LING 570: Hw5
Author: Dhwani Serai (dserai)
Date: 3rd November 2022

As usual, the example files are stored under **570/hw5/examples/.**

Q1 (25 points): Are the following true or false?

- If true, please provide a proof:
  - You can assume that regular languages and FSAs are equivalent, so are regular relations and FSTs.
  - For instance, if you want to prove a relation R is regular, you just need to show that you can build an FST that transduces x into y for every pair (x, y) in R.
- If false, provide a counterexample:
  - for instance, if (a) is false, you need to find a regular relation R and shows the corresponding L is not regular.

(a) Let R be a relation. Let L = {x | there exists y, such that (x, y) $\in$ R}. If R is a regular relation, then L is a regular language.

Ans: True

Proof: We know that since R is a regular relation there can be a FST A which can be constructed for Relation R where x is an input and y is an output. (Which is also regular)

$A = <\Phi, \Sigma1, \Sigma2, \delta, q0, \gamma>$

Similarly we can create an FSA B from FST A by removing the output symbols from A.

$B = <\Phi, \Sigma1, \delta, q0, F>$

Where the set of states will be the same for FSA and FST

- Set of input symbols also remain the same
- Transition function also remains the same
- Just removing the output symbols which transduce every x into y and removing the output transition function
- Final state of the FSA will be the same as the Final state of FST

Every set of strings which were getting accepted in Relation R will also be accepted by the FSA (because of reasons mentioned above) so the FSA will also be regular

Since FSA B is regular its equivalent language L will also be a regular language.

Hence, proved

(b) Let L1 and L2 be two languages. Let R be the cross product of L1 and L2. That is, R = {(x, y) | x ∊ L1 and y ∊ L2}. If L1 and L2 are regular languages, then R is a regular relation.

Ans: True
Proof: if L1 and L2 are regular languages then let A and B be their corresponding FSAs where
A = <Φ1, Σ1, δ1, q10 ,F1> and B = <Φ2, Σ2, δ2, q20 ,F2> (A,B are regular since FSA and their corresponding Language are equivalent)
Then according to the definition of Relation R the FST will be
Q = <Φ1, Σ1, Σ2, δ, q0, γ> where

- Φ1 : is the set of states corresponding to Language L1 i.e. FSA A
- δ : Φ1 x Σ1 (transition function) (which is also equivalent to δ1)
- γ: Φ1 -> Σ2 (output function)

Since the transition function for FSA A (L1) and FST Q is the same because the set of states are the same just there is an output symbol corresponding to the input.
So, every string in Language L1 will be accepted by FST Q because of the transition function; the output will depend on the output function γ which is dependent on the regular Language(L2)

But since L1 is a regular language and every string in L1 will be accepted by FST Q we can say that the Relation R corresponding to FST Q is also a regular relation

Hence proved

Q2 Q3 and Q4 notes:

- Initially for expand_fst I had taken an approach similar to the one in hw4 but here I have also added the input symbols in the output because of the output format which was required in q3 and q4
- In expand_fst I have also added additional symbols like # and * to distinguish between the part of speech tags in the morph_rules and the input_symbols
- Later for q4 formatting I have used two additional python scripts split_str.py and format_op.py split_str is to split the input word (for eg: “cuts”) into “c” “u” “t” “s” to make it in a Carmel format
- format_op.py takes the output from expand_fst.py and removes the additional symbols like # and * and turns the output into the desired format.
- Both of these additional python scripts are being called from morph_acceptor.sh