Theory of Computation: Lecture 1

Morgan McCarty

07 September 2023

1. Theory of Computation
   - What is computation?
     - Math-checking if no. prime
     - Verifying logic
     - Navigational routing
     - Comparing strings $\rightarrow$ looking up, sorting, etc.
   - All computation relative to computer? Can we do ths w/o the computer?
   - What is a computer? Modeling a computer
     - Supported operations
     - Determinsitic (usually)
     - Output
     - Input
   - Are there fundamental limits on what is computable?
     Yes: Halting problem
     - Does a program actually halt?
   - Areas of the course: (in order of decreasing complexity)
     (a) Computability: what can be computed given enough time and space
     (b) Complexity: how fast/efficiently can we solve a problem
     (c) Automata: what problems can we solve given very limited space (constant)
   - Why does this matter?
     - Checking correctness of a program
     - Knowinig what functions can be computed quickly and which cannot (security)
   - Goals of the course:
     (a) Understand notions of computability
     (b) Understand limitations of computability
     (c) What can be doen with weaker forms of computability
     (d) Computational relation to formal languages
2. Mathematical Review
   - Sets:
     - Unordered group of elements (finite or infinite)
     - E.g.

    - $\mathbb{N} = \{1, 2, \ldots\}$
    - $\mathbb{N} \cup 0$
    - $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$
    - $\emptyset = \{\}$
  - Set Operations:
    - $X \cup Y = \{x \mid x \in X \vee x \in Y\}$
    - $X \cap Y = \{x \mid x \in X \wedge x \in Y\}$
    - $\bar{X}$: negaton of set relative to universe
    - $X \setminus Y = X - Y = \{x \mid x \in X \wedge x \notin Y\} = X \cap \bar{Y}$
- Logic:
  - $\wedge$: and
  - $\vee$: or
  - $\implies$: implication
  - $\alpha \implies \beta$
    "if α is truse, then β is true"
    "if Corina gets fed, then Ariel sleeps in"
  - Negation of $\alpha \implies \beta$ is $\alpha \wedge \bar{\beta} \equiv \bar{\alpha} \vee \beta$
  - Satisfiable: the formula has a set of boolean assignments so that the entire thing evaluates to true
- Proof Techniques:
  - Proof by induction
    - Induction Prove: $\forall n \in \mathbb{N}_0, P(n)$
    - $P(n)$: predicate (boolean statement about n)
    - <u>Base Case</u>: $P(0)$ (or $P(1)$)
    - <u>Inductive Step</u>: Assume $P(n)$ is true, show $P(n+1)$ is true
    - E.g.
      - Show $5 + 10 + 15 + \cdots + 5n = \frac{5n(n+1)}{2} \forall n \in \mathbb{N}$
      - Base Case: $n = 1$
        $5 = \frac{5(1)(2)}{2} = 5$
      - Inductive Step: We know for $n = k$
        $5 + 10 + 15 + \cdots + 5k = \frac{5k(k+1)}{2}$
      - Show for $n = k + 1$:
        $5 + 10 + 15 + \cdots + 5k + 5(k+1) = \frac{5k(k+1)}{2} + 5(k+1)$
        $= \frac{5k(k+1)+10(k+1)}{2} = \frac{5(k+1)(k+2)}{2}$
  - Proof by contradiction
    - Assume the opposite of proof statement and show that it leads to a contradiction of a known fact
    - E.g.
      - Show $\sqrt{2}$ is irrational (cannot be written as $\frac{p}{q}$ where $p, q \in \mathbb{Z}$ and $q \neq 0$)
      - Assume $\sqrt{2}$ is rational
      - $\sqrt{2} = \frac{p}{q}$ where $p, q \in \mathbb{Z}$ and $q \neq 0$
      - p and q are relatively prime (no common factors)
      - $q\sqrt{2} = p$

    - $2q^2 = p^2$
    - p^2 is even $\implies$ p is even
    - $p = 2k$ for some $k \in \mathbb{Z}$
    - $2q^2 = (2k)^2 = 4k^2$
    - $q^2 = 2k^2$
    - q^2 is even $\implies$ q is even
    - p and q are both even, but this contradicts the fact that p and q are relatively prime
    - $\therefore \sqrt{2}$ is irrational
  - Proof by construction
  - Proof by contrapositive
  - Proof by reduction

- <u>Alphabet</u>
  - Σ: finite set of symbols ("letters", "elements")
  - E.g.
    - $\Sigma = \{0, 1\}$, $\Sigma = \{a, b, c\}$
  - A string over Σ ($w \in \Sigma$) is a finite sequence of symbols from Σ. Σ^* is the set of all strings over Σ.
  - E.g.
    - $w = 010101$, $w = 101010$, or $w = 0000$
    - $w =$ aabab, $w =$ ababab, or $w =$ aaa
    - ϵ is the empty string

- <u>Language</u>
  - A language over Σ is a set of strings over Σ
  - E.g.
    - $L = \{a, ab, aa, bb, \ldots\}$ is a language over $\Sigma = \{a, b\}$
  - Language is a subset from Σ^*
  - How to decide what's in a language?
    - Total list
    - Can we do better?
  - <u>Machine</u> to decide the language
  - $x \in \Sigma^* \rightarrow \boxed{M} \rightarrow Y$ or N
  - Often define an L by the describetion of the M
    M <u>accepts</u> some strings and <u>rejects</u> others
    M <u>defines</u> a language $L(M) = \{x | M \text{ accepts } x\}$

- <u>Strings</u>
  - Concatenation: $x \cdot y$
    abc $\cdot$ aab = abcaab
  - Empty string: ϵ a $\cdot \epsilon = \epsilon \cdot a = a$
  - Length: $|a|$ is the number of elements in a String a
    $|\text{aab}| = 3$, $|\epsilon| = 0$
  - Prefix: $x, y \in \Sigma^*$, x is a prefix of y if $\exists z \in \Sigma^*$ such that $x \cdot z = y$
    If $x = y$, then $z = \epsilon \iff x$ is a prefix of y and y is a prefix of x
    Something is always a prefix of itself

- Suffix: x is a suffix of y if $\exists z \in \Sigma^*$ such that $z \cdot x = y$
  If $x = y$, then $z = \epsilon \iff x$ is a suffix of y and y is a suffix of x
  Something is always a suffix of itself
- Substring: x is a substring of y if $\exists z_1, z_2 \in \Sigma^*$ such that $z_1 \cdot x \cdot z_2 = y$
  If $x = y$, then $z_1 = z_2 = \epsilon \iff x$ is a substring of y and y is a substring of x z_1 and z_2 can be ϵ
- Lexigraphical Ordering over String:
  Requires order on Σ can be used to order strings

- Countable and Uncountable Sets
  - $A : \{1, 2, 3\}$, $B : x, y, z$
  - Function f that maps A to B (one-to-one)
  - $f(1) = x$, $f(2) = y$, $f(3) = z$
  - $|A| = |B| = 3$ then $\exists f$ that is one-to-one correspondence
  - Works for finite sets and countably infinite sets
  - $\mathbb{N}$, $2\mathbb{N}$, $f : \mathbb{N} \to 2\mathbb{N}$
  - If infinite set can be mapped to $\mathbb{N}$, then it is <u>countably</u> infinite
  - If infinite set cannot be mapped to $\mathbb{N}$, then it is <u>uncountably</u> infinite