The general idea

Given two finite-state machines (FSMs), call them A and B, the 'product construction' of A and B can implement many different operations on the languages or relations that A and B recognize.

The product construction builds a new machine out of the cross product of the states of A and the states of B. Let $A \times B = (Q, I, F, \delta)$. If $A = (Q_A, I_A, F_A, \delta_A)$ and $B = (Q_B, I_B, F_B, \delta_B)$ then we have

$$Q = Q_A \times Q_B = \{(q_a, q_b) \mid q_a \in Q_A, q_b \in Q_B\} .$$

Which of these states are initial, final, and how the delta transition is defined depends on the particulars of the FSM (acceptor, tranducer) and what the product construction is aiming to compute.

Intersection of regular languages

$$I = I_A \times I_B = \{(q_a, q_b) \mid q_a \in I_A \textbf{ and } q_b \in I_B\} .$$
$$F = F_A \times F_B = \{(q_a, q_b) \mid q_a \in F_A \textbf{ and } q_b \in F_B\} .$$

Finally, $\delta\big((q_a, q_b), \sigma)\big) = (q'_a, q'_b)$ if and only if **both** $\delta_A(q_a, \sigma) = q'_a$ **and** $\delta_B(q_b, \sigma) = q'_b$.

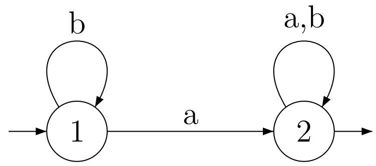
Union of regular languages

$$I = I_A \times I_B = \{(q_a, q_b) \mid q_a \in I_A \textbf{ and } q_b \in I_B\} .$$
$$F = F_A \times F_B = \{(q_a, q_b) \mid q_a \in F_A \textbf{ or } q_b \in F_B\} .$$

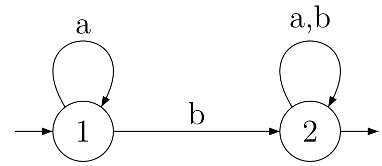
Finally, $\delta\big((q_a, q_b), \sigma)\big) = (q'_a, q'_b)$ if and only if **both** $\delta_A(q_a, \sigma) = q'_a$ **and** $\delta_B(q_b, \sigma) = q'_b$.

Exercise

A:



B:



1. In words, explain the sets of strings $L(A)$ and $L(B)$.

2. Using the product construction for intersection, produce the FSA which accepts the intersection of $L(A)$ and $L(B)$ above.

3. Using the product construction for union, produce the FSA which accepts the intersection of $L(A)$ and $L(B)$ above.

Composition of Literal Transducers

Recall that a literal string-to-string transducer is one where the input and output labels on the transitions are a single symbol from Σ or the 'empty' label ϵ.

Like the construction for automata intersection, I and F are determined by conjunction.

$$I = I_A \times I_B = \{(q_a, q_b) \mid q_a \in I_A \textbf{ and } q_b \in I_B\} \ .$$
$$F = F_A \times F_B = \{(q_a, q_b) \mid q_a \in F_A \textbf{ and } q_b \in F_B\} \ .$$

For clarity transitions in literal transducers are written as (q, a, b, r) where q is the origin state, r is the terminal state, and $a, b \in \Sigma \cup \{\epsilon\}$. Then

$$\big((q_a, q_b), x, y, (q'_a, q'_b)\big) \in \delta$$

if and only if **either**

$$\exists z \text{ such that } (q_a, x, z, q'_a) \in \delta_A \textbf{ and } (q_b, z, y, q'_b) \in \delta_B$$

or

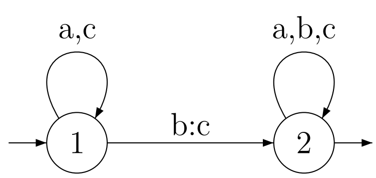
$$y = \epsilon \textbf{ and } (q_a, x, y, q'_a) \in \delta_A \textbf{ and } q_b = q'_b \in Q_B$$

or

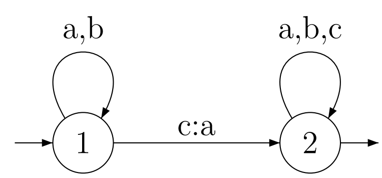
$$x = \epsilon \textbf{ and } (q_b, x, y, q'_b) \in \delta_B \textbf{ and } q_a = q'_a \in Q_A$$

Exercise

A:



B:



1. Explain in words the transformation the literal transducers do to strings.

2. What is A(abacabaca)? What is B(abacabaca)? What is A∘B(abacabaca)? What is A∘B(acabacaba)? Note $f \circ g(x) = g(f(x))$.

3. Compute the literal transducer which computes the composition of A∘B. Using this machine, check A∘B(abacabaca) and A∘B(acabacaba).

Co-emission product of stochastic regular languages

For stochastic regular languages there is a probability distribution I over the states which indicates the probability starting there. So

$$\sum_{q \in Q} I(q) = 1$$

There is also a probability associated to each state in Q for being final, denoted F.

For clarity, transitions in stochastic automata are written as (q, a, p, r) where q is the origin state, r is the terminal state, and $a \in \Sigma \cup \{\epsilon\}$ and p is a probability between 0 and 1. Note that for each $q \in Q$

$$\sum_{\exists a,p,r[(q,a,p,r)\in\delta]} p + F(q) = 1$$

Finally, $\delta\big((q_a, q_b), \sigma, p, (q'_a, q'_b)\big)$ if and only if **both** $(q_a, \sigma, p_a, q'_a) \in \delta_A$ **and** $(q_b, \sigma, p_b, q'_b) \in \delta_B$ **and** $p = \frac{p_a \times p_b}{Z(q)}$ where $Z(q)$ is a normalization term. Details on the co-emission product are given in [1, 2].

References

[1] Enrique Vidal, Franck Thollard, Colin de la Higuera, Francisco Casacuberta, and Rafael C. Carrasco. Probabilistic finite-state machines-part I. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1013–1025, 2005.

[2] Enrique Vidal, Frank Thollard, Colin de la Higuera, Francisco Casacuberta, and Rafael C. Carrasco. Probabilistic finite-state machines-part II. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1026–1039, 2005.