Regular expressions

R	$L(R)$
ε	$\{\varepsilon\}$
$\emptyset$	$\emptyset$
a	$\{\mathsf{a}\}$ (for any $\mathsf{a} \in \Sigma$)
$R_1 + R_2$	$L(R_1) \cup L(R_2)$
$R_1 R_2$	$\{xy \mid x \in L(R_1), y \in L(R_2)\}$
R_1^*	$\{x_1 \ldots x_n \mid n \in \mathbb{N}_0, x_1, \ldots, x_n \in L(R_1)\}$.

Closure under union and concatenation

Theorem: If L_1 and L_2 are regular languages, then the following two languages are also regular:

1. $L_1 \cup L_2$,
2. $L_1 L_2 = \{xy \mid x \in L_1, y \in L_2\}$.

Proof: Let R_1, R_2 be regular expressions such that $L(R_1) = L_1, L(R_2) = L_2$. Then

$$L_1 \cup L_2 = L(R_1) \cup L(R_2) = L(R_1 + R_2)$$

and

$$L_1 L_2 = L(R_1)L(R_2) = L(R_1 R_2).$$

Closure under complementation

Theorem: If $L \subseteq \Sigma^*$ is a regular language, then the following language is also regular:

$$L^{\complement} = \{x \in \Sigma^* \mid x \notin L\}.$$

Proof: Let M be a DFA that recognises L. Let M' be the DFA obtained from M by making all states that are not final states of M final states of M' and vice versa. Then M' recognises $L^{\complement}$:

M' accepts string x

$\iff$ the computation of M' on input x ends in a final state of M'

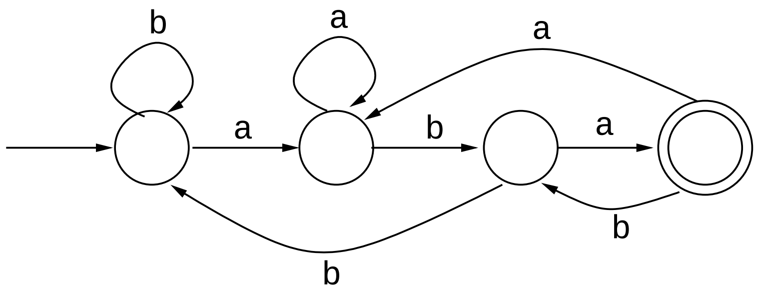
$\iff$ the computation of M' on input x does not end in a final state of M

$\iff$ the computation of M on input x does not end in a final state of M

$\iff$ M does not accept x

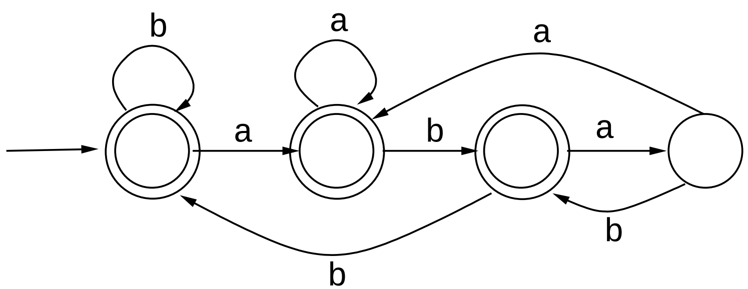
$\iff$ $x \in L^{\complement}$.

Example



A DFA for the language

$L = \{x\mathtt{aba} \mid x \in \{\mathtt{a}, \mathtt{b}\}^*\}$.



A DFA for $L^{\complement}$.

Closure under intersection and difference

Theorem: If L_1 and L_2 are regular languages, then the following two languages are also regular:

1. $L_1 \cap L_2$,
2. $L_1 \setminus L_2 = \{x \mid x \in L_1 \text{ and } x \notin L_2\}$.

Proof: We use the facts that

$$L_1 \cap L_2 = \left(L_1^{\complement} \cup L_2^{\complement}\right)^{\complement} \qquad \text{(by DeMorgan's rule)}$$

and

$$L_1 \setminus L_2 = L_1 \cap L_2^{\complement} = \left(L_1^{\complement} \cup L_2\right)^{\complement}.$$

Constructions

Problem: Suppose we are given DFAs for L_1 and L_2.

How do we construct DFAs for

$$L_1^{\complement}, \quad L_1 \cup L_2, \quad L_1 \cap L_2, \quad L_1 \setminus L_2 \quad ?$$

Complementation

We have already seen a construction.

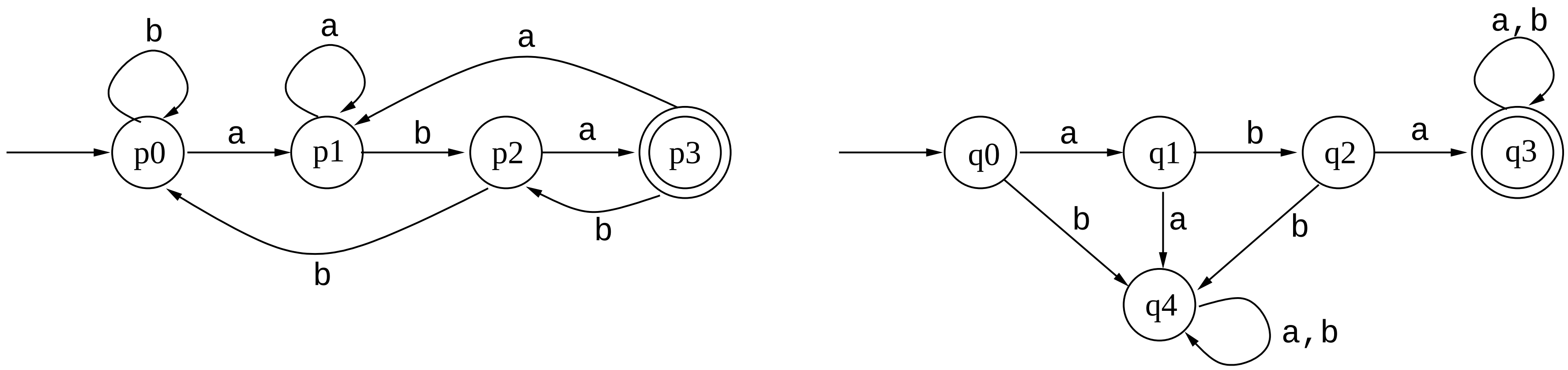
Union

1. Compute an NFA with ε-transitions.
2. Convert it to a DFA.

Intersection and difference

Combine the constructions for complementation and union.

Example



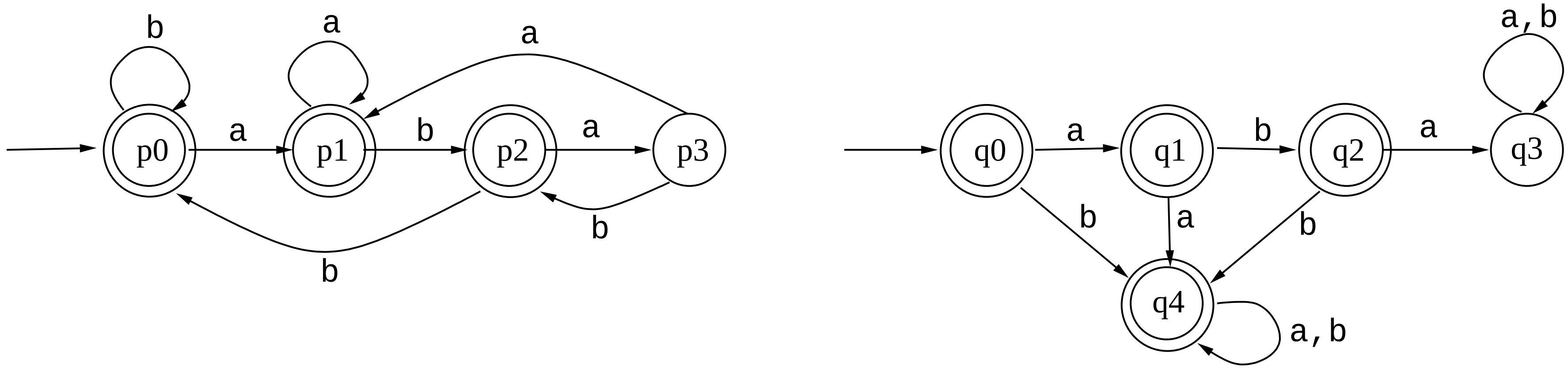
M_1 M_2

Construct a DFA for

$$L(M_1) \cap L(M_2) = \left(L(M_1)^{\complement} \cup L(M_2)^{\complement}\right)^{\complement}.$$

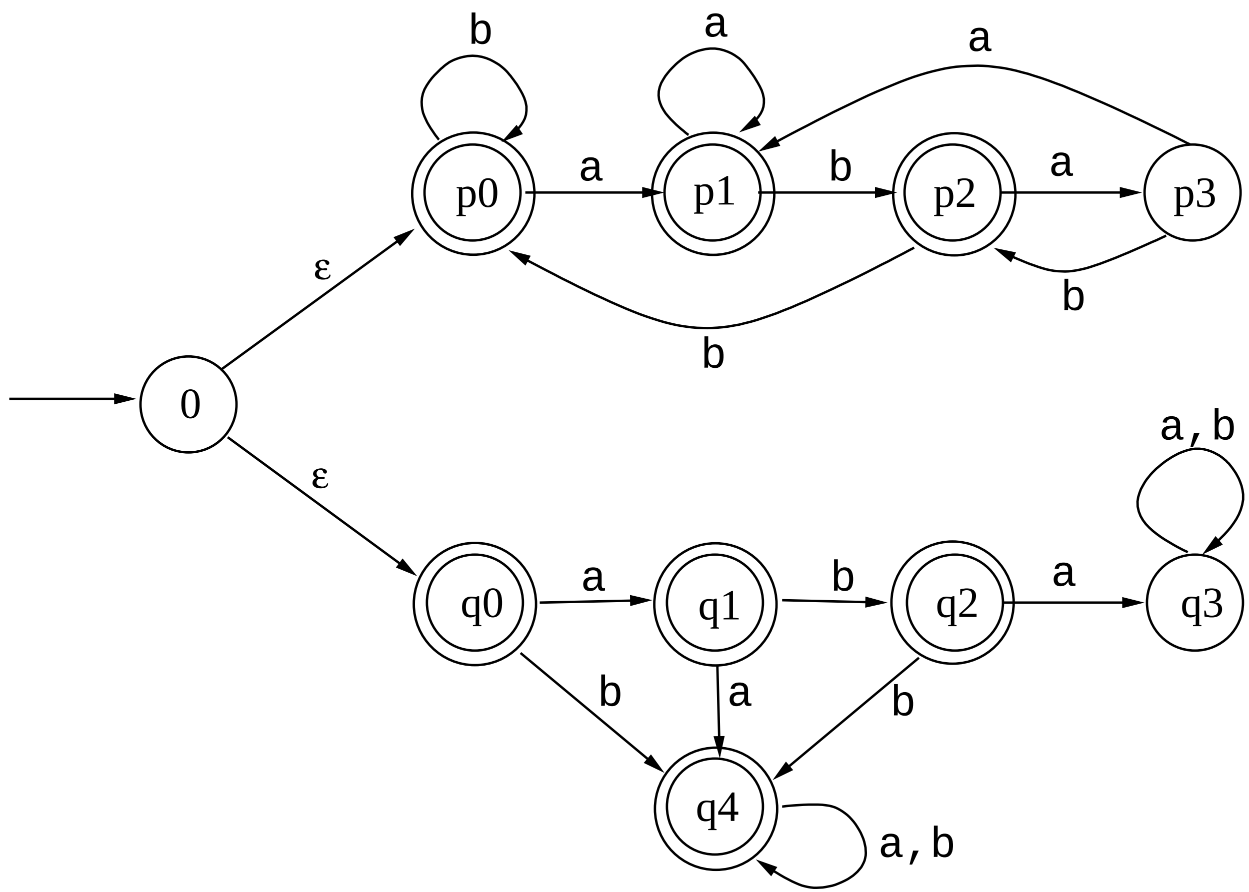
Example (cont'd)

Step 1: Construct DFAs for $L(M_1)^{\complement}$ and $L(M_2)^{\complement}$.



Example (cont'd)

Step 2: Construct an NFA with ε-transitions for $L(M_1)^{\complement} \cup L(M_2)^{\complement}$.



Example (cont'd)

Step 3: Convert the NFA with ε-transitions for $L(M_1)^{\complement} \cup L(M_2)^{\complement}$ to a DFA.

δ	a	b
$\{0, p0, q0\}$	$\{p1, q1\}$	$\{p0, q4\}$
$\{p1, q1\}$	$\{p1, q4\}$	$\{p2, q2\}$
$\{p0, q4\}$	$\{p1, q4\}$	$\{p0, q4\}$
$\{p1, q4\}$	$\{p1, q4\}$	$\{p2, q4\}$
$\{p2, q2\}$	$\{p3, q3\}$	$\{p0, q4\}$
$\{p2, q4\}$	$\{p3, q4\}$	$\{p0, q4\}$
$\{p3, q3\}$	$\{p1, q3\}$	$\{p2, q3\}$
$\{p3, q4\}$	$\{p1, q4\}$	$\{p2, q4\}$
$\{p1, q3\}$	$\{p1, q3\}$	$\{p2, q3\}$
$\{p2, q3\}$	$\{p3, q3\}$	$\{p0, q3\}$
$\{p0, q3\}$	$\{p1, q3\}$	$\{p0, q3\}$

Example (cont'd)

Step 4: Complement the DFA for $L(M_1)^{\complement} \cup L(M_2)^{\complement}$.

Result:

$$\Big(Q, \Sigma, \{0, p0, q0\}, \big\{\{p3, q3\}\big\}, \delta\Big),$$

where

$$\begin{aligned} Q = \big\{&\{0, p0, q0\}, \{p1, q1\}, \{p0, q4\}, \{p1, q4\}, \{p2, q2\}, \{p2, q4\}, \\ &\{p3, q3\}, \{p3, q4\}, \{p1, q3\}, \{p2, q3\}, \{p0, q3\}\big\}, \end{aligned}$$

Union vs. Intersection vs. Difference

The automata we construct for $L(M_1) \cap L(M_2)$, $L(M_1) \cup L(M_2)$, and $L(M_1) \setminus L(M_2)$ only differ in their final states:

Intersection: Final states (of the powerset automaton) are states that contain a final state of M_1 **and** a final state of M_2.

Union: Final states (of the powerset automaton) are states that contain a final state of M_1 **or** a final state of M_2.

Difference: Final states (of the powerset automaton) are states that contain a final state of M_1 **but not** a final state of M_2.

Minimal DFAs

For every DFA M there exists a DFA M' such that

- $L(M) = L(M')$.

- M' has fewer states then any other automaton M'' with $L(M'') = L(M)$.

  Thus M' is an optimal, **minimal** DFA for the language $L(M)$.

- M' is unique up to re-naming states.

- M' can be computed by an efficient algorithm.

No analogous result holds for NFAs.

Equivalence testing

Theorem: $L(M) = \emptyset$ iff there is no path in M from the initial state to any final state.

Lemma: $L_1 = L_2$ iff $L_1 \setminus L_2 = \emptyset$ and $L_2 \setminus L_1 = \emptyset$.

Theorem: There is an efficient algorithm that decides whether $L(M_1) = L(M_2)$.

Proof: Construct DFAs for $L_1 \setminus L_2$ and $L_2 \setminus L_1$, and test if both have a path from the initial state to a final state.

Equations for regular expressions

We can manipulate regular expressions using the following identities.

1. $R + R = R = R + \emptyset$
2. $R + S = S + R$
3. $(R + S) + T = R + (S + T)$
4. $(RS)T = R(ST) = RST$
5. $R\varepsilon = \varepsilon R = R$
6. $R\emptyset = \emptyset R = \emptyset$
7. $(R + S)T = RT + ST$
8. $R(S + T) = RS + RT$
9. $R^*R^* = (R^*)^* = R^* = RR^* + \varepsilon$
10. $RR^* = R^*R$
11. $\varepsilon^* = \emptyset^* = \varepsilon$
12. $(R + S)^* = (R^*S^*)^*$
 $= (R^*S)^*R^* = (R^* + S^*)^*$
13. $(RS)^*R = R(SR)^*$

The regular expressions on either side of the "$=$" signs represent the same regular language. These identities are **schematic**, i.e., the names R, S and T stand for any regular expressions.

Example

$$
\begin{aligned}
0(10)^*1 + (01)^* &= (01)(01)^* + (01)^* && \text{(13 with } R = 1 \text{ and } S = 0.\text{)} \\
&= (01)(01)^* + (01)(01)^* + \varepsilon && \text{(9 with } R = (01).\text{)} \\
&= (01)(01)^* + \varepsilon && \text{(1 with } R = (01)(01)^*.\text{)} \\
&= (01)^* && \text{(9 with } R = (01) \text{ again.)}
\end{aligned}
$$

Justification of the equations

Example: $(RS)^*R = R(SR)^*$

$$w \in L((RS)^*R)$$

$$\iff \exists v \in L((RS)^*), z \in L(R) : \; w = vz$$

$$\iff \exists n \in \mathbb{N}_0, v_1, \ldots, v_n \in L(RS), z \in L(R) : \; w = v_1 v_2 \ldots v_n z$$

$$\iff \exists n \in \mathbb{N}_0, x_1, \ldots, x_n \in L(R), y_1, \ldots, y_n \in L(S), z \in L(R) :$$

$$w = x_1 y_1 x_2 y_2 \ldots x_n y_n \; z$$

$$\iff \exists x \in L(R), n \in \mathbb{N}_0, y'_1, \ldots, y'_n \in L(S), z_1, \ldots, z_n \in L(R) :$$

$$w = x \; y_1 z_1 y_2 z_2 \ldots y_n z_n$$

$$\iff \exists x \in L(R), \in \mathbb{N}_0, u_1, \ldots, u_n \in L(SR) : \; w = x u_1 \ldots u_n$$

$$\iff \exists x \in L(R), u \in L((SR)^*) : \; w = xu$$

$$\iff w \in L(R(SR)^*).$$