Minimization of Finite Automata

Prof. (Dr.) K.R. Chowdhary

Former Professor & Head, Department of Computer Sc. & Engineering
MBM Engineering College, Jodhpur

K.R. Chowdhary

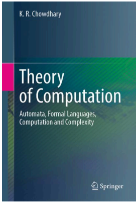
Theory of Computation

Automata, Formal Languages, Computation and Complexity

Focuses on pedagogy in its writing, that represents a refreshing approach

Ensures comprehensive and enjoyable learning

Undergone a rigorous classroom testing



Introduction

- Each DFA defines a unique language but reverse is not true.
- Larger number of states in FA require higher memory and computing power.
- An *NFA* of n states result to 2^n maximum number of states in an equivalent *DFA*, therefore design of *DFA* is crucial.
- Minimization of a *DFA* refers to detecting those states whose absence does not affect the language acceptability of *DFA*.
- A reduced Automata consumes lesser memory, and complexity of implementation is reduced. This results to faster execution time, easier to analyze.

Some definitions

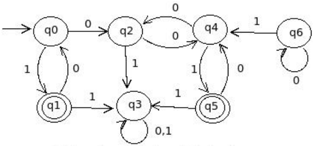
- Unreachable states: If there does not exist any q', such that $\delta^*(q_0, w) = q'$, then q' is unreachable/unaccessible state.
- Dead state: $\forall a$, $a \in \Sigma$, q is dead state if $\delta(q, a) = q$ and $q \in Q - F$.
- Reachability: $FA\ M$ is accessible if $\exists w$, $w \in \Sigma^*$, and $(q_0, w) \vdash^* (q, \varepsilon)$ for all $q \in Q$. $\vdash^*$ is called **reachability** relation.
- Indistinguishable states: Two states are indistinguishable if their behavior are indistinguishable with respect to each other. For example, p, q are indistinguishable if $\delta^*(p, w) = \delta^*(q, w) = r \in Q$ for all $w \in \Sigma^*$.
- k-equivalence: p, q are $k-$equivalence if:

  $\delta^*(q, w) \in F \Leftrightarrow \delta^*(p, w) \in F$,

  for all $w \in \Sigma^*$ and $|w| \leq k$; written as $p \sim_k q$.

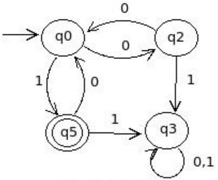
  If they are equivalent for all k, then $p \sim k$. The $p \sim q$ and $p \sim_k q$ are equivalent relations.

Minimization Example



Finite automata to be minimized

- q_6 has no role, hence it can be removed.
- q_1, q_5 are indistinguishable states because their behavior is identical for any string supplied at these states. These are called equivalent states, and can be merged.
- In merging of two equivalent states, one state is eliminated, and the state which remains will have in addition, all incoming transitions from the removed state.
- Similarly, the states q_0, q_4 are also indistinguishable states, hence they can also be merged. q_3 is dead state.



Minimized FA

Formalism for minimization

1. Identify and remove all unreachable states: find all reachable states R, the non-reachable states are $Q - R$.

$$
\begin{aligned}
& R = \{q_0\} \\
& \textit{while } \exists p,\ p \in R \wedge \exists a, a \in \Sigma, \\
& \textit{and } \delta(p,a) \notin R \\
& \{ \\
& R = R \cup \delta(p,a) \\
& \}
\end{aligned}
$$

2. Identify and merge of indistinguishable states.
3. Identify and merge of dead states.
4. A sequence w is accepted if $\delta^*(q,w) \in F$

Indistinguishability is an equivalence relation. Let $p, q, r \in Q$. Let $p \equiv q$, if they are indistinguishable. So,

$p \equiv p$; *reflexive*

$p \equiv q \Leftrightarrow q \equiv p$; *symmetry*

$p \equiv q, q \equiv r \Rightarrow p \equiv r$; *transitivity*, $\therefore$, indistinguishablity is an equivalence relation.

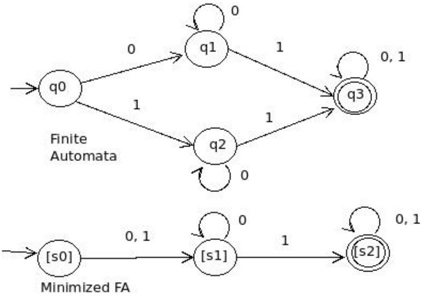
Formalism for minimization

- Let $x, y \in \Sigma^*$, then x and y are said to be *equivalent with respect to* L (i.e. $x \approx_L y$), if for some $z \in \Sigma^*$, $xy \in L$ iff $yz \in L$.
- $\approx_L$ relation is *reflexive, symmetric, and transitive*, $\therefore$, it is *equivalence* relation, which divides the language set L into equivalence classes.
- For a *DFA* M; $x, y \in \Sigma^*$ are equivalent with respect to M, if x, y both drive M from a state q_0 to same state q',

  $\delta^*(q_0, x) = q'$ and
  $\delta^*(q_0, y) = q'$,

  $\therefore$, $x \approx_M y$

Minimization Example#1



1. There is no unreachable state
2. Indistinguishable states

   q_1, q_2 are indistinguishable, and q_0, q_3 are distinguishable
3. Reduced automata: The set of distinguishable states are:

   $[s_0] = \{q_0\}, [s_1] = \{q_1, q_2\}, [s_2] = \{q_3\}$.

   Start and final states are $[s_0], [s_2]$.

Minimization Algorithm

The minimization algorithm is based on the following theorem:

Theorem

Let $\delta(p,a) = p'$ and $\delta(q,a) = q'$, for $a \in \Sigma$. If p', q' are distinguishable then so are p, q.

Proof.

If p', q' are distinguishable by wa then p, q are distinguishable by string w. □

Minimization Algorithm(Table Filling Algorithm)

1. Remove inaccessible/unreachable states:

   delete $Q - Q_R$, where Q_R is set of accessible states.

2. Marking distinguishable states:

   - Mark p, q as distinguishable, where $p \in F, q \notin F$
   - For all marked pairs p, q and $a \in \Sigma$, if $\delta(p, a), \delta(q, a)$ is already marked distinguishable then mark p, q as distinguishable.

3. Construct reduced automata:

   - Let the set of indistinguishable(equivalent) states be sets $[p_i], [q_j], \dots$ such that $\forall i, j \; [p_i] \cap [q_j] = \phi$ and $[p_i] \cup [q_j] \cup \cdots = Q_R$.
   - For each $\delta(p_i, a) = q_j$, add an edge from $[p_i]$ to $[q_j]$

4. Mark the start and final states:

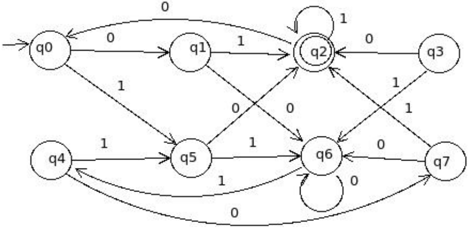
   - if $q_0 \in [p_i]$ then mark $[p_i]$ as start state,
   - if $q_f \in F$, then mark $[q_f]$ as final state.

Implementation of Table Filling Algorithm

Steps:

1. Let $M = (Q, \Sigma, \delta, s, F)$. Remove all the non-reachable states.
2. For $p \in F$ and $q \in Q - F$, put "x" in table at (p, q). This shows that p, q are distinguishable.
3. If $\exists w$, such that $\delta^*(p, w) \in F$ and $\delta^*(q, w) \notin F$, mark (p, q) as distinguishable.
4. Recursion rule: if $\delta^*(p, w) = r, \delta^*(q, w) = s$, and (r, s) were earlier proved distinguishable, then mark (p, q) also distinguishable in the table.

Example: Table Filling algorithm to minimize a FA

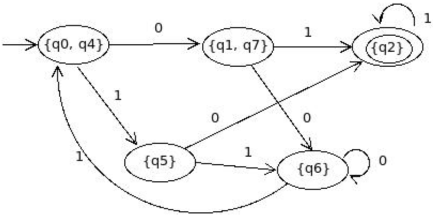


- Consider that we want to minimize the FA shown above. The state q_3 is unreachable, so it can be dropped.
- Next, we mark the distinguishable states at begin as final and non-final states. and make their entries in table as $(q_2, q_0), (q_2, q_1), (q_4, q_2), (q_5, q_2), (q_6, q_2), (q_7, q_2)$ and indicate these by mark "x."

	q_0	q_1	q_2	q_4	q_5	q_6
q_1	x					
q_2	x	x				
q_4		x	x			
q_5	x	x	x	x		
q_6	x	x	x	x	x	
q_7			x	x	x	x

- Next we consider the case $\delta(q_0, 1) = q_5, \delta(q_1, 1) = q_2$. Since (q_5, q_2) are already marked distinguishable, therefore, (q_0, q_1) are also distinguishable.
- Like this we have filled the table shown above. The unmarked are indistinguishable states.

Example: Table Filling algorithm to minimize a FA...



- Only states pairs which are not marked distinguishable are $\{q_0, q_4\}$ and $\{q_1, q_7\}$. The automata shown in figure above is reduced automata.

Bibliography

Chowdhary, K.R. (2025). Minimization of Finite Automata. In: Theory of Computation. Springer, Singapore. https://doi.org/10.1007/978-981-97-6234-7_4