Theory of Computation- Lec 1

Terms:

- Computation
- Programs
- Algorithms (i.e., theory of algorithms)
  - Step-by-step description of how to carry out computing of a function
- Input/Output of an Algorithm
- Input:x, output: f(x), function f
- Define a function: f:X→Y
- Partial function
- Example: is_prime(x)

Basic Goal

- To identify class of functions
  - Set1: {f: there are algorithms to compute them}
  - Set2: {f: no algorithms to compute them}
- Characteristic function of set X
- i.e., set-membership
- A computing device (which computes function f)
  - f is a Recognizer, if ...
  - f is an Acceptor, if ...
  - f is a Generator, if ...

Representation of Objects

- Set of finite strings?
- Set of natural numbers, N
- Set of binary digits, B
- An Alphabet, Σ, is a finite set of symbols.
- Symbols
- Sting: a finite **sequence** of symbols chosen from an alphabet
- The empty string, ε, has zero occurences of symbols.
- String length,
- Σ^* is set of all finite strings over Σ
- A formal language L over the alphabet Σ is a subset of Σ^*

Language Examples

- {ε, 01, 0011, 000111, ...}

- Set of strings of 0's and 1's with equal number of each
  - {ε, 01, 10, 0011, 0101, 1001, ...}
- Set of binary numbers whose value is a prime
  - {10, 11, 101, 111, 1011, ...}
- Σ*
  - A language for any alphabet Σ
- {ε}
  - Lang consisting of only the empty string

Our concern...

- Set membership **problem** of formal language
- input, output: strings
  - Assume Σ is an alphabet and L is a lang over Σ
  - Given a string w in Σ^*, decide whether or not w is in L
- **Models of computation**: Automata of various types
  - Finite State Automata
  - Pushdown Automata
  - Linear-Bounded Automata
  - Turing Machine
  - Grammars: regular, context-free, context-sensitive

Inductive Defs

- Form:
  - $f(0, y) = g(y)$
  - $f(n+1, y) = h(n, y, f(n,y))$
- Example1:
  - $y^0 = 1$
  - $y^{n+1} = y^n \cdot y$ $\quad g(y) = 1,\ h(x,y,z) = z \cdot y$
- Example2:
  - Cumulative sum of c_i $(i=0, \ldots n)$
  - …
  - $G(y) = c_0$ $\quad , h(x,y,z) = z + c_{x+1}$

Inductive defs (cont'd)

- Boolean function:
  - Inputs:x_1, ... x_n; outputs: y_1, ... y_m
  - Interested in case when m=1
- Such a boolean fcn can be represented as a finite table of 2^n entries
  - Or a boolean expression, consisting of
    - Boolean vars,
    - Boolean constants
    - Boolean operations (and, or, not)

Boolean Function

- The Cartesian product:

$X_1 \times X_2 \times \ldots \times X_n = \{(x_1, x_1, \ldots, x_n): x_1 \varepsilon X_1, \ldots, x_n \varepsilon X_n\}$

$X^n = X_1 \times X_2 \times \ldots \times X_n$ when $X_1 = X_2 = \ldots = X_n$

- A boolean function f is any function $f: B^n \rightarrow B^m$
- Interested in case where m=1
- Repr1:
  - A finite table with 2^n entries
- Repr2:
  - A boolean expression consists of
  - Boolean vars
  - Boolean constants
  - Boolean operators

Boolean Expr (inductive defs)

- Any boolean var is a boolean expression, and any constant is a boolean expression.
- If e_1 and e_2 are boolean exprs, then so are (e_1 **or** $e_{2)}$, (e_1 **and** e_2) and (**not** e_1)
- Every boolean expr with n vars represents some boolean function f: $B^n \rightarrow B$
- Theorem: Every boolean function f: $B^n \rightarrow B$ is represented by some boolean expr with n vars.

Finite Memory Devices (Finite State Automata, Finite State Machines)

- Alphabet, Σ, a finite set of symbols
- Σ^* = set of all finite strings over Σ
- Language L over the alphabet Σ = a subset of Σ^*

- Problem:
  - A lang L,
  - Given a string x, to decide if x belongs to L

Intro to FSA

- By an example
  - L is $\{x \text{ in } \{0,1\}^* : x \text{ has even number of 0's}\}$
  - Given a binary string x (x=0011010), is x in L?
- Way to process strings:
  - Start scanning from left-end
  - Go over the symbols one at a time
  - As soon as string is entirely scanned, answer is ready

State Transition Diagram

Automata is defined by followings:

- A set of states
- The input alphabet
- The state transition function, a mapping ...
- Start state
- Set of final, or accepting states

- A FSA M has five components. $(Q, \Sigma, \delta, q_0, F)$
- M is (...)

Example2

Another example:

- L_1 is {x in $\{0,1\}^*$: x has an even number of 0's **and** has an even number of 1's}
- Given any binary string x (x=0011010), is x in L?

- Define an FSA.

  i.e., x=0010010110 is given. Seen so far?

State Transition Diagram for Ex2

- i.e., x=0010010110 is given. Seen so far?

- FSA M = (...)

Example3

- {x01y: x, y are any strings of 0’s and 1’s}

Extention

- Recall δ
- Extend it with δ head
- Instead of symbol, use string as input to δ head

- Inductive definition for δ head
  - Base:
  - Iductive part:

- Given any M =
- L(M) = set of strings accepted by M
- L(M) = $\{x \text{ in } \Sigma^* : \delta \text{ head } (q_0, x) \text{ in } F\}$

Ultimately

- Given a language L_1
- Find FSA M
  - such that $L(M) = L_1$