Programming Languages: Homework #8

Due on May 7, 2019 at 9:00am

Erin Keith Section 101

Michael DesRoches

Problem 1

Solution

Problem 2

```
(14pts) Write the rules for a predicate tally(E,L,N), which succeeds if N is the
number of occurrences of element E in list L. The following query shows an
example of using this predicate:

?-tally(3, [1,2,3,1,2,3],N).
N = 2
```

Solution

```
tally(_,[],0).
tally(E,[E|L],N) :- tally(E,L,M), N is M+1.
tally(E,[F|L],N) :- not(E=F), tally(E,L,N).
```

Problem 3

```
(15pts) Define predicates and/2, or/2, nand/2, nor/2, xor/2, and equ/2 (for
logical equivalence) which succeed or fail according to the result of their
respective operations; e.g.and(A,B) will succeed, if and only if both A and B
succeed. Note that A and B can be Prolog goals (not only the constants true
and fail).

?-and(true, true).
true
```

Solution

```
and(A,B) :- call(A), call(B).

or(A,_) :- A, !.
or(_,B) :- B.

nand(A,B) :- not(and(A,B)).
nor(A,B) :- not(or(A,B)).
equ(A,B) :- or(and(A,B), and(not(A),not(B))).
xor(A,B) :- not(equ(A,B)).
```

Problem 4

```
(15pts) Write the rules for a predicate gcd(X,Y,G), whichdeterminesthe greatest
common divisor of two positive integer numbers. Use Euclid's algorithm:
https://www.khanacademy.org/computing/computer-science/cryptography/
modarithmetic/a/the-euclidean-algorithm?-gcd

(36, 63, G).
G = 9
```

Solution

```
gcd(0,X,X) :- X > 0, !.
gcd(X,Y,G) :- X >= Y, X1 is X-Y, gcd(X1,Y,G).
gcd(X,Y,G) :- X < Y, X1 is Y-X, gcd(X1,X,G).
```

Problem 5

```
(Extra Credit −10 pts) Write the rules for a predicate flatten(A,B), which
succeeds if A is a list (possibly containing sublists), and B is a list
containing all elements in Aand its sublists, but all at the same level. The
following query shows an example of using this predicate:

?-flatten([1, [2, [3, 4]], 5], L).
L = [1, 2, 3, 4, 5]
```

Solution

```
flatten(A,B) :- flatten(A,[],B).
flatten(Var, T, [Var|T]) :- var(Var), !.
flatten([],T,T) :- !.
flatten([H|T],TailList,List) :- !, flatten(H,FlatTail,List), flatten(T,TailList,FlatTail).
flatten(NonList,T,[NonList|T]).
```