- Project #2 - due today. Demos this week!
- Last week!
- Practice exam questions on web page.

"In Prolog, as in most halfway decent programming languages, there is no tension between writing a beautiful program an writing an efficient program. If your Prolog code is ugly, the chances are that you either don't understand your problem or don't understand your programming language, and in neither case does your code stand much chance of being efficient. In order to ensure your program is efficient, you need to know what it is doing, and if your code is ugly, you will find it hard to analyse."

Richard A. O'Keefe, "The Craft of Prolog", 1990.

Since Last midterm

- difference lists, definite clause grammars and natural language interfaces to databases
- computer algebra and calculus
- Triples are universal representations of relations, and are the basis for RDF, and knowledge graphs
- URIs/IRIs provide constants that have standard meanings
- Ontologies define the meaning of symbols used in information systems.
- You should know what the following mean: RDF, IRI, rdf:type, rdfs:subClassOf, rdfs:domain, rdfs:range
- Complete knowledge assumption and negation as failure
- Extra-logical predicates

Today

- Unification and Proofs.

Or

Prolog has an "or" written as ;
Defined by

```
(A ; _) :- call(A).
(_ ; B) :- call(B).
```

Reasoning with Variables

- An instance of an atom or a clause is obtained by uniformly substituting terms for variables. Every instance of the same variable is replaced by the same term.
- A substitution is a finite set of the form $\{V_1/t_1, \ldots, V_n/t_n\}$, where each V_i is a distinct variable and each t_i is a term.
- The application of a substitution $\sigma = \{V_1/t_1, \ldots, V_n/t_n\}$ to an atom or clause e, written $e\sigma$, is the instance of e with every occurrence of V_i replaced by t_i.

Application Examples

The following are substitutions:

$$\sigma_1 = \{X/A, Y/b, Z/C, D/e\}$$
$$\sigma_2 = \{A/X, Y/b, C/Z, D/e\}$$
$$\sigma_3 = \{A/V, X/V, Y/b, C/W, Z/W, D/e\}$$

The following shows some applications:

$$p(A, b, C, D)\sigma_1 = p(A, b, C, e)$$
$$p(X, Y, Z, e)\sigma_1 = p(A, b, C, e)$$
$$p(A, b, C, D)\sigma_2 = p(X, b, Z, e)$$
$$p(X, Y, Z, e)\sigma_2 = p(X, b, Z, e)$$
$$p(A, b, C, D)\sigma_3 = p(V, b, W, e)$$
$$p(X, Y, Z, e)\sigma_3 = p(V, b, W, e)$$

Application Examples

Given the substitution:

$$\sigma = \{X/A, Y/b, Z/C, D/e\}$$

$foo(D, Z, C, A)\sigma$ is

A $foo(D, Z, C, A)$

B $foo(e, C, C, A)$

C $foo(D, C, C, X)$

D $foo(e, C, C, X)$

E $foo(e, C, Z, A)$

Application Examples

Given the substitution:

$$\sigma = \{X/A, Y/b, Z/C, D/e\}$$

$foo(W, b, C, A)\sigma$ is

A $foo(X, Y, Z, D)$

B $foo(b, b, C, Y)$

C $foo(W, Y, C, X)$

D $foo(W, b, C, A)$

E $foo(W, Y, C, A)$

Unifiers

- Substitution σ is a unifier of e_1 and e_2 if $e_1\sigma = e_2\sigma$.
- Substitution σ is a most general unifier (mgu) of e_1 and e_2 if
  - σ is a unifier of e_1 and e_2; and
  - if substitution σ' also unifies e_1 and e_2, then $e\sigma'$ is an instance of $e\sigma$ for all atoms e.
- If two atoms have a unifier, they have a most general unifier.
- If there are more than one most general unifiers, they only differ in the names of the variables.

Unification Example

A yes

B no

C I'm not sure

Is the substitution a unifier of $p(A, b, C, D)$ and $p(X, Y, Z, e)$:

$\sigma_1 = \{X/A, Y/b, Z/C, D/e\}$ yes

$\sigma_2 = \{Y/b, D/e\}$ no

$\sigma_3 = \{X/A, Y/b, Z/C, D/e, W/a\}$ yes

$\sigma_4 = \{A/X, Y/b, C/Z, D/e\}$ yes

$\sigma_5 = \{X/a, Y/b, Z/c, D/e\}$ no

$\sigma_6 = \{A/a, X/a, Y/b, C/c, Z/c, D/e\}$ yes

$\sigma_7 = \{A/V, X/V, Y/b, C/W, Z/W, D/e\}$ yes

$\sigma_8 = \{X/A, Y/b, Z/A, C/A, D/e\}$ yes

Which are most general unifiers? σ_1, σ_4

procedure $unify(t_1, t_2)$ ▷ Returns mgu of t_1 and t_2 or $\bot$.

  $E := \{t_1 = t_2\}$ ▷ Set of equality statements

  $S := \{\}$ ▷ Substitution

  while $E \neq \{\}$ **do**

    select and remove $x = y$ from E

    if y is not identical to x **then**

      if x is a variable **then**

        replace x with y in E and S

        $S := \{x/y\} \cup S$

      else if y is a variable **then**

        replace y with x in E and S

        $S := \{y/x\} \cup S$

      else if x is $p(x_1, \ldots, x_n)$ and y is $p(y_1, \ldots, y_n)$ **then**

        $E := E \cup \{x_1 = y_1, \ldots, x_n = y_n\}$

      else

        return $\bot$ ▷ t_1 and t_2 do not unify

  return S ▷ S is mgu of t_1 and t_2

Examples

- unify $p(A, b, C, D)$ and $p(X, Y, Z, e)$
  $\{A/X, Y/b, C/Z, D/e\}$
- unify $p(A, b, A, D)$ and $p(X, X, Z, Z)$
  $\{A/b, X/b, Z/b, D/b\}$
- unify $p(A, b, A, d)$ and $p(X, X, Z, Z)$
  $\bot$
- unify $n([sam, likes, prolog], L2, I, C1, C2)$ and
  $n([P|R], R, P, [person(P)|C], C)$
  $\{P/sam,\ R/[likes, prolog],\ L2/[likes, prolog],\ I/sam,$
  $C1/[person(sam)|C2],\ C/C2\}$

Top-down Propositional Proof Procedure (recall)

- Idea: search backward from a query to determine if it is a logical consequence of KB.
- An answer clause is of the form:

  $$yes \text{ :- } a_1,\ a_2,\ \ldots,\ a_m$$

- The (SLD) resolution of this answer clause on atom a_1 with the clause in the knowledge base:

  $$a_1 \text{ :- } b_1,\ \ldots,\ b_p$$

  is the answer clause

  $$yes \text{ :- } b_1,\ \cdots,\ b_p,\ a_2,\ \cdots,\ a_m.$$

  A fact in the knowledge base is considered as a clause where $p = 0$.