CS3510: Artificial Intelligence

Assignment 1 (10%)*: Prolog — due Friday, 8 March, 2002, 12.00 noon

Peter Lucas, University of Aberdeen

1. Give the Prolog definition for the predicate `replace_n(N, X, Y, L1, L2)`, which succeeds when `L2` is a copy of `L1` into which `N` occurrences of `X` have been replaced by `Y` starting at the beginning of `L1`. (Assume that `N` is always given.)

   Use your program to find the solutions to the following three queries:

   ```
   ?- replace_n(2, a, b, [a, a, b, a, c], L).

   ?- replace_n(3, X, Y, [a, b, b, a, b, b], [a, c, c, a, c, b]).

   ?- replace_n(3, X, g(e), [a, b, b, a, b, a, b], L).
   ```

2. Define a Prolog predicate `eval(Expr, Value)` that given an arithmetic expression `Expr` in prefix form with binary operators `add` and `mult`, standing for addition and multiplication, respectively, returns the value `Value` of that arithmetic expression. For example, the value of the arithmetic expression `mult(2, add(mult(3, 2), add(1, 9)))` is equal to `32`. Take into account that adding 0 to an expression does not change its value; similarly, take into account that multiplying an expression by 1 does not change its value.

   Use your program to find the solutions to the following two queries:

   ```
   ?- eval(mult(3, add(mult(4, 2), add(1, 9))), Value).

   ?- eval(add(6, mult(7, add(2, 4))), Value).
   ```

3. Causal medical knowledge can be described easily in Prolog using a tertiary relation of the form `causes(Cause, Effect, Comment_list)`, where `Cause` represents the cause, `Effect` one of the effects of the given cause, and `Comment_list` a list of remarks, explaining the relationship between the given cause and its effect in more detail.

   The following causal medical knowledge is now becoming more and more relevant given the decline in MMR (Measles, Mumps, Rubella) immunisation among children in Great Britain, as parents need again be aware of the signs and symptoms of these diseases as was also true more than 20 years ago:

*This exercise is part of the formal assessment of the course, and the work done must be your own. You should look at the section on Cheating snd Plagiarism in your student handbook.

```
causes(mumps, 'enlarged salivary glands',
       ['in particular one or both of the parotid glands are enlarged
         and painful']).
causes(measles, fever, ['the fever can be quite high']).
causes(measles, 'Koplik spots',
       ['these are small white spots in the mouth',
        'typically they start bleeding when rubbed']).
causes(measles, conjunctivitis,
       ['this is associated with watery eyes',
        'other ocular signs may also occur']).
causes(rubella, rash, ['the rash starts on the face, and then spreads to
       the trunk and arms']).
causes(rubella, fever, [none]).
causes(fever, 'bodytemp > 38', [none]).
causes(mumps, orchitis, ['this is inflammation of the testes']).
causes(orchitis, 'testicular shrinkage',
       ['this occurs in about 50% of male patients',
        'it is a potential cause of sterility']).
```

Write a Prolog program which includes among others the predicate `effects_of(Cause)`, which when called prints all possible effects, including comments, of the given cause, e.g.

```
?- effects_of(measles).
measles causes fever.
comments: the fever can be quite high.
fever causes bodytemp > 38.
comments: none.
measles causes Koplik spots.
comments: these are small white spots in the mouth,
          typically they start bleeding when rubbed.
measles causes conjunctivitis.
comments: this is associated with watery eyes,
          other ocular signs may also occur.
```

Your program's output should conform *exactly* to the format above, including indentation, newlines and periods at the end of sentences.

Hand in two copies of your code including the results of running the code, as follows. Use the command `script` under Unix (see manual page 'script'), and then run the following commands (assuming all your programs are stored in one file called `progs.pl`):

```
unix% script
unix% cat progs.pl
unix% pl
?- [progs].

?- replace_n(2, a, b, [a, a, b, a, c], L).
```

```
?- replace_n(3, X, Y, [a, b, b, a, b, b], [a, c, c, a, c, b]).

?- replace_n(3, X, g(e), [a, b, b, a, b, a, b], L).

?- eval(mult(3, add(mult(4, 2), add(1, 9))), Value).

?- eval(add(6, mult(7, add(2, 4))), Value).

?- effects_of(measles).

?- effects_of(mumps).

?- effects_of(rubella).
unix%
^D
unix% enscript -o -typescript | psnup -2 | lpr -Plj4_205
```