Properties of Regular Languages

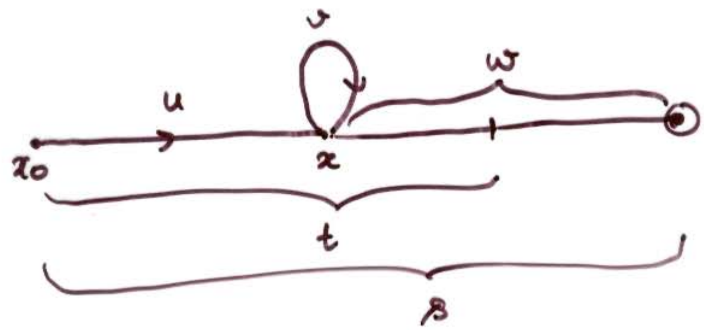
- Closure properties: Regularity preserved under choice, concatenation, Kleene Closure (by definition).

  K_1, K_2 regular $\Rightarrow$ $K_1 \cap K_2$ regular (consider $G_1 || G_2$)

  K regular $\Rightarrow$ K^c regular (consider $(\bar{G})^c$)

- Pumping Lemma: K regular $\Rightarrow$ $\exists\, m$ such that $\forall\, s \in K$, $|s| \geq m$: $s = uvw$ with $|uv| \leq m$, $|v| \geq 1$, and $uv^iw \in K$ for each i.

Proof: DFSM G s.t. $L_m(G) = K$. Set $m = |X|$.
Pick $s \in K$ with $|s| \geq m$. Let $t \leq s$ be such that $|t| = m$.
$\Rightarrow$ execution of t visits at least one state twice. x be 1^{st} such state.



- Application of Pumping Lemma: $K = \{a^i b^i \mid i \geq 1\}$ is not-regular.

Proof: Suppose for contradiction K is regular. Pick $s = a^m b^m \in K$, where m as in pumping lemma.

So $s = a^m b^m = uvw$, $|uv| \leq m$, $|v| \geq 1$, $uv^iw \in K$

$|uv| \leq m \Rightarrow uv \leq a^m \Rightarrow u = a^j$, $v = a^k$, $j+k \leq m$, $k \geq 1$

so $w = a^{m-(j+k)} b^m$.

Choose $i = 2m$. Then $uv^iw = a^j a^{2km} a^{m-(j+k)} b^m = a^{2km+m-k} b^m$.

Since $m-k \geq 0$ & $k \geq 1$, $2km + m - k \geq 2m$, a contradiction.

Review of equivalence relation

Given a set X, a relation R on X is a subset $R \subseteq X \times X$

$(x, y) \in R$, then x related to y, written xRy

Example: $X = \mathbb{N}$, set of naturals, xRy iff $x \bmod 5 = y \bmod 5$

$4R9$, $9R14$, $2 \not R 0$

R is equivalence relation if (i) reflexive : xRx
(ii) symmetric : $xRy \Rightarrow yRx$
(iii) transitive : $xRy, yRz \Rightarrow xRz$.

Examples: mod 5 relation is an equivalence
"brother" relation is transitive but not symmetric

An equivalence relation R denoted by $\cong_R$.

Equivalence class or coset $[x]_R \subseteq X$ of $x \in X := \{ y \mid y \cong_R x \}$

Example: $[4]_{\text{mod } 5} = \{4 + 5k \mid k \in \mathbb{N}\} = \{4, 9, 14, 19, 24, \ldots\}$

Thm: The set of all equivalence classes $\{[x]_R \mid x \in X\}$ form a partition of X

X 

(i) $\bigcup_{x \in X} [x]_R = X$ covers X (obvious)

(ii) $[x]_R \neq [y_R] \Rightarrow [x]_R \cap [y_R] = \phi$ different $\Rightarrow$ pairwise disjoint

suppose for contradiction, $[x]_R \cap [y_R] \neq \phi$.
Pick $z \in [x]_R \cap [y_R] \Rightarrow z \cong_R x \wedge z \cong_R y \Rightarrow x \cong_R y \Rightarrow [x]_R = [y]_R$.

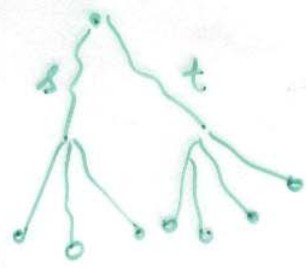
Example:
$[0]_{\text{mod } 3} = \{0, 3, 6, 9 \ldots\}$
$[1]_{\text{mod } 3} = \{1, 4, 7, 10, \cdots\}$ $\Rightarrow$ pair-wise disjoint
$[2]_{\text{mod } 3} = \{2, 5, 8, 11, \cdots\}$

$[0]_{\text{mod } 3} \cup [1]_{\text{mod } 3} \cup [2]_{\text{mod } 3} = \mathbb{N} \Rightarrow$ covers $\mathbb{N}$

Index of $\cong_R$ is no. of distinct equivalence classes of $\cong_R$.

Myhill Nerode Characterization

- Another characterization of regular languages



- Equivalence relation on Σ^* induced by K:

$$[s \cong t\ (R_K)] \Leftrightarrow [K\backslash\{s\} = K\backslash\{t\}]$$

$$\Leftrightarrow [\forall u \in \Sigma^*:\ su \in K \Leftrightarrow tu \in K]$$

Note: $s \cong t\ (R_K) \Rightarrow \forall u:\ su \cong tu\ (R_K)$

R_K is "right invariant" (wrt concatenation)

- Equivalence relation on Σ^* induced by G:

$$[s \cong t\ (R_G)] \Leftrightarrow [\alpha(x_0, s) = \alpha(x_0, t)] \vee [\alpha(x_0, s), \alpha(x_0, t) \text{ undefined}]$$

Note: $s \cong t\ (R_G) \Rightarrow [s \cong t\ (R_{L(G)})] \wedge [s \cong t\ (R_{L_m(G)})]$

R_G "refines" $R_{L(G)}$ and $R_{L_m(G)}$

- Example: $K = (ad)^*$

$[\varepsilon]\ (R_K) = (ad)^*$
$[a]\ (R_K) = a(da)^*$
$[d]\ (R_K) = \Sigma^* - pr(K)$

Myhill Nerode Characterization

<u>Thm</u> : The following are equivalent:

1. K is regular
2. K is union of some equivalence classes of a right-inavariant equivalence ~~class~~ relation of finite index
3. R_K is finite index

<u>Pf</u>: $(1 \Rightarrow 2)$ Let G be a DFSM with $L_m(G) = K$.

$$K = \{ [s](R_G) \mid s \in K \} \Rightarrow R_G \text{ is the required equivalence}$$

$(2 \Rightarrow 3)$ Suppose R is the given equivalence relation.

Suffices to show that R refines R_K, i.e., $s \cong t\,(R) \Rightarrow s \cong t\,(R_K)$

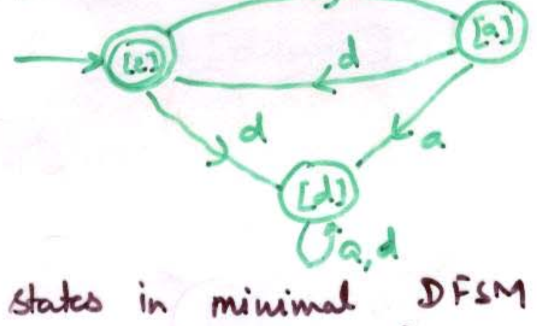
$$s \cong t\,(R)$$
$$\Rightarrow \forall u: su \cong tu\ (R) \qquad (R \text{ right-inv.})$$
$$\Rightarrow \forall u: su \in K \Leftrightarrow tu \in K \qquad (K \text{ union of equivalence classes of } R)$$
$$\Rightarrow s \cong t\,(R_K)$$

$(3 \Rightarrow 1)$ Construct a DFSM G:

$$X = \{ [s](R_K) \mid s \in \Sigma^* \}$$
$$X_m = \{ [s](R_K) \mid s \in K \}$$
$$x_0 = [\varepsilon](R_K)$$
$$\alpha([s](R_K), \sigma) = [s\sigma](R_K)$$

This implies: # of states in minimal DFSM $\leq |R_K| - 1$

<u>Example</u>: $K = (ad)^*$.



$[\varepsilon](R_K) = (ad)^*$

$[a](R_K) = a(da)^*$

$[d](R_K) = \Sigma^* - pr(K)$.

<u>Remark</u>: # of states in minimal DFSM for $K = |R_K| - 1$.

(since R_G refines $R_{L_m(G)}$) $\Rightarrow$ (# of states in G) $+ 1 \geq |R_K|$
$\Rightarrow$ # of states in minimal DFSM $\geq |R_K| - 1$)

Algorithms for Regular languages.

Emptiness: $K = \phi$? Construct DFSM G such that $L_m(G) = K$.

$$[K=\phi] \Leftrightarrow [Re_G(x_0) \cap X_m = \phi], \text{ where}$$

$Re_G(x_0)$ = set of reachable states from x_0.

Containment: $[K_1 \subseteq K_2] \Leftrightarrow [K_1 \cap K_2^c = \phi]$.

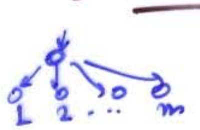
Algorithm for Computing Reachability:

1. Initiation Step: $Re_G^{-1}(x_0) := \phi$; $Re_G^{0}(x_0) = \{x_0\}$; $k = 0$

2. Iteration step:

$$Re_G^{k+1}(x_0) := Re_G^{k}(x_0) \cup \left\{ x \in X - Re_G^{k}(x_0) \;\middle|\; \exists x' \in Re_G^{k}(x_0) - Re_G^{k-1}(x_0), \sigma \in \Sigma : \alpha(x', \sigma) = x \right\}$$

3. Termination step: $Re_G^{k+1}(x_0) = Re_G^{k}(x_0)$, then $Re_G(x_0) = Re_G^{k}(x_0)$, stop; else $k := k+1$, goto step 2.



Complexity: G has m states $\Rightarrow O(m)$ steps to compute $Re_G(x_0)$.

$$f = O(g) \text{ if } f(\cdot) \le c\, g(\cdot)$$

Definitions: G is called accessible if $Re_G(x_0) = X$
G is called co-accessible if $Re_G(x) \cap X_m \neq \phi \quad \forall x \in X$.
trim = accessible + co-accessible.

- It is always possible to find a lang.-equivalent trim SM for G.
- minimal SM must be trim.

Algorithm for SM: minization.

- Follows from Myhill Nerode Characterization that a trim DFSM G is minimal iff

$$\left[\forall s : \left(\alpha(x,s), \alpha(x',s)\right) \in X_m \times X_m\right] \Rightarrow \left[x = x'\right]$$

- Aggregate x and x' if they are unequal.

- Consider a trim DFSM G with $L_m(G) \neq \Sigma^*$ (otherwise minimal is trivial)

  construct $\bar{G} = (\bar{X}, \Sigma, \bar{\alpha}, x_0, X_m)$

  Then a pair of states $(x, x') \in X_m \times (\bar{X} - X_m)$ must not be aggregated.

Initialize: $A_0 := (X_m \times X_m) \cup (\bar{X} - X_m \times \bar{X} - X_m)$ pairs which can be aggregated

$B_0 := (\bar{X} \times \bar{X}) - A_0$ pairs which ~~cannot be~~ must not be aggregated

$k := 0$

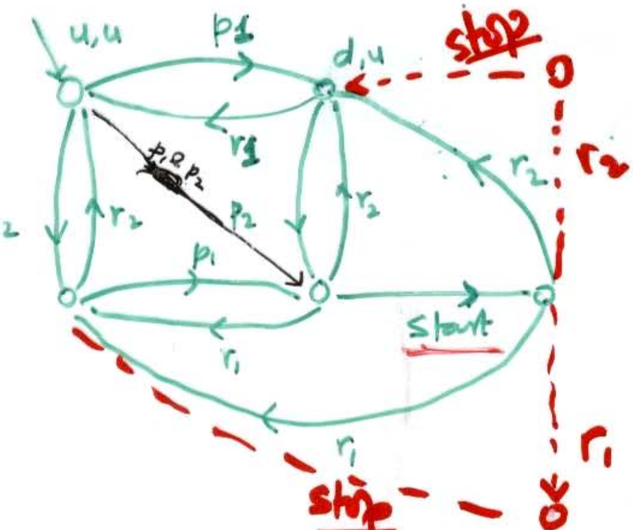
Iteration: $A_{k+1} := A_k - \{(x, x') \mid \exists \sigma \text{ s.t. } (\alpha(x,\sigma), \alpha(x',\sigma)) \in B_k\}$

$B_{k+1} := (\bar{X} \times \bar{X}) - A_{k+1}$

Termination: stop if $A_{k+1} = A_k$; else $k := k+1$ and iterate.

"Button tie-down problem"

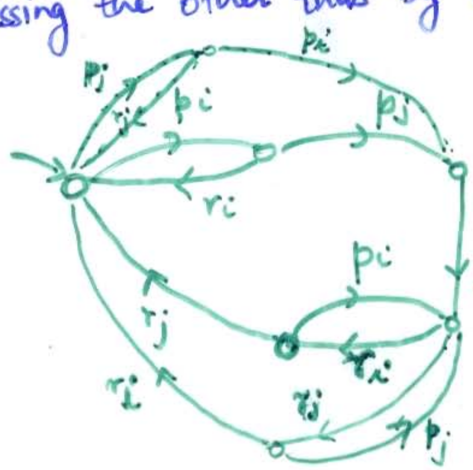
- Safety feature provided in potentially hazardous machines (saw mill, die press) by requiring machine be only allowed to start when both hands are pressing a push-button (cann't start machine by one hand)



p_i = push$_i$
r_i = release$_i$
u = "up"
d = "down"

uncontrolled plant

- Spec: Successive start events must be blocked unless separated by execution of both the push-button events (cann't start the machine by taping down one button, and pressing the other thus by using only one hand)



desired specification

Source: "Basic Control Systems Eng.", Prentice Hall, 1997