National Institute of Standards and Technology

- Each year bad programming costs the U.S. economy more than $60 billion in revenue. In other words, what we Americans lose each year to faulty code is greater than the gross national product of most countries.
- ***One of the great ironies is that computer science should be the most mathematical of all the sciences,***
- ***Computers are essentially mathematical engines that should behave in precisely predictable ways.***
- ***yet software is some of the flakiest engineering there is, full of bugs and security issues.***

Steve Omohundro, Stanford University Phi Beta Kappa, and Ph.D. in physics U.C. Berkeley.

Problems:

Alphabets, Strings, and Languages

communicate information

Examples of an Alphabet

$\sum = \{a, b, \ldots, z\}$ - alphabetic

$\sum = \{a, b, \ldots, z\} \cup \{0, 1, \ldots, 9\}$ - alphanumeric

$\sum = \{0, 1\}$ – bit values

For $\sum$ a nonempty set $= \{a\}$

$\sum^* = \{\lambda, a, aa, aaa, \ldots\}$

Each element is unique

$\sum^*$ is countably infinite

Problems:

Alphabets, Strings, and Languages

Definition 2.11

Alphabet Σ is a set of symbols

Language is a set of strings over alphabet Σ

Σ^* is a countable set of all strings over Σ

λ, an empty string $\in \Sigma^*$

recursive strings

Basis: $\lambda \in \Sigma^*$

Recursive Step:

if $w \in \Sigma^*$ and $\forall\, a \in \Sigma$ then $wa \in \Sigma^*$

Problems:

Closure:

$w \in \Sigma^*$ iff *w obtained from λ with finite number of recursive steps*

Alphabets, Strings, and Languages

For string $w \in \sum^*$, $w^0 = \lambda \in \sum^*$

$w^k = w \ldots w\,(k \text{ times}) = w^{k-1}w$, $w \in \sum^*$, $\forall k > 0$

For symbol $a \in \sum$, $a^0 = \lambda \in \sum^*$

$a^k = a \ldots a\,(k \text{ times}) = a^{k-1}a$, $a \in \sum^*$, $\forall\, k > 0$

Problems:

Number strings of $length(k) = n^k$, $n = |\sum|$

$\sum = \{a, b, c\}$, elements of $\sum^*$ include:

Strings of Length 0: λ

Strings of length $1 = (3)^1 = 3$: *a, b, c*

Strings of length $2 = (3)^2 = 9$: *aa, ab, ac, ba, bb, bc, ca, cb, cc*

$\sum^* = \{\lambda, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, \ldots\}$

Alphabets, Strings, and Languages

Definition 2.1.3: String Concatenation

$u, v \in \sum^*$, concatenation of uv is a binary operation on $\sum^*$

Basis: if $length(v) = 0$ then $v = \lambda$ and $uv = u$

Recursive Step:

if $length(v) = n > 0$ then $v = wa$, $a \in \sum$, $w \in \sum^*$, $length(w) = n - 1$

and $uv = u(wa) = uwa$

Problems:

Concatenation example:

Associative but not Commutative Property

$\sum = \{a, b, c\}$

$u = ab$, $v = ca$, $w = bb$

$uv = abca$ $\qquad$ $vw = cabb$

$(uv)w = abcabb$ $\qquad$ $u(vw) = abcabb$

Alphabets, Strings, and Languages

String Reversal

$u \in \sum^*$, reversal of u, denoted u^R is defined as follows:

Basis:

if $length(u) = 0$ then $u = \lambda$, $\lambda^R = \lambda$

Problems:

Recursive Step:

If $length(u) = n > 0$ then

$u = wa$, $Length(w) = n - 1$, $a \in \sum$, and $u^R = \mathrm{a}w^R$

Example:

$(\lambda sam)^R = m(\lambda sa)^R = ma(\lambda s)^R = mas(\lambda)^R = mas\lambda = mas$

Alphabets, Strings, and Languages

Strings over Σ of length 3

Language Definitions

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots \; = \; \Sigma^* = \lambda \cup \Sigma \cup \Sigma\Sigma \cup \Sigma\Sigma\Sigma \cup \ldots$

Any language over alphabet $\Sigma \subseteq \Sigma^*$

Number of possible languages is number of subsets over $\Sigma^* = |\, P(\Sigma^*) \,|$

By definition, $\{a\}$ $a \in \Sigma$, $\Phi \subseteq \Sigma^*$, $\Sigma^* \subseteq \Sigma^*$ are **languages**

Language strings satisfy syntactical requirements.

Language specification requires unambiguous string description.
Infinite languages with simple syntax are described recursively.

Recursive Language Specification

Example 2.2.1: Language L *from* $\sum = \{a, b\}$,

strings begin with a and have ***even*** length.

Basis: ***aa***, ***ab*** $\in L$

recursive strings

Recursive Step:

if $u \in L$ then uaa, uab, uba, and $ubb \in L$

Concatenate two elements to right of previous string

Closure:

String $u \in L$ iff obtained from basis element with finite number of applications of recursive step

$$L = \{aa\}\{\Sigma\Sigma\}^* \cup \{ab\}\{\Sigma\Sigma\}^* = \{aa\}\{\Sigma^2\}^* \cup \{ab\}\{\Sigma^2\}^*$$

Recursive Language Specification

Language L *from* $\sum = \{a, b\}$

number ***b's*** equal number ***a's*** and ***b's*** follow the ***a's***

Basis: $\lambda \in L$

Recursive Step: *If* $\boldsymbol{x} \in L$ *then* $\boldsymbol{axb} \in L$

Closure: *String* $\boldsymbol{x} \in L$ iff obtained from basis element with finite number of applications of recursive step

$L_0 = \{\lambda\}$

$L_1 = L_0 \cup \{ab\} = \{\lambda, ab\}$

$L_2 = L_1 \cup \{ab, aabb\} = \{\lambda, ab, aabb\}$

$L_3 = L_2 \cup \{ab, aabb, aaabbb\}$

$=\{\lambda, ab, aabb, aaabbb\}$

omit duplicate strings

Recursive Language Specification

Stable ***M/M/1*** **Model:** Language L *from* $\sum = \{a, d\}$

number ***d's*** equal number ***a's*** and

n^{th} d in the string is preceded by exactly ***n a's***

Basis: $\lambda \in L$

Recursive Step: *If* $x \in L$ *then* axd, adx, *and* $xad \in L$

Closure: *String* $x \in L$ iff obtained from basis element with finite number of applications of recursive step

omit duplicate strings

$L_0 = \{\lambda\}$

$L_1 = L_0 \cup \{ad\} = \{\lambda, ad\}$

$L_2 = L_1 \cup \{aadd, adad\} = \{\lambda, ad, aadd, adad\}$

$L_3 = L_2 \cup \{aaaddd, adaadd, aaddad, aadadd, adadad\}$

$= \{\lambda, ad, aadd, adad, aaaddd, adaadd, aaddad, aadadd, adadad\}$

Recursive Language Specification

Problem 2.18: Language L *from* $\Sigma = \{a, b, c\}$ and length ≥ 4

Basis: $wxyz \in L, \forall\, w, x, y, z \in \Sigma$

$u \in L, \forall\ u \in \Sigma^4, |\Sigma^4| = 81$

Basis is set of all strings of length 4 from Σ
81 strings

Recursive Step:

if $u \in L$ then $ua \in L$, $ub \in L$, and $uc \in L$

(if $u \in L$ then $uv \in L, \forall\ v \in \Sigma$)

Closure:

String $u \in L$ iff obtained from basis element with finite number of applications of recursive step

$L = \{u \in \Sigma^* \mid |u| \geq 4\}$

Recursive Language Specification

Problem 2.25: Language L *from* $\sum = \{a, b, c\}$

Every b is followed by at least one c

Basis: $\lambda \in L$

Recursive Step:

if $u \in L$ then $ua \in L$, $uc \in L$, and $ubc \in L$

Closure:

String $u \in L$ iff obtained from basis element with finite number of applications of recursive step

Includes λ

$L = \{\{a\} \cup \{c\} \cup \{bc\}\}^* = \{a, c, bc\}^*$

Concatenating Alphabet Σ on Itself

Definition 2.2.2:

$\Sigma^* = \bigcup_{i=0}^{\infty} \Sigma^i$ = Kleene Star

strings of all lengths from alphabet Σ

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots = \lambda \cup \Sigma \cup \Sigma\Sigma \cup \Sigma\Sigma\Sigma \cup \ldots$

$\{\lambda\}$

$\Sigma^+ = \bigcup_{i=1}^{\infty} \Sigma^i = \Sigma\Sigma^*$

excludes λ

Concatenating Set on Itself

For set X, $X^0 = \{\lambda\}$,

Concatenation of X on itself $\boldsymbol{n}$ times, X^n, $X^0 = \{\lambda\}$

$X^k = X^{k-1}X = X\,X^{k-1}$, $k > 0$

$X^n = \{x_1, \ldots x_i \ldots, x_n \mid x_i \in X\}$

Set of $|X|^n$ distinct concatenations from X

$X^* = \bigcup_{i=0}^{\infty} X^i = X^0 \cup X^1 \cup X^2 \cup X^3 \cup \ldots = \lambda \cup X \cup XX \cup XXX \cup \ldots$

$X^+ = \bigcup_{i=1}^{\infty} X^i$, $X^+ = XX^*$

Concatenating Set on Itself

Example:

$X = \{a, b, c\}$

$X^0 = \{\lambda\}$

$X^1 = \{a, b, c\}$

$X^2 = \{aa, ab, ac, ba, bb, bc, ca, cb, cc\}$

$|X^3| = |X^2| * |X^1| = 3^3 = 3^2 3^1 = 9 * 3 = 27$ length(3) in $\sum^*$

Concatenating Set on Itself

Example:

$X = \{abb, ba\}$

$X^0 = \{\lambda\}$

$X^1 = \{abb, ba\}$

$X^2 = \{abb\underline{abb}, abbba, baabb, baba\}$

$X^3 = \{abb\underline{abb}abb, abb\underline{abb}ba, abbba\underline{abb}, abbbaba,$

$baabb\underline{abb}, baabbba, babaabb, bababa\}$

$|X^3| = |X^1| * |X^2| = 2^3 = 2^1 2^2 = 2 * 4 = 8$

Set of $|X|^n$ distinct strings concatenated from elements of X

Concatenating Set on Itself

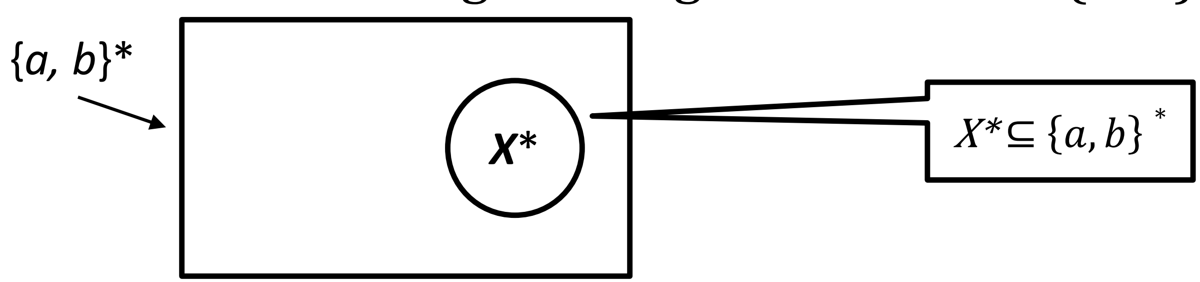
Concatenation of set $\{a, b\}$ on itself $= \{aa, ab, ba, bb\}$

For set $X = \{aa, ab, ba, bb\}$,

$X^{*} = \bigcup_{i=0}^{\infty} X^{i}$ = set of even length strings based on set $\{aa, ab, ba, bb\}$

$\{a, b\}^{*}$ = set of all odd and even length strings

$\{a, b\}^{*} - X^{*}$ = set of odd length strings based on set $\{a, b\}$



Concatenating Distinct Sets

For sets X and Y,

XY represents concatenation of X and Y

$XY = \{xy \mid x \in X, y \in Y\}$

Number of elements in $XY = |XY| = |X|\,|Y|$

$\mathrm{X} = \{a, b, c\}$, $\mathrm{Y} = \{\lambda, abb, ba\}$

$\mathrm{XY} = \{a, b, c, aabb, aba, babb, bba, cabb, cba\}$

$\mathrm{X} = \{a, b, c\}$, $\mathrm{Y} = \{abb, ba\}$

$\mathrm{XY} = \{aabb, aba, babb, bba, cabb, cba\}$

Regular Sets

Definition 2.3.1: ***Regular sets*** over alphabet Σ

Basis: Φ, $\{\lambda\}$, and $\{a \mid a \in \Sigma\}$ are ***regular sets*** over Σ , $|\Sigma|+2$ sets

Recursive Step:

if X and Y are ***regular sets*** over Σ then

$X \cup Y$, XY, and X^* are ***regular sets*** over Σ

regular set derivable from ***Basis*** *by* ***union, concatenation, Kleene star*** *operations*

Closure:

X is a ***regular set*** on Σ, if it can be obtained from basis with finite number of applications of recursive step

Regular language **is defined by a *regular set***

Regular Set Languages

Regular set (***language*** L) from $\sum = \{a, b, \ldots, z\}$

$L = \{w \in \sum^* \mid w$ is a regular substring$\}$

$L = \{\{a\} \cup \{b\} \cup \ldots \cup \{z\}\}^* \{r\}\{e\}\{g\}\{u\}\{l\}\{a\}\{r\}\{\{a\} \cup \{b\} \cup \ldots \cup \{z\}\}^*$

$\{a\}, \{b\}, \ldots, \{z\}$ are ***regular sets*** ***(Basis)***

$\{r\}\{e\}\{g\}\{u\}\{l\}\{a\}\{r\}$ is a ***regular set*** ***(Concatenation)***

$\{a\} \cup \{b\} \cup \ldots \cup \{z\}$ is a ***regular set*** ***(Union)***

$\{\{a\} \cup \{b\} \cup \ldots \cup \{z\}\}^*$ is a ***regular set*** ***(Kleene Star)***

$L = \sum^* \{r\}\{e\}\{g\}\{u\}\{l\}\{a\}\{r\} \sum^*$

regular set language ***(Concatenation)***

Regular Set Languages

Example 2.3.1 based on Example 2.2.5

$L = \{a, b\}^*\{bb\}\{a, b\}^*$ from $\sum = \{a, b\}$

$\{a\}$, $a \in \{a, b\}$ and $\{b\}$, $b \in \{a. b\}$ ***regular set*** ***(Basis)***

$\{b\}\{b\} = \{bb\}$ ***regular set*** ***(Concatenation)***

$\{a\} \cup \{b\} = \{a. b\}$ ***regular set*** ***(Union)***

$\{a. b\}^*$ ***regular set*** ***(Kleene Star)***

$\mathrm{L} = \{a, b\}^*\{bb\}\{a, b\}^*$ ***regular set language******(Concatenation)***

Regular Set Languages

Example 2.3.2

$L = \{a\}\{a, b\}^*\{b\}\{a, b\}^*\{a\}$ from $\sum = \{a, b\}$

$\{a\}$, $a \in \{a. b\}$ and $\{b\}$, $b \in \{a. b\}$ are ***regular sets*** ***(Basis)***

$\{a\} \cup \{b\} = \{a. b\}$ is a ***regular set*** ***(Union)***

$\{a. b\}^*$ is a ***regular set*** ***(Kleene Star)***

$L = \{a\}\{a, b\}^*\{b\}\{a, b\}^*\{a\}$

regular set language ***(Concatenation)***

Regular Set Languages

Example 2.2.5 *Language L from* $\sum = \{a, b\}$

bb embedded within any number of a's and b's

$$L = \{a, b\}^*\{bb\}\{a, b\}^* = \bigcup_{i=0}^{\infty}\{a, b\}^i \{bb\} \bigcup_{i=0}^{\infty}\{a, b\}^i$$

$$L = \sum^*\{bb\}\sum^*$$

Union Regular Set Languages

Example 2.2.6 *Language L from* $\sum = \{a, b\}$
any number of a's and b's preceded by aa or followed bb

$L_1 = \{aa\}\{a, b\}^*$ preceded by ***aa***

$L_2 = \{a, b\}^*\{bb\}$ followed by ***bb***

$L_3 = L_1 \cup L_2$ union of two languages

$L_3 = \{aa\}\sum^* \cup \sum^*\{bb\}$

Kleene Star Regular Set Languages

Example 2.2.7 *Language L from* $\Sigma = \{a, b\}$

$L_1 = \{bb\}$ $\quad$ $L_1^* = \{bb\}^*$

$L_2 = \{\lambda, bb, bbbb\}$ $\quad$ $L_2^* = \{\lambda, bb, bbbb\}^*$

$L_1^* = L_2^*$ $\quad$ strings with even number of b's

Regular Expressions

Definition 2.3.2: ***Regular expressions*** over alphabet Σ

Basis: Φ, λ, and $a \in \Sigma$ are ***regular expressions*** over Σ

Recursive Step:

if u and v are ***regular expressions*** over Σ then

$(u \cup v)$, uv and u^* are ***regular expressions*** over Σ

Closure:

u is a ***regular expression*** over Σ if obtained from basis elements with finite number of applications of recursive step

regular expressions omit squiglies of set notation

Rules for Regular expressions

Union and Concatenation are associative

Operator precedence

  Kleene Star

  Concatenation and Union

$u^+ = uu^*$ and $u^2 = uu$

Regular Expressions

Set $\{bawab \mid w \in \{a, b\}^*$ is a regular expression over $\{a, b\}$

	Regular Set	Expression	Justification
1	$\{a\}$	$\boldsymbol{a}$	Basis
2	$\{b\}$	$\boldsymbol{b}$	Basis
3	$\{b\}\{a\} = \{ba\}$	$\boldsymbol{ba}$	concatenate 2,1
4	$\{a\}\{b\} = \{ab\}$	$\boldsymbol{ab}$	concatenate 1,2
5	$\{a\}\cup\{b\} = \{a, b\}$	$\boldsymbol{a \cup b}$	union 1,2
6	{a, b}*	$\boldsymbol{(a \cup b)^*}$	Kleene star
7	{ba} {a, b}*	$\boldsymbol{ba(a \cup b)^*}$	concatenate 3,6
8	{ba} {a, b}*{ba}	$\boldsymbol{ba(a \cup b)^*ab}$	concatenate 7,4

regular expressions omit squiglies of set notation

Regular Expressions

Example 2.3.4

For strings over $\{a, b\}$ joined by string aa or bb

Regular Set

$\{\{a\}\cup\{b\}\}^*aa\{\{a\}\cup\{b\}\}^* \cup \{\{a\}\cup\{b\}\}^*bb\{\{a\}\cup\{b\}\}^*$

$\{a, b\}^*aa\{a, b\}^* \cup \{a, b\}^*bb\{a, b\}^*$

Regular Expression

$(a \cup b)^*aa\,(a \cup b)^* \cup (a \cup b)^*bb\,(a \cup b)^*$

Regular Expressions

Example 2.3.5

Regular Expression for ***all*** strings over $\{a, b\}$ with any number of a's before, between, and after b

$a^* b a^* b a^*$

Example 2.3.6

Regular Expression for ***at least 2 b's*** in each string

$a^* b a^* b (a \cup b)^*$

$(a \cup b)^* b a^* b a^*$

$(a \cup b)^* b (a \cup b)^* b (a \cup b)^*$

others ?

Regular Expressions

Example 2.3.9

Regular Expression for ***all*** strings over $\{a, b\}$ that does not end in aaa

$$(a \cup b)^*(b\lambda \cup ba \cup baa) \cup \lambda \cup a \cup aa$$

Regular Expressions

Example 2.3.8

Regular Expression for ***all*** strings over $\{a, b\}$ that begin with *ba*, end with *ab*, and contains substring *aa*

ba aa and *ab*

aa and *ab*

$$ba(a \cup b)^*aa(a \cup b)^*ab \cup baab \cup$$

$$baa(a \cup b)^*ab \cup ba(\mathrm{a} \cup \mathrm{b})^*aab$$

ba and *aa*

Regular Expressions

Regular Expression for Language L *from* $\sum = \{0, 1\}$ that contains substrings *00* ***or*** *11*

$$L = \underline{(0 \cup 1)^*}00\,\underline{(0 \cup 1)}^* \cup \underline{(0 \cup 1)^*}11\underline{(0 \cup 1)^*}$$

$$L = (0 \cup 1)^*(00 \cup 11)(0 \cup 1)^*$$

Regular Expressions

Regular Expression for Language L *from* $\sum = \{0, 1\}$

contains substrings *00* ***and*** *11*

$$L = (0 \cup 1)^*00(0 \cup 1)^*11(0 \cup 1)^* \cup$$
$$(0 \cup 1)^*11(0 \cup 1)^*00(0 \cup 1)^*$$

$\in \sum^*$

any substring in the middle

$$L = (0 \cup 1)^*(00(0 \cup 1)^*11 \cup (11(0 \cup 1)^*00)(0 \cup 1)^*$$

begins and ends with any substring

Regular Expressions

Example 2.3.10

Regular Expression $c^*(b \cup ac^*)^*$ over $\{a, b, c\}$ ***does not*** contain substring ***bc***

b concatenated with anything results in ***ba...*** (precludes ***bc***)

$c^*(b \cup ac^*)^* = c^*(b \cup ac^0 \cup ac^1 \cup ac^2 \ldots)^*$

$= c^*(b \cup a \cup ac \cup acc \ldots)^*$

$(b \cup ac^x)^y$

x = number of c's

y = number of concatenated substrings

$string\ acabacc = c^0((ac^1)(ac^0)(b)(ac^2))$

$string\ bbaaacc \equiv c^0((b)(b)(ac^0)(ac^0)(ac^2))$

$string\ bbaaacc \equiv c^0((b)^2(ac^0)^2(ac^2))$

Regular Expression Identities

1. $\Phi u = u\Phi = \Phi$	9. $u\,(v \cup w) = \mathrm{u}v \cup uw$
2. $\lambda u = u\lambda = u$	10. $(u \cup v)w = \mathrm{uw} \cup vw$
3. $\Phi^* = \Phi^0 = \lambda$	11. $(uv)^* u = \mathrm{u}(vu)^*$
4. $\lambda^* = \lambda$	12. $(u \cup v)^* = (u^* \cup v)^* =$
5. $u \cup v = v \cup u$	$= u^*(u \cup v)^* = (u \cup vu^*)^* =$
6. $u \cup \Phi = u$	$= (u^*v^*)^* = u^*(vu^*)^* =$
7. $u \cup u = u$	$= (u^*v)^* u^*$
8. $u^* = (u^*)^*$	

Regular Expressions

Example 2.3.11

For **Regular Expression** over $\{a, b\}$, ***aa*** is not a substring and any ***a*** must be followed by at least one ***b*** or terminate the string (any number of prefix ***b***'s satisfy the requirements)

$$b^*(ab^+)^* \cup b^*(ab^+)^*a = b^*(ab^+)^*\lambda \cup b^*(ab^+)^*a =$$

$$b^*(ab^+)^*(\lambda \cup a) \equiv b^*(abb^*)^*(\lambda \cup a) \equiv$$

$ab^+ = abb^*$

$$(b^*ab)^*b^*(\lambda \cup a) \equiv (b \cup ab)^*b^*(\lambda \cup a) \equiv$$

$u^*(vu^*)^* = (u^*v)^*u^*$

$(u^*v)^*u^* = (u \cup v)^*u^*$

$$(b \cup ab)^*(\lambda \cup a)$$

$(u \cup v)^*u^* = (u \cup v)^*$

Regular Expression → Language

Basis: regular expressions Φ, λ, and a correspond to languages Φ, $\{\lambda\}$. and $\{a \mid a \in \Sigma\}$ respectively

$L(\Phi) = \Phi$, $L(\lambda) = \{\lambda\}$. $\mathrm{L}(a) = \{a \mid a \in \Sigma\}$

Recursive Step:

if u and v are ***regular expressions*** over Σ then

$L(uv) = L(u)L(v)$,

$L(u \cup v) = L(u) \cup \mathrm{L}(v)$, and

Union
set $L(u)$ with set $L(v)$

$L(u^*) = L(u)^*$ are ***regular languages*** over Σ

Closure:

$L(u)$ and $L(v)$ are a ***regular languages*** over Σ if obtained from basis with finite number of recursive step applications

Regular Expression - Language

For each of the following, $\Sigma = \{a, b\}$

Example 2.3.1

$(a \cup b)^* bb(a \cup b)^*$ — *Regular Expression*

$L = \{a, b\}^*\{bb\}\{a, b\}^*$ — *Regular Set - Language*

Example 2.3.2

$a(a \cup b)^* b(a \cup b)^* a$ — *Regular Expression*

$L = \{a\}\{a, b\}^*\{b\}\{a, b\}^*\{a\}$ — *Regular Set - Language*

Example 2.3.3

$ba(a \cup b)^*ab$ — *Regular Expression*

Set $\{bawab \mid w \in \{a, b\}^*\}$ — *Regular Set*

$(a \cup b)^*$ regular expression $\rightarrow$ $w \in \{a, b\}^*$ regular set

Language - Regular Expression

$n_a(w)$ = number a's in w

Regular Expression for Language L *from* $\sum = \{0, 1\}$

$L_1 = \{n_0(w) = 0\}$

RE: 1^*

$L_2 = \{n_0(w) = 1\}$

RE: 1^*01^*

$L_3 = \{n_0(w) = 2\}$

RE: $1^*01^*01^*$

$L_4 = \{n_0(w) \text{ even}\}$

RE: $(1^*01^*01^*)^*$

$L_5 = \{n_0(w) = \text{odd}\} = L_2 L_4$

RE: $1^*01^*(1^*01^*01)^*$

$L_6 = \{n_0(w) \geq 2\}$

RE: $1^*01^*0(0 \cup 1)^*$

$L_7 = \{0\text{'s followed by at least one } 1\}$

RE: $(1 \cup 01^+)^*$

$L_8 = \{0\text{'s followed by at least one } 1\}$

RE: $(1 \cup 01^+)^+$ *does not include* λ

Recursive Language Specification

Language L *from* $\sum = \{a, b\}$

number ***b's*** equal number ***a's*** and ***b's*** follow ***a's***

Basis: $\lambda \in L$

Recursive Step: *If* $\boldsymbol{x} \in L$ *then* $\boldsymbol{axb} \in L$

Closure: *String* $\boldsymbol{x} \in L$ iff obtained from basis element with finite number of applications of recursive step

$L_0 = \{\lambda\}$

$L_1 = L_0 \cup \{ab\} = \{\lambda, ab\}$

$L_2 = L_1 \cup \{ab, aabb\} = \{\lambda, ab, aabb\}$

$L_3 = L_2 \cup \{ab, aabb, aaabbb\}$

$=\{\lambda, ab, aabb, aaabbb\}$

omit duplicate strings

Regular Sets/Expressions

a^*b^* is a ***Regular Expression*** for ***Regular Set*** $\{a\}^*\{b\}^*$

From **Definition 2..1.2,** Language over alphabet $\sum$ is subset of $\sum^*$

$L = \{\lambda, ab, aabb, aaabbb, \ldots\} \subseteq \{a\}^*\{b\}^*$

L is a language over $\{a, b\}$ but not a ***Regular Expression***

Regular Expressions

Definition 2.3.2: ***Regular expressions*** over alphabet Σ

Basis: Φ, λ, and $a \in \Sigma$ are ***regular expressions*** over Σ

Recursive Step:

if u and v are ***regular expressions*** over Σ then

$(u \cup v)$, ***uv*** and u^* are ***regular expressions*** over Σ

concatenate Regular Expressions

successive concatenations and unions

Closure:

u is a ***regular expression*** over Σ if obtained from basis elements with finite number of applications of recursive step

Search Strings

Operation	Symbol	Extended RE	Regular Expression
Concatenate		ab	ab
		$\{a\text{-}c\}\ \{AB\}$	$aA \cup aB \cup bA \cup bB \cup cA \cup cB$
Kleene Star	*	$\{ab\}^*$	$(a \cup b)^*$
Disjunction	|	$\{ab\}^*\vert A$	$(a \cup b)^* \cup A$
One or more	+	$\{ab\}+$	$(a \cup b)+$
Zero or one	?	$a?$	$(a \cup \lambda)$
One character	.	$a.a$	$\{a\}\Sigma\{a\} \rightarrow a(a \cup b)a \;\; for\ \Sigma = \{a, b\}$
n times	$\{n\}$	$a\{4\}$	$aaaa \equiv a^4$
n or more times	$\{n,\}$	$a\{4,\}$	$aaaaa^*$
n to m times	$\{n, m\}$	$a\{4, 6\}$	$aaaa \cup aaaaa \cup aaaaaa$

Search Strings

Extended RE	Regular Expression
computer	*computer*
$\{a\text{-}z\}\,\{0\text{-}9\}$	$a0 \cup a1 \cup \ldots \cup z8 \cup z9$
$\{a\text{-}z\}^*$	$(a \cup b \cup \ldots \cup z)^*$
$\{a \mid b \mid c\}^* \equiv \{a\text{-}c\}^*$	$(a \cup b \cup c)^*$
$a^* \mid b^* \mid c^*$	$a^* \cup b^* \cup c^*$
$\{a\text{-}z\}+$	$(a \cup b \cup \ldots \cup z)^+ \equiv (a \cup b \cup \ldots \cup z)\,(a \cup b \cup \ldots \cup z)^*$
$ab(c?)$	$ab(c \cup \lambda) \equiv abc \cup ab$
$(abc)?$	$abc \cup \lambda$
$a.c$	$\{a\}\Sigma\{c\} \rightarrow a(a \cup b \cup c)c \quad for\ \Sigma = \{a, b, c\}$
$a\{8\}$	$aaaaaaaa \equiv a^8$
$a\{8,\}$	a^8a^*
$a\{4, 6\}$	$aaaa \cup aaaaa \cup aaaaaa$

GREP Search Strings

[A-Z]	upper case letters
[a-z]	lower case letters
[A-Z a-z] or [a-z A-Z]	all letters
[0-9]	decimal digits
[A-Z a-z 0-9]	alphanumeric

3 or 4 characters concatenated with 3 or 4 digits

[A-Z] {3,4}[0-9] {3,4} – $(26^3 + 26^4)(10^3 + 10^4)$

or

3 or 4 digits concatenated with 3 or 4 characters

[0-9] {3,4} [A-Z] {3,4} – $(10^3 + 10^4)(26^3 + 26^4)$

GREP Search Strings

32 bit IP Address e.g. 255.255.255.255

RE: $(0 \cup 1)^{32}$

GREP: [01]{32}

([0-9] | [1-9] [0-9] | 1 [0-9] [0-9] | 2 [0-4] [0-9] | 25 [0-5])

0-9 | 10-99 | 100-199 | 200-249 | 250-255

Numeric Literals

53 9.37 -992 -49.06 4.0 4.

+49E-7 -49e7 -3.92E-3

[+-]?([0-9] | [1-9] [0-9]+)

single digit or two or more digits with no leading zeroes

optional sign

(.[0-9]*)?

optional floating point number

([Ee][+-]?([0-9] | ([1-9] [0-9]+)))?

optional scientific notation