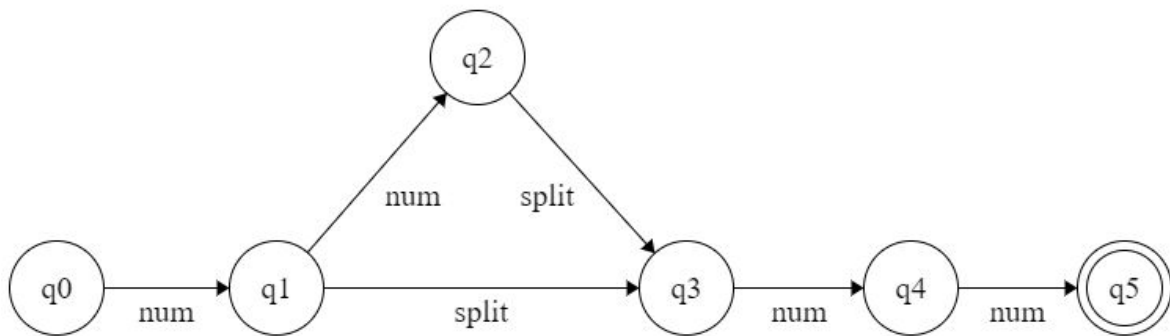


1: Σχήμα του ντετερμινιστικού αυτομάτου πεπερασμένων καταστάσεων (DFA) που αντιστοιχεί στη ζητούμενη προδιαγραφή :



Η πρώτη κατάσταση είναι η Q0 και η τελευταία η Q5

2:Πίνακας μεταβάσεων :

```
td = { 'q0':{ 'number':'q1' },  
       'q1':{ 'number':'q2','split':'q3' },  
       'q2':{ 'split':'q3' },  
       'q3':{ 'number':'q4' },  
       'q4':{ 'number':'q5' }  
}
```

Από την αρχική κατάσταση αν πάρουμε οποιονδήποτε αριθμό πάμε στην Q1

Από την Q1 παίρνουμε είτε αριθμό είτε διαχωριστικό. Αν πάρουμε αριθμό πάμε στην Q2 και αν πάρουμε διαχωριστικό πάμε στην Q3

Από την Q2 πρέπει να πάρουμε διαχωριστικό για να πάμε στην Q3

Από την Q3 πρέπει να πάρουμε αριθμό για να πάμε στην Q4

Από την Q4 πρέπει να πάρουμε αριθμό για να πάμε στην Q5

3: Περιγραφή κώδικα :

```
if state in transition_table and c in transition_table[state]:
    if state == 'q1' and c=='number':
        if text[pos-1]<='1' and text[pos-1] >='0':
            state = transition_table[state][c] # set new state
            pos += 1 # advance to next char
        elif text[pos-1] == '2':
            if text[pos]<='3' and text[pos]>='0':
                state = transition_table[state][c]
                pos+=1
            else: return 'ERROR_TOKEN',pos
    elif state=='q1' and c=='split':
        state = transition_table[state][c]
        pos+=1
    elif state == 'q3' and text[pos]>='0' and text[pos]<='5':
        state = transition_table[state][c]
        pos+=1
    elif state=='q0' or state=='q2' or state=='q4':
        state = transition_table[state][c]
        pos+=1
    else:
        return 'ERROR_TOKEN',pos
else: # no transition found

    # check if current state is accepting
    if state in accept_states:
        return accept_states[state],pos

    # current state is not accepting
    return 'ERROR_TOKEN',pos
```

Το πρώτο if και elif ελέγχουν την κατάσταση Q1. Αν έχει πάρει αριθμό που σημαίνει ότι υπάρχει και δεύτερος αριθμός ελέγχει αν ο πρώτος είναι από 0 έως 2. Αν ήταν από 0 έως 1 δέχεται οποιοδήποτε αριθμό, αν ήταν 2 δέχεται από 0 έως 3. Αλλιώς ελέγχει αν πήρε διαχωριστικό. Ο τρίτος έλεγχος γίνεται πάνω στην Q3 γιατί το πρώτο στοιχείο των λεπτών μπορεί να είναι από 0 έως 5 και αν είναι το δέχεται. Η επόμενη συνθήκη ελέγχει αν είναι οποιοδήποτε άλλο state από το Q1 ή Q3 που έχουν περιορισμούς και αν όχι δέχεται οποιοδήποτε ψηφίο. Τέλος αν δεν ικανοποιήθηκε καμία από τις προηγούμενες συνθήκες σημαίνει πως η είσοδος δεν είναι αποδεκτή και επιστρέφει : ERROR_TOKEN. Ο υπόλοιπος κώδικας είναι βασισμένος στο παράδειγμα.

3:Αποτελέσματα :

Αποτελέσματα εξόδου για άκυρες μορφές ώρας:

```
give some input>23:61  
unrecognized input at pos 4 of 23:61
```

```
give some input>52.35  
unrecognized input at pos 2 of 52.35
```

Αποτελέσματα για έγκυρες μορφές ώρας:

```
give some input>3:59  
token: TIME_TOKEN string: 3:59
```

```
give some input>12.32  
token: TIME_TOKEN string: 12.32  
-
```