Languages

Because sets of strings are so important, let's give them a name. A *formal language* (usually called only a language) over alphabet Σ is a set of strings over Σ.

Since languages are sets, we inherit the usual set operations $A \cup B$, $A \cap B$, $\overline{A}$ (where the universe of A is taken to be Σ^*).

Because a language A is a set of strings specifically, we can also define more specific operations.

> $A \cdot B = \{uv \mid u \in A \text{ and } v \in B\}$, that is, the set of all strings obtained by concatenating a string of A and a string of B.
>
> $A^0 = \{\epsilon\}$
>
> $A^1 = A$
>
> $A^2 = A \cdot A$
>
> $A^3 = A \cdot A \cdot A$
>
> etc...
>
> $A^* = A^0 \cup A^1 \cup A^2 \cup A^3 \cup \cdots = \bigcup_{k \geq 0} A^k$

The * operation is called the *Kleene star*.

Some properties that are easy to verify:

> $\varnothing \cdot A = A \cdot \varnothing = \varnothing$
>
> $\{\epsilon\} \cdot A = A \cdot \{\epsilon\} = A$

Example: $\Sigma = \{\mathtt{a}, \mathtt{b}\}$, and $A = \{\mathtt{aa}, \mathtt{bb}\}$. Then $A^* = \{\epsilon, \mathtt{aaaa}, \mathtt{aabb}, \mathtt{bbaa}, \mathtt{bbbb}, \mathtt{aaaaaa}, \mathtt{aaaabb}, \mathtt{aabbaa}, \mathtt{aabbbb}, \mathtt{bbaaaa}, \mathtt{bbaabb}, \mathtt{bbbbaa}, \mathtt{bbbbbb}, \ldots\}$.

This explains why I wrote Σ^* for the set of all strings: if we consider Σ as a set of strings, each of length 1, then Σ^* according to the above definition indeed gives the set of all strings over alphabet Σ.

The operations $\cup$, $\cdot$, and * are called the *regular operations*.

Deterministic Finite Automata

A *finite automaton* is a structure

$$M = (Q, \Sigma, \Delta, s, F)$$

where

- Q is a finite set of states
- Σ is a finite alphabet
- $\Delta \subseteq Q \times \Sigma \times Q$ is a transition relation: $\langle p, a, q \rangle \in \Delta$ when there is a transition from state p to state q labeled by the symbol a
- $s \in Q$ is the start state
- $F \subseteq Q$ is a set of final states.

A finite automaton M is *deterministic* if for every state of M and every symbol of the alphabet there is excatly one transition out of that state labeled with that symbol. The transition relation Δ for a deterministic finite automaton (DFA) is a function from $Q \times \Sigma$ to Q.

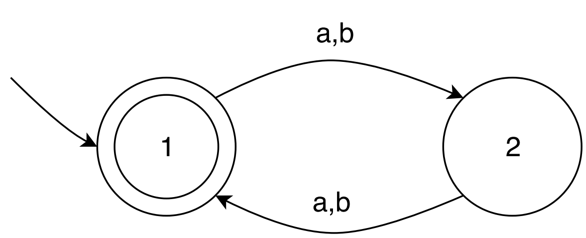
Finite automaton M *accepts* string $u = a_1 \ldots a_k$ if there is a path in M starting with s labeled by $a_1, a_2, \ldots, a_k$, and ending up in a final state. For a deterministic finite automaton, this is equivalent to starting in the start state and following the one and only path labeled by the symbols in the input string.

Formally: $M = (Q, \Sigma, \Delta, s, F)$ accepts $u = a_1 \ldots a_k$ if there exists $q_0, q_1, \ldots, q_k \in Q$ such that $q_0 = s$, $q_k \in F$, and $\langle q_{i-1}, a_i, q_i \rangle \in \Delta$ for all $1 \leq i \leq k$.

The language accepted by M is

$$L(M) = \{u \mid M \text{ accepts } u\}$$

Example: The following finite automaton accepts exactly the strings over $\{\mathtt{a}, \mathtt{b}\}$ of even length:



$$M_{even} = (\{1, 2\}, \{\mathtt{a}, \mathtt{b}\}, \Delta_{even}, 1, \{1\})$$

where the transitions go from state 1 to state 2 and back no matter the symbol:

$$\Delta_{even} = \{\langle 1, \mathtt{a}, 2\rangle, \langle 1, \mathtt{b}, 2\rangle, \langle 2, \mathtt{a}, 1\rangle, \langle 2, \mathtt{b}, 1\rangle\}$$

Definition: A language A is *regular* if there exists some finite automaton M which accepts A.