7.6. Insuring generality

7.6.0. Overview

Although the idea of a general argument is not the last addition we will make to the perspective on proofs introduced in 2.2, it is the key idea needed for the derivations of this chapter and the next.

7.6.1. How generality can fail
To be able to generalize about what is said using a specific name, what is argued must not depend on what this name refers to; and there is more than one way that this can fail to be so.

7.6.2. Multiply general arguments
Arguments that establish multiply general conclusions must be general in several dimensions independently.

Glen Helman 01 Aug 2013

7.6.1. How generality can fail

The examples considered so have not placed much emphasis on the choice of the term used in a general argument. In many of them, any term could be used. And, in cases where this is not true (such as the second example of 7.5.5), the need to use care in choosing a term was accidental. The derivations happened to already contain terms that might naturally be chosen; but, if different letters had appeared (or we were less inclined to choose letters from the beginning of the alphabet), the natural first choice would always work. That will no longer be so when we consider conclusions involving multiple generality, so, before considering them, we will look more closely at the requirements for a term to be independent.

The most basic requirement is that we not rely on special assumptions about the term from which we hope to generalize. We cannot conclude Everything is turned on from The amplifier is turned on, so we cannot generalize on the amplifier in the latter sentence if our justification for it is simply having that sentence as a premise. That is, the following derivation clearly must be disallowed

```
             | Ta        (2)
             |  (a)
             | | ●
     2 QED   | | Ta      1
ERROR 1 UG   | ∀x Tx
```

and it is ruled out by the requirement that the term flagging a general argument appear only to the right of the its scope line.

But, of course, that requirement rules out many other derivations, too, and among them are some that involve no logical error. As was noted in 7.5.3, the appearance of a term among the assumptions does not imply a use of special information about it in drawing a given conclusion, and we have ruled any occurrence of a term outside a scope line it flags, whether this occurrence is in an assumption or elsewhere. The chief virtue of the severe restriction is simplicity in its statement, and this simiplicity comes at little cost since, in the derivations we will consider, there will never be a shortage of new terms to use. (In principle, there can never be a shortage in any sort of derivation if we allow new terms to be generated by devices such as the addition of primes or subscripts.)

Even when it is not needed, the use of a new term does make clear just what sort of argument is provided for the many instances of the generalization other than the one from which we generalize. As one example of this, consider the following argument showing that Everything is turned on really does follow if the premise is extended to say The amplifier is turned on and so is everything else.

```
          | Ta ∧ ∀x (¬ x = a → Tx)   2
          |  (b)
  2 Ext   | | Ta                      (6)
  2 Ext   | | ∀x (¬ x = a → Tx)       b:4
          | | | ¬ Tb                  (5), (6)
  4 UI    | | | ¬ b = a → Tb          5
  5 MTT   | | | b = a                 b—a
          | | | ●
  6 Nc=   | | | ⊥
  3 IP    | | Tb                      1
  1 UG    | ∀x Tx
```

This analysis uses a paraphrase of else as other than it that will be discussed in 8.3.1.

The requirement that the term we generalize on does not appear in any assumption is enough to rule out many unwarranted generalizations but it does not exclude them all. To see why, suppose we are arguing from the assumption Everything is like itself. One conclusion we can draw is Wabash is like Wabash and, in doing so, we have certainly used no special assumptions about Wabash. But this conclusion says that Wabash has the property of being like Wabash, and that makes it an instance of the generalization Everything is like Wabash. Nevertheless generalizing to that conclusion is surely unwarranted. Here is what this argument might look like in a derivation.

```
             | ∀x Lxx    b:2
             |  (b)
     2 UI    | | Lbb     (3)
             | | ●
     3 QED   | | Lbb     1
ERROR 1 UG   | ∀x Lxb
```

The problem with this argument is that even though the term Wabash stands in no special relation to the assumptions, it does stand in a special relation to the universal conclusion Everything is like Wabash. In particular, it plays a special role in the predicate that the conclusion claims to be universal.

These considerations lay behind the second requirement for a general argument: if we wish to generalize from an instance $\theta\tau$ to a universal $\forall x\ \theta x$, the term τ should not appear in our conclusion; that is, it should not appear in the predicate θ. Just as in the case of the first requirement, this is more than is strictly necessary: even if a term has occurrences other than those on which we generalize (i.e., has occurrence left behind in the predicate), this fact may not have been exploited in the argument for it, and the argument might have gone through with any other term. And our approach in derivations is stricter still since we require that the term we generalize on appear nowhere after its scope and not merely that it not appear in the immediately following universal. But, as in other cases, the justification for this restrictiveness is that the restriction is both easy to enforce and easy to satisfy.

The final issue affecting generalization concerns cases where the term we generalize on does not itself appear outside the general argument but contains vocabulary which does. Suppose our assumption is Everything has its bad side. We can conclude Wabash has its bad side. But we cannot go on to conclude Wabash has everything, as in the first derivation below (where d: [_'s bad side] and typographical limitations force a boxed rather than a circled flag).

```
             | ∀x Hx(dx)    b:2
             |  [db]
     2 UI    | | Hb(db)     (3)
             | | ●
     3 QED   | | Hb(db)     1
ERROR 1 UG   | ∀y Hby
```

```
             |  [db]
             | | | ∀x Hx(dx)            b:3
     3 UI    | | | Hb(db)               (4)
             | | | ●
     4 QED   | | | Hb(db)               2
     2 CP    | | ∀x Hx(dx) → Hb(db)     1
ERROR 1 UG   | ∀y (∀x Hx(dx) → Hby)
```

Now the instance from which this conclusion would generalize is an instance for the term Wabash's bad side and this term does not appear in either the assumption or the conclusion, so it satisfies both of the requirements we have imposed so far. And the same issue can arise when vocabulary is shared with the conclusion, as in the second derivation, which is an attempt to show

∀y (∀x Hx(dx) → Hby)—i.e., Everything is such that (Wabash has it if everything has its bad side)—to be a tautology by deriving it from no premises at all.

A requirement that the term we generalize on not share vocabulary with sentences outside the scope line would rule out derivations like thse, and it would be more than enough to insure that an argument is general. Indeed, in the case of a compound term, it would be enough to require that the main functor not appear outside the scope line (so, in the examples above, the real problem lies in the occurences of the functor [_ 's bad side] not the occurences of the term Wabash). However, it is easier simply to prohibit generalization on compound terms. Unanalyzed terms that satisfy the first two requirements clearly share no vocabulary with the assumptions or conclusion so, for those terms, the first two requirements are enough.

Glen Helman 01 Aug 2013

7.6.2. Multiply general arguments

Although we could enforce the requirements that the term we generalize on have no connection with assumptions or the conclusion simply by setting aside a special group of letters for general arguments, that would not be enough to handle cases where a conclusion is multiply general. For, to establish such a conclusion, we need more than one general argument, and the terms used in such arguments must be independent of one another.

The derivation below is a simple illustration of this.

	∀x ∀y Rxy	b:3
	ⓐ	
	ⓑ	
3 UI	∀y Rby	a:4
4 UI	Rba	(5)
	●	
5 QED	Rba	2
2 UG	∀y Rya	1
1 UG	∀x ∀y Ryx	

We begin by applying the planning rule to the universal conclusion, introducing a as the term on which we will generalize. When the rule is applied a second time at stage 2, a second new term is introduced, and it must be independent of the first. That is insured by the rule because, since the term a will appear outside the scope line of the second general argument, a new term must be used to flag this new scope line.

The effects of not using independent terms is shown in the following faulty derivation, which attempts to conclude that R holds between every pair of objects from the assumption that it is reflexive.

	∀x Rxx	a:3
	ⓐ	
	ⓐ	
3 UI	Raa	(4)
	●	
4 QED	Raa	2
ERROR 2 UG	∀y Ray	1
1 UG	∀x ∀y Rxy	

Here the error lies in the use of UG planned for at stage 2, for the premise really would entail the conclusion if it entailed ∀y Ray. And it is innermost scope line that violates the requirement that the flagging term not appear outside the part of the derivation marked by the line.

The recognition of multiple generality in the Middle Ages was a real advance beyond Aristotle's theory of syllogisms (in the narrow sense of 7.5.6). The argument shown below is the sort of pattern the medieval logicians were trying to account for. Both the premise and the conclusion assert affirmative generalizations. But the restricting and quantified predicates of the conclusion themselves involve generalization, and it is the relation that the premise establishes between these generalizations that makes the conclusion follow. The theory of syllogisms did not provide the means to analyze predicates, so it was not able to account for the impact of the premise in this sort of example.

All dogs are mammals

Everything that affects all mammals affects all dogs

	∀x (Dx → Mx)	b:5
	ⓐ	
	∀y (My → Fay)	b:7
	ⓑ	
	Db	(6)
5 UI	Db → Mb	6
6 MPP	Mb	(8)
7 UI	Mb → Fab	8
8 MPP	Fab	(9)
	●	
9 QED	Fab	4
4 CP	Db → Fab	3
3 UG	∀z (Dz → Faz)	2
2 CP	∀y (My → Fay) → ∀z (Dz → Faz)	1
1 UG	∀x (∀y (My → Fxy) → ∀z (Dz → Fxz))	

Since the general term thing does not restrict generalizations, the restriction in the conclusion comes solely from the relative clause that affects all mammals, and the whole sentence would be represented using restricted quantifiers as (∀x: x affects all mammals) x affects all dogs The derivation begins at stage 1 with planning for the unrestricted universal conclusion. At stage 2 we plan for the new conditional goal and at stages 3 and 4 for the universal and conditional that represent the claim a affects all dogs; we do this by introducing a new independent term and supplying a supposition that begins exploitation of the two resources in stages 5-8. Notice that the argument for Fab is doubly general—i.e., it falls within the scope of two independent terms.

Glen Helman 01 Aug 2013

7.6.s. Summary

1 There are a number of reasons why it may not be legitimate to generalize from what has been shown for a given term. The argument may rest on assumptions that special to this term. The predicate we would like to assert generally may contain the term. The term, while not itself appearing in an assumption or the result of the generalization, may share vocabulary with one or the other, and the argument may depend on this connection. These possibilities can all be avoided by requiring that the term we generalize on be an unanalyzed term and not appear outside the scope line whose goal we generalize. These requirements are more stringent than necessary on logical grounds, but they are simple to state and cost us little since they can be met simply by introducing a new unanalyzed term in any general argument.

2 While the chance of illegitimate generalization could be avoided in many cases also by using a special set of terms in general arguments, this would not handle cases of multiply general conclusions, where we need to have general arguments in the scope of other general arguments. In this case, the requirements insure that independent terms are independent of one another and represent multiple independent dimensions of generality.

Glen Helman 01 Aug 2013

7.6.x. Exercise questions

1. Use the system of derivations to establish the following. You may use detachment and attachment rules.

a. ∀x ∀y (Rxy → ¬ Ryx) ⊨ ∀x ∀y (¬ x = y → ¬ (Rxy ∧ Ryx))

b. ∀x ∀y (¬ x = y → ¬ (Rxy ∧ Ryx))
∀x ¬ Rxx
―――――――――――
∀x ∀y (Rxy → ¬ Ryx)

c. ∀x ∀y ∀z ((Rxy ∧ Ryz) → Rxz), ∀x ¬ Rxx ⊨ ∀x ∀y (Rxy → ¬ Ryx)

d. Everyone loves everyone who loves anyone
―――――――――――
If anyone loves anyone, then everyone loves everyone

e. ∀x ∀y Rxy, ∀x (∀y Ryx → (Fx → Gx)) ⊨ ∀x (Fx → Gx)

f. Al said everything he remembered
Al is a person who said nothing
―――――――――――
Anyone who remembered nothing forgot everything
Al forgot everything

2. Choose one of each alternative pair of premises (enclosed in square brackets) and one of each alternative pair of words or phrases in the conclusion in such way that the resulting argument is valid. Then analyze the premises and conclusion and construct a derivation to confirm its validity. You may use detachment and attachment rules.

a. Everyone watched every snake
[Every cobra is a snake | Every snake is a reptile]
―――――――――――
Everyone watched every [cobra | reptile]

b. No one watched every snake
[Every cobra is a snake | Every snake is a reptile]
―――――――――――
No one watched every [cobra | reptile]

c. No one watched any snake
[Every cobra is a snake | Every snake is a reptile]
―――――――――――
No one watched any [cobra | reptile]

d. Everyone who likes every snake was pleased
[Every cobra is a snake | Every snake is a reptile]
―――――――――――
Everyone who likes every [cobra | reptile] was pleased

e. Everyone who likes any snake was pleased
[Every cobra is a snake | Every snake is a reptile]
―――――――――――
Everyone who likes any [cobra | reptile] was pleased

For more exercises, use the exercise machine.

Glen Helman 01 Aug 2013

7.6.xa. Exercise answers

a.

	∀x ∀y (Rxy → ¬ Ryx)	a:6
	ⓐ	
	ⓑ	
	¬ a = b	
	Rab ∧ Rba	5
5 Ext	Rab	(8)
5 Ext	Rba	(9)
6 UI	∀y (Ray → ¬ Rya)	b:7
7 UI	Rab → ¬ Rba	8
8 MPP	¬ Rba	(9)
	●	
9 Nc	⊥	4
4 RAA	¬ (Rab ∧ Rba)	3
3 CP	¬ a = b → ¬ (Rab ∧ Rba)	2
2 UG	∀y (¬ a = y → ¬ (Ray ∧ Rya))	1
1 UG	∀x ∀y (¬ x = y → ¬ (Rxy ∧ Ryx))	

b.

	∀x ∀y (¬ x = y → ¬ (Rxy ∧ Ryx))	a:5
	∀x ¬ Rxx	a:9
	ⓐ	
	ⓑ	
	Rab	(7),(10)
	Rba	(7)
5 UI	∀y (¬ a = y → ¬ (Ray ∧ Rya))	b:6
6 UI	¬ a = b → ¬ (Rab ∧ Rba)	8
7 Adj	Rab ∧ Rba	X,(8)
8 MTT	a = b	a—b
9 UI	¬ Raa	(10)
	●	
10 Nc=	⊥	4
4 RAA	¬ Rba	3
3 CP	Rab → ¬ Rba	2
2 UG	∀y (Ray → ¬ Rya)	1
1 UG	∀x ∀y (Rxy → ¬ Ryx)	

c.

	∀x ∀y ∀z ((Rxy ∧ Ryz) → Rxz)	a:5
	∀x ¬ Rxx	a:10
	ⓐ	
	ⓑ	
	Rab	(8)
	Rba	(8)
5 UI	∀y ∀z ((Ray ∧ Ryz) → Raz)	b:6
6 UI	∀z ((Rab ∧ Rbz) → Raz)	a:7
7 UI	(Rab ∧ Rba) → Raa	9
8 Adj	Rab ∧ Rba	X,(9)
9 MPP	Raa	(11)
10 UI	¬ Raa	(11)
	●	
11 Nc	⊥	4
4 RAA	¬ Rba	3
3 CP	Rab → ¬ Rba	2
2 UG	∀y (Ray → ¬ Rya)	1
1 UG	∀x ∀y (Rxy → ¬ Ryx)	

d. (∀x: Px) (∀y: Py) (∀z: Pz ∧ Lzx) Lyz

(∀x:Px) (∀y:Py) (Lxy → (∀z:Pz) (∀w:Pw) Lzw)

	∀x (Px → ∀y (Py → ∀z ((Pz ∧ Lzx) → Lyz)))	b:10, a:17
	ⓐ	
	Pa	(15), (18)
	ⓑ	
	Pb	(11)
	Lab	(15)
	ⓒ	
	Pc	(20)
	ⓓ	
	Pd	(13), (22)
10 UI	Pb → ∀y (Py → ∀z ((Pz ∧ Lzb) → Lyz))	10
11 MPP	∀y (Py → ∀z ((Pz ∧ Lzb) → Lyz))	d:12
12 UI	Pd → ∀z ((Pz ∧ Lzb) → Ldz)	13
13 MPP	∀z ((Pz ∧ Lzb) → Ldz)	a:14
14 UI	(Pa ∧ Lab) → Lda	16
15 Adj	Pa ∧ Lab	X, (16)
16 MPP	Lda	(22)
17 UI	Pa → ∀y (Py → ∀z ((Pz ∧ Lza) → Lyz))	18
18 MPP	∀y (Py → ∀z ((Pz ∧ Lza) → Lyz))	c:19
19 UI	Pc → ∀z ((Pz ∧ Lza) → Lcz)	20
20 MPP	∀z ((Pz ∧ Lza) → Lcz)	d:21
21 UI	(Pd ∧ Lda) → Lcd)	23
22 Adj	Pd ∧ Lda	X, (23)
23 MPP	Lcd	(24)
	●	
24 QED	Lcd	9
9 CP	Pd → Lcd	8
8 UG	∀w (Pw → Lcw)	7
7 CP	Pc → ∀w (Pw → Lcw))	6
6 UG	∀z (Pz → ∀w (Pw → Lzw))	5
5 CP	Lab → ∀z (Pz → ∀w (Pw → Lzw))	4
4 CP	Pb → (Lab → ∀z (Pz → ∀w (Pw → Lzw)))	3
3 UG	∀y (Py → (Lay → ∀z (Pz → ∀w (Pw → Lzw))))	2
2 CP	Pa → ∀y (Py → (Lay → ∀z (Pz → ∀w (Pw → Lzw))))	1
1 UG	∀x (Px → ∀y (Py → (Lxy → ∀z (Pz → ∀w (Pw → Lzw)))))	

It would be easy to get lost in this argument, but the basic structure has just three parts: planning what must be shown (stages 1-9) and then applying the premise twice (stages 10-16 and 17-23) to take us first from Lab to Lda and then from Lda to Lcd. After stage 9, we have Lcd as the goal and Lab among the resources, and we also know that a, b, c, and d are all people. The premise tells us that anyone who loves is loved by everyone. It will then follow from Lab that the predicate [L _ a] is true of everyone, and it will follow from any predication of [Ld _] of a person that Lcd. Since Lda is both [L _ a]d and [Ld _]a, it can link the two applications of the premise.

e.

	∀x ∀y Rxy	b:7
	∀x (∀y Ryx → (Fx → Gx))	a:3
	ⓐ	
	Fa	(10)
3 UI	∀y Rya → (Fa → Ga)	5
	¬ Ga	(11)
	ⓑ	
7 UI	∀y Rby	a:8
8 UI	Rba	(9)
	●	
9 QED	Rba	4
6 UG	∀y Rya	5
	Fa → Ga	10
10 MPP	Ga	(11)
	●	
11 Nc	⊥	5
5 RC	⊥	4
4 IP	Ga	2
2 CP	Fa → Ga	1
1 UG	∀x (Fx → Gx)	

There is not much alternative to the using RC to exploit ∀y Rya → (Fa → Ga). Although ∀y Rya follows from the premise, it is not an instance of it and thus does not come by UI; and, although the resources Fa and ¬ Ga together entail ¬ (Fa → Ga), we have no attachment rule implementing this entailment. So we do not have an opportunity to apply either MPP or MTT.

f. (∀x: Rax) Sax

Pa ∧ ∀x ¬ Sax

(∀x: Px ∧ ∀y ¬ Rxy) ∀z Fxz

∀x Fax

	∀x (Rax → Sax)	c:9
	Pa ∧ ∀x ¬ Sax	1
	∀x ((Px ∧ ∀y ¬ Rxy) → ∀z Fxz)	a:3
1 Ext	Pa	(7)
1 Ext	∀x ¬ Sax	c:10
	ⓑ	
3 UI	(Pa ∧ ∀y ¬ Ray) → ∀z Faz	5
	¬ Fab	(14)
	●	
7 QED	Pa	6
	ⓒ	
9 UI	Rac → Sac	11
10 UI	¬ Sac	(11)
11 MTT	¬ Rac	(12)
	●	
12 QED	¬ Rac	8
8 UG	∀y ¬ Ray	6
6 Cnj	Pa ∧ ∀y ¬ Ray	5
	∀z Faz	b:13
13 UI	Fab	(14)
	●	
14 Nc	⊥	5
5 RC	⊥	4
4 IP	Fab	2
2 UG	∀x Fax	

There were many other approaches that might have been attempted at stage 3. The key to seeing the approach that was taken is thinking through the content of the resources at that point. Since we have Al is a person and Al said nothing (the resources added at stage 1), the first premise should allow us to conclude that Al is a person who remembered nothing. The third premise should thus allow us to reach the goal of showing that Al forgot b. Stage 3 is a first step along these lines but we are not be able to add the resource needed to apply MPP to this conditional, so stages 4 and 5 set out to exploit it to complete a *reductio*.

2. a.

Everyone watched every snake	(∀x: Px) (∀y: Sy) Wxy
Every cobra is a snake	(∀x: Cx) Sx
Everyone watched every cobra	(∀x: Px) (∀y: Cy) Wxy

	∀x (Px → ∀y (Sy → Wxy))	a:5
	∀x (Cx → Sx)	b:7
	ⓐ	
	Pa	(6)
	ⓑ	
	Cb	(8)
5 UI	Pa → ∀y (Sy → Way)	6
6 MPP	∀y (Sy → Way)	b:9
7 UI	Cb → Sb	8
8 MPP	Sb	(10)
9 UI	Sb → Wab	10
10 MPP	Wab	(11)
	●	
11 QED	Wab	4
4 CP	Cb → Wab	3
3 UG	∀y (Cy → Way)	2
2 CP	Pa → ∀y (Cy → Way)	1
1 UG	∀x (Px → ∀y (Cy → Wxy))	

b.

No one watched every snake	(∀x: Px) ¬ (∀y: Sy) Wxy
Every snake is a reptile	(∀x: Sx) Rx
No one watched every reptile	(∀x: Px) ¬ (∀y: Ry) Wxy

```
          │ ∀x (Px → ¬ ∀y (Sy → Wxy))      a:3
          │ ∀x (Sx → Rx)                   b:9
          ├─
          │ ⓐ
          │ │ │ Pa                         (4)
          │ │ ├─
3 UI      │ │ │ Pa → ¬ ∀y (Sy → Way)       4
4 MPP     │ │ │ ¬ ∀y (Sy → Way)            6
          │ │ │
          │ │ │ │ ∀y (Ry → Way)            b:11
          │ │ │ ├─
          │ │ │ │ ⓑ
          │ │ │ │ │ │ Sb                   (10)
          │ │ │ │ │ ├─
9 UI      │ │ │ │ │ │ Sb → Rb              10
10 MPP    │ │ │ │ │ │ Rb                   (12)
11 UI     │ │ │ │ │ │ Rb → Wab             12
12 MPP    │ │ │ │ │ │ Wab                  (13)
          │ │ │ │ │ │ ●
          │ │ │ │ │ ├─
13 QED    │ │ │ │ │ │ Wab                  8
8 CP      │ │ │ │ │ Sb → Wab               7
7 UG      │ │ │ │ ∀y (Sy → Way)            6
6 CR      │ │ │ │ ⊥                        5
5 RAA     │ │ │ ¬ ∀y (Ry → Way)            2
2 CP      │ │ Pa → ¬ ∀y (Ry → Way)         1
1 UG      │ ∀x (Px → ¬ ∀y (Ry → Wxy))
```

c.

No one watched any snake	(∀x: Sx) (∀y: Py) ¬ Wyx
Every cobra is a snake	(∀x: Cx) Sx
No one watched any cobra	(∀x: Cx) (∀y: Py) ¬ Wyx

```
          │ ∀x (Sx → ∀y (Py → ¬ Wyx))      a:5
          │ ∀x (Cx → Sx)                   a:3
          ├─
          │ ⓐ
          │ │ │ Ca                         (4)
          │ │ ├─
3 UI      │ │ │ Ca → Sa                    4
4 MPP     │ │ │ Sa                         (6)
5 UI      │ │ │ Sa → ∀x (Px → ¬ Wxa)       6
6 MPP     │ │ │ ∀x (Px → ¬ Wxa)            (7)
          │ │ │ ●
          │ │ ├─
7 QED     │ │ │ ∀x (Px → ¬ Wxa)            2
2 CP      │ │ Ca → ∀x (Px → ¬ Wxa)         1
1 UG      │ ∀x (Cx → ∀y (Py → ¬ Wyx))
```

The relative simplicity of this derivation is due to the fact that the difference between the first premise and the conclusion is not deeply embedded in their structures.

d.

Everyone who likes every snake was pleased
Every snake is a reptile
Everyone who likes every reptile was pleased

(∀x: Px ∧ (∀y: Sy) Lxy) Dx
(∀x: Sx) Rx
(∀x: Px ∧ (∀y: Ry) Lxy) Dx

```
          │ ∀x ((Px ∧ ∀y (Sy → Lxy)) → Dx)    a:4
          │ ∀x (Sx → Rx)                      b:11
          ├─
          │ ⓐ
          │ │ │ Pa ∧ ∀y (Ry → Lay)            3
          │ │ ├─
3 Ext     │ │ │ Pa                            (7)
3 Ext     │ │ │ ∀y (Ry → Lay)                 b:13
4 UI      │ │ │ (Pa ∧ ∀y (Sy → Lay)) → Da     6
          │ │ │
          │ │ │ │ ¬ Da                        (6)
          │ │ │ ├─
6 MTT     │ │ │ │ ¬ (Pa ∧ ∀y (Sy → Lay))      7
7 MPT     │ │ │ │ ¬ ∀y (Sy → Lay)             8
          │ │ │ │
          │ │ │ │ │ ⓑ
          │ │ │ │ │ │ │ Sb                  (12)
          │ │ │ │ │ │ ├─
11 UI     │ │ │ │ │ │ │ Sb → Rb             12
12 MPP    │ │ │ │ │ │ │ Rb                  (14)
13 UI     │ │ │ │ │ │ │ Rb → Lab            14
14 MPP    │ │ │ │ │ │ │ Lab                 (15)
          │ │ │ │ │ │ │ ●
          │ │ │ │ │ │ ├─
15 QED    │ │ │ │ │ │ │ Lab                 10
10 CP     │ │ │ │ │ │ Sb → Lab              9
9 UG      │ │ │ │ │ ∀y (Sy → Lay)           8
8 CR      │ │ │ │ ⊥                         5
5 IP      │ │ │ Da                            2
2 CP      │ │ (Pa ∧ ∀y (Ry → Lay)) → Da       1
1 UG      │ ∀x ((Px ∧ (∀y: Ry) Lxy) → Dx)
```

e.

Everyone who likes any snake was pleased
Every cobra is a snake
Everyone who likes any cobra was pleased

(∀x: Sx) (∀y: Py ∧ Lyx) Dy
(∀x: Cx) Sx
(∀x: Cx) (∀y: Py ∧ Lyx) Dy

```
          │ ∀x (Sx → ∀y ((Py ∧ Lyx) → Dy))    a:5
          │ ∀x (Cx → Sx)                      a:3
          ├─
          │ ⓐ
          │ │ │ Ca                            (4)
          │ │ ├─
3 UI      │ │ │ Ca → Sa                       4
4 MPP     │ │ │ Sa                            (6)
5 UI      │ │ │ Sa → ∀y ((Py ∧ Lya) → Dy)     6
6 MPP     │ │ │ ∀y ((Py ∧ Lya) → Dy)          (7)
          │ │ │ ●
          │ │ ├─
7 QED     │ │ │ ∀y ((Py ∧ Lya) → Dy)          2
2 CP      │ │ Ca → ∀y ((Py ∧ Lya) → Dy)       1
1 UG      │ ∀x (Cx → ∀y ((Py ∧ Lyx) → Dy))
```

Glen Helman 01 Aug 2013