The Theory of Insults

P = George’s mother is ugly.

Q = George’s mother is more stupid than Harry.

R = Harry’s mother is ugly.

S = Harry’s mother is more stupid than George.

Let (Um x) = x’s mother is ugly.

Then P = (Um George) and R = (Um Harry).

Let (MS x y) = x’s mother is more stupid than y.

So Q = (MS George Harry),
S = (MS Harry George).

```
  Predicate symbols and Function Symbols

    (ugly x) = x is ugly.

    (mother x) = the mother of x


 P = (ugly (mother George))

 Q = (ugly (mother Harry))

 T = (nice (mother George))


    (more x y) = x is more than y.

    (stupidity x) = the stupidity of x


   (and (ugly (mother George))
        (more (stupidity (mother George))
              (stupidity Harry)))

   (and (ugly (mother Harry))
        (more (stupidity (mother Harry))
              (stupidity George)))

   (nice (mother George))
```

Variables and quantifiers

There is no one stupider than George.

```
(not
 (Exists (x)
   (more (stupidity x)
         (stupidity George))))
```

Anyone stupider than Harry is Ugly.

```
(All (x)
   (implies (more (stupidity x)
                  (stupidity Harry))
            (ugly x)))
```

Why do we believe this argument?

```
(All (x) (implies (man x) (mortal x)))
(man Socrates)
-------------------------------------
(mortal Socrates)
```

Method of interpretation!

Domain

Predicate symbols map to predicates

Function symbols map to functions

Individual Constant symbols map to distinguished members of the domain.

If P is true in an interpretaion we say that the interpretation is a model of P.

An argument is valid if there is no interpretation in which the consequent is false and the antecedents are true.

```
      (All (x) (Exists (y) P))
      ------------------------
      (Exists (y) (All (x) P))

 This is not valid:

   Let the domain be the integers.
   Let P be (<= x y).

   (All (x) (Exists (y) (<= x y))) is
   true, because for any integer you
   choose there is always a bigger one.
   So we have a model for the antecedent.
   In the model the consequent
   (Exists (y) (All (x) (<= x y)))
   is false, because there is no greatest
   integer.
```

```
      (Exists (y) (All (x) P))
      ------------------------
      (All (x) (Exists (y) P))
```

This is a valid rule: Suppose the rule were invalid, then there is a model of the antecedent in which the consequent is false.

If the consequent is false there is an x0 for which there is no y that makes P true. But the antecedent claims that there is a y0 for which P is true, independent of the choice of x. This is a contradiction, so the rule is valid.

Remember

```
1.                        Premise        {1}
    All y [(greek y) --> (human y)]

2.                        Premise        {2}
    All x [(human x) --> (mortal x)]

3.                        Premise        {3}
    (greek *G)

4.                        (US 1 *G y)    {1}
    (greek *G) --> (human *G)

5.                         (MP 4 3)    {1 3}
    (human *G)

6.                         (US 2 *G x)   {2}
    (human *G) --> (mortal *G)

7.                         (MP 6 5)  {1 2 3}
    (mortal *G)

8.                         (CP 7 3)    {1 2}
    (greek *G) --> (mortal *G)

9.                         (UG 8 z *G) {1 2}
    All z [(greek z) --> (mortal z)]
```

Rule 15. Universal Specification

Restriction: In this rule t may be a term composed of constants or arbitrary individuals.

```
n     All x P         ---          d
-------------------------------------
m     S[t;x;P]    (US n t x)       d
```

Rule 16. Universal Generalization

Restriction: Here t must be some arbitrary individual not occurring in any line of d, and x may not appear in P.

```
n     P                  ---       d
-------------------------------------
m    All x S[x;t;P]   (UG n x t)  d
```

```
BAD!

10. All w (mortal w)  (UG 7 w *G) {1 2 3}

11. All w (human w)     (UG 5 w *G) {1 3}

12. All w (greek w)       (UG 3 w *G) {3}


GOOD!

13.                             (UG 6 z *G) {2}
    All z [(human z) --> (mortal z)]
```

```
Rule 17. Existential Specification

  Restriction: ???

n     Exists x P     ---          d
--------------------------------------
m     S[t;x;P]    (ES n t x)      d


Rule 18. Existential Generalization

  Restriction: Here t is a term which
  contains no variables, and x may not
  appear in P.

n     P                 ---        d
--------------------------------------
m    Exists x S[x;t;P] (EG n x t)  d
```

Consider the following fallacious derivation (dependencies are all {1}).

```
1. (All (x) (Exists (y) (< x y))) Premise
2. (Exists (y) (< *A y))      (US 1 *A x)
3. (< *A a)                    (ES 2 a y)
4. (All (x) (< x a))          (UG 3 x *A)
5. (Exists(y)(All(x)(< x y)))  (EG 4 x a)
```

Where is the bug?

The problem is that a depends on *A.

Patch: Make it explicit: (a *A).

Rule 17. Existential Specification

Restriction: In this rule t is a term composed of a new function symbol applied to the list of the arbitrary individuals occurring in P.

```
n     Exists x P     ---            d
-------------------------------------
m     S[t;x;P]    (ES n t x)        d
```

```
1. (All (x) (Exists (y) (< x y)))  Premise
2. (Exists (y) (< *A y))       (US 1 *A x)
3. (< *A (a *A))           (ES 2 (a *A) y)
4. (All (x) (< x (a x)))       (UG 3 x *A)
5. STUCK.
```