

## Μεταγλωτιστές 2018

### Προγραμματιστική Εργασία #1

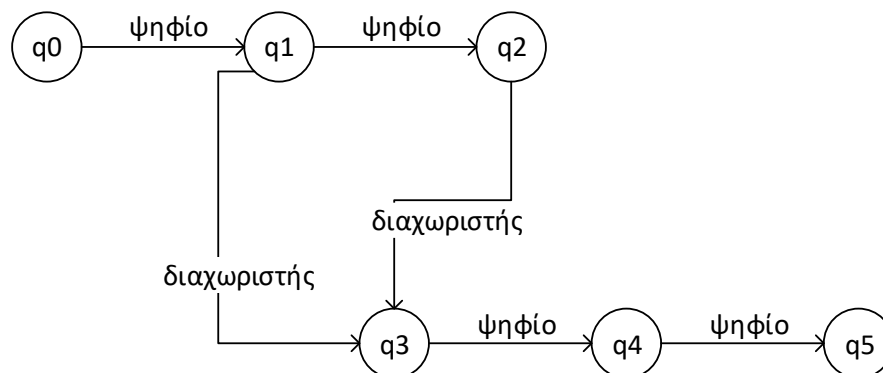
#### Μέμου Παρασκευή

Π2013092

Για την υλοποίηση της εργασίας χρησιμοποιήθηκε ως βάση το πρότυπο αρχείο `adhoclex2a.py`. Στις μεθόδους του πρότυπου προστέθηκε και η μέθοδος `is_number` που επιστρέφει το χαρακτήρα 'd' αν ο χαρακτήρας εισόδου είναι αριθμός και το χαρακτήρα 'o' αν δεν είναι. Η αρχική έκδοση της μεθόδου μπορεί να βρεθεί στο σύνδεσμο <https://www.pythoncentral.io/how-to-check-if-a-string-is-a-number-in-python-including-unicode/>. Η μέθοδος αυτή τροποποιήθηκε ώστε να βοηθήσει στην ομαδοποίηση των χαρακτήρων.

Υπάρχουν τρεις ομάδες χαρακτήρων. Η πρώτη συμβολίζεται με το χαρακτήρα 'd' και περιλαμβάνει τα ψηφία που μπορεί να εισάγει ο χρήστης, η δεύτερη συμβολίζεται με το χαρακτήρα 's' και περιλαμβάνει τους δύο πιθανούς διαχωριστές και η τρίτη συμβολίζεται με το χαρακτήρα 'o' και περιλαμβάνει όλους τους υπόλοιπους χαρακτήρες. Η τοποθέτηση κάθε χαρακτήρα εισόδου στην κατάλληλη ομάδα γίνεται στη μέθοδο `getchar` με τη συνδρομή της `isnumber` που αναφέρθηκε προηγουμένως.

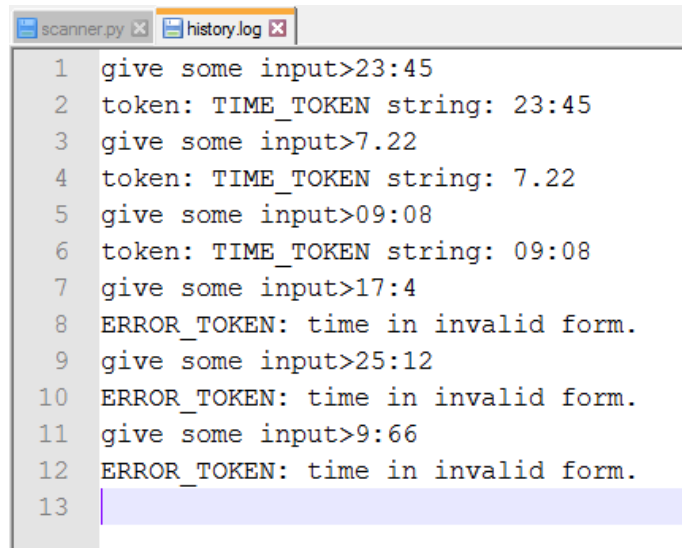
Από το ντετερμινιστικό αυτόματο πεπερασμένων καταστάσεων DFA προκύπτει ο πίνακας μεταβάσεων που μπορεί να βρεθεί στον κώδικα ορισμένο ως `td`. Η μόνη αποδεκτή κατάσταση είναι η `q5` και το DFA απεικονίζεται στο ακόλουθο σχήμα. Αν σε οποιοδήποτε σημείο στην είσοδο δοθεί διαφορετικός χαρακτήρας, τότε επιστρέφεται το token σφάλματος.



Στο παραπάνω σχήμα δεν περιλαμβάνονται κάποιες επιπλέον συνθήκες που καθορίζουν αν θα γίνει η μετάβαση εκτός της εισόδου διαφορετικού χαρακτήρα. Πιο συγκεκριμένα όταν βρισκόμαστε στην κατάσταση `q1` υπάρχουν μπορεί να λάβουμε είτε διαχωριστή είτε ψηφίο. Στην πρώτη περίπτωση οποιοδήποτε ψηφίο που είχαμε λάβει

στην προηγούμενη μετάβαση γίνεται δεκτό και προχωρούμε στην επόμενη κατάσταση. Στη δεύτερη περίπτωση ελέγχουμε το προηγούμενο ψηφίο που είχαμε λάβει. Αν αυτό ήταν μικρότερο του 2, τότε το τρέχων ψηφίο μπορεί να πάρει οποιαδήποτε τιμή και προχωράμε στην επόμενη κατάσταση. Αν αυτό ήταν ίσο με 2, τότε μόνο αν το τρέχων ψηφίο είναι μικρότερο του 4, προχωράμε στην επόμενη κατάσταση. Σε οποιαδήποτε άλλη περίπτωση επιστρέφουμε το token σφάλματος. Επιπλέον, όταν βρισκόμαστε στην κατάσταση q3 ελέγχουμε αν το τρέχων ψηφίο που δεχόμαστε (το πρώτο από τα 2 των λεπτών) είναι μικρότερο του 6 και μόνο τότε προχωράμε στην επόμενη κατάσταση, διαφορετικά επιστρέφουμε το token σφάλματος. Η περιγραφή αυτή αντιστοιχεί στις γραμμές 57-76.

Για να ελεγχτεί η ορθότητα του προγράμματος, δόθηκαν ως είσοδος οι δοκιμαστικές τιμές εισόδου που παρατέθηκαν στην εκφώνηση και τα αποτελέσματα ήταν σωστά όπως μπορεί να φανεί και στο αρχείο history.log του οποίου τα περιεχόμενα παραθέτουμε κι εδώ.



```
1 give some input>23:45
2 token: TIME_TOKEN string: 23:45
3 give some input>7.22
4 token: TIME_TOKEN string: 7.22
5 give some input>09:08
6 token: TIME_TOKEN string: 09:08
7 give some input>17:4
8 ERROR_TOKEN: time in invalid form.
9 give some input>25:12
10 ERROR_TOKEN: time in invalid form.
11 give some input>9:66
12 ERROR_TOKEN: time in invalid form.
13
```

Για τη δημιουργία του αρχείου αυτού, εφόσον δεν επιτρεπόταν η εισαγωγή άλλων modules ανοίξαμε ένα αρχείο για εγγραφή δεδομένων και αναπαράγαμε το περιεχόμενο των εντολών input ή print στην εντολή f.write στις γραμμές 107, 117, 121.