Finite State Machines

- G is a finite SM if $|X| < \infty$.

  Notations: ε-NFSM, NFSM, DFSM.

- Language model equivalence of ε-NFSM and NFSM:

  Given ε-N.FSM G, construct NFSM $G' := (X, \Sigma, \alpha', x_0, X'_m)$

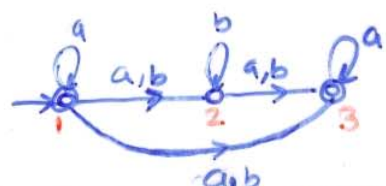
$$\alpha'(x,\sigma) := \varepsilon^*_G\left(\alpha\left(\varepsilon^*_G(x), \sigma\right)\right) \doteq \alpha^*(x,\sigma)$$

$$X'_m := \begin{cases} X_m \cup x_0 & \text{if } X_m \cap \varepsilon^*_G(x_0) \neq \phi \\ X_m & \text{otherwise.} \end{cases}$$

Then $L_m(G') = L_m(G)$; $L(G') = L(G)$.

Example:

G



G'

- Language model equivalence of NFSM and DFSM:

  Given NFSM $G := (X, \Sigma, \alpha, x_0, X_m)$, construct DFSM

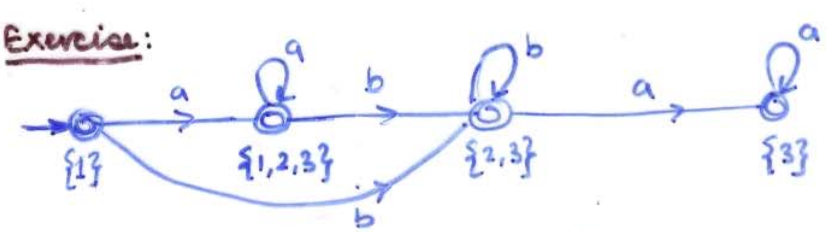
$$\mathcal{G} := (\mathcal{X}, \Sigma, \hat{\alpha}, \{x_0\}, \mathcal{X}_m)$$

$$\mathcal{X} := 2^X, \ \mathcal{X}_m := \{\hat{x} \in \mathcal{X} \mid \hat{x} \cap X_m \neq \phi\}$$

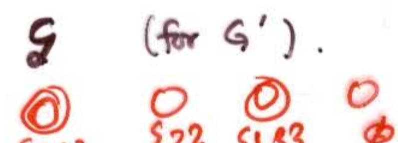
$$\hat{\alpha}(\hat{x}, \sigma) := \bigcup_{x \in \hat{x}} \alpha(x,\sigma)$$

Then $L_m(\mathcal{G}) = L_m(G)$ and $L(\mathcal{G}) = L(G)$. Known as power set construction.

Exercise:



$\mathcal{G}$ (for G').



- Remark: A deterministic DES with finite states can always be modeled as a DFSM.

Regular Languages

- We will now characterize the class of languages that can be represented by FSMs. (useful for development of algorithms).

- Regular Class($\mathcal{L}_R$): $\quad \phi, \{\varepsilon\}, \{\sigma\} \in \mathcal{L}_R \subseteq 2^{\Sigma^*}$

$$K, K_1, K_2 \in \mathcal{L}_R \Rightarrow K_1 + K_2, K_1 \cdot K_2, K^* \in \mathcal{L}_R.$$

For simplicity of notation, regular expressions are used for regular langs.

- $\phi, \varepsilon, \sigma$ are regular expressions
- r, r_1, r_2 regular expressions $\Rightarrow r_1 + r_2, r_1 r_2, r^*$ regular expressions.

Example: $(a+b)^*$, $a^* b$, etc. are regular
$+a$, $a \cdot + 3$ not regular.

$\boxed{\text{Given a DFSM } G \text{, there exists a regular exp. } r \text{ such that } L(r) = L_m(G). \text{ Conversely, given regular exp. } r \text{, there exists DFSM } G \text{ s.t. } L_m(G) = L(r).}$

Proof: ($\Rightarrow$) Label states of G by $1, \ldots, m$. Define regular expressions inductively for each $i, j \leq m$:

$$r_{ij}^0 = \begin{cases} +_{\{\sigma \in \Sigma \mid \alpha(i,\sigma) = j\}} \sigma & \text{if } i \neq j \\ \left(+_{\{\sigma \in \Sigma \mid \alpha(i,\sigma) = j\}} \sigma\right) + \varepsilon & \text{otherwise} \end{cases}$$

$$r_{ij}^k = r_{ik}^{k-1} \left(r_{kk}^{k-1}\right)^* r_{kj}^{k-1} + r_{ij}^{k-1} \quad \forall k \leq m$$

Then $L(r_{ij}^k)$ = set of strings starting from i, ending at j, and visiting states with label no larger than k.

Clearly, $L(r_{ij}^k)$ is regular and $L_m(G) = L\left(+_{j \in X_m} r_{1j}^m \right)$

Regular language: Equivalence with DFSM (ctnd.)

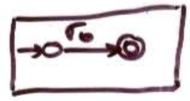
($\Leftarrow$) Since DFSM is "lang. equivalent" to ε-NFSM, it suffices to show existence of an ε-NFSM G with $L_m(G) = L(r)$. Shown by induction on number of operations in r:

base step: # of operations $= 0$

$\Rightarrow r = \phi$, or $r = \varepsilon$, or $r = \sigma_0$ for some $\sigma_0 \in \Sigma$.

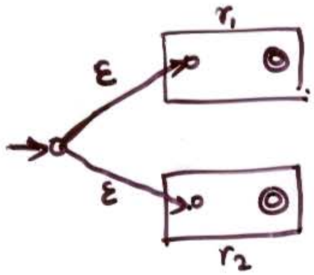
$r = \phi$

$r = \varepsilon$

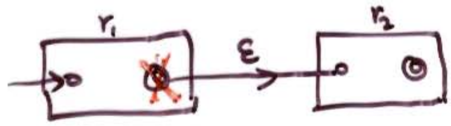


$r = \sigma_0$

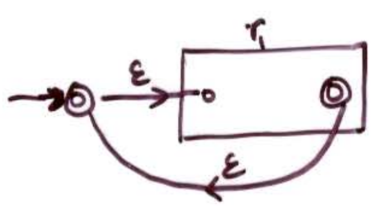
Induction step: suppose $r = r_1 + r_2$; then



Next suppose $r = r_1 \cdot r_2$; then



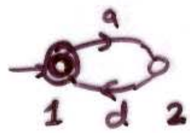
Finally suppose $r = r_1^*$; then



- Corollary: Given a language model (K_m, K), there exists a DFSM G such that $(L_m(G), L(G)) = (K_m, K)$ iff K_m and K both are regular.

Regular language : Equivalence with DFSM

Example:



$L_m(G) = L(r_{11}^2)$

$$r_{11}^0 = \varepsilon$$

$$r_{12}^0 = a$$

$$r_{21}^0 = d$$

$$r_{22}^0 = \varepsilon$$

$$r_{11}^1 = r_{11}^0 (r_{11}^0)^* r_{11}^0 + r_{11}^0 = \varepsilon(\varepsilon)^* \varepsilon + \varepsilon = \varepsilon$$

$$r_{12}^1 = r_{11}^0 (r_{11}^0)^* r_{12}^0 + r_{12}^0 = \varepsilon(\varepsilon)^* a + a = a$$

$$r_{21}^1 = r_{21}^0 (r_{11}^0)^* r_{11}^0 + r_{21}^0 = d(\varepsilon)^* \varepsilon + d = d$$

$$r_{22}^1 = r_{21}^0 (r_{11}^0)^* r_{12}^0 + r_{22}^0 = d(\varepsilon)^* a + \varepsilon = da + \varepsilon$$

$$r_{11}^2 = r_{12}^1 (r_{22}^1)^* r_{21}^1 + r_{11}^1 = a(da+\varepsilon)^* d + \varepsilon = a(da)^* d + \varepsilon = (ad)^*$$

Definitions: $Re_G(x)$ = set of reachable states from x in G

$$= \{ x' \in X \mid \exists\, s \in \Sigma^* \text{ s.t. } x' \in \alpha^*(x,s) \}.$$

- G is called accessible if $Re_G(x_0) = X$
- G is called co-accessible if $\forall x \in X : Re_G(x) \cap X_m \neq \phi$
- G is trim if it is accessible + co-accessible.
- It is always possible to construct a language equivalent trim state machine for any given state machine.