Def: Let DFAs

$A_1 := \langle Q_1, \Sigma, \delta_1, q_1, F_1 \rangle$

$A_2 := \langle Q_2, \Sigma, \delta_2, q_2, F_2 \rangle$

be given

We define the product DFA

$$A = \langle Q_1 \times Q_2, \Sigma, \delta, (q_1, q_2), F_1 \times F_2 \rangle$$

where, for all $q \in Q_1$ and $r \in Q_2$ and $a \in \Sigma$

$$\delta((q, r), a) = (\delta_1(q, a), \delta_2(r, a)).$$

Notation: $A := A_1 \wedge A_2$

Lemma: Let A_1, A_2, A be as above, and let $w \in \Sigma^*$ be arbitrary. Let $A_1(w)$ be the end state of A_1's computational trace on input w $(A_1(w) \in Q_1)$. Similarly for A_2 and A.

Then $A(w) = (A_1(w), A_2(w))$.

Proof: Induction on $|w|$ (length of w).

Base case: $|w| = 0$. Then $w = \varepsilon$.

And $A_1(\varepsilon) = q_1$ and $A_2(\varepsilon) = q_2$

Also, $A(\varepsilon) = (q_1, q_2)$

$= (A_1(\varepsilon), A_2(\varepsilon))$. // base case

Inductive case: $|w| > 0$. Then there exist unique $x \in \Sigma^*$ and unique $a \in \Sigma$ such that $w = xa$. (such principal prefix of w)

Then $|x| = |w| - 1 < |w|$.

By inductive hypothesis,

$$A(x) = (A_1(x), A_2(x))$$

Then

$$A(w) = A(xa) \overset{\text{"obvious"}}{=} \delta(A(x), a)$$

$$\overset{\text{ind hyp}}{=} \delta((A_1(x), A_2(x)), a)$$

$$\underset{\text{def of } \delta}{=} (\delta_1(A_1(x), a), \delta_2(A_2(x), a))$$

$$\underset{\text{"obvious" fact applied to } A_1 \text{ \& } A_2}{=} (A_1(xa), A_2(xa))$$

$$\underset{w = xa}{=} (A_1(w), A_2(w)) \quad [\text{match stmt for } w]$$

$\therefore$ By induction, Lemma holds for all $w \in \Sigma^*$. ▨

Cor: A_1, A_2, A as above.

$$L(A) = L(A_1) \cap L(A_2)$$

Proof: For any $w \in \Sigma^*$

$w \in L(A) \overset{\text{def of } L(A)}{\iff} A$ accepts w

$\overset{\text{def of A accepting}}{\iff} A(w) \in F_1 \times F_2$

$\iff (A_1(w), A_2(w)) \in F_1 \times F_2$ — $A(w) = (A_1(w), A_2(w))$ by the Lemma

$\iff A_1(w) \in F_1$ & $A_2(w) \in F_2$ — def of cartesian product

$\iff A_1$ accepts w and A_2 accepts w — def of acceptance

$\iff w \in L(A_1)$ and $w \in L(A_2)$ — def of $L(\cdots)$

$\iff w \in L(A_1) \cap L(A_2)$. — def of $\cap$

To summarize: $\forall w \in \Sigma^*$

$$w \in L(A) \iff w \in L(A_1) \cap L(A_2)$$

$\therefore L(A) = L(A_1) \cap L(A_2)$. ▨

Cor: If L_1 & L_2 are regular languages, then $L_1 \cap L_2$ is regular.

That is REG_Σ is closed under intersection.

Cor: REG_Σ is closed under union and all other Boolean set ops.

Ex: $L_1 \cup L_2 = \overline{\overline{L_1} \cap \overline{L_2}}$

$L_1 - L_2 := \{w : w \in L_1 \text{ \& } w \notin L_2\}$

$L_1 - L_2 = L_1 \cap \overline{L_2}$

$L_1 \triangle L_2 := (L_1 - L_2) \cup (L_2 - L_1)$

$= \{w : w \in L_1 \text{ or } w \in L_2 \text{ but not both}\}$

$= (L_1 \cup L_2) - (L_1 \cap L_2)$

(symmetric difference)

Note: $L_1 = L_2 \iff L_1 \triangle L_2 = \emptyset$.

Application: string matching.

Def: $w, x \in \Sigma^*$ (Σ arbitrary alphabet)

say that

x is a prefix of w if $\exists y \in \Sigma^*,\ w = xy$

x is a suffix of w if $\exists y \in \Sigma^*\ w = yx$.

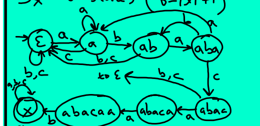
x is a substring of w if x is a prefix of some suffix of w, equiv, $\exists y, z \in \Sigma^*,\ w = yxz$

Fix $x \in \Sigma^*$ want a DFA S_x that accepts $w \in \Sigma^*$ iff w has x as a substring.

S_x = "search for x in w"

Ex: $x = abacaab$ ($\Sigma = \{a,b,c\}$)

S_x: 8 states ($8 = |x| + 1$)

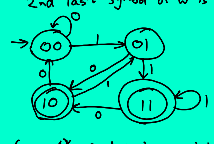


Idea: State of S_x represents the longest prefix of x that is a suffix of the input read so far.

Knuth-Morris-Pratt algo to build S_x given x.

Ex: $\Sigma = \{0,1\}$. DFA for

$L := \{w \in \Sigma^* : |w| \geq 2 \text{ \& 2nd last symbol of } w \text{ is } 1\}$



$L' := \{w \in \Sigma^* : \text{3rd to last symbol of } w \text{ is } 1\}$

$8 = 2^3$ states is necessary & sufficient for a DFA recog. L'.

Nondeterminism:

Informally: an NFA (nondet. finite automaton) has an unrestricted transition diagram (and # of edges leaving a state with a given label (incl. 0))

NFA recognizing L':