1η Εργασία θεωρίας υπολογισμού και αυτομάτων

Ονοματεπώνυμο: **Χρήστος Δερετζής**
Α.Μ.: **dai18039**

Περιγραφή Προγράμματος

Για την δημιουργία του πεπερασμένου αυτόματου, Αρχικά διαβάζουμε το αρχείο txt με τη μέθοδο **readDataFromFile**, η οποία δέχεται ως παράμετρο το όνομα του αρχείου που θέλουμε να διαβάσουμε. Αρχικά, διαβάζει από το αρχείο τον αριθμό των καταστάσεων του αυτόματου, την αρχική του κατάσταση, τον αριθμό τον τελικό του καταστάσεων, τις τελικές καταστάσεις και τον αριθμό των μεταβιβάσεων του αυτόματου. Για την ανάγνωση των μεταβάσεων του αυτόματου, χρησιμοποιείται η δομή δεδομένων της Python dictionary. Συγκεκριμένα, χρησιμοποιούμε 2 διαφορετικά dictionaries, ένα για τις κανονικές μεταβάσεις του αυτομάτου και ένα για τις ε-μεταβάσεις του αυτομάτου. Κάθε στοιχείο του dictionary έχει την μορφή key: value. Στο dictionary για την μοντελοποίηση των μεταβάσεων των αυτομάτων χρησιμοποιούμε ως key ένα tuple(δομή δεδομένων της python), όπου το πρώτο της στοιχείο είναι η αρχική κατάσταση και το δεύτερο στοιχείο της το στοιχείο του αλφαβήτου που απαιτείται για να γίνει η μετάβαση. Ως value χρησιμοποιείται ένα tuple με την τελική κατάσταση της μετάβασης. Αν για κάποιο ζεύγος (αρχική κατάσταση, στοιχείο) έχουμε παραπάνω από μία τελικές καταστάσεις, τότε αυτό προστίθεται στο αντίστοιχο tuple της τελικής κατάστασης. Τις ε-μεταβάσεις τις κωδικοποιούμε με το στοιχείο ‘@’.

Για τη μοντελοποίηση του αυτόματου δημιουργούμε μία νέα κλάση **Automaton**, η οποία έχει ως γνωρίσματα της τις μεταβάσεις του αυτόματου, τις ε-μεταβάσεις του αυτόματου, την αρχική κατάσταση του αυτόματου, ένα σύνολο με τις τελικές καταστάσεις του αυτομάτου, ένα σύνολο με τις τρέχουσες καταστάσεις του αυτομάτου, τον αριθμό των μεταβάσεων και ένα σύνολο με το αλφάβητο του αυτομάτου. Η κλάση Automaton έχει τέσσερις μεθόδους: transition_to_next_state, is_state_accepted, calculate_alphabet και run_automaton.

Η μέθοδος **is_state_accepted** ελέγχει αν κάποιο από το σύνολο των τρεχουσών καταστάσεων ανήκει στο σύνολο των τελικών καταστάσεων.

Η μέθοδος **calculate_alphabet** υπολογίζει το αλφάβητο του αυτόματου από το dictionary των μεταβάσεων του αυτόματου.

Η μέθοδος **transition_to_next_state** υπολογίζει τις επόμενες καταστάσεις του αυτόματου από τις τρέχουσες καταστάσεις και το input του. Συγκεκριμένα για κάθε μία από τις τρέχουσες καταστάσεις του αυτόματου το αυτόματο ελέγχει μέσω των 2 dictonaries των μεταβάσεων αν υπάρχει ε-μετάβαση στο αυτόματο, αφαιρεί την τρέχουσα κατάσταση από το σύνολο των τρεχουσών καταστάσεων του αυτόματου και προσθέτει σε ένα σύνολο τις επόμενες καταστάσεις του αυτόματου. Στην συνέχεια υπολογίζει τις επόμενες καταστάσεις του αυτόματου που δεν είναι

ε-μεταβιβάσεις και προσθέτει σε ένα σύνολο τις επόμενες καταστάσεις του αυτόματου. Τέλος, ενημερώνει τις τρέχουσες καταστάσεις του αυτόματου με τις επόμενες.

Η μέθοδος **run_automaton** είναι η μέθοδος με την οποία ελέγχουμε αν μία λέξη ανήκει στο αυτόματο. Συγκεκριμένα υπολογίσουμε το αλφάβητο του αυτόματου και προσθέτουμε στο σύνολο των τρεχουσών καταστάσεων την αρχική κατάσταση. Στη συνέχεια για κάθε γράμμα της λέξης ελέγχουμε αρχικά αν το αυτόματο ανήκει στο αλφάβητο του Και στη συνέχεια για κάθε γράμμα μεταβάλλουμε την επόμενη κατάσταση του αυτόματου. Μετά το τέλος του for loop, Ελέγχουμε αν η λέξη ανήκει στο αυτόματο.

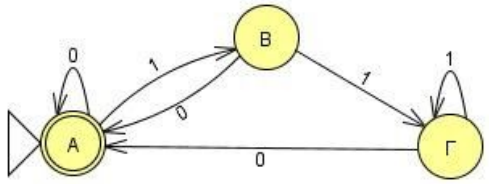
Στην **main** μέθοδο αρχικά διαβάζουμε το αρχείο του κειμένου του αυτόματου. Στη συνέχεια δημιουργούμε ένα αντικείμενο της κλάσης Automaton με τα στοιχεία που πήραμε από το διάβασμα του αυτομάτου. Στη συνέχεια ο χρήστης εισάγει όσες λέξεις θέλει και ελέγχει αν η λέξη ανήκει στο αυτόματο. Αν ο χρήστης εισάγει exit, τότε το πρόγραμμα τερματίζει.

Για να τρέξουμε το πρόγραμμα εισάγουμε σε command line:

python FA.py example_files/name_of_file

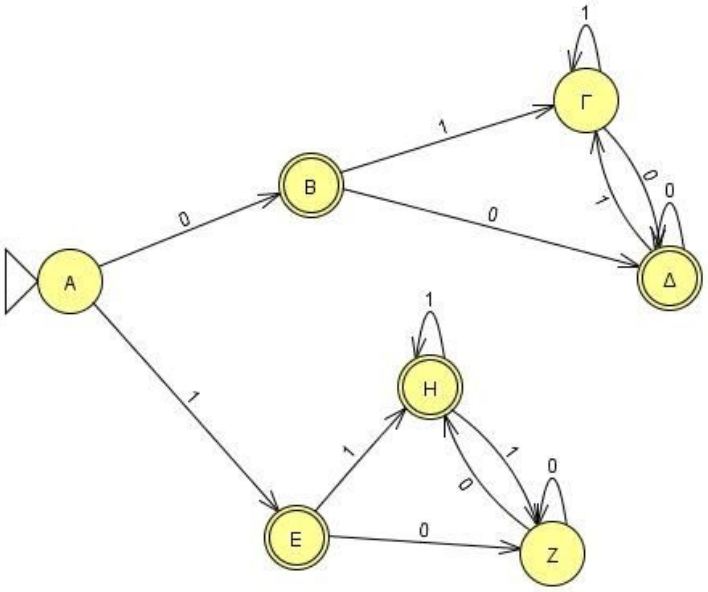
Παραδείγματα Αυτόματων

1. Ντετερμινιστικό Αυτόματο, όπου η λέξη w δεν τελειώνει σε 11



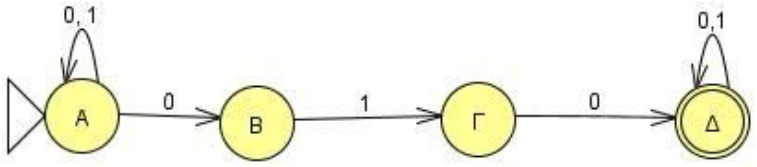

```
C:\Users\user\Documents\UoM\7th_semester\Θεωρία Υπολογισμών και Αυτομάτων\Projec
t_01>python FA.py example_files\FiniteAutomaton01.txt
Number of states:  3
--------------------------------------
Start state:  1
--------------------------------------
Number of final states:  2
--------------------------------------
Final States:  {1, 2}
--------------------------------------
Number of transitions:  6
--------------------------------------
Transitions :  {(1, '0'): (1,), (1, '1'): (2,), (2, '0'): (1,), (2, '1'): (3,), (3, '0'): (1,), (3, '1'): (3,)}
Transitions of e:  {}
Enter a word: 0011001
The word is accepted!
Enter a word: 000110011
The word is not accepted
Enter a word: 00110
The word is accepted!
Enter a word: 011
The word is not accepted
Enter a word: 00110011001
The word is accepted!
Enter a word: 101010110101
The word is accepted!
Enter a word: 100011000100010000111111110
The word is accepted!
Enter a word: 100000000000000011100101011
The word is not accepted
Enter a word: exit

C:\Users\user\Documents\UoM\7th_semester\Θεωρία Υπολογισμών και Αυτομάτων\Project_01>
```

2. Ντετερμινιστικό Αυτόματο, όπου η λέξη w ξεκινά και τελειώνει με το ίδιο σύμβολο



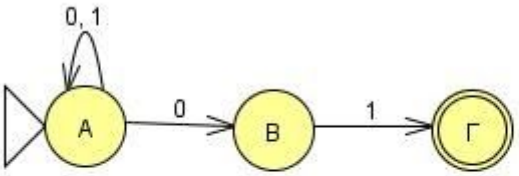

```
C:\Users\user\Documents\UoM\7th_semester\Θεωρία Υπολογισμών και Αυτομάτων\Project_01>python FA.py example_files\FiniteAutomaton02.txt
Number of states:  7
---------------------------------------
Start state:  1
---------------------------------------
Number of final states:  4
---------------------------------------
Final States:  {2, 4, 5, 7}
---------------------------------------
Number of transitions:  14
---------------------------------------
Transitions :  {(1, '0'): (2,), (1, '1'): (5,), (2, '0'): (4,), (2, '1'): (3,), (3, '0'): (4,), (3, '1'): (3,), (4, '1'): (3,), (4, '0'): (4,), (5, '0'): (6,), (5, '1'): (7,), (6, '0'): (6,),
(6, '1'): (7,), (7, '1'): (7,), (7, '0'): (6,)}
Transitions of e:  {}
Enter a word: 00101
The word is not accepted
Enter a word: 0010010
The word is accepted!
Enter a word: 00110010
The word is accepted!
Enter a word: 100010010
The word is not accepted
Enter a word: 1000100101
The word is accepted!
Enter a word: exit
```

3. Μη-Ντετερμινιστικό Αυτόματο, όπου η λέξη w περιέχει την υπολέξη 010



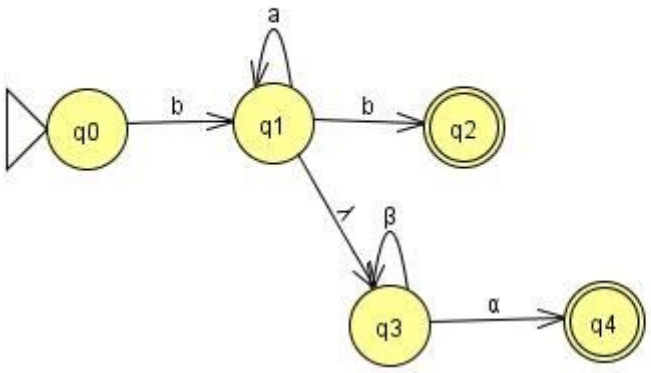

```
C:\Users\user\Documents\UoM\7th_semester\Θεωρία Υπολογισμών και Αυτομάτων\Project_01>python FA.py example_files\Non-Fini
Number of states:  4
--------------------------------------
Start state:  1
--------------------------------------
Number of final states:  1
--------------------------------------
Final States:  {4}
--------------------------------------
Number of transitions:  7
--------------------------------------
Transitions :  {(1, '0'): (1, 2), (1, '1'): (1,), (2, '1'): (3,), (3, '0'): (4,), (4, '0'): (4,), (4, '1'): (4,)}
Transitions of e:  {}
Enter a word: 110
The word is not accepted
Enter a word: 0001
The word is not accepted
Enter a word: 001100
The word is not accepted
Enter a word: 00100100011
The word is accepted!
Enter a word: 010101010
The word is accepted!
Enter a word: 1100
The word is not accepted
Enter a word: 1011110
The word is not accepted
Enter a word: 0100101110
The word is accepted!
Enter a word: exit
```

4. Μη-Ντετερμινιστικό Αυτόματο, όπου η λέξη w τελειώνει σε 01



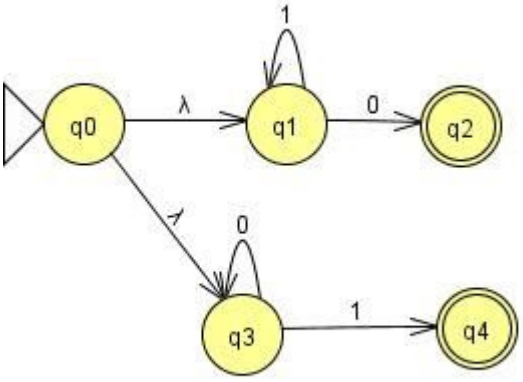

```
Number of states:  3
--------------------------------------
Start state:  1
--------------------------------------
Number of final states:  1
--------------------------------------
Final States:  {3}
--------------------------------------
Number of transitions:  4
--------------------------------------
Transitions :  {(1, '0'): (1, 2), (1, '1'): (1,), (2, '1'): (3,)}
Transitions of e:  {}
Enter a word: 001001001
The word is accepted!
Enter a word: 001010
The word is not accepted
Enter a word: 0011
The word is not accepted
Enter a word: 001100010011101
The word is accepted!
Enter a word: 000100001000110
The word is not accepted
Enter a word: 0101000000000000100101
The word is accepted!
Enter a word: exit
```

5. Μη-Ντετερμινιστικό Αυτόματο με ε-μεταβάσεις




```
Number of states:  5
--------------------------------------
Start state:  1
--------------------------------------
Number of final states:  2
--------------------------------------
Final States:  {3, 5}
--------------------------------------
Number of transitions:  6
--------------------------------------
Transitions :  {(1, 'b'): (2,), (2, 'a'): (2,), (2, 'b'): (3,), (4, 'b'): (4,), (4, 'a'): (5,)}
Transitions of e:  {(2, '@'): (4,)}
Enter a word: bb
The word is accepted!
Enter a word: ba
The word is not accepted
Enter a word: bbb
The word is not accepted
Enter a word: babababab
The word is not accepted
Enter a word: baaaabbbbba
The word is accepted!
Enter a word: bcbbs
The letter (c) does not exist on the alphabet.
The word is not accepted
Enter a word: bab
The word is accepted!
Enter a word: bbbaaabba
The word is not accepted
Enter a word: baaba
The word is accepted!
Enter a word: exit
```

6. Μη-Ντετερμινιστικό Αυτόματο με ε-μεταβάσεις




```
Number of states:  5
--------------------------------------
Start state:  1
--------------------------------------
Number of final states:  2
--------------------------------------
Final States:  {4, 5}
--------------------------------------
Number of transitions:  6
--------------------------------------
Transitions :  {(2, '1'): (2,), (2, '0'): (4,), (3, '0'): (3,), (3, '1'): (5,)}
Transitions of e:  {(1, '@'): (2, 3)}
Enter a word: 1010
The word is not accepted
Enter a word: 1111-
The letter (-) does not exist on the alphabet.
The word is not accepted
Enter a word: 1110
The word is accepted!
Enter a word: 000001
The word is accepted!
Enter a word: 00010010
The word is not accepted
Enter a word: 0010010010
The word is not accepted
Enter a word: 0011001
The word is not accepted
Enter a word: 000000001
The word is accepted!
Enter a word: 1111110
The word is accepted!
Enter a word: exit
```