

Ling 570 HW2

Olga Whelan

MY ASSIGNMENT IS LATE BECAUSE I FORGOT THAT SUBMISSION DATE WAS YESTERDAY :(((I GOT CONFUSED WITH TIME ZONES - I AM USUALLY NEVER LATE

Q1 (2)

- yes, the results are the same
- `carmel -k 1 fsa7 wfst1` returns the best path for the wfst1-transducer

`-sli wfst1` prepends the input to wfst1, is interpreted as a separate sequence of symbols and translated into a transducer

(Q3 notes)

`fsa_acceptor.sh` calls `test.sh` and `cleanout.py`. It also generates temporary files `fn1` and `fn2` at every run.

command line: `./fsa_acceptor.sh q2/fsa1 /dropbox/13-14/570/hw2/examples/ex > q3/ex.fsa1`

Q4

I used a dictionary of lists *fsmStates* to store the input FSA (created in function *make_fsa*). States are the keys of the dictionary; the value of each key is a list of tuples. Each tuple contains the input symbol and the state it transits to.

My code only handles DFAs. I spent too much time making it work for the case of DFA and did not have enough time left to make the NFA aspect work. If the FSA is non-deterministic, all strings will be rejected. This means that files `ex.fsa1` and `ex.fsa2` in `q4` are incorrect.

The work on NFA treatment is supplied in *nfa_acceptor.py*. It does not work correctly. The idea was to follow all possible branches and to check whether any of them end in a final state. I attempted to implement the algorithm described in J&M 2.2.5 and I think I am very close. At the point of submission I was stuck with addressing my data structures.

This file can be run with `python2.7 nfaacceptor.py q2/fsa1 /dropbox/13-14/570/hw2/examples/ex > q4/ex.fsa1`

I apologize for the delay and for the messy code.