

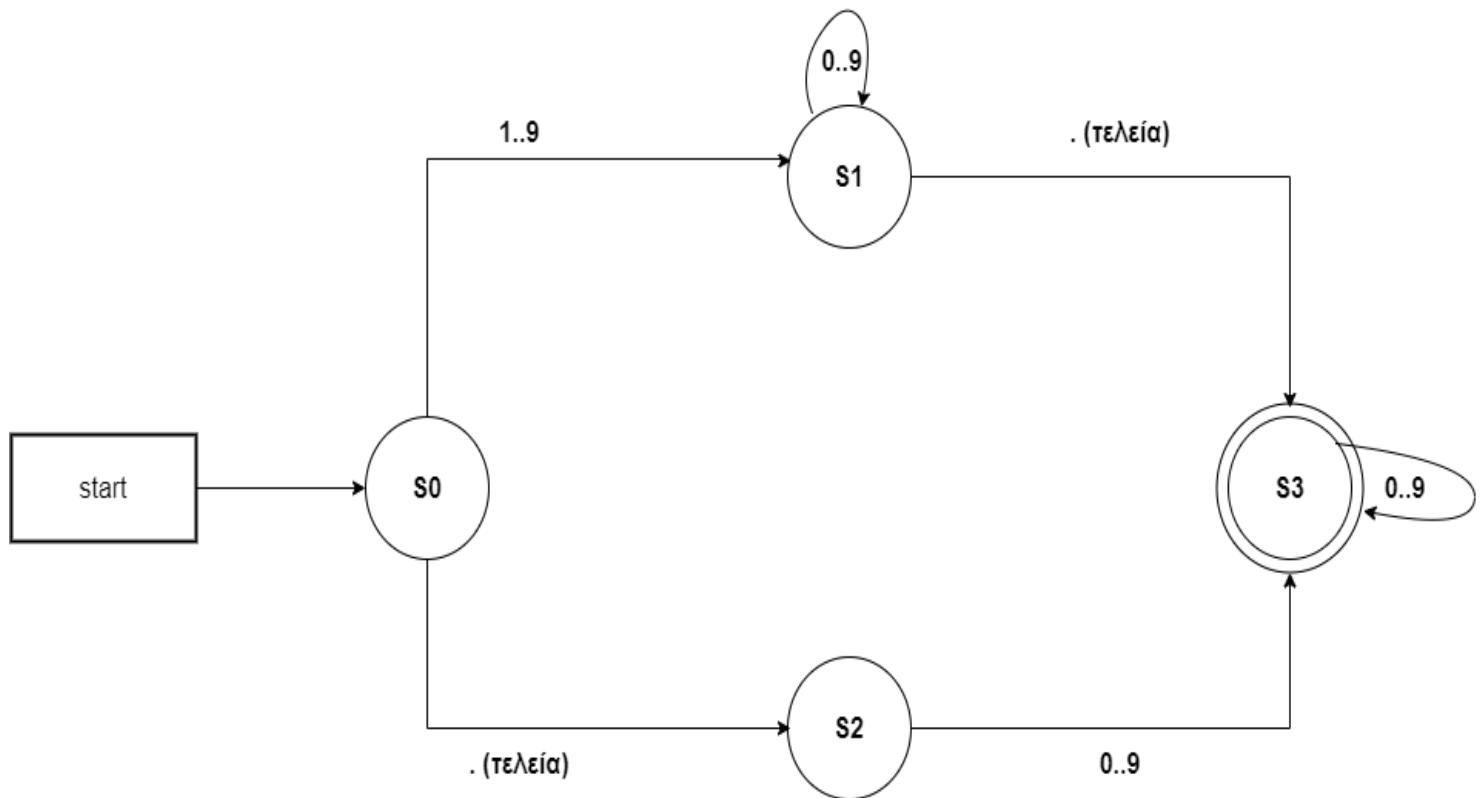
Μεταγλωτιστές 2020

Προγραμματιστική Εργασία #1

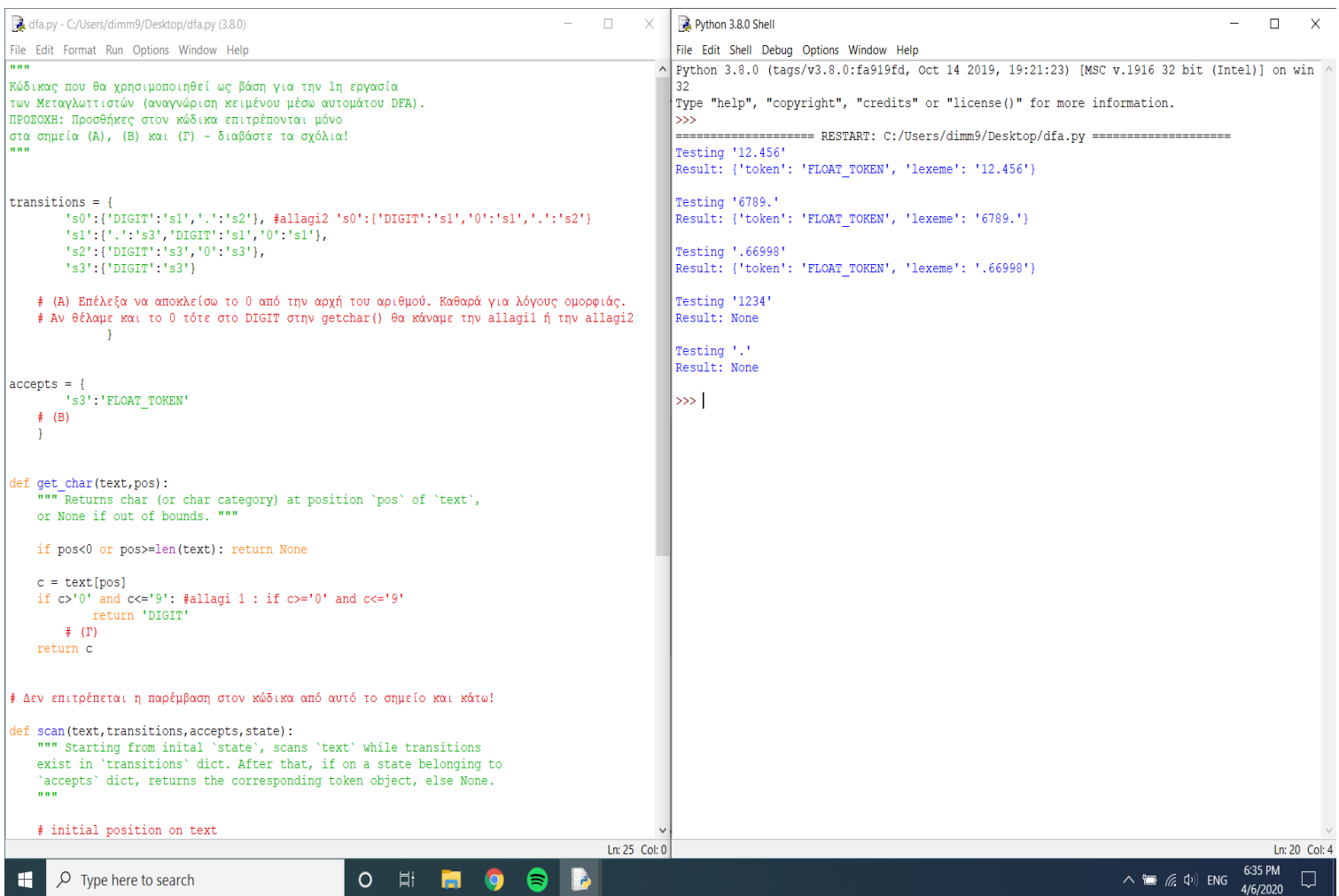
Ονοματεπώνυμο: Δημήτρης Μισαηλίδης

ΑΜ: Π2017066

Σχήμα αυτόματου: (βρίσκεται και σε png στο repository)



Αποτελέσματα εξόδου (screenshot) του προγράμματος (υπάρχει και στο [github repository](#))



```
dfc.py - C:/Users/dimm9/Desktop/dfc.py (3.8.0)
File Edit Format Run Options Window Help

"""
Κώδικας που θα χρησιμοποιηθεί ως βάση για την 1η εργασία
των Μεταγλωττιστών (αναγνώριση κειμένου μέσω αυτομάτου DFA).
ΠΡΟΣΟΧΗ: Προσθήκες στον κώδικα επιτρέπονται μόνο
στα σημεία (A), (B) και (Γ) - διαβάστε τα σχόλια!
"""

transitions = {
    's0':{'DIGIT':'s1','.': 's2'}, #allagi2 's0':{'DIGIT':'s1','0':'s1','.': 's2'}
    's1':{'.': 's3','DIGIT':'s1','0':'s1'},
    's2':{'DIGIT':'s3','0':'s3'},
    's3':{'DIGIT':'s3'}

    # (A) Επίλεξε να αποκλείσω το 0 από την αρχή του αριθμού. Καθαρά για λόγους ομορφιάς.
    # Αν θέλαμε και το 0 τότε στο DIGIT στην getch() θα κάναμε την allagi1 ή την allagi2
}

accepts = {
    's3': 'FLOAT_TOKEN'
    # (B)
}

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]
    if c>'0' and c<='9': #allagi 1 : if c>='0' and c<='9'
        return 'DIGIT'
    # (Γ)
    return c

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

def scan(text,transitions,accepts,state):
    """ Starting from initial 'state', scans 'text' while transitions
    exist in 'transitions' dict. After that, if on a state belonging to
    'accepts' dict, returns the corresponding token object, else None.
    """

    # initial position on text

Python 3.8.0 Shell
File Edit Shell Debug Options Window Help

Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/dimm9/Desktop/dfc.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

>>> |
```

Αναφορά σε πηγές που χρησιμοποίησα

- pdf και site του μαθήματος