Prolog Lab 1

University of Wales, Cardiff

This exercise is not assessed, but it is in your interest to practise this, because SOON there might be one that is.

1. Which of the following terms :

```
apple   '_5764'   Mary   _atom
bar(5,_,a)   student   atom's   123.87.4
[1,_,t]   []   -2365   g(V)   fun(X,_)
```

will succeed when used instead of X in the queries :

var(X)
nonvar(X)
atom(X)
atomic(X)
number(X)
integer(X)
float(X)
compound(X)
lst(X)

Note: You do not have to type anything as a program, all of these have to be evaluated as queries. Open the **Query** *dialog from the* **Eval** *menu.*

2. Using the definition from the notes :

```
buy( Person, Thing) :-
   needs( Person, Thing),
   price( Thing, Price),
   can_pay( Person, Price).
```

and a simple database :

price(cornflakes,2).
price(frosties,3).
price(branflakes,2).
price(chocolate_cake,1).

etc .. add a few of your own.

and needs(antonia, frosties), needs(mark, branflakes).
and can_pay(antonia,4).
can_pay(mark,2).

Now **compile** *your program from within the MacProlog environment. The compile option can be found under the* **Eval** *menu.*

ask *queries* from the MacProlog32 of who can buy what. Also change queries to inquire what things mark can buy. Such as :

buy(mark,frosties)
buy(mark,_)

3. Type in the following program :

likes(ellen,tennis).
likes(tom,swimming).
likes(tom,football).
likes(tom,rugby).
likes(eric,swimming).

Try out the following *goals* :

Note : Click on the **Next** *button in the* **Query** *dialog to get more solutions.*

likes(ellen,tennis)
likes(Who,swimming)
likes(eric,What)

Now add the following rule :

likes(eric,X) :- likes(ellen,X).
Try out the goals :

likes(eric,Sport)
likes(Who,Sport)

Now add the following facts :

likes(john,football).
likes(john,boxing).
likes(john,swimming).

Try out the goal,

likes(tom,X), likes(john,X).

4. See if you can *Save* and then *Print* your program.

Omer Rana
January 1997.