Strings

Tuesday, August 25, 2020 9:56 AM

Math Prelims.

Strings

A string is a finite sequence of symbols from a finite set of symbols Σ. ← also called the alphabet

e.g. $\Sigma = \{0, 1\}$

$X = 010111$ is a string over Σ

ex. $\Sigma = \{a, b, \dots z\}$

ϵ (epsilon) empty string

$|X|$: length of a string.

ex. $|010111| = 6$

Concatenation:

Xy is concatenation strings X and y

ex $X=$ hello, $y=$ world, $Xy =$ helloworld

Sometimes $X \circ Y$

Fact: $|XY| = |X| + |Y|$

Usually: w, x, y, z, mean strings (by convention)

and a, b, c are symbols.

Σ^* is the set of all strings over Σ

Qs. – is $\emptyset$ (← empty set) a string?

no

– is $\{\epsilon\}$ a string?

no

– is $\epsilon\epsilon$? yes $\epsilon\epsilon = \epsilon$

– $\Sigma = \{0, 1, \dots 9, .\}$

is $0.999\bar{9}$ ← an infinite sequence NOT a string.

– How many strings over $\{0,1\}$ are there?

∞

Inductive def'n of strings.

Strings over Σ:

- ϵ is a string
- aX is a string if X is a string and $a \in \Sigma$

Length:

$$\text{length}(X) = \begin{cases} 0 & \text{if } X = \epsilon \quad \leftarrow (I) \\ 1 + \text{length}(w) & \text{if } X = aw \end{cases}$$

Inductive Proofs:

Ex. Lemma: $\underline{|xy| = |x| + |y|}$

- Induction on $|x|$
- Base Case.

  Assume $|x| = 0$, $x = \epsilon$

  $|x| = |\epsilon| = 0$ ← by (I)

  $|x| + |y| = 0 + |y| = |\epsilon y|$

  $= |y|$ by ←(1)

$$\text{Concat}(x,y) = \begin{cases} y & x = \epsilon \leftarrow (1) \\ a \cdot (w \cdot y) & x = aw \end{cases}$$

- Inductive Case
  - Inductive Hypothesis

    (Strong IH): Suppose for $\underline{n \geq 0}$,

    IH holds. $|x| \leq n$, $|x| + |y| = |xy|$
  - Inductive Step: For $n+1$,

    Suppose $\underline{|x| = n+1}$, goal: $|x| + |y| = |xy|$

    $x = aw$ for some $a \in \Sigma$, $w \in \Sigma^*$

    $1 + |w| = |x|$.

    $xy = (aw)y$

    $= a(wy)$

    $|wy| = |w| + |y|$ ← by IH to w, y

    $|xy| = 1 + |w| + |y|$

    $= 1 + |wy|$

    $= (1 + |w|) + |y|$

    $= |x| + |y|$

Choice of inductive variable

$|x|$? $|y|$? $|x| + |y|$?

$|x| \cdot \quad |y| \cdot \quad |x|+|y| \cdot$

Languages

A Language is a set of strings.

ex. All valid C programs.

ex. All grammatically correct english sentences

L_{Eng}

"The ball flew." $\in L_{Eng}$

"The ball flyn't" $\notin L_{Eng}$.

Set Concatenation:

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

Union

$$X \cup Y = \{ w \mid w \in X, \text{ or } w \in Y \}$$

Powers: $X^n = \underbrace{X \cdot X \cdot X \cdot \dots \cdot X}_{n}$

inductively:

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

$$X^{1+n} = X \cdot (X^n)$$

Kleene Star $X^* = \bigcup_{n \in \mathbb{N}} X^n$ ← infinite union.

(Σ^*) all strings

$$X^+ = \bigcup_{n \geq 1} X^n$$

Exs. – What is $\emptyset^0$? $\{\}$? ✗ $\{\epsilon\}$ ✓

– If $|L| = 2$, what's $|L^3|$?

$L^3 = L \cdot (L^2)$ 2? $= L \cdot L \cdot L = \{xwz \mid x \in L, x \in L, z \in L$

$2^3 = 8$ ✓

Can't be ~~more~~ more than 8. (Could be less)

– $\emptyset^*$? $\{\}$?

$\{\epsilon\}$? ✓

– Does X^n contain ϵ ?

Why languages?

– We'll focus now on computing language membership or language recognition

Given string x, is $x \in L$?

Algorithm: M recognizes L if $M(x)$ outputs 1 if

0 if x

in finite time.

– Generally care about computing other things.

ex. $f : \Sigma^* \rightarrow \{0, 1\}$

boolean functions from strings to 1-bit output

Bijection between boolean functions and languages.

Let $L_f = \{ X \mid f(X) = 1 \}$

M recognizes $L_f \iff$ M computes f.

How many languages are there?

And first result in computability.

Powerset: $\mathbb{P}(X)$ all possible subsets of X.

2^X $\mathbb{P}(\{a,b,c\}) =$

$\{ \{\}, \{a\}, \{b\}, \{c\},$
$\{a,b\}, \{b,c\}, \{a,c\},$
$\{a,b,c\} \}$ size 8

$|\mathbb{P}(X)| = 2^{|X|}$

X is Countably infinite.

- if exists a bijection from X to $\mathbb{N}$.

- Can enumerate all elements in order.

eg. $\Sigma = \{0,1\}$

Σ^* is countably infinite.

0, 00, 000, ... X

$\epsilon, 0, 1, 00, 01, 10, 11, 000, \ldots$ ✓

$\Sigma^* \times \Sigma^* = \{(x,y) \mid x \in \Sigma^*, y \in \Sigma^*\}$

also countably infinite.

Cantor's

Thm: $\mathcal{P}(\Sigma^*)$ is uncountably infinite.

Consequence:

There exists some boolean function

$f: \Sigma^* \to \{0,1\}$

s.t. No C program computes f.

language, set of strings,

$L_{cprog} \subseteq \Sigma^*$

Any mapping from

$L_{cprogs} \to \mathcal{P}(\Sigma^*)$

can't be a bijection by Thm.