Endless Loops

Background

In Chapter 8, we introduced quantified statements in addition to the truth functional compounds we had been examining in Chapters 6 and 7. In Chapters 6 and 7, we studied the rules of inference for truth functional compounds, and developed a "proof strategy" that allowed us to prove or refute arguments composed of truth functional compounds. So with the addition of a new class of statements in Chapter 8--quantified statements--we added three new rules of inference ("Reverse Squiggle," "Drop Existential," and "Drop Universal") and made changes to our "proof strategy" to accommodate these new rules.

> (*A complete description of the final proof strategy covering all the derivations up through Chapter 8 can be found online at: http://homepages.wmich.edu/~baldner/strategyrevised.pdf*)

This proof strategy has a remarkable feature: we can *prove* that it always works.

That is, for any argument composed of truth functional compounds and quantified statements (of the sort introduced in Chapter 8), the proof strategy provides instructions that, in a finite number of steps, will *always* either prove the argument valid or invalid. Following these instructions does not require understanding anything about the "meaning" or the "truth" of "falsity" of any of the formulas that occur in a derivation. The rules simply tell you what groups of symbols you may or must write on the next line, given the groups of symbols that appear or don't appear on previous lines. So the validity or invalidity of arguments containing only these kinds of statements is (as it is sometimes called) "*machine computable*." It can be "computed," in a finite number of "computational steps," by a "mechanical" process that requires no intelligence or understanding, but simply an ability to discriminate a limited class of "objects" and to follow rules for arranging and re-arranging them.

This proof strategy, then, always works for any argument composed of truth functional compounds and the kinds of quantified statements introduced in Chapter 8. In Chapter 8 we had not yet introduced *relations*, and so quantified wffs could contain at most one quantifier, e.g.:

> $(x)Px$
> $(\exists x)Px$

(Of course, we could still create truth functional compounds composed of multiple quantified wffs, such as "$((x)Px \supset (\exists x)Px)$". But the quantified wffs contained in them were always of one of these two forms.)

But in Chapter 9, we introduced quantified relational statements, and these can involve two or more quantifiers in a single (truth-functionally atomic) wff., e.g.:

(x)(y)Pxy
(∃x)(y)(∃z)Pxyz
etc.

Since relations can have any (finite) number of terms, truth functionally atomic quantified relational statements can have any (finite) number of initial quantifiers, and these quantifiers can occur in any order (universal-universal; existential-universal-existential; etc.), as the examples illustrate.

With this, things get more interesting. In 1936, Alonzo Church *proved* that there was no "machine computable" strategy for proving the validity or invalidity of arguments containing quantified relational statements. This has many interesting implications for philosophy, mathematics, and computer science. But what it means for us in this class is:

> ***There is no mechanical proof strategy that will always work for refuting arguments that contain quantified relations.***

The Proof Strategy developed in Chapter 8 will work *sometimes,* but in other cases, it will produce refutations that "loop" back upon themselves endlessly, and so *never end.* These are known as "endless loops." The skill you must master for refuting arguments involving quantified relations is knowing when you have entered an endless loop.

With this introduction, we can now actually begin our discussion of "endless loops."

Endless Loops

Endless loops can arrive in refutations involving quantified relational statements where ***there is an existential quantifier within the scope of a universal quantifier.*** You need to understand what this means, as these are the *only* cases where endless loops can arise.

In these cases, you might say, you find an existential quantifier that is "embedded" inside a universally quantified statement. Consequently, you cannot apply the Drop Existential rule to it until you first "uncover" it by applying the Drop Universal rule to the universally quantified statement is is "embedded" or "contained" in.

Consider the following example:

(x)(∃y)Fxy

This statement begins with a universal quantifier followed by an existential quantifier. Our rules for quantified statements tell us three important things here:

1) use a *new* constant when dropping an existential;
2) drop *all* existentials before dropping *any* universals, unless this is impossible.
(*This includes making an assumption, if the rules call for it, before dropping an any universal. So, if there is a universal we could drop but there is still an unbroken complex wff where one side or the other is an existential wff, we must make an assumption and break that wff before dropping the universal. We do this to make sure we drop any existential, even one that may occur only after making a second assumption, before dropping any universal. If you are not clear on this, see Steps 2.3a and 2.3b in the Proof Strategy for Chapter 8.*)
3), before refuting, drop *all* universals with *all* constants already in use.

But in this example, we have no choice but to drop a universal before dropping an existential, as we don't *have* an existential to drop until after we have dropped the initial universal. There is no way to derive the "embedded" existential -- "(∃y)" -- until we first drop the initial universal -- "(x)."

This is the defining feature of refutations that lead to endless loops: there is an existential quantifier that occurs within the scope of a universal quantifier. In these cases, we must drop a universal before dropping an existential. And when we do, will need to use a different constant than the one we used to drop the universal.

(*This situation can arise in a number of cases. Consider, "(x)~(y)Fxy." After we drop the first universal, we are left with a negated universal. We then apply "Reverse Squiggle" and "uncover" an existential quantifier. This can happen in various ways. But the end result is always that there is an existentially quantified wff that we can derive only after dropping a universal.*)

So, suppose we start with our initial example:

(x)(∃y)Fxy

This statement begins with a universal quantifier. Let's drop it with the constant "a," giving us:

(∃y)Fay

We can now drop the existential, but, according to our rules, we must use a new constant, one not already in use. Since we can't use "a," let's use "b," giving us:

Fab

At this point we would continue applying inference rules. If the argument is valid, we will eventually derive a contradiction. But suppose we get no contradiction. Can we refute? According to our Proof Strategy, before refuting we must drop all universals with all existing constants. But we now have a new constant, "b," that we introduced when we dropped the existential (i.e., when we dropped the existential that we derived only after we dropped the universal in our original statement). So, before we could refute, our rules tell us we must drop "(x)(∃y)Fxy" again with this new constant, "b." This gives us:

(∃y)Fby

According to our Chapter 8 rules, we can refute only when the are no more inference rules to apply. But we now have a new existential statement, and so, according to our Chapter 8 Proof Strategy, we must apply the Drop Existential rule. When we drop existentials, we must use a new constant. Since we can't use "a" or "b," we introduce "c," giving us:

Fbc

Can we refute *now*?

But wait! We now have *another* new constant, "c," so we need to drop that first universal *again* with "c." And this will "uncover" *another* new existential. Dropping it will, in turn, produce *another* new constant, requiring us to drop that same universal *another* time, etc., etc., ***ad infinitum***.

Welcome, my friends, to an ***endless loop***.

What do we do now?

As we said above, there simply *is no* strategy that will always work in a finite number of steps. This has been *proven* by math-logics geeks. So, what's a mere logic student to do? And, more to the point, how will your obsessive-compulsive logic teacher grade you on the test?

Some of you have done quite well in the course thus far by doing lots of Logicola exercises. But, in this case, Logicola exercises are not always instructive. With endless loops, Logicola does all the "hard" parts *for* you.

You may or may not have noticed, but when you have entered an endless loop, Logicola *tells* you this on the bottom of the screen. It says:

> WARNING -- endless loop! Type "REFUTE" to finish the argument.
> Scoring is disabled in case you want to derive a few more steps.

This won't happen on the test! So, on the test, *you* will have to know when you have entered an endless loop, and when you can stop writing new lines and say "ENDLESS LOOP--REFUTE."

So, let me repeat:

> ***On the test, you will be required to know when you can stop deriving new lines and write "Endless Loop." This is something that Logicola did for you.***

When can you do this?

When you have an existential quantifier that occurs within the scope of a universal quantifier, you have no choice but to drop the initial universal in order to derive the "embedded" existential. Dropping that existential will require a constant that has not already been used. Before refuting, you then need to drop the universal (the one that has an existential "embedded" in it) again with the new constant. You must then keep going until you derive the "embedded" existential a second time. At this point, you *know* you are in an endless loop because you have repeated the first step in an infinitely repeating loop.

> ***When you find an existential "embedded" within the scope of a universal quantifier and you have dropped that initial universal quantifier, you can stop deriving new lines when the next line would involve introducing a third new constant.***

OK, that is the rule for stopping and writing "ENDLESS--LOOP." But what about "Refutation Boxes?"

This is something else that Logicola does for you. Creating refutation boxes for endless loops is not automatic. There is no rule I can give you for how to create them. In fact, if you read the footnote on p. 219 of our text, you will note that Logicola does not itself create the refutation boxes for endless loops. Gensler simply has Logicola display a refutation box that Gensler, through his own thinking, has arrived at.

Creating refutation boxes for endless loops is not automatic, and requires some thought and imagination. This is not something we have reviewed much in class, and Logicola always does it for.

> **So, on the test,** ***you will not be required to create refutation boxes for refutations that lead to endless loops. Instead, I will give you extra credit if you can refute the argument.***

But be careful: this *only* applies to refutations that lead to endless loops. For any other refutation (any not leading to an endless loop) you *are* required to create a refutation box.

I will not review here suggestion for how to refute endless loops. You can find suggestions for how to to this on the bottom of p. 221 of our text. Understanding those suggestions is how you can earn extra credit.

So, to repeat the rules for endless loops:

When to stop deriving new lines:

> ***In an endless loop you can stop deriving new lines***
> ***when your next step would be to introduce a third new constant***
> ***after dropping the universal in which this existential is "embedded."***

Refutation Boxes:

> ***You are not required to create refutation boxes for endless loops.***
> *(But this applies only to refutations that lead to endless loops!)* ***Instead,***
> ***I will give extra credit if you can come up with a successful***
> ***refutation box for any refutation that leads to an endless loop.***