COS210 - Theoretical Computer Science
Finite Automata and Regular Languages (Part 2)

Standard and Extended Transition Function

DFA $M = (Q, \Sigma, \delta, q, F)$, state $r \in Q$, symbol $a \in \Sigma$, string $w = w_1 ... w_n$

Standard transition function $\delta : Q \times \Sigma \to Q$:

- $\delta(r, a) = r'$ with $r \xrightarrow{\text{a}} r'$
- $\delta(r, a)$ is the state that is reached from r by taking the a-labelled transition

Standard and Extended Transition Function

DFA $M = (Q, \Sigma, \delta, q, F)$, state $r \in Q$, symbol $a \in \Sigma$, string $w = w_1 ... w_n$

Standard transition function $\delta : Q \times \Sigma \to Q$:

- $\delta(r, a) = r'$ with $r \xrightarrow{\text{a}} r'$
- $\delta(r, a)$ is the state that is reached from r by taking the a-labelled transition

Extended transition function $\bar{\delta} : Q \times \Sigma^* \to Q$:

- Σ^* is the set of all strings over Σ, including the empty string ϵ

Example:
Σ = {0, 1}
Σ* = {ε, 0, 1, 00, 01, ...}

Standard and Extended Transition Function

DFA $M = (Q, \Sigma, \delta, q, F)$, state $r \in Q$, symbol $a \in \Sigma$, string $w = w_1 ... w_n$

Standard transition function $\delta : Q \times \Sigma \rightarrow Q$:

- $\delta(r, a) = r'$ with $r \xrightarrow{\text{a}} r'$
- $\delta(r, a)$ is the state that is reached from r by taking the a-labelled transition

Extended transition function $\bar{\delta} : Q \times \Sigma^* \rightarrow Q$:

- Σ^* is the set of all strings over Σ, including the empty string ϵ
- $\bar{\delta}(r, w_1 \ldots w_n) = r'$ with $r \xrightarrow{w_1} \ldots \xrightarrow{w_n} r'$
- $\bar{\delta}(r, w_1 \ldots w_n)$ is the state that is reached from r by sequentially taking the w_1-, $\ldots w_n$-labelled transitions
- $\bar{\delta}$ is based on a recursive application of δ

Example:
Σ = {0, 1}
Σ* = {ε, 0, 1, 00, 01, …}

Standard and Extended Transition Function

DFA $M = (Q, \Sigma, \delta, q, F)$, state $r \in Q$, symbol $a \in \Sigma$, string $w = w_1 ... w_n$

Standard transition function $\delta : Q \times \Sigma \to Q$:

- $\delta(r, a) = r'$ with $r \xrightarrow{\text{a}} r'$
- $\delta(r, a)$ is the state that is reached from r by taking the a-labelled transition

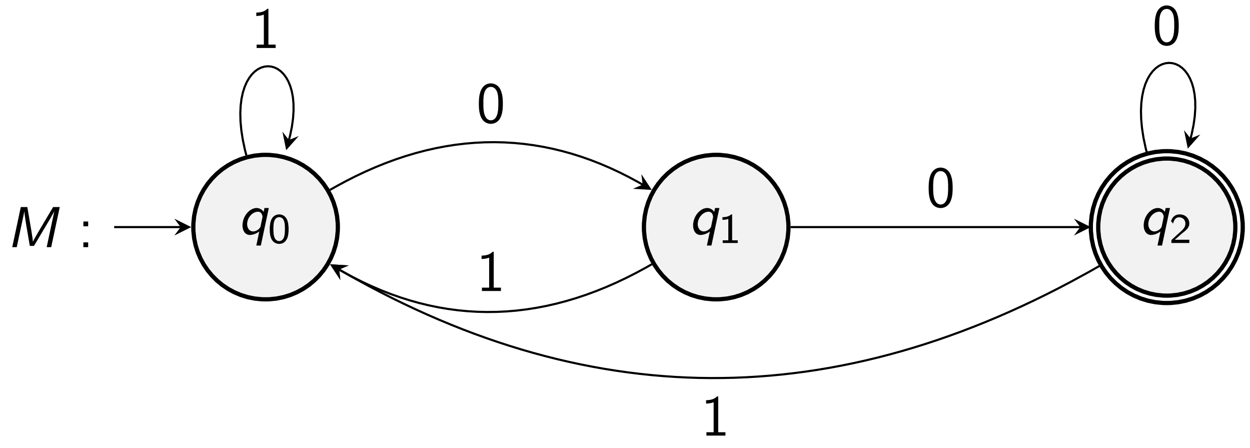
Extended transition function $\bar{\delta} : Q \times \Sigma^* \to Q$:

- Σ^* is the set of all strings over Σ, including the empty string ϵ
- $\bar{\delta}(r, w_1 \dots w_n) = r'$ with $r \xrightarrow{w_1} \dots \xrightarrow{w_n} r'$
- $\bar{\delta}(r, w_1 \dots w_n)$ is the state that is reached from r by sequentially taking the w_1-, $\dots w_n$-labelled transitions
- $\bar{\delta}$ is based on a recursive application of δ

Example:
Σ = {0, 1}
Σ* = {ε, 0, 1, 00, 01, …}

Acceptance of a string w can now be simply defined as $\bar{\delta}(q, w) \in F$

Extended Transition Function: Example



Is $w = 100$ accepted by M?

- $\delta(q_0, 1) = q_0, \;\; \delta(q_0, 0) = q_1, \;\; \delta(q_1, 0) = q_2 \in F$
- $\bar{\delta}(q_0, 100) = q_2 \in F$

Operations on Regular Languages

The following operations can be applied to languages:

- The **union** of languages A and B:

$$A \cup B = \{w : w \in A \text{ or } w \in B\}$$

Example:

$$A = \{work\},\ \ B = \{sheet, space\} \quad \rightarrow \quad A \cup B = \{work, sheet, space\}$$

Operations on Regular Languages

The following operations can be applied to languages:

- The **union** of languages A and B:

$$A \cup B = \{w : w \in A \text{ or } w \in B\}$$

Example:

$$A = \{work\},\ \ B = \{sheet, space\} \quad \rightarrow \quad A \cup B = \{work, sheet, space\}$$

- The **concatenation** of A and B:

$$AB = \{ww' : w \in A \text{ and } w' \in B\}$$

Example:

$$A = \{work\},\ \ B = \{sheet, space\} \quad \rightarrow \quad AB = \{worksheet, workspace\}$$

Operations on Regular Languages

- The **star** of language A:

$$A^* = \{u_1 u_2 \cdots u_k : k \geq 0 \text{ and } u_i \in A \text{ for all } i = 1, 2, \cdots, k\}$$

all possible combinations of strings from A glued together

Example:

$$A = \{a, b\} \quad \rightarrow \quad A^* = \{\epsilon, a, b, aa, bb, ab, ba, aba, abba \ldots\}$$

Alternative Definition of the Star Operation

Definition

Let A be a language, A^k, $k \geq 0$, and A^* are defined as

$$\begin{aligned} A^0 &= \{\epsilon\} \\ A^1 &= A \\ A^2 &= AA \\ A^3 &= AAA \\ &\vdots \\ A^k &= \underbrace{A \ldots A}_{k \text{ times}} \end{aligned}$$

and

$$A^* = A^0 \cup A^1 \cup \ldots$$

Closure of Operations

In many practical cases **closure** is an important property when it comes to the application of operations.

Closure is when an operation on elements of a set X results in an element of the same set X.

Example:

- $1 \in \mathbb{N},\ 2 \in \mathbb{N}$
- $1 + 2 \ = \ 3 \in \mathbb{N}$ ($\mathbb{N}$ closed under addition)
- $1/2 \ = \ 0.5 \notin \mathbb{N}$ ($\mathbb{N}$ not closed under division)

Is the set of regular languages closed under Union, Concatenation, Star?

Closure of Operations on Languages

Let R be the set of all regular languages over some alphabet Σ:

Definition

R is closed under an binary operator *bop* if for all $A, B \in R$

$$bop(A, B) \in R$$

i.e. if $bop(A, B)$ is regular.

Definition

R is closed under an unary operator *uop* if for all $A \in R$

$$uop(A) \in R$$

i.e. if $uop(A)$ is regular.

Closure of the Union Operation

Theorem (Closure of Union)

The set of regular languages R over Σ is closed under the Union *operation.*

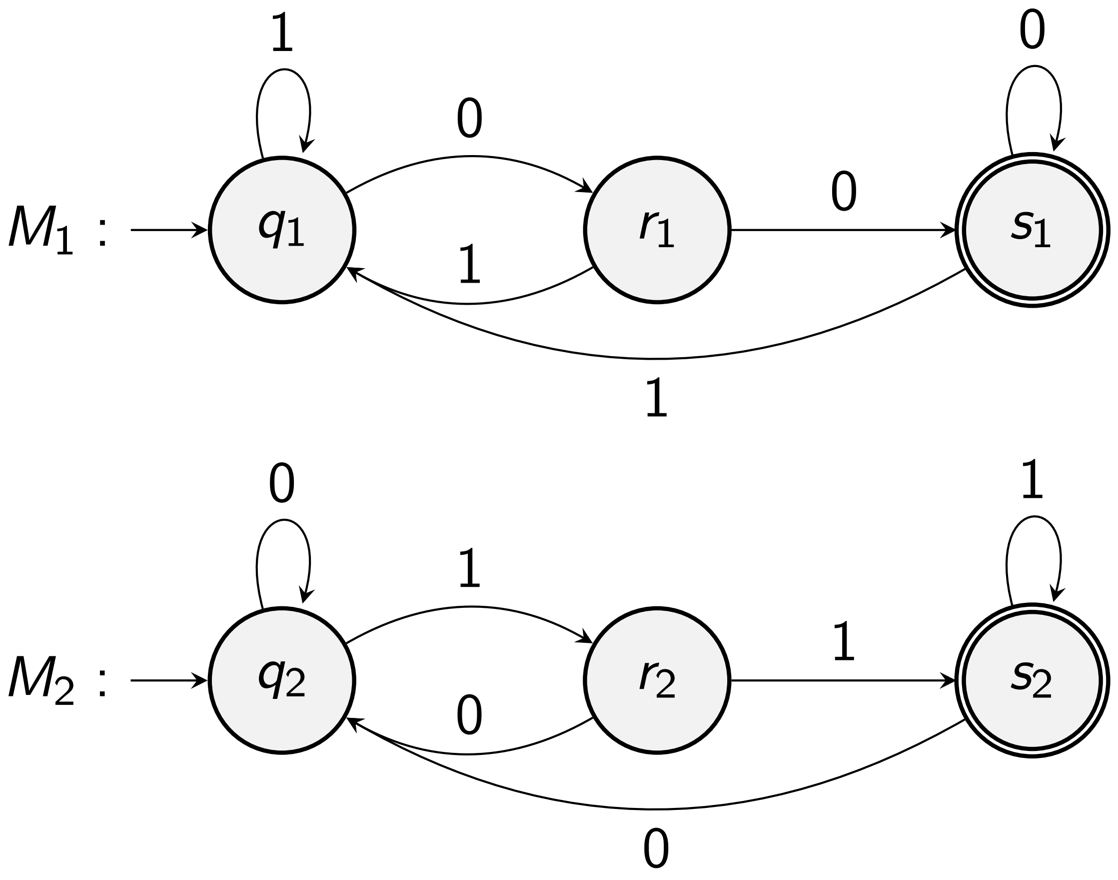
Proof Idea:

- we show that for any two regular languages A and B the language $A \cup B$ is regular as well
- since A and B are regular there exist automata $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ that accept A and B respectively
- to prove $A \cup B$ regular we construct automaton M that accepts $A \cup B$
- we therefore show that for all strings $w \in \Sigma^*$:

$$\underbrace{M_1 \text{ accepts } w}_{A} \quad \text{OR} \quad \underbrace{M_2 \text{ accepts } w}_{B} \iff \underbrace{M \text{ accepts } w}_{A \cup B}$$

Example: How to construct M based on M_1 and M_2

We let M_1 and M_2 run simultaneously with the same input w



$L(M_1) = \{w : \text{ends with } 00\}$

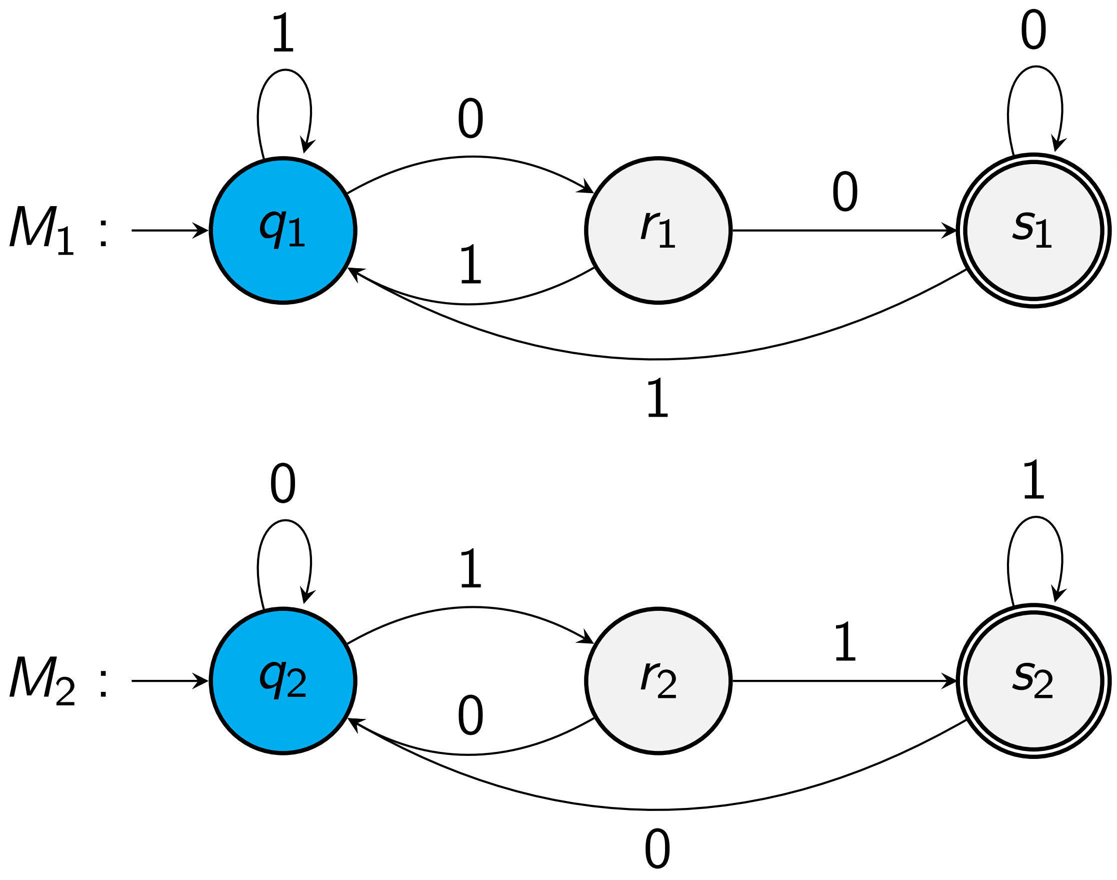
$L(M_2) = \{w : \text{ends with } 11\}$

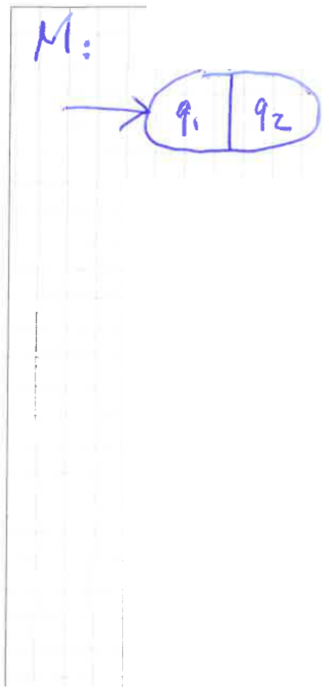
As states of M we use combined states (t_1, t_2) where $t_1 \in Q_1$ and $t_2 \in Q_2$

M to be constructed shall be in accept state if M_1 or M_2 is in accept state

Example: How to construct M based on M_1 and M_2

Run M_1 and M_2 simultaneously with $\mathbf{w} = \mathbf{1011}$

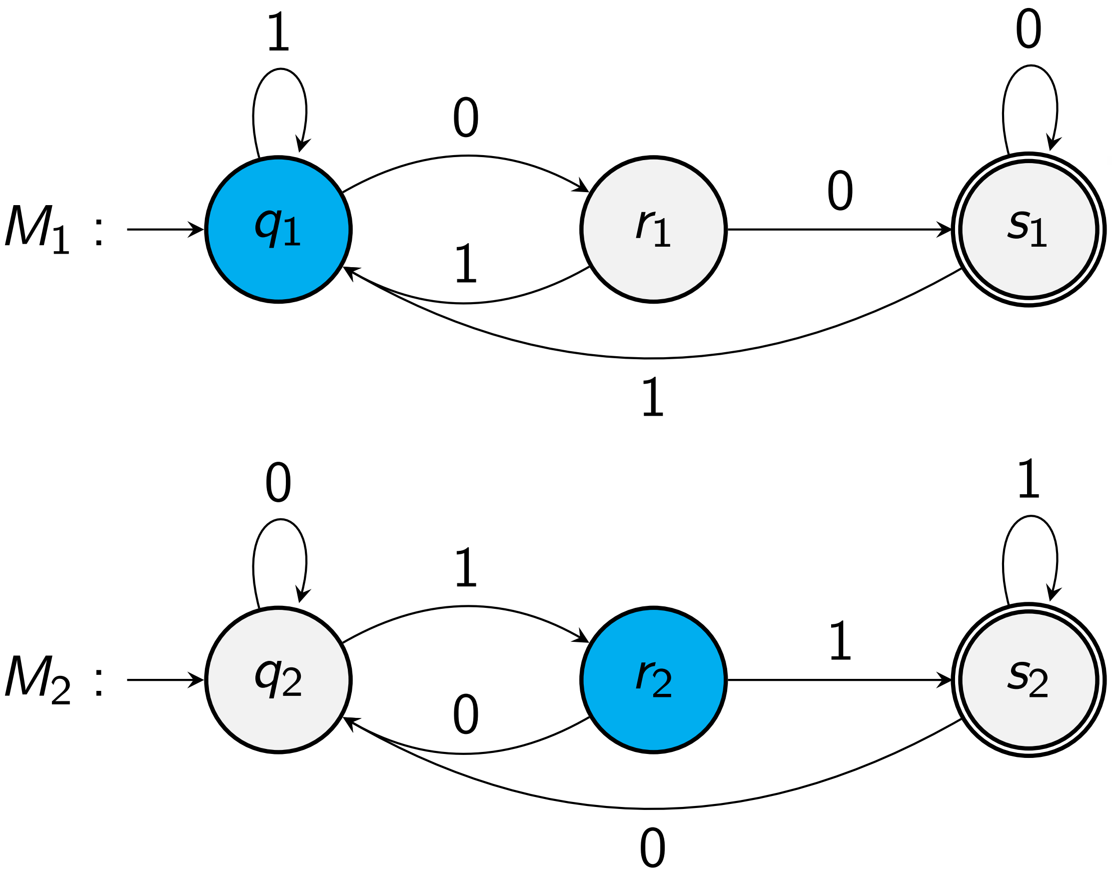


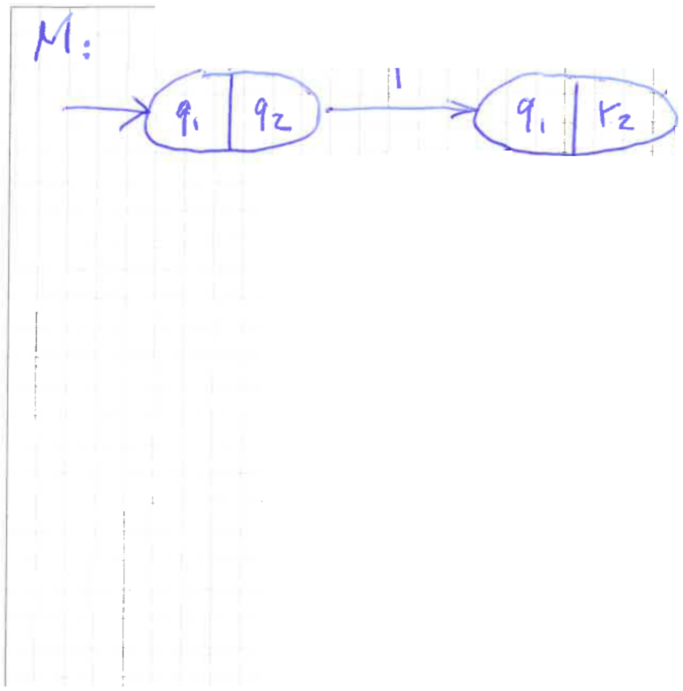


current combined state of M_1 and M_2 is (q_1, q_2),
which shall be the current state of M

Example: How to construct M based on M_1 and M_2

Run M_1 and M_2 simultaneously with $\mathbf{w} = \boxed{\mathbf{1}}\mathbf{011}$

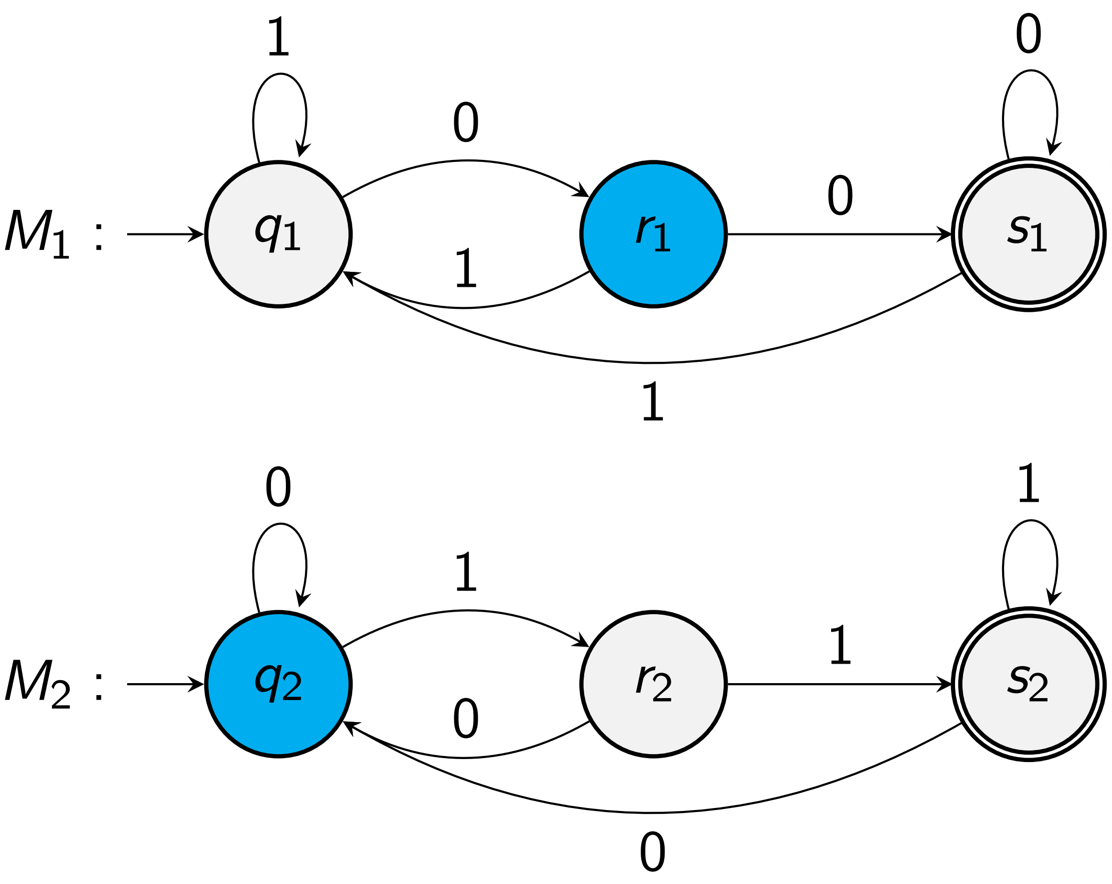


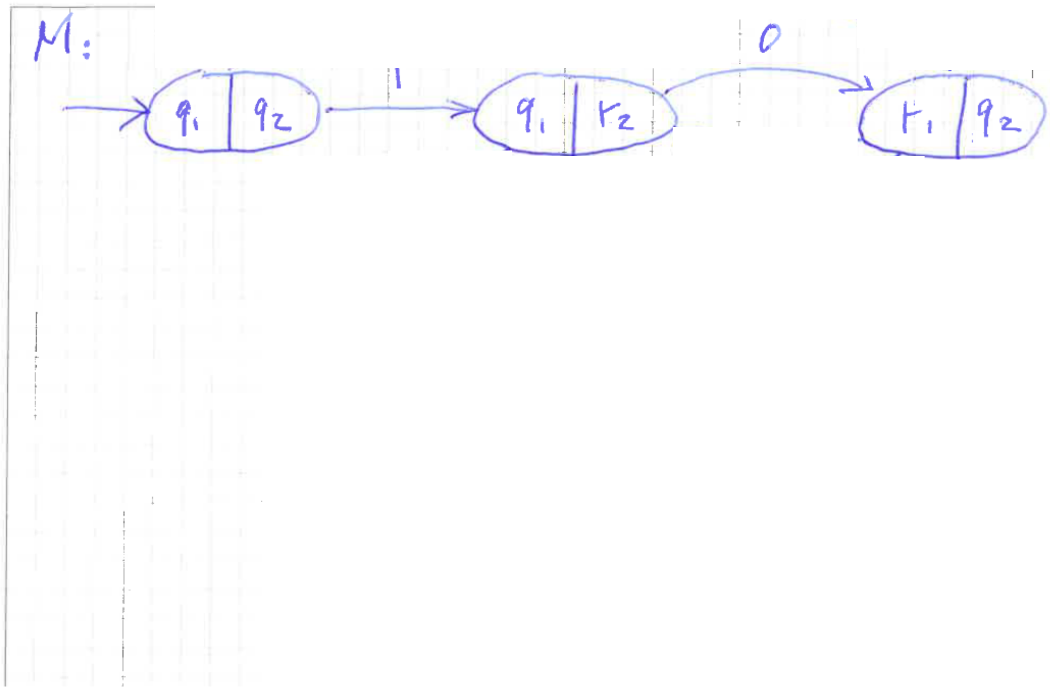


current combined state of M_1 and M_2 is (q_1, r_2),
which shall be the current state of M

Example: How to construct M based on M_1 and M_2

Run M_1 and M_2 simultaneously with $\mathbf{w} = \mathbf{1}\boxed{\mathbf{0}}\mathbf{11}$

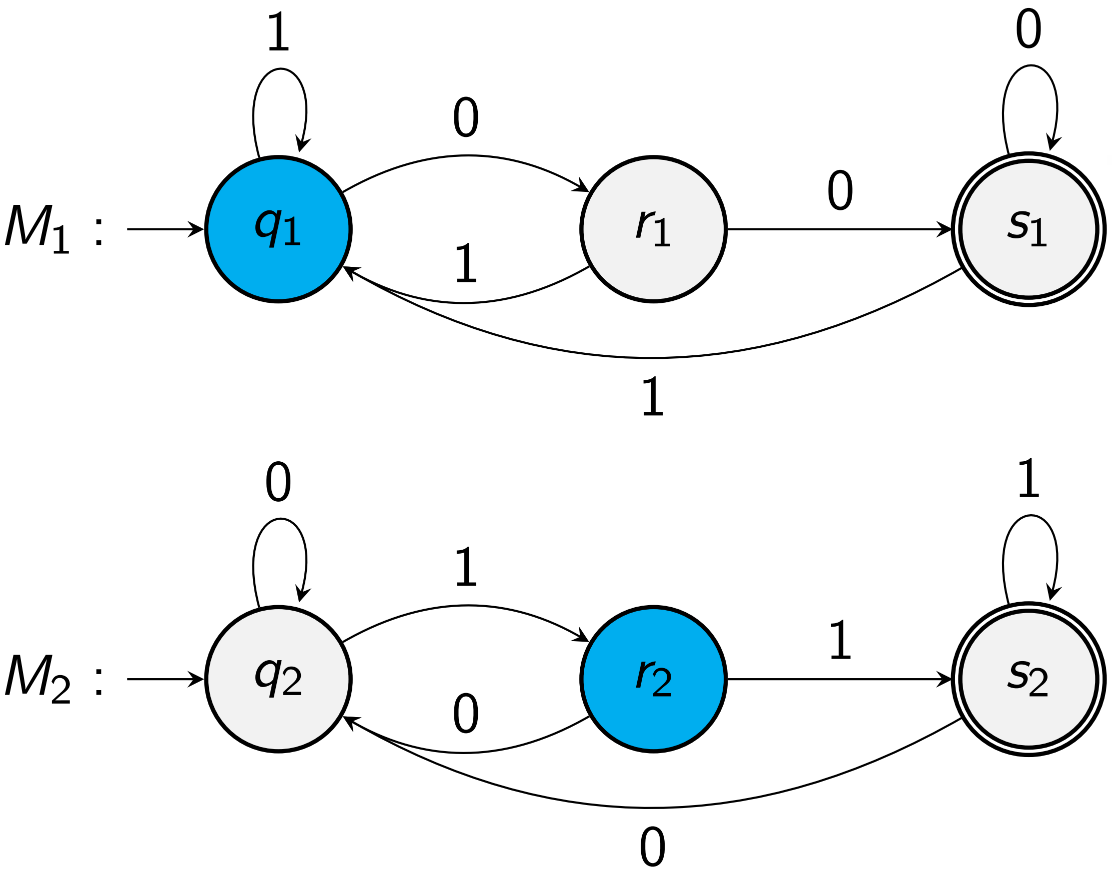


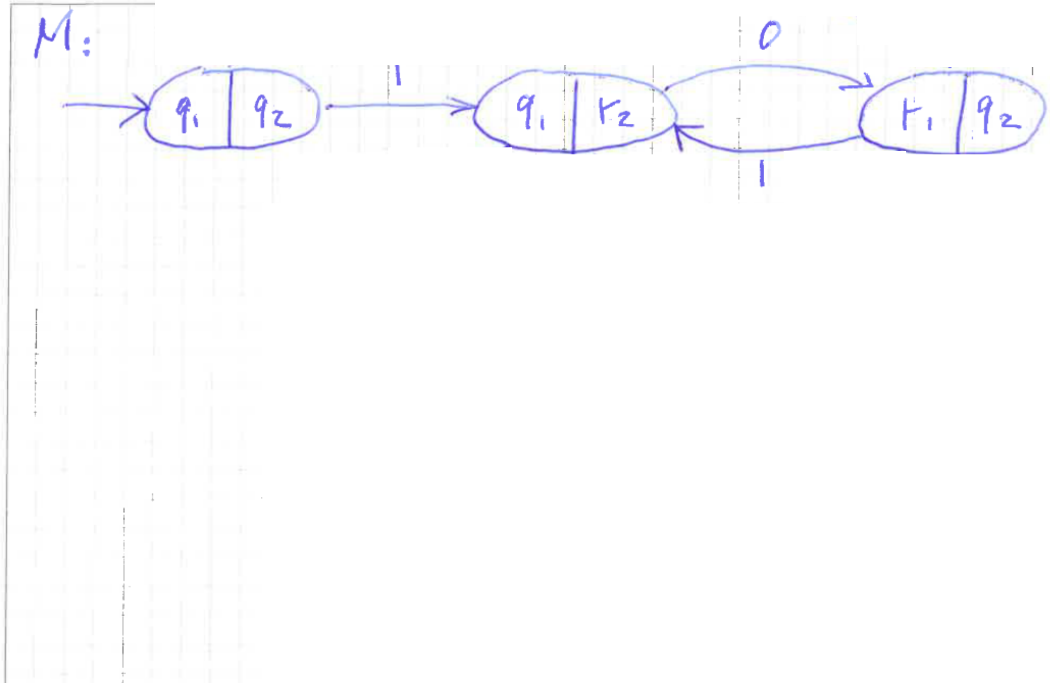


current combined state of M_1 and M_2 is (r_1, q_2),
which shall be the current state of M

Example: How to construct M based on M_1 and M_2

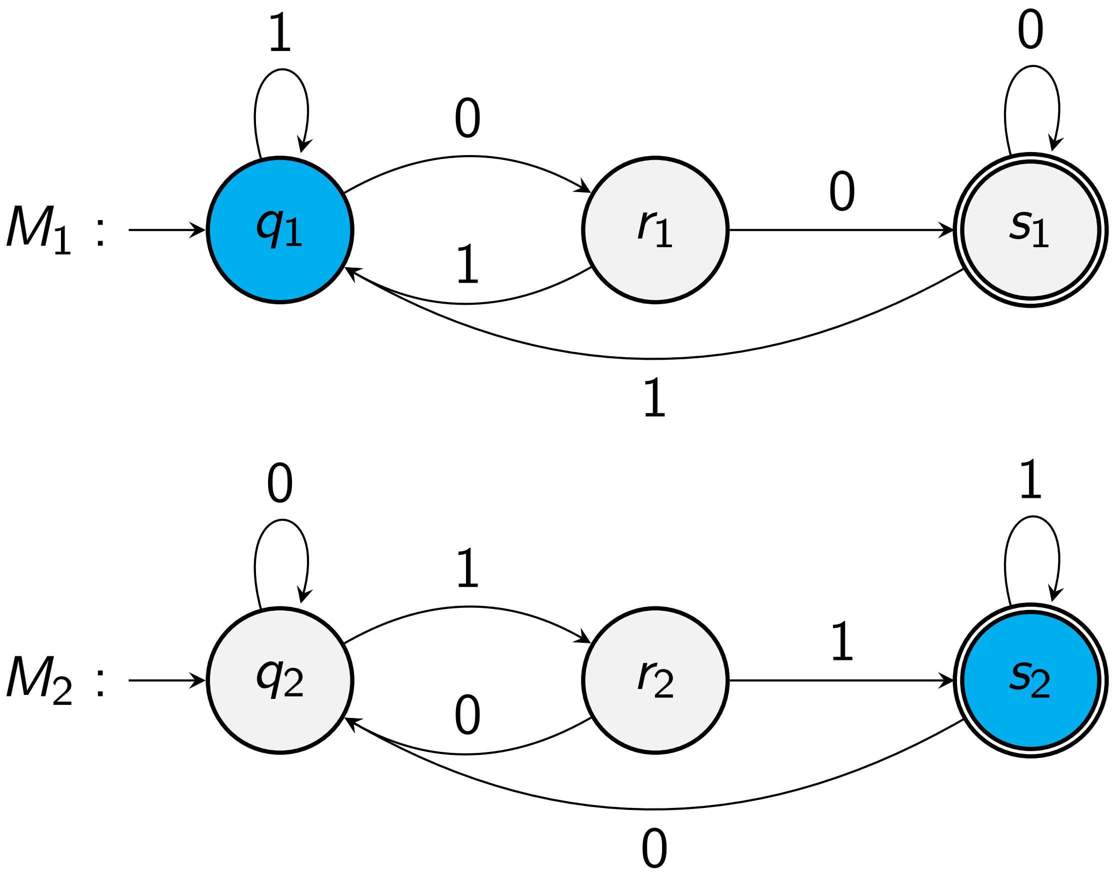
Run M_1 and M_2 simultaneously with $\mathbf{w} = \mathbf{10\boxed{1}1}$

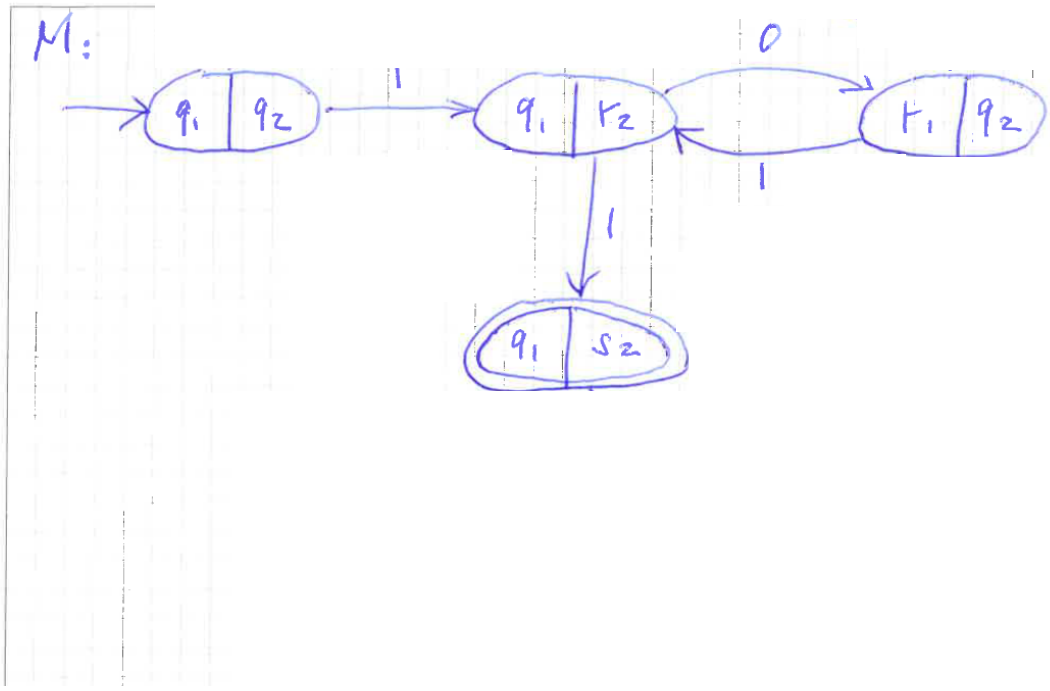




current combined state of M_1 and M_2 is (q_1, r_2),
which shall be the current state of M

Run M_1 and M_2 simultaneously with $\mathbf{w} = \mathbf{101}\boxed{\mathbf{1}}$

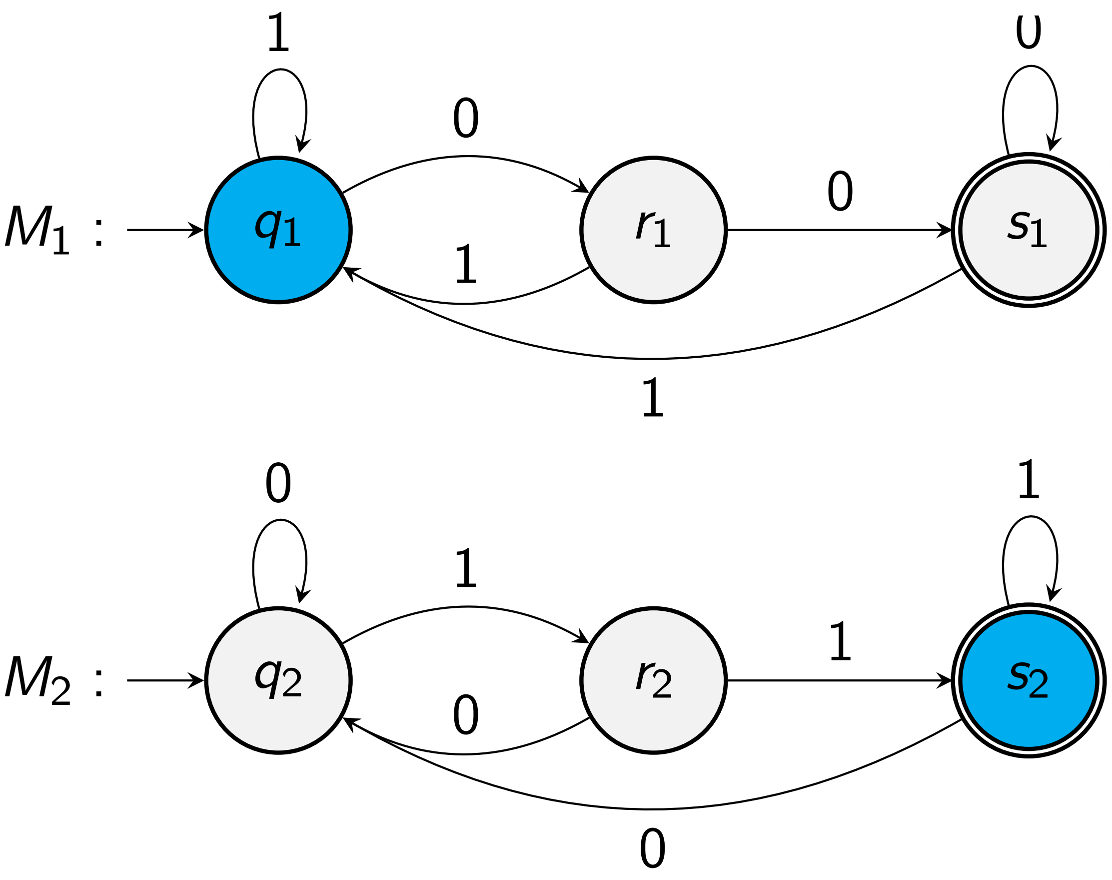


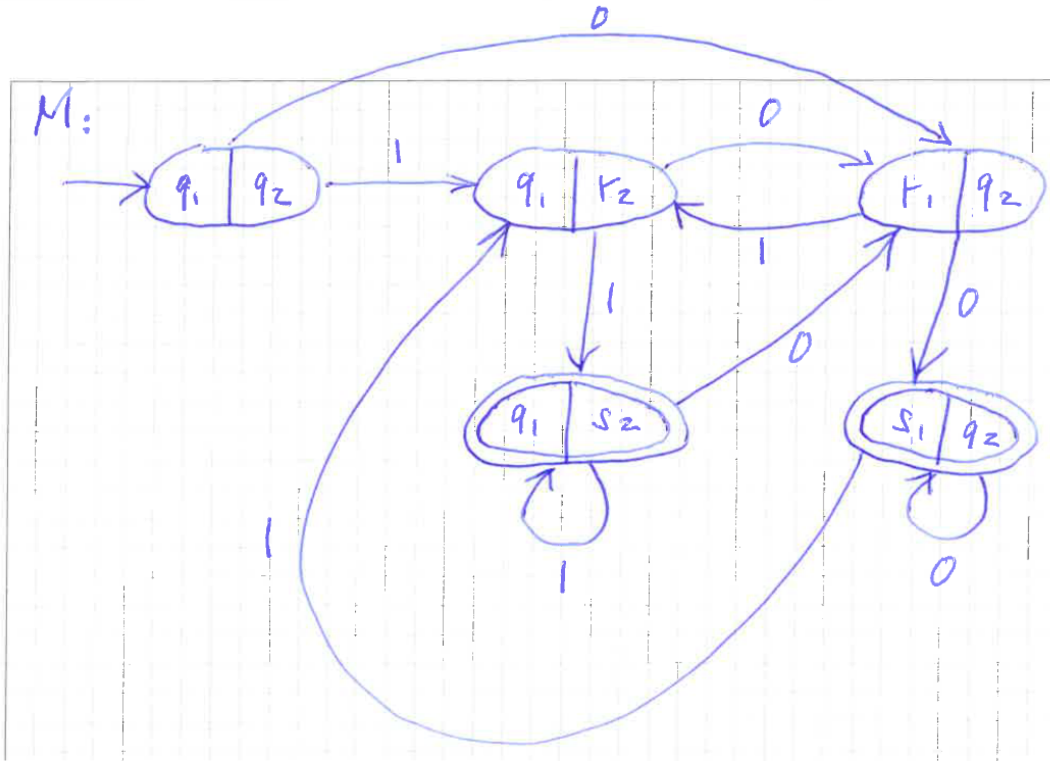


current combined state of M_1 and M_2 is (q_1, s_2),
which shall be the current state of M and an **accepting state** of M

Example: How to construct M based on M_1 and M_2

Run M_1 and M_2 simultaneously





Closure of the Union Operation

Formal Proof of Theorem (Closure of Union):

Given $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$

we construct $M = (Q, \Sigma, \delta, q, F)$ as follows:

- $Q = Q_1 \times Q_2 = \{(t_1, t_2) : t_1 \in Q_1 \text{ and } t_2 \in Q_2\}$
  set of all combined states of M_1 and M_2
- Σ is the same alphabet as that of M_1 and M_2
- $q = (q_1, q_2)$ the combined start state of M_1 and M_2
- $F = \{(t_1, t_2) : t_1 \in F_1 \text{ OR } t_2 \in F_2\}$
  set of all combined states where M_1 or M_2 is in an accepting state

Closure of the Union Operation

Formal Proof of Theorem (Closure of Union):

Given $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$

we construct $M = (Q, \Sigma, \delta, q, F)$ as follows:

- **transition function** $\delta : Q \times \Sigma \to Q$ with

$$\delta((t_1, t_2), a) = \underbrace{(\delta_1(t_1, a), \delta_2(t_2, a))}_{(t_1', t_2')}$$

  for all $t_1 \in Q_1$, $t_2 \in Q_2$, and $a \in \Sigma$.

This completes the construction of M.

Formal Proof of Theorem (Closure of Union):

It remains to show that for all strings $w \in \Sigma^*$:

$$M \text{ accepts } w \iff M_1 \text{ accepts } w \;\; \text{OR} \;\; M_2 \text{ accepts } w$$

which is the same as:

$$\bar{\delta}((q_1, q_2), w) \in F \iff \bar{\delta}_1(q_1, w) \in F_1 \;\; \text{OR} \;\; \bar{\delta}_2(q_2, w) \in F_2$$

Closure of the Union Operation

Formal Proof of Theorem (Closure of Union):

It remains to show that for all strings $w \in \Sigma^*$:

$$M \text{ accepts } w \iff M_1 \text{ accepts } w \;\text{ OR }\; M_2 \text{ accepts } w$$

which is the same as:

$$\bar{\delta}((q_1, q_2), w) \in F \iff \bar{\delta}_1(q_1, w) \in F_1 \;\text{ OR }\; \bar{\delta}_2(q_2, w) \in F_2$$

Proof of Equivalence:

$$\bar{\delta}((q_1, q_2), w) \in F$$

$$\Leftrightarrow \quad \big(\bar{\delta}_1(q_1, w), \bar{\delta}_2(q_2, w)\big) \in F \qquad \text{(Def. of } \delta\text{)}$$

$$\boxed{\delta((t_1, t_2), a) = (\delta_1(t_1, a), \delta_2(t_2, a))}$$

Formal Proof of Theorem (Closure of Union):

It remains to show that for all strings $w \in \Sigma^*$:

$$M \text{ accepts } w \iff M_1 \text{ accepts } w \ \text{ OR } \ M_2 \text{ accepts } w$$

which is the same as:

$$\bar{\delta}((q_1, q_2), w) \in F \iff \bar{\delta}_1(q_1, w) \in F_1 \ \text{ OR } \ \bar{\delta}_2(q_2, w) \in F_2$$

Proof of Equivalence:

$$\begin{aligned} & \bar{\delta}((q_1, q_2), w) \in F & \\ \Leftrightarrow \quad & \big(\bar{\delta}_1(q_1, w), \bar{\delta}_2(q_2, w)\big) \in F & \text{(Def. of } \delta\text{)} \\ \Leftrightarrow \quad & \big(\bar{\delta}_1(q_1, w), \bar{\delta}_2(q_2, w)\big) \in \{(r_1, r_2) : r_1 \in F_1 \text{ OR } r_2 \in F_2\} & \text{(Def. of } F\text{)} \end{aligned}$$

$$\boxed{F = \{(t_1, t_2) : t_1 \in F_1 \text{ OR } t_2 \in F_2\}}$$

Closure of the Union Operation

Formal Proof of Theorem (Closure of Union):

It remains to show that for all strings $w \in \Sigma^*$:

$$M \text{ accepts } w \iff M_1 \text{ accepts } w \;\text{ OR }\; M_2 \text{ accepts } w$$

which is the same as:

$$\boxed{\bar{\delta}((q_1, q_2), w) \in F \iff \bar{\delta}_1(q_1, w) \in F_1 \;\text{ OR }\; \bar{\delta}_2(q_2, w) \in F_2}$$

Proof of Equivalence:

$$\begin{aligned}
& \bar{\delta}((q_1, q_2), w) \in F \\
\Leftrightarrow \quad & \big(\bar{\delta}_1(q_1, w), \bar{\delta}_2(q_2, w)\big) \in F && \text{(Def. of } \delta\text{)} \\
\Leftrightarrow \quad & \big(\bar{\delta}_1(q_1, w), \bar{\delta}_2(q_2, w)\big) \in \{(r_1, r_2) : r_1 \in F_1 \text{ OR } r_2 \in F_2\} && \text{(Def. of } F\text{)} \\
\Leftrightarrow \quad & \bar{\delta}_1(q_1, w) \in F_1 \;\text{ OR }\; \bar{\delta}_2(q_2, w) \in F_2 \quad \square
\end{aligned}$$

Closure of the Union Operation

Theorem (Closure of Union)

The set of regular languages R over Σ is closed under the Union *operation.*