CSCU9Y4 Spring 2018

Programming Language Paradigms

Tutorial 18/7

Reading for this tutorial: Supplementary Material 7, and doing the lab sheets.

Sections 1.8.2, 10 intro, 10.1, 10.3, 10.6 of Introduction to Programming Languages (the other sections will provide richer background, especially 10.2, but you don't need them for these examples), Chapter 10 of Comparative Programming Languages. Alternatively Ch 15 of Watt, or Ch 16 and 2.13 of Sebesta. Or google "Prolog" or "Logic Programming".

1. Supposing we have the following Prolog database indicating supervisors details in a company.

   `supervisor_of(Boss,Employee)` is read as 'Boss is the supervisor of Employee'.

```
supervisor_of(bill, freda).
supervisor_of(fred, bill).
supervisor_of(freda, jane).
supervisor_of(freda, allan).
supervisor_of(allan, ashley).
supervisor_of(jane, emily).
supervisor_of(emily, anne).
supervisor_of(emily, linda).
```

(i) Write a query to find out who is supervised by allan.

(ii) Write a predicate **`above_and_below(Person)`** to output the name of the employeee who supervises **`Person`**, and the name of one employee whom **`Person`** supervises. Check your solution with suitable tests.

(iii) Some employees supervise more than one person. Write a predicate **`all_supervisees(Person)`** that will list all the people supervised by **`Person`**.

(iv) Trace through the steps taken by Prolog when issued with the goal:

```
all_supervisees(emily).
```

(v) Supposing we wish to find out all those that are higher up in a direct line of command than an individual. So for example, freda is more senior than jane, bill is more senior than freda, fred is more senior than bill. So freda, bill and fred are all more senior than jane.
Write a predicate **`senior_to(Person)`** that outputs the names of all people more senior than **`Person`**.
What assumptions are you making about the **`supervisor_of`** relationship here?

(vi) Trace through the steps taken by Prolog when issued with the goal:

```
senior_to(jane).
```

(vii) Taking the predicate **`senior_to(Person)`** as a template, form a new predicate **`junior_of(Person)`** that lists all staff who are junior to a given member of staff **`Person`** (i.e. lower down in the management hierarchy – all supervisees, and all their supervisees, and so on). Check your solution with suitable tests. What is the significant difference between traversing the **`supervisor_of`** relationship for juniors compared to seniors?

2. Logic programming often uses arguments to deduce whether something is true or false. Give an example of a deductive argument and an inductive argument explaining the difference between each. (Don't just give the examples from the lecture materials!)

3. Prolog is an *interpreted* language rather than a *compiled* one. Explain what this means, and what difference it makes to the user.