MA410 Prolog Practical 5 - Unifiers and Proof Techniques

1 Most General Unifers

(a) Open Prolog and run these examples:

i. `?- f(g(X)) = f(Y), t(X) = t(u).`
ii. `?- g(X,y) = g(x,Y).`
iii. `?- g(d(F),B,C,d(E)) = g(E,a(F),b(B),d(d(t))).`
iv. `?- h(A,B,A) = h(r,C,A), h(C,r,C) = h(B,A,b).`
v. `?- [prolog, is, cool] = [H | T].`
vi. `?- [prolog, is, cool] = [H1, H2 | T].`
vii. `?- [There, is, three, erreurs, in, this, sentence] = [find | THEM].`

Task: Make up three funny examples (include one example that fails) and e-mail these to me with the subject heading "`Lab Proof Techniques - MGUs`".

2 Top-Down Proof (No Variables)

2.1 Tracing in Prolog

The `trace` command in Prolog allows you to see all of the goals that are executed as part of the query, in sequence, and also whether they succeed or not. It also allows you to see what steps occur as Prolog backtracks. We will use this as a tool to understand how prolog performs top-down proofs.

2.2 Examples

- Go to the *demonstrations* section on the course website and run the top-down demo there at `top_down_demo.pdf`. See if you understand it.

2.2.1 Example 1

(a) Download `lab5td1.pl` and consult it in Prolog.

(b) Run the query `?- trace.` followed by `?- health.`

Prolog should prompt you with:

```
Call:  (7) health ?
```

At this point press return and continue to press return until all the sequences are finished and it returns the answer `true`. Prolog shows you how it performs a top-down proof in each of the steps.

(c) To turn off trace and debugging, use the queries '`?- notrace.`' and '`?- nodebug.`' respectively.

2.2.2 Example 2

(a) Restart Prolog.

(b) Download `lab5td2.pl` and consult it in Prolog.

(c) Again run query `?- trace.` and then `?- health.`

Continue to press return as before. Eventually, Prolog on this occasion should return the answer `false`.

- Compare the steps taken in prolog to that of the demonstration.

3 Top-Down Proof (With Variables)

3.1 Example 1: Lion Loose at NUIG

(a) Go to the *demonstrations* section and run the first example "Lion Loose at NUIG" example there at `top_down_with_vars_demo.pdf`. Note that SLD stands for "Selective Linear Definite" and SLD Resolution is esentially the top-down proof method with variables.

(b) Download and edit the file `lab5lion.pl`.

(c) Compare the code with the demonstration file.

(d) Load it into prolog.

(e) Run the query: `?- trace.`

and this time followed by: `?- caught(X).`

Keep pressing return (trying to understand the process prolog is using at the same time) until the first answer '`X = lecturer`' is given. Then press ';' and search for the next answer.

Note how prolog applies MGUs and backtracks.

3.2 Example 2: Ancestor Example

This example illustrates recursion.

(a) Run the second example in the file `top_down_with_vars_demo.pdf`.

(b) Download and edit the file `lab5anc.pl`.

(c) Compare the code with the demonstration file.

(d) Load it into prolog.

(e) Run the query: `?- trace.`

followed by: `?- ancestor(X, liam).`

Keep pressing return (compare the process as in the demonstration) until the first answer '`X = padhraic`' is given. Then press ';' and repeat the process to find all answers.

Note how prolog backtracks in this example and reapplies the same definition.

Task: Replace the characters `lion`, `lecturer` and `nuig_student` with some alternative crazy characters. E-mail code with subject heading "`Lab Proof Techniques - Chasing Gone Wrong`".