Finite automata theory and formal languages (DIT321, TMV027)

Nils Anders Danielsson

2019-02-18

Today

- Various algorithms.
- Equivalence of states.

Some old algorithms

Some algorithms we have already seen

- (ε-)NFA to DFA. (Can be slow.)
- DFA to (ε-)NFA. (Fast.)
- FA to RE. (Can be slow.)
- RE to ε-NFA. (Fast.)

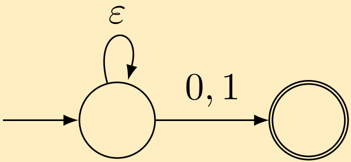
Empty?

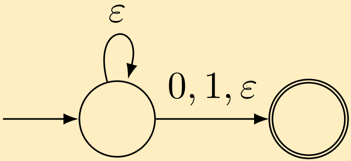
Is the language empty?

- For an FA: If there is no path from the start state to an accepting state.
- For a regular expression:

$$
\begin{aligned}
&empty \in RE(\Sigma) \to Bool\\
&empty(\emptyset) &&= \text{true}\\
&empty(\varepsilon) &&= \text{false}\\
&empty(a) &&= \text{false}\\
&empty(e_1 e_2) &&= empty(e_1) \vee empty(e_2)\\
&empty(e_1 + e_2) &&= empty(e_1) \wedge empty(e_2)\\
&empty(e^*) &&= \text{false}
\end{aligned}
$$

Which of the following regular expressions/ε-NFAs over $\{ 0, 1 \}$ represent the empty language?

1. $\emptyset + \varepsilon$
2. $\emptyset + \emptyset^*$
3. $\emptyset^+$
4. $(\emptyset 01 + 10(\emptyset + \varepsilon\emptyset))^+$
5. 

6. 

Member?

Is the string a member of the language?

- For a DFA: Move from state to state, check if the last state is accepting.
- For an NFA or ε-NFA:
  - Keep track of a set of states.
  - Or convert to a DFA. (This could be much less efficient.)
- For a regular expression:
  - Convert to an ε-NFA.
  - Or use Brzozowski derivatives. (At least in some cases less efficient.)

Equivalence of states

Equivalence of states

For a DFA $(Q, \Sigma, \delta, q_0, F)$:

- Two states $p, r \in Q$ are *equivalent* $(p \sim r)$ if

$$\forall w \in \Sigma^*.\ \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(r, w) \in F.$$

- Two states that are not equivalent are *distinguishable*.

Which of the following properties does the $\sim$ relation always satisfy?

1. It is reflexive.
2. It is symmetric.
3. It is antisymmetric.
4. It is transitive.

Equivalence of states

To find out which states are equivalent:

- Create a matrix where rows and columns are labelled by states.
- Mark every accepting state as distinguishable from every non-accepting state.
- Repeat until no further changes are possible:
  - Mark two states $p, q \in Q$ as distinguishable if there is some $a \in \Sigma$ for which $\delta(p, a)$ and $\delta(q, a)$ have already been marked as distinguishable.
- States that have not been marked as distinguishable are equivalent.

Equivalence of states

Note:

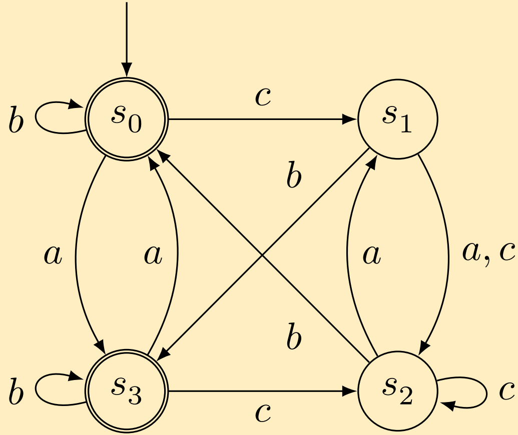
- The $\sim$ relation is reflexive, so one can skip the diagonal.
- The $\sim$ relation is symmetric, so one can skip, say, the elements below the diagonal.

(Assuming that row and column labels are ordered in the same way.)

Equivalence of states

- The $\sim$ relation is an equivalence relation.
- The equivalence classes partition the set of states.

How many equivalence classes does the $\sim$ relation for the following DFA have?



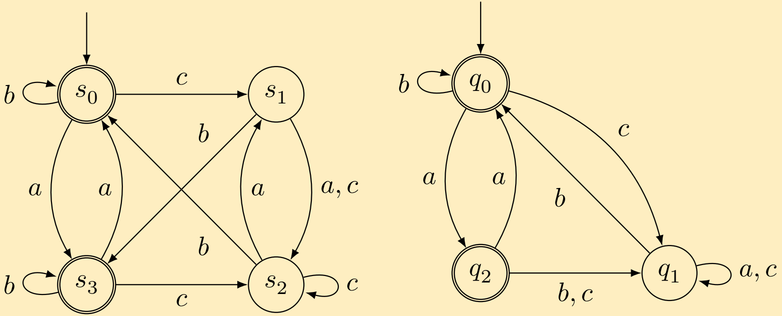
Equality of languages

Equality of languages

To find out if two languages, represented by the DFAs $(Q_1, \Sigma, \delta_1, q_{01}, F_1)$ and $(Q_2, \Sigma, \delta_2, q_{02}, F_2)$ with $Q_1 \cap Q_2 = \emptyset$, are equal:

- Create the DFA $(Q_1 \cup Q_2, \Sigma, \delta_1 \cup \delta_2, q_{01}, F_1 \cup F_2)$.
- The languages are equal iff $q_{01} \sim q_{02}$.

Are the languages over $\{ a, b, c \}$ denoted by the following DFAs equal?



Equality of languages

Note:

- One can skip entries for which the row label and column label belong to the same DFA.

Minimisation

Minimisation

Given a DFA $A = (Q, \Sigma, \delta, q_0, F)$ one can construct a minimal (in terms of the number of states) DFA that represents the same language.

Minimisation

1. Remove non-accessible states:

$$A' = (Acc(q_0), \Sigma, \delta', q_0, F \cap Acc(q_0))$$
$$\delta'(q, a) = \delta(q, a)$$

2. Replace the set of states with equivalence classes of equivalent states:

$$A'' = (Acc(q_0)/\sim, \Sigma, \delta'', [q_0], F'')$$
$$\delta''([q], a) = [\delta(q, a)]$$
$$F'' = \{\ [q] \mid q \in F \cap Acc(q_0)\ \}$$

Exercise: Check that A'' is a well-formed DFA. Prove that it accepts the same language as A.

Minimisation

Why is the constructed DFA minimal?

- Take any DFA $B = (Q_B, \Sigma, \delta_B, q_B, F_B)$ that represents the same language.
- Combine A'' and B like in the language equality checking algorithm (renaming states if necessary).
- We have $[q_0] \sim q_B$.
- Hence every accessible state $\widehat{\delta''}([q_0], w) = \left[\hat{\delta}(q_0, w)\right]$ of A'' (and thus every state of A'') is equivalent to a state of B, $\widehat{\delta_B}(q_B, w)$.

Minimisation

Consider the following function:

$$f \in Acc(q_0)/{\sim} \to Q_B/{\sim}$$
$$f\Big(\big[\hat{\delta}(q_0, w)\big]\Big) = \big[\widehat{\delta_B}(q_B, w)\big]$$

This is a proper definition, because
if $\hat{\delta}(q_0, u) \sim \hat{\delta}(q_0, v)$ then $\widehat{\delta_B}(q_B, u) \sim \widehat{\delta_B}(q_B, v)$.

Minimisation

- The function f is injective:

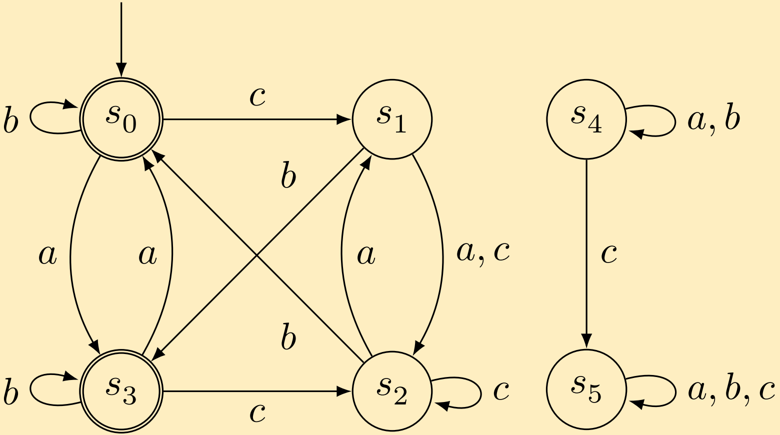
$$
\begin{aligned}
& f\left(\left[\hat{\delta}(q_0, u)\right]\right) = f\left(\left[\hat{\delta}(q_0, v)\right]\right) \Leftrightarrow \\
& \left[\widehat{\delta_B}(q_B, u)\right] = \left[\widehat{\delta_B}(q_B, v)\right] \Leftrightarrow \\
& \widehat{\delta_B}(q_B, u) \sim \widehat{\delta_B}(q_B, v) \Leftrightarrow \\
& \hat{\delta}(q_0, u) \sim \hat{\delta}(q_0, v)
\end{aligned}
$$

- Thus $Q_B/\sim$ is at least as large as $Acc(q_0)/\sim$...
- ...and Q_B is at least as large as $Q_B/\sim$.

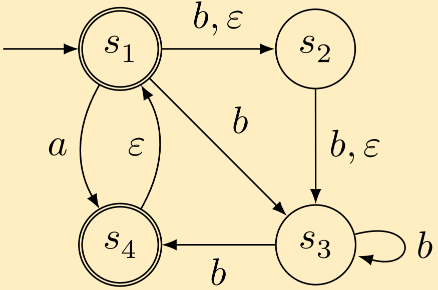
Minimisation

In fact, the minimised DFA is equal
(up to renaming of states)
to every other minimal DFA for the same language.

Minimise the following DFA.



Minimise the following ε-NFA over $\{ a, b \}$.



Today

- Is the language empty?
- Is the string a member of the language?
- Equivalence of states.
- Are the languages equal?
- Minimisation of DFAs.

Next lecture

- Context-free grammars.

- Deadline for the next quiz: 2019-02-21, 10:00.
- Deadline for the fourth assignment: 2019-02-24, 23:59.