5.1.2
Algorithm for converting NFA to DFA

Recall I

Extending the transition function to strings

Definition

For NFA $N = (Q, \Sigma, \delta, s, A)$ and $q \in Q$ the $\epsilon\text{reach}(q)$ is the set of all states that q can reach using only ϵ-transitions.

Definition

Inductive definition of $\delta^* : Q \times \Sigma^* \to \mathcal{P}(Q)$:

- if $w = \varepsilon$, $\delta^*(q, w) = \epsilon\text{reach}(q)$
- if $w = a$ where $a \in \Sigma$: $\delta^*(q, a) = \epsilon\text{reach}\left(\bigcup_{p \in \epsilon\text{reach}(q)} \delta(p, a)\right)$
- if $w = ax$: $\delta^*(q, w) = \epsilon\text{reach}\left(\bigcup_{p \in \epsilon\text{reach}(q)} \bigcup_{r \in \delta^*(p,a)} \delta^*(r, x)\right)$

Recall II

Formal definition of language accepted by N

Definition

A string w is accepted by NFA N if $\delta_N^*(s, w) \cap A \neq \emptyset$.

Definition

The language $L(N)$ accepted by a NFA $N = (Q, \Sigma, \delta, s, A)$ is

$$\{w \in \Sigma^* \mid \delta^*(s, w) \cap A \neq \emptyset\}.$$

Subset Construction

NFA $N = (Q, \Sigma, s, \delta, A)$. We create a DFA $D = (Q', \Sigma, \delta', s', A')$ as follows:

- $Q' = \mathcal{P}(Q)$
- $s' = \epsilon\text{reach}(s) = \delta^*(s, \epsilon)$
- $A' = \{X \subseteq Q \mid X \cap A \neq \emptyset\}$
- $\delta'(X, a) = \cup_{q \in X} \delta^*(q, a)$ for each $X \subseteq Q$, $a \in \Sigma$.

Subset Construction

NFA $N = (Q, \Sigma, s, \delta, A)$. We create a DFA $D = (Q', \Sigma, \delta', s', A')$ as follows:

- $Q' = \mathcal{P}(Q)$
- $s' = \epsilon\text{reach}(s) = \delta^*(s, \epsilon)$
- $A' = \{X \subseteq Q \mid X \cap A \neq \emptyset\}$
- $\delta'(X, a) = \cup_{q \in X} \delta^*(q, a)$ for each $X \subseteq Q$, $a \in \Sigma$.

Subset Construction

NFA $N = (Q, \Sigma, s, \delta, A)$. We create a DFA $D = (Q', \Sigma, \delta', s', A')$ as follows:

- $Q' = \mathcal{P}(Q)$
- $s' = \epsilon\text{reach}(s) = \delta^*(s, \epsilon)$
- $A' = \{X \subseteq Q \mid X \cap A \neq \emptyset\}$
- $\delta'(X, a) = \cup_{q \in X} \delta^*(q, a)$ for each $X \subseteq Q$, $a \in \Sigma$.

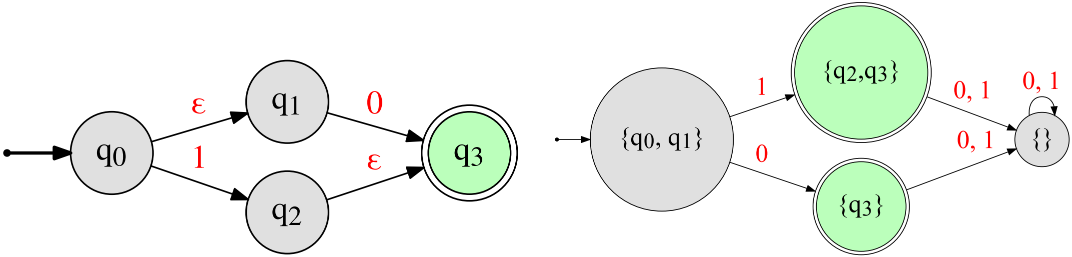
Subset Construction

NFA $N = (Q, \Sigma, s, \delta, A)$. We create a DFA $D = (Q', \Sigma, \delta', s', A')$ as follows:

- $Q' = \mathcal{P}(Q)$
- $s' = \epsilon\text{reach}(s) = \delta^*(s, \epsilon)$
- $A' = \{X \subseteq Q \mid X \cap A \neq \emptyset\}$
- $\delta'(X, a) = \cup_{q \in X} \delta^*(q, a)$ for each $X \subseteq Q$, $a \in \Sigma$.

Incremental construction

Only build states reachable from $s' = \epsilon\text{reach}(s)$ the start state of D



$\delta'(X, a) = \cup_{q \in X} \delta^*(q, a)$.

An optimization: Incremental algorithm

- Build D beginning with start state $s' == \epsilon\text{reach}(s)$
- For each existing state $X \subseteq Q$ consider each $a \in \Sigma$ and calculate the state $U = \delta'(X, a) = \cup_{q \in X} \delta^*(q, a)$ and add a transition.

  To compute $Z_{q,a} = \delta^*(q, a)$ - set of all states reached from q on character a
  - Compute $X_1 = \epsilon\text{reach}(q)$
  - Compute $Y_1 = \cup_{p \in X_1} \delta(p, a)$
  - Compute $Z_{q,a} = \epsilon\text{reach}(Y) = \cup_{r \in Y_1} \epsilon\text{reach}(r)$
- If U is a new state add it to reachable states that need to be explored.

An optimization: Incremental algorithm

- Build D beginning with start state $s' == \epsilon\text{reach}(s)$
- For each existing state $X \subseteq Q$ consider each $a \in \Sigma$ and calculate the state $U = \delta'(X, a) = \cup_{q \in X} \delta^*(q, a)$ and add a transition.

  To compute $Z_{q,a} = \delta^*(q, a)$ - set of all states reached from q on character a
  - Compute $X_1 = \epsilon\text{reach}(q)$
  - Compute $Y_1 = \cup_{p \in X_1} \delta(p, a)$
  - Compute $Z_{q,a} = \epsilon\text{reach}(Y) = \cup_{r \in Y_1} \epsilon\text{reach}(r)$
- If U is a new state add it to reachable states that need to be explored.

An optimization: Incremental algorithm

- Build D beginning with start state $s' == \epsilon\text{reach}(s)$
- For each existing state $X \subseteq Q$ consider each $a \in \Sigma$ and calculate the state $U = \delta'(X, a) = \cup_{q \in X} \delta^*(q, a)$ and add a transition.

  To compute $Z_{q,a} = \delta^*(q, a)$ - set of all states reached from q on character a
  - Compute $X_1 = \epsilon\text{reach}(q)$
  - Compute $Y_1 = \cup_{p \in X_1} \delta(p, a)$
  - Compute $Z_{q,a} = \epsilon\text{reach}(Y) = \cup_{r \in Y_1} \epsilon\text{reach}(r)$
- If U is a new state add it to reachable states that need to be explored.

THE END

...

(for now)