Name : Aniket Iyer
PID : A17047762

CSE 105 Project:

1) My application takes in dates from the users who are born in November and December in the MM/DD format, and it accepts only those dates which correspond to the Sagittarius zodiac sign. I chose it so that it filters out all Sagittarian's from a given list of people.

2) The alphabet for my example is {0,1,2,3,4,5,6,7,8,9,/}

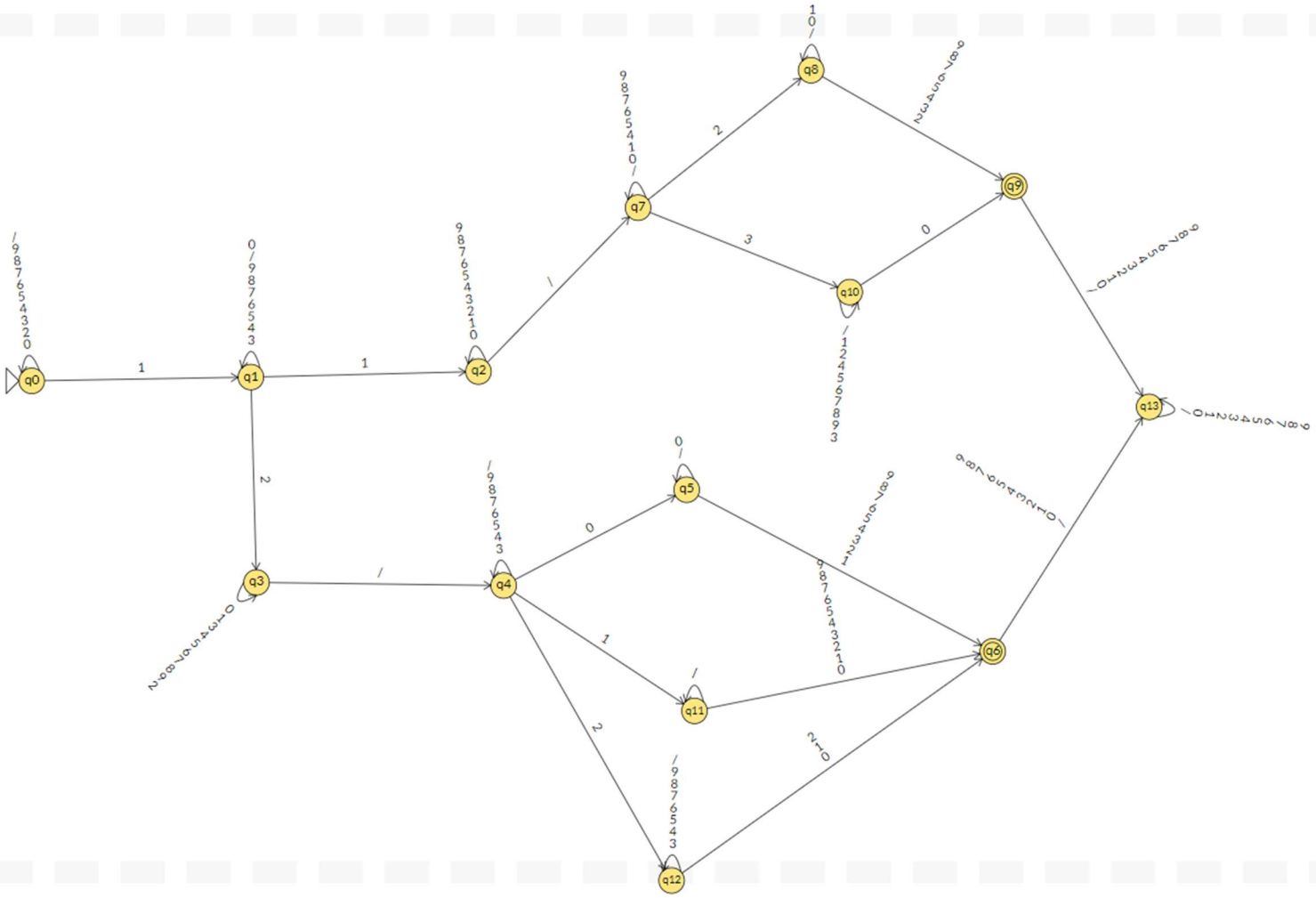
3) The set of strings over this alphabet that are important are strings which are of the MM/DD format, where the dates correspond to the Sagittarius zodiac sign. The dates should be between 21st of November and 22nd of December. (11/21 – 12/22).

4) One example of a string that is in the set is 11/27 as the month is 11 and the DD is between 27 and since, the date lies between the 21st of November and the 22nd of December, it is a valid example. Other examples are 12/06, 11/29, 12/15, and so on. Non-examples include strings such as 05/23, 06/98, 76/23. Although they are strings over the alphabet, they are not strings which we require. As we can see in the above examples, the MM and DD are not in the required range and hence, they are non-examples. I chose these example strings because we need them to be valid inputs for our applications. Since valid inputs for our applications are dates in the month of November and December that correspond to the Sagittarius zodiac sign, the language we are trying to define should have strings pertaining to the MM/DD format of dates between 21st of November and 22nd of December.

Name : Aniket Iyer
PID : A17047762

5) My language is regular as there exists a DFA D such that L(D) = the language we have defined above, i.e., the set of strings of the format MM/DD such that MM is either 11 or 12 and DD is anything between 01 and 31 depending on the month. If it is 11, then any date between 01 to 30 is a valid DD and if it is 12, then any date between 01 and 31 is a valid DD. (Since there are 31 days in December and only 30 days in November).

Pasted below is the DFA D :



Name : Aniket Iyer
PID : A17047762

Justifying the construction of the above DFA :

1) We start at our start state q0. As we know a valid input will start only with 1. Hence, we move to the next state q1 if we read a 1 otherwise we stay put and the computation keeps on looping on q0 until the end of the string and it is eventually rejected.
2) Once we move to q1, we can see that it could be either a 1 or a 2 as it should be either November or December. On reading a 1 or 2, we transition to different states, namely q2 and q3 respectively, as different sets of dates correspond to Sagittarius in November and December. While November dates in Sagittarius range from November 21st to November 30th, December dates in Sagittarius range from December 1st to December 22nd. That is precisely we have two different subdivisions in our DFA to handle the computation of November dates and December dates. If we don't read either a 1 or 2 in our second state, we keep on looping on q1 until the end of the string and eventually it gets rejected.
3) Now we either move to q2 or q3 and in both cases, it is necessary for us to read a /. This is because to be a valid string in our language it needs to have a / in its 3rd position. If we read a / we transition to either q7 (in November subdivision) or q4 (in December subdivision). Otherwise, we just stay put in q2 or q3 and eventually the computation loops in either of these states and the string eventually gets rejected.
4) Now, we are done processing the MM and / part of the string. We are just left with checking if the dates are valid, i.e., between 21 and 30 for the November subdivision and between 01 and 22 for the December subdivision.
5) We construct our November subdivision in such a way that we accept only the strings having 2 or 3 at the 4th position. All other strings end up looping on q7 and hence, those get rejected.

Name : Aniket Iyer
PID : A17047762

6) Once, we read a 2 or a 3, we need to make sure that the string succeeding it should be between 0 to 9 if we read a 2 or 0 if we read a 3. All the other strings end up looping on either q8 and q10 and thus, eventually get rejected.
7) Now, if we read the required strings after the 2 or 3, we transition to the accept state as we know that these strings are valid dates and lie in the range of the Sagittarius zodiac sign.
8) We also need to make sure that there could be strings which could have the valid date as a mere substring and hence, we have to make sure to send all the strings in the alphabet to a sink state from the accept state and hence we send all the strings from q9 to q13.
9) We could use a similar logic in the December subdivision to justify the construction of the DFA. We check for DD between 01 and 22 and only accept those strings which satisfy that condition. All other strings end up looping in either states q5, q11 or q12 and hence, eventually get rejected. We also make sure to send all strings from q6 to q13 (our sink state) for the same reason as mentioned in point 8.
10) Hence, I have justified the construction of the DFA D which recognizes the language L which is essentially the set of strings which correspond to the dates between 21st November and 22nd December in the valid MM/DD format!