Formal Languages

Ken Q Pu, Faculty of Science, Ontario Tech University

Expressing problems and solutions

Problem:

67 * 4.5 = ?

Answer:

67 * 4.5 = 301.5

Problem:

Given two numbers, x and y, what is (x * y)?

Answer:

?

To express the answer to the multiplication problem, we need to work with a more generic form of solutions: *formal languages*

Problems and instances

Problem:

67 * 4.5 = ?

An instance of the MULT problem

Problem:

Given two numbers, x and y, what is (x * y)?

The problem: MULT

Languages and Computation

- Problems and solutions are characterized by symbolic strings.
- Computation is
  - Determine membership of a set of string
  - Mapping between sets of strings
- Algorithms are mappings from problem space to solution space
- Implementations are mappings from finite strings to finite strings
  - Finite Automaton (not covered in this course)
  - Turing Machine
  - Lambda Calculus

Definitions

- An alphabet is a finite set of symbols, written as Σ.
- A string is a finite sequence of symbols from Σ.
- The (infinite) set of all possible (finite length) strings is written as Σ^*.
- A language is a subset of Σ^*.

String Encoding

Thesis:

Given any mathematically defined decision problem P, there exists a string encoding of all of its instances.

$$\textbf{ENC} : \text{instances}(P) \rightarrow \Sigma^*$$

Algorithm (decision procedure)

Given an encoding **Enc** of a decision problem P, an algorithm is a string processing function:

$$\textbf{alg} : \Sigma^* \longrightarrow \text{boolean}$$

such that for all instances $x \in A$, we have

$$P(x) = \textbf{alg}(\textbf{Enc}(x))$$

Why only decision problems?

- We definitely want to do more than decisions.
- It turns out that general purpose computation is only superficially more complex than their decision counter parts.
- Same definitions, theories, and computational results apply to both general purpose computing and decision problems.

Decision counter parts of general purpose computing

Integer multiplication

- Input: x, y
- Output: x * y

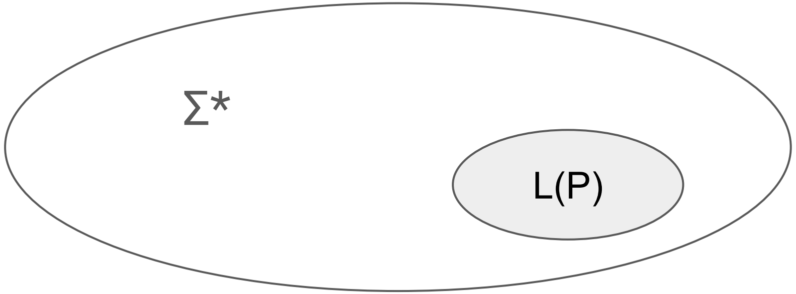
Verification of integer multiplication

- Input: x, y, z
- Output: check if z = x * y

Language and decision problems

For every decision problem P there exists a language L(P) defined as the encodings of "good inputs".

$$L(P) = \{\ Enc(x) : x \in B\ \}$$



When working with languages, we call the decision problem by a different name: ***recognition of the language.***

Some decision problems and their languages

PRIME:

- Decision problem: is ***x*** a prime number?
- Language: the (infinite) set of all prime numbers

Syntax checking of HTML?

- Decision problem: Is this HTML valid?
- Language: the set of all valid possible HTML pages

Multiply

- Decision problem: is ***z*** the produce of ***x*** and ***y***?
- Language: the set of all possible triples (***x***, ***y***, ***x********y***).

String encoding of inputs: examples

- We work with only three symbols: $\Sigma = \{ 0, 1, \_ \}$
- PRIME: we use the binary representation
    - Input: 17
    - Encoding: 10001
- MULT: we need to encode a triple (x, y, z). This can be done using the separator symbol to join Enc(x), Enc(y), Enc(z)
    - Input: 4 * 5 = 20?
    - Encoding: 100_101_10100

String Processing Methods

- List all elements of L(P)
  - Impractical for large languages
  - Impossible for infinite languages (like PRIME)
- Use regular expressions
  - This corresponds to a state machines, also known as finite state automata (FSA)
  - Most useful languages cannot be described by regular expressions (like PRIME)
  - Did you know that programming language syntax is not regular?

String Processing Methods

- Use context free grammar (CFG) (not covered in this course)
  - This corresponds to FSA with a stack storage
  - Most programming languages can be decided by CFG
- **Turing Machine**
  - This corresponds to FSA with a movable HEAD and a tape storage
  - All Python-decidable problems can be recognized by TM