES
```

Ο FSM επεξεργάστηκε επιτυχώς τη συμβολοσειρά "test 3 \n\n" και επέστρεψε YES, υποδεικνύοντας ότι η είσοδος ήταν αποδεκτή και ότι ο FSM ολοκλήρωσε τη διαδικασία σωστά.

3

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe string.fsm
```

```
"test 5 \ \ "
```

```
^Z
```

```
YES
```

Ο FSM διαχειρίστηκε σωστά την είσοδο "test 5 \ \", επιστρέφοντας το YES. Αυτό σημαίνει ότι η είσοδος, η οποία περιλαμβάνει διαφυγές των χαρακτήρων διαφυγής (backslashes) \, έχει επεξεργαστεί σωστά.

4

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe string.fsm
```

```
"test 4 \"antonis\" "
```

```
^Z
```

```
YES
```

Ο FSM σου επεξεργάστηκε επιτυχώς τη συμβολοσειρά "test 4 \"antonis\" ", καταλήγοντας σε ένα YES. Αυτό υποδηλώνει ότι ο FSM κατανόησε και διαχειρίστηκε σωστά την εισαγωγή διαφυγής για τα εισαγωγικά (\), επιτρέποντας την εμφάνιση εισαγωγικών μέσα στη συμβολοσειρά χωρίς να τερματίζουν την ανάγνωση της συμβολοσειράς.

5

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe string.fsm
```

```
""
```

```
^Z
```

```
YES
```

Ο FSM διαχειρίζεται σωστά και τις κενές συμβολοσειρές που περικλείονται από διπλά εισαγωγικά " ". Το γεγονός ότι ο FSM επέστρεψε YES υποδεικνύει ότι αυτή η είσοδος θεωρήθηκε έγκυρη.

Άκυρα παραδείγματα:

1

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe string.fsm
```

```
10
```

```
fsm: in string.fsm, state 'bad' input 0 not accepted
```

Το αποτέλεσμα που παίρνουμε όταν δίνουμε την είσοδο 10 δείχνει ότι ο FSM αντιμετωπίζει δυσκολίες με την επεξεργασία αριθμητικών εισόδων εκτός συμβολοσειρών. Το μήνυμα σφάλματος "state 'bad' input 0 not accepted" υποδηλώνει ότι ο FSM δεν έχει ορίσει μια κατάλληλη διαδρομή ή κατάσταση για τη διαχείριση του χαρακτήρα '0'.

2

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe string.fsm
```

```
test 1
```

```
fsm: in string.fsm, state 'bad' input e not accepted
```

Το αποτέλεσμα που παίρνουμε με την είσοδο test 1 και το μήνυμα σφάλματος που αναφέρει ότι ο χαρακτήρας 'e' δεν γίνεται αποδεκτός στην κατάσταση 'bad', υποδηλώνει ότι ο FSM δεν έχει κατάλληλη μετάβαση για να χειριστεί κείμενο εκτός συγκεκριμένων καταστάσεων που αναμένουν συμβολοσειρές περικλεισμένες σε εισαγωγικά.

3

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe string.fsm
```

```
"test 3 \"
```

```
^Z
```

NO

Η είσοδος "test 3 \" και το αποτέλεσμα NO επιβεβαιώνουν τη σωστή λειτουργία του FSM σε περιπτώσεις όπου η συντακτική της εισόδου δεν είναι σωστή. Σε αυτή την περίπτωση, ο χαρακτήρας διαφυγής (\) χρησιμοποιείται λανθασμένα στο τέλος της συμβολοσειράς, υπονοώντας ότι θα ακολουθούσε κάποιος ειδικός χαρακτήρας αλλά η είσοδος τελειώνει απότομα. Αυτό αφήνει τα διπλά εισαγωγικά ανοιχτά και καθιστά τη συμβολοσειρά ανολοκλήρωτη και άκυρη.

4

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe string.fsm
"test
"
fsm: in string.fsm, state 'bad' input " not accepted
```

Το αποτέλεσμα που παίρνουμε με την είσοδο "test\n" οδήγησε στην κατάσταση 'bad', υποδηλώνοντας ότι ο FSM δεν μπόρεσε να διαχειριστεί σωστά τον χαρακτήρα νέας γραμμής (\n) εντός των διπλών εισαγωγικών και ακολούθως το κλείσιμο με διπλά εισαγωγικά (").

5

```
C:\Users\USER\Desktop\MΕΤΑΓΛΩΤΤΙΣΤΕΣ\ΚΩΔΙΚΑΣ FSM>fsm.exe string.fsm
"test \ n"
fsm: in string.fsm, state 'bad' input n not accepted
```

Το αποτέλεσμα που παίρνουμε δείχνει ότι ο FSM αντιμετωπίζει προβλήματα με την επεξεργασία της εισόδου "test \ n", ιδιαίτερα με την παρουσία του χαρακτήρα 'n' μετά τον χαρακτήρα διαφυγής \ και ένα κενό διάστημα μεταξύ τους.

Εννιαίο (αρχείο final.fsm)

START=SZ

SZ: a-z A-Z _ -> S0
 * -> S13
 " -> S1
 0 -> S7
 1-9 -> S6

S0: a-z A-Z 0-9 _ -> S0
 * -> S13
 \n -> GOOD

S1: * -> S2
 \\ -> S4
 " -> S3

S2: * -> S2
 " -> S3
 \\ -> S4

S3: \n -> GOOD
 * -> S13

S4: * -> S13
 \\ -> S5
 " -> S5
 n -> S13

S5: * -> S1
 \\ -> S4
 " -> S3

S6: 0-9 -> S6
 . -> S10

\n -> GOOD

* -> S13

S7: 0 -> S9

X -> S8

x -> S8

\n -> GOOD

1-9,A-F,a-f,* -> S13

. -> S10

S8: 1-9 -> S8

A-F -> S8

a-f -> S8

\n -> GOOD

X -> S13

x -> S13

* -> S13

S9: 0-7 -> S9

\n -> GOOD

* -> S13

S10: 0-9 -> S10

e,E -> S11

* -> S13

\n -> GOOD

S11: -,+ -> S12

0-9 -> S12

* -> S13

S12: 0-9 -> S12

\n -> GOOD

* -> S13

S13: * -> S13

 \n -> BAD

GOOD(OK):

- SZ (κατάσταση έναρξης)

- Κατευθύνει σε διαφορετικές FSM με βάση τον αρχικό χαρακτήρα:
 - → S0 για αναγνωριστικά.
 - `` → S1 για συμβολοσειρές.
 - `0` → S7 για το μηδέν που μπορεί να οδηγήσει σε δεκαεξαδικό, οκταδικό ή ένα κινητό δεκαδικό.
 - `1-9` → S6 για δεκαδικούς αριθμούς.
 - ``*` → S13 για οποιονδήποτε άλλο χαρακτήρα, που οδηγεί σε χειρισμό σφάλματος.

- S0: Αναγνωριστικά

- Χειρίζεται αναδρομικά αλφάβητα, ψηφία και υπογράμμιση για τον σχηματισμό αναγνωριστικών.
- Τερματίζει την επεξεργασία αναγνωριστικών και δηλώνει GOOD σε newline.
- Μεταβαίνει σε S13 σε μη έγκυρους χαρακτήρες για αναγνωριστικά.

- S1 έως S5: Συμβολοσειρές

- Το S1 ξεκινά την επεξεργασία συμβολοσειρών, χειριζόμενο το escaping (`\`), το τέλος της συμβολοσειράς (``), ή οποιονδήποτε άλλο χαρακτήρα (`*`).
- Το S2 συνεχίζει την επεξεργασία συμβολοσειρών με τις ίδιες επιλογές όπως το S1.
- Το S3 σηματοδοτεί το τέλος μιας ακολουθίας συμβολοσειρών και ελέγχει αν υπάρχει newline για να δηλώσει GOOD.
- Το S4 χειρίζεται ακολουθίες διαφυγής και μεταβάσεις με βάση το αν ο επόμενος χαρακτήρας είναι διαφυγής, εισαγωγικά ή `n`.
- Το S5 επεξεργάζεται τον χαρακτήρα μετά από μια ακολουθία διαφυγής και επιστρέφει στην επεξεργασία συμβολοσειρών.

- S6: Δεκαδικοί αριθμοί

- Επεξεργάζεται συνεχή ψηφία και μεταβαίνει στην επεξεργασία κινητής υποδιαστολής με `.`.
- Δηλώνει GOOD σε newline, υποδεικνύοντας το τέλος ενός αριθμού.
- Μεταβαίνει σε επεξεργασία σφαλμάτων σε μη έγκυρους χαρακτήρες.

- S7: Μηδενικοί αριθμοί προθέματος

- Χειρίζεται οκταδικούς αριθμούς που ακολουθούνται από `0-7`.
- Μεταβαίνει σε δεκαεξαδική επεξεργασία με `X` ή `x`.
- Χειρίζεται ένα απλό μηδέν ακολουθούμενο από μια νέα γραμμή ως GOOD.
- Χειρίζεται αριθμούς κινητής υποδιαστολής αν ακολουθείται από `.`.
- Μεταβαίνει σε σφάλμα σε κάθε μη έγκυρο χαρακτήρα, συμπεριλαμβανομένων των αλφαβήτων και των μη έγκυρων ψηφίων.

- S8: Εξαδενικοί αριθμοί

- Επεξεργάζεται δεκαεξαδικούς αριθμούς (`1-9`, `A-F`, `a-f`).
- Δηλώνει GOOD σε νέα γραμμή.
- Μεταβαίνει σε σφάλμα σε άκυρους χαρακτήρες ή πρόσθετα δεκαεξαδικά προθέματα.

- S9: Οκταδικοί αριθμοί

- Επεξεργάζεται οκταδικούς αριθμούς (`0-7`).
- Δηλώνει GOOD σε νέα γραμμή.
- Μεταβαίνει σε σφάλμα σε οποιονδήποτε άλλο χαρακτήρα.

- S10: Αριθμοί κινητής υποδιαστολής

- Χειρίζεται το κλασματικό μέρος των αριθμών κινητής υποδιαστολής μετά το δεκαδικό σημείο.
- Μεταβαίνει στο χειρισμό επιστημονικής σημειογραφίας με `e` ή `E`.
- Δηλώνει GOOD σε newline.
- Μεταβαίνει σε σφάλμα σε οποιονδήποτε άλλο χαρακτήρα.

- S11 και S12: Συμβολισμός

- Ο S11 χειρίζεται την αρχή του εκθέτη, δεχόμενος σύμβολα ή ψηφία.

- Ο S12 συνεχίζει τον εκθέτη με ψηφία.
- Δηλώνει GOOD σε newline.
- Μεταβαίνει σε σφάλμα σε οποιονδήποτε μη έγκυρο χαρακτήρα.

- S13: Χειρισμός σφαλμάτων

- Χειρίζεται συνεχώς οποιονδήποτε χαρακτήρα μέχρι να εμφανιστεί νέα γραμμή, οπότε δηλώνει ΚΑΚΟ, υποδεικνύοντας σφάλμα στην αναγνώριση συμβόλων.

Έγκυρα παραδείγματα:

1

```
PS C:\Users\filip\OneDrive\Desktop\METAGLOTISTES> ./fsm final.fsm
```

filippos

^Z

YES

2

```
PS C:\Users\filip\OneDrive\Desktop\METAGLOTISTES> ./fsm final.fsm
```

123

^Z

YES

3

```
PS C:\Users\filip\OneDrive\Desktop\METAGLOTISTES> ./fsm final.fsm
```

"Hello world!"

^Z

YES

4

```
PS C:\Users\filip\OneDrive\Desktop\METAGLOTISTES> ./fsm final.fsm
```

007

^Z

YES

5

```
PS C:\Users\filip\OneDrive\Desktop\METAGLOTISTES> ./fsm final.fsm
```

0X12

^Z

YES

6

```
PS C:\Users\filip\OneDrive\Desktop\METAGLOTISTES> ./fsm final.fsm
```

0xAF

^Z

YES

7

```
PS C:\Users\filip\OneDrive\Desktop\METAGLOTISTES> ./fsm final.fsm
```

12.123

^Z

YES

8

```
PS C:\Users\filip\OneDrive\Desktop\METAGLOTISTES> ./fsm final.fsm
```

0.1

^Z

YES

9

```
PS C:\Users\filip\OneDrive\Desktop\METAGLOTISTES> ./fsm final.fsm
```

12.1e+9

^Z

YES

10

```
PS C:\Users\filip\OneDrive\Desktop\METAGLOTISTES> ./fsm final.fsm
```

"string \\ \\"

^Z

YES

Άκυρα παραδείγματα:

1

```
PS C:\Users\filip\OneDrive\Desktop\METAGLOTISTES> ./fsm final.fsm
```

```
90identifier
```

```
^Z
```

```
NO
```

2

```
PS C:\Users\filip\OneDrive\Desktop\METAGLOTISTES> ./fsm final.fsm
```

```
"string \"
```

```
^Z
```

```
NO
```

3

```
PS C:\Users\filip\OneDrive\Desktop\METAGLOTISTES> ./fsm final.fsm
```

```
" Hy this is Wrong \n"
```

```
^Z
```

```
NO
```

4

```
PS C:\Users\filip\OneDrive\Desktop\METAGLOTISTES> ./fsm final.fsm
```

```
@
```

```
^Z
```

```
NO
```

5

```
PS C:\Users\filip\OneDrive\Desktop\METAGLOTISTES> ./fsm final.fsm
```

```
1234"Hy"
```

```
^Z
```

```
NO
```

6

```
PS C:\Users\filip\OneDrive\Desktop\METAGLOTISTES> ./fsm final.fsm
```

```
1_Identifier
```

```
^Z
```

NO

7

PS C:\Users\filip\OneDrive\Desktop\METAGLOTISTES> ./fsm final.fsm

200e

^Z

NO

8

PS C:\Users\filip\OneDrive\Desktop\METAGLOTISTES> ./fsm final.fsm

200e

^Z

NO

Αρμοδιότητες μελών:

Για την εκπόνηση της συγκεκριμένης εργασίας υπήρξε δια ζώσης και εξ' αποστάσεως συνεργασία μεταξύ των μελών της ομάδας, που αναγράφονται στο εξώφυλλο. Αναλυτικότερα, τα εξής μέλη ασχολήθηκαν με το σύνολο των τμημάτων της εργασίας:

ΓΕΩΡΓΙΑ ΠΑΠΑΘΑΝΑΣΙΟΥ (ΠΑΔΑ-21390279) :Συνταγή και σχεδίαση για τους Εννιαίους πίνακες μεταβάσεις και Διάγραμμα μεταβάσεων με συνεργασία Παναγιώτη Σπυροπάλη.

ΠΑΝΑΓΙΩΤΗΣ ΣΠΥΡΟΠΑΛΗΣ (ΠΑΔΑ-19390218) : Συνταγή και σχεδίαση για τους
Εννιαίους πίνακες μεταβάσεις και Διάγραμμα μεταβάσεων με συνεργασία
Γεωργία Παπαθανασίου

ΦΙΛΙΠΠΟΣ ΠΑΠΑΓΕΩΡΓΙΟΥ(ΠΑΔΑ-21390174) : Δημιουργία πίνακες μεταβασιών
και διαγράμματα μεταβασιών όλων το τοκενς , των κανονικών εκφρασεων,συνταγή
εννίαιο fsm και testing, καθώς σύνταξη και παρουσιάσής word.

ΣΑΝΤΑΣ ΑΝΤΩΝΙΟΣ (ΠΑΔΑ-21390199):Δημιουργία και testing για τους κωδικές
fsm για τα τοκενς