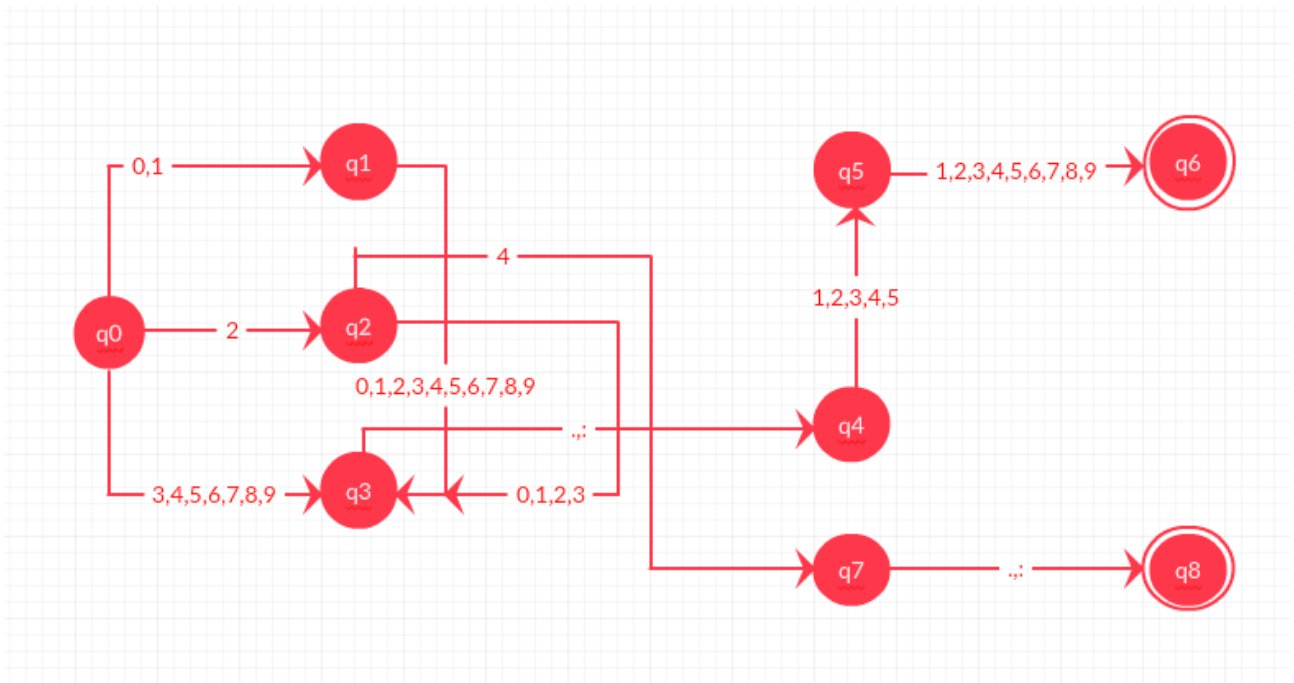


1) Το σχήμα DFA που αντιστοιχεί στον τρόπο λύσης μου, με αρχική κατάσταση την q0 και τελική την q6/q8:



2) Πίνακας Μεταβάσεων:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		0	1	2	3	4	5	6	7	8	9	.	:
2	q0	q1	q1	q2	q3	q3	q3	q3	q3	q3	q3	-	-
3	q1	q3	q3	q3	q3	q3	q3	q3	q3	q3	q3	-	-
4	q2	q3	q3	q3	q3	q7	-	-	-	-	-	-	-
5	q3	-	-	-	-	-	-	-	-	-	-	q4	q4
6	q4	-	q5	q5	q5	q5	q5	-	-	-	-	-	-
7	q5	-	q6	q6	q6	q6	q6	q6	q6	q6	q6	-	-
8	q6	-	-	-	-	-	-	-	-	-	-	-	-
9	q7	-	-	-	-	-	-	-	-	-	-	q8	q8
10	q8	-	-	-	-	-	-	-	-	-	-	-	-

στον παραπάνω πίνακα βλέπουμε σε ποια κατάσταση οδηγείται κάθε στιγμή το αυτόματο. (όπου υπάρχει “-” δεν γίνεται δεκτό).

3) Πρέπει να υλοποιήσουμε ένα DFA όπου θα αναγνωρίζει ενδείξεις ώρας. Με βάση το υπόδειγμα προγράμματος `rython3` που υλοποιήσαμε στο εργαστήριο, το πρόγραμμα ζητάει από τον χρήστη να δώσει μια τιμή/ώρα, αν η τιμή γίνει αποδεκτή επιστρέφει `TIME_TOKEN`, σε κάθε άλλη περίπτωση επιστρέφει `ERROR_TOKEN`, δηλαδή το input μας είναι λάθος.

4) Παραδείγματα κώδικα:

βάζοντας μια εσφαλμένη τιμή (πχ 12:7) παίρνουμε το παρακάτω αποτέλεσμα

```
zabzuki@zabzuki-Z97-D3H:~/Desktop/GitHub/python/compilers1718a1$ python3 scanner.py
give some input>12:7
unrecognized input at pos 4 of 12:7
```

βάζοντας δύο δεκτές τιμές (πχ 13:55, 17:21) παίρνουμε τα παρακάτω αποτελέσματα

```
zabzuki@zabzuki-Z97-D3H:~/Desktop/GitHub/python/compilers1718a1$ python3 scanner.py
give some input>13:55
token: TIME TOKEN string: 13:55
```

```
zabzuki@zabzuki-Z97-D3H:~/Desktop/GitHub/python/compilers1718a1$ python3 scanner.py
give some input>17:21
token: TIME TOKEN string: 17:21
```