

**Άσκηση Θεωρία Υπολογισμού**

**Μαλωνάς Κων/νος**

**A.M: 45226**

**Εξ: 11**

## RunAutomato

Σε αυτή την κλάση διαβάζουμε την περιγραφή του αυτομάτου με το input object. Παίρνουμε τις καταστάσεις, τους κανόνες του αυτομάτου και το string από το input.txt σε περίπτωση που θέλουμε να διαβάσουμε είσοδο από το input.txt και το εκχωρούμε στο runs.

Επιλέγουμε με 0 αν θέλουμε να δώσουμε είσοδο από το πληκτρολόγιο. Εκχωρούμε τιμές στο αντικείμενο autom τύπου AutomataMap. Στην μεταβλητή currentActions παίρνουμε τις πιθανές ενέργειες που μπορούμε να κάνουμε από την κατάσταση που βρισκόμαστε. Αν βρεθούμε σε τελική κατάσταση ρωτάμε τον χρήστη αν θέλει να συνεχίσει να δίνει σύμβολα. Αν όχι τερματίζει το πρόγραμμα.

Αν έχουμε επιλέξει με 1 η είσοδος να δίνετε από το input.txt τότε δημιουργούμε αρχικά ένα object aList τύπου ArrayList με αντικείμενα AutomataMap. Για κάθε κατάσταση στην οποία βρισκόμαστε παίρνουμε τις πιθανές ενέργειες για κάθε σύμβολο που παίρνουμε στο εσωτερικό for. Στην συνέχεια μεταβαίνουμε στην αντίστοιχη κατάσταση με την updateCurrentState. Τέλος ελέγχουμε αν εκεί που έχουμε μεταβεί μετά το διάβασμα του τελευταίου συμβόλου είναι τελική κατάσταση. Αν ναι τότε το αυτόματο αποδέχεται αλλιώς απορρίπτει.

## Objects

sc: Scanner object για είσοδο από το πληκτρολόγιο

input: FileUtil object για διάβασμα από αρχείο

rules: ArrayList object που περιέχει object τύπου Rule

runs: ArrayList object που περιέχει String array object για το διάβασμα συμβολοσειράς από το input.txt

Από το αρχείο dfa.txt παίρνουμε την περιγραφή του αυτομάτου.

numOfStates: Ο αριθμός των καταστάσεων του αυτομάτου

actions: Οι ενέργειες που επιτρέπεται να κάνουμε.

finalStates: Οι τελικές καταστάσεις

currentState = Η αρχική κατάσταση που βρισκόμαστε

## **mode == 0**

Instantiation του aum object τύπου AutomataMap. Καλούμε την getPossibleActions μέθοδο που επιστρέφει ένα string από τις δυνατές ενέργειες που μπορούμε να κάνουμε. Στην συνέχεια καλούμε την updateCurrentState που παίρνει το σύμβολο που έχουμε δώσει από το πληκτρολόγιο και εκχωρεί την νέα κατάσταση που έχουμε μεταβεί στην currentState. Στην συνέχεια ελέγχουμε αν βρισκόμαστε σε τελική κατάσταση και αν ο χρήστης θέλει να συνεχίσει να δίνει σύμβολα. Αν όχι το πρόγραμμα τερματίζει.

## **mode == 1**

Κάνουμε declare το aList object που περιέχει object AutomataMap type. Στο εξωτερικό for παίρνουμε στο e την πρώτη γραμμή του runs που περιέχει την συμβολοσειρά που θα διαβάσει το αυτόματο. Δημιουργούμε temp variables που τις εκχωρούμε στο aList. Με το εσωτερικό for παίρνουμε ξεχωριστά κάθε ένα στοιχείο του string που περιέχεται στο runs. Στην συνέχεια με την getPossibleActions παίρνουμε για κάθε στοιχείο της πρώτης γραμμής του αρχείου (get(0)) τις πιθανές ενέργειες. Με την updateCurrentState μεταβαίνουμε στην κατάσταση που υποδεικνύει ο κανόνας. Έπειτα ελέγχουμε αν βρισκόμαστε σε τελική κατάσταση μετά το διάβασμα και του τελευταίου συμβόλου της γραμμής. Αν ναι τυπώνεται μήνυμα επιτυχίας. Στο τέλος από το διάβασμα κάθε γραμμής την διαγράφουμε από το aList τις τιμές και τις δίνουμε ξανά από την αρχή για το διαβασμα της επόμενης συμβολοσειράς.

## **Rule**

### **Attributes**

currentState: Εκχωρείται η κατάσταση που βρισκόμαστε.

transition: Εκχωρείται το σύμβολο που επιτρέπεται να δώσουμε στην συγκεκριμένη κατάσταση.

finalState: Τελική κατάσταση.

### **Constructor**

Με τον constructor εκχωρούμε τιμές στα attributes της κλάσης. Επίσης υπάρχουν get και set μέθοδοι για το currentState, transition και finalState.

### **toString**

Η toString μέθοδος επιστρέφει την κατάσταση που είμαστε, το transition που επιτρέπεται και την κατάσταση που θα βρεθούμε.

## **FileUtil**

### **Attributes**

numOfStates: Αριθμός των καταστάσεων.

actions: Επιτρεπτές ενέργειες, κινήσεις που μπορούμε να κάνουμε.

startingState: Αρχική κατάσταση.

finalStates: Τελική κατάσταση.

### **getMoves()**

Με αυτή την μέθοδο παίρνουμε το ArrayList με το string που θα διαβάσει το αυτόματο. Παίρνουμε το string από το input.txt και αποθηκεύουμε ξεχωριστά τα σύμβολα του string μέσα στο ArrayList runs και το επιστρέφουμε.

### **createAllRules()**

Με αυτή την μέθοδο παίρνουμε από το αρχείο dfa.txt τον αριθμό των καταστάσεων, το αλφάβητο της μηχανής, την αρχική κατάσταση και τέλος τους κανόνες για κάθε κατάσταση που βρισκόμαστε και κάθε επόμενο σύμβολο που διαβάζει η μηχανή. Τέλος υπάρχουν get και set μέθοδοι για τα attributes.

## **AutomataMap**

### **Attributes**

rules: Arraylist rules με αντικείμενα Rule.

actions: Οι ενέργειες που επιτρέπεται να κάνουμε.

numOfStates: Ο αριθμός των καταστάσεων.

startingState: Αρχική κατάσταση.

finalStates: Τελικές καταστάσεις.

### **isFinalState()**

Έλεγχος αν το `currentState == finalStates[i]`. Άμα βρισκόμαστε σε τελική κατάσταση επιστρέφει `true` αλλιώς `false`.

Υπάρχουν `get` και `set` μέθοδοι για τα `attributes`.

### **getPossibleActions()**

Ελέγχουμε σε ποια κατάσταση βρισκόμαστε. Ελέγχουμε ποιοι κανόνες αντιστοιχούν για την κατάσταση μας. Τέλος επιστρέφεται το `string` με τα σύμβολα που επιτρέπεται να δώσουμε σαν είσοδο αναλόγως την κατάσταση που βρισκόμαστε.

### **updateCurrentState()**

Αρχικά εκχωρούμε στο `action` την ενέργεια που έχει διαβάσει. Στην συνέχεια ελέγχουμε αν είναι επιτρεπτή αυτή η ενέργεια μέσα στην `if()`. Τέλος βρίσκουμε το `rule` για το `currentState` μας και το `action` που διαβάστηκε και θέτουμε το `currentState = rule.getFinalState()` – 48 για να πάμε στην επόμενη κατάσταση.