Lecture 4: The product construction: Closure under intersection and union

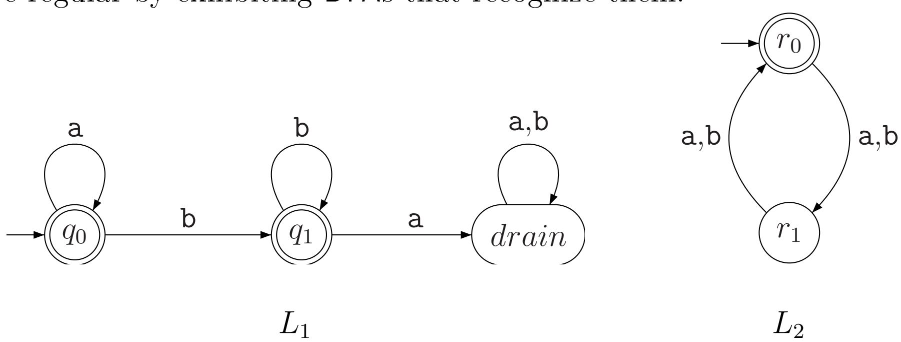
28 January 2010

This lecture finishes section 1.1 of Sipser and also covers the start of 1.3.

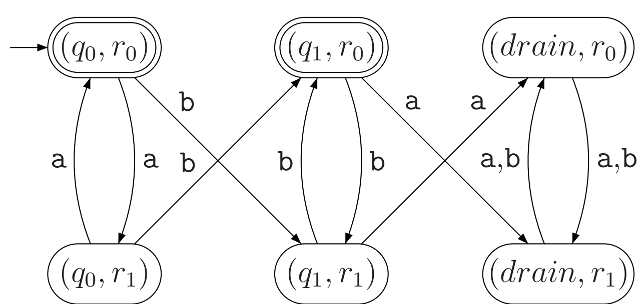
1 Product Construction

1.1 Product Construction: Example

Let $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and L is the set of strings in Σ^* that have the form $\mathtt{a}^*\mathtt{b}^*$ and have even length. L is the intersection of two regular languages $L_1 = \mathtt{a}^*\mathtt{b}^*$ and $L_2 = (\Sigma\Sigma)^*$. We can show they are regular by exhibiting DFAs that recognize them.



We can run these two DFAs together, by creating states that remember the states of both machines.



Notice that the final states of the new DFA are the states (q, r) where q is final in the first DFA ***and*** r is final in the second DFA. To recognize the union of the two languages, rather than the intersection, we mark all the states (q, r) such that either q or r are accepting states in the their respective DFAs.

State of a DFA after reading a word w. In the following, given a DFA $M = (Q, \Sigma, \delta, q_0, F)$, we will be interested in the state the DFA M is in, after reading the a string w. Let us denote by $\delta^* : Q \times \Sigma^* \to Q$ the function such that $\delta^*(q, w)$ is the state the DFA will land in, if started from state q and fed the word w.

Formally, we define $\delta^* : Q \times \Sigma^* \to Q$ using the following *inductive definition*:

- $\delta^*(q, \epsilon) = q$ for every $q \in Q$,
- $\delta^*(q, wa) = \delta(\delta^*(q, w), a)$ for each $q \in Q, w \in \Sigma^*, a \in \Sigma$.

Note that by the above definition of δ^*, we have that $w \in L(M)$ iff $\delta^*(q_0, w) \in F$.

2 Product Construction: Formal construction

We are given two DFAs $M = (Q, \Sigma, \delta, q_0, F)$ and $M' = (Q', \Sigma, \delta', q'_0, F')$ both working over the same alphabet Σ. A ***product automaton*** of M and M' is an automaton

$$N = \Big(\mathcal{Q},\ \Sigma,\ \delta_N,\ (q_0, q'_0)\ ,\ F_N \Big)\ ,$$

where $\mathcal{Q} = Q \times Q'$, and $\delta_N : \mathcal{Q} \times \Sigma \to \mathcal{Q}$. Also, for any $q \in Q$, $q' \in Q'$ and $c \in \Sigma$, we require

$$\delta_N(\underbrace{(q, q')}_{\text{state of } N}, c\ ) = \Big(\delta(q, c),\ \delta'(q', c) \Big)\ . \tag{1}$$

The set $F_N \subseteq \mathcal{Q}$ of accepting states is free to be whatever we need it to be, depending on what we want N to recognize. For example, if we would like N to accept the intersection $L(M) \cap L(M')$ then we will set $F_N = F \times F'$. If we want N to recognize the union language $L(M) \cup L(M')$ then $F_N = (F \times Q')\ \cup \cup (Q \times F')$.

Lemma 2.1 *For any input word $w \in \Sigma^*$, the product automata N of the DFAs $M = (Q, \Sigma, \delta, q_0, F)$ and $M' = (Q', \Sigma, \delta', q'_0, F')$, is in state (q, q') after reading w, if and only if (i) M in the state q after reading w, and (ii) M' is in the state q' after reading w. In other words, $\delta^*_N((q_0, q'_0), w) = (\delta^*(q_0, w), \delta'^*(q'_0, w))$.*

Proof: The proof is by induction on the length of the word w.

If $w = \epsilon$ is the empty word, then N is initially in the state (q_0, q'_0) by construction, where q_0 (resp. q'_0) is the initial state of M (resp. M'). As such, the claim holds in this case.

Formally, $\delta^*_N((q_0, q'_0), \epsilon) = (q_0, q'_0) = (\delta^*(q_0, \epsilon), \delta'^*(q'_0, \epsilon))$ (by definition of δ^*_N, δ^* and δ'^*).

Otherwise, $|w| > 0$, and let us hence assume $w = w_1 a$ ($w_1 \in \Sigma^*, a \in \Sigma$), and assume the induction hypothesis that the claim is true for all input words of length strictly smaller than $|w|$ (in particular $|w_1|$).

Let (q_{k-1}, q'_{k-1}) be the state that N is in after reading the string w_1. By the induction hypothesis, as $|w_1| = k - 1$, we know that M is in the state q_{k-1} after reading $\widehat{w}$, and M' is in the state q'_{k-1} after reading $\widehat{w}$.

Let $q_k = \delta(q_{k-1}, a) = \delta(\delta^*(q_0, w_1)\ ,\ a) = \delta(q_0, w)$ and

$$q'_k = \delta'(q'_{k-1}, a) = \delta'(\delta'(q'_0, w_1)\ ,\ a) = \delta'(q'_0, w)\ .$$

As such, by definition, M (resp. M') would in the state q_k (resp. q'_k) after reading w.

Also, by the definition of its transition function, after reading w the DFA N would be in the state

$$\begin{aligned}\delta_N((q_0, q'_0), w) &= \delta_N(\delta^*_N((q_0, q'_0), w_1)\,, a) = \delta_N\big((q_{k-1}, q'_{k-1}),\ a\big)\\ &= \big(\delta(q_{k-1}, a),\ \delta(q'_{k-1}, a)\big) = (q_k,\ q'_k)\,,\end{aligned}$$

(see Eq. (1)). This establishes the claim. ■

Lemma 2.2 *Let $M = (Q, \Sigma, \delta, q_0, F)$ and $M' = (Q', \Sigma, \delta', q'_0, F')$ be two given DFAs. Let N be a product automaton with set of accepting states is $F \times F'$. Then $L(N) = L(M) \cap L(M')$.*

Proof: $w \in L(N)$ iff $\delta^*_N((q_0, q'_0), w) \in F \times F'$
iff $(\delta^*(q_0, w), \delta'^*(q'_0, w)) \in F \times F'$ (by Lemma 2.1)
iff $\delta^*(q_0, w) \in F$ and $\delta'^*(q'_0, w) \in F'$
iff $w \in L(M)$ and $w \in L(M')$.

More verbosely:

If $w \in L(M) \cap L(M')$, then let $q_w = \delta^*(q_0, w) \in F$ and $q'_w = \delta'^*(q'_0, w) \in F'$. By Lemma 2.1, this implies that $\delta^*_N((q_0, q'_0), w) = (q_w, q'_w) \in F \times F'$. Namely, N accepts the word w, implying that $w \in L(N)$, and as such $L(M) \cap L(M') \subseteq L(N)$.

Similarly, if $w \in L(N)$, then $(p_w, p'_w) = \delta^*_N((q_0, q'_0), w)$ must be an accepting state of N. But the set of accepting states of N is $F \times F'$. That is $(p_w, p'_w) \in F \times F'$, implying that $p_w \in F$ and $p'_w \in F'$. Now, by Lemma 2.1, we know that $\delta^*(q_0, w) = p_w \in F$ and $\delta'^*(q'_0, w) = p'_w \in F'$. Thus, M and M' both accept w, which implies that $w \in L(M)$ and $w \in L(M')$. Namely, $w \in L(M) \cap L(M')$, implying that $L(N) \subseteq L(M) \cap L(M')$.

Putting the above together proves the lemma. ■