University of Auckland

07.363

Functional and Logic Programming

Christian Collberg

October 7, 1994

Homework 2

Understanding Prolog 1 – I

```
| ?- append([], [1, 2|B], C),
        append([3,4], [5], B).
   B = [3,4,5],
   C = [1,2,3,4,5] ;
   no

| ?- bagof(X,
      Y^append(X, Y, [1,2,3,4]), Xs).
   X = _9050,
   Y = _9067,
   Xs = [[],[1],[1,2],
         [1,2,3],[1,2,3,4]] ;
   no
```

Slide 20–1

Understanding Prolog 1 – II

```
| ?- L=[1,2], member(X, L),
        delete(X, Y, L).
   L = [1,2],
   X = 1,
   Y = [1,1,2] ;

   L = [1,2],
   X = 2,
   Y = [2,1,2] ;
   no
```

Slide 20–2

Understanding Prolog 2 – I

```
balance(john, 100).
balance(sue, 200).
balance(mary, 100).
balance(paul, 500).

| ?- bagof(Name,
      balance(Name, Amount), Names).
   Name = _9248,
   Amount = 100, Names=[john,mary];

   Name = _9248,
   Amount = 200, Names = [sue] ;

   Name = _9248,
   Amount = 500, Names = [paul] ;
   no
```

Slide 20–3

Understanding Prolog 2 – II

```
| ?- bagof(Name,
     Amount^balance(Name,Amount), Names).
  Name = _9248,
  Amount = _9267,
  Names = [john,sue,mary,paul] ;
  no

| ?- bagof(Name,
     Name^balance(Name, Amount), Names).
  Name = _9248, Amount = 100,
  Names = [john,mary] ;

  Name = _9248, Amount = 200,
  Names = [sue] ;

  Name = _9248, Amount = 500,
  Names = [paul] ;
  no
```

Slide 20–4

Understanding Prolog 3

Describe (in English) what the following predicate does:

```
% Both arguments to
% bbb are lists.
bbb([], []).
bbb(A, [X|F]) :-
        append(F, [X], A).
```

Answer

A one step right circular shift.

Slide 20–5

Understanding Prolog 4

Given the following program

```
a(1, 2).
a(3, 5).
a(R, S) :- b(R, S), b(S, R).

b(1, 3).
b(2, 3).
b(3, T) :- b(2, T), b(1, T).
```

list the first answer to this query:

```
?- a(X, Y), b(X, Y)
   X = 3
   Y = 3 ;
   no
```

Slide 20–6

Understanding Prolog 5

```
f(1, one).
f(s(1 ), two).
f(s(s(1)), three).
f(s(s(s(X))), N) :- f(X, N).
```

What are the results of these queries?

1. `?- f(s(1), A).` ⇒ `A=two; no`
2. `?- f(s(s(1), two).` ⇒ `no`
3. `?- f(s(s(s(s(s(s(1)))))), C).` ⇒ `C=1; no.`
4. `?- f(D, three).` ⇒ `D=s(s(1)); D=s(s(s(s(s(1)))))`

Slide 20–7

Prolog Programming 1

Write a pedicate that sums the absolute differences between two integer lists:

```
?- sum_abs_diffs([1,2,3], [5,4,2], X).
     X = 7 % abs(1-5)+abs(2-4)+abs(3-2)
```

Answer

```
sum_abs_diff([], [], 0).
sum_abs_diff([X|T1], [Y|T2], N) :-
   X >= Y, D is X - Y,
   sum_abs_diff(T1, T2, N1),
   N is N1 + D.
sum_abs_diff([X|T1], [Y|T2], N) :-
   X < Y, D is Y - X,
   sum_abs_diff(T1, T2, N1),
   N is N1 + D.
```

Slide 20–8

Prolog Programming 2 – I

Write a Prolog predicate which transposes a rectangular matrix given in row-major order.

Transpose:

$$\begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} \Rightarrow \begin{vmatrix} 1 & 3 \\ 2 & 4 \end{vmatrix}$$

$$\begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix} \Rightarrow \begin{vmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{vmatrix}$$

Answer:

```
tr([[]|_], []).
tr(A, [F|B]) :-
   ex(A, F, R), tr(R, B).
```

Slide 20–9

Prolog Programming 2 – II

```
% Extract the first element of each list.
% ex(ListOfLists,
%        ListOfFirstElements, Remainder)
ex([], [], []).
ex([[X|T1]| T2], [X|F], [T1| R]) :-
   ex(T2, F, R).

?- tr([[1, 2], [3, 4]], B).
   B = [[1,3],[2,4]]

?- tr([[1, 2, 3], [4, 5, 6],
         [7, 8, 9]], B).
   B = [[1,4,7],[2,5,8],[3,6,9]]

?- ex([[1, 2], [3, 4]], F, R).
   F =[1,3]
   R = [[2],[4]]
```

Slide 20–10

Prolog Programming 3 – I

```
%country(name, population, capital).
country(sweden, 8823, stockholm).
country(usa, 221000, washington).
country(france, 56000, paris).
country(denmark, 3400, copenhagen).

% city(name, in_country, population).
city(lund, sweden, 88).
city(new_york, usa, 5000).
city(paris, usa, 1).  % Paris, Texas.
city(copenhagen, denmark, 1200).
city(aarhus, denmark, 330).
city(odense, denmark, 120).
city(stockholm, sweden, 1300).
city(gothenburg, sweden, 350).
city(washington, usa, 3400).
city(paris, france, 2000).
city(marseilles, france, 1000).
```

Slide 20–11

Prolog Programming 3 – II

```
% Which countries have cities with the same
% name as capitals of other countries?
same_name :-
   country(L1, _, S), city(S, L2, _),
   not(L1 = L2),
   write(S), write(' '), write(L1),
   write(' '), write(L2), nl, fail.
same_name.
| ?- same_name.
      paris france usa

% In how many countries do more than 1/3 of
% the population live in the capital?
big_cap(L) :-
   country(L, X, C), city(C, L, Y), 3 * Y > X.
count_big_capital(N) :-
   bagof(C, big_cap(C), L), length(L, N).
| ?- big_cap(N).
   N = denmark
| ?- count_big_capital(N).
   N = 1
```

Slide 20–12

Prolog Programming 3 – III

```
% Which capitals have a population more
% than 3 times larger than that of the
% secondmost populous city?
capital_3times_bigger(C) :-
   country(L, _, C),
   city(C, L, X),
   three_times_bigger(C, L, X).

three_times_bigger(C, L, X) :-
   city(S, L, Y),
   not(C = S),
   X =< 3 * Y, !, fail.
three_times_bigger(_, _, _).
| ?- capital_3times_bigger(C).
   C = stockholm ;
   C = copenhagen ;
   no
```

Slide 20–13

Prolog Programming 4

Write a Prolog predicate that extracts all words immediately following "the" in a given list of words.

```
find([], []).
find([_], []).
find([the, X | T1], [X|T2]) :-
   find([X|T1], T2).
find([X | T1], T2) :-
   X \= the, find(T1, T2).

?- find([the, man, closed, the,
        door, of, the, house], M).
   [man,door,house]
?- find([the, the], M).
   [the]
?- find([man, the], M).
   []
```

Slide 20–14

Prolog Programming 5

```
long_dist_call(From_Cit, Cnt, To_Cit, Cnt, LN)-->
   {long_dist_access_code(LDAC), member(Cnt/LC, LDAC)}, LC,
   {area_code(Cnt, ACL), member(To_Cit/AC, ACL)}, AC,
   LN. % Local Number

int_call(From_Cit, From_Cnt, To_Cit, To_Cnt, LN)-->
   {int_access_code(ICL), member(From_Cnt/IC, ICL)}, IC,
   {Cnt_code(CCL), member(To_Cnt/CC, CCL)}, CC,
   {area_code(Cnt, ACL), member(To_Cit/AC, ACL)}, AC,
   LN. % Local Number
```

Slide 20–15

Logic 1

1. A quick brown fox jumped over the lazy dog.
   - $\exists F \bullet \mathrm{fox}(F) \wedge \mathrm{quick}(F) \wedge \mathrm{brown}(F) \wedge \exists D \bullet \mathrm{dog}(D) \wedge \mathrm{lazy}(D) \wedge \mathrm{jumped\_over}(F, D)$
2. Some quick brown foxes jumped over all the lazy dogs.
   - $\forall D \bullet (\mathrm{dog}(D) \wedge \mathrm{lazy}(D)) \Rightarrow \exists F \bullet \mathrm{fox}(F) \wedge \mathrm{quick}(F) \wedge \mathrm{brown}(F) \wedge \mathrm{jumped\_over}(F, D)$
3. All drivers must make at least one pit stop if the race is longer than 200 miles.
   - $\mathrm{race\_longer\_than\_200} \Rightarrow \forall D \bullet \mathrm{driver}(D) \Rightarrow \exists P \bullet \mathrm{pit\_stop}(P) \wedge \mathrm{make}(D, P)$

Slide 20–16

Logic 1

4. All students sitting the 363 test are happy with some of the questions.
   - $\forall S \bullet (\mathrm{student}(S) \wedge \mathrm{sits}(S, \mathrm{363\_exam})) \Rightarrow \exists Q \bullet \mathrm{question}(Q) \wedge \mathrm{happy\_with}(S, Q)$
5. Every tree in Albert Park has some leaves on it.
   - $\forall T \bullet (\mathrm{tree}(T) \wedge \mathrm{in\_a\_p}(T)) \Rightarrow \exists L \bullet \mathrm{leaf}(L) \wedge \mathrm{on}(L, T)$
6. Some buildings have all offices either air conditioned or centrally heated.
   - $\exists B \bullet \mathrm{building}(B) \wedge (\forall O \bullet (\mathrm{office}(O) \wedge \mathrm{in}(O, B)) \Rightarrow (\mathrm{a\_c}(O) \vee \mathrm{c\_h}(O)))$

Slide 20–17

Logic 2 – I

$(P \wedge Q) \Rightarrow R, P \wedge \neg R \models \neg Q$

P	Q	R	$(P \wedge Q) \Rightarrow R$	$P \wedge \neg R$	$\models$	$\neg Q$	
T	T	T	T	F		F	
T	T	F	F	T		F	
T	F	T	T	F		T	
T	F	F	T	T		T	$\Leftarrow$
F	T	T	T	F		F	
F	T	F	T	F		F	
F	F	T	T	F		T	
F	F	F	T	F		T	

Slide 20–18

Logic 2 – II

$\neg R \Rightarrow \neg P, \neg R \Rightarrow \neg Q \models (P \vee Q) \Rightarrow R$

P	Q	R	$\neg R \Rightarrow \neg P$	$\neg R \Rightarrow \neg Q$	$\models$	$(P \vee Q) \Rightarrow R$	
T	T	T	T	T		T	$\Leftarrow$
T	T	F	F	F		F	
T	F	T	T	T		T	$\Leftarrow$
T	F	F	F	T		F	
F	T	T	T	T		T	$\Leftarrow$
F	T	F	T	F		F	
F	F	T	T	T		T	$\Leftarrow$
F	F	F	T	T		T	$\Leftarrow$

Slide 20–19