02156 Exercises-08

Jørgen Villadsen

2021

Exercise 1

Consider the following definition of a particular directed graph:

```
edge(a,c). edge(a,b). edge(b,a).
```

Explain the results of the following 5 queries without executing them.

```
?- findall(X,edge(X,Y),L).

?- findall(Y,edge(X,Y),L).

?- bagof(X,edge(X,Y),L).

?- bagof(Y,edge(X,Y),L).

?- setof(X,edge(X,Y),L).

?- setof(Y,edge(X,Y),L).
```

Exercise 2

Consider the formula:

$$\forall x p(x) \rightarrow \forall y p(y)$$

The file `qed.pl` available on CampusNet (Program files folder in the top folder) also contains a program `test` that checks a proof in the Hilbert system $\mathcal{H}$:

```
?- test([
     deduce([], all(X, p(X)) => p(a)),
     deduce([], all(Y, all(X, p(X)) => p(Y))),
     deduce([], (all(Y, all(X, p(X)) => p(Y))) =>
                (all(X, p(X)) => all(Y, p(Y)))),
     deduce([], all(X, p(X)) => all(Y, p(Y)))
   ]).
```

Printout:

```
  1. |- (Ax1p(x1) => p(a))            Axiom 4
  2. |- Ax1(Ax2p(x2) => p(x1))        Gen 1
  3. |- (Ax1(Ax2p(x2) => p(x1)) =>
        (Ax2p(x2) => Ax1p(x1)))       Axiom 5
  4. |- (Ax1p(x1) => Ax2p(x2))        MP 3,2

Yes
```

Use `test` to check a proof of the following formula:

$$\forall x(p(x) \rightarrow q(x)) \rightarrow (\forall x p(x) \rightarrow \forall x q(x))$$

Hint: Theorem 8.16 is a good starting point.

Exercise 3

Provide a proof in the Gentzen system $\mathcal{G}$ and in the Hilbert system $\mathcal{H}$ of the following formula:

$$\forall x(p(x) \to q(x)) \to (\exists x p(x) \to \exists x q(x))$$

Hint: Consider using the C-rule and Theorem 8.14 for the proof in the Hilbert system $\mathcal{H}$.

Exercise 4

Consider a program `sublist(?List1,?List2)` that succeeds iff `List1` is a sublist of `List2` (the empty list is a sublist of any list).

```
sublist(A,B) :- append(_,C,B), append(A,_,C).

?- sublist(X,[1,2,3]).

X = [] ;

X = [1] ;

X = [1, 2] ;

X = [1, 2, 3] ;

X = [] ;

X = [2] ;

X = [2, 3] ;

X = [] ;

X = [3] ;

X = [] ;

No
```

Consider the following 4 programs:

```
sublist1(A,B) :- append(C,_,B), append(_,A,C).

sublist2([],_).
sublist2(A,B) :- A = [_|_], append(_,C,B), append(A,_,C).

sublist3(A,B) :- append(A,_,B).
sublist3(A,[_|B]) :- sublist3(A,B).

sublist4([],[]).
sublist4(A,[_|B]) :- sublist4(A,B).
sublist4([X|A],[X|B]) :- sublist4(A,B).
```

Compare each of these programs with the given `sublist` result.

Note that SWI-Prolog has a `sublist/3` predicate but the purpose of this predicate is different.