#2
EXAMPLES OF
AUTOLOG QUANTIFIER
METALOGIC
1/19/2019

§1 ∀

Consider a FOL problem having coherent first-order logic form:

```
∀(X)(p(X)).
∀(X)(p(X)→q(X)).
--------------------
∀(X)(q(X))?
```

An autolog problem translation might look like this

```
dom(X) => p(X).
p(X) => q(X).
true => dom(a).
q(a) => goal.
```

This translation supports a tree proof

```
true
  |
dom(a)
  |
 p(a)
  |
 q(a)
  |
 goal
```

solving the problem using a coherent "lemma".
The lemma supplies an arbitrary 'a' as a witness
and derives q(a). We call the translation a lemma
because ∀ is not explicitly represent in the formulation.

Such proofs argue using nonempty domains.
In FOL, the original problem would hold for empty domains.

A similar formulation of the problem using types might be

```
∀(X:t)(p(X)).
∀(X:t)(p(X)→q(X)).
--------------------
∀(X:t)(q(X)).
```

and an autolog unfolding ...

```
X:t => p(X).
X:t, p(X) => q(X).
true => a:t.
q(a) => goal.
```

having a similar proof.

Is it possible to formulate ∀ more directly using autolog?

Consider the following version

```
true => ∀(X:t)(p(X)).        //1
X:t, p(X) => q(X).           //2
∀(X:t)(q(X)) => goal.        //3
```

along with the following metalogical axioms

```
    ∀(X:T)(P(X)) => @T:T, P(@T).  //4
    P(@T) => ∀(X:T)(P(X)).        //5
```

In particular, note the change of variable in the first metalogic axiom //4. The reason for this will be illustrated in the following proof tree.

```
     true
       |                         //1
     ∀(x:t)(p(x))
       |                         //4
      @t:t
       |                         //4
     p(@t)
       |                         //2
     q(@t)
       |                         //5
     ∀(y:t)(q(y))
       |                         //3
     goal
```

@ is a built-in autolog unary operator.
@T represents an arbitrary object of type T.

The problem formulation as a lemma is obviously simpler than the metalogic formulation.

EXERCISE 1. In the previous autolog formulation use

```
true => ∀(X:t)(p(X)→q(X)).   //2
```

and add the metalogic axiom

```
∀(X:T, P(X)→Q(X)), Z:T, P(Z) => Q(Z).  //6
```

(a) Construct an autolog proof tree for this version.
(b) Now replace //6 by two metalogic axioms

```
∀(X:T)(P(X)→Q(X)), Z:T => P(Z)→Q(Z).  //6
P(Z)→Q(Z), P(Z) => Q(Z).              //7
```

and construct an autolog proof tree for this version.
Try to solve exercise before looking at following answers.
The exercise illustrates that metalogic may have many and various formulations.

---

§1(a) solution for Exercise 1(a)

```
true => ∀(X:t)(p(X)).                  //1
true => ∀(X:t)(p(X)→q(X)).             //2
∀(X:t)(q(X)) => goal.                  //3
∀(X:T)(P(X)) => @T:T, P(@T).           //4
P(@T) => ∀(X:T)(P(X)).                 //5
∀(X:T, P(X)→Q(X)), Z:T, P(Z) => Q(Z).  //6
```

```
     true
       |                                //1
  ∀(x:t)(p(x))
       |                                //4
      @t:t
       |                                //4
     p(@t)
       |                                //2
 ∀(y:t)(p(y)→q(y))
       |                                //6
     q(@t)
       |                                //5
  ∀(y:t)(q(y))
       |                                //3
     goal

   x and y are new constants
   injected by the prover
```

----

§1(b) solution for Exercise 1(b)

```
true => ∀(X:t)(p(X)).                       //1
true => ∀(X:t)(p(X)→q(X)).                  //2
∀(X:t)(q(X)) => goal.                       //3
∀(X:T)(P(X)) => @T:T, P(@T).                //4
P(@T) => ∀(X:T)(P(X)).                      //5
∀(X:T)(P(X)→Q(X)), Z:T => P(Z)→Q(Z).        //6
P(Z)→Q(Z), P(Z) => Q(Z).                    //7
```

```
          true
            |                          //1
      ∀(x:t)(p(x))
            |                          //4
           @t:t
            |                          //4
          p(@t)
            |                          //2
    ∀(y:t)(p(y)→q(y))
            |                          //6
      p(@t)->q(@t)
            |                          //7
          q(@t)
            |                          //3
       ∀(z:T)(q(z))
            |                          //3
          goal

     x, y and z are new constants
     injected by the prover
```
==

§2 ∃

Consider a FOL problem having coherent first-order logic form:

```
∃(X:t)(p(X)).
∀(X:t)(p(X)→q(X)).
--------------------
∃(X:t)(q(X))?
```

Notice that we are considing a problem with a cogent relationship to the problem we started with in §1.

A simple coherent unfolding would look something like this

```
true =>  X:t, p(X).
X:t, p(X) => q(X).
A:t, q(A) => goal.
```

which has a simple proof

```
true
  |
 a:t
  |
 p(a)
  |
 q(a)
  |
 goal
```

Here is a autolog metalogic formulation of the problem

```
true => ∃(X:t)(p(X)).          //1
true => ∀(X:t)(p(X)→q(X)).     //2
∃(X:t)(p(X)) => goal.          //3
```

EXERCISE 2. Add metalogic axioms regarding ∃ to our theory that will yield a proof for this problem. Try this on your own before looking at the following solution.

We need an instantiation rule for asserted existentials, and an evidentiary rule for existentials.

```
     ∃(X:T)(P(X)) => Z:t, P(Z).   //4   ∃  assertion
     X:t, P(X) => ∃(X:t)(P(X)).   //5   ∃  evidence
     ∀(X:T, P(X)→Q(X)), Z:T, P(Z) => Q(Z). // 6
```

Here is a proof tree for the metalogic formulation

```
          true
            |                //1
      ∃(a:t)(p(a))
            |                //4
           a:t
            |                //4
          p(a)
            |                //2
    ∀(b:t)(p(b)→q(b))
            |                //6
           q(a)
            |                //5
       ∃(a:t)(p(a))
            |                //3
           goal
```

A subtle point: What happens when we use the meta axiom

```
     Z:t, P(Z) => ∃(X:t)(P(X)).
```

for meta axiom 5?

==

§3   ∀∃

We explore expansions for ∀∃ metalogic forms

```
     ∀(X:t)(∃(Y:s)(p(X,Y))).  //  assertion
     ∀(X:t)(∃(Y:s)(P(X,Y))?   //  goal -- to prove
```

A simple lemma formulation, without metalogic expansions might be

```
true => @t:t          // arbitrary element of type t
X:t => Y:s, p(X,Y).  // assert ∀(X:t)(∃(Y:s)(p(X,Y)))
true => y:s.          // assume y of type s
Y:s, p(@t,Y) => goal. // ∀(X:t)(∃(Y:s)(p(X,Y)))?
```

which has a simple proof tree

```
   true
     |
   @t:t
     |
    y:s
     |
  p(@t,y)
     |
   goal
```

EXERCISE 3. Formulate the ∀∃ problem using metalogic quantifier translations, as in previous sections, and provide a relevant proof tree for your formulation of the problem.

---

§4 ∀∃ Skolem function metalogic

Let us rework §3 using a Skolem functions in the ∀∃ assertion.

```
∀(X:t)(p(X,sk(X))).      // assertion
```

```
∀(X:t)(∃(Y:s)(P(X,Y))?   // goal -- to prove
```

Here, sk is a new Skolem function which produces witnesses sk(X) for the 2nd argument of p(X,-), for an arbitrary X:t

The lemma formulation might be as follows

```
true => @t:t            // arbitrary element of type t
true => sk:t→s.         // type signature for sk
sk:T→S, X:T=> sk(X):S.  // type transfer for function
X:t => p(X,sk(X)).      // assert ∀(X:t)(p(X,sk(X))
Y:s, p(@t,Y) => goal.   // ∀(X:t)(∃(Y:s)(p(X,Y)))?
```

and here is a proof tree

```
    true
      |
    sk:t→s
      |
    @t:t
      |
  sk(@t):s
      |
     y:s
      |
  p(@t,sk(@t))
      |
    goal
```

EXERCISE 4. Same as for exercise 3, but use the Skolem function version of the problem.

Notice that a Skolem functions and universal arguments play sort of a dual role in our expansion of ∃ and ∀ expressions.

In coherent form FOL, any problem that proves with the use of Skolem functions is provable without the use of Skolem functions.

It is conjectured here that any autolog ∀∃ problem that proves with the use of Skolem functions has a proof without using Skolem functions. For this to be the case for autolog, it is required that axioms for the problem include appropriate equality results for skolem function applications, such as

```
sk:T→S, X:T, Y:T, X=Y => sk(X)=sk(Y).
```

as well as sufficient other axioms for reasoning with equalities.