Minimal automata

1 Introduction

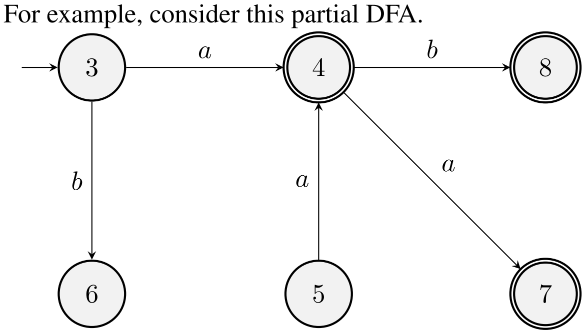
When we convert a regex into a DFA or partial DFA, we sometimes obtain an automaton with more states than are needed. Happily, this can be cut down in a systematic way to obtain a *minimal* automaton.

2 Reachable, hopeful and equivalent states

In an automaton, a state s is *reachable* when there is a path from an initial state to s. It is *hopeful* when there is a path from s to an accepting state.

Two states s and t are *equivalent* when the language of s (the set of words that are acceptable starting from s) is the same as the language of t.

For example, consider this partial DFA.



- Is state 5 reachable?
- Is state 6 reachable?
- Is state 5 hopeful?
- Is state 6 hopeful?
- Are states 4 and 7 equivalent?
- Are states 7 and 8 equivalent?

3 Minimal automata

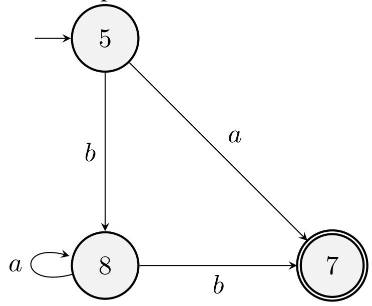
A partial DFA is *minimal* when it satisfies three conditions:

- Every state is reachable.
- Every state is hopeful.[1]
- Every pair of equivalent states are equal.

[1] When studying total DFAs, this condition is not imposed.

The above example satisfies none of these conditions!

Here is a partial DFA that's minimal. (Call it Automaton A.)

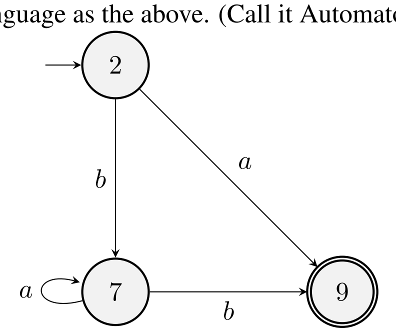


Proof of minimality:

- 5 is reachable via ε, and hopeful via a.
- 8 is reachable via b, and hopeful via b.
- 7 is reachable via a, and hopeful via ε.
- 7 and 5 are inequivalent, since 7 accepts ε and 5 doesn't.
- 7 and 8 are inequivalent, since 7 accepts ε and 8 doesn't.
- 5 and 8 are inequivalent, since 8 accepts b and 5 doesn't.

4 Uniqueness

There can be more than one minimal PDFA for the same language. For example, here's a minimal PDFA with the same language as the above. (Call it Automaton B.)



There's exactly one isomorphism between A and B. What is it?

Generally, if M and N are minimal PDFAs for the same language, there's a unique isomorphism between M and N.

5 How to minimize an automaton

Remember that when we're in state s and read the letter a, the resulting state is called $\delta(s, a)$.

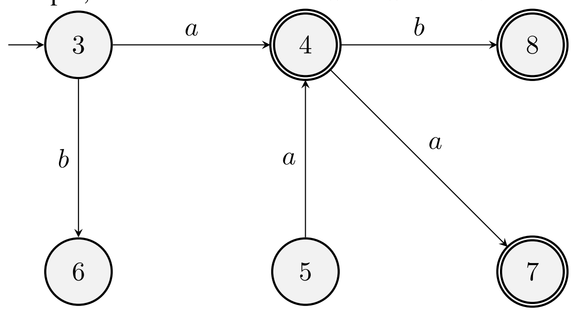
Let's note that if s and t are equivalent states, then

- s and t are either both accepting or both rejecting;
- the states $\delta(s, a)$ and $\delta(t, a)$ are also equivalent.

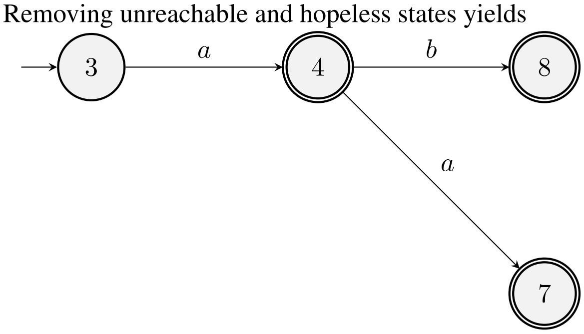
To minimize a partial DFA:

- Remove all the unreachable states.
- Remove all the hopeless states.
- Identify each set of equivalent states.

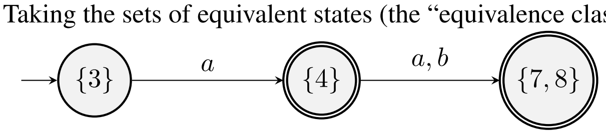
For example, here's the automaton we saw before:



Removing unreachable and hopeless states yields



Taking the sets of equivalent states (the "equivalence classes") yields the following:



To finish off, just to be sure you haven't missed anything, prove that this is minimal.