

Natural Language - MP1

A fair amount of the theory surrounding this assignment has been discussed at length with various other students, which in turn has influenced some of the design decisions for some transducers. No code or solutions have been exchanged, although similar conclusions for some design choices might have been reached, based on these discussions. The academic discussion was found very valuable, and helped us better understand the theory and complete the assignment. In particular, a number of major design options were discussed and investigated with the members of group 74.

Two other sources were utilized in designing the transducer R2A. Two github repositories, containing transducers designed for translation between Roman and Arabic numbers were studied and used as inspiration for our developed transducer. (Source: https://github.com/akshay993/FST_RomanNumerals_HinduArabicNumerals , <https://github.com/pmorvalho/ist-LN>). As our solutions contained errors, we decided to implement this solution, in order to be able to complete the other subtasks. Therefore, credit is given to the user github user akshay993 (<https://github.com/akshay993>) , for the design of this transducer. We are aware that this transducer only reached to 100 and not 4000, and thus some later transducers of the R2A are only able to write years in the format of "YY" and not "YYYY".

We did not manage to find a suitable design for the leapyear transducer, and thus have not included this in the final deliverance. The assignments work was performed cooperatively within the group, sharing the workload as evenly as possible.

Discussion of the options

For all transducers, we made a sketch of the FSTs by hand before making the textfiles. The first two transducers in part one were pretty straight forward, and the d2dddd was made by just extending the d2dd transducer. Copy, skip and date2year was also very straightforward to make. Reaching part two of the project, we had some difficulties with understanding the logic. A2R was made by just inverting the R2A transducer made in part 1 using the built-in invert function in openFST. BirthR2A took the longest time, as we had some issues understanding what to do with the slash within the birth date.

Discussing with one of the assistants at the office hours we finally understood that the transducers had to be made by repeating the built-in compose and concat functions in openFST. This also helped understand the logic of the next two problems K) and L). The transducer BirthA2T was made by first using two different copy-functions where one of them was not made in part 1. The month was transferred to text using the mm2mmm function in part 1 a), and the rest was just copied in the same manner as the first part of the birth date. After being familiar with the invert function, we quickly realised that we just had to compose an inverted version of both birthA2T and birthR2A to create birthT2R.