Properties of regular languges

Effective properties, ie. algorithmic

Closure properties

1. Closed under union. L_1, L_2 regular $\Rightarrow L_1 \cup L_2$ regular
2. Closed under product L_1, L_2 regular $\Rightarrow L_1 \cdot L_2$ regular
3. Closed under Kleene* L regular $\Rightarrow L^*$ regular
4. Closed under complementation: L regular $\Rightarrow \overline{L}$ regular

   Proof Let $M = (Q, \Sigma, \delta, q_0, F)$ accept L regular
   Then $M' = (Q, \Sigma, \delta, q_0, Q-F)$ accepts $\overline{L}$,
   since a string x is accepted by M
   $\Leftrightarrow$ x is not accepted by M'.

5. Closed under intersection: L_1, L_2 regular $\Rightarrow L_1 \cap L_2$ regular

   Proof $L_1 \cap L_2 = \overline{(\overline{L_1} \cup \overline{L_2})}$

6. All finite sets are regular.

   Proof A single string is easily shown to be regular.
   Then apply property 1.

Properties 1-6 imply that regular sets are the smallest class containing all finite sets and "closed" under union, product (concatenation) complement & Kleene*.

Structure Preserving Maps

Example Let R be the real numbers, with operations addition $(+)$ and multiplication $(\cdot)$.

Let
$$f : R \to R$$
$$a \mapsto e^a$$

Note that:
$$f(a+b) = e^{(a+b)} = e^a \cdot e^b = f(a) \cdot f(b).$$

Such a structure preserving map between two "algebraic" sets is called a homomorphism

Example Let R be the real numbers and M_2 be two by two matrices. Consider addition in R and matrix addition in M_2.

Let
$$g : R \to M_2$$
$$x \longmapsto \begin{bmatrix} x & 0 \\ 0 & x \end{bmatrix}$$

Note that:
$$g(x+y) = \begin{bmatrix} x+y & 0 \\ 0 & x+y \end{bmatrix} = \begin{bmatrix} x & 0 \\ 0 & x \end{bmatrix} + \begin{bmatrix} y & 0 \\ 0 & y \end{bmatrix} = g(x) + g(y)$$

We consider homomorphisms and their generalization, called substitutions for strings.

Substitution Property

Let Σ and Δ be two alphabets.

Consider a mapping $f : \Sigma \to 2^{(\Delta^*)}$ (Range is the power set of Δ^*)

We can extend f to Σ^* by:

$$f(\lambda) = \{\lambda\}$$

$$f(x \cdot a) = f(x) \cdot f(a) \qquad x \in \Sigma^*, \; a \in \Sigma.$$

We can also extend f to the domain $2^{(\Sigma^*)}$ by:

$$f(L) = \bigcup_{x \in L} f(x) \quad \text{where } L \subseteq 2^{(\Sigma^*)}.$$

The mapping f is called a <u>substitution</u>

Example $\Sigma = \{0,1\}$ $\Delta = \{a,b,c\}$

Let $f(0) = |ab^*|$ and $f(1) = \{ac\}$.

Then

$$f(011) = |ab^* ac\, ac| \quad \text{or} \quad ab^* acac$$

and

$$f(011^*) = ab^* ac(ac)^*$$

If $f(a)$ is a regular language for $a \in \Sigma$, we call the substitution a regular substitution. We will only consider regular substitutions.

Theorem

Regular sets are closed under (regular) substitutions.

Proof: Let $R \subset \Sigma^*$ be a regular language.
We need to show that $f(R)$ is a regular language.
By definition $f(a)$ is a regular language for $a \in \Sigma$.

We call easily show that:

$$f(L_1 \cup L_2) = f(L_1) \cup f(L_2)$$
$$f(L_1 \cdot L_2) = f(L_1) \cdot f(L_2)$$
$$f(L^*) = (f(L))^*.$$

We can thus prove this by brocking up a regular language into primitive expressions and then applying the operations to get regular sets in the domain & range.

Definition A homomorphism (in Automata Theory) is a substitution h such that $h(a)$ is a single string in Δ, for $a \in \Sigma$. That is:

$$h : \Sigma \to \Delta^*$$
$$a \mapsto x \qquad x \in \Delta^*.$$

We can extend h to Σ^* by

$$h(xa) = h(x) \cdot h(a) \quad \text{for} \quad x \in \Sigma^*, a \in \Sigma.$$

and $h(\lambda) = \lambda$.

Note 1. If $w = a_1 a_2 \cdots a_n$, then $h(w) = h(a_1) \cdot h(a_2) \cdots h(a_n)$.

Note 2. For $L \subset \Sigma^*$, $h(L) = \{h(w) : w \in L\}$ called the homomorphic image of L.

Example $h(0) = ab$, $h(1) = b$, $h(2) = a$ for $\Sigma = \{0,1,2\}$

Then $h(0110) = abbbab$

$h(122) = baa$

Let $h : \Sigma \to \Gamma^*$ be a homomorphism.

By definition, $h^{-1}(w) = \{x \mid h(x) = w\}$ for $w \in \Gamma^*$.

Similarly $h^{-1}(L) = \{x \mid h(x) \in L\}$ for $L \subset \Gamma^*$.

Example $h : \Sigma \to \Gamma^*$, $\Sigma = \{0,1,2,3\}$, $\Gamma = \{a,b\}$.

$0 \mapsto abaab$

$1 \mapsto aabb$

$2 \mapsto abab$

$3 \mapsto aabb$.

$h^{-1}(aabb) = \{1,3\}$

$h^{-1}(\{aabb, abaab\}) = \{0,1,3\}$

$h^{-1}(ab) = \Phi = \{\ \}$.

Note that $h(00) = abaababaab$.

Then $h^{-1}(abaababaab) = 00$.

Would it be possible for $h(0)$ to be aba?

Theorems The class of regular languages is closed under homomorphisms and inverse homomorphisms.

Proof. A homomorphism is a substitution hence regular languages are closed under homomorphisms.

Let $h : \Sigma \to \Gamma^*$ be a homomorphism and consider L a regular language in Γ^*. There must exist a dfa $M_L = \langle Q, \Gamma, \delta_\Gamma, q_0, F \rangle$ that accepts L. Define M_Σ to be $\langle Q, \Sigma, \delta_\Sigma, q_0, F \rangle$ where $\delta_\Sigma(q, a) = \delta_\Gamma(q, h(a))$ for $a \in \Sigma$.

We can prove by induction on the length of x, that $x \in L_\Sigma \Leftrightarrow h(x) \in L_\Gamma$. Hence regular languages are closed under inverse homomorphisms.

<u>Examples of the use</u> of these theorems.

Example. Prove that $L = \{a^n b^n c^{2n} : n \geq 1\}$ is not regular.

Define $h(a) = 0 \quad h(b) = 0 \quad h(c) = 1$.

Now, $h(L) = \{0^{2n} 1^{2n} : n \geq 1\}$ which we can easily show is not regular. How?

Example. $L = \{a^n b a^n \mid n \geq 1\}$ is not regular.

Define

$$\begin{array}{ll} h_1(a) = a & h_2(a) = 0 \\ h_1(b) = ba & h_2(b) = 1 \\ h_1(c) = a & h_2(c) = 1 \end{array}$$

Note that: $h_2(h_1^{-1}(\{a^n b a^n \mid n \geq 1\}) \cap a^* b c^*) = \{0^n 1^n \mid n \geq 1\}$.

Note: $h_1^{-1}(L) = (a+c)^n\, b\, (a+c)^{n-1}$.

Quotients of Languages

Given L_1, L_2 try to define $\frac{L_1}{L_2}$ s.t. $\frac{L_1}{L_2} \cdot L_2 = L_1$
This does not quite work!
However we have:

The right quotient of L_1 with L_2 is:
$L_1/L_2 = \{ x : x \cdot y \in L_1 \text{ for some } y \in L_2 \}$

Example $L_1 = 1^*01 + 0^*$
$L_2 = 0^*1$

What is L_1/L_2? It includes $1^* + 0$
How about 1^*0?
How about 00?

Note that $L_1/L_2 \cdot L_2 \neq L_1$.

Theorem. The class of regular sets is closed under quotients with arbitrary sets. (But this closure is not effective.)

Theorem The class of regular sets is closed under quotients with ~~arbit~~ regular sets, and this closure is effective.

The Pumping Lemma for Regular Languages

Suppose a regular language L is accepted by a dfa with n states, and let $m \geq n$.

Consider the string $w = a_1 a_2 a_3 a_4 \ldots a_m$ and assume $w \in L$. Using the dfa we know that:

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \xrightarrow{a_3} q_3 \xrightarrow{a_4} q_4 \cdots \xrightarrow{a_m} q_m$$

where q_0 is the initial state, q_m is a final state.
Note that each $a_i \in \Sigma$ and $q_i \in Q$.
Since $m \geq n$, some q_i must repeat, an $\exists\, x, y, z$ s.t.

$$q_0 \xrightarrow[x]{} q_i \underbrace{\xrightarrow[y]{} q_i}_{\text{This may repeat 1 or more times.}} \xrightarrow[z]{} q_m$$

for strings x, y, z with $|xy| \leq n$, $|y| \geq 1$ and furthermore $xy^iz \in L$ for all $i \geq 0$. Note that the "middle part" y can be "pumped."

Lemma. (The pumping lemma)
Let L be a regular language. Then, there exists a constant n s.t. if w is any word in L with $|w| \geq n$, then $w = xyz$ with $|xy| \leq n$ and $|y| \geq 1$ and for $i \geq 0$, we have $xy^iz \in L$.

(Note: such a string actually exists if L is an infinite language)

Example. Let $\Sigma = \{a, b\}$. Let $L = \{ w \in L^*, \text{ s.t. } N_a(w) < N_b(w) \}$
Prove that L is not regular.

Proof. Let n be as in the lemma. Let w be
$a^n b^{n+1}$, an element of L.
$\therefore a^n b^{n+1} = xyz$ with $|xy| \le n$ and $|y| \ge 1$.
and $xy^i z \in L$.
However, by construction xy must be all a's.
$\therefore$ y must be all a's.
$\therefore$ for $i = 2$ $xyyz \in L$, etc.
and xy^i must have more than n a's for
some i. and $\therefore$ xy^iz must have more a's than b's
Contradiction!

Example Prove that $L = \{ 0^{i^2} \mid i \ge 1 \}$ is not regular.
Assume L is regular. Choose n as in the lemma.
Let $w = 0^{n^2}$. By the lemma,
$w = xyz$ with $|xy| \le n$ and $|y| \ge 1$ & $xy^iz \in L$ $i \ge 0$,
But $|xy^2z| \le \underbrace{n^2}_{\text{length of } w} + \underbrace{n}_{\text{max length } y} = n(n+1) < (n+1)^2$

$\therefore xy^2z \notin L$, a contradiction.

Decision Algorithms for Regular Languages

Note: A language is given in standard representation if it is described by an fsa, regular grammar or regular expression.

Lemma. The set of strings accepted by an fsa with n states is:

(a) nonempty $\Leftrightarrow$ the fsa accepts a string with $|w| < n$.

Proof. $\Leftarrow$ (obvious)

Proof $\Rightarrow$. Use the pumping lemma. Let w be the shortest string accepted and assume $|w| \geq n$. By the pumping lemma $w = xyz$ is accept $\Rightarrow$ xz is accepted. But xz is then shorter than the shortest string.

(b) infinite $\Leftrightarrow$ the fsa accepts w with $n \leq |w| < 2n$.

Proof $\Leftarrow$ obvious using pumping lemma.

Proof $\Rightarrow$ By pumping lemma $\exists\, w$ s.t. $|w| \geq n$ and $w \in L$. Suppose $|w| \geq 2n$. and is the shortest such string.

$\therefore$ $w = xyz$ & xz is shorter than w.

$\therefore$ $|xz| < 2n$, and $|xz| \geq n$ since or y was removed and $1 \leq |y| \leq n$.

Theorem (Membership)

Given a regular language L and $w \in \Sigma^*$, there exists an algorithm for determing if $w \in L$.

Proof. Test w on the fsa for L.

Theorem (Empty, Finite, Infinite)

(a) There exists an algorithm to decide if a regular language L is empty.

Proof. Check strings of length upto n.

(b) There exists an algorithm to check if a regular language L is infinite.

Proof. Check all strings w with length $n \le |w| < 2n$.

(c) There exist an algorithm to check if a regular language L is finite.

Proof. Same as (b).

Theorem (Accept the Same Language)

There exists an algorithm to decide if two regular languages L_1 and L_2 are equal ($L_1 = L_2$).

Proof. Consider $L_3 = (L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2)$

Check if L_3 is empty. If so, the languages L_1 & L_2 are equal.

Note. Pictorially: