9

Natural Deduction for PL

When we put forward the semantics for SL in terms of truth-tables in Chapter 3, this gave us an algorithmic way of checking whether or not an argument is valid: we simply construct the truth-table for each sentence in the argument and check if there's any row in which all of the premises are true and the conclusion is false. We realized, at the end of last chapter, that, in order to determine whether an argument of PL is valid, we can't simply check all of the models to see if there's any model in which all of the premises are true and the conclusion is false—there's simply too many possible models! Accordingly, whereas, in SL, using a deductive system wasn't actually necessary to determine the validity of arguments, in PL, it is. In this chapter, we'll put forward rules for the quantifiers to add to the natural deduction system introduced in Chapter 5.

9.1 Universal Quantifier Rules

As with the other connectives from SL, each quantifier will be given an introduction rule, which lets us conclude a sentence containing that quantifier as its main connective from other things, and an elimination rule, which lets us conclude other things from a sentence containing that quantifier as its main connective. Let's start with the universal quantifier rules. Consider first the elimination rule, known as *universal instantiation*:

Universal Quantifier Elimination Rule:

j	$\forall u(X)$	
	$\vdots$	
k	$X[^n/_u]$	$\forall_E\ j$

This says that if know that $\forall u(X)$ is true, then we can conclude the truth of any substitution instance of X. So, for instance, consider the proof of "Socrates is mortal" from "All humans are mortal" and "Socrates is a human."

1	$\forall x(Hx \rightarrow Mx)$	prem.
2	Hs	prem.
3	$Hs \rightarrow Ms$	$\forall_E$ 1
4	Ms	$\rightarrow_E$ 2, 3

So, since we know that every human is mortal, we can eliminate the quantifier and conclude in line 3 that if Socrates is human, then he is mortal. Then we simply apply modus ponens to conclude that Socrates is mortal.

Consider now what rule we should have for *introducing* a formula of the form $\forall uX$. Consider, for instance, the simple formula $\forall x(Fx)$. This says that *everything* is F. How on earth could we go about proving that? One way, of course, would be to consider literally every thing and prove of each and every thing that it is F. That, however, would take way too long. Indeed, since our domains can contain an infinite number of things, it could take forever. Our strategy, then, will not be to consider *all* of the things, but, rather, to consider an *arbitrary* thing, something about which we have no information at all, and conclude of this thing that it is F. If we're able to conclude of an arbitrary thing that it's F, then the reasoning that we just did will generalize to everything, and so we're able to conclude that everything is F.

To formulate this idea officially as a rule, let us introduce a new kind of rule for flagging a name that functions arbitrarily in the context of a proof:

> **Arbitrary Name Rule:** You can write down $\boxed{n}$ as a line of a proof or as part of the justification of a line of a proof only if n hasn't occurred in any prior line of the proof and doesn't occur in the premises or conclusion.

A line of a proof with a boxed name n can be understood as saying *consider an arbitrary object that we'll call* n, or *let* n *be an arbitrary object*. Unlike the other lines of a proof we have rules for so far, in a line of this sort, you're not actually asserting something. Rather, you're stipulating a use of a name as referring to an arbitrary object. The rule for such a stipulation is that *n* can't be anything that we've already said things about, anything we already have knowledge about by way of our premises, or anything about which we're trying to conclude something. We'll call names in our proof that we've introduced in accordance with this rule "Arbitrary names" or "A-names" for short.

With the boxed name rule, we can now provide the official rule for introducing a universal quantifier that formally codifies the idea informally expressed above. Suppose we want to conclude $\forall u(X)$, where X presumably contains some free occurences of the variable u. To conclude this, we'll try to conclude a *substitution instance* of X with respect to the name n, but with the crucial caveat *that the name n is arbitrary*. If we do this, we can *universally generalize* on the name n, replacing all occurences of it with u and sticking a universal quantifier in front of it. Thus, for instance, if we can conclude Fa for an arbitrary name a, then we can conclude $\forall x(Fx)$. In general, our rule is the following:

Universal Quantifier Introduction Rule:

j	$\boxed{n}$	
	$\vdots$	
k	$X[^{n}/_{u}]$	
l	$\forall u(X)$	$\forall_I$ j-k

Where X contains neither n nor any A-name that depends upon n (i.e. any name m from which there is a path: $\boxed{m} \rightarrow \boxed{n}$*)*

You'll note that this rule comes with a restriction written below. The first part of the restriction simply says that, when we universally generalize on the name n, we need to substitute *all* of the occurrences of n with u. The second part is closely related, but let's ignore it for the moment and get back to it once we've introduced the existential quantifier rules. So, ignoring this restriction, the universal quantifier introduction rule says that, in order to introduce a formula of the form $\forall u(X)$, where u (presumably) occurs freely in X, we first have to introduce an arbitrary name n. Then, we do some reasoning and, in the same subproof, conclude the substitution instance $X[^{n}/_{u}]$ in which all occurences of u

have been substituted with n. Having done that, we can universally generalize on n, concluding the universally quantified statement, $\forall u(X)$. To illustrate how this rule works, consider the following proof of $\forall x((Fx \wedge Gx) \rightarrow Fx)$:

1	$\boxed{a}$	
2	$Fa \wedge Ga$	asm
3	Fa	$\wedge_E$ 2
4	$(Fa \wedge Ga) \rightarrow Fa$	$\rightarrow_I$ 2-3
5	$\forall x((Fx \wedge Gx) \rightarrow Fx)$	$\forall_I$ 1-4

Intuitively, this proof reads as follows. Consider an arbitrary object a. Now suppose that $Fa \wedge Ga$. It follows, by conjunction elimination, that it's true that Fa. Accordingly, by the conditional introduction rule, we can conclude $(Fa \wedge Ga) \rightarrow Fa$. Since, a was arbitrary, we can conclude that this holds of every object. So, we can conclude $\forall x((Fx \wedge Gx) \rightarrow Fx)$.

The reason we have to introduce an arbitrary name before universally generalizing on that name should be obvious. If we didn't have that restriction, we could reason from the claim that "Kermit is green" to the sentence "Everything is green" as follows.

1	Gk	prem.
2	$\forall x(Gx)$	$\forall_I$ 1? *incorrect use of universal quantifier rule!*

Our rules prevent this fallacious reasoning by making it such that, in order to conclude $\forall x(Gx)$, you need to first introduce an arbitrary name and then universally generalize on that name, and it's one of the restrictions on introducing an arbitrary name that this name can't occur in any of the premises.

It's very important that the substitution instance $X[^n/_u]$ with the arbitrary name n from which you infer the universally quantified sentence occurs *in the same subproof* as your boxed name. To see why, consider the following incorrect proof of $\forall x(Fx \rightarrow \forall y(Fy))$.

1	$\boxed{a}$	
2	$\quad Fa$	asm.
3	$\quad \forall y(Fy)$	$\forall_I$ 1-2? *incorrect use of universal quantifier rule!*
4	$Fa \rightarrow \forall y(Fy)$	$\rightarrow_I$ 2-3
5	$\forall x(Fx \rightarrow \forall y(Fy))$	$\forall_I$ 1-4

The conclusion here says that everything is such that if it's F then everything is F. Clearly, we shouldn't' be able to prove this, so something has gone wrong. The problem is that we assumed, on line 2, that a is F, and so *in the context of that subproof*, the name a is no longer occurring arbitrarily. So, we can't universally generalize from something we conclude about a in the context of that subproof given that assumption. Our rules prohibit this by making it such that, if we want to introduce a universal quantifier in the context of a subproof, we have to introduce a new boxed name in the context of that subproof. This ensures that the hypothetical thing that we're reasoning about in the context of that subproof is genuinely arbitrary when we make a universal generalization.

9.2 Existential Quantifier Rules

As we've said, our universal quantifier introduction rule is known as *universal generalization*, whereas our universal quantifier elimination rule is known as *universal instantation*. For the existential quantifier, we rules of the same basic form: an introduction rule of *existential generalization* and an elimination rule of *existential instantiation*. The rule of existential generalization has none of the restrictions that that the rule of universal generalization has. It is simply the following rule:

Existential Introduction:

j	$X[^n/_u]$	
	$\vdots$	
k	$\exists u(X)$	$\exists_I$ j

So, our introduction rule let's us existentially generalize, concluding $\exists u(X)$ from any substitution instance of X, where all free occurences of u have been substituted with any name *n*. For instance, if we want to conclude $\exists x(Fx)$, we can do so by concluding *Fa*, *Fb*, or any other formula of the form *Fn*, for any name *n*. The basic idea is just that if we know of any *particular* thing that it's *F*—be it *a*, *b*, *c*, or what have you—we can conclude that there exists *something* that's *F*. In this case, *n* doesn't need to be arbitrary. Moreover, unlike the case of universal generalization, it's perfectly permissible to leave some occurences of the name on which we're generalizing in the formula *X*. That is, when we existentially generalize, we're not required to substitute *all* occurrences of *n* with *u*—we're permitted to substitute only *some*. So, for instance, if we know that Alice loves herself, not only can we conclude that someone loves themself, but we can also conclude that Alice loves someone and that someone loves Alice. That is, given *Laa*, not only can we conclude $\exists x(Lxx)$, but we can also conclude $\exists x(Lax)$ and $\exists x(Lxa)$.

The elimination rule for the existential quantifier, on the other hand, requires that we instantate with an arbitrary name. The rule is the following:

Existential Elimination Rule:

j	$\exists u(X)$	
	$\vdots$	
k	$X[^n/_u]$	$\exists_E\ j, \boxed{n}$

If X contains another A-name m, then n depends upon m, and an arrow is drawn from $\boxed{n}$ *to* $\boxed{m}$

You'll note that the additional condition under this rule is related to the restriction on the universal quantifier introduction rule, but, once again, let's ignore this condition for the moment. The elimination rule let's us move from $\exists u(X)$ to $X[^n/_u]$ insofar as *n* is a new arbitrary name. The basic thought is that, if we have an existentially quantified sentence such as $\exists x(Fx)$, then we know that there's something that's *F*, and so we can stipulate an arbitrary name *n* to refer to that thing, whatever it is, that is *F*. You can think of application of this rule as saying *let's call the thing that's F "n"* or *let n be the thing that's F*.

This sort of step is very common in the context of mathematical proofs. For instance, if I want to prove that there is an infinite number of primes, I suppose that I have a list of prime numbers $p_1, p_2 \ldots p_3$. Now, I know that, for every set of numbers, there exists a number that is the sum of all of those numbers plus one. So, I can say "Let $a = p_1, p_2 \ldots p_3 + 1$," and I can then go on to show that either a is prime or there is some prime factor of a not in this list. It's crucial that the name "a" here is arbitrary. For instance, I can't say "Let $7474967 = p_1, p_2 \ldots p_3 + 1$," and then go on to show that 7474967 is prime, since the name "7474967" is already in use!

To see how these rules work, consider the simple proof of $\exists x(Fx) \wedge \exists x(Gx)$ from $\exists x(Fx \wedge Gx)$:

1	$\exists x(Fx \wedge Gx)$	prem.
2	$Fa \wedge Ga$	$\exists_E$, 1, $\boxed{a}$
3	Fa	$\wedge_E$ 2
4	$\exists x(Fx)$	$\exists_I$ 3
5	Ga	$\wedge_E$ 2
6	$\exists x(Gx)$	$\exists_I$ 5
7	$\exists x(Fx) \wedge \exists x(Gx)$	$\wedge_I$ 4, 6

Our premise, $\exists x(Fx \wedge Gx)$, tells us that there's something that's both F and G. In line 2, we stipulate a new arbitrary name a for this "something," writing down the substitution instance $Fa \wedge Ga$. In order to make sure that our use of a is genuinely arbitrary here, we need to check that a is not used in any line above line 2, and it's not used in the conclusion we're trying to prove. It's not, so writing $Fa \wedge Ga$ in line 2 is a correct use of the $\exists_E$ rule. It follows by conjunction elimination that Fa, and so, in line 4, we conclude by $\exists_I$ that there's something that's F. Likewise, it follows by conjunction elimination that Ga, and so we conclude in line 6 that there's something that's G. Putting lines 4 and 6 together with conjunction introduction, we conclude $\exists x(Fx) \wedge \exists x(Gx)$: there's something that's F and there's something that's G.

9.3 Some Proofs Using These Rules

Let's now consider a few proofs that combine the universal and the existential quantifier rules. Consider how we prove $\exists x(Fx) \rightarrow \exists x(Gx)$ from $\forall x(Fx \rightarrow Gx)$:

1	$\forall x(Fx \to Gx)$	prem.
2	$\quad \exists x(Fx)$	asm.
3	$\quad Fa$	$\exists_E$ 2, $\boxed{a}$
4	$\quad Fa \to Ga$	$\forall_E$ 1
5	$\quad Ga$	$\to_E$ 3, 4
6	$\quad \exists x(Gx)$	$\exists_I$ 5
7	$\exists x(Fx) \to \exists x(Gx)$	$\to_I$ 2-6

Note the difference between line 3 and line 4. When we eliminate an *existential* quantifier, we have to do so with an arbtrary name, since we only know there's *something* of which X is true. So, on line 3, a has to be new, which we indicate by putting $\boxed{a}$ as part of the justification for that line. When we eliminate a *universal* quantifier, on the other hand, we can use *any* name, since we know that X is true of *everything*. So, on line 4, we can use $\forall_E$ on line 1 to conclude $Fa \to Ga$, even though a is already in use. Note that, if we tried to do things in the other order, introducing $Fa \to Ga$ on line 3, we wouldn't be able to introduce Fa on line 4, since that name would already be in use. In general, when we want to establish a logical relation between an existentially quantified sentence and a universally quantified one, it's a good strategy to existentially eliminate *first*, using a new name, and then universally eliminate *second*, using that same name.

Let's now look at the proofs of the most important equivalences in PL: *negated quantifier equivalences*. These tell us that $\neg\exists u(X)$ is equivalent to $\forall u \neg(X)$, and $\neg\forall u(X)$ is equivalent to $\exists u(\neg X)$. We can prove these facts in this general notation, but, for simplicity, let us just prove the equivalence of $\forall x \neg(Fx)$ and $\neg\exists x(Fx)$. The two proofs go as follows:

1	$\forall x(\neg Fx)$	prem.
2	$\quad \exists x(Fx)$	asm.
3	$\quad Fa$	$\exists_E$, 2, $\boxed{a}$
4	$\quad \neg Fa$	$\forall_E$ 1
5	$\neg\exists x(Fx)$	$\neg_I$ 2-4

1	$\neg\exists x(Fx)$	prem.
2	$\boxed{a}$	
3	$\quad Fa$	asm.
4	$\quad \exists x(Fx)$	$\exists_I$ 3
5	$\quad \neg\exists x(Fx)$	reit. 1
6	$\neg Fa$	$\neg_I$ 3-5
7	$\forall x(\neg Fx)$	$\forall_I$ 2-6

The strategy for these proofs is straightforward. For the proof on the left, we want to prove a negation, so we start by assuming the negated sentence $\exists x(Fx)$, and hoping to derive a contradiction. We then use $\exists_E$ on our assumption to get Fa and then $\forall_E$ on our premise to get $\neg Fa$. Once again, it's important that we use existential elimination *first* here, since the name we use with existential elimination has to be arbitrary, whereas the name we use with universal elimination can already be in use. For the proof on the right, we want to prove a universally quantified sentence, so we start by stipulating an arbitrary name a. We want to prove $\neg Fa$, so we assume Fa, hoping to derive a contradiction. To do this, we use existential introduction to get $\exists x(Fx)$, and then we reiterate our premise to get the contradiction. The proofs of the equivalence of $\neg\forall x(Fx)$ and $\exists x\neg(Fx)$ are left as an exercise for you to do yourself. One is very simple, and one is a bit tricky!

9.4 Dependence Among A-Names

Now that we've gotten a sense of how our quantifier rules work, let us return to consider the restriction on the universal quantifier introduction rule: that, when we conclude some substitution instance of X containing A-name n, which enables us to conclude $\forall u(X)$, this formula X over which our universal quantifier ranges can't contain n nor any A-name that depends upon n. The first part of this restriction is straightforward. The fact that X cannot contain n simply means that, when we universally generalize on n in the substitution instance $X[^n/_u]$, we must replace *all* occurrences of n with u. We've already mentioned this restriction when we introduced the rule, but let's now see why it's important. Consider the attempted proof of the clearly false claim that there is some number that is equal to every number, with the bad step marked with a hash-mark:

1. Every number is equal to itself.
2. Now, let a be an arbitrary number.
3. It follows that a is equal to a.
4. # Since a was arbitrary, every number is equal to a.
5. So, there's some number that is equal to every number.

Clearly, this is bad reasoning, and the problem is clearly with step 4: when you universally generalize on some arbitrary name, you can't keep any occurrences on that name in the universally quantified formula that you conclude. So, the only universally quantified formula you can conclude from line 3 is the one you already have: every number is equal to itself. Formally representing this argument, our proof system blocks the fallacious step 4:

1	$\forall x(Exx)$	prem.	
2	$\boxed{a}$		
3	Eaa	$\forall_E$ 1	
4	$\forall x(Exa)$	$\forall_I$ 2-3?	*incorrect use of universal quantifier rule!*
5	$\exists y(\forall x(Exy))$	$\exists_I$ 4	

If one wants to apply $\forall_I$ to the formula on line 3, to conclude a universally quantified formula over the variable x, one needs to replace *all* occurances of a with x. Thus, the only thing one can conclude is the premise one already has: $\forall x(Exx)$.

Let's now turn to the the second part of the restriction—that beyond just not containing n, X also can't contain any name that *depends upon* n. In the last chapter, we showed that the argument with the premise $\forall x(\exists y(Ryx))$ and the conclusion $\exists y(\forall x(Ryx))$ is invalid by constructing a model in which the premise was true and the conclusion was false. The example we considered as an instance of this argument was a philosophical one, where Ryx was y causes x. But let's now switch up the example to a mathematical one, supposing our domain is real numbers and Ryx is y is the cube root of x. Consider now the following bad line of reasoning, with the bad step marked with a hash-mark:

1. Every number has a cube root.
2. Now, let a be an arbitrary number.
3. It follows that there exists a number that is the cube root of a.
4. Let us call this number b.
5. # Since a is arbitrary, and b is it's cube root, it follows that every number has b is as its cube root.

6. So, there's some number that is the cube root of every number.

The name b in the above "proof" designates an arbitrary number but one whose value *depends upon* the value of a. The notion of dependence here is the same as the notion of dependence between variables that you might have learned in grade school algebra. In an equation such as $b = \sqrt[3]{a}$, b is the *dependent* variable, whereas a is the *independent* variable: the value of b is defined only in relation to a. Accordingly, if we want to universally generalize on the arbitrary name a—moving from a claim about a to a claim not about a *particular* thing but about *every*thing—for the same reason our claim can't contain a, it can't contain b either.

Our proof system generalizes this notion of dependence, and codifies this restriction by making us draw an arrow every time we use existential elimination on a formula containing an A-name. To see this, consider the following attempted proof of $\forall x(\exists y(Ryx))$ from $\exists y(\forall x(Ryx))$, which the problematic step is blocked by the restriction on $\forall_I$ rule:

1	$\forall x(\exists y(Ryx))$	prem.
2	$\boxed{a}$	
3	$\exists y(Rya)$	$\forall_E$ 1
4	Rba	$\exists_E$, 3, $\boxed{b}$ *note:* ***b depends upon a***
5	$\forall x(Rbx)$	$\forall_I$ 2-4? *incorrect use of the universal quantifier rule!*
6	$\exists y(\forall x(Ryx))$	$\exists_I$ 5

When we apply $\exists_E$ on line 3, instantiating with the new A-name b to conclude Rba 4, we have to check to see if this formula contains any other A-name. It contains, the name a, and so we have to draw an arrow from $\boxed{b}$ to $\boxed{a}$. Since Rba contains b, and b depends upon a, we can't universally generalize on a and conclude $\forall x(Rbx)$, as we incorrectly do in line 5. In general, in order to universally generalize on an A-name n in a formula X, we need to to be sure that X contains no A-name on which n depends. That is, there must be no arrow (or path of arrows) going from any A-name X contains to n. So, in the context of this proof, we know that we can't universally generalize on a in the context of the formula *Rba*, since *Rba* contains some occurrences of *b*, and there's an arrow going from *b* to *a*. If we do want to universally generalize, we have to get the *b*'s out of this formula. So, for instance, we could existentially generalize first to conclude $\exists x(Rxa)$, and then universally generalize to conclude $\forall y(\exists x(Rxy))$, a sentence which, of course, is entailed $\forall x(\exists y(Ryx)$ (since it's just the same formula with the variables switched).

To see the contrast between the above bad case in which the $\forall_I$ rule is broken and a good case in which the rule is followed, consider the following correct proof of the opposite direction of entailment:

1	$\exists y(\forall x(Ryx))$	prem.
2	$\boxed{a}$	
3	$\forall x(Rbx)$	$\exists_E$, 1, $\boxed{b}$
4	Rba	$\forall_E$ 3
5	$\exists y(Rya)$	$\exists_I$ 4
6	$\forall x(\exists y(Ryx)$	$\forall_I$ 2-5

Here, when we introduce the A-name *b* in line 3, *b doesn't* depend upon *a*, since *a* is not in existentially quantified formula that is instantiated with *b*. Moreover, *a* doesn't depend upon *b*, since it's only the context of existential elimination that A-names can be introduced as dependent upon other A-names.

To further illustrate these notions, let's consider a more involved proof. Consider the proof of $\forall u(\exists v(\forall s(\exists t(Ruvst))))$ from $\forall x(\exists y(\forall z(\exists w(Rxyzw))))$. Now, we'd normally never want to prove such a thing, but, since we're just swapping out all of the variables with different variables, we know that if the first formula is true, then the second formula must also be true. So, we should be able to prove this in our system, and it will be instructive to do so. Here's the proof, with the arrows indicating the relations of dependence between A-names:

1	$\forall x(\exists y(\forall z(\exists w(Rxyzw))))$	prem.
2	$\boxed{a}$	
3	$\exists y(\forall z(\exists w(Rayzw)))$	$\forall_E$ 1
4	$\forall z(\exists w(Rabzw))$	$\exists_E$ 3, $\boxed{b}$
5	$\boxed{c}$	
6	$\exists w(Rabcw)$	$\forall_E$ 4
7	$Racbd$	$\boxed{d}$
8	$\exists t(Rabct)$	$\exists_I$ 7
9	$\forall s(\exists t(Rabst))$	$\forall_I$ 5-8
10	$\exists v(\forall s(\exists t(Ravst)))$	$\exists_I$ 9
11	$\forall u(\exists v(\forall s(\exists t(Ruvst))))$	$\forall_I$ 2-10

Let's walk through this proof step by step. The conclusion we want to end up with is a universally quantified sentence, and so we start by stipulating a new arbitrary name *a* on line 2, hoping to conclude the substitution instance of the universally quantified formula shown on line 10. We universally instantiate on line 3, and then, on line 4, we existentially instantiate with a new name *b*. Since the formula we're existentially instantiating on contains the name *a*, we draw an arrow from *b* to *a* to show that *b* depends upon *a*. In line 5, we introduce a new name *c*, which doesn't depend on any names. In line 6, we universally instantiate with that name. In line 7, we existentially instantiate with a new name, *d*. Since the formula from which we're existentially instantiating contains the names *a*, *b*, and *c*, it depends upon all of these names. So, we draw an arrow from *d* to *c* and from *d* to *b*. We don't need to draw an arrow directly from *d* to *a* (though doing so wouldn't hurt), since if we show that *d* depends on *b* and that *b* depends on *a*, we've thereby shown that *d* depends on *a*. In line 8, we existentially generalize on *d*. In line 9, we universally generalize on the name *c*. In order to make sure we can do this, we have to check that neither of the names in this formula, *a* or *b*, depend upon *c*. Neither of them do, so our use of $\forall_I$ here is legal. In line 10, we existentially generalize on *b*. Finally, in line 11, we universally generalize on *a*, arriving at our conclusion.

You'll almost never have to prove something with this many nested quantifiers, so don't feel overwhelmed at the prospect of constructing such a proof. However, make sure you understand why the arrows in the above proof must be drawn the way that they are drawn, and how they restrict the inferences we're allowed to make. Note, for instance, that, if we wanted to universally generalize on *a* rather than *c* in line 9, we couldn't since the formula we'd universally generalizing contains *b*, which depends upon *a*.

Historical Note:

The general sort of existential quantifier rules that we're using were put forward by W.V.O. Quine in 1950 and also in a popular textbook by Irving Copi in 1954 (though Copi's original formulation was incorrect and it took several iterations of corrections after 1954 for the rules to be stated correctly). The specific approach to these rules adopted here, thinking of dependence relations between arbitrary names and drawing arrows to indicate these relations, is owed to Kit Fine, who systematically developed this approach to quantification in his 1985 book *Reasoning with Arbitrary Objects*. Fine actually argues in that book that there *really are* arbitrary objects. That is, in addition to particular human beings like Socrates or particular numbers like the number 4, there are such things as arbitrary human beings and arbitrary numbers. That's a bit of contentious metaphysics that we don't need to subscribe to in order to make use of the formal tools Fine develops in spelling out this idea.

9.5 Proofs, Validity, and Invalidity

As with the natural deduction system for SL, which was sound and complete with respect to the semantics of SL, this system is sound and complete with respect to the semantics of PL. Accordingly, as long as you follow the rules for the quantifiers, you'll never be able to prove some conclusion from some set of premises if the argument with those premises and that conclusion isn't valid.

Moreover, you'll be able to prove the conclusion of any valid argument from its premises. It might be tricky, but, if the conclusion does follow from the premises, there's a way to do it. There is, however, one contrast with SL worth reiterating.

In SL, if you're trying to prove some conclusion from some set of premises and you're stuck, you can always construct the truth-table to be sure that the argument really is valid, and so there really is some proof of it to be constructed. In PL, however, if you get stuck trying to prove an conclusion from some set of premises, and you wonder whether the argument really is valid, insofar as the argument is relatively complex, there is no foolproof way to confirm that it is indeed valid other than actually proving it. The best thing to do, in a case where you really seem stuck, is to reason about whether it could possibly be invalid, using where you've gotten stuck to aid you in thinking about how you could try to construct a countermodel. For instance, if you've done a bunch of a proof there's one step from X to Y that you know you need but you can't seem to get, see if you an construct a model in which X is true and Y is false.

Let's take a simple example to illustrate this strategy. Suppose we're trying to prove $\exists x(\neg Px)$ from $\forall x((Px \wedge Rx) \rightarrow Qx)$ and $\exists x(\neg Qx)$. We've started the proof and gotten this far:

1	$\forall x((Px \wedge Rx) \rightarrow Qx)$	prem.
2	$\exists x(\neg Qx)$	prem.
3	$\neg Qa$	$\exists_E$, 2, $\boxed{a}$
4	$(Pa \wedge Ra) \rightarrow Qa$	$\forall_E$, 1
5	$\neg(Pa \wedge Ra)$	MT, 3, 4
6	$\vdots$	??
7	$\neg Pa$	??
8	$\exists x(\neg Px)$	want to conclude by $\exists_I$ from 6

We're stuck, since we don't have any rules that will get us from $\neg(Pa \wedge Ra)$ to $\neg Pa$, given what we have. So, rather than banging our head against the wall to try to get the proof to go through, let's pause and see if we're trying to prove an invalid argument by trying to construct a model in which $\neg(Pa \wedge Ra)$ is true and Pa is false, where, once again a something that's not Q. Of course, this is easy to do:

Domain: {1}

a: 1

Q: {}

P: {1}

R: {}

Checking that this makes the premises of the argument true and the conclusion false, we see that the reason we weren't able to prove the conclusion from the premises is that the argument is invalid!

Finally, one small sidenote. You'll see that I've written "MT" in the above proof for "Modus Tollens," where I'ved inferred $\neg(Pa \wedge Ra)$ from $(Pa \wedge Ra) \rightarrow Qa$ and $\neg Qa$. We learned the trick for officially making this inference in Chapter 6: you need start a subproof with $Pa \wedge Ra$, use conditional elimination (modus ponens) to get Qa, and then reiterate $\neg Qa$ into that subproof to get a contradiction and conclude $\neg(Pa \wedge Ra)$ by negation introduction. In the SL proofs you did in Chapters 5 and 6, I required that you actually do the proofs using only the official rules for the sentential connectives. In this chapter, since our focus in on the rules for the quantifiers, you're free to use the shortcuts for making SL inferences from Chapter 6, as I have here. However, in the excercises I've given you for this chapter, all of the focus is on mastering the quantifier rules, and so you won't actually need to use any of these tricks to make the proofs shorter.

9.6 Wrapping Up

Finally, to conclude, let us consider once more the argument from the Nāgarjūna that we've now represented in PL as follows:

Premise 1: $\forall x \forall y(Cxy \rightarrow (Iyx \vee Eyx))$
Premise 2: $\forall x \forall y(Cxy \rightarrow \neg Iyx)$
Premise 3: $\forall x \forall y(Cxy \rightarrow \neg Eyx)$
Conclusion: $\neg(\exists x \exists y(Cxy))$

The proof of this argument is straightforward (I've compressed some successively applications of quantifier elimination rules into single steps):

1	$\forall x \forall y(Cxy \rightarrow (Iyx \lor Eyx))$	prem.
2	$\forall x \forall y(Cxy \rightarrow \neg Iyx)$	prem.
3	$\forall x \forall y(Cxy \rightarrow \neg Eyx)$	prem.
4	$\vert\ \exists x(\exists y(Cxy))$	asm.
5	$\vert\ Cab$	$\exists_E$, $\boxed{a}$, $\exists_E$, $\boxed{b}$, 4
6	$\vert\ Cab \rightarrow (Iab \lor Eab)$	$\forall_E$, $\forall_E$, 1
7	$\vert\ Cab \rightarrow \neg Iab$	$\forall_E$, $\forall_E$, 2
8	$\vert\ Cab \rightarrow \neg Eab$	$\forall_E$, $\forall_E$, 3
9	$\vert\ Iab \lor Eab$	$\rightarrow_E$ 6
10	$\vert\ \neg Iab$	$\rightarrow_E$ 7
11	$\vert\ Eab$	$\lor_E$ 9, 10
12	$\vert\ \neg Eab$	$\rightarrow_E$ 8
13	$\neg\exists x(\exists y(Cxy))$	$\neg_I$ 4-12

We make the same basic inferences as we did before when we represented this argument in SL, but now our proof system captures the *generality* that the argument is meant to have. We suppose that there exist some two objects that stand in a causal relation, and so we consider a pair of *arbitrary* objects, a and b, which are supposed to stand in a causal relation. We then derive a contradiction: a must be both extrinsic to b and not extrinsic to b. Having concluded this about two arbitrary objects, we can conclude that there does not exist *any* objects that stand in a causal relation.

9.7 Exercises

9.1 Each of the following attempted proofs contain some mistake. Identify at which line the mistake occurs, and say what the mistake is (1pt each).

a) ———

1	Ca	prem.
2	$\exists x(Bx)$	prem.
3	Ba	$\exists_E$ 2, $\boxed{a}$
4	$Ca \wedge Ba$	$\wedge_I$ 1, 3
5	$\exists x(Cx \wedge Bx)$	$\exists_I$ 4

b) ————

1	$\forall x(Lxx)$	prem.
2	$\boxed{a}$	
3	Laa	$\forall_E$ 1
4	$\forall x(Lax)$	$\forall_I$ 2-3

c) ————

1	$\exists x(Px)$	prem.
2	Pa	$\exists_E$ 1, $\boxed{a}$
3	$\forall x(Px)$	$\forall_I$ 2

d) ————

1	$\forall x(\exists y(Fxy \wedge (Gy \rightarrow Gx)))$	prem.
2	$\boxed{a}$	
3	$\exists y(Fay \wedge (Gy \rightarrow Ga))$	$\forall_E$ 1
4	$Fab \wedge (Gb \rightarrow Ga)$	$\exists_E$ 3, $\boxed{b}$
5	$Gb \rightarrow Ga$	$\wedge_E$ 4
6	$\mid$ Gb	asm.
7	$\mid$ Ga	$\rightarrow_E$ 5, 6
8	$\mid$ $\forall z(Gz)$	$\forall_I$ 2-7
9	$Gb \rightarrow \forall z(Gz)$	$\rightarrow_I$ 6-8
10	Fab	$\wedge_E$ 4
11	$Fab \wedge (Gb \rightarrow \forall z(Gz)))$	$\wedge_I$ 9, 10
12	$\exists y(Fay \wedge (Gy \rightarrow \forall z(Gz)))$	$\exists_I$ 11
13	$\forall x(\exists y(Fxy \wedge (Gy \rightarrow \forall z(Gz))))$	$\forall_I$ 2-12

e) ————

1	$\forall x(Fx \wedge Gx)$	prem.
2	$\forall x(Fx \rightarrow \exists y(Lyx))$	prem.
3	$\boxed{a}$	
4	$Fa \rightarrow \exists y(Lya)$	$\forall_E$ 2
5	$Fa \wedge Ga$	$\forall_E$ 1
6	Fa	$\wedge_E$ 5
7	$\exists y(Lya)$	$\rightarrow_E$ 4, 6
8	Lba	$\exists_E$ 7, $\boxed{b}$
9	$\forall x(Lbx)$	$\forall_I$ 3-8
10	$\exists y(\forall x(Lyx))$	$\exists_I$ 9
11	Ga	$\wedge_E$ 5
12	$\forall x(Gx)$	$\forall_I$ 3-11
13	$\forall x(Gx) \wedge \exists y(\forall x(Lyx))$	$\wedge_I$ 10, 12

9.2 Prove the following (2 pts each):

a) $\forall x(Fx \vee Gx) \vdash \exists x(\neg Fx) \rightarrow \exists x(Gx)$

b) $\neg\exists x(Fx \wedge Gx) \vdash \forall x(Fx \rightarrow \neg Gx)$

c) $\exists x(Fx \wedge Gax), \forall y(\exists x(Gxy) \rightarrow \neg Hy) \vdash \exists x(Fx \wedge \neg Hx)$

9.3 Prove the other negated quantifier equivalence. That is, prove the following (2 pts each):

a) $\neg\forall x(Fx) \vdash \exists x(\neg Fx)$

b) $\exists x(\neg Fx) \vdash \neg\forall x(Fx)$

9.4 Prove the following, making sure to draw arrows to indicate the relations of dependence between A-names:

a) $\forall x(\exists y(\forall z(Rxyz))) \vdash \forall w(\exists u(\forall v(Rwuv)))$

b) $\exists x(\forall y(\exists z(Rxyz))) \vdash \exists w(\forall u(\exists v(Rwuv)))$

9.5 The existential quantifier elimination rule we introduced is different than the rule that Gentzen originally introduced, which you'll find most other introductory books (at least, ones that give you a proof system in which each quantifier is given an introduction and an elimination rule). That alternative rule, formulated in our notation, is the following:

Gentzen's rule:

j	$\exists u(X)$	
k	$\quad X[^n/_u]$	$\boxed{n}$
	$\quad \vdots$	
l	$\quad Y$	
m	Y	gentzen, $j, k\text{-}l$

where n doesn't occur in Y

The idea is similar to the proof by cases rule for eliminating disjunctions. If you can conclude Y from a specific substitution instance of X, where the name substituted is arbitrary, then you can conclude Y from the existentially quantified sentence which says that there is some true substitution instance.

Using these rule makes formulating the rules a bit simpler, since, without an existential instantiation rule, we wouldn't need to define the relation of dependence between A-names. However, the trade off is that our proofs end up being much more simple and intuitive whereas, with this rule, a lot of proofs that that should be quite simple become quite tricky and unintuitive.

For this problem, suppose I gave you a system which contained this existential elimination rule instead of ours, and, using only this elimination rule, along with the rest of our rules, prove the following:

a) $\forall x(Fx \to Gx) \vdash \exists x(Fx) \to \exists x(Gx)$ (2pts)

b) *Bonus:* $\forall x \neg(Fx) \vdash \neg\exists x(Fx)$ (+2pts)