Natural Language Mini Project 1 Report

A work made by: António Venâncio: 50%, 93689; Francisco Rodrigues: 50%, 93711; Group 47

Description of the adopted solution:

- For the solution of the transducer mm2mmm.fst we made two transitions from the initial states, by reading the first input digit. One for the months 1 to 9 and another for the ones from 10 to 12. From there it just reads the last input digit and matches it to the respective Arabic month.

- The transducers d2dd.fst and d2dddd.fst have a similar solution. In both the number of initial transitions is equal to the number of digits that we want to produce. For example, the d2dd.fst transducer, there is an initial transition for the case that the input is only one digit and there's also a transition for the case that there is two digits. The same process is applied in d2dddd.fst having 4 transitions.

- The copy.fst and skip.fst transducers are the same state machines. They have only one transition for each individual digit to the final state. In the case of the copy.fst the output will be the input digit, while in the skip.fst the output will be an epsilon transition.

- For a matter of simplicity date2year.fst skips every first 6 digits it encounters. Then it copies to the output tape the following 4 digits

- The leap.fst transducer was achieved through the mathematical method of calculating it, where the corresponding year must be fully divisible by 4, with the exception of century years (for example 1700, 1800 and 1900) which must be fully divisible by 400. However, leap.fst only considers years between 1901 and 2099, so the only century year is the year 2000 and it is divisible by 400. This makes possible for the transducer only have the need to check if the year is divisible by 4 to be functional, being this the though process given for the solution.

- The transducer R2A.fst can be divided into 4 'sections' of automata, each one tackling each digit, starting with the thousands, then hundreds, dozens and units. Each section takes into account the roman values which can occur, for example in the hundreds digit, the roman values that are applied to are the following: C, D and M. The following thought process is applied equally throughout all the sections, where the only difference being the symbols involved. All the 4 sections are connect together then to take in consideration digits valued at 0, for example MMII, which corresponds to 2001.

- The A2R.fst expected output is exactly the opposite of R2A.fst. So the solution is only making the operation fstinvert to obtain the desired output.

- By composing the transducers R2a.fst and d2dd.fst we can transform a Roman number to a two digit Arabic number, solving the problem for numbers below 10. The same approach goes for the year portion of the date, the only difference is instead of using the d2dd.fst we use d2dddd.fst for obtaining a four digit number. Finally we concatenate the first obtained transducers with itself and then we concatenate that transducer with the second developed fst obtaining the desired birthA2R.fst.

-For birthA2T we simply copy every single digit from the input tape except for the month portion, where we use fstconcat with the transducer mm2mmm.fst to obtain the written Arabic month format.

- By inverting both birthA2T.fst and birthR2A.fst (and by omission composing them) we can first obtain the date format Arabic-Number from the Arabic-text date and then turning the Arabic-Number format to the Roman format.

- When we compose birthR2A.fst with the fst date2year we can first obtain the Arabic-Number date format and with that output it results in the Arabic number year of that date. Finally, by composing the resulting fst with the leap.fst we can know if the Roman year is a leap year or not.

Contribution: For the making of this project, we both agreed in dividing it in equal parts. Meaning the transducers were divided according to their level of difficulty, resulting in a similar effort project.