

# Computational Linguistics

LIN/CSE 467/567 – Spring 2018

Homework 1

Due date: **Feb 16th**

## Instructions

Note: **Answers not submitted as specified below will not be graded.**

Upload to UBLearn a single file containing your answers to the two questions below. The text file must be named as 'hw1UBitNAME.txt', where UBitNAME is your UB email's name. For example, if your email were `nazgul@buffalo.edu` then your homework file must be named 'hw1nazgul.txt'. Don't forget to click the button after uploading the file, or it will not be submitted.

Students can work with others but must ultimately create their own individual answers. Any homework assignments that are sufficiently similar may be considered instances of plagiarism, and will be penalized and reported. See syllabus for more information about UB's official policy on ethical behavior.

## Exercises

Roman numerals were developed in the 8th century BC, went through several stages of development, and remained in common use into the 14th and 15th centuries. They are still used today in certain contexts such as in book prefaces, introductions and chapters, acts in plays, movements in music, movie sequels, names of monarchs and popes, timepieces, buildings (construction year), Olympic games, Super Bowl, and Wrestlemania events. The standard Roman numerals from 1 to 20 (including subtractive notation for 4 and 9) are as follows: I, II, III, IV, V, VI, VII, VIII, IX, X, XI, XII, XIII, XIV, XV, XVI, XVII, XVIII, XIX, XX.

1. Use the website at <https://www.draw.io/> to draw an FST that simultaneously achieves two things: (1) checks whether the input is a well-formed roman numeral between 1 and 20, and (2) outputs the corresponding Hindu-Arabic numeral. For example, the FST should accept strings like VIII, IX, and XIV, and output 8, 9 and 14, respectively, while rejecting strings like VIIII, and VX. The (ideal) solution involves an FST with fewer than 10 states. When you are done drawing your FST, please open the *File* tab, select *Embed as*, and click *HTML*. Then, click the 'Create' button. You can click 'Preview' in order to make sure your FST is rendered correctly in HTML. Finally, copy the highlighted HTML text into your answer file 'h1UBitNAME.txt'.

[3 points]

2. Implement your FST in SWI Prolog, so that a query like `q0([x,v,i,i],N)` yields `N = [1,7]` and `q0(N,[8])` yields `N = [v,i,i,i]` but *any* query involving an ill-formed Roman numeral like `q0([v,i,v],N)` yields 'false'. Simply paste your code to your answer file 'h1UBitNAME.txt' after the HTML code for the FST. Your answer should consist of the Prolog code implementing your answer to exercise 1 above.

[3 points]