Formal Language Theory
an Introduction

Prof. A. Morzenti

ALPHABET Σ : any ***finite*** set of symbols $\Sigma = \{a_1, a_2, \dots a_k\}$

cardinality of the alphabet $|\Sigma| = k$

String: a sequence ($\Rightarrow$ordered) of alphabet elements (possibly repeated)

Language: any set of strings

$\Sigma = \{a, b, \mathrm{c}\}$ $\quad L_1 = \{ab, ac\}$ $\quad L_2 = \{bc, bbc\}$ $L_3 = \{abc, aabbcc, aaabbbccc, \dots\}$

The strings of a language are called its ***sentences*** or ***phrases***

Language ***cardinality***: the number of its sentences

$$|L_2| = |\{bc, bbc\}| = 2 \qquad |\varnothing| = 0$$

Number of occurrences of a symbol in a string $|bbc|_b = 2$, $\quad |bbc|_a = 0$

With a slight *abuse of notation* sometimes we denote with Σ

both the alphabet and

the language of all strings of length 1

length of a string x: $|x|$
number of its elements

$$\boxed{\begin{array}{l} |bbc| = 3 \\ |abbc| = 4 \end{array}}$$

string equality : two strings are equal if and only if (***iff***, for short)

- have the same length
- their elements, from left to right, coincide

$x = a_1 a_2 \dots a_h \qquad y = b_1 b_2 \dots b_k$

$x = y$ iff $h = k$ and $a_i = b_i$ for all $i = 1 \dots h$

$bbc \neq bcb \neq bc$

OPERATIONS ON STRINGS /1

CONCATENATION (product): $x \cdot y$ or xy for short

$x = a_1 a_2 \dots a_h$ $y = b_1 b_2 \dots b_k$ $x \cdot y = xy = a_1 a_2 \dots a_h b_1 b_2 \dots b_k$

- associative $(xy)z = x(yz)$
- length $|xy| = |x| + |y|$

EMPTY STRING (or ***null string***) ε is the neutral element for concatenation

for any x, $x\varepsilon = \varepsilon x = x$

length of ε: $|\varepsilon| = 0$

NOTICE: $\boldsymbol{\varepsilon}$ is ***NOT*** the empty set: $\varepsilon \neq \varnothing$

SUBSTRINGS: if $x=uyv$ (NB: both u and v can be ε) then

- y is a substring of x
- y is a ***proper substring*** iff $u \neq \varepsilon$ or $v \neq \varepsilon$
- u is a ***prefix*** of x
- v is a ***suffix*** of x

EXAMPLES

if $x = abccbc$ then

prefixes: *a*, *ab*, *abc*, *abcc*, *abccb*, *abccbc*

suffixes: *c*, *bc*, *cbc*, *ccbc*, *bccbc*, *abccbc*

substrings: …, *bc*, *cc*, *cb*, *abc*, *bcc*, …

OPERATIONS ON STRINGS /2

REFLECTION x^R

$$x = a_1 a_2 \ldots a_h$$
$$x^R = a_h a_{h-1} \ldots a_2 a_1$$
$$(x^R)^R = x$$
$$(xy)^R = y^R x^R$$
$$\varepsilon^R = \varepsilon$$

$$x = atri \qquad x^R = irta$$
$$x = bon \qquad y = ton$$
$$xy = bonton$$
$$(xy)^R = y^R x^R = notnob$$

REPETITION: m-th power ($m \geq 1$) of string x: concatenation of x with itself m-1 times

$$x^m = \underbrace{xxx\ldots x}_{1\,2\,3\,\ldots\,m}$$

$$\begin{pmatrix}\text{inductive}\\ \text{definition}\end{pmatrix} \begin{cases} x^m = x^{m-1}x, & m > 0 \\ x^0 = \varepsilon \end{cases}$$

$$x = ab \quad x^0 = \varepsilon \quad x^1 = x = ab \quad x^2 = (ab)^2 = abab$$
$$y = a^3 = aaa \quad y^3 = a^3a^3a^3 = a^9$$
$$\varepsilon^0 = \varepsilon \quad \varepsilon^2 = \varepsilon$$

OPERATOR PRECEDENCE: repetition and reflection take precedence over concatenation

$$ab^2 = abb \qquad (ab)^2 = abab$$
$$ab^R = ab \qquad (ab)^R = ba$$

OPERATIONS ON LANGUAGES /1

OPERATIONS ARE TYPICALLY DEFINED ON A LANGUAGE
BY EXTENDING THE STRING OPERATION TO ALL ITS PHRASES

REFLECTION L^R: $L^R = \{ x \mid \exists y \, (y \in L \wedge x = y^R) \}$ def. by ***characteristic predicate***

Prefixes(L) $= \{ y \mid y \neq \varepsilon \wedge \exists x \, \exists z \, (x \in L \wedge x = yz \wedge z \neq \varepsilon) \}$ NB: *proper* prefixes

Prefix-free language L: no proper prefix of its phrases $\in L$: Prefixes(L) $\cap L = \varnothing$

EXAMPLE: $L_1 = \{ x \mid x = a^n b^n \wedge n \geq 1 \}$ is prefix-free: $a^2b^2 \in L_1$ $a^2b \notin L_1$

EXAMPLE: $L_2 = \{ x \mid x = a^m b^n \wedge m > n \geq 1 \}$ is not prefix-free: $a^4b^3 \in L_2$ $a^4b^2 \in L_2$

OPERATIONS ON LANGUAGES / 2
Operations defined over two arguments

CONCATENATION

$$\boxed{L'L'' = \left\{ xy \mid x \in L' \wedge y \in L'' \right\}}$$

m-th POWER
(inductive definition)

$$L^m = L^{m-1}L, \; m > 0$$
$$L^0 = \left\{ \varepsilon \right\}$$

NB: $\{\varepsilon\} \neq \varnothing$

NB: consequences

$$\boxed{\varnothing^0 = \left\{ \varepsilon \right\} \quad L.\varnothing = \varnothing.L = \varnothing \quad L.\left\{ \varepsilon \right\} = \left\{ \varepsilon \right\}.L = L}$$

OPERATIONS ON LANGUAGES / 3

EXAMPLES

$L_1 = \{ a^i \mid i \geq 0, i \text{ even} \} = \{\varepsilon, a^2, a^4, \ldots\}$

$L_2 = \{ b^j a \mid j \geq 1, j \text{ odd} \} = \{ba, b^3a, b^5a, \ldots \}$

$L_1 L_2 = \{ a^i b^j a \mid (i \geq 0, i \text{ even}) \wedge (j \geq 1, j \text{ odd}) \} =$

$= \{ \varepsilon ba, a^2ba, a^4ba, \ldots \varepsilon b^3a, a^2b^3a, \ldots \}$

$$(L_1)^2 = \{\varepsilon, a^2, a^4, a^6, \ldots\}\{\varepsilon, a^2, a^4, a^6, \ldots\} =$$

$$= \{\varepsilon, \varepsilon a^2, \varepsilon a^4, \ldots, a^2\varepsilon, a^4, \ldots, a^4\varepsilon, a^6 \ldots\} = L_1$$

for each pair of even numbers h and k, $h+k$ is even, hence $a^{h+k} \in L_1$

PAY ATTENTION: the language L^m in general does ***not*** contain ***only*** phrases of L repeated m times

$$\{x \mid x = y^m \wedge y \in L\} \subset L^m$$

$$m = 2 \quad L_1 = \{a, b\}$$

$$\{a^2, b^2\} \subset L_1^{\,2} = \{a^2, ab, ba, b^2\}$$

OPERATIONS ON LANGUAGES / 4

Finite length strings:

The power operator allows one to define concisely the language of strings whose length is not greater than a given integer K

$$\boxed{\begin{array}{l} L=\{\varepsilon,a,b\}^3 \quad K=3 \\ L=\{\varepsilon,a,b,aa,ab,ba,bb,aaa,\ldots bbb\} \end{array}}$$

Notice the role of ε
It allows one to obtain
all strings of length <K (0, 1, 2)

To rule out the empty string: $\boxed{L=\{a,b\}\{\varepsilon,a,b\}^2}$

OPERATIONS ON LANGUAGES / 5

SET THEORETIC OPERATIONS: the customary ones are defined:
union, intersection, difference, inclusion, strict inclusion, equality

$\cup \quad \cap \quad \setminus \quad \subseteq \quad \subset \quad =$

UNIVERSAL LANGUAGE: the set of all strings over the alphabet Σ, of any length, including 0 (i.e., string ε)

$$L_{universal} = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots$$

COMPLEMENT of a language L over alphabet Σ is the set difference with respect to (w.r.t.) the universal language (i.e., the set of strings over Σ that $\notin L$)

$$\neg L = L_{universal} \setminus L$$

hence $\quad L_{universal} = \neg \varnothing$

OPERATIONS ON LANGUAGE / 6

EXAMPLES

The complement of a *finite* language is *always infinite*

$$\boxed{\neg\left(\{a,b\}^2\right)=\varepsilon\cup\{a,b\}\cup\{a,b\}^3\cup\ldots}$$

The complement of an *infinite* one is *not necessarily finite*

$$\boxed{L=\left\{a^{2n}\mid n\geq 0\right\}\quad \neg\mathrm{L}=\left\{a^{2n+1}\mid n\geq 0\right\}}$$

Examples of the difference operation among languages

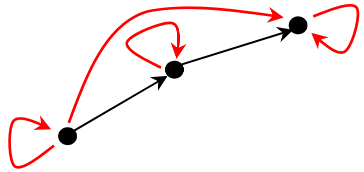
$$\boxed{\begin{array}{l}\Sigma=\{a,b,c\}\\ L_1=\left\{x\mid |x|_a=|x|_b=|x|_c\geq 0\right\}\\ L_2=\left\{x\mid |x|_a=|x|_b\wedge|x|_c=1\right\}\end{array}}$$

$L_1 \setminus L_2 = \varepsilon \cup \{\ x \mid |x|_a = |x|_b = |x|_c \geq 2\ \}$
(same number of *a*, *b*, *c*, but not =1)

A frequently used algebraic operation: reflexive and transitive closure *R** of a relation *R*

Given a set A and a relation $R \subseteq A \times A$, $(a_1, a_2) \in R$ is also denoted as $a_1 R\, a_2$

R* is a ***relation*** defined by:



- $x\, R^*\, x \quad \forall\, x \in A$, (reflexive) and
- $x_1\, R\, x_2 \;\wedge\; x_2\, R\, x_3 \wedge \ldots\, x_{n\text{-}1}\, R\, x_n \Rightarrow x_1\, R^*\, x_n$ (transitive)

If we see $a\, R\, b$ as ***a step*** in relation R, $x\, R^*\, y$ seen as

a chain of n ≥ 0 steps

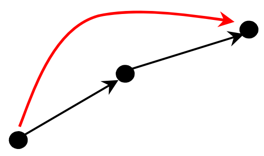
Example: if $R = \{(a, b), (b, c)\}$ then

$R^* = \{\underline{\mathbf{(a, a), (b, b), (c, c),}}\ (a, b), (b, c), \underline{\mathbf{(a, c)}}\ \}$

A variant: transitive closure R^+ of a relation R

Similarly, ***transitive*** (non reflexive) ***closure R***$^+$:
k*-th *power R^k with with $\boldsymbol{n \geq 1}$

$$x_1\, R\, x_2 \;\wedge\; x_2\, R\, x_3 \wedge \ldots\, x_{n\text{-}1}\, R\, x_n \Rightarrow x_1\, R^+\, x_n$$



Example: if relation R is the ***adjacency*** relation on a graph
R^+ is the ***reachability in one or more steps***

Example: if $R = \{(a, b), (b, c)\}$ then
$R^+ = \{ (a, b), (b, c), \underline{\boldsymbol{(a, c)}} \}$

Similarly, the ***closure*** of a ***set*** A under an ***operation*** (function)
is obtained from A by adding to it all elements obtained
by applying the operation any number of times

OPERATIONS ON LANGUAGES / 7

STAR OPERATOR: reflexive transitive closure under the concatenation operation (also called ***Kleene star***)

$$L^* = \bigcup_{h=0\ldots\infty} L^h = L^0 \cup L^1 \cup L^2 \ldots = \varepsilon \cup L^1 \cup L^2 \ldots$$

$$L = \{ab, ba\} \quad L^* = \{\varepsilon, ab, ba, abab, abba, baab, baba, \ldots\}$$

(L is finite $\quad L^*$ is infinite)

It is the union of all the powers of the language

Every string of the star language L^* can be chopped into substrings $\in L$

The star language L^* can be equal to the base language L

$$\boxed{L = \{a^{2n} \mid n \geq 0\} \quad L^* = \{a^{2n} \mid n \geq 0\} \equiv L}$$

OPERATIONS ON LANGUAGES / 8

If we take Σ as the base language, then Σ^* contains all the strings built on that alphabet (it is the ***universal language*** of alphabet $\boldsymbol{\Sigma}$)

We often say that L is a language on alphabet Σ by writing $L \subseteq \Sigma^*$

PROPERTIES OF THE STAR OPERATOR

- monotonicity (with * the set increases): $L \subseteq L^*$
- closure under concatenation: if $x \in L^*$ and $y \in L^*$ then $xy \in L^*$
- idempotence: $(L^*)^* = L^*$
- commutativity of star and reflection $(L^*)^R = (L^R)^*$

Furthermore: $\varnothing^* = \{ \varepsilon \}$ $\{ \varepsilon \}^* = \{ \varepsilon \}$ NB: these are cases where $\boldsymbol{L^*}$ is finite

Example of idempotence: We already noticed that, for $\boldsymbol{L = \{ a^{2n} \mid n \geq 0 \}}$, it holds $\boldsymbol{L^* = L}$

This derives from idempotence, because we have $\boldsymbol{L = L_0^*}$ for $\boldsymbol{L_0 = \{ aa \} = \{ a^2 \}}$

OPERATIONS ON LANGUAGES / 9

Example on the STAR OPERATOR

language of identifiers $\boldsymbol{I}$ as character strings that start with a letter and include any number of letters and digits

$$\boldsymbol{\Sigma_A} = \{\, \boldsymbol{a, b, \ldots, z, A, B, \ldots, Z} \,\} \quad \boldsymbol{\Sigma_N} = \{\, \mathbf{0, 1, 2, \ldots, 9} \,\}$$

$$\boldsymbol{I} = \Sigma_A\, (\Sigma_A \cup \Sigma_N)^*$$

if we stipulate $\Sigma = \Sigma_A \cup \Sigma_N$

language I_5 of identifiers of maximal length 5

$$\boldsymbol{I_5} = \Sigma_A\, (\, \Sigma \cup \{\, \varepsilon \,\}\,)^4$$

OPERATIONS ON LANGUAGES / 10

CROSS OPERATOR $\boldsymbol{L^+}$: transitive closure (non reflexive) under concatenation
The union does ***not*** include the first power $\boldsymbol{L^0}$
Useful but not indispensable, it can be derived from the star operator *****:

$$\boldsymbol{L^+ = L \cdot L^*}$$

$$L^+ = \bigcup_{h=1\ldots\infty} L^h = L^1 \cup L^2 \cup \ldots$$

$$\{ab, bb\}^+ = \{ab, bb, ab^3, b^2ab, abab, b^4, \ldots\}$$

$$\{\varepsilon, aa\}^+ = \{\varepsilon, a^2, a^4, \ldots\} = \{a^{2n} \mid n \geq 0\}$$

if $\boldsymbol{\varepsilon \in L}$ then $\boldsymbol{L^+ = L^*}$

Typically, a given language can be defined in different ways using different operators

Example: language L of strings of length ≥4: $\boldsymbol{L = \Sigma^4\, \Sigma^*}$ and also $\boldsymbol{L = (\Sigma^+)^4}$

OPERATIONS ON LANGUAGES / 11

QUOTIENT OPERATOR $\boldsymbol{L_1 / L_2}$: it shortens the phrases of $\boldsymbol{L_1}$ by cutting off a suffix that belongs to L_2 . NB: **forward** slash (backward slash denotes set difference)

$$L = L_1 / L_2 = \{ y \mid \exists x \in L_1 \; \exists z \in L_2 \, (x = yz) \}$$

Example: $L_1 = \{a^{2n} b^{2n} \mid n > 0 \}$ $\qquad$ $L_2 = \{b^{2n+1} \mid n \geq 0 \}$

$$L_1 / L_2 = \{a^r b^s \mid (r \geq 2, \quad r \text{ even}) \wedge (1 \leq s < r, \quad s \text{ odd}) \}$$

$$= \{ a^2 b, a^4 b, a^4 b^3, \dots \}$$

$L_2 / L_1 = \varnothing$ $\qquad$ because no string in L_2 has a string in L_1 as a suffix