CS370

Symbolic Programming
Declarative Programming

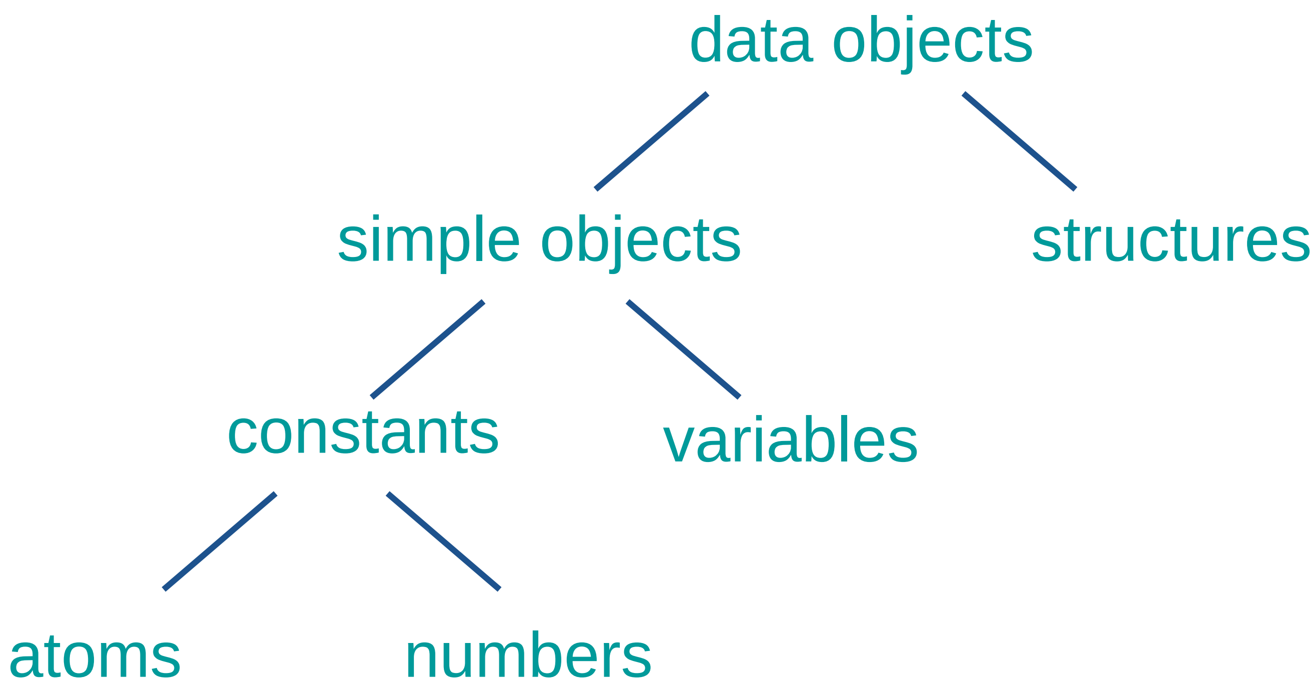
LECTURE 3: Syntax and Meaning of Prolog Programs

Jong C. Park
park@cs.kaist.ac.kr
Computer Science Department
Korea Advanced Institute of Science and Technology
http://nlp.kaist.ac.kr/~cs370

Syntax and Meaning of Prolog Programs

- Data objects
- Matching
- Declarative meaning of Prolog programs
- Procedural meaning
- Example: monkey and banana
- Order of clauses and goals
- The relation between Prolog and logic

Data Objects



Data objects

⦿ Atoms

- Strings of letters, digits and '_', starting with a lowercase letter
  - anna, nil, x25, x_25, miss_Jones, sarah_kerrighan
- Strings of special characters
  - <--->, ======>, ..., .:., ::=
- Strings of characters enclosed in single quotes
  - 'Tom', 'South_America', 'Sarah Kerrighan'

Data objects

Numbers

- Integers
  - 1
  - 1313
  - 0
  - -97
- Real numbers
  - 3.14
  - -0.0035
  - 100.2

Data Objects

Variables

- Strings of letters, digits, and '_' that start with an uppercase letter or '_'
  - X, Result, Object2, _x23, _23
- Anonymous variable: _
  - hasachild(X) :- parent(X,_)
  - somebody_has_child :- parent(_,_).
- Lexical scope of a variable

Data Objects

Structures

- Structures with 3 components
  - Examples
    - date(1,may,2001)
    - date(Day,may,2001)
- Terminology
  - Term
  - Functor
  - Argument
  - Principal functor

father(bob)

Data Objects

⊙ Structures

- Geometric objects in 2D space
  - Examples
    - P1 = point(1,1)
    - P2 = point(2,3)
    - S = seg(P1,P2)
      = seg(point(1,1),
      point(2,3))
    - T = triangle(point(4,2),
      point(6,4),
      point(7,1))
  - Any other ways?
    -

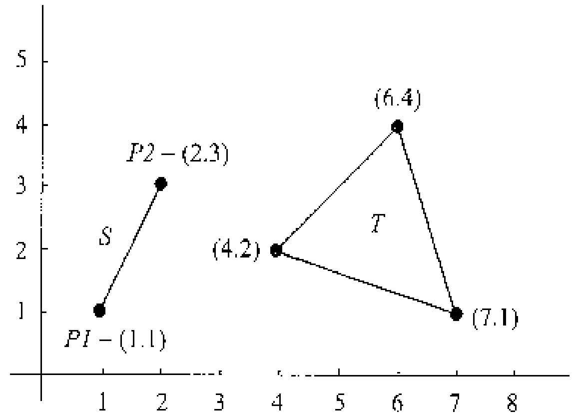


Figure 2.3 Some simple geometric objects.

Data Objects

Structures

- Representing objects in 3D space
  - Alternatives
    - point3(X,Y,Z)
    - point(X,Y,Z)
  - What are the pros and cons?
    -

Matching

- **Two terms match if**
  - they are identical, or
  - the variables in both terms can be instantiated to objects in such a way that after the substitution of variables by these objects the terms become identical
- **Examples**
  - date(D,M,2001) and date(D1,may,Y1)?
  - date(D,M,2001) and date(D1,M1,1444)?
  - date(X,Y,Z) and point(X,Y,Z)?

Matching

- **Given two terms S and T:**
  - If S and T are then S and T match only if they are the same object.
  - If S is a and T is anything, then they match, and S is instantiated to T. Conversely, if T is a variable then T is instantiated to S.
  - If S and T are then they match only if they have the same principal functor and all their corresponding components match.

Matching

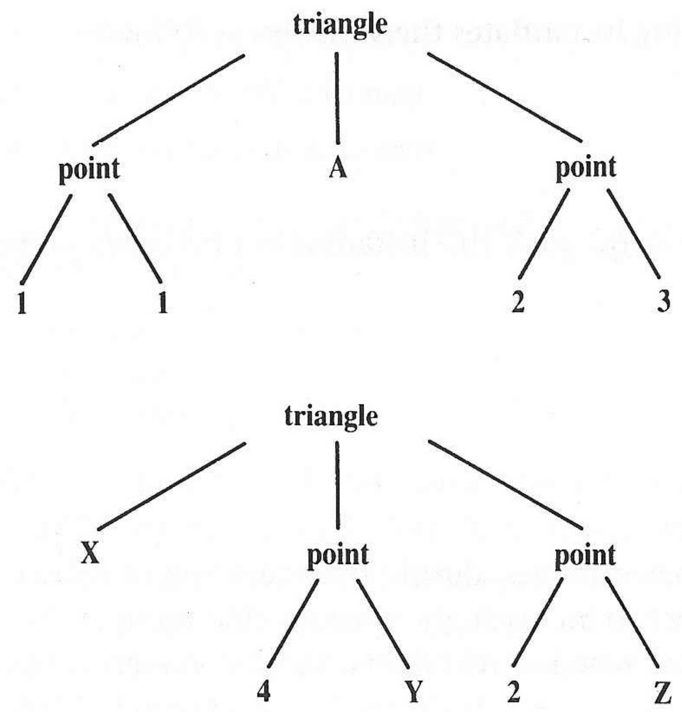


Figure 2.7 Matching **triangle(point(1,1), A, point(2,3)) = triangle(X, point(4,Y), point(2,Z))**.

Matching

⦿Example use of matching

- Recall the representation for line segments
  - S = seg(P1,P2) = seg(point(1,1),point(2,3))
- Define a piece of program for recognizing horizontal and vertical line segments

```
vertical(seg(point(X,Y),point(X,Y1)).
horizontal(seg(point(X,Y),point(X1,Y)).

?-
```

Prolog programs

Meanings

- P :- Q, R.
- declarative meaning
  - P is true if Q and R are true.
  - From Q and R follows P.
- procedural meaning
  - To solve problem P, first solve the subproblem Q and then the subproblem R.
  - To satisfy P, first satisfy Q and then R.

Prolog programs

Instances and variants of a clause

- Example
  - hasachild(X) :- parent(X,Y).
- A variant of a clause C is the clause C where each variable is substituted by another variable.
  - hasachild(A) :- parent(A,B).
  - hasachild(X1) :- parent(X1,X2).
- An instance of a clause C is the clause C with each of its variables substituted by some term.
  - hasachild(peter) :- parent(peter,Z).

Prolog programs

- **Given a program and a goal G, the declarative meaning says:**
  - A goal G is true iff:
    - there is a clause C in the program such that
    - there is a clause instance I of C such that
      - the head of I is identical to G, and
      - all the goals in the body of I are true.
- **A question is true if all of its goals are true for the same instantiation of variables.**

Prolog programs

Conjunction and disjunction of goals

- Conjunction of goals
  - P :- Q, R.
- Disjunction of goals
  - P :- Q; R.
    - P :- Q.
    - P :- R.
  - P :- Q, R; S, T, U.
    - P :- Q, R.
    - P :- S, T, U.

Summary

- Simple objects in Prolog are **atoms**, **variables** and **numbers**.
- Structured objects (**structures**) are used to represent objects that have several components.
- Structures are constructed by means of **functors**. Each functor is defined by its **name** and **arity**.
- The **type** of object is recognized entirely by its syntactic form.

Summary

- The **lexical scope** of variables is one clause. Thus the same variable name in two clauses means two different variables. **Structures** can be naturally pictured as **trees**. Prolog can be viewed as language for processing trees.
- The **matching** operation takes two terms and tries to make them identical by instantiating the variables in both terms.
- Matching, if it succeeds, results in the most general instantiation of variables.

Summary

- The **declarative semantics** of Prolog defines whether a goal is true wrt a given program, and if it is true, for what instantiation of variables it is true.
- A **comma** between goals means the conjunction of goals.
- A **semicolon** between goals means the disjunction of goals.