Let S =

$\{ \pi : \pi$ is a permutation of $\{1, 2, 3, \ldots, n\}$

for some integer $n \geq 1 \}$.

(a) List the elements of S for n= 1, 2, and 3.

(b) Prove that the set S is countable by explicitly describing a bijection between S and the natural numbers.

(c) How does your bijection number the permutations you listed for part (a)?

CSC 320:

Mathematics of computation

Α - Alpha (Al-fah)	Ν - Nu
Β - Beta (Bay-tah)	Ξ - Xi (Zie)
Γ - Gamma	Ο - Omicron
Δ - Delta	Π - Pi (pie)
Ε - Epsilon	Ρ - Rho (row)
Ζ - Zeta	Σ - Sigma
Η - Eta (ate-ah)	Τ - Tau (tahw)
Θ - Theta	Υ - Upsilon
Ι - Iota (I -oh-ta)	Φ - Phi (fie)
Κ - Kappa	Χ - Chi (kie)
Λ - Lamda	Ψ - Psi (sigh)
Μ - Mu	Ω - Omega

And suddenly there it was, the perfect opening for Tommy's novel, lying at the bottom of his bowl of Alphabet Soup.

Mathematics of Computation

Introduction to the mathematics of formal language theory:

1. Alphabets, strings, languages.
2. Mathematical operations on strings: concatenation, substring, prefix, suffix, reversal.
3. Union, concatenation and Kleene star for languages.

These definitions lead up to our first class of languages- the regular languages.

Review: Representing Data

Alphabet: finite set of symbols

Ex. { a, b, c, … , z}

String: finite sequence of alphabet symbols

Ex. abaab, hello, cccc

Inputs and outputs of computations:

represented by strings.

ε represents an empty string (length 0)

Empty Set and Empty String

Students frequently are confused about the empty set and an empty string.

Empty set = Φ = { } = set with 0 elements.

Empty string= ε = `` '' = string with 0 symbols.

Example:

The set {ε} is a set containing one string.

Language: set of strings

First names of students taking CSC 320:

{Adrian, Alexander, Amanda, Braden, Brandon, Brodie, Catlin, Christopher, Daniel, Ding, Dong, Erik, Ethan, Evian, Gordon, Grigory, Hao, Heng, Jonathan, Jordan, Lin, Matt, Matthew, Mauricio, Meghan, Michael, Ramtin, Richard, Rui, Scott, Wenchong, Xiuxia, Xiyu, Zhuoli}

Strings over {a,b} with even length:

{ε , aa, ab, ba, bb, aaaa, aaab, aaba, ...}

Syntactically correct JAVA programs.

Notational Conventions

x, y, z real numbers

k, n integers

A, B matrices

p, q primes

$\Sigma, \Sigma', \Sigma_1\ \Sigma_2$ alphabets

a, b, c, 0, 1 symbols

u, v, w, x, y, z strings

L, L_1, L_2 languages

Concatenation of strings x and y denoted $x \cdot y$ or simply xy means write down x followed by y.

Theorem: For any string w, $\varepsilon \cdot w = w = w \cdot \varepsilon$.

String v is a prefix of w if w= v y for some string y.

String v is a suffix of w if w= x v for some string x.

String v is a substring of w if there are strings x and y such that w= x v y.

The reversal of string w, denoted w^R, is w "spelled backwards".

Operations on Languages:

1. Complement of L defined over Σ = $\overline{L}$

   = { $w \in \Sigma^*$: w is not in L }

2. Concatenation of Languages $L_1 \cdot L_2 = L_1 L_2 =$

   $\{w = x \cdot y$ for some $x \in L_1$ and $y \in L_2\}$

3. Kleene star of L, L^* = { $w = w_1 w_2 w_3 \ldots w_k$ for some $k \geq 0$ and $w_1, w_2, w_3, \ldots, w_k$ are all in L}

4. $L^+ = L \cdot L^*$

(Concatenate together one or more strings from L.)

Σ^* = set of all strings over alphabet Σ

Language over Σ - any subset of Σ^*

Examples: $\Sigma = \{0, 1\}$

$L_1 = \{ w \in \Sigma^* :$ w has an even number of 0's$\}$

$L_2 = \{ w \in \Sigma^* :$ w is the binary representation of a prime number with no leading zeroes$\}$

$L_3 = \Sigma^*$

$L_4 = \{ \} = \Phi$

$L_5 = \{ \varepsilon \}$

$L_2 = \{w \in \{0,1\}^*$: w is the binary representation of a prime with no leading zeroes$\}$

The complement is:

$\{w \in \{0,1\}^*$: w is the binary representation of a number which is not prime which has no leading 0's or w starts with 0$\}$

Note: 1 is not prime or composite. The string 1 is in the complement since it is not in L.

Matrix multiplication:

$$\begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 1 & 3 \end{bmatrix}$$

Concatenation:

ab · bb = abbb

bb · ab = bbab

Regular Languages over Alphabet Σ:

[Basis] 1. Φ and $\{\sigma\}$ for each $\sigma \in \Sigma$ are regular languages.

[Inductive step] If L_1 and L_2 are regular languages, then so are:

2. $L_1 \cdot L_2$,

3. $L_1 \cup L_2$, and

4. $L_1{}^*$.

Regular expressions over Σ:

[Basis] 1. Φ and σ for each $\sigma \in \Sigma$ are regular expressions.

[Inductive step] If α and β are regular expressions, then so are:

2. $(\alpha\beta)$

3. $(\alpha \cup \beta)$ and

4. α^*

Note: Regular expressions are strings over

$\Sigma \cup \{ \; (\; , \;) \; , \; \Phi \; , \cup \; , \; * \; \}$

for some alphabet Σ.

Precedence of Operators

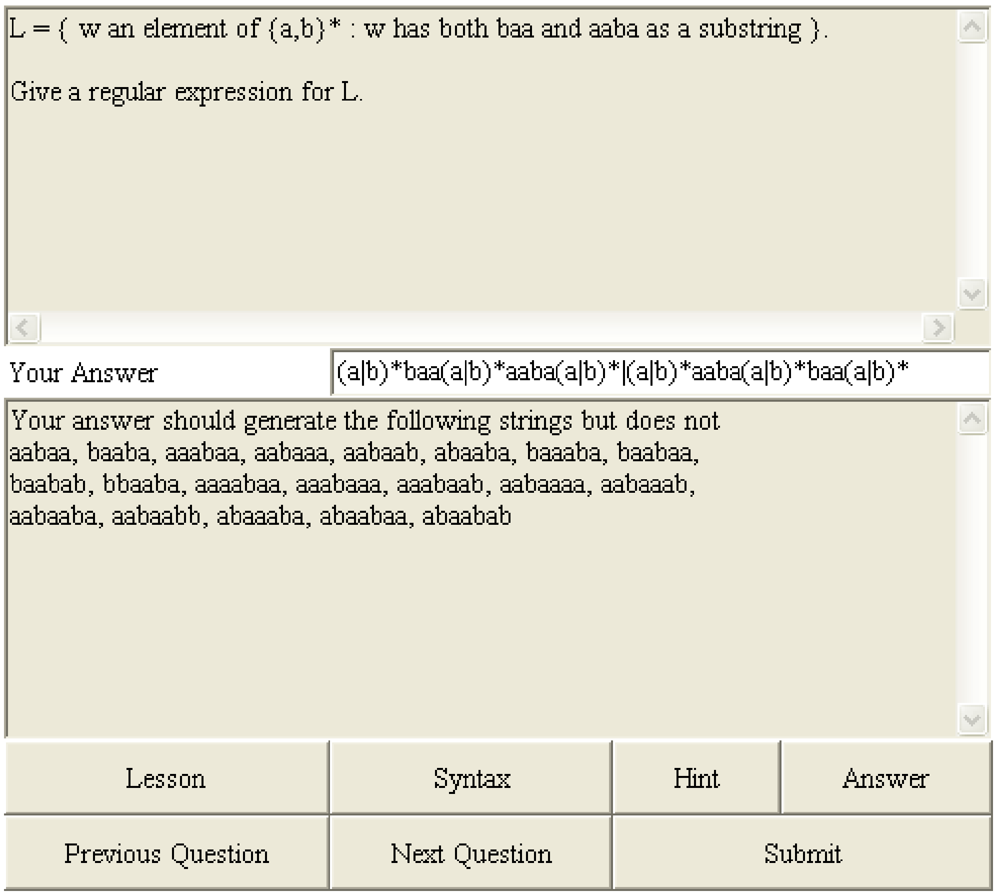
Exponents	highest	Kleene star
Multiplication	⇩	Concatenation
Addition	lowest	Union

Prove the following languages over $\Sigma=\{0,1\}$ are regular by giving regular expressions for them:

1. {w: w has odd length}
2. {w: w contains 0011}
3. {w: w does not contain 01}
4. {w: w starts and ends with the same symbol ($|w| \geq 1$)}

TUTORIAL: L = { w an element of {a,b}* : w has both baa and aaba as a substring }.

(a|b)*baa(a|b)*aaba(a|b)*|(a|b)*aaba(a|b)*baa(a|b)*



MISSING:
aabaa,
baaba,
...

See home page for link to regular expression tutorial