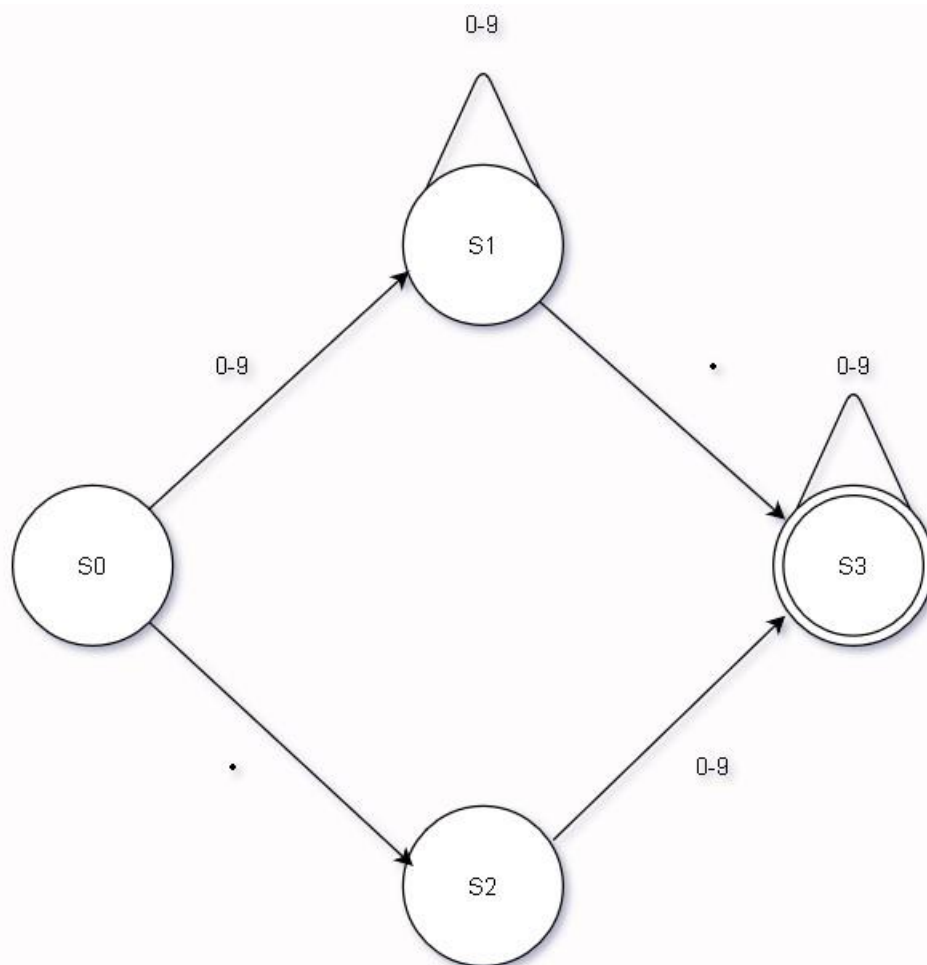


Μεταγλωττιστές 2020

Προγραμματιστική Εργασία #1

Ονοματεπώνυμο: ΠΑΠΑΠΕΤΡΟΥ ΑΘΑΝΑΣΗΣ

Α.Μ.: Π2017149



```

def get_char(text,pos):
    if pos<0 or pos>=len(text):
        return None
    c=text[pos]
    if c>='0' and c<='9':
        return 'DIGIT'
    return c

def scan(text,transitions,accepts,start):
    pos = 0
    state = start
    while True :
        c=get_char(text,pos)
        if state in transitions and c in transitions[state]:
            state = transitions[state][c]
            pos+=1
        else:
            if state in accepts:
                return {'token': accepts[state], 'lexeme':text[:pos]}
            return None

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

accepts = {'s3': 'FLOAT_TOKEN'}

text = input('Give')
m=scan(text,transitions,accepts,'s0')
print(m)

for test in ['12.456','6789.','66998','1234','.']:
    m = scan(test,transitions,accepts,'s0')
    print("Testing '{}' \nResult: {} \n".format(test,m))

```

```

Give
None
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

```

```
>>>
```