Θεωρία Υπολογισμού
Deterministic Finite Auto

Oulis Evangelos
A.M. 711151051

January 25, 2019

1 Εισαγωγή

Σκοπός της άσκησης είναι η δημιουργία ενός ντετερμινιστικού πεπερασμένου αυτόματου στη μνήμη του υπολογιστή και αναγνώριση λέξεων (σωστών και λάθος) από αυτόν.

Ο κώδικας πρέπει να διαβάζει ένα αρχείο το οποίο θα περιέχει μία περιγραφή ενός ντετερμινιστικού πεπερασμένου αυτόματου, η αποθήκευσή του στη μνήμη του υπολογιστή καθώς και η αναγνώριση εισόδων του χρήστη περνώντας τις μέσα από το αυτόματο. Η απάντησή του θα είναι είτε θετική, δηλαδή ότι η λέξη ήταν σωστή είτε αρνητική, δηλαδή ότι η λέξη δεν ήταν σωστή.

2 Προγραμματισμός

Για τον προγραμματισμό του ντετερμινιστικού πεπερασμένου αυτόματου χρησιμοποίησα την γλώσσα προγραμματισμού Python (v3). Η επιλογή της συγκεκριμένης γλώσσας έγινε για τον λόγο ότι η Python είναι αρκετά δυναμική και ότι περιέχει δομές όπως οι λίστες και οι κατάλογοι, οι οποίες κάνανε σε σύντομο χρόνο την υλοποίηση αυτού του προγράμματος.

Συγκεκριμένα ο κώδικας ξεκινώντας διαβάζει από το αρχείο, διαβάζοντας γραμμή-γραμμή, το πλήθος των καταστάσεων του αυτόματου, το αλφάβητό του, την αρχική κατάσταση καθώς και τη(ις) τελική(ες) κατάσταση(εις).

Έπειτα με την ίδια μέθοδο, γραμμή-γραμμή, διαβάζει εξαντλητικά (μέχρι τέλους) για κάθε κατάσταση και είσοδο την επόμενη κατάσταση. Κάθε κόμβος αποθηκεύεται σε έναν κατάλογο στην Python (dict()) με χαρακτηριστικό κλειδί τον κωδικό της κατάστασης (0, 1, 2, ...n). Σε κάθε περιεχόμενο του καταλόγου αποθηκεύουμε για κάθε κατάσταση με το αντίστοιχο κλειδί μία λίστα η οποία αποτελείται από λίστες τόσες όσες και οι εξερχόμενες ζεύξεις από τον αντίστοιχο κόμβο. Κάθε υπο-λίστα κρατά έναν χαρακτήρα και την επόμενη κατάσταση που πρέπει να μεταβεί.

Στο τέλος κάθε εισόδου χρήστη ελέγχουμε στον κώδικα εάν η κατάσταση που έχουμε καταλήξει είναι μία από τις τελικές καθώς και αν διάβασε κάποιον χαρακτήρα ο οποίος δεν άνηκε στο αλφάβητό μας. Αν κατάσταση που έχουμε καταλήξει είναι μία από τις τελικές και δεν διαβάστηκε κάποιος χαρακτήρας που δεν άνηκε στο αλφάβητο τότε η λέξη ήταν σωστή, δηλαδή σύμφωνα με την γραμματική που έχουμε περιγράψει στο αρχείο **dfa.txt**.

Τέλος το πρόγραμμα δέχεται εισόδους μέχρις ότου ο χρήστης δώσει **exit()**.

Παράδειγμα αναπαράστασης ντετερμινιστικού πεπερασμένου αυτόματου σε αντικείμενο καταλόγου και εσωτερικά λίστας στην Pythom.

Για παράδειγμα:

{
'0' : [['1', '1'], ['0', '0']]
}

Όπου περιγράφει ότι για την κατάσταση '0' ή 'q0', εάν η είσοδος είναι '1' τότε μεταβαίνουμε στην κατάσταση '1' ή 'q1' και εάν η είσοδος είναι '0' μεταβαίνουμε στην κατάσταση '0' ή 'q0'.

Γενικά:

{
'κατάσταση 1' :
[['είσοδος 1', 'επόμενη κατάσταση'], ['είσοδος 2', 'επόμενη κατάσταση']],

'κατάσταση 2' :
[['είσοδος 1', 'επόμενη κατάσταση'], ['είσοδος 2', 'επόμενη κατάσταση']],

....
'κατάσταση n" :
[['είσοδος 1', 'επόμενη κατάσταση'], ['είσοδος 2', 'επόμενη κατάσταση']],
}

3 Αρχεία προς παράδοση

Στον φάκελο βρίσκεται ο πηγαίος κώδικας Python 'dfa.py' και ένα ενδεικτικό αυτόματο εισόδου 'dfa.txt' το οποίο είναι ίδιο με της εκφώνησης. Για την εκτέλεση του προγράμματος εκτελούμε την εντολή **python3 dfa.py** ή **python dfa.py**.