

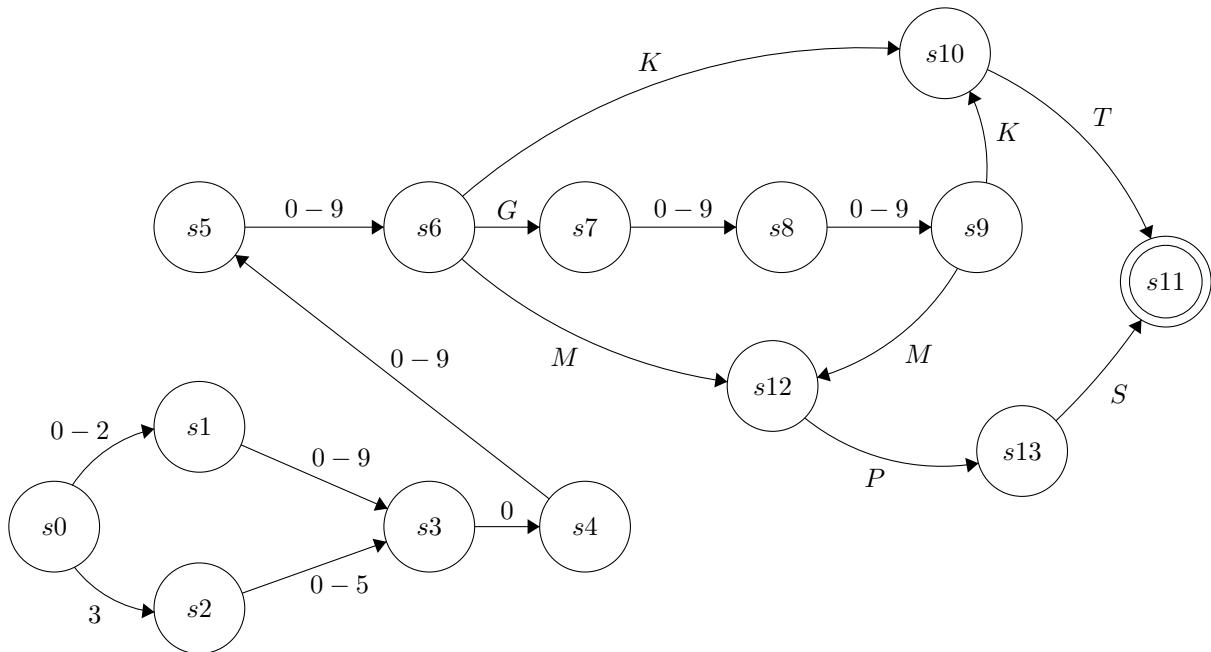
Μεταγλωτιστές 2019

Προγραμματιστική Εργασία 1

Γεώργιος Μαντέλλος
Π2016149

Μάρτιος 2019

- 1 Σχήμα ντετερμινιστικού αυτομάτου πεπερασμένων καταστάσεων (DFA)



- 2 Πίνακας Μεταβάσεων

	0	1	2	3	4	5	6	7	8	9	G	K	T	M	P	S
→s0	s1	s1	s1	s2												
s1	s3	s3	s3	s3	s3	s3	s3	s3	s3	s3						
s2	s3	s3	s3	s3	s3	s3										
s3	s4															
s4	s5	s5	s5	s5	s5	s5	s5	s5	s5	s5						
s5	s6	s6	s6	s6	s6	s6	s6	s6	s6	s6						
s6											s7	s10		s12		
s7	s8	s8	s8	s8	s8	s8	s8	s8	s8	s8						
s8	s9	s9	s9	s9	s9	s9	s9	s9	s9	s9						
s9												s10		s12		
s10													s11			
*s11																
s12															s13	
s13																s11

Όπως φαίνεται από το παραπάνω σχήμα, στον άξονα των x, βρίσκονται οι διαθέσιμες μεταβάσεις ενώ στον y οι καταστάσεις αποδοχής. Η αρχική μεταβίαση είναι η s0 και συμβολίζεται με δεξί βέλος και η τελική είναι η s11 που συμβολίζεται με *. Συνοπτικά, αναλύοντας την πρώτη μετάβαση, από την κατάσταση s0 με 0, 1 ή 2 πάμε στην κατάσταση s1 ενώ με 3 στην κατάσταση s2.

3 Συνοπτική περιγραφή του κώδικα

```

16  # **Σημείο #3**: Προαιρετικά, προσθέστε τις δικές σας ομαδοποιήσεις
17  if c == '0':
18      return 'DIGIT == 0'
19  if c == '1':
20      return 'DIGIT == 1'
21  if c == '2':
22      return 'DIGIT == 2'
23  if c == '3':
24      return 'DIGIT == 3'
25  if c == '4':
26      return 'DIGIT == 4'
27  if c == '5':
28      return 'DIGIT == 5'
29  if c >= '6' and c <= '9':
30      return 'DIGIT >= 6 and DIGIT <= 9'
31
32  if c == 'G':
33      return 'INPUT is G'
34
35  if c == 'K':
36      return 'INPUT is K'
37  if c == 'T':
38      return 'INPUT is T'
39
40  if c == 'M':
41      return 'INPUT is M'
42  if c == 'P':
43      return 'INPUT is P'
44  if c == 'S':
45      return 'INPUT is S'
46
47  return c    # anything else

```

Σχήμα 1: Μέρος κώδικα για την σύμπτυξη της εισόδου

Αρχικά, οι πιθανές αποδεκτές εισοδοι, είναι τα νούμερα από το 0 έως το 9 και τα γράμματα G, K, T, M, P και S. Έτσι, η σύμπτυξη αυτών γίνεται με τον παραπάνω τρόπο. Εάν το πρόγραμμα δεχτεί έναν από τους παραπάνω αριθμούς ή χαρακτήρες, τότε, επιστρέφει ένα string αναλόγως την είσοδο, ώστε να μπορέσουμε

να αναζητήσουμε στο λεξικό την επόμενη μετάβαση, αν αυτή υπάρχει. Η σύμπτυξη έγινε κατα αυτόν τον τρόπο, μιας και για τα νούμερα απο το 0 μέχρι και το 5, υπάρχουν διαφορετικές μεταβάσεις, συνεπώς μας ενδιαφέρει να γνωρίζουμε τον ακριβή αριθμό. Από το 6 και μέχρι το 9, οι μεταβάσεις είναι οι ίδιες για όλα τα νούμερα μέσα σε αυτό το υποσύνολο, συνεπώς δεν μας νοιάζει ποιος αριθμός είναι ακριβώς, αρκεί να είναι εντός ορίων. Το ίδιο ισχύει και για τους χαρακτήρες. Επειδή κάθε χαρακτήρας αντιστοιχεί σε διαφορετική είσοδο, επιθυμούμε να γνωρίζουμε ποιος χαρακτήρας είναι ακριβώς.

4 Αποτελέσματα εξόδου για έγκυρες και άκυρες μορφές εισόδου

```
[geocfu@icarus compilers1819a1]$ python3 scanner.py
give some input>12099KT
token: WIND TOKEN text: 12099KT
[geocfu@icarus compilers1819a1]$ python3 scanner.py
give some input>29099KT
token: WIND TOKEN text: 29099KT
[geocfu@icarus compilers1819a1]$ python3 scanner.py
give some input>35099KT
token: WIND TOKEN text: 35099KT
[geocfu@icarus compilers1819a1]$ python3 scanner.py
give some input>23085G99MPS
token: WIND TOKEN text: 23085G99MPS
[geocfu@icarus compilers1819a1]$ python3 scanner.py
give some input>35000G00KT
token: WIND TOKEN text: 35000G00KT
[geocfu@icarus compilers1819a1]$ python3 scanner.py
give some input>05000KT
token: WIND TOKEN text: 05000KT
```

Σχήμα 2: Παράδειγμα ορθής εισόδου

```
[geocfu@icarus compilers1819a1]$ python3 scanner.py
give some input>36099KT
unrecognized input at position 1 of 36099KT
[geocfu@icarus compilers1819a1]$ python3 scanner.py
give some input>35099G1MPS
unrecognized input at position 7 of 35099G1MPS
[geocfu@icarus compilers1819a1]$ python3 scanner.py
give some input>350G99KT
unrecognized input at position 3 of 350G99KT
[geocfu@icarus compilers1819a1]$ python3 scanner.py
give some input>10999G99MPS
unrecognized input at position 2 of 10999G99MPS
```

Σχήμα 3: Παράδειγμα λανθασμένης εισόδου