Universität Konstanz

Introduction to Computation for the Social Sciences

Assignment 10

Prof. Dr. Karsten Donnay, Marius Giebenhain, Stefan Scholz
Winter Term 2019 / 2020

Please solve the exercises below and commit your solutions to our GitHub Classroom until Jan, 28th midnight. Submit all your code in one executable file (*py* / *ipynb*) and your text in one text file (*txt* / *md* / *pdf*). You can score up to 10 points in this assignment. You will get individual feedback in your repository.

Exercise 1: Efficient Time Iteration in Observational Data (5 Points)

We have discussed a number of sorting and search algorithms in great detail. This exercise now explores a very practical application: efficient, systematic comparisons of time-stamped data. We explore the efficiency gains of different approaches in several steps. Please use the empirical conflict event data we used in *Assignment 6* (*ACLED_South-Sudan_2017.csv*) to test your algorithm. *Note: Use event_date as the timestamp.*

a) Begin by writing a simple function `slidingwindow` that tests for each pair of events (i.e. a given timestamp) whether they lie within X days of each other (before and after). The function should for each event return a list of those events that do.

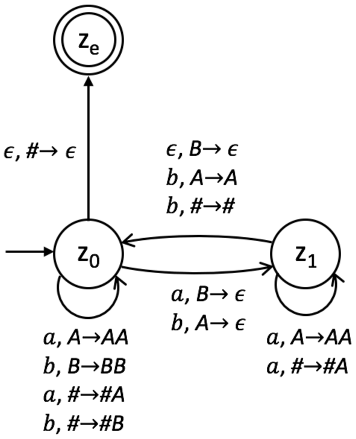
b) Now use we optimize the performance of this function by first sorting the data by timestamp and then doing the comparison by iterating through the sorted index of events (forward and backward). In doing so, we use the minimal number of steps to decide if an event before or after a given event in the index is within our X-day window.

c) Systematically compare the performance of the two variants when sorting our empirical conflict event data using the `time` module in Python.

Exercise 2: Push-down Automata and Turing Machines (5 Points)

In the lecture, we discussed push-down automata as representations of context-free languages and Turing machines as representations of context-sensitive and recursively enumerable languages. Here, we take a closer look at both.

a) First, consider the following push-down automaton with three states $M = (\{z_0, z_1, z_2\}, \{a, b\}, \{A, B\}, \delta, z_0, \#)$ where transitions are given by:

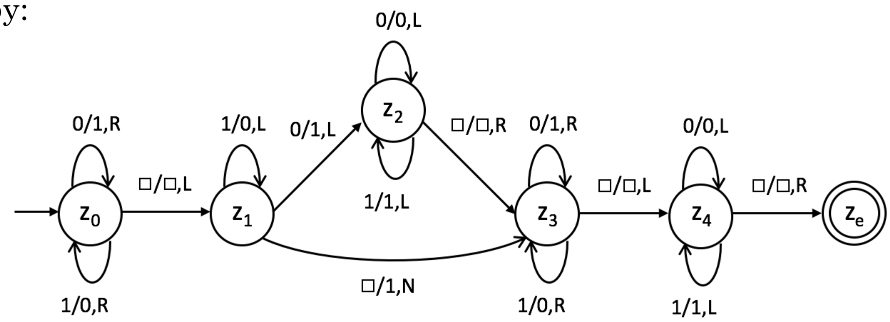


i. Write down the exact sequence of state changes of the automaton if you enter the word *ababbbabb*. Use triples of the following form: `(state, input, stack status) → (state, input, stack status)`

ii. Which context free language does this push-down automaton recognize?

iii. Is the automaton deterministic or nondeterministic?

b) Now consider the following deterministic Turing Machine with six states $TM = (\{z_0, z_1, z_2, z_3, z_4, z_e\}, \{0,1\}, \{0,1,\square\}, \delta, z_0, \square, \{z_e\})$ where transitions are given by:



iv. Write down the exact sequence of state changes of the Turing Machine if you enter the word 1010010. Use the $\alpha z \beta$ notation introduced in the lecture.

v. Which operation does the Turing machine perform?

vi. How is this related to the Turing machine we saw in the lecture?