CS 603: Programming Languages

Lecture 31

Spring 2004

Department of Computer Science
University of Alabama
Joel Jones

Overview

- Questions on MP?
- Prolog Semantics (starting at 10.2.3 of text)
  - Logical Interpretation
  - Procedural Interpretation

Semantics: Logical Interpretation

- Prolog program: list of clauses C_1, ..., C_n and goal g, where each clause has the form G :- H_1, ..., H_n
- Prolog logical interpretation: prove right hand side then left hand side is true, finding variables that satisfy the constraints
- In other words, to satisfy the query g, find values for the variables occurring in g such that it can be proven using the clauses as inference rules

Example of Logical Interpretation

```
member(X, [Y|M]) :- member (X,M).
```

represents the assertion: for any values X, Y, and M, if X occurs in M, then X occurs in [Y|M]. Thus from

```
member(3,[4,3])
```

we can infer

```
member(3,[7,4,3]).
```

Similarly,

```
member(X,[X|L]).
```

states that member(X,[X|L]) is true no matter what the values of X and L

Logical Interpretation (cont.)

- But this is very precise, as we haven't really pinned down what is going on with variables.
- Our goals is a formulation of logical inference rules, which happen to be nondeterministic.

Logical: Definitions

- Theorem 10.1: A *substitution* θ is a function θ from terms to terms that satisfies...
  - Substitution must be finite
  - Expressed as set of bindings
  - If θ_1 and θ_2 are substitutions, so is $\theta_1 \circ \theta_2$
- Definition 10.2: The application of a substitution to a goal is the goal obtained by applying the substitution to each argument of the goal

Logical: Precise Inference Rules

- Goal *g* is satisfiable using database *D* and substitution θ^n as the judgment ...
- In the general case, a query can have more than one goal, so ...
- In the logical interpretation, satisfaction of different goals is independent, but requires that the same substitution satisfy them all
- See **LOGICALQUERIES** on p. 447

Logical: Precise Inference Rules (cont.)

- A substitution θ satisfies a query g if there is some clause in the database such that another substitution makes the conclusion the same as g, and under that substitution we can prove all the premises
- See **LOGICALQUERY** on p. 447
- Summarizing, the logical interpretation says that to satisfy query g, there must be a pair of substitutions θ and θ' such that θ applied to the original query is the same as θ' applied to the head of some clause, and the subgoals produced by applying θ' to the subgoals of that clause are all satisfiable.

Logical: How make deterministic?

- *Which* clause $C \sqcup D$ do we choose when applying rule **LogicalQuery**?
- Given g and G, how do we discover a pair of substitutions that cause a match ... ?
- Unification

Logical: Definitions for Unification

- Definition 10.3: A substitution θ_1 is *more general* than a substitution θ_2 if there exists a θ_3 such that $\theta_2 = \theta_3 \circ \theta_1$. The more general a substitution is, the fewer things it changes.
- Definition 10.4: *Unification* is the process of finding, for given goals g_1 and g_2, a substitution θ that unifies g_1 and g_2. Furthermore, θ must be a *most general* substitution.
- Definition 10.5: A *renaming of variables* is a substitution θ_ρ in which θ_ρ (VAR(*X*)) is always a variable, never an application or an integer.