Mid-Semester Thesis Presentation I

Improving known bounds on size of 2DFAs solving particular interesting problems

Advisor: Christos Kapoutsis
Student: Qasim Nadeem

Agenda

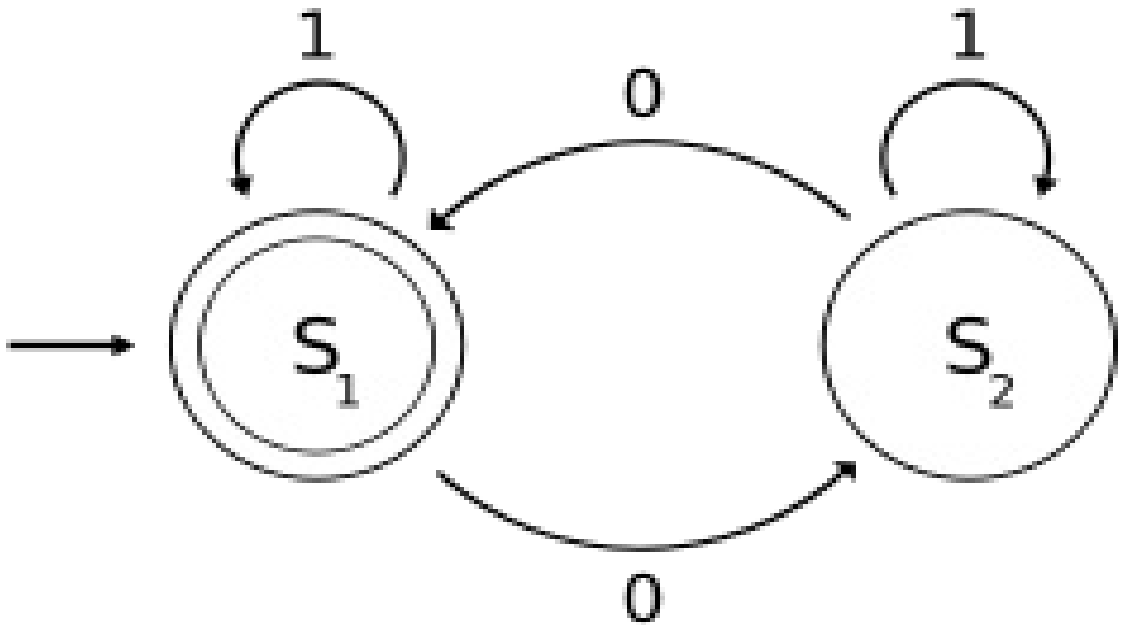
1. Description of the general area
2. Our thesis focus and progress
3. Summary of prior work
4. Future outline

Deterministic Finite Automaton

Recall that

M = (S,Σ,δ,q0,F)

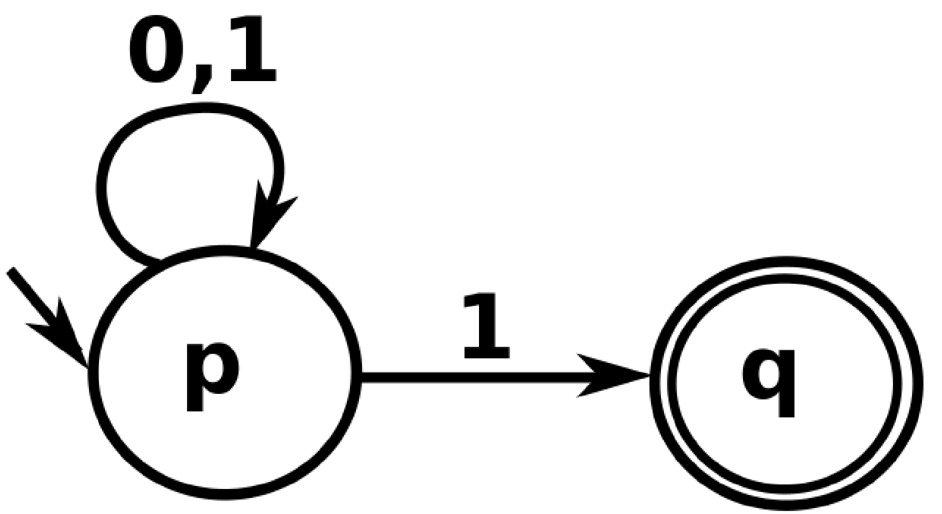
δ: S×(Σ∪{⊢,⊣}) ⇀ S



Non-Deterministic FA

The same EXCEPT that in any transition can move to any of a SET of states

δ: $S \times (\Sigma \cup \{\vdash, \dashv\}) \rightharpoonup P(S)$



- Of course these were *one*-way finite automata..
- 1DFA and 1NFA respectively
- READ head could move both ways too..
- Thus, giving us 2DFA and 2NFA

Does a 2-way head help?!

- Not in terms of computability..
- Kleene's theorem + conversions across the 4 types..
- All of them only recognize regular languages..
- But what about Complexity?

Finite Automata (FA) complexity

- The resource to consider now is size (as in number of states)
- We now care about how # of states required grow for language families
- It's normally easiest to talk about state complexity for 1DFAs because they need to carry forward all of the critical information, otherwise they'll be fooled
- 2DFAs much harder to reason about because they need not carry all critical information, can move back/forth when needed

One-Way Liveness (OWL)

- EXAMPLE: Sakoda/Sipser (1978) is an important early paper in the area. They gave the names 1,2 N/D, gave concept of homomorphic reductions and provided/proved complete problems for 1N/2N etc.
- A 1DFA needs exactly 2^h – 1 states to solve OWL_h (easy to prove)
- A 1NFA can solve OWL_h with h states (can not do with less?****)
- What about 2DFA? Key open problem! Because OWL happens to be the hardest problem in this 1N → 2D conversion! (OWL in 1N and all probs in 1N h-reduce to OWL)
- Sipser/Sakoda (1978) gave this idea of homo-morphic reductions under which 1,2 N/D are closed. this can be thought of as the poly-time reductions we use in NP.
- Meaning that if OWL_h can be solved with p(h) states on a 2DFA then any NFA can be converted to an at most polynomially larger 2DFA..
- Thus, above is actually asking whether 1N \in 2D?

Two-Way Liveness (TWL)

- Also given by SS (78). Same idea as OWL, except that now you can have backward arrows too.
- Clearly, 2NFA can solve it with h states again.. but unknown under 2DFA once more.
- The above conversion asks whether 2D=2N (?), since TWL complete for 2N.
- Many open problems surrounding these two big problems namely whether 1N\in2D, and whether 2N=2D?

Our Focus I

- I've read SS (1978), Sieferas' manuscript (1973), Kapoutsis (2009 and 2011). Readin Sipser currently (Sweeping automata - 1980)
- Our focus right now is improving bounds of OWL for constant sized inputs, with or without restricted alphabet.
- We already know a bunch of stuff. Already talked about how a 1DFA needs exactly 2^h-1 states and 1N needs exactly h.

Our Focus II

- We know some bounds on OWL_h_2, OWL_h_3 already.
- We know simple O(h) and O(h^2) 2DFAs for these two. You can extend that idea to O(h^n) 2DFA for OWL_h_n.
- However, we want to bring the upper and lower bound as close as possible.
- **OPEN 1**: Exact size of smallest 2DFA for OWL_h_2. So far we have 2h.
- OWL_h_2 has an exact LB of h/2 (using a visited state pair argument) and UB of 2h.
- **OPEN 2**: Exact size of smallest 2DFA for OWL_h_3.
- LB for OWL_h_3 is O(h^2/lgh) (using visited state pair argument again)
- We know though of a O(h^2/lgh)-state algorithm for OWL_3^h that Christos found. Exact bounds not completely clear yet. We're still working on OWL_h_2.
- Interesting thing is that the exact LB arguments above don't even use the entire alphabet. Only a restricted alphabet.
- In these past two weeks, we got an UB of h for the restricted alphabet. Found by brute forcing automata for h=2, and then saw that it can be generalized. Trying to bring the LB up to h now.

Future outline

- Bring the LB upto h (which is what we feel should be the case)
- Work on OPEN 1 and OPEN 2. Can start thinking about OWL_h_4 and so on..
- Can also start looking at TWL_h_2, TWL_h_3 etc
- Read and understand others' advanced work in the field example the Sipser paper (1980) on sweeping automata doing now.

Any questions?