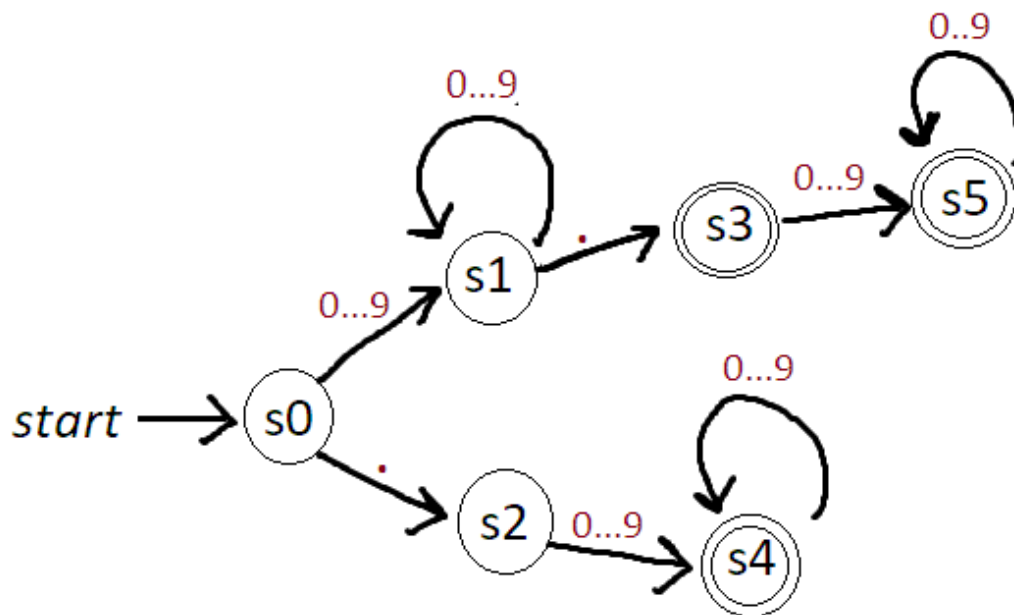


Μεταγλωττιστές 2020

Προγραμματιστική Εργασία #1

Ονοματεπώνυμο: Φουτσιτζή Σοφρονία

A.M.: Π2017063



Αυτόματο για την αναγνώριση κλασματικών αριθμών

Το συγκεκριμένο αυτόματο πεπερασμένων καταστάσεων (FA) αναγνωρίζει κλασματικούς αριθμούς στις μορφές: ✓ 123.56 ✓ 123. ✓ .56789. Ωστόσο, δεν αναγνωρίζει αριθμούς χωρίς την τελεία/υποδιαστολή πχ. 12178 αλλά ούτε και την τελεία μόνο.

- Στην αρχή βρισκόμαστε στην κατάσταση s0, την αρχική κατάσταση.
- Εάν ο πρώτος χαρακτήρας εισόδου είναι 0 έως 9 ( digit) μεταβαίνουμε στην κατάσταση s1, εάν ο πρώτος χαρακτήρας εισόδου είναι τελεία (.) μεταβαίνουμε στην κατάσταση s2.

- Εάν μεταβούμε από την αρχική κατάσταση στη κατάσταση s1( δηλαδή ο πρώτος χαρακτήρας εισόδου είναι αριθμός (0 έως 9)), ο χαρακτήρας εισόδου μπορεί να είναι είτε αριθμός και να παραμείνουμε στην ίδια κατάσταση s1, είτε τελεία οπότε να μεταβούμε στην κατάσταση s3. Η κατάσταση s3 μπορεί να είναι μία τελική μας κατάσταση(κατάσταση αποδοχής), αφού το αυτόματό μας αναγνωρίζει και κλασματικούς αριθμούς της μορφής «12433.». Επίσης, είναι εφικτό να δωθεί κάποιος αριθμούς και να μεταβούμε στην κατάσταση s5, όπου μπορούμε να δώσουμε όσα ακόμα ψηφία/αριθμούς επιθυμούμε (χωρίς να είναι απαραίτητο). Η κατάσταση s5 είναι και αυτή μία τελική κατάσταση, μία κατάσταση αποδοχής, αφού το αυτόματο μας αναγνωρίζει κλασματικούς της μορφής πχ 123.232.
- Εάν μεταβούμε από την αρχική κατάσταση στη κατάσταση s2(δηλαδή ο πρώτος χαρακτήρας εισόδου είναι τελεία), ο επόμενος χαρακτήρας εισόδου πρέπει να είναι αριθμός για να μεταβούμε στην κατάσταση s4, όπου μπορούμε να δώσουμε όσους ακόμα αριθμούς επιθυμούμε (χωρίς να είναι απαραίτητο). Η κατάσταση s4 είναι και αυτή μία τελική κατάσταση, μία κατάσταση αποδοχής, αφού το αυτόματο μας αναγνωρίζει κλασματικούς αριθμούς της μορφής πχ 0.345.
- Οι καταστάσεις s1 και s2 ΔΕΝ είναι καταστάσεις αποδοχής διότι το αυτόματο δεν αναγνωρίζει ακέραιους αριθμούς ή μονάχα την τελεία.

### Αποτελέσματα εξόδου από την υλοποίηση του προγράμματος

The image shows a code editor window titled 'dfa.py' and a terminal window titled 'root@kali: ~'.

**Code Editor (dfa.py):**

```

transitions = {'s0':{'DIGIT':'s1', '.':'s2'},
               's1':{'DIGIT':'s1', '.':'s3'},
               's3':{'DIGIT':'s5'},
               's5':{'DIGIT':'s5'},
               's2':{'DIGIT':'s4'},
               's4':{'DIGIT':'s4'}}

#Προσθήκη 2!
accepts = {'s3':'FLOAT_TOKEN',
            's4':'FLOAT_TOKEN',
            's5':'FLOAT_TOKEN'}

def get_char(text,pos):
    """ Returns char (or char category) at position 'pos' of 'text'
    or None if out of bounds. """

    if pos<0 or pos>=len(text): return None

    c = text[pos]

    #Προσθήκη 3!
    # Με την προθήκη που έκανα:
    # Όταν έχει ως είσοδο 0...9 επιστρέφει Digit
    if c>='0' and c<='9':
        return 'DIGIT'
    return c

# Δεν επιτρέπεται η παρέμβαση στον κώδικα από αυτό το σημείο και κάτω!

```

**Terminal Output:**

```

root@kali:~# python3 dfa.py
Testing '12.456'
Result: {'token': 'FLOAT_TOKEN', 'lexeme': '12.456'}

Testing '6789.'
Result: {'token': 'FLOAT_TOKEN', 'lexeme': '6789.'}

Testing '.66998'
Result: {'token': 'FLOAT_TOKEN', 'lexeme': '.66998'}

Testing '1234'
Result: None

Testing '.'
Result: None

root@kali:~#

```