CS310 : Automata Theory 2019

Lecture 4: Subset construction

Instructor: Ashutosh Gupta

IITB, India

Compile date: 2019-01-10

Announcements

- Join piazza (some students have not joined yet; all communications via piazza)
- First quiz timing? 7PM 23(W),24(T),25(F),26(S) Jan30

What we have seen?

We have seen two fundamental definitions of automata theory

- DFA
- NFA

Reverse question

Theorem 4.1
For each NFA $A = (Q, \Sigma, \delta, q_0, F)$, there is a DFA A' such that $L(A) = L(A')$.

Wait! First non-trivial theorem. Can you see why this may be true?

NFAs have more transitions, but still do not recognize more languages.

¯_(ツ)_/¯

Meta comment on automata theory

Most of the theorems in automata theory are about

constructing an automaton

that satisfies certain property and recognizes a given language.

Subset construction

We had NFA $A = (Q, \Sigma, \delta, q_0, F)$.

Let us construct the following DFA

$$A' = (\mathfrak{p}(Q), \Sigma, \delta', \{q_0\}, F'),$$

where

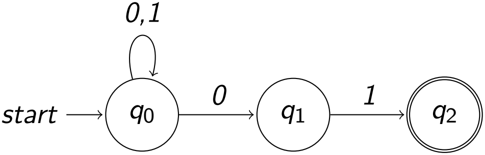
- for each $S \subseteq Q$, $\delta'(S, a) \triangleq \cup_{q \in S}\delta(q, a)$,and
- $F' \triangleq \{S \subseteq Q | S \cap F \neq \emptyset\}$, i.e., all subsets of Q that have states from F.

We will prove that $L(A') = L(A)$.

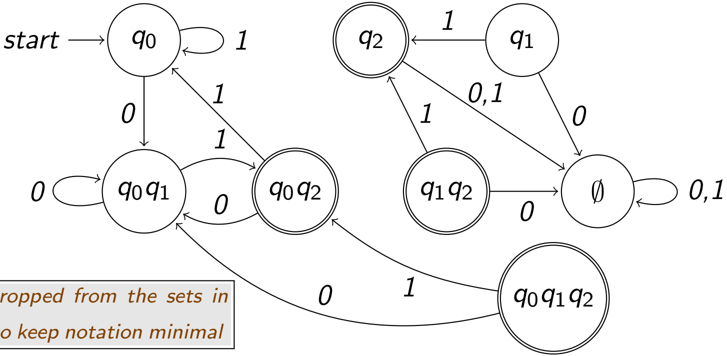
First let us see the construction on an example.

Example: subset construction

Example 4.1



The following is a DFA obtained by subset construction.

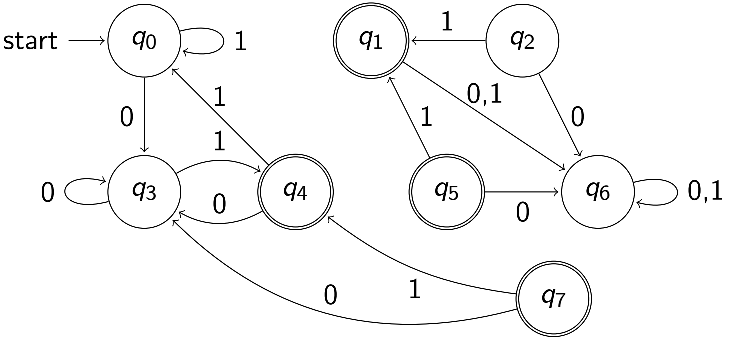


$\{..,..\}$ is dropped from the sets in the states to keep notation minimal

Example: continued

The names of states have no meaning.

By giving them ordinary names, it is clear that the following is just a DFA.



Proof of correctness of subset construction

Proof of theorem 4.1.

Let us recall! We had NFA $A = (Q, \Sigma, \delta, q_0, F)$ and we constructed DFA

$$A' = (\mathfrak{p}(Q), \Sigma, \delta', \{q_0\}, F')$$

where for each $S \subseteq Q$, $\delta'(S, a) \triangleq \cup_{q \in S} \delta(q, a)$ and $F' \triangleq \{S \subseteq Q | S \cap F \neq \emptyset\}$.

We will first prove

$$\hat{\delta}(q_0, w) = \hat{\delta}'(\{q_0\}, w)$$

by induction on the length of w.

...

Exercise 4.1

How does the above equality type-check?

Proof of correctness of subset construction (contd.)

Proof of theorem 4.1(contd.).

base case:

Let $w = \epsilon$.

$\hat{\delta}(q_0, \epsilon) = \{q_0\}$ due to the def of NFA extended transitions

$\hat{\delta'}(\{q_0\}, \epsilon) = \{q_0\}$ due to the def of DFA extended transitions

Therefore,

$$\hat{\delta}(q_0, \epsilon) = \hat{\delta'}(\{q_0\}, \epsilon)$$

...

Proof of correctness of subset construction (contd.)

Proof of theorem 4.1(contd.).

induction step:

Let $w = xa$, where x is a word in Σ^* and a is a letter in Σ.

Due to induction hypothesis, we assume $\hat{\delta}(q_0, x) = \hat{\delta'}(\{q_0\}, x) = S$.

Due to the definition of $\hat{\delta}$, $\hat{\delta}(q_0, xa) = \cup_{q \in S}\delta(q, a)$.

Due to the definition of $\hat{\delta'}$, $\hat{\delta'}(\{q_0\}, xa) = \delta'(S, a) = \bigcup_{q \in S} \delta(q, a)$.

Since δ' is defined in the terms of δ, we apply the definition.

Therefore, $\hat{\delta}(q_0, xa) = \hat{\delta'}(\{q_0\}, xa)$. ...

Proof of correctness of subset construction (contd.)

Proof of theorem 4.1(contd.).

claim: $L(A) = L(A')$

If w is accepted by A, $\hat{\delta}(q_0, w)$ has a state q such that $q \in F$.

- iff, $q \in \hat{\delta'}(\{q_0\}, w)$, which is $\hat{\delta'}(\{q_0\}, w) \cap F \neq \emptyset$.
- iff, $\hat{\delta'}(\{q_0\}, w) \in F'$.
- iff, w is accepted by A'.

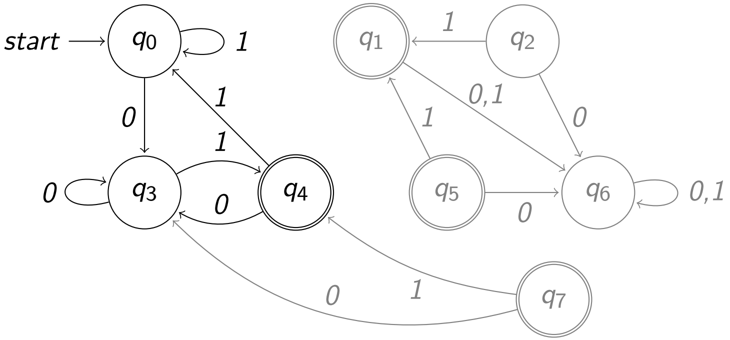
□

Complexity of the subset construction

- If NFA has n states DFA potentially has 2^n states. exponential blowup
- However, not all states are reachable most of the times.

Example 4.2

Consider our example.



Out of 8 states, only 3 states are reachable from the initial state.

Idea alert! : exponential blowup

Exponential	Bad!
Polynomial	Good!

Incremental generation of DFA

Algorithm 4.1: $\mathrm{NFA2DFA}($ NFA $A = (Q, \Sigma, \delta, q_0, F)$ $)$

Output: DFA $A' = (Q', \Sigma, \delta', \{q_0\}, F')$
$Q' := \emptyset$;
$\delta' := \emptyset$;
$F' := \emptyset$;
$worklist := \{\{q_0\}\}$;
while $worklist \neq \emptyset$ **do**
  choose $S \in worklist$;
  $worklist := worklist \setminus \{S\}$;
  if $S \in Q'$ **then continue**;
  $Q' := Q' \cup \{S\}$;
  if $S \cap F \neq \emptyset$ **then** $F' := F' \cup \{S\}$;
  foreach $a \in \Sigma$ **do**
    $S' := \cup_{q \in S} \delta(q, a)$;
    $\delta' := \delta'[(S, a) \mapsto S']$;
    $worklist := worklist \cup \{S'\}$

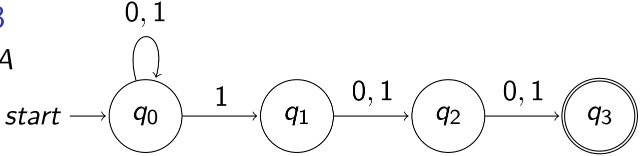
return $(Q', \Sigma, \delta', \{q_0\}, F')$

The above algorithm avoids exponential blow up, if the output DFA does not have exponentially many states.

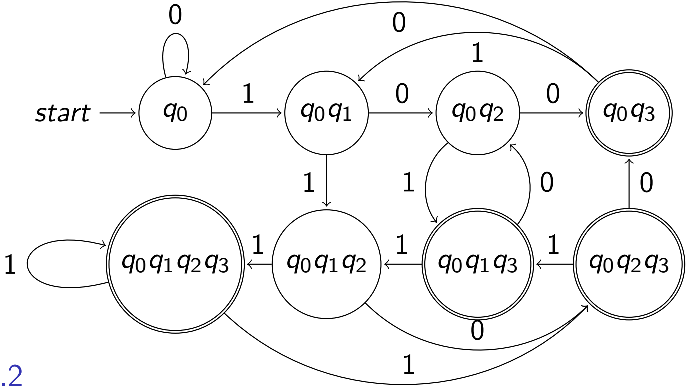
Example: incremental DFA construction

Example 4.3

Consider NFA



Let us construct an equivalent DFA:



Exercise 4.2

What fraction of subset states are reached?

Worst case example

Theoretically, there is a potential exponential explosion.

How do we know there is necessary explosion for some NFA's?

Exercise 4.3

a. Does one such example be enough?

b. How do we prove that there is no small equivalent DFA for an NFA?

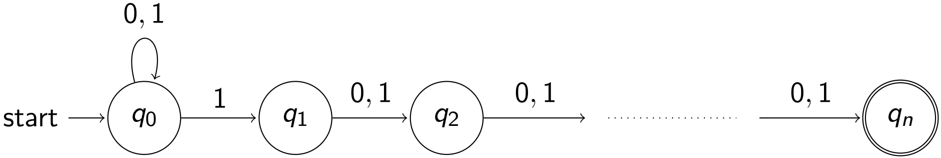
Proving lower size bound

- The evidence of blow up is a family of examples.
  - For each number n, there is a larger example
- For each example in family, we show that there is a contradiction if a small DFA exists.

Exponential blow up family I

For some n, $L_n = \{w | n\text{th symbol from the end is } 1\}$.

The following NFA recognizes L_n.



Since we do not know when the word is going to end, DFA needs to keep the record of last n symbols.

Exponential blow up family II

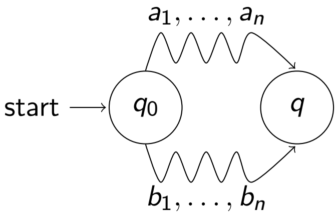
Theorem 4.2

No DFA states fewer than 2^n states can recognize L_n.

Proof.

Let us suppose such a DFA A exists.

There must be two different words $a_1, ..., a_n$ and $b_1, ..., b_n$ such that both of them end up in the same state q of A.(why?)



Let us suppose a_i and b_i are different. ...

Exponential blow up family II

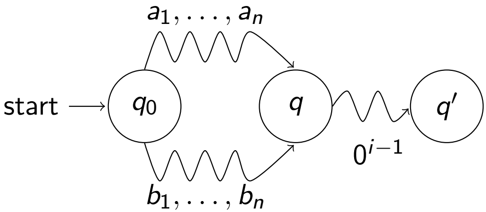
Proof(contd.)

Without loss of generality(why?), we can assume $a_i = 0$ and $b_i = 1$.

Now consider words

- $w = a_1, ..., a_i, ..., a_n, 0^{i-1}$, which is not in L_n.
- $w' = b_1, ..., b_i, ..., b_n, 0^{i-1}$, which is in L_n.

Since A is DFA, w amd w' will finish in the same state of A, say q'.(why?)



A will either accept or reject both w or w' simultaneously. Contradiction. □

Beyond regular languages

NFAs and DFAs can recognize same regular languages.

If we add more features in the automaton, would we cover more languages?

To be continued...

End of Lecture 4