A language is **regular** over $\Sigma$ if it can be built from $\emptyset$, \{$\Lambda$\}, and \{a\} for every $a \in \Sigma$, using operators union ($\cup$), concatenation (\cdot) and Kleene star ($^*$).

Example: $(\{a\} \cup \{b\})^* \{a\} \{a\} (\{a\} \cup \{b\})^* \cup (\{a\} \cup \{b\})^* \{b\} \{b\} (\{a\} \cup \{b\})^*$

Simplify: $(a \cup b)^* a a (a \cup b)^* \cup \ldots$

Final simplification: $(a \cup b)^* a a (a \cup b)^* + \overset{\text{reg. expression}}{\leftarrow}$
A language is regular over $\Sigma$ if it can be built from $\emptyset$, $\{ \Lambda \}$, and $\{ a \}$ for every $a \in \Sigma$, using operators union ($\cup$), concatenation ($.$) and Kleene star ($^*$).

Recursive definition of regular languages over $\Sigma$:

1) $\emptyset$, $\{ \Lambda \}$ and $\{ a \}$ for every $a \in \Sigma$ are regular
2) if $L_1$ and $L_2$ are regular then $L_1 \cup L_2$, $L_1 \cdot L_2$, $L_1^*$ are regular
3) no other language regular
Recursive definition of regular expressions over $\Sigma$:

1) $\emptyset$, {$\Lambda$}, and {a} for every $a \in \Sigma$ are regular languages represented by regular expressions $\emptyset$, $\Lambda$, $a$, respectively.

2) If $L_1$, $L_2$ are regular languages represented by regular expressions $r_1$ and $r_2$, then $L_1 \cup L_2$, $L_1L_2$, and $L_1^*$ are regular languages represented by regular expressions $r_1+r_2$, $r_1 \cdot r_2$, $r_1^*$, respectively.

3) No other expressions are regular expressions.
Regular Expressions

Comments:
- For convenience, we will also use $r^+$ and $r^i$ for $i \in \mathbb{N}$.
- The priority of operators in decreasing order: Kleene’s star, concatenation, union.

Parenthesize:

(a + (b^{*}c))
(a + (b(c^{*})))
Regular Expressions

Simplify the following regular expressions:

- \( a^* (a+\Lambda) = a^* \)
- \( a^* a^* = a^* \)
- \( (a^* b^*)^* = (a+b)^* \)
- \( (a+b)^* ab (a+b)^* + a^* b^* = (a+b)^* \)

Strings not in here:
- all a's
- all b's
- \( \Lambda \)
- \( b^* a^* \)
Regular Expressions

Give a regular expression for each of the following:

- the language of all strings over \{a,b\} of even length,
  \[(aa + ab + ba + bb)^* = (a + b)^2]^*\]

- the language of all strings over \{a,b,c\} in which all a’s precede all b’s,
  \[(a+c)^* (b+c)^*\]

- the language of all strings over \{a,b\} with length greater than 3,
  \[(a+b)^3 \cdot (a+b)^*\]

- the language of all odd-length strings over \{a,b\} containing the string bb,
  \[((a+b)^2)^* (a+b) bb (a+b)^2)^* + (a+b)^2 bb (a+b) ((a+b)^2)^*\]

- the language of all strings over \{a,b\} that do not contain the string aaa.
  \[(\Lambda + a + aa) \left( b (\Lambda + a + aa) \right)^*\]

Is the language \(L = \{ a^k b^k \mid k \in \mathbb{N} \}\) regular?

\(L \subseteq a^* b^*\)

NO, we'll see a proof.

\(L \notin a^* b^*\)
Finite Automata

Our simplest model of computation:
- has a tape with input, read once from left to right
- has only a single memory register with the state

[Sections 3.2, 3.3]
Example: the language of all strings over \{a,b\} with an even number of a's.
Finite Automata

More examples:
- Language of strings over \{a,b\} with at least 3 a’s.
- Language of strings over \{a,b\} containing substring aaba.
- Language of strings over \{a,b\} not containing substring aaba.
- Language of strings over \{a,b\} with even number of a’s and odd number of b’s.

[Sections 3.2, 3.3]
A (deterministic) **finite automaton** (FA, in some books also abbreviated as DFA) is a 5-tuple \((Q, \Sigma, q_0, A, \delta)\) where

- **\(Q\)** is a finite set of states
- **\(\Sigma\)** is a finite alphabet (input symbols)
- **\(q_0 \in Q\)** is the initial state
- **\(A \subseteq Q\)** is a set of accepting states
- **\(\delta : Q \times \Sigma \rightarrow Q\)** is the transition function
Finite Automata

Comment: We saw that we can represent a finite automaton by a transition diagram.

Draw the transition diagram for \( ( \{ q_a, q_b \}, \{ a, b \}, q_a, \{ q_b \}, \delta ) \) where \( \delta \) is given by the following table:

\[
\begin{align*}
(q_a, a) & \rightarrow q_a \\
(q_a, b) & \rightarrow q_b \\
(q_b, a) & \rightarrow q_a \\
(q_b, b) & \rightarrow q_b
\end{align*}
\]

Which strings does this FA accept?
Finite Automata

Defining the computation of an FA $M=(Q,\Sigma,q_0,A,\delta)$.

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

1) For every $q \in Q$, let $\delta^*(q,\lambda) = q$

2) For every $q \in Q$, $y \in \Sigma^*$, and $\sigma \in \Sigma$, let $\delta^*(q,y\sigma) = \delta(\delta^*(q,y),\sigma)$

We say that a string $x \in \Sigma^*$ is **accepted by $M$** if $\delta^*(q_0, x) \in A$.

A string which is not accepted by $M$ is **rejected by $M$**.

The **language accepted by $M$**, denoted by $L(M)$ is the set of all strings accepted by $M$. 

[Sections 3.2, 3.3]
Kleene’s Thm: A language L over Σ is regular iff there exists a finite automaton that accepts L.

Recall $L = \{ a^k b^k \mid k \geq 0 \}$. Is it regular?

Suppose we have a FA $M = (Q, \Sigma, q_0, A, \delta)$ which accepts $L$ ($= L(M)$).

Look at $a^i$ - infinite # of such strings but only $|Q|$ (finite) # states.

Therefore $\exists i \neq j$ s.t. $\delta^*(q_0, a^i) = \delta^*(q_0, a^j)$.

Since M is assumed to be accepting L, then $\delta^*(q_0, a^i b^i) \in A$.

Then $\delta^*(q_0, a^i b^i) = \delta^*(q_0, a^i b^i) \in A$

Therefore $a^i b^i$ must be accepted by M but shouldn’t be. ✷
Operations on Finite Automata

Let \( M_1 = (Q_1, \Sigma, q_1, A_1, \delta_1) \) and \( M_2 = (Q_2, \Sigma, q_2, A_2, \delta_2) \) be two FA’s.

**Union:**

We know that \( L(M_1) \cup L(M_2) \) is regular. Construct an FA \( M \) such that \( L(M) = L(M_1) \cup L(M_2) \).

\[
M = (Q, \Sigma, q_0, A, \delta)
\]

\[
Q = Q_1 \times Q_2
\]

\[
q_0 = (q_1, q_2)
\]

\[
A = A_1 \times Q_2 \cup Q_1 \times A_2
\]

\[
\delta((p_1, p_2), \sigma) = \delta_1(p_1, \sigma), \delta_2(p_2, \sigma)
\]

for \( p_1 \in Q_1, p_2 \in Q_2, \sigma \in \Sigma \)
Let $M_1 = (Q_1, \Sigma, q_1, A_1, \delta_1)$ and $M_2 = (Q_2, \Sigma, q_2, A_2, \delta_2)$ be two FA's.

**Intersection:**

Is $L(M_1) \cap L(M_2)$ is regular?

Construct an FA $M$ such that $L(M) = L(M_1) \cap L(M_2)$.

Same as before except

$$A = A_1 \times A_2$$
Let $M_1 = (Q_1, \Sigma, q_1, A_1, \delta_1)$ be an FA.

**Complement:**

Is $L(M_1)'$ regular?

Construct an FA $M$ such that $L(M) = L(M_1)'$.

$$M = (Q_2, \Sigma, q_2, Q_2 - \bar{A}_1, \delta_2)$$
Operations on Finite Automata

Let $M_1 = (Q_1, \Sigma, q_1, A_1, \delta_1)$ and $M_2 = (Q_2, \Sigma, q_2, A_2, \delta_2)$ be two FA's.

**Difference:**

Construct an FA $M$ such that $L(M) = L(M_1) - L(M_2)$.

*Lazy approach:* construct $M_3$ s.t. $L(M_3) = L(M_2)'$

$M$ s.t. $L(M) = L(M_1) \cap L(M_3)$

"Cartesian-product" automaton approach:

$$ A = A_1 \times (Q_2 - A_2) $$
Closure Properties

We just saw that the class of (regular languages) is closed under union, intersection, complement, and difference.

Let \( L = \{ x \in \{a,b\}^* \mid x \text{ has the same number of } a\text{'s and } b\text{'s } \} \).

Is \( L \) regular? NO, because:

Recall \( L_1 = \{ a^k b^k \mid k \geq 0 \} \) not regular.

Suppose \( L \) is regular (\( \exists \) a FA \( M \) s.t. \( L(M) = L \)).

\[ L \cap a^* b^* = L_1 \]

We know \( a^* b^* \) is regular, therefore \( L \cap a^* b^* \) is regular, but \( L \cap a^* b^* = L_1 \) which is not regular.

Thus, \( L \) can't be regular.