Finite Automata

- basic computational model: limited amount of memory
- example: controller for an automatic door
Finite Automata

**Formal definition:**

A **finite automaton** (FA) is a 5-tuple \((Q, \Sigma, \delta, q_0, F)\), where

- \(Q\) is a **finite set of states**
- \(\Sigma\) is a (finite) alphabet
- \(\delta: Q \times \Sigma \rightarrow Q\) is the **transition function**
- \(q_0 \in Q\) is the **start state**
- \(F \subseteq Q\) is the set of **accept states**

**Pictorial representation:** state diagram
Another (more abstract) example:
- accept all strings over \( \{0,1\} \) that start with 1 and end with 0

\[100101\]
Let $M=(Q, \Sigma, \delta, q_0, F)$ be a FA. The language of $M$ (accepted / recognized by $M$) is $L(M)$.

Formally: need the definition of computation:

**$M$ accepts** $w=w_1w_2...w_n$ if there exist states $r_0, r_1, ..., r_n$ in $Q$ such that

- $r_0 = ? \not\in Q_0$
- $\delta(r_{i-1}, w_i) = r_i \quad \forall i \in \{1, ..., n\}$
- $r_n \in F$

A language is **regular** if there exists a FA that recognizes it.
Designing FAs

Examples - languages over \{0,1\} consisting of strings:
- with odd number of 1’s
- that contain 001 as a substring
- that are even length and do not contain 00 as a substring

A language that cannot be accepted by a FA?

\[ \{0^k 1^k \mid k \geq 0\} \]
Let $A$ and $B$ be languages. The following three language operations are called the regular operations:

- union: $A \cup B$
- concatenation: $A.B$
- star: $A^*$

The natural numbers are closed under multiplication but not division. For any two natural numbers $x, y$, the product $x \cdot y$ is a natural number.

What about the class of regular languages?
Thm 1.25: The class of regular languages is closed under the union operation.

Proof: If $A, B$ are regular lang., then $A \cup B$ is a regular lang.  $\leftarrow$ WANT TO PROVE

Have $M_A$ accepting $A$ and $M_B$ accepting $B$, we want to construct $M_{A \cup B}$ for $A \cup B$

Let $M_A = (Q_A, \Sigma, \delta_A, q_{0A}, F_A)$

$$M_{A \cup B} = (Q_A \times Q_B, \Sigma, \delta_{A \cup B}(q_{0A}, q_{0B}), F_A \times Q_B \cup Q_A \times F_B)$$

(note: if proving closedness under $N$, then $F_{A \cup B} = F_A \times F_B$)

$$\delta_{A \cup B}((p_A, p_B), \sigma) = (\delta_A(p_A, \sigma), \delta_B(p_B, \sigma)) \quad \forall p_A \in Q_A, p_B \in Q_B, \sigma \in \Sigma$$

$\square$

Note: Thm $AB$: the class of regular lang. is closed under complement.

Pf idea: switch accept & non-accept states.
Thm 1.26: The class of regular languages is closed under the concatenation operation.

\[ w = w_1 w_2 \ldots w_n \]  
- want to check if in \( A \cdot B \)

i.e. can split \( w \) into \( u \cdot v \) where \( u \in A \) and \( v \in B \)

feels like we need to "guess" the split