18.6

DFA to Regular Expression

Back to Regular Languages

We saw the following two theorems previously.

Theorem 18.1.

For every NFA $\boldsymbol{N}$ *over a finite alphabet* Σ *there is* DFA $\boldsymbol{M}$ *such that* $\boldsymbol{L}(\boldsymbol{M}) = \boldsymbol{L}(\boldsymbol{N})$.

Theorem 18.2.

For every regular expression $\boldsymbol{r}$ *over finite alphabet* Σ *there is a* NFA $\boldsymbol{N}$ *such that* $\boldsymbol{L}(\boldsymbol{N}) = \boldsymbol{L}(\boldsymbol{r})$.

We claimed the following theorem which would prove equivalence of NFAs, DFAs and regular expressions.

Theorem 18.3.

For every DFA $\boldsymbol{M}$ *over a finite alphabet* Σ *there is a regular expression* $\boldsymbol{r}$ *such that* $\boldsymbol{L}(\boldsymbol{M}) = \boldsymbol{L}(\boldsymbol{r})$.

Back to Regular Languages

We saw the following two theorems previously.

Theorem 18.1.

For every NFA $\boldsymbol{N}$ over a finite alphabet Σ there is DFA $\boldsymbol{M}$ such that $\boldsymbol{L}(\boldsymbol{M}) = \boldsymbol{L}(\boldsymbol{N})$.

Theorem 18.2.

For every regular expression $\boldsymbol{r}$ over finite alphabet Σ there is a NFA $\boldsymbol{N}$ such that $\boldsymbol{L}(\boldsymbol{N}) = \boldsymbol{L}(\boldsymbol{r})$.

We claimed the following theorem which would prove equivalence of NFAs, DFAs and regular expressions.

Theorem 18.3.

For every DFA $\boldsymbol{M}$ over a finite alphabet Σ there is a regular expression $\boldsymbol{r}$ such that $\boldsymbol{L}(\boldsymbol{M}) = \boldsymbol{L}(\boldsymbol{r})$.

DFA to Regular Expression

Given DFA $M = (Q, \Sigma, \delta, q_1, F)$ want to construct an equivalent regular expression r.

Idea:

- Number states of DFA: $Q = \{q_1, \ldots, q_n\}$ where $|Q| = n$.
- Define $L_{i,j} = \{w \mid \delta(q_i, w) = q_j\}$. Note $L_{i,j}$ is regular. Why?
- $L(M) = \cup_{q_i \in F} L_{1,i}$.
- Obtain regular expression $r_{i,j}$ for $L_{i,j}$.
- Then $r = \sum_{q_i \in F} r_{1,i}$ is regular expression for $L(M)$ – here the summation is the `or` operator.

Note: Using q_1 for start state is intentional to help in the notation for the recursion.

A recursive expression for $\mathbf{L_{i,j}}$

Define $L_{i,j}^{k}$ be set of strings w in $L_{i,j}$ such that the highest index state visited by M on walk from q_i to q_j (not counting end points i and j) on input w is at most k.

Claim:

$$L_{i,i}^{0} = \{a \in \Sigma \mid \delta(q_i, a) = q_i\}^*$$

$$L_{i,j}^{0} = L_{i,i}^{0}\{a \in \Sigma \mid \delta(q_i, a) = q_j\}L_{j,j}^{0} \qquad \text{if } i \neq j$$

$$L_{i,j}^{k} = L_{i,j}^{k-1} \cup \left(L_{i,k}^{k-1} \cdot L_{k,k}^{k-1} \cdot L_{k,j}^{k-1}\right) \qquad i \neq j$$

$$L_{i,i}^{k} = \left(L_{i,i}^{k-1} \cup L_{i,k}^{k-1} \cdot L_{k,k}^{k-1} \cdot L_{k,i}^{k-1}\right)^*$$

$$L_{i,j} = L_{i,j}^{n}.$$

A recursive expression for $\mathbf{L_{i,j}}$

Claim:

$$L^0_{i,i} = \{a \in \Sigma \mid \delta(q_i, a) = q_i\}^*$$

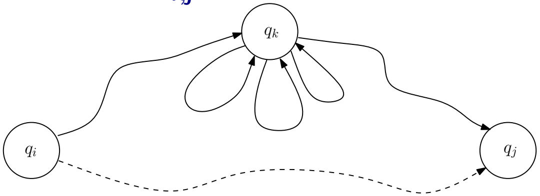
$$L^0_{i,j} = L^0_{i,i}\{a \in \Sigma \mid \delta(q_i, a) = q_j\}L^0_{j,j} \qquad \text{if } i \neq j$$

$$L^k_{i,j} = L^{k-1}_{i,j} \cup \left(L^{k-1}_{i,k} \cdot L^{k-1}_{k,k} \cdot L^{k-1}_{k,j}\right) \qquad i \neq j$$

$$L^k_{i,i} = \left(L^{k-1}_{i,i} \cup L^{k-1}_{i,k} \cdot L^{k-1}_{k,k} \cdot L^{k-1}_{k,i}\right)^*$$

$$L_{i,j} = L^n_{i,j}.$$

Proof: by picture



A recursive expression for $\mathbf{L_{i,j}}$

Claim:

$$
\begin{aligned}
L^0_{i,i} &= \{a \in \Sigma \mid \delta(q_i, a) = q_i\}^* \\
L^0_{i,j} &= L^0_{i,i}\{a \in \Sigma \mid \delta(q_i, a) = q_j\}L^0_{j,j} && \text{if } i \neq j \\
L^k_{i,j} &= L^{k-1}_{i,j} \cup \left(L^{k-1}_{i,k} \cdot L^{k-1}_{k,k} \cdot L^{k-1}_{k,j}\right) && i \neq j \\
L^k_{i,i} &= \left(L^{k-1}_{i,i} \cup L^{k-1}_{i,k} \cdot L^{k-1}_{k,k} \cdot L^{k-1}_{k,i}\right)^* \\
L_{i,j} &= L^n_{i,j}.
\end{aligned}
$$

The desired language is

$$L(M) = \cup_{q_i \in F} L_{1,i} = \cup_{q_i \in F} L^n_{1,i}$$

A regular expression for $\mathbf{L(M)}$

$$r_{i,i}^{0} = \left(\sum_{a\in\Sigma:\delta(q_i,a)=q_i} a\right)^{*}$$

$$r_{i,j}^{0} = r_{i,i}^{0}\left(\sum_{a\in\Sigma:\delta(q_i,a)=q_j} a\right) r_{j,j}^{0} \qquad \text{if } i \neq j$$

$$r_{i,j}^{k} = r_{i,j}^{k-1} + r_{i,k}^{k-1} r_{k,k}^{k-1} r_{k,j}^{k-1} \qquad i \neq j$$

$$r_{i,i}^{k} = \left(r_{i,i}^{k-1} + r_{i,k}^{k-1} \cdot r_{k,k}^{k-1} \cdot r_{k,i}^{k-1}\right)^{*}$$

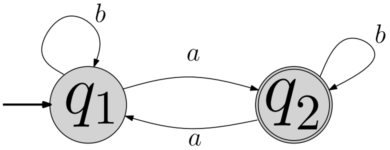
$$r_{i,j} = r_{i,j}^{n}.$$

The desired regular expression is: reg-expression$(M) = \sum_{q_i\in F} r_{1,i} = \sum_{q_i\in F} r_{1,i}^{n}$.

Example

$$r^0_{1,1} = r^0_{2,2} = b^*$$

$$r^0_{1,2} = r^0_{2,1} = b^*ab^*$$



$$r^1_{1,1} = (r^0_{1,1} + r^0_{1,1}r^0_{1,1}r^0_{1,1})^* = b^*$$

$$r^1_{2,2} = \left(r^0_{2,2} + r^0_{2,1}r^0_{1,1}r^0_{1,2}\right)^* = (b^* + b^*ab^*b^*b^*ab^*)^* = (b^* + ab^*a)^*$$

$$r^1_{1,2} = r^0_{1,2} + r^0_{1,1}r^0_{1,1}r^0_{1,2} = b^*ab^* + b^*b^*ab^* = b^*ab^*.$$

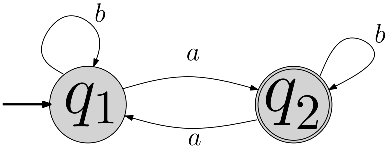
$$r^1_{2,1} = r^0_{1,2} + r^0_{2,1}r^0_{1,1}r^0_{1,1} = b^*ab^*$$

$$r^2_{1,1} = \left(r^1_{1,1} + r^1_{1,2}r^1_{2,2}r^1_{2,1}\right)^* = \cdots$$

$$r^1_{2,2} = \cdots$$

Example

$r^0_{1,1} = r^0_{2,2} = b^*$ $\qquad$ $r^0_{1,2} = r^0_{2,1} = b^*ab^*$



$$r^1_{1,1} = (r^0_{1,1} + r^0_{1,1}r^0_{1,1}r^0_{1,1})^* = b^*$$

$$r^1_{2,2} = (r^0_{2,2} + r^0_{2,1}r^0_{1,1}r^0_{1,2})^* = (b^* + b^*ab^*b^*b^*ab^*)^* = (b^* + ab^*a)^*$$

$$r^1_{1,2} = r^0_{1,2} + r^0_{1,1}r^0_{1,1}r^0_{1,2} = b^*ab^* + b^*b^*ab^* = b^*ab^*.$$

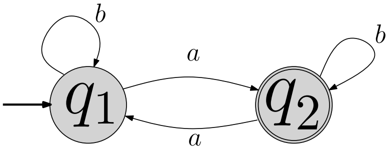
$$r^1_{2,1} = r^0_{1,2} + r^0_{2,1}r^0_{1,1}r^0_{1,1} = b^*ab^*$$

$$r^2_{1,1} = (r^1_{1,1} + r^1_{1,2}r^1_{2,2}r^1_{2,1})^* = \cdots$$

$$r^1_{2,2} = \cdots$$

Example

$r^0_{1,1} = r^0_{2,2} = b^*$ $\qquad$ $r^0_{1,2} = r^0_{2,1} = b^*ab^*$



$$r^1_{1,1} = (r^0_{1,1} + r^0_{1,1}r^0_{1,1}r^0_{1,1})^* = b^*$$

$$r^1_{2,2} = \left(r^0_{2,2} + r^0_{2,1}r^0_{1,1}r^0_{1,2}\right)^* = (b^* + b^*ab^*b^*b^*ab^*)^* = (b^* + ab^*a)^*$$

$$r^1_{1,2} = r^0_{1,2} + r^0_{1,1}r^0_{1,1}r^0_{1,2} = b^*ab^* + b^*b^*ab^* = b^*ab^*.$$

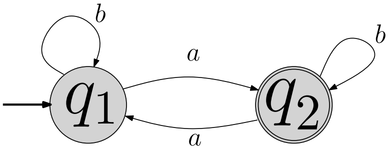
$$r^1_{2,1} = r^0_{1,2} + r^0_{2,1}r^0_{1,1}r^0_{1,1} = b^*ab^*$$

$$r^2_{1,1} = \left(r^1_{1,1} + r^1_{1,2}r^1_{2,2}r^1_{2,1}\right)^* = \cdots$$

$$r^1_{2,2} = \cdots$$

Example

$r^0_{1,1} = r^0_{2,2} = b^*$ $\qquad$ $r^0_{1,2} = r^0_{2,1} = b^*ab^*$



$$r^1_{1,1} = (r^0_{1,1} + r^0_{1,1}r^0_{1,1}r^0_{1,1})^* = b^*$$

$$r^1_{2,2} = \left(r^0_{2,2} + r^0_{2,1}r^0_{1,1}r^0_{1,2}\right)^* = (b^* + b^*ab^*b^*b^*ab^*)^* = (b^* + ab^*a)^*$$

$$r^1_{1,2} = r^0_{1,2} + r^0_{1,1}r^0_{1,1}r^0_{1,2} = b^*ab^* + b^*b^*ab^* = b^*ab^*.$$

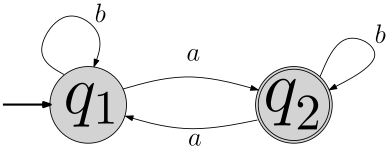
$$r^1_{2,1} = r^0_{1,2} + r^0_{2,1}r^0_{1,1}r^0_{1,1} = b^*ab^*$$

$$r^2_{1,1} = \left(r^1_{1,1} + r^1_{1,2}r^1_{2,2}r^1_{2,1}\right)^* = \cdots$$

$$r^1_{2,2} = \cdots$$

Example

$r^0_{1,1} = r^0_{2,2} = b^*$ $\qquad$ $r^0_{1,2} = r^0_{2,1} = b^*ab^*$



$$r^1_{1,1} = (r^0_{1,1} + r^0_{1,1} r^0_{1,1} r^0_{1,1})^* = b^*$$

$$r^1_{2,2} = \left(r^0_{2,2} + r^0_{2,1} r^0_{1,1} r^0_{1,2}\right)^* = (b^* + b^*ab^*b^*b^*ab^*)^* = (b^* + ab^*a)^*$$

$$r^1_{1,2} = r^0_{1,2} + r^0_{1,1} r^0_{1,1} r^0_{1,2} = b^*ab^* + b^*b^*ab^* = b^*ab^*.$$

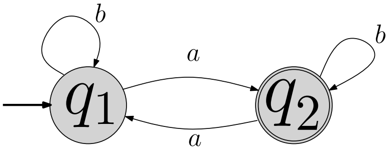
$$r^1_{2,1} = r^0_{1,2} + r^0_{2,1} r^0_{1,1} r^0_{1,1} = b^*ab^*$$

$$r^2_{1,1} = \left(r^1_{1,1} + r^1_{1,2} r^1_{2,2} r^1_{2,1}\right)^* = \cdots$$

$$r^1_{2,2} = \cdots$$

Example

$$r^0_{1,1} = r^0_{2,2} = b^* \qquad r^0_{1,2} = r^0_{2,1} = b^*ab^*$$



$$r^1_{1,1} = (r^0_{1,1} + r^0_{1,1}r^0_{1,1}r^0_{1,1})^* = b^*$$

$$r^1_{2,2} = \left(r^0_{2,2} + r^0_{2,1}r^0_{1,1}r^0_{1,2}\right)^* = (b^* + b^*ab^*b^*b^*ab^*)^* = (b^* + ab^*a)^*$$

$$r^1_{1,2} = r^0_{1,2} + r^0_{1,1}r^0_{1,1}r^0_{1,2} = b^*ab^* + b^*b^*ab^* = b^*ab^*.$$

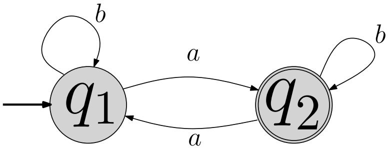
$$r^1_{2,1} = r^0_{1,2} + r^0_{2,1}r^0_{1,1}r^0_{1,1} = b^*ab^*$$

$$r^2_{1,1} = \left(r^1_{1,1} + r^1_{1,2}r^1_{2,2}r^1_{2,1}\right)^* = \cdots$$

$$r^1_{2,2} = \cdots$$

Example

$$r^0_{1,1} = r^0_{2,2} = b^* \qquad r^0_{1,2} = r^0_{2,1} = b^*ab^*$$



$$r^1_{1,1} = (r^0_{1,1} + r^0_{1,1}r^0_{1,1}r^0_{1,1})^* = b^*$$
$$r^1_{2,2} = \left(r^0_{2,2} + r^0_{2,1}r^0_{1,1}r^0_{1,2}\right)^* = (b^* + b^*ab^*b^*b^*ab^*)^* = (b^* + ab^*a)^*$$
$$r^1_{1,2} = r^0_{1,2} + r^0_{1,1}r^0_{1,1}r^0_{1,2} = b^*ab^* + b^*b^*ab^* = b^*ab^*.$$
$$r^1_{2,1} = r^0_{1,2} + r^0_{2,1}r^0_{1,1}r^0_{1,1} = b^*ab^*$$
$$r^2_{1,1} = \left(r^1_{1,1} + r^1_{1,2}r^1_{2,2}r^1_{2,1}\right)^* = \cdots$$
$$r^1_{2,2} = \cdots$$

Correctness

Similar to that of Floyd-Warshall algorithms for shortest paths via induction.

The length of the regular expression can be exponential in the size of the original DFA.

THE END

...

(for now)