

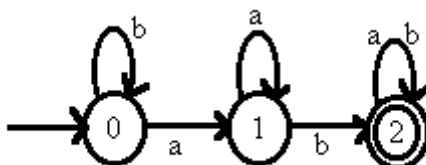
# ΥΠΟΛΟΓΙΣΤΙΚΗ ΛΟΓΙΚΗ ΚΑΙ ΛΟΓΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

2018-2019

## Εργασία 2

### (Προγραμματισμός σε Prolog - Λίστες)

Ένα ντετερμινιστικό αυτόματο πεπερασμένων καταστάσεων (deterministic finite state automaton, DFSA) δέχεται ως είσοδο ένα αλφαριθμητικό και ελέγχει αν είναι αποδεκτό ή όχι. Σε κάθε DFSA υπάρχει ακριβώς μία αρχική και τουλάχιστον μία τελική κατάσταση. Στο παρακάτω διάγραμμα, το DFSA αποτελείται από 3 καταστάσεις (0,1,2), η αρχική κατάσταση είναι η 0 και τελική κατάσταση η 2, ενώ οι κατευθυνόμενες ακμές δείχνουν τις δυνατές μεταβάσεις από μία κατάσταση σε μία άλλη (π.χ. από την 0 μεταβαίνουμε στην 1 με το σύμβολο a, από την 2 παραμένουμε στην 2 με το σύμβολο a ή με το σύμβολο b, κλπ).



Το αυτόματο δέχεται ως είσοδο ένα αλφαριθμητικό και εξετάζει τα σύμβολα από τα οποία αποτελείται με τη σειρά. Τα αποδεκτά σύμβολα του αλφαριθμητικού είναι 2 (a,b). Κάθε σύμβολο το οδηγεί σε μία κατάσταση. Αν στο τέλος του αλφαριθμητικού βρεθεί το αυτόματο σε τελική κατάσταση τότε το αλφαριθμητικό γίνεται αποδεκτό. Το παραπάνω αυτόματο θα έκανε αποδεκτά αλφαριθμητικά όπως τα ab, abb, babbb, aabb αλλά όχι και τα bb, aaa, baa, bba.

- 1) Να αναπαραστήσετε το παραπάνω DFSA στην Prolog.
- 2) Να δημιουργήσετε ένα predicate `check(X,Y)` που ελέγχει αν τα περιεχόμενα της λίστας συμβόλων (αλφαριθμητικό) X είναι συμβατά με το DFSA. Αν η λίστα X περιλαμβάνει σύμβολα που δεν είναι ούτε a ούτε b τότε να τα απομακρύνει από την λίστα X. Η λίστα Y περιλαμβάνει το αποτέλεσμα μετά την απομάκρυνση των ακατάλληλων συμβόλων.

```
?-check([a,b,b,a],Y).  
Y=[a,b,b,a];  
false  
?-check([a,1,b,c,b],Y).  
Y=[a,b,b];  
false
```

- 3) Να δημιουργήσετε ένα predicate `parse(X)` που να ελέγχει αν η λίστα συμβόλων (αλφαριθμητικό) `X` είναι αποδεκτή από το DFSA. Παράδειγμα:

```
?-parse([a,b,b]).
true
?-parse([b,b,b]).
false
```

- 4) Να δημιουργήσετε ένα predicate `parseable(X,Y)` που εφόσον η λίστα συμβόλων (αλφαριθμητικό) `X` δεν είναι αποδεκτή από το DFSA το μετατρέπει σε αποδεκτό προσθέτοντας ή/και αφαιρώντας σύμβολα. Η λίστα `Y` περιλαμβάνει την νέα αποδεκτή λίστα συμβόλων. Πρέπει να επιστρέφεται ακριβώς μία τιμή στο `Y`. Παράδειγμα:

```
?-parseable([a,b,b],Y).
Y=[a,b,b];
false
?-parseable([b,b,b],Y).
Y=[a,b,b,b]
false
```

- 5) Να δημιουργήσετε ένα predicate `parse_all(N)` που σχηματίζει όλα τα δυνατά αλφαριθμητικά που αποτελούνται από σύμβολα `a,b` και έχουν μήκος `N`, ελέγχει αν το καθένα από αυτά είναι αποδεκτό από το DFSA και εμφανίζει τα αποδεκτά αλφαριθμητικά. Παράδειγμα:

```
?-parse_all(2).
ab;
false

?-parse_all(3).
aab;
aba;
abb;
bab;
false
```