University of Auckland

07.363

Functional and Logic Programming

Christian Collberg

October 7, 1994

Arithmetic

Equality I

`X = Y, X \= Y`
: Can `X` be unified/**not** be unified with `Y`?

`X == Y, X \== Y`
: Do `X` and `Y` have identical/different structure?

`a=b`	`no`	`a(X) = a(Y)`	`yes`
`a=a`	`yes`	`a(X) == a(Y)`	`no`
`a==a`	`yes`	`a(X) \== a(Y)`	`yes`
`a==b`	`no`	`a(X) == a(X)`	`yes`

Slide 5–1

Equality II

`X < Y, X > Y`
: Is `X` less than/greater than `Y`?

`X=< Y, X >= Y`
: Is `X` less than or equal/greater than or equal to `Y`?

`5 < 7`	`yes`
`5 <= 7`	`yes`
`5 <= X`	`error (if X is uninst.)`
`X <= 7`	`error (if X is uninst.)`

Slide 5–2

Equality III

`X` **is** `E`
: Give `X` the value of the expression `E`.

`E1 =:= E2, E1 =\= E2`
: Do the expressions `E1` and `E2` evaluate to the same/different values?

X **is** 7+5	12
X **is** Y+14	error (if Y is uninst.)
Y+2 =:= 2	error (if Y is uninst.)
5+2 =:= 14/2	yes

Slide 5–3

Arithmetic In Logic I

```
0            0
s(0)         1
s(s(0))      2
s(s(s(0)))   3

nat(0).
nat(s(X)) :- nat(X).

?- nat(s(s(s))).
      yes

?- nat(X).
      X = 0 ;
      X = s(0) ;
      X = s(s(0)) ;
```

Slide 5–4

Arithmetic In Logic II

```
plus(0, X, X) :- nat(X).
plus(s(X), Y, s(Z)):- plus(X,Y,Z).

minus(X, Y, Z) :- plus (Z, Y, X).

?- plus(s(s(0)), s(0), Z).
      Z = s(s(s(0)))

?- plus(X, Y, s(s(s(0)))).
   X = 0, Y = s(s(s(0))) ;
   X = s(0), Y = s(s(0)) ;
   X = s(s(0)), Y = s(0) ;
   X = s(s(s(0))), Y = 0
```

Slide 5–5

Arithmetic In Logic II

```
less_or_eq(0, X) :- nat(X).
less_or_eq(s(X), s(Y)) :- less_or_eq(X, Y).

times(0, X, 0) :- nat(X).
times(s(X), Y, Z) :- times(X, Y, W), plus(W,Y,Z).

fac(0, s(0)).
fac(s(N), F) :- fac(N, F1), times(s(N), F1, F).
```

Slide 5–6

Arithmetic In Logic III

```
fac(0, s(0)).
fac(s(N), F) :- fac(N, F1), times(s(N), F1, F).
?- fac(s(s(s(0))), Z).
      Z = s(s(s(s(s(s(0))))))

?- fac(Z, s(s(s(s(s(s(0))))))).
      Z = s(s(s(0)))
```

Slide 5–7

Prolog Arithmetic I

Differences between Logic Arithmetic and Prolog Arithmetic:

1. Programs become specialized for a single use (`plus` can't be used to define `minus`).
2. No recursive structure of numbers.

Slide 5–8

Prolog Arithmetic II

```
fac1(0, s(0)).
fac1(s(N), F) :- fac1(N, F1),
   times(s(N), F1, F).

fac2(N, F) :-
   N > 0, N1 is N-1,
   fac2(N1, F1), F is N * F1.
fac2(0, 1).

?- fac2(3, Z).
      Z = 6

?- fac2(Z, 6).
      error
```

Slide 5–9

Prolog Arithmetic III

X + Y	sum
X - Y	difference
X ∗ Y	product
X / Y	quotient
X mod Y	remainder

`X is 5 + 4`	`9`
`X is 5 + X`	`error`
`X is 15 / 2`	`7.5`
`X is 15 mod 2`	`1`

Slide 5–10

Arithmetic – Prince of Wales I

```
reigns(rhodri, 844, 878).
reigns(anarawd, 878, 916).
reigns(hywel_dda, 916, 950).
reigns(lago_ad_idwal, 950, 979).
reigns(hywel_ab_ieuaf, 979, 985).
reigns(cadwallon, 985, 986).
reigns(maredudd, 986, 999).

prince(X, Y) :-
   reigns(X, A, B),
   Y >= A,
   Y =< B.
```

Slide 5–11

Arithmetic – Prince of Wales II

```
?- prince(cadwallon, 986).
   yes

?- prince(charles, 1994).
   no

?- prince(X, 900).
   X = anarawd

?- prince(X, 979).
   X = lago_ab_idwal ;
   X = hywel_ab_ieuaf ;
   no
```

Slide 5–12

Arithmetic In Logic IV

How can we represent S as the sum of 2 numbers?

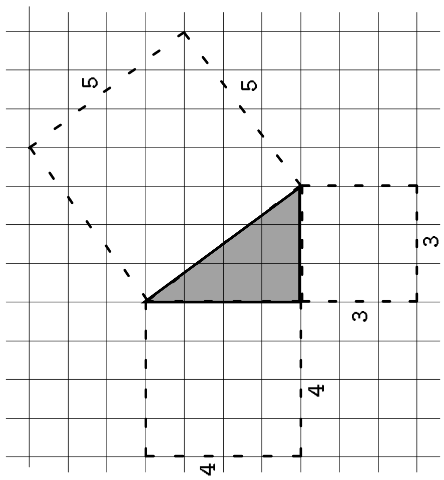
% minus(S, D1, D2) -- $S - D_1 = D_2$

```
minus(S, S, 0).
minus(S, D1, D2) :-
      S > 0, S1 is S-1,
      minus(S1, D1, D3),
      D2 is D3 + 1.

?- minus(3, X, Y).
   X = 3, Y = 0 ;
   X = 2, Y = 1 ;
   X = 1, Y = 2 ;
   X = 0, Y = 3
```

Slide 5–13

Pythagorean Triples I



Slide 5–14

Pythagorean Triples II

```
is_int(0).
is_int(X) :- is_int(Y), X is Y+1.

pythag(X, Y, Z) :-
      int_triple(X, Y, Z),
      Z*Z =:= X*X + Y*Y.

int_triple(X, Y, Z) :-
      is_int(S),
      minus(S, X, S1), X > 0,
      minus(S1, Y, Z), Y > 0, Y > 0.

minus(S, S, 0).
minus(S, D1, D2) :-
      S > 0, S1 is S-1,
      minus(S1, D1, D3),
      D2 is D3 + 1.
```

Slide 5–15

Pythagorean Triples III

```
?- pythag(X, Y, Z).
X = 4, Y = 3, Z = 5 ;

X = 3, Y = 4, Z = 5 ;

X = 8, Y = 6, Z = 10 ;

X = 6, Y = 8, Z = 10 ;

X = 12, Y = 5, Z = 13 ;

X = 5, Y = 12, Z = 13 ;

X = 12, Y = 9, Z = 15
```

Slide 5–16