CSCC24 – Principles of Programming Languages

Anya Tafliovich[1]

[1] with thanks to S.McIlraigh, G.Penn, P.Ragde

Logic Programming and Prolog

Logic programming languages are neither procedural nor functional.

- Specify **relations** between objects
  - `larger(3,2)`
  - `father(tom,jane)`
- Separate logic from control:
  - Programmer declares **what** facts and relations are true.
  - System determines **how** to use facts to solve problems.
  - System **instantiates** variables in order to make relations true.

Prolog

Suppose we state these **facts**:

```
male(charlie).
female(alice).
male(bob).
female(eve).
```

```
parent(charlie,bob).
parent(eve,bob).
parent(charlie,alice).
parent(eve,alice).
```

We can then make **queries**:

```
?- male(charlie).
true

?- male(eve).
false

?- female(Person).
Person = alice;
Person = eve;
false
```

```
?- parent(Person, bob).
Person = charlie;
Person = eve;
false

?- parent(Person, bob),
   female(Person).
Person = eve;
false
```

Prolog

We can also state **rules**, such as this one:

```
sibling(X, Y) :- parent(P, X),
                 parent(P, Y).
```

Then the queries become more interesting:

```
?- sibling(charlie, eve).
false

?- sibling(bob, Sib).
Sib = bob;
Sib = alice;
Sib = bob;
Sib = alice;
false
```