

## ΛΥΣΗ ΕΡΩΤΗΜΑΤΟΣ (Α)

### Τεκμηρίωση

Για την λύση του προβλήματος κατασκευάσαμε ένα Ντετερμινιστικό Πεπερασμένο αυτόματο(ΝΠΑ).Αυτό είναι ένα υπολογιστικό μοντέλο αποτελούμενο από έναν πεπερασμένο αριθμό καταστάσεων και μια συνάρτηση μετάβασης, μέσω της οποίας καθορίζονται οι μεταβάσεις από κατάσταση σε κατάσταση, ανάλογα με την είσοδο που δέχεται το αυτόματο. Η έξοδος του αυτόματου θα είναι είτε αποδοχή είτε απόρριψη της εισόδου . Έχει 3 καταστάσεις( $k_0, k_1, k_2$ ) , 3 σύμβολα εισόδου( $X, Y$ , κενό), 2 σύμβολα στοίβας ( $X, \$$ )

$$K = \{ k_0, k_1, k_2 \}$$

$$T = \{ X, Y, E \}$$

$$V = \{ X, \$ \}$$

$$F = \{ k_2 \}$$

### Συναρτήσεις μετάβασης:

Στην αρχή βρισκόμαστε στην κατάσταση  $k_0$  όπου με κενή στοίβα και είσοδο μεταβιβαζόμαστε στην κατάσταση  $k_1$  και προστίθεται το σύμβολο '\$' στην στοίβα. Έπειτα από την κατάσταση  $k_1$  υπάρχουν 4 ενδεχόμενα:

1. Αν είναι στην κατάσταση  $k_1$  στην κορυφή της στοίβας βρίσκεται το '\$' και διαβάσει 'X' τότε παραμένω στην κατάσταση  $k_1$  και προσθέτω 'X' στην κορυφή (πάνω από το '\$').
2. Αν είναι στην κατάσταση  $k_1$  στην κορυφή της στοίβας βρίσκεται το 'X' και διαβάσει 'Y' τότε παραμένω στην κατάσταση  $k_1$  και αφαιρώ το 'X' από την κορυφή.
3. Αν είναι στην κατάσταση  $k_1$  στην κορυφή της στοίβας βρίσκεται το 'X' και διαβάσει 'X' τότε παραμένω στην κατάσταση  $k_1$  και προσθέτω 'X' στην κορυφή .
4. Αν είναι στην κατάσταση  $k_1$  στην κορυφή της στοίβας βρίσκεται το '\$' και διαβάσει το κενό τότε μεταβιβαζόμαι στην κατάσταση  $k_2$  (τελική) αφαιρώ το '\$' από την κορυφή , και η συμβολοσειρά κρίνεται αποδεκτή.

$$P(k_0, e, e) = \{ (k_1, \$) \}$$

$$P(k_1, \$, 'X') = \{ (k_1, 'X') \}$$

$$P(k_1, 'X', 'X') = \{ (k_1, 'X') \}$$

$$P(k_1, 'X', 'Y') = \{ (k_1, e) \}$$

$$P(k_1, \$, e) = \{ (k_2, e) \}$$

## Παραδείγματα λειτουργίας

Παραδείγματα εκφράσεων που αναγνωρίζονται:

- xxyyxy
- XXYYXY
- XxYyxY
- abcXabcY

```
Insert expression:abcXYabcxY
INPUT:
STACK: $
STATE: k1

INPUT: X
STACK: $ X
STATE: k1

INPUT: Y
STACK: $
STATE: k1

INPUT: x
STACK: $ x
STATE: k1

INPUT: Y
STACK: $
STATE: k1

INPUT:
STACK: $
STATE: k2

Valid
Process finished with exit code 0
```

Παραδείγματα εκφράσεων που **δεν** αναγνωρίζονται:

- xyyx
- yxxy
- abcXyy

```
Insert expression:xyyx
INPUT:
STACK: $
STATE: k1

INPUT: x
STACK: $ x
STATE: k1

INPUT: y
STACK: $
STATE: k1

INPUT: y
STACK: $
STATE: k1

Not Valid
Process finished with exit code 0
|
```

## Οδηγίες χρήσης

1. Υποθέτοντας ότι έχουμε εγκαταστημένο έναν GCC compiler
2. Ανοίγουμε ένα παράθυρο cmd ή terminal όπου είναι στο path όπου είναι το αρχείο ASKSH1.cpp
3. Εκτελούμε την εντολή `g++ -o CompilersTask ASKSH1.cpp`
4. Το εκτελούμε με την εντολή `./CompilersTask` σε terminal ή `CompilersTask` σε cmd
5. Ύστερα πληκτρολογούμε την έκφραση που θέλουμε η μηχανή να εξετάσει . Εκτυπώνεται η αλληλουχία βημάτων και αν είναι αποδεκτή εκτυπώνει στην οθόνη Valid διαφορετικά Not Valid.

- Σε περίπτωση που η έκφραση είναι πάνω από 10 χαρακτήρες (X,Y) τότε μας δίνεται η επιλογή να μην την εκτυπώσουμε ολόκληρο γράφοντας No ή N. Όταν το επιλέξουμε εκτυπώνεται μόνο η πρώτη και η τελευταία αλληλουχία βημάτων με ενδιάμεσα τρεις τελείες.

```
Insert expression: xXxxXxYyYyYY
```

```
The expression string seems big and it might take some time to be printed do you wish to continue?[Y/N]:N
```

```
INPUT:
```

```
STACK: $
```

```
STATE: k1
```

```
*
```

```
*
```

```
*
```

```
INPUT: Y
```

```
STACK: $
```

```
STATE: k1
```

```
INPUT:
```

```
STACK: $
```

```
STATE: k2
```

```
Valid
```

```
Process finished with exit code 0
```

- Η μηχανή δέχεται όλα τα σύμβολα απλά αν δεν είναι X ή Y τα αγνοεί

## Βασικά εργαλεία

- Το πρόγραμμα είναι φτιαγμένο σε γλώσσα c++ σε έκδοση STANDARD 14
- Ως δομή δεδομένων χρησιμοποιήθηκε στοίβα από την βιβλιοθήκη της c++