Theory of Computations

Formal Modeling Concepts

Lecturer: Dr. Manar Elkady

Modeling

- What is a Model?
  - A Model is a purposeful abstract representation of reality
    - Purpose
      - You may have more than one model for the same reality for different purpose
    - Abstract
      - Does not represent instances
      - Schematic description

Modeling notation

- Textual
  - Pseuducode, structured English, programs,
- Visual
  - Flow charts, UML, ERD
- Formal
  - Mathematics, logic, graphs

Models

- Enhance better understanding
- Higher level of thinking
- Important in design
- Better communication
- Formal: a medium for proof
  - The existence a required property
  - The absence of a property

Formal Models

- Set Theory
- Graph Theory
- Formal Logic

Sets

- A set is a collection (group) of elements
  - Notation { }, {a, x, book} {s | student (s)}
  - Ordering
  - Repetition
  - Explicit/ Implicit
  - Cardinality (Function)
  - Φ : Empty Set
- Tupple (List)
  - Notation( )
  - Order
  - Repetition

Relationships

- Logical (Boolean)
  - Answers a question
  - Membership:
    - *(Input: An item & a Set, Output: Boolean)*
  - Subset:
    - *(Input: Two Sets, Output: Boolean)*
  - Equivalence:
    - *(Input: Two Sets, Output: Boolean)*
    - Note: Subset & Equivalence

Operations

- Produce results
  - Union: R U S
    - *(Input: Two Sets, Output: a Set)*
    - *A set contains elements in either of input sets*
    - *Commutative, i.e., R U S = S U R*
  - Intersection
    - *(Input: Two Sets, Output: a Set)*
    - *A set contains elements in both input sets*
    - *Commutative*
  - Difference
    - *(Input: Two Sets, Output: a Set)*
    - *A set contains elements in one but not in the second*
    - *Not Commutative*

Operations

- Cartesian Product
- Two sets S and T
  - $S \times T = \{(s, t) \mid s \in S \ \& \ t \in T\}$
  - Cardinality
  - Order

Relation

- A relation, R, between two sets S & T
  - Meaningful association between an element in S with an element in T
  - A set of pairs, one element from S and one element from T
    - $R = \{(s, t) \mid s \varepsilon S \ \& \ t \varepsilon T\}$
    - A subset from Cartesian Product
    - Cardinality = number of elements: |R|
  - Arity: binary (2), 3, 4, …number of sets in the relation

Functions

- Two sets, X and Y
- A function F: X -> Y or $y = f(x)$.
  - X is the Domain
  - Y is the Co-domain
  - y is the Image of *x* under f
  - Range, the set of images
  - A function is a relation with unique images
  - Cardinality is a function

Sets of Numbers

- N: the set of *Natural Numbers*
- Z: the set of *Integers*
  - N with 0 and negative integers
- Q: the set of *Rational Numbers*
  - Fractions: two numbers from N or Z
- R: The set of *Real Numbers*
  - *Non rational numbers, roots*
    - The set of *Imaginary numbers*

Graph theory

Graphs

- A graph G is a construct of two finite sets, V={v_1, v_2, …, v_n} a set of vertices and a set of edges E={e_1, e_2, …, e_m}.
- Each edge is represented as a pair of vertices from V: $e_i=(v_j, v_k)$, where v_j and $v_k \varepsilon$ V.
  - Usually, vertices represent objects, or elements, and edges represent relationship
  - Vertices and Edges may be called Nodes and Links

Graphs

- Edges may be directed or undirected
  - Direction may be represented, visually, by an arrow
- A *directed* graph is a type of graph in which each edge (v_i , v_j) is directed, i.e., from v_i to v_j but not vice versa, otherwise it is *undirected*.

Graphs

- A *path*, from v_1 to v_k in a graph G(V, E) is a sequence of vertices (v_1, v_2, …, v_{k-1}, v_k) such that {(v_1, v_2), (v_2, v_3), …, (v_{k-1}, v_k)} exist in E.
- Length of a path is the number of edges in the path

Graphs

- A *cycle* in a graph G is a path (v_i, …, v_i)
- A *cyclic graph* is a graph in which there is any cycle, otherwise it is *acyclic*.

Graphs

- *Connected* graph is a graph in which there is a path from every vertex to all other vertices, otherwise it is disconnected.

Trees

- A *tree* is a connected acyclic graph.
- *Root :* a vertex with out-edges, no in-edges
- *Leaves: Vertices with* in-edges, no out-edges
- *Height* of a tree: length of the path from root to a leaf.
  - Sometimes called depth of a tree.

Graphs in CS

- A network is a graph
  - Network of computers
  - Network of people (social network)
- Trees are commonly used in CS
  - Indices in data and database
  - Search tree
- An ontology is a graph or a tree of concepts

Logic

Logic

- What is logic?

Logic

- Logic is the branch of science that studies the relationship between premises (Assumptions) & results (Conclusion).
- Logic is a branch of philosophy.
- In CS, Symbolic Logic
  - Well-Formed-Formulae (WFF)
    - Propositional
    - Predicate

Propositional Logic

- *Atomic Formula* is a proposition.
- A *proposition* is a statement that can be either true or false, but not both.
  - P: It is hot
  - Q: The air conditioner is on
  - R: Lights are on
    - P and Q: it is hot and the air conditioner is on

Propositional Logic

- WFF in Propositional Logic:
  - A proposition is a formula (Atomic Formula)
  - If F is a formula, then ~F (not F) is also a formula
  - If F & G are Formulae, then
    - F and G (Conjunction) is a formula
    - F or G (Disjunction) is a formula
    - If F then G is a formula
    - F iff G is a formula (if and only if)
  - Nothing Else is a formula
  - Note: Algebraic rules & DeMorgan's

Evaluation of a Formula

- A Proposition has a value (True or False)
- An *interpretation* of a formula is an assignment of each of its atoms by a value (true or false).
- A formula has a value (T or F) under each interpretation.

Evaluation of a Formula

- A formula is said to be *valid* if it is true under all its interpretations, otherwise it is *invalid*.
- A formula is said to be *inconsistent* if it is false under all its interpretations, otherwise it is *consistent*.
  - A valid formula is called a “tautology”
  - An inconsistent formula is called a contradiction

Predicate Logic

- A *predicate* is a logic statement that has an argument. Usually the predicate applies to the argument(s).
  - Student (John): John is a student
  - Has-book(John): John has book
  - Likes(John, Mary): John Likes Mary
  - Give(John, Mary, book)
  - An argument may be a variable
    - Student(x) has x as a variable
      - When x is substituted with a value, it is called an instance of the predicate.
      - Instantiation is the process.

Predicate Logic

- WFF in Predicate Logic
  - A predicate is a formula (Atomic Formula)
  - If F is a formula, then ~F (not F) is also a formula
  - If F & G are Formulae, then
    - F and G (Conjunction) is a formula
    - F or G (Disjunction) is a formula
    - If F then G is a formula
    - F iff G is a formula

Predicate Logic

- If F is a formula that contains a variable x defined over a domain D, then

  (“***For All***” x) F is evaluated as true only if it is true for each x ε D.
- If F is a formula that contains a variable x defined over a domain D, then

  (“***There Exists***” x) F is evaluated as true if at least one value of x ε D makes F true.
- Nothing Else is a formula

Notes on Quantifiers

- For All & There Exists are called quantifiers.
  - For all= for each= for every
  - There exist = for some = for at least one
  - ~for all (x) = there exist (~x)
  - ~there exist(x) = for all (~x)
  - ~for all (x) =/= for all (~x)
- A quantifier has a scope of variables.
- If all the variables in a formula are quantified, it is called *bound*, otherwise it is *loose*.

Evaluation of Formula

- The same definitions of Validity and Consistency apply in general to predicates.
- “For all x” F(x) is true if F is true for all values of x ε D
- “There Exists x” F(x) is true if any value of x ε D makes F true

EXAMPLES

- All basketball players are tall
  - $\forall X$ play(X, basketball) $\rightarrow$tall(X)
- John like anyone who likes books
  - like(X,book) $\rightarrow$like(john,X)
- Nobody likes taxes
  - $\neg \exists X$ likes(X,taxes)
- There is a person who writes computer class
  - $\exists X$ write(X,computer class)
- John did not study but he is lucky
  - $\neg$study(john) $\wedge$ luky(john)
- All cats and dogs are animals
  - $\forall X \forall Y$ cats(X) $\wedge$dogs(Y) $\rightarrow$animals(X)$\wedge$animals(Y)

Logical Consequence

- Given a set of formulae, G,

  we can say F is a logical consequence G,

  G |= F, if we can prove F using a subset of G.

  - {F U G} is consistent
  - {~F U G} is inconsistent

Proof Procedure

- The path from the assumptions to the result is called the *proof procedure*.

Logic in CS

- In AI there are many useful application of Logic.
  - Knowledge & knowledge base
    - Knowledge is composed of logic statements
    - Inference is the mechanism of using knowledge to infer more statements from asserted ones.
  - Semantic Web
    - RDF a language of semantic web based on logic, each RDF statement is a clause (OAV)
    - OWL a language for semantic web based on logic.
    - Association rules and logic statements