CSEN502
Theory of Computation

Course Material

- **Textbook:** Sipser, Michael (2006). *Introduction to the Theory of Computation (2nd edition)*, Thomson Course Technology.
- **Internet:**

  `met.guc.edu.eg/courses/Winter2021_CSEN502.aspx`

Disclaimer!

These slides are not meant to be comprehensive lecture notes; only sketchy remarks and pointers. The material presented here is not sufficient for studying for the course. Your main sources for studying are (i) the text and (ii) your own lecture notes.

Five Reasons Why You Should Study Theory of Computation!

1. It is a degree requirement.
2. It purports to answer the following seemingly simple questions.
   (a) What is a problem?
   (b) How do you determine the degree of hardness of a given problem? (Is there such a thing?)
   (c) Are there unsolvable problems? (And how can you know that a problem is unsolvable without trying to solve it?)
   (d) What is an algorithm?
   (e) What is a computer?
3. Its tools are applied all over the place in computer science.
4. It is good for your intellect.
5. In its own cryptic way, it is fun after all.

Course Assessment

- Your grade in this course will be based on your scores in one midterm exam, one final exam, and your best $n - 2$ of n quiz scores.
- Weights

Quizzes	40%
Midterm	20%
Final Exam	40%

Homework Assignments

- Weekly, mostly pencil-and-paper style, occasional programming tasks.
- Designed to
  1. give you practice in applying the concepts covered in class,
  2. give you a chance to assess the level of your understanding, and
  3. prepare you for the in-class quizzes
- Homeworks will not be submitted for grading, though you are encouraged to discuss your answers with us.

Quizzes

- Mostly weekly.
- Towards the end of the Monday lecture (unless otherwise announced).
- First quiz: Week starting October 16, covers this lecture.
- **No makeup quizzes.**

Mathematical Preliminaries

Lecture 1

October 3, 2021

Sets

- A set is a collection of entities regarded as a unit.
- These entities are referred to as the **members** or **elements** of the set.
- By "entity" we mean practically anything: numbers, symbols, strings, people, objects, events, or even other sets.
  - Though not necessary, members of a typical set have some properties in common.

Formal Representation of a Set

- One way of formally representing a set is to list its elements between curly braces. (Another way is using **Venn diagrams**.)
  - $\{1, 3, 5\}$
  - $\{a, b, c, d, e, f\}$
  - $\{\{1, 3, 5\}, \{a, b, c, d, e, f\}\}$
- Note that $\{1, 3, 5\}$, $\{5, 1, 3\}$, and $\{1, 1, 3, 3, 5, 5\}$ are different formal representations of the same set.
  - Can you relate this to what you know about abstract data types and data structures?
- How can you formally represent a set of people? or events?
  - Note that, ultimately, we can express only sets of strings of symbols.

Extensions and Intensions

- Lists between curly braces or Venn diagrams are **extensional** representations of sets.
  - *A representation is extensional if it describes a set by explicitly representing all of its elements.*
- If the set is infinite we either rely on intuition (which is not always a good idea):
  - $\{1, 3, 5, 7, \ldots\}$

  or use an **intensional** representation. [a]
  - *A representation is intensional if it describes a set by specifying a property that is shared by all and only the elements of the set.*

---

[a] For an interesting discussion of the distinction between intensions and in*t*entions, see `www.cse.buffalo.edu/~rapaport/intensional.html`.

Intensional Representations

- Give an intensional representation for the set $\{1, 3, 5, 7, \ldots\}$.

The Language of Set Theory

- $\mathcal{U}$ denotes the **universe of discourse**, or the set of all entities under consideration.
- $\varnothing \equiv \{\}$.
- "x is a member of A" $\equiv x \in A$.
- Everything else is defined in terms of $\in$.
  - $A \subseteq B \equiv$ for every $x \in \mathcal{U}$, if $x \in A$ then $x \in B$.
  - $A = B \equiv A \subseteq B$ and $B \subseteq A$.
  - $A \subset B \equiv A \subseteq B$ and $B \not\subseteq A$.
  - $A \cap B = \{x | x \in A \text{ and } x \in B\}$.
  - $A \cup B = \{x | x \in A \text{ or } x \in B\}$.
  - $\overline{A} \equiv \{x | x \in \mathcal{U} \text{ and } x \notin A\}$.

Power Sets

- The power set of A is the set of all subsets of A
  - What is the power set of $\{1, 3, 5\}$?

Sequences and Tuples

- Sequences vs. sets.
  - $(1, 3, 5)$ vs. $\{1, 3, 5\}$.
- A finite sequence with k elements is called a **k-tuple**.
  - A 2-tuple is called an ordered **pair**.
  - A 3-tuple is called an ordered **triple**.
  - A 4-tuple is called an ordered **quadruple**.
- Cartesian product: $A \times B = \{(x, y) | x \in A \text{ and } y \in B\}$.
  - What is $\{1, 3, 5\} \times \{a, b\}$?

Functions

- A function sets up an input/output relationship; it maps the input into the output.
- A function always produces the same output for the same input.
- The set of possible inputs is the **domain** of the function.
- The set of possible outputs is the **range** of the function.
  - $f : D \longrightarrow R$.
- Note that D could be the Cartesian product of several sets.
  - For k sets, f is said to be a k-ary function.
- R **can be the Cartesian product of several sets.**

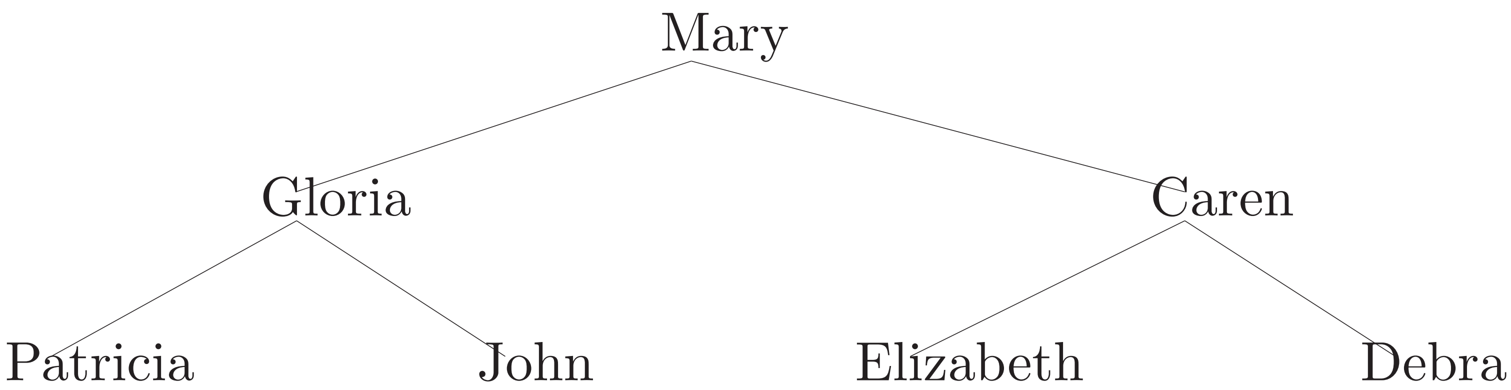
Representations of Functions

- Extensional
  - Sets of pairs.
  - Tables.
- Intensional
  - A recipe for how to compute the output for a general input.

Relations

- You can think of a relation as a function whose range is {TRUE, FALSE}.
- For convenience, a relation is represented as the set of input tuples for which the output is TRUE.
- **Notation:** If R is a binary relation, then $(x, y) \in R$ is often represented as xRy.
- Example
  - The kinship relations world: parent-of, ancestor-of, sister-of, etc.

Example: Family Tree



- Let $S = \{$ Mary, Gloria, Caren, Patricia, John, Elizabeth, Debra $\}$.
- Give an extensional definition of the relation R_p, where $xR_py \leftrightarrow x$ is a parent of y.

Properties of Relations

- Let $R \subseteq S \times S$. R is
  - **reflexive** if and only for every $x \in S, xRx$,
  - **symmetric** if and only if for every $x, y \in S, xRy \rightarrow yRx$,
  - **anti-symmetric** if and only if for every $x, y \in S, xRy$ and $yRx \rightarrow x = y$,
  - **asymmetric** if and only if for every $x, y \in S, xRy \rightarrow$ it is not the case the yRx, and
  - **transitive** if and only if for every $x, y, z \in S, xRy$ and $yRz \rightarrow xRz$
- An **equivalence relation** is a relation that is reflexive, symmetric, and transitive.

Graphs

- Nodes and edges.
- Directed and undirected graphs.
  - **Degree**, **indegree**, **outdegree**.
- Graphs as depictions of binary relations.
  - The kinship relations graph.

More Graph Terminology

- **Subgraph:** a subset of the depicted relation on a subset of the ground set.
  - The text's version is weaker.
- **Path:** a sequence of nodes connected by edges.
  - A graph is **connected** if every two nodes have a path between them.
  - A directed graph is **strongly connected** if a directed path connects every two nodes.
- **Simple Path:** a path that doesn't repeat any nodes.
- **Cycle:** A path that starts and ends at the same node.
- **Simple Cycle:** a cycle that contains at least three nodes and repeats only the first and last nodes.

Trees

- **Tree:** a connected, undirected graph with no simple cycles.
- **Root:** a distinguished node in the tree.
- **Leaf:** a node, other than the root, with degree one.

Points to take home

- Formal representation of sets.
- Set operations.
- Power sets.
- Tuples.
- Cartesian products.
- Functions.
- Relations.
- Reflexivity, symmetry, transitivity.
- Graphs.

Next time

- Proof Techniques.