1 Nested-word automata (8 points)

Show that NWAs (over finite words) are closed under concatenation.

Careful: The first nested-word might have unmatched calls and similarly the second NWA might have unmatched returns. Make sure they match properly.

Important: Specify in your construction what are the input and output NWAs. I don’t require a proof of correctness, but clearly state the invariants maintained by your construction (e.g., after reading the nested word $a_1...$ the NWA $C$ is in state $q$ with stack $p_1...p_n$ iff $...$).

2 Probabilistic model checking (6 points)

Exercise 10.3 on page 899 of Principles of model checking.

3 Symbolic transducers (7 points)

We refer to the paper “Symbolic Finite State Transducers: algorithms and applications” by Veanes et al.

Provide the code. Use Bek (http://rise4fun.com/bek) to model the following two functions:

- $deletealpha$, that deletes all the alphabetic characters in $[a - z]$ from a string. E.g., $deletealpha(abil33[]) = 133[$.
- $stutter$, that duplicates every charter of a string. E.g., $stutter(abcabc) = aabbeccabbec$.

Now use Bek to prove that

- $deletealpha$ is idempotent: running it twice in a row is equivalent to running it once;
• deletealpha and stutter commute.
• the output of stutter always has even length.

4  \( L^* \) (6 points)

We briefly mentioned in class that the following problem is NP-Complete. Given two DFAs \( A \) and \( B \), such that \( L(A) \cap L(B) = \emptyset \), find the minimal DFA \( C \) that separates them. That is

• \( L(A) \subseteq L(C) \);
• \( L(B) \cap L(C) = \emptyset \);
• every \( C' \) with the two properties above has at least \( n \) states, where \( n \) is the number of states in \( C \).

On the other hand we saw that \( L^* \) can learn a DFA accepting a given regular language \( R \) in polynomial time. Let’s say we adapt \( L^* \) in the following way:

• Membership queries: given a string \( w \), mark it as positive example if \( w \in L(A) \) and negative if \( w \in L(B) \)
• Equivalence queries: is the conjectured automaton \( M \) such that \( L(A) \subseteq L(M) \) and \( L(B) \cap L(M) = \emptyset \)? If not provide a counterexample.

Why doesn’t this algorithm work?

5  Reactive Synthesis: 6 points

Solve the following reactive synthesis problem. A depiction of the situation is given in Figure 1.

You have to synthesize a controller for a traffic light that has the following inputs, outputs, and requirements.

Input signals

• C: car is waiting on farm road
• P: pedestrians want to cross the highway
• Eh: emergency vehicle on highway
• Ef: emergency vehicle on farm road
Figure 1: Road Intersection

Outputs signals

- h: highway light is green
- f: farm road light is green

Requirements

- Lights should not be both green
- Lights should both turn green every now and then
- If pedestrians approach, farm light should go green (i.e. on input, the system should immediately output f).

You have to:

- Formalize the requirements in LTL;
- Is the system realizable? If so provide a controller otherwise provide a winning strategy for the environment.