CS3231

Lecturer: Sanjay Jain (COM1 03–36; sanjay@comp.nus.edu.sg
Web Page: http://www.comp.nus.edu.sg/∼sanjay/cs3231.html
Text Book: Hopcroft, Motwani and Ullman — Introduction to Automata Theory, Languages and Computation, 3rd edition.
Tutorials:

- start in second week; 11 or 12 tutorials
- Questions for tutorial will be given in class (previous week). Due at the start of the tutorial

Grading:

- 2 midterms (25 % each)
- 1 final (50 %)

Why Study Theory of Computation

- Computational devices appear all over the place – besides the computers, several products have computational devices inside them: TV, vending machines, mobile phones, . . .
- Though models and forms of computation may differ, essential idea underlying them remain the same.

  Example: vending machines may be despensing soft drinks, snacks, train tickets, . . .
- Theory of computation helps you develop an understanding of what computation is, what are its limits, what can be done and what cannot be done, how much resources (time, memory, . . .) are needed to do a job, etc.
- We will mostly focus on acceptance of languages: automata, grammars, expressions, etc.

Formal Proofs

If you studied geometry by traditional methods, you would have seen how one proves validity of a statement by using a chain of arguments.

- Deductive proofs:

  Example: if $x \geq 4$, then $2^x \geq x^2$.

- Modus Ponens:

  $A \Rightarrow B$, A, then we can conclude B.

- Proof by Contradiction.

  Example: Suppose U, the universal set, is infinite. Then for any set A, either A is infinite of $\overline{A}$ is infinite (where $\overline{A} = U - A$).

-

Counterexample.
Example: All primes are odd (False: counterexample is 2).

Formal Proofs

- Contrapositive: $A \Rightarrow B$ can be proved by showing $\neg B \Rightarrow \neg A$.
- Equivalence (iff, if and only if)
  - Suppose x is a real number. Then $x = \lfloor x \rfloor = \lceil x \rceil$ iff x is an integer.
  - $\lfloor r \rfloor$ is the largest integer which is less than or equal to r
  - $\lceil r \rceil$ is the smallest integer which is greater than or equal to r
- Converse
  - Converse of $A \Rightarrow B$ is $B \Rightarrow A$
  - Usage when we prove iff type statements

Formal Proofs

- Inductive Proofs.
  - $1 + 2 + 3 + \ldots + n = n(n + 1)/2$.

Formal Proofs

- Inductive Proofs

  Basic form: prove

  - base case ($n = 1$)
  - induction step: If statement holds for $n = k$, then it holds for $n = k + 1$.

  Course of values induction: prove

  - base case ($n = 1$)
  - induction step: If statement holds for smaller values of n ($n = 1$ to k), then it holds for $n = k + 1$.

  Structural Induction

  Example: If a claim holds for all trees of height at most k, then the claim holds for all trees of height $k + 1$.

Formal Proofs

Mutual Induction: showing several claims to be true simultaneously.

Quantifiers: $(\exists x)[P(x)]$ and $(\forall x)[P(x)]$

Sometimes also use $(\stackrel{\infty}{\exists} x)[P(x)]$, $(\exists! x)[P(x)]$.

Central Concepts of Automata Theory

- Alphabets
  - Alphabet is a finite non-empty set of symbols. We usually use the symbol Σ to denote the alphabet.
  - $\{0, 1\}$.
  - $\{A, B, \ldots, Z\}$
  - $\{0, A, s\}$
- Strings:

  Finite sequence of symbols chosen from a given alphabet. For example, $010001, ACB, 0sAss0$.
- Empty String: ϵ. Sometimes Λ is also used.

Central Concepts of Automata Theory

- Length of a string: number of symbols in the string.

  Example: length of 01001 is 5.

- Powers of an Alphabet:
  - $\Sigma^1 = \{0, 1\}$
  - $\Sigma^2 = \{00, 01, 10, 11\}$
  - $\Sigma^0 = \{\epsilon\}$
  - $\Sigma^{\leq 2} = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2$

- Concatenation of Strings:

  $x = 00$, $y = 10$, then $x \cdot y = xy = 0010$.

- Substring.
- Subsequence.

- Languages: A set of strings (over an alphabet).

  $L = \{00, 11, 01, 110\}$.

  $L = \emptyset$.

  $L = \{x \mid x \text{ is a binary representation of a prime number }\}$.

1. $L_1 \cdot L_2 = \{xy \mid x \in L_1, y \in L_2\}$.
When there is no confusion, we often drop $\cdot$, $L_1 L_2$ represents $L_1 \cdot L_2$.
2. $L^* = \{x_1 x_2 \ldots x_n \mid x_1, x_2, \ldots, x_n \in L, n \in N\}$.
Here n can be 0 (thus $\epsilon \in L^*$).
$L^* = \{\epsilon\} \cup L \cup LL \cup LLL \cup \cdots$.
3. $L^+ = \{x_1 x_2 \ldots x_n \mid x_1, x_2, \ldots, x_n \in L, n \geq 1\}$.
$L^+ = L \cup LL \cup LLL \cup \cdots$.

Number of strings over any fixed finite alphabet Σ is countable:
We do the proof for $\Sigma = \{0, 1\}$.
The same idea can be generalized to prove the result for larger alphabets.
For any string x, let $m(x) =$ value of $1x$ as binary number.
Thus we are mapping string ϵ to 1, string 0 to 2,
string 1 to 3, string 00 to 4 and so on.
Thus, number of strings is countable.

Number of languages over any non-empty alphabet is uncountable.
Consider $\Sigma = \{a\}$.
Then consider binary representation of any real number in $[0, 1)$: $0.b_0 b_1 \cdots$.
Let L be the language $\{a^i \mid b_i = 1\}$.
Thus, we have formed a language corresponding to each real number in the interval $[0, 1)$.
Thus, the number of languages is atleast as large as the set of real numbers, which is uncountable.