Introduction to Logic Programming

York University CSE 3401

Vida Movahedi

Overview

- Programming Language Paradigms
  - Logic Programming
  - Functional Programming
- Brief review of Logic
  - Propositional logic
  - Predicate logic

Why Logic Programming?

- View of the world imposed by a language

  A programming language tends to impose a certain view of the world on its users.

- Semantics of the programming languages

  To program with the constructs of a language requires thinking in terms of the semantics of those constructs

Programming Language Paradigms

- (1) Imperative programming
  - Semantics: state based
  - Computation viewed as state transition process
  - Categories:
    - Procedural
    - Object Oriented
    - Other non-structured
  - For example: C, Pascal, Turing are in the Procedural category, steps of computation describe state changing process

Programming Language Paradigms

- (2) Declarative Programming
  - Focus is on logic (WHAT) rather than control (HOW)
  - Categories:
    - Logic Programming: Computation is a reasoning process, e.g. Prolog
    - Functional Programming: Computation is the evaluation of a function, e.g. Lisp, Scheme, ...
    - Constrained Languages: Computation is viewed as constraint satisfaction problem, e.g. Prolog (R)
- Level of language
  - Low level
    - has a world view close to that of the computer
  - High level
    - has a world view closer to that of the specification (describing the problem to be solved, or the structure of the system to be presented)

Logic Programming

- Based on *first order predicate logic*
- A programmer *describes* with formulas of predicate logic
- A *mechanical problem solver* makes inferences from these formulas

Propositional Logic (review)

- Alphabet
  - Variables, e.g. p, q, r, ..., p_1, ..., p', ...
  - Constants: T and F
  - Connectives: $\{\neg, \wedge, \vee, \rightarrow, \equiv\}$
    - or {~, &, #, ->, <-> in some books}
  - Brackets: (and)

- Well-formed-formula (wff)
  - All variables and constants are wffs.
  - If A and B are wffs, then the following are also wffs.

$$(\neg A),\ \ (A \wedge B),\ \ (A \vee B),\ \ (A \rightarrow B),\ \ (A \equiv B)$$

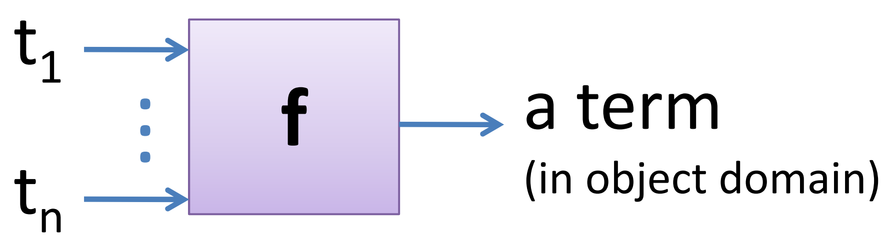
  - Priority of connectives, and rules for removing brackets

Propositional Logic (cont.)

- Semantics and truth tables
  - true (1) and false (0)
  - state
  - Tautologies: true in all possible states
- Satisfiable
  - A formula A is satisfiable iff there is at least one state v where v(A)=true
  - A set of formulae X is satisfiable (or consistent) iff there is at least one state v where for every formula A in X, v(A)=true.
- Contradiction: (unsatisfiable, inconsistent)
  - If A is a tautology, ¬A is a contradiction

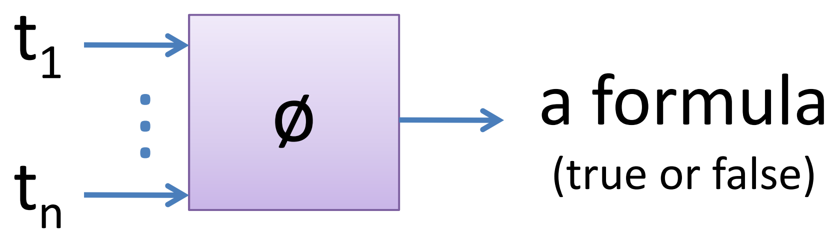
Predicate Logic (review)

- Alphabet
  - Alphabet of propositional logic
  - Object variables, e.g. x, y, z, ..., x_1, ..., x', ...
  - Object constants, e.g. a, b, c, ...
  - Object equality symbol =
  - Quantifier symbols $\forall$ (and $\exists$)
  - and some functions & predicates
- Term
  - An object variable or constant, e.g. x, a
  - A function f of n arguments, where each argument is a term, e.g. $f(t_1, t_2, \ldots t_n)$



Predicate Logic (cont.)

- Atomic formula
  - A Boolean variable or constant
  - The string t = s, where t and s are terms
  - A predicate ø of n arguments where each argument is a term , e.g. ø(t_1, t_2, ...t_n)



- Well-formed formula
  - Any atomic formula
  - If A and B are wffs, then the following are also wffs.

$(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, $(A \equiv B)$, $((\forall x)A)$, $((\exists x)A)$

Examples

- Numbers
  - Object constants: 1, 2, 3, ...
  - Functions: +, -, *, /, ...
  - Predicates: >, <, ...
  - Examples of wffs: $>(x, y) \rightarrow > (+(x,1), y)$

    Or the familiar notation: $x > y \rightarrow x+1 > y$

    Another example: $x! = z \rightarrow (x+1)! = (x+1) * z$

- Sets
  - Object constants: {1}, {2,3},...
  - Functions: $\bigcup$, $\bigcap$,...
  - Predicates: $\subset$, $\subseteq$,...
  - A wff: $(x \bigcap y) \subseteq (x \bigcup y)$

More Examples

- Our world
  - Object variables: X, Y, ...
    - upper case in PROLOG
  - Constants such as: john, mary, book, fish, flowers, ...
    - Note lower case in PROLOG
  - Functions: distance(point1, X), wife(john)
  - Predicates: owns(book, john), likes(mary, flowers), ...
  - true and false in PROLOG
    - Relative to PROLOG's knowledge of the world
    - False whenever it cannot find it in its database of facts (and rules)