COS210 - Theoretical Computer Science
Finite Automata and Regular Languages (Part 7)

Regular Expressions

The following theorem holds:

Theorem (1)

Let L be a language, then:

$$L \text{ is regular} \\ \Leftrightarrow \\ \text{there exists a regular expression } R \text{ that describes } L$$

$\Longleftarrow$: (proven in Lecture 8)

Theorem (1A)

Every regular expression R describes a language $L(M)$ where M is a finite automaton.

$\Longrightarrow$:

Theorem (1B)

For every finite automaton M, the language $L(M)$ can be described by a regular expression R.

Intermediate Result: Recurrence Relation

Lemma

Let B, C, and L be languages over an alphabet Σ such that

$$\epsilon \notin B \quad \text{and} \quad L = BL \cup C$$

then

$$L = B^*C$$

Proof:

Two parts:

- $B^*C \subseteq L$
- $L \subseteq B^*C$

Intermediate Result: Recurrence Relation

Part 1: $B^*C \subseteq L$

Proof by Induction:
Each string of B^*C is of the form $b_k \dots b_1 c$ where $b_k, \dots, b_1 \in B$ and $c \in C$.

We show that $b_k \dots b_1 c \in L$.

Base Case $k = 0$:

String of B^*C is of the form c

$\implies$ $c \in C$

$\implies$ $c \in BL \cup C$ (since $C \subseteq BL \cup C$)

$\implies$ $c \in L$ (since $L = BL \cup C$)

Intermediate Result: Recurrence Relation

Proof by Induction:

Hypothesis:

- Assume that for each $b_k \dots b_1 c \in B^* C$ we also have $b_k \dots b_1 c \in L$

Inductive Step:

- Show that if $\underbrace{\underbrace{b_{k+1}}_{\in B} \underbrace{b_k \dots b_1 c}_{\in L}}_{\in BL} \in B^* C$, then $b_{k+1} b_k \dots b_1 c \in L$

$\Longrightarrow$ $b_{k+1} b_k \dots b_1 c \in BL \cup C$ (since $BL \subseteq BL \cup C$)

$\Longrightarrow$ $b_{k+1} b_k \dots b_1 c \in L$ (since $L = BL \cup C$)

$\Longrightarrow$ We can conclude that $B^* C \subseteq L$

Intermediate Result: Recurrence Relation

Part 2: $L \subseteq B^*C$

Proof by Induction:
Let $l \in L$ and $|l|$ the length of l.

We show that $l \in B^*C$.

Base Case $|l| = 0$:

$\implies \quad l = \epsilon$

and $\quad l \in BL \cup C$ (since $L = BL \cup C$)

$\implies \quad l \notin BL$ (since $\epsilon \notin B$)

$\implies \quad l \in C$

$\implies \quad l \in B^*C$ (since $C \subseteq B^*C$)

Lemma

Let B, C, and L be languages over an alphabet Σ such that

$$\epsilon \notin B \quad \textit{and} \quad L = BL \cup C$$

then

$$L = B^*C$$

Intermediate Result: Recurrence Relation

Hypothesis:

- Assume that for each $l \in L$ with length $|l| \leq k$: $l \in B^*C$

Inductive Step:

- Show that if $l \in L$ with length $|l| = k + 1$, then $l \in B^*C$

$\Longrightarrow$ $l \in BL \cup C$ (since $L = BL \cup C$)

$\Longrightarrow$ $l \in BL$ (**Case 1**) or $l \in C$ (**Case 2**)

Case 2:

$l \in C$

$\Longrightarrow$ $l \in B^*C$ (since $C \subseteq B^*C$)

Intermediate Result: Recurrence Relation

Hypothesis:

- Assume that for each $l \in L$ with length $|l| \leq k$: $l \in B^*C$

Inductive Step:

- Show that if $l \in L$ with length $|l| = k + 1$, then $l \in B^*C$

$\implies$ $l \in BL \cup C$ (since $L = BL \cup C$)

$\implies$ $l \in BL$ (**Case 1**) or $l \in C$ (**Case 2**)

Case 1:

$l \in BL$

$\implies$ $l = l_1 l_2$ where $l_1 \in B$ and $l_2 \in L$

$\implies$ $|l_2| < |l|$ (since $l_1 \neq \epsilon$)

$\implies$ $|l_2| \leq k$

$\implies$ $l_2 \in B^*C$ (Hypothesis)

$\implies$ $l_1 l_2 \in BB^*C$

$\implies$ $l_1 l_2 \in B^*C$ (since $BB^* = B^*$)

$\implies$ $l \in B^*C$

We can conclude that $L \subseteq B^*C$ and also $L = B^*C$ $\square$

$|l_1| \geq 1$

$|l_1 l_2| = |l|$

Converting a DFA to a regular expression:

Back to Theorem 1B:

Theorem (1B)

For every finite automaton M, the language $L(M)$ can be described by a regular expression R.

Proof:

- Let $M = (Q, \Sigma, \delta, q, F)$ be the DFA with language $L(M)$
- For each state $r \in Q$ we define the language:

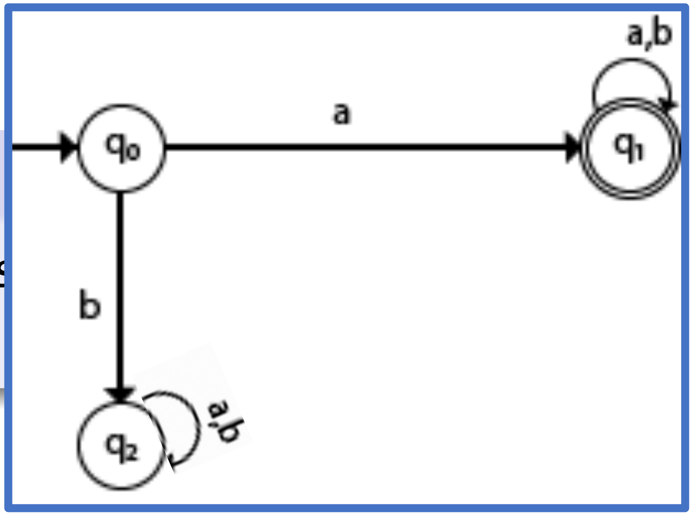
$$L_r = \{w : \text{ starting in the state } r\text{, the run over } w \text{ in } M \text{ ends in a state of } F.\}$$

Converting a DFA to a regular expression:

Back to Theorem 1B:

Theorem (1B)

For every finite automaton M, the language $L(M)$ can be des
regular expression R.



Proof:

- Let $M = (Q, \Sigma, \delta, q, F)$ be the DFA with language $L(M)$
- For each state $r \in Q$ we define the language:

$$L_r = \{w : \text{ starting in the state } r, \text{ the run over } w \text{ in } M \text{ ends in a state of } F.\}$$

$Lq_0 = \{w : w \text{ starts with } a\}$
$Lq_1 = \{w : w \text{ is an arbitrary string over } \{a, b\}\}$
$Lq_2 = \emptyset$

Converting a DFA to a regular expression:

Proof cont:

- How to represent each language L_r as a regular expression:
- If non-accepting state $r \notin F$ then we claim that

$$\boxed{L_r \;=\; \bigcup_{a\in\Sigma} a \cdot L_{\delta(r,a)}}$$

L_r is the union over all symbols a where a is concatenated with the language of the a-successor state of r

(dot $\cdot$ denotes concatenation)

Converting a DFA to a regular expression:

Proof cont:

- How to represent each language L_r as a regular expression:
- If non-accepting state $r \notin F$ then we claim that

$$L_r = \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}$$

L_r is the union over all symbols a where a is concatenated with the language of the a-successor state of r

(dot $\cdot$ denotes concatenation)

- We prove the equivalence by showing that the following relations hold:

$$\boxed{\begin{array}{l} L_r \subseteq \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \\ \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \subseteq L_r \end{array}}$$

Converting a DFA to a regular expression:

Proof cont:

- Part 1: show $L_r \subseteq \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}$
- Equivalent to: If string $w \in L_r$, then $w \in \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}$

Converting a DFA to a regular expression:

Proof cont:

- Part 1: show $L_r \subseteq \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}$
- Equivalent to: If string $w \in L_r$, then $w \in \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}$
- Let P be the run over w in M
- Since $r \notin F$, the run P must contain at least one transition.
- Let $r' = \delta(r, b)$ be the second state of P where b is a the first symbol of w.

Converting a DFA to a regular expression:

Proof cont:

- Part 1: show $L_r \subseteq \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}$
- Equivalent to: If string $w \in L_r$, then $w \in \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}$
- Let P be the run over w in M
- Since $r \notin F$, the run P must contain at least one transition.
- Let $r' = \delta(r, b)$ be the second state of P where b is a the first symbol of w.
- We can rewrite $w = bv$ where v is the remaining part of w
- Run $P' = P \backslash \{r\}$ over v starts in r' and ends in state of F.

$\Longrightarrow \quad v \in L_{r'} = L_{\delta(r,b)}$

Converting a DFA to a regular expression:

Proof cont:

- Part 1: show $L_r \subseteq \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}$
- Equivalent to: If string $w \in L_r$, then $w \in \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}$
- Let P be the run over w in M
- Since $r \notin F$, the run P must contain at least one transition.
- Let $r' = \delta(r, b)$ be the second state of P where b is a the first symbol of w.
- We can rewrite $\boxed{w = bv}$ where v is the remaining part of w
- Run $P' = P \backslash \{r\}$ over v starts in r' and ends in state of F.

$$\implies \quad v \in L_{r'} = L_{\delta(r,b)}$$

$$\implies \quad w \in b \cdot L_{\delta(r,b)}$$

Converting a DFA to a regular expression:

Proof cont:

- $\boxed{\text{Part 1: show } L_r \subseteq \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}}$
- Equivalent to: If string $w \in L_r$, then $w \in \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}$
- Let P be the run over w in M
- Since $r \notin F$, the run P must contain at least one transition.
- Let $r' = \delta(r, b)$ be the second state of P where b is a the first symbol of w.
- We can rewrite $w = bv$ where v is the remaining part of w
- Run $P' = P \backslash \{r\}$ over v starts in r' and ends in state of F.

$$\Longrightarrow \quad v \in L_{r'} = L_{\delta(r,b)}$$
$$\Longrightarrow \quad w \in \boxed{b \cdot L_{\delta(r,b)}}$$
$$\Longrightarrow \quad w \in \boxed{\bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}}$$

$\subseteq$

Converting a DFA to a regular expression:

Proof cont:

- Part 2: show $\bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \subseteq L_r$
- Equivalent to: If string $w \in \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}$, then $w \in L_r$

Converting a DFA to a regular expression:

Proof cont:

- Part 2: show $\bigcup\limits_{a \in \Sigma} a \cdot L_{\delta(r,a)} \subseteq L_r$
- Equivalent to: If string $w \in \bigcup\limits_{a \in \Sigma} a \cdot L_{\delta(r,a)}$, then $w \in L_r$
- Let $w \in \bigcup\limits_{a \in \Sigma} a \cdot L_{\delta(r,a)}$
- There is a symbol $b \in \Sigma$ and a string $v \in L_{\delta(r,b)}$ such that $w = bv$

Converting a DFA to a regular expression:

Proof cont:

- Part 2: show $\bigcup_{a\in\Sigma} a \cdot L_{\delta(r,a)} \subseteq L_r$
- Equivalent to: If string $w \in \bigcup_{a\in\Sigma} a \cdot L_{\delta(r,a)}$, then $w \in L_r$
- Let $w \in \bigcup_{a\in\Sigma} a \cdot L_{\delta(r,a)}$
- There is a symbol $b \in \Sigma$ and a string $v \in L_{\delta(r,b)}$ such that $w = bv$
- Let P' be the run over v in M starting in state $\delta(r,b)$.
- Since $v \in L_{\delta(r,b)}$, this run ends in a state of F.

Converting a DFA to a regular expression:

Proof cont:

- Part 2: show $\bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \subseteq L_r$
- Equivalent to: If string $w \in \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}$, then $w \in L_r$
- Let $w \in \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}$
- There is a symbol $b \in \Sigma$ and a string $v \in L_{\delta(r,b)}$ such that $\boxed{w = bv}$
- Let P' be the run over v in M starting in state $\delta(r, b)$.
- Since $v \in L_{\delta(r,b)}$, this run ends in a state of F.
- Let P be the run that starts in r, takes the transition to $\delta(r, b)$, and then follows P'
- This run is over the string w and ends in state of F

$\Longrightarrow \quad w \in L_r$

Converting a DFA to a regular expression:

Proof cont:

- $\boxed{\text{Part 2: show } \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \subseteq L_r}$
- Equivalent to: If string $w \in \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}$, then $w \in L_r$
- Let $w \in \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}$
- There is a symbol $b \in \Sigma$ and a string $v \in L_{\delta(r,b)}$ such that $w = bv$
- Let P' be the run over v in M starting in state $\delta(r, b)$.
- Since $v \in L_{\delta(r,b)}$, this run ends in a state of F.
- Let P be the run that starts in r, takes the transition to $\delta(r, b)$, and then follows P'
- This run is over the string w and ends in state of F

$\Longrightarrow \quad w \in L_r$

Converting a DFA to a regular expression:

Proof cont:

- For non-accepting states $r \notin F$ we now have:

$$\boxed{L_r = \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}}$$

Converting a DFA to a regular expression:

Proof cont:

- For non-accepting states $r \notin F$ we now have:

$$L_r = \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}$$

- For accepting states $r \in F$ we will show:

$$\boxed{L_r = \epsilon \cup \left(\bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \right)}$$

Converting a DFA to a regular expression:

Proof cont:

- For non-accepting states $r \notin F$ we now have:

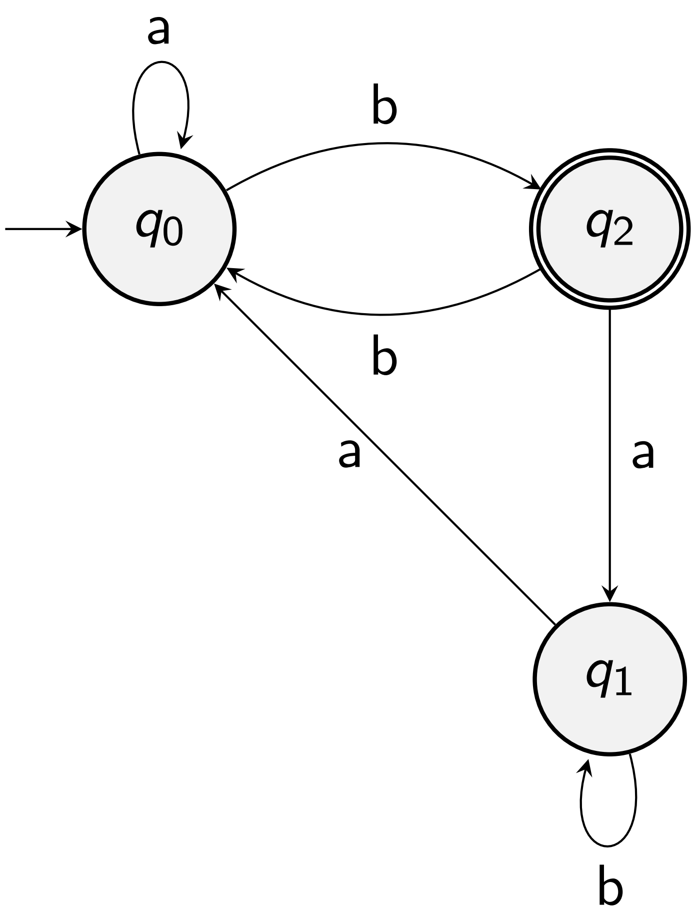
$$\boxed{L_r = \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)}}$$

- For accepting states $r \in F$ we will show:

$$\boxed{L_r = \epsilon \cup \left(\bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \right)}$$

- The proof requires to solve the system of equations above
- We have $|Q|$ equations and $|Q|$ unknowns , one for each state of M
- Before the general proof we will consider an example

Example:



- $M = (Q, \Sigma, \delta, q, F)$
- $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, $q = q_0$, $F = \{q_2\}$
- and

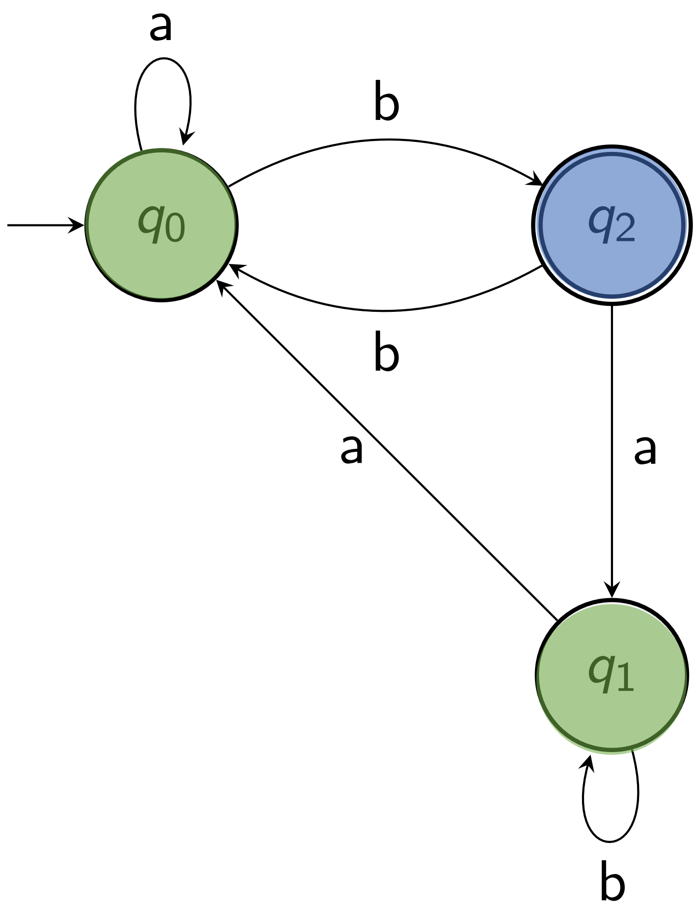
δ	a	b
q_0	q_0	q_2
q_1	q_0	q_1
q_2	q_1	q_0

Converting DFA to a Regular Expression

General form of equations:

$$L_r = \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \text{ if } r \notin F$$

$$L_r = \epsilon \cup \left(\bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \right) \text{ if } r \in F$$



Converting DFA to a Regular Expression

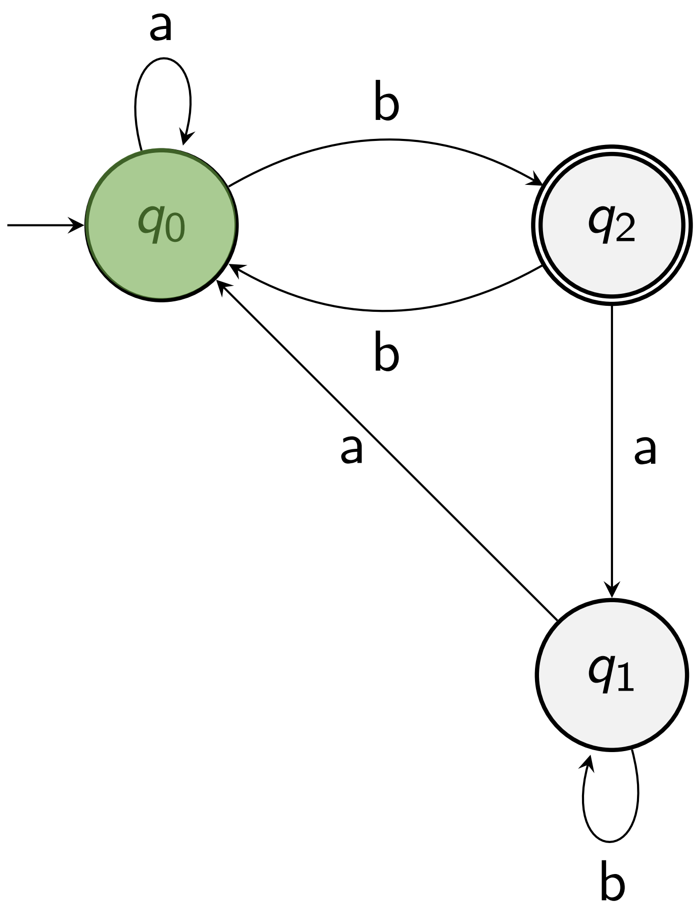
General form of equations:

$$\boxed{L_r = \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \text{ if } r \notin F}$$

$$L_r = \epsilon \cup \left(\bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \right) \text{ if } r \in F$$

For our example we get:

$$L_{q_0} = a \cdot L_{q_0} \cup b \cdot L_{q_2}$$



Converting DFA to a Regular Expression

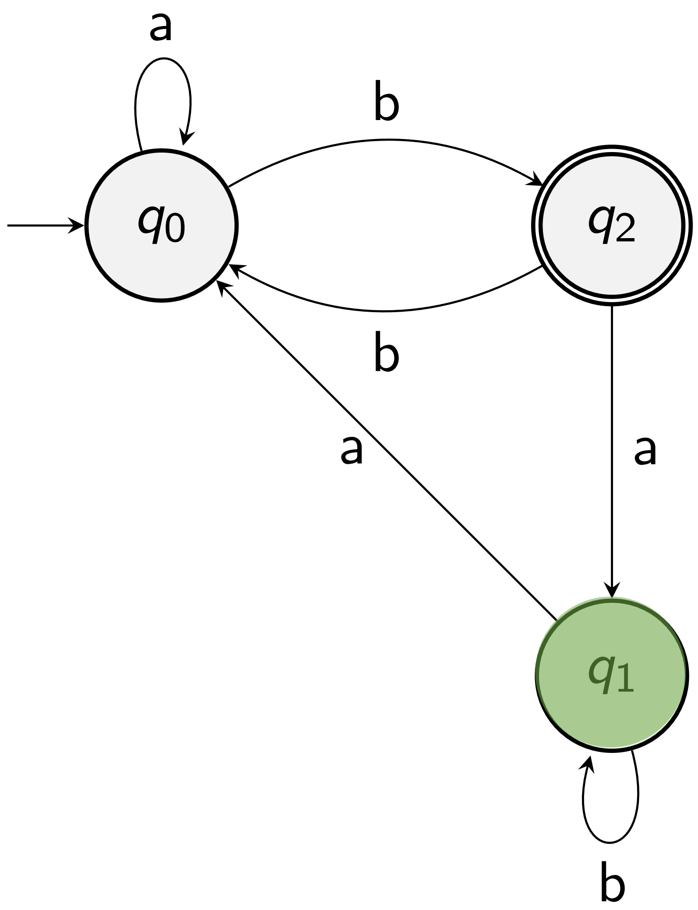
General form of equations:

$$\boxed{L_r = \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \text{ if } r \notin F}$$

$$L_r = \epsilon \cup \left(\bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \right) \text{ if } r \in F$$

For our example we get:

$$L_{q_0} = a \cdot L_{q_0} \cup b \cdot L_{q_2}$$
$$L_{q_1} = a \cdot L_{q_0} \cup b \cdot L_{q_1}$$



Converting DFA to a Regular Expression

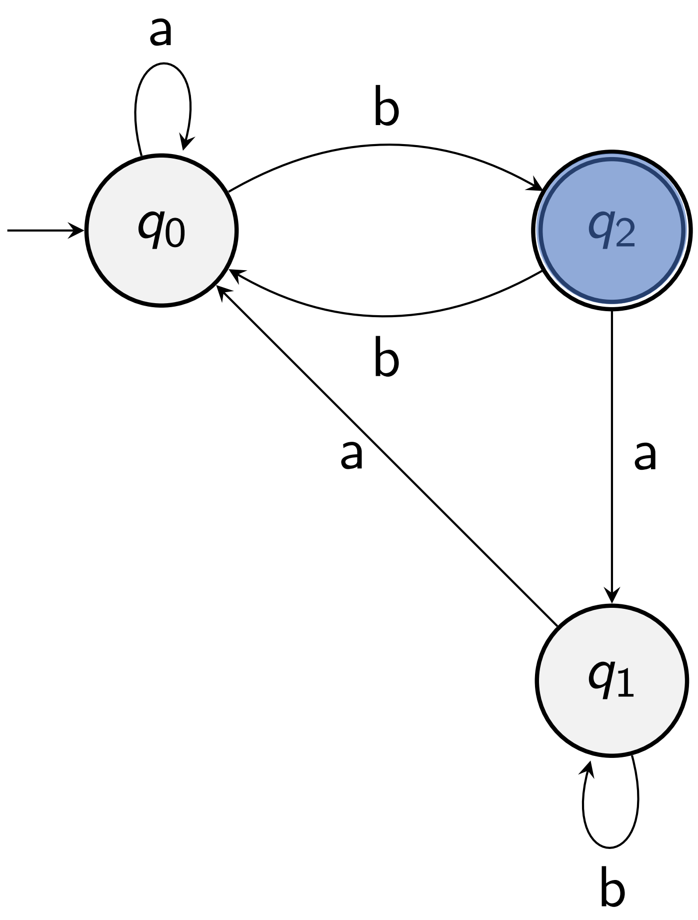
General form of equations:

$$L_r = \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \text{ if } r \notin F$$

$$\boxed{L_r = \epsilon \cup \left(\bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \right) \text{ if } r \in F}$$

For our example we get:

$$L_{q_0} = a \cdot L_{q_0} \cup b \cdot L_{q_2}$$
$$L_{q_1} = a \cdot L_{q_0} \cup b \cdot L_{q_1}$$
$$L_{q_2} = \epsilon \cup a \cdot L_{q_1} \cup b \cdot L_{q_0}$$



Converting DFA to a Regular Expression

We now need to solve:

$$\begin{aligned} L_{q_0} &= a \cdot L_{q_0} \cup b \cdot L_{q_2} \\ L_{q_1} &= a \cdot L_{q_0} \cup b \cdot L_{q_1} \\ L_{q_2} &= \epsilon \cup a \cdot L_{q_1} \cup b \cdot L_{q_0} \end{aligned}$$

Processing L_{q_0}:

$$\begin{aligned} L_{q_0} &= a \cdot L_{q_0} \cup b \cdot L_{q_2} \\ &= a \cdot L_{q_0} \cup b \cdot (\epsilon \cup a \cdot L_{q_1} \cup b \cdot L_{q_0}) \quad \text{(substitution of } L_{q_2}\text{)} \end{aligned}$$

Converting DFA to a Regular Expression

We now need to solve:

$$\begin{aligned} L_{q_0} &= a \cdot L_{q_0} \cup b \cdot L_{q_2} \\ L_{q_1} &= a \cdot L_{q_0} \cup b \cdot L_{q_1} \\ L_{q_2} &= \epsilon \cup a \cdot L_{q_1} \cup b \cdot L_{q_0} \end{aligned}$$

Theorem (Regular Expression Standard Equivalences)

Let R_1, R_2, and R_3 be regular expressions. The following equivalences hold:

1. $R_1\emptyset = \emptyset R_1 = \emptyset$
2. $R_1\epsilon = \epsilon R_1 = R_1$
3. $R_1 \cup \emptyset = \emptyset \cup R_1 = R_1$
4. $R_1 \cup R_1 = R_1$
5. $R_1 \cup R_2 = R_2 \cup R_1$
6. $R_1(R_2 \cup R_3) = R_1R_2 \cup R_1R_3$
7. $(R_1 \cup R_2)R_3 = R_1R_3 \cup R_2R_3$
8. $R_1(R_2R_3) = (R_1R_2)R_3$
9. $\emptyset^* = \epsilon$
10. $\epsilon^* = \epsilon$
11. $(\epsilon \cup R_1)^* = R_1^*$
12. $(\epsilon \cup R_1)(\epsilon \cup R_1)^* = R_1^*$
13. $R_1^*(\epsilon \cup R_1) = (\epsilon \cup R_1)R_1^*$
14. $R_1^*R_2 \cup R_2 = R_1^*R_2$
15. $R_1(R_2R_1)^* = (R_1R_2)^*R_1$
16. $(R_1 \cup R_2)^* = (R_1^*R_2)^*R_1^* = (R_2^*R_1)^*R_2^*$

Processing L_{q_0}:

$$\begin{aligned} L_{q_0} &= a \cdot L_{q_0} \cup b \cdot L_{q_2} \\ &= a \cdot L_{q_0} \cup b \cdot (\epsilon \cup a \cdot L_{q_1} \cup b \cdot L_{q_0}) && \text{(substitution of } L_{q_2}\text{)} \\ &= a \cdot L_{q_0} \cup b\epsilon \cup ba \cdot L_{q_1} \cup bb \cdot L_{q_0} && \text{(standard equivalence 6)} \end{aligned}$$

Converting DFA to a Regular Expression

We now need to solve:

$$
\begin{aligned}
L_{q_0} &= a \cdot L_{q_0} \cup b \cdot L_{q_2} \\
L_{q_1} &= a \cdot L_{q_0} \cup b \cdot L_{q_1} \\
L_{q_2} &= \epsilon \cup a \cdot L_{q_1} \cup b \cdot L_{q_0}
\end{aligned}
$$

Theorem (Regular Expression Standard Equivalences)

Let R_1, R_2, and R_3 be regular expressions. The following equivalences hold:

1. $R_1\emptyset = \emptyset R_1 = \emptyset$
2. $R_1\epsilon = \epsilon R_1 = R_1$
3. $R_1 \cup \emptyset = \emptyset \cup R_1 = R_1$
4. $R_1 \cup R_1 = R_1$
5. $R_1 \cup R_2 = R_2 \cup R_1$
6. $R_1(R_2 \cup R_3) = R_1R_2 \cup R_1R_3$
7. $(R_1 \cup R_2)R_3 = R_1R_3 \cup R_2R_3$
8. $R_1(R_2R_3) = (R_1R_2)R_3$
9. $\emptyset^* = \epsilon$
10. $\epsilon^* = \epsilon$
11. $(\epsilon \cup R_1)^* = R_1^*$
12. $(\epsilon \cup R_1)(\epsilon \cup R_1)^* = R_1^*$
13. $R_1^*(\epsilon \cup R_1) = (\epsilon \cup R_1)R_1^*$
14. $R_1^*R_2 \cup R_2 = R_1^*R_2$
15. $R_1(R_2R_1)^* = (R_1R_2)^*R_1$
16. $(R_1 \cup R_2)^* = (R_1^*R_2)^*R_1^* = (R_2^*R_1)^*R_2^*$

Processing L_{q_0}:

$$
\begin{aligned}
L_{q_0} &= a \cdot L_{q_0} \cup b \cdot L_{q_2} \\
&= a \cdot L_{q_0} \cup b \cdot (\epsilon \cup a \cdot L_{q_1} \cup b \cdot L_{q_0}) && \text{(substitution of } L_{q_2}\text{)} \\
&= \boxed{a \cdot L_{q_0}} \cup \boxed{b\epsilon} \cup ba \cdot L_{q_1} \cup \boxed{bb \cdot L_{q_0}} && \text{(standard equivalence 6)} \\
&= \boxed{(a \cup bb) \cdot L_{q_0}} \cup \boxed{b} \cup ba \cdot L_{q_1} && \text{(standard equivalences 2,5,6)}
\end{aligned}
$$

Converting DFA to a Regular Expression

Remaining system to be solved:

$$L_{q_0} = (a \cup bb) \cdot L_{q_0} \cup b \cup ba \cdot L_{q_1}$$
$$L_{q_1} = b \cdot L_{q_1} \cup a \cdot L_{q_0}$$

Converting DFA to a Regular Expression

Remaining system to be solved:

$$L_{q_0} = (a \cup bb) \cdot L_{q_0} \cup b \cup ba \cdot L_{q_1}$$
$$L_{q_1} = b \cdot L_{q_1} \cup a \cdot L_{q_0}$$

We can utilize our new lemma:

Lemma

$$L = BL \cup C \;\Rightarrow\; L = B^* C$$

Choose L, B, C as follows:

$$\underbrace{L_{q_1}}_{L} = \underbrace{b}_{B} \cdot \underbrace{L_{q_1}}_{L} \cup \underbrace{a \cdot L_{q_0}}_{C}$$

Converting DFA to a Regular Expression

Remaining system to be solved:

$$\begin{aligned} L_{q_0} =& (a \cup bb) \cdot L_{q_0} \cup b \cup ba \cdot L_{q_1} \\ L_{q_1} =& b \cdot L_{q_1} \cup a \cdot L_{q_0} \end{aligned}$$

We can utilize our new lemma:

Lemma

$$L = BL \cup C \;\Rightarrow\; L = B^*C$$

Choose L, B, C as follows:

$$\underbrace{L_{q_1}}_{L} = \underbrace{b}_{B} \cdot \underbrace{L_{q_1}}_{L} \cup \underbrace{a \cdot L_{q_0}}_{C}$$

We get:

$$\underbrace{L_{q_1}}_{L} = \underbrace{b^*}_{B^*} \underbrace{a \cdot L_{q_0}}_{C}$$

Converting DFA to a Regular Expression

Remaining system to be solved:

$$
\begin{aligned}
L_{q_0} &= (a \cup bb) \cdot L_{q_0} \cup b \cup ba \cdot L_{q_1} \\
L_{q_1} &= b \cdot L_{q_1} \cup a \cdot L_{q_0}
\end{aligned}
$$

We can utilize our new lemma:

Lemma

$$L = BL \cup C \;\Rightarrow\; L = B^* C$$

Choose L, B, C as follows:

$$\underbrace{L_{q_1}}_{L} = \underbrace{b}_{B} \cdot \underbrace{L_{q_1}}_{L} \cup \underbrace{a \cdot L_{q_0}}_{C}$$

We get:

$$\underbrace{L_{q_1}}_{L} = \boxed{\underbrace{b^*}_{B^*} \underbrace{a \cdot L_{q_0}}_{C}}$$

Result can now be substituted back into the equation for L_{q_0}

Converting DFA to a Regular Expression

Equation after substitution:

$$
\begin{aligned}
L_{q_0} &= \boxed{(a \cup bb) \cdot L_{q_0}} \cup b \cup \boxed{ba \cdot \overbrace{b^* a \cdot L_{q_0}}^{L_{q_1}}} \\
&= (a \cup bb \cup bab^* a) \cdot L_{q_0} \cup b \qquad \text{(standard equivalence 6)}
\end{aligned}
$$

Converting DFA to a Regular Expression

Equation after substitution:

$$\begin{aligned} L_{q_0} &= (a \cup bb) \cdot L_{q_0} \cup b \cup ba \cdot \overbrace{b^* a \cdot L_{q_0}}^{L_{q_1}} \\ &= (a \cup bb \cup bab^* a) \cdot L_{q_0} \cup b \qquad \text{(standard equivalence 6)} \end{aligned}$$

Utilization of the lemma again:

$$\underbrace{L_{q_0}}_{L} = \underbrace{(a \cup bb \cup bab^* a)}_{B} \cdot \underbrace{L_{q_0}}_{L} \cup \underbrace{b}_{C}$$

Lemma

Let B, C, and L be languages over an alphabet Σ such that

$$\epsilon \notin B \quad \text{and} \quad L = BL \cup C$$

then

$$L = B^* C$$

Converting DFA to a Regular Expression

Equation after substitution:

$$
\begin{aligned}
L_{q_0} &= (a \cup bb) \cdot L_{q_0} \cup b \cup ba \cdot \overbrace{b^* a \cdot L_{q_0}}^{L_{q_1}} \\
&= (a \cup bb \cup bab^* a) \cdot L_{q_0} \cup b \qquad \text{(standard equivalence 6)}
\end{aligned}
$$

Utilization of the lemma again:

$$
\underbrace{L_{q_0}}_{L} = \underbrace{(a \cup bb \cup bab^* a)}_{B} \cdot \underbrace{L_{q_0}}_{L} \cup \underbrace{b}_{C}
$$

Lemma

Let B, C, and L be languages over an alphabet Σ such that

$$\epsilon \notin B \quad \text{and} \quad L = BL \cup C$$

then

$$L = B^* C$$

We get

$$
L_{q_0} = L = B^* C = (a \cup bb \cup bab^* a)^* b
$$

which is the regular expression that describes the language of M.

Converting DFA to a Regular Expression

$$L(M) \;=\; (a \cup bb \cup bab^*a)^*b$$