Problems, Languages, and Encoding

1/11/21 10:51 AM

Where were we? Oh yeah, universal computation.

Def. A problem has an input and an output.

- We need a language / symbol set to talk about it!

Def. • Alphabet Σ

e.g. $\Sigma_{eng} = \{a\ b\ c\ \dots\ z\} \cup \{A, \dots Z\} \cup \{0 \dots 9\} \cup \{\text{punctures}\} \cup \dots$

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

Strings/words Σ^k empty string $\varepsilon \in \Sigma^0$

Σ_{eng} "This_is_a_string" $\in \Sigma_{eng}^{16}$

$0011001 \in \Sigma_{binary}$

• Concatenation $x \cdot y$ / xy

apple • pie = applepie

Σ^* : set of all strings

$x, y \in \Sigma^*$ $\quad x \cdot y \in \Sigma^*$ $\quad \varepsilon \cdot \text{apple} = \text{apple}$

Def. A language L is a subset of Σ^*.

e.g. Set of all words in a English dictionary.

Set of all binary string w/ exactly one 0

$= \{0, 01, 10, 011, 101, 110, \dots\}$

Set of all compilable Python program.

→ Set of all C++ program printing Hello, world!

input: X
output: Hello, world!

Intuition. Language is a formal realization of a problem.

SORTED?
input: integer array A[1..n].
output: is A sorted? ~~sorted array A[1..n]~~

Def. Decision problem (yes/no)

SORTED? : $\Sigma^* \longrightarrow \{\text{yes}, \text{no}\}$

$\Sigma = \{\text{digits}\} = \{0, 1, 2, \dots\} \cup \{/\}$

$A = [0, 2, 7, 11] \rightsquigarrow w \in \Sigma^*$

~~$A = [0, 2, 7, 11]$~~ $\quad w = 0/2/7/11$

Problem —encoding→ Language

In class I wrote {integers}; this could work, w/o need of delimiters, as 11 itself will be a character instead of word; the problem is now Σ has infinite size, which we don't like.

Def. Language of a decision problem P (w.r.t. some encoding)

$\{w \in \Sigma^* : P(w) = \text{yes}\}$ the set of positive inputs.

input: Set of integers $\{x_1, \dots, x_n\}$.
output: Is $x_1^2 + \dots + x_n^2 < 0$?

$L = \{\} =: \emptyset \neq \{\varepsilon\}$

$x_1/x_2/\dots/x_n$

input: Set of positive integers $\{x_1, \dots, x_n\}$.
output: Is $x_1^2 + \dots + x_n^2 \leq 0$?

$L = \{\varepsilon\}$

Operations on languages.

Question. What can we solve without any knowledge of the problem?

Def. Let L_A, L_B languages

• Union $L_A \cup L_B := \{w : w \in L_A \text{ or } w \in L_B\}$

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots = \bigcup_{k \geq 0} \Sigma^k$ — In class I incorrectly wrote L instead of Σ

ODD
input: an integer N
output: is N odd?

EVEN
input: an integer N
output: is N even?

$L(\text{ODD}) := \{1, 3, \dots\}$ $\quad L(\text{EVEN}) := \{2, 4, \dots\}$

$L(\text{ODD}) \cup L(\text{EVEN}) = \{0, \dots, 9\}^*$

• Concatenation $L_A \cdot L_B := \{xy : x \in L_A, y \in L_B\}$

$= \{w : w = xy, x \in L_A, y \in L_B\}$

$L_A = \{\varepsilon, \text{west\_}\}$ $\quad \{\varepsilon\} \cdot L = L$

$L_B = \{\text{lebanon}, \text{virginia}\}$ $\quad \emptyset \cdot L = \emptyset$

$L_A \cdot L_B = \{\text{lebanon}, \text{virginia}, \text{west\_lebanon}, \text{west\_virginia}\}$

• Kleene-Star $L^* = \{x_1 x_2 \dots x_k : \forall k \geq 0 \ x_i \in L_A \ \forall i \in \{1 \dots k\}\}$

$= \{\varepsilon\} \cup L \cup L \cdot L \cup L \cdot L \cdot L \cup \dots$

$L = \{ba\}$

$L^* = \{\varepsilon\ \text{ba}\ \text{baba}\ \text{bababa}\ \dots\}$

$L_B = \{a, ba\}$

$L_B^* = \{\varepsilon, a, ba, aa, aba, baa, baba, \dots\}$

Def. A language L (over Σ) is regular if L equals:

- $\emptyset$ — return no
- $\{w\}$ — some word w. return yes if input is w
- $A \cup B$ — A, B reg. conditionals. if-else.
- $A \cdot B$ — " — sequencing
- A^* — A reg. iteration while-loop. (no counters!)

Def. Regular expression: $\emptyset, w, A+B, AB, A^*$

$(01)^* \neq (01)^*$ $\quad (0+1)^* = (0^*1^*)^*$

$\{0, 01, 011, \dots\}$ $\{\varepsilon, 01, 0101, \dots\}$ $\{0, 1\}^* = \{$alternating between strings of 0s and 1s$\}$

Question. Can all languages be represented as reg. exp.?

Can all problems be solved w/o brains?