COL352 Lecture 3

Contents

1 Regular Languages

Recall the definition of a DFA from the last lecture - and consider δ.

We'll try to extend δ to a certain $\hat{\delta}$, so that we get a function from $Q \times \Sigma^*$ to Q, such that $\hat{\delta}(q, w)$ is the state reached by starting from q and following transitions labelled by symbols in w.
Formally, let

$$\hat{\delta}(q, \epsilon) = q \quad \forall q \in Q.$$

$$\hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a) \quad \forall q \in Q, x \in \Sigma^*, a \in \Sigma.$$

Definition 1. *A DFA $(Q, \Sigma, \delta, q_0, A)$ is said to accept a string w if $\hat{\delta}(q_0, w) \in A$.
It is said to reject a string w if $\hat{\delta}(q_0, w) \notin A$.*

Definition 2. *The language recognised by D is the set $\{w \in \Sigma^* \mid D \text{ accepts } w\}$, and is denoted by $\mathcal{L}(D)$, and is also sometimes called the language of D.*

Definition 3. *A language is said to be regular if it is recognised by some DFA.*

Example 1. *$\Sigma = \{0, 1\}$, $L_k = \{w \mid w \text{ is the binary representation of a multiple of } k\}$ Then L_k is regular for all $k \in \mathbb{N}$.*

Example 2. *$\Sigma = \{a, b\}$, $END_w = \{x \mid x \text{ ends in } w\}$ for some $w \in \Sigma^*$. Then END_w is regular for all $w \in \Sigma^*$.*

Claim 1. *Let $L \subseteq \Sigma^*$ be a regular language. Then $\Sigma^* \setminus L$ is also regular.*

Proof. Suppose $D = (Q, \Sigma, \delta, q_0, A)$ be a DFA that accepts L. Then consider the DFA $D' = (Q, \Sigma, \delta, q_0, Q \setminus A)$. Consider the language $\mathcal{L}(D')$.
$w \in L \iff \hat{\delta}(q_0, w) \in A \iff \hat{\delta}(q_0, w) \notin Q \setminus A \iff w \notin \mathcal{L}(D')$.
So we have $\mathcal{L}(D') = \Sigma^* \setminus L$, and hence we are done. ■

In other words, the class of regular languages is closed under complementation.

Claim 2. *If $L_1, L_2 \in \Sigma^*$ are regular languages, then $L_1 \cap L_2$ is also regular.*

Proof. Suppose $D_1 = (Q_1, \Sigma, \delta_1, q_1, A_1)$ is a DFA recognising L_1. Suppose $D_2 = (Q_2, \Sigma, \delta_2, q_2, A_2)$ is a DFA recognising L_2. Let D be a DFA defined by

$$D = (Q_1 \times Q_2, \Sigma, \delta, (q_1, q_2), A_1 \times A_2).$$

where δ is defined as

$$\delta((s_1, s_2), a) = (\delta_1(s_1, a), \delta_2(s_2, a)).$$

Now we claim that D recognizes $L_1 \cap L_2$.

Subclaim 1. $\hat{\delta}((s_1, s_2), w) = (\hat{\delta_1}(s_1, w), \hat{\delta_2}(s_2, w))$

Proof. We proceed by induction on $|w|$. When $|w| = 0$, we have $\hat{\delta}((s_1, s_2), \epsilon) = (s_1, s_2) = (\hat{\delta_1}(s_1, w), \hat{\delta_2}(s_2, w))$, so we are done in this case. Now suppose $w = xa$ for some $x \in \Sigma^*, a \in \Sigma$. Then we have

$$\begin{aligned}
\hat{\delta}((s_1, s_2), xa) &= \delta(\hat{\delta}((s_1, s_2), x)) \\
&= \delta((\hat{\delta_1}(s_1, x), \hat{\delta_1}(s_1, x)), a) \\
&= (\delta_1(\hat{\delta_1}(s_1, x), a), \delta_2(\hat{\delta_2}(s_2, x), a)) \\
&= (\hat{\delta_1}(s_1, xa), \hat{\delta_2}(s_2, xa))
\end{aligned}$$

whence we are done. ■

Now we use this claim to see the following sequence of equivalences:

$$\begin{aligned}
w \in L_1 \cap L_2 &\iff w \in L_1 \wedge w \in L_2 \\
&\iff D_1 \text{ accepts } w \wedge D_2 \text{ accepts } w \\
&\iff \hat{\delta_1}(q_1, w) \in A_1 \wedge \hat{\delta_2}(q_2, w) \in A_2 \\
&\iff (\hat{\delta_1}(q_1, w), \hat{\delta_2}(q_2, w)) \in A_1 \times A_2 \\
&\iff D \text{ accepts } w
\end{aligned}$$

whence we are done. ■

In other words, the class of regular languages is closed under intersection.

Claim 3. *If $L_1, L_2 \in \Sigma^*$ are regular languages, then $L_1 \cup L_2$ is also regular.*

Proof. We show two proofs.

1. $L_1 \cup L_2 = \Sigma^* \setminus ((\Sigma^* \setminus L_1) \cap (\Sigma^* \setminus L_2))$, so using closure properties under complementation, we are done.
2. (sketch) use the same DFA as before, but replace A to $\{(s_1, s_2) \mid s_1 \in A_1 \vee s_2 \in A_2\}$ (which is equivalent to saying that $A = Q_1 \times Q_2 \setminus ((Q_1 \setminus A_1) \times (Q_2 \setminus A_2)))$

■

Corollary 1. *Every finite language L is regular.*

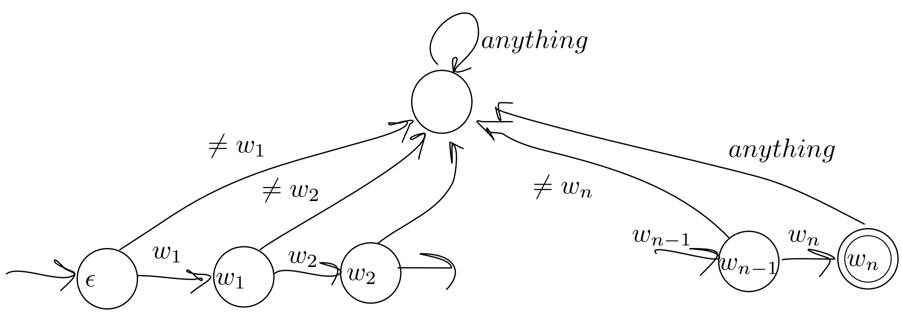

Figure 1: Automaton for string matching

Proof. If $|L| = 1$, then we are done (make a DFA to recognize a single word). Else, use closure under union and induction on $|L|$. ■

Note 1. *Note that this doesn't extend to all languages L, since we never said that the union of a countable collection of regular languages is regular.*