Tree Trimming: Four Non-Branching Rules for Priest's *Introduction to Non-Classical Logic*

Marilynn Johnson
Department of Philosophy
The Graduate Center, City University of New York

Abstract

In *An Introduction to Non-Classical Logic: From If to Is*[1] Graham Priest presents branching rules in Free Logic, Variable Domain Modal Logic, and Intuitionist Logic. I propose a simpler, non-branching rule to replace Priest's rule for universal instantiation in Free Logic, a second, slightly modified version of this rule to replace Priest's rule for universal instantiation in Variable Domain Modal Logic, and third and fourth rules, further modifying the second rule, to replace Priest's branching universal and particular instantiation rules in Intuitionist Logic. In each of these logics the proposed rule leads to tableaux with fewer branches. In Intuitionist logic, the proposed rules allow for the resolution of a particular problem Priest is grappling with throughout the chapter. In this paper, I demonstrate that the proposed rules can greatly simplify tableaux and argue that they should be used in place of the rules given by Priest.

1 Introduction

In Chapter 13 of *An Introduction to Non-Classical Logic: From If to Is* Graham Priest (2008) introduces the following rule for universal instantiation in Free Logic:

$$\begin{tikzcd}[column sep=tiny] & \forall x A \arrow[dl] \arrow[dr] & \\ \neg\mathfrak{E}a & & A_x(a) \end{tikzcd}$$

In this rule, a is any constant already on the branch; if there is no constant already on the branch, one is introduced. Priest slightly adapts this rule to suit Variable Domain Modal Logic in Chapter 15 and to suit Intuitionist Logic in Chapter 20.

[1] As the aim of this paper is to present improved quantifier rules for various systems of logic presented in Priest (2008), I will adopt much of his terminology, notation, and formatting. For straightforward comparison with the text, citations will follow Priest's formatting of (x, y, z) with x denoting the chapter, y the section, and the z the subsection, rather than providing page numbers.

I propose a simpler rule to replace Priest's rule for universal instantiation in Free Logic, a second, slightly modified version of this rule to replace Priest's rule for universal instantiation in Variable Domain Modal Logic, and third and fourth rules, further modifying the second rule, to replace Priest's branching universal and particular instantiation rules in Intuitionist Logic.

In essence, each of these rules replaces a branching rule that indicates a formula of the form $\forall xA$ is true when either $\neg\mathfrak{E}a$ is true or A is true of any other object with a non-branching rule that is applied only when there is an object said to exist on the branch. This idea is applied across the instances of similarly branching rules in Priest's chapters on Free Logic, Variable Domain Modal Logic, and Intuitionist Logic. In each of these logics the proposed rule leads to simpler tableaux. In Intuitionist logic, the proposed rules allow for the resolution of a particular problem Priest grapples with throughout the chapter: a desire that resulting tableaux be both perspicuous and not unwieldy.

In each of the following three sections I begin by introducing a system of logic, including presenting Priest's universal and particular instantiation rules for the system. I then complete a tableau that closes with Priest's rule. After completion of the closed tableau, I explain the rule I propose and complete the previous example with the proposed rule replacing Priest's rule. I will follow this by completing an example that does not close with Priest's rule and generating a countermodel from the open tableau. Then, I complete the previous open tableau and generate a counterexample with the proposed rule. Lastly, I provide the soundness and completeness proofs for the system of logic.

2.1 Universal Instantiation in Free Logic

The tableau procedure[2] is essentially a test to see whether or not the premises and negated conclusion of some inference leads to a contradiction (1.5). If the premises and negated conclusion do lead to a contradiction this indicates that the inference is valid; if the premises and negated conclusion do not lead to a contradiction this indicates that the inference is not valid.

A tableau is complete when every rule that can be applied has been applied (1.4.5). A branch of a tableau closes as soon as there is a contradiction, formulas A and $\neg A$, on the branch. Because subsequent steps will not undo the contradiction, the branch may be closed as soon as the contradiction appears, whether or not all the rules on that branch have been applied. An 'X' at the bottom of a branch is used to indicate closure. A tableau is closed when every branch on the tableau has been closed (1.4.6).

Most of the rules of Classical Logic tableaux carry over to Free Logic tableaux. They

[2] Priest explains his choice of the tableau method in the Preface to the First Edition (2008) in the following way, "constructing tableau proofs, and so 'getting a feel' for what is, and is not, valid in logic, is very easy ... tableaux have great pedagogical attractions" (xviii). Although my rule supports this view, the aim of this paper is reasonably narrow in scope in adopting this cause and certainly does not go into anything that could be considered an argument for the tableau method in general.

are the following (1.4.4):

$$\begin{tikzcd} & A \supset B \arrow[dl] \arrow[dr] & \\ \neg A & & B \end{tikzcd} \qquad \begin{tikzcd} \neg(A \supset B) \arrow[d] \\ A \\ \neg B \end{tikzcd}$$

$$\begin{tikzcd} & A \lor B \arrow[dl] \arrow[dr] & \\ A & & B \end{tikzcd} \qquad \begin{tikzcd} \neg(A \lor B) \arrow[d] \\ \neg A \\ \neg B \end{tikzcd}$$

$$\begin{tikzcd} & \neg(A \land B) \arrow[dl] \arrow[dr] & \\ \neg A & & \neg B \end{tikzcd} \qquad \begin{tikzcd} A \land B \arrow[d] \\ A \\ B \end{tikzcd}$$

$$\begin{tikzcd} & A \equiv B \arrow[dl] \arrow[dr] & \\ A & & \neg A \\ B & & \neg B \end{tikzcd} \qquad \begin{tikzcd} & \neg(A \equiv B) \arrow[dl] \arrow[dr] & \\ A & & \neg A \\ \neg B & & B \end{tikzcd}$$

$$\begin{tikzcd} \neg\neg A \arrow[d] \\ A \end{tikzcd}$$

The following two rules carry over to Free Logic from Classical First Order Logic (12.4.1):

$$\begin{tikzcd} \neg\forall x A \arrow[d] \\ \exists x \neg A \end{tikzcd} \qquad \begin{tikzcd} \neg\exists x A \arrow[d] \\ \forall x \neg A \end{tikzcd}$$

The systems of Free Logic dispense with the assumption of Classical Logic that every object in the domain exists[3]. This means that constants can refer to non-existent objects. This allows us to distinguish between something "existing" in the sense of being within the domain, D, of objects we wish to reason over, and of those objects, "existence proper" in the sense of being within the sub-domain of objects, E, that we are to say exist.

The division between what exists and what does not exist is relatively straightforward in Free Logic: certain propositions about Barack Obama, such as, 'Barack Obama lives at 1600 Pennsylvania Avenue', are true because of the way the physical world is; certain propositions about Sherlock Holmes, such as, 'In the story, Sherlock Holmes lives at 221B

[3] See Priest 13.8 for a historical overview.

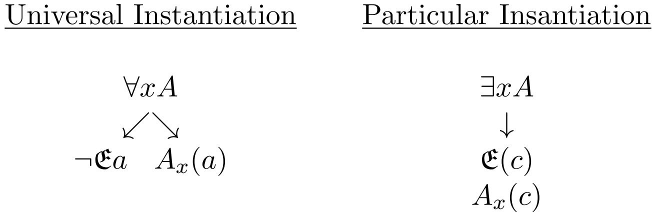
Baker Street', are true because of decisions made by Sir Arthur Conan Doyle. We will look at a system that treats more complicated notions of existence in section four.

To express existence and nonexistence, an existence predicate, $\mathfrak{E}$, is introduced to the machinery of classical logic (12.6.2-4). $\mathfrak{E}a$ denotes 'a exists'; $\neg\mathfrak{E}a$ denotes 'a does not exist'.

The truth conditions and rules for Free Logic[4] are the same as those of Classical Logic (above), except for modifications in Universal Instantiation and Particular Instantiation. In Free Logic, if C is a set of constants such that every object in D has a name in C, then the truth conditions for the quantifiers are (13.2.6):

$$v(\forall xA) = 1 \text{ iff for all } c \in C \text{ such that } v(\mathfrak{E}c) = 1,\ v(A_x(c)) = 1 \text{ (otherwise it is 0);}$$
$$v(\exists xA) = 1 \text{ iff for some } c \in C \text{ such that } v(\mathfrak{E}c) = 1,\ v(A_x(c)) = 1 \text{ (otherwise it is 0).}$$

This is reflected in Free Logic tableaux by the following rules (13.10.2):



The Particular Instantiation Rule for $\exists xA$ introduces some object that exists, represented by c, a new constant to the branch, which has the property A. The Universal Instantiation Rule for $\forall xA$ says that either an object does not exist (left branch) or it has the property A (right branch). The Universal Instantiation Rule is applied for all constants on the branch. If there is no constant on the branch when the rule is to be instantiated, one is introduced.

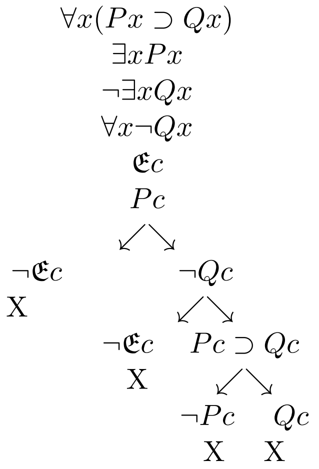
In cases where there is a constant, a, predicated by the existence predicate, $\mathfrak{E}$, on the branch before the Universal Instantiation Rule is applied, the left branch of the rule always closes because the $\mathfrak{E}a$ already on the branch contradicts the introduced $\neg\mathfrak{E}a$. In certain tableaux where this occurs several times, the left branch repeatedly is created only to be subsequently closed by the contradiction presented. This is seen in the following problem from Chapter 13 (13.10.2):

[4] In the chapter on Free Logic, Priest provides rules suitable for Positive Free Logic and Negative Free Logic (13.4). With the supplementation of the Universal and Particular Instantiation Rules to the rules of Classical First Order Logic, the system arrives at Positive Free Logic. By adding the Negativity Constraint Rule to these, the system arrives at Negative Free Logic. This means that the rules I provide here could apply to either Positive or Negative Free Logic, depending on whether or not one also works with the Negativity Constraint Rule (13.4.3).

2.2 Closed Tableaux in Free Logic

A. Closed tableau completed using Priest's Universal Instantiation Rule:

$$\forall x(Px \supset Qx),\ \exists x Px \models \exists x Qx$$



I propose avoiding such series of branches and closures by simplifying the rule: instantiate $\forall xA$ on a branch only if there is already a constant on the branch that refers to something that is said to exist. In reading an interpretation off the branch, if it is not explicitly stated in the tableau that an object exists, it is understood that the object does not exist.[5]

The Proposed Universal Instantiation Rule[6] can be stated as such:

$$\begin{array}{c} \forall xA \\ \mathfrak{E}a \\ \downarrow \\ A_x(a) \end{array}$$

[5] This feature will be addressed in detail in section 2.3.

[6] It may be asserted that although the proposed rule leads to tableaux with fewer branches, Priest's rule is "more intuitive". Talk of intuition in such a case seems to reflect merely the sort of rule one is accustomed to – a property of the evaluator rather than an objective property of the rule itself. Indeed, it was because there was something that seemed unintuitive (whatever that means – here taking it to mean what seemed unnatural to me) about generating multiple branches only to close them immediately after that led to the proposed rule. However, because of the flimsiness of such claims about intuitiveness I will not assert that the proposed rule is "more intuitive" and for the same reason I do not think claims that rest on intuition are persuasive against this proposal.

a follows the existence predicate, $\mathfrak{E}$, on the branch, which means that it refers to an existent object. In the absence of a constant that follows the existence predicate on the branch, $\forall xA$ is not instantiated.

To demonstrate how this rule would simplify tableaux and to show the contrast between tableaux created with the two rules, I will first apply the rule to the example above and will then apply it to a set of formulas that generates a countermodel.

B. Closed tableau from above completed using Proposed Universal Instantiation Rule:

$$\forall x(Px \supset Qx),\ \exists xPx \models \exists xQx$$

$$\begin{array}{c}
\forall x(Px \supset Qx) \\
\exists xPx \\
\neg\exists xQx \\
\forall x\neg Qx \\
\mathfrak{E}a \\
Pa \\
\neg Qa \\
Pa \supset Qa
\end{array}$$

$$\begin{array}{cc}
\swarrow & \searrow \\
\neg Pa & Qa \\
\text{X} & \text{X}
\end{array}$$

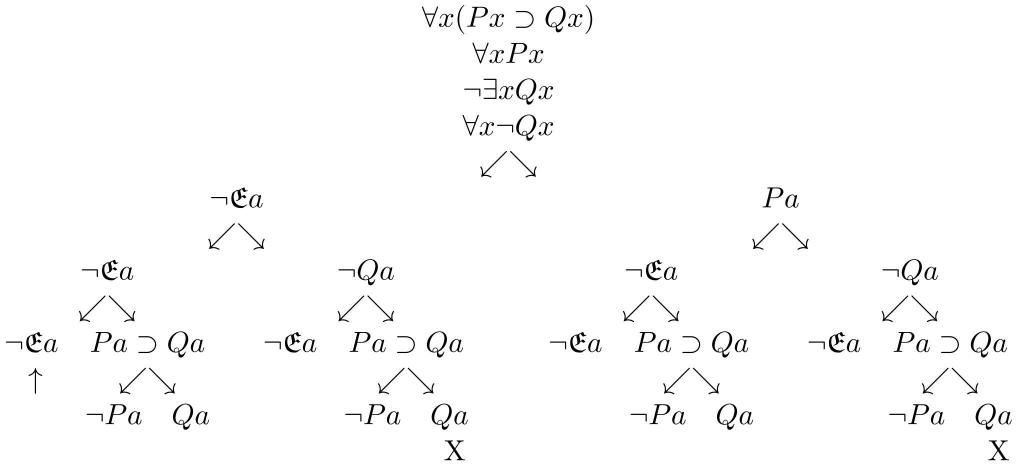
Tableau A has 9 lines and branches 3 times; Tableau B has 9 lines and branches once.

2.3 Open Tableaux in Free Logic

I will now apply Priest's Universal Instantiation Rule and the Proposed Universal Instantiation Rule to an example that generates a countermodel.

C. Open tableau completed using Priest's Universal Instantiation Rule:

$$\forall x(Px \supset Qx),\ \forall xPx \models \exists xQx$$



In Free Logic, an interpretation is a triple $< D, E, v >$ where the domain, D, is a non-empty set consisting of all objects, E is the inner domain of D, the set of all *existent* objects, and v assigns every formula a truth value. The procedure to read a countermodel off an open branch in Free Logic is to assign a distinct object, δ_a, for every constant, a, that occurs on the branch. The valuation of a predicate, P, is the set of n-tuples assigned to constants that follow P on the branch (12.4.8). For Priest, if a predicate or constant does not occur on the branch, it may be assigned any value. Priest calls this the "don't care" condition (12.4.8).

Because the domain, D, is a *non-empty* set, if there is no constant present in the tableau, an arbitrary object, δ, must be introduced to D when generating the countermodel (14.3.8). The introduced object will not exist. It may or may not be within the extension of any predicate other than the existence predicate.

For the countermodel determined by the open branch of tableau C there is a constant, a, on the branch, so one need not be introduced. The countermodel generated by the tableau is: $D = \{\delta_a\},\ E = \{\emptyset\}$, and $v(a) = \delta_a$, $v(P) = \emptyset$, $v(Q) = \emptyset$. This can be depicted as:

Countermodel:	δ_a
$\mathfrak{E}$	X
P	X
Q	X

In the table above, an 'X' indicates the object in the column is not in the extension of the predicate in the row; a check would indicate that the object in the column is in the extension of the predicate in the row (12.4.8). $\forall x(Px \supset Qx)$ and $\forall xPx$ are trivially true because there is no existent object in the domain. $\exists xQx$ is false because there is not an existent object that has the property Q. The counterexample demonstrates that $\forall x(Px \supset Qx), \forall xPx \models \exists xQx$ is invalid.

Tableau C completed with the Proposed Universal Instantiation Rule for Free Logic is as follows.

D. Open tableau completed using Proposed Universal Instantiation Rule:

$$\forall x(Px \supset Qx),\ \forall xPx \models \exists xQx$$

$$\begin{array}{c} \forall x(Px \supset Qx) \\ \forall xPx \\ \neg\exists xQx \\ \forall x\neg Qx \\ \uparrow \end{array}$$

The Proposed Universal Instantiation Rule is not applied because the there is no constant that follows the existence predicate on the branch and the tableau is complete.

As noted above, in the countermodel procedure detailed by Priest, if a predicate is not on the branch, "the value given to it by v is a *don't care* condition: it can be anything one likes" (12.4.8). That is, on the Priest model of reading a countermodel off tableaux, if a constant or predicate is not on the branch, it may be said to exist or not exist, to be satisfied or not satisfied, respectively. However, with tableaux generated using the Proposed Universal Instantiation Rule this no longer holds. On the model I propose here, predicates other than the existence predicate continue to fall within the "don't care" condition, but existent objects do not. This means that an object is said to exist only if it follows the existence predicate on the open branch. If an object does not do this, it is understood to not exist. It is no longer a "don't care" condition.

The countermodel is generated following a similar procedure as above, except that here the denotation of a is in the extension of E if and only if $\mathfrak{E}a$ is on the branch. That is, if it is not explicitly stated on the branch that an object is an existent object, the set of existent objects is taken to be empty. Because formula D is invalid, existence and nonexistence are easily reflected in the countermodel by including those objects which are stated to exist in the tableau in the model and excluding those which are not stated to exist in the tableau from the model.

Generation of the countermodel for Tableau D requires keeping in mind the subtlety I mentioned above: because there is no constant present in the tableau, when generating the countermodel, an arbitrary object, denoted by δ_a, is introduced so that domain D is non-empty (14.3.8). δ_a is not in the extension of $\mathfrak{E}$, P or Q. This may be illustrated as follows and may be verified in the same way as Tableau C.

Countermodel:	δ_a
$\mathfrak{E}$	X
P	X
Q	X

Tableau C has 8 lines and branches 11 times; Tableau D has 4 lines and does not branch. Both executions arrived at the same countermodel. As is demonstrated by this example executed two different ways, the proposed rule can simplify tableaux.

2.4 Soundness and Completeness in Free Logic

Priest's Locality Lemma (13.7.2), Denotation Lemma (13.7.3), and Corollary (13.7.4) are unchanged.

Theorem: *The tableaux of free logic are sound with respect to their semantics.*

The proof is as in (13.7.5). The only difference in the Soundness Lemma is in the case of universal instantiation. For universal instantiation, we have the following:

Soundness is proven by demonstrating that, if we assume that on some interpretation, I, everything on the branch thus far is true, application of the rule in question maintains truth on at least one of the branches, on some interpretation I'. The proposed Universal Instantiation Rule is applied when two conditions are met: $\forall xA$ and $\mathfrak{E}a$ are both on the branch. Consider an interpretation, I, on which, by the Inductive Hypothesis, these two conditions are met. The truth of $\forall xA$ means that for every object in the inner domain (every existent object) that object satisfies A. That is, for all $d \in E$ $A_x(k_d)$ is true. Let the constant a refer to such a d. Then since $\mathfrak{E}a$ is on the branch, d is in the inner domain. By the Denotation Lemma, $A_x(k_d)$ is true in I if and only if $A_x(a)$ is true. $A_x(a)$ is true in I. Thus, we can take I' to be I.

Theorem: *The tableaux of free logic are complete with respect to their semantics.*

Completeness is proven by demonstrating that for every formula, A, on an open branch, B

If A is on B then $v(A) = 1$, and

If $\neg A$ is on B then $v(A) = 0$.

The open branch, B, induces an interpretation that is defined as follows. Let C be the set of all constants on B. There are two domains at play: D and E. D consists of every object that has been named on the branch. D cannot be empty so if there is no constant on the branch we must introduce a constant that denotes an arbitrary object. The second domain, E, is the inner domain or class of existent objects. E consists of only those objects on the branch which are named by some constant, c, that is preceded by the existence predicate,

$\mathfrak{E}$. For the extension of the predicates, if it is not stated on the branch that some predicate applies to some object, then the object is not in the extension of that predicate.

The only cases that involve the modified rule of universal instantiation are the cases in the induction for the truth of a universally quantified sentence, and falsity for an existentially quantified sentence. These are as follows.

Suppose that $\forall xA$ is on the branch. We must show that $\forall xA$ is true in the induced interpretation. Take some object in the domain, d. Let c denote d. If $d \in E$, $\mathfrak{E}c$ is on the branch. Since $\forall xA$ and $\mathfrak{E}c$ are on the branch, $A_x(c)$ will also be on the branch. By the Inductive Hypothesis, $v(A_x(c)) = 1$. By the Denotation Lemma $v(A_x(k_d)) = 1$. Thus $v(\forall xA) = 1$.

Suppose that $\neg\exists xA$ is on the branch. This means that $\forall x\neg A$ is also on the branch. Reasoning exactly as before, it follows that $v(\forall x\neg A) = 1$. And since $\forall x\neg A$ is logically equivalent to $\neg\exists xA$, $v(\neg\exists xA) = 1$.

The tableaux of Free Logic are sound and complete with respect to their semantics with the Proposed Universal Instantiation Rule replacing Priest's Universal Instantiation Rule.

3.1 Universal Instantiation in Variable Domain Modal Logic

With a slight modification, the proposed Universal Instantiation Rule may be carried over to Variable Domain Modal Logic.

Variable Domain Modal Logic is Free Logic supplemented with the possibility and necessity operators from Constant Domain Modal Logic.[7] As such, the notion of possible worlds is reflected in Priest's slight adaptation of his Free Logic Rules. His Variable Domain Modal Logic Rules is below (15.4.1). A world is given in the rules as i.

Universal Instantiation

$$\begin{tikzcd} & \forall xA,\ i \arrow[dl] \arrow[dr] & \\ \neg\mathfrak{E}a,\ i & & A_x(a),\ i \end{tikzcd}$$

Particular Instantiation

$$\begin{tikzcd} \exists xA,\ i \arrow[d] \\ \mathfrak{E}c,\ i \\ A_x(c),\ i \end{tikzcd}$$

Application of the branching rule for Universal Instantiation in Variable Domain Modal Logic often leads to a series of branches and immediate closures if the object, a, is already on the branch for the world, i. This is similar to the branching demonstrated above in Free Logic Tableaux A and C. The difference is that in Free Logic, the left branch would close if the object were anywhere on the branch; whereas with Variable Domain Modal Logic, the left branch will only close only if the object is on the branch *and* in the same world as that for which the rule is applied. Priest 15.4.2 provides an example below. All Variable Domain Modal Logic tableaux are here done in logic K, meaning that there are

[7] See Priest 15.10 for a historical overview.

no constraints on accessibility relations. The proposed technique also extends to stronger modal logics.

3.2 Closed Tableaux in Variable Domain Modal Logic

E. Closed tableau completed using Priest's Universal Instantiation Rule:

$$\Box\forall x(A \supset B) \supset (\Box\forall xA \supset \Box\forall xB)$$

$$\begin{array}{c}
\neg(\Box\forall x(A \supset B) \supset (\Box\forall xA \supset \Box\forall xB)),\ 0 \\
\Box\forall x(A \supset B),\ 0 \\
\neg(\Box\forall xA \supset \Box\forall xB),\ 0 \\
\Box\forall xA,\ 0 \\
\neg\Box\forall xB,\ 0 \\
\Diamond\neg\forall xB,\ 0 \\
0r1 \\
\neg\forall xB,\ 1 \\
\exists x\neg B,\ 1 \\
\mathfrak{E}a, 1 \\
\neg B_x(a),\ 1 \\
\forall xA,\ 1 \\
\forall x(A \supset B),\ 1
\end{array}$$

$$\begin{array}{cccc}
 & & \swarrow \qquad \searrow & \\
\neg\mathfrak{E}a,\ 1 & & A_x(a),\ 1 & \\
\text{X} & & \swarrow \qquad \searrow & \\
 & \neg\mathfrak{E}a,\ 1 & & A_x(a) \supset B_x(a),\ 1 \\
 & \text{X} & & \swarrow \qquad \searrow \\
 & & \neg A_x(a),\ 1 & B_x(a),\ 1 \\
 & & \text{X} & \text{X}
\end{array}$$

Echoing the Free Logic Rule presented above, is the following proposed rule for Universal Instantiation in Variable Domain Modal Logic:

Universal Instantiation

$$\begin{array}{c}
\forall xA,\ i \\
\mathfrak{E}a,\ i \\
\downarrow \\
A_x(a),\ i
\end{array}$$

Tableau E completed with the Proposed Universal Instantiation Rule for Variable Domain Modal Logic goes as follows.

F. Closed tableau completed using Proposed Universal Instantiation Rule:

$$\Box\forall x(A \supset B) \supset (\Box\forall xA \supset \Box\forall xB)$$

$$\begin{array}{c}
\neg(\Box\forall x(A \supset B) \supset (\Box\forall xA \supset \Box\forall xB)),\ 0 \\
\Box\forall x(A \supset B),\ 0 \\
\neg(\Box\forall xA \supset \Box\forall xB),\ 0 \\
\Box\forall xA,\ 0 \\
\neg\Box\forall xB,\ 0 \\
\Diamond\neg\forall xB,\ 0 \\
0r1 \\
\neg\forall xB,\ 1 \\
\exists x\neg B,\ 1 \\
\mathfrak{E}a, 1 \\
\neg B_x(a),\ 1 \\
\forall xA,\ 1 \\
\forall x(A \supset B),\ 1 \\
A_x(a),\ 1 \\
A_x(a) \supset B_x(a),\ 1 \\
\swarrow \quad \searrow \\
\begin{array}{cc}
\neg A_x(a),\ 1 & B_x(a),\ 1 \\
\text{X} & \text{X}
\end{array}
\end{array}$$

Tableau E has 16 lines and branches 3 times; Tableau F has 16 lines and branches once.

3.3 Open Tableaux in Free Logic

I will now apply Priest's Universal Instantiation Rule and the Proposed Universal Instantiation Rule to a formula that generates a countermodel in Variable Domain Modal Logic. This example is chosen because it demonstrates how the modification to the Free Logic Rule I introduced above affects the application of the rule here for Variable Domain Modal Logic. There is not a dramatic reduction in branches as was seen in Tableaux C and D, but an example with similarly dramatic results could be constructed.

G. Open tableau (Barcan Formula) completed using Priest's Universal Instantiation Rule:

$$\forall x\Box Px \supset \Box\forall xPx$$

$$\begin{array}{c}
\neg(\forall x\Box Px \supset \Box\forall xPx),\ 0\\
\forall x\Box Px,\ 0\\
\neg\Box\forall xPx,\ 0\\
\Diamond\neg\forall xPx,\ 0\\
0r1\\
\neg\forall xPx,\ 1\\
\exists x\neg Px,\ 1\\
\mathfrak{E}a,\ 1\\
\neg Pa,\ 1\\
\swarrow\searrow\\
\begin{array}{cc}
\neg\mathfrak{E}a,\ 0 & \Box Pa,\ 0\\
\uparrow & Pa,\ 1\\
 & \text{X}
\end{array}
\end{array}$$

A countermodel is read off an open branch of the tableau following the same basic procedure as above. In Variable Domain Modal Logic, an interpretation is the quadruple $< D, W, R, v >$ where the domain, D, is the non-empty domain, W is a non-empty set of worlds, R is a binary accessibility relation on W, and v assigns every formula a truth value at a world. Thus, for the countermodel determined by the open branch of Tableau G, $W = \{w_0, w_1\}$, w_0Rw_1, $D = \{\delta_a\}$, $v(w_0) = Dw_0 = vw_0(\mathfrak{E}) = \emptyset$, $v(w_1) = D_{w_1} = v_{w_1}(\mathfrak{E}) = \{\delta_a\}$, $v_{w_0}(P) = v_{w_1}(P) = \emptyset$, and $v(a) = \delta_a$. This can be depicted as:

$$\begin{array}{cccccccc}
 & \delta_a & & & & & \delta_a\\
\mathfrak{E} & \text{X} & w_0 & \longrightarrow & w_1 & \mathfrak{E} & \surd\\
P & \text{X} & & & & P & \text{X}
\end{array}$$

$\forall x\Box Px$ is true at w_0 because there are no objects that exist; as such, it is trivially the case that for every object that exists at w_0 that object is P. $\Box\forall xPx$ is false because there is some world accessible from w_0 – w_1 – at which there is an existent object and this world does not have the property P. The countermodel demonstrates that $\forall x\Box Px \supset \Box\forall xPx$ is invalid.

Although it is not the case in this example, if there were no constant introduced in the tableau, an arbitrary object, δ, would need to be introduced to the domain of D (14.3.8).

H. Open tableau (Barcan Formula) completed using Proposed Universal Instantiation Rule:

$$\forall x \Box Px \supset \Box \forall x Px$$

$$\begin{array}{c}
\neg(\forall x \Box Px \supset \Box \forall x Px),\ 0 \\
\forall x \Box Px,\ 0 \\
\neg \Box \forall x Px,\ 0 \\
\Diamond \neg \forall x Px,\ 0 \\
0r1 \\
\neg \forall x Px,\ 1 \\
\exists x \neg Px,\ 1 \\
\mathfrak{E}a,\ 1 \\
\neg Pa,\ 1 \\
\uparrow
\end{array}$$

Although the constant a is on the branch, the Universal Instantiation Rule is not applied because it is not in the same world as the universal quantifier. In the example, the existent object is in world 1 but $\forall x \Box Px$ is at world 0 and thus the Proposed Universal Instantiation Rule is not applied. If there had been an existent object at world 0, line 2 could have been instantiated. This aspect of the example highlights this difference between the rule I introduced above for Free Logic, and the rule I introduce here for Variable Domain Modal Logic.

The countermodel is generated following the same procedure as above and may be verified in the same way.

$$\begin{array}{ccccccc}
 & \delta_a & & & & & \delta_a \\
\mathfrak{E} & \mathrm{X} & w_0 & \longrightarrow & w_1 & \mathfrak{E} & \surd \\
P & \mathrm{X} & & & & P & \mathrm{X}
\end{array}$$

Tableau G has 11 lines and branches 1 time; Tableau H has 9 lines and does not branch. Both executions arrived at the same countermodel. As is demonstrated by this example executed two different ways, the proposed rule can greatly simplify tableaux in Variable Domain Modal Logic.

3.4 Soundness Completeness in Variable Domain Modal Logic

The Locality Lemma (15.9.3) and Denotation Lemma (15.9.4) are unchanged.

Theorem: *The tableaux of variable domain K are sound with respect to their semantics.*

The proof is as in (15.9.5). The only difference in the Soundness Lemma is in the case of universal instantiation. For universal instantiation, we have the following:

Soundness is proven by demonstrating that, if we assume that on some interpretation, I, everything on the branch thus far is true, application of the rule in question maintains truth on at least one of the branches, on some interpretation, I'. Let f be a function that shows interpretation I to be faithful to the branch. Consider an application of the rule:

$$\begin{gathered} \forall xA,\ i \\ \mathfrak{E}a,\ i \\ \downarrow \\ A_x(a),\ i \end{gathered}$$

Assume that in I $\forall xA$ is true at $f(i)$ and Ea true at $f(i)$. This means that $\forall d \in D_{f(i)}$ and $A_x(k_d)$ true at $f(i)$. Let d be such that $v(a) = v(k_d)$. Because $\mathfrak{E}a$ true at $f(i)$ this means $d \in D_{f(i)}$. This means $A_x(k_d)$ is true at $f(i)$. By the Denotation Lemma (15.9.4) $A_x(a)$ is true at $f(i)$. Hence, I is faithful to the branch, and we can take I' to be I.

Theorem: *The tableaux of variable domain K are complete with respect to their semantics.* (The proof is as in 15.9.6 with a small modification.)

Completeness is proven by demonstrating that for every formula, A, on an open branch, B

If $A,\ i$ is on B then $v_{W_i}(A) = 1$, and

If $\neg A,\ i$ is on B then $v_{W_i}(A) = 0$.

The open branch, B, induces an interpretation $< D, W, R, v >$ that is defined as follows. Let C be the set of all constants on the branch. $W = \{w_i : i \text{ occurs on B}\}$. $w_i R w_j$ if and only if irj occurs on B. $D = \{\delta_a : a \in C\}$ (or if C is empty, $D = \{\delta\}$, for some arbitrary δ). For all constants, a, on B, $v(a) = \delta_a$. For every n-place predicate on B (including $\mathfrak{E}$), $< \delta_{a_1} ..., \delta_{a_n} > \in v_{W_i}$ if and only if $Pa_1...a_n,\ i$ is on B. $D_{W_i} = v(w_i) = v_{W_i}(\mathfrak{E}) = \{\delta_a : \mathfrak{E}a,\ i \text{ occurs on B}\}$. This means $\delta_a \in D_{W_i}$ if and only if $\mathfrak{E}a,\ i$ is on the branch.

The only cases that involve the modified universal instantiation rule are the cases in the induction for the truth of the universally quantified sentence, and falsity for an existentially quantified sentence. These are as follows.

Suppose that $\forall xA,\ i$ is on the branch. We must show that $\forall xA$, is true in the induced interpretation at world w_i. This means for $\forall d \in D_{W_i} A_x(k_d)$ is true at w_i. Suppose that $d \in D_{W_i}$. Let c denote d. Because $d \in D_{W_i}$ this means that $\mathfrak{E}c,\ i$, is on B. So, we have applied the modified universal instantiation rule and $A_x(c),\ i$ is on B. By the Induction Hypothesis, $A_x(c)$ is true at w_i and $A_x(k_d)$ is true at w_i by the Denotation Lemma. Thus $v_{W_i}(\forall xA) = 1$.

Suppose that $\neg\exists xA,\ i$ is on the branch. This means that $\forall x\neg A,\ i$ is also on the branch. Reasoning exactly as before, it follows that $v_{W_i}(\forall x\neg A) = 1$. And since $\forall x\neg A,\ i$ is logically equivalent to $\neg\exists xA,\ i$ $v_{W_i}(\neg\exists xA) = 1$.

The tableaux of variable domain K are sound and complete with respect to their semantics with the Proposed Universal Instantiation Rule replacing Priest's Universal Instantiation Rule.

4.1 Universal and Particular Instantiation in Intuitionist Logic

The same basic idea may be carried over to Intuitionist Logic, which has two rules each for Universal and Particular Instantiation – one that branches and one that does not. The principle I have applied thus far will apply to both Intuitionist Logic rules that involve branching, for Universal Instantiation, as above, and for Particular Instantiation, for the first time here.

Intuitionist Logic provides the machinery to account for the sort of questions about existence and non-existence that I have been discussing, but now considered with respect to mathematics.[8] Something exists in Intuitionist Logic when there is a construction of it. A new construal of truth and falsity is needed to account for the fact that we are now working with constructs. This also has repercussions for negation: a proof of $\neg A$ is a proof that there is no proof of A (6.2.7). The symbol $\neg$ is introduced to capture this new sense of negation. That is necessary because, in Intuitionist Logic, if there is not a proof that something is the case, this does not mean that there is a proof that there is no proof that it is the case. A new notion of the conditional is also required. A proof of $A \sqsupset B$ is "a construction that, given any proof of A, can be applied to give a proof of B" (6.2.7). $\forall xA$ is true when for everything that there is or could be a construction of, there is a proof that it is A. $\exists xA$ is true when there is something that there is a construction of, and there is a proof that this thing satisfies A.

$\neg A$ in Intuitionism is stronger than falsity (untruth) because it indicates there is a proof that there is no proof of A. Because in Intuitionism A and $\neg A$ are not complements, we can no longer indicate that A is false at a world with '$\neg A$'. Instead, to denote falsity, the rules for each operator are twofold: one for truth, denoted with a plus sign before the world, and one for falsity, denoted with a minus sign before the world.

Tableaux of Intuitionist Logic are done in logic K$\rho\tau$, meaning that the accessibility relations, R, are reflexive and transitive, but not symmetric – historically known as S4. With Variable Domain Modal Logic, the reflexivity relations may be set in any way that serves the purposes of the proof. With Intuitionist Logic, there are conceptual reasons for requiring R to be reflexive and transitive. We have transitivity and reflexivity because once something is proven to be true it cannot be *un*proven. If something is proven to not be true in the future that means it was not really true in the past, although it may have been thought to be so. There is no symmetry because when evidence is found in the future that proves something to be the case, this does not mean there was evidence that proved it to be true at that point in the past before it was proven.

[8] See Priest 6.8 for a historical overview.

Now that I have explained the motivation for this system of logic, I will explain the rules Priest gives in the chapter.

We must introduce a line of the form $\mathfrak{E}a,\ +0$ for every object, a, given in the premises or conclusion; if there is no constant given in the premises or conclusion, we must introduce an existent object to world 0 at the beginning of the tableau (20.3.7; 20.4.1).

Every atomic sentence that is true at some world is true at every accessible world (6.3). This holds true for sentences using the existence predicate, meaning every object that has been constructed at a world also exists at all accessible worlds. This is captured by the Heredity Rule (20.4.1):

$$\begin{array}{c} Pa_1...a_n,\ +i \\ irj \\ \downarrow \\ Pa_1...a_n,\ +j \end{array}$$

The rules for Universal and Particular Instantiation are as follows (20.4.1):

$$\begin{array}{cc}
\underline{\text{Universal Instantiation}} & \underline{\text{Particular Instantiation}} \\
\\
\begin{array}{c} \forall xA,\ +i \\ irj \\ \swarrow\ \searrow \\ \mathfrak{E}a,\ -j \quad A_x(a),\ +j \end{array} &
\begin{array}{c} \exists xA,\ -i \\ \swarrow\ \searrow \\ \mathfrak{E}a,\ -i \quad A_x(a),\ -i \\ \\ \end{array} \\
\\
\begin{array}{c} \forall xA,\ -i \\ \downarrow \\ irj \\ \mathfrak{E}c,\ +j \\ A_x(c),\ -j \end{array} &
\begin{array}{c} \exists xA,\ +i \\ \downarrow \\ \mathfrak{E}c,\ +i \\ A_x(c),\ +i \\ \\ \end{array}
\end{array}$$

Application of this set of rules often leads to a series of branches and swift closures if the object, a, is already on the branch for the world, i, or if the heredity rule can be applied to close the branch. This looks similar to the branching demonstrated above in Free Logic Tableaux A and C and Variable Domain Modal Logic Tableaux D and F. Priest 20.4.2 provides such an example below.

4.2 Closed Tableaux in Intuitionist Logic

I. Closed tableau completed using Priest's Universal Instantiation Rule:

$$\forall x Px \sqsupset Pa$$

$$\begin{array}{c}
\mathfrak{E}a,\ +0 \\
\forall x Px \sqsupset Pa,\ -0 \\
0r0 \\
0r1,\ 1r1 \\
\forall x Px,\ +1 \\
Pa,\ -1 \\
\swarrow \searrow \\
\begin{array}{cc}
\mathfrak{E}a,\ -1 & Pa,\ +1 \\
\mathfrak{E}a,\ +1 & \text{X} \\
\text{X} &
\end{array}
\end{array}$$

The branching that was seen with the rules for Free and Variable Domain Modal Logic is exacerbated with this set of rules because now there is also a branching rule for the particular quantifier. Priest writes that the resulting tableaux are "perspicuous, but can be unwieldy, due to the branching delivered by the second pair of quantifier rules" (20.5) and names this sort of tableaux *tableaux of kind 1.*

Priest proposes *tableaux of kind 2* that require "extra book-keeping" but that simplify tableaux by eliminating the existence predicate, requiring instead information which is on the tableau, but only implicitly. He proposes replacing the rules above with the following (20.5.4):

$$\begin{array}{cc}
\underline{\text{Universal Instantiation}} & \underline{\text{Particular Instantiation}} \\
 & \\
\begin{array}{c} \forall xA,\ +i \\ irj \\ \downarrow \\ A_x(a),\ +j \end{array} & \begin{array}{c} \exists xA,\ -i \\ \downarrow \\ A_x(a),\ -i \\ \\ \end{array} \\
 & \\
\begin{array}{c} \forall xA,\ -i \\ \downarrow \\ irj \\ A_x(c),\ -j \end{array} & \begin{array}{c} \exists xA,\ +i \\ \downarrow \\ A_x(c),\ +i \\ \\ \end{array}
\end{array}$$

For the top particular instantiation rule the object a must "belong to" i and for the top universal instantiation rule, the object a must "belong to" j (20.5.3). For an object to "belong to" a world that means it exists at that world. As one applies these rules, one must use information implicit on the branch, as it is not explicitly visible anywhere on the branch. Tableaux of this form are less unwieldy than those of kind 1 but they are less perspicuous.

I propose a revised version of these rules in the model of those I have presented above. It has the reduction in branches of tableaux of kind 2 and the perspicuity of tableaux of kind 1. These rules essentially spell out what was kept in ones head in tableaux of kind 2. In this rule $\mathfrak{E}a,\ i$ indicates a belongs to i. I propose the following rules for Intuitionist Logic to replace the current branching rules of tableaux of kind 1 and the top rules of tableaux of kind 2:

$$\begin{array}{cc}
\underline{\text{Universal Instantiation}} & \underline{\text{Particular Instantiation}} \\
 & \\
\forall xA,\ +i & \exists xA,\ +i \\
irj & \mathfrak{E}c\ +i \\
\mathfrak{E}a\ +j & \downarrow \\
\downarrow & A_x(c),\ +i \\
A_x(a),\ +j &
\end{array}$$

Because it is a requirement of this rule that there exists an object in the relevant domain for the rule to be applied, it is advisable to apply the Heredity Rule as soon as there is a new variable or world to apply it to.

Tableau I completed with the Proposed Universal Instantiation Rule for Intuitionist Logic goes as follows.

J. Closed tableau completed using Proposed Rules:

$$\forall xPx \sqsupset Pa$$

$$\begin{array}{c}
\mathfrak{E}a,\ +0 \\
\forall xPx \sqsupset Pa,\ -0 \\
0r0 \\
0r1,\ 1r1 \\
\mathfrak{E}a,\ +1 \\
\forall xPx,\ +1 \\
Pa,\ -1 \\
Pa\ +1 \\
\text{X}
\end{array}$$

Tableau I has 8 lines and branches once; Tableau J has 8 lines and does not branch.

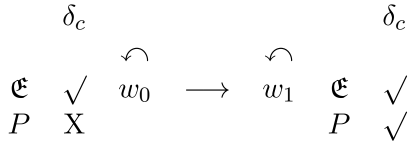
4.3 Open Tableaux in Intuitionist Logic

K. Open tableau completed using Priest's False Particular Instantiation Rule (20.4.2):

$$\exists x(Px \vee \neg Px)$$

$$\begin{array}{cc}
\multicolumn{2}{c}{\mathfrak{E}c,\ +0}\\
\multicolumn{2}{c}{\exists x(Px \vee \neg Px),\ -0}\\
\multicolumn{2}{c}{0r0}\\
\swarrow & \searrow\\
\mathfrak{E}c,\ -0 & Pc \vee \neg Pc,\ -0\\
\text{X} & Pc,\ -0\\
 & \neg Pc,\ -0\\
 & 0r1,\ 1r1\\
 & Pc,\ +1\\
 & \mathfrak{E}c,\ +1\\
 & \uparrow
\end{array}$$

To read a countermodel off the open branch, we will use the basic structure as above to account for the quantifiers, supplemented with the methods of propositional Intuitionist Logic (6.4.8). If the branch states that some node is true at some world, it is indicated as true in the countermodel. If some node is stated to be false at some world, it is indicated as false in the countermodel. The countermodel given by the open branch of tableau K, $W = \{w_0, w_1\}$, w_0Rw_0, w_0Rw_1, w_1Rw_1, $D = \{\delta_c\}$, $\mathrm{v(w_0)} = D_{w0} = v(w_0)(\mathfrak{E}) = \{\delta_c\}, v(w_1) = Dw_1 = vw_1(\mathfrak{E}) = \{\delta_c\}$, $vw_0(P) = \emptyset$, $vw_1(P) = \{\delta_c\}$, and $v(c) = \delta_c$. This can be depicted as:



Recall that $\neg Px$ means that there is a proof that there is no proof that Px. $\neg Px$ fails at w_0 because Pc holds at w_1 as given in the right side of the model and thus there can be no proof that there is no proof that Px. Px also fails at w_0 as given in the left side of the model. Thus, for c, it is not the case that it either is Pc or $\neg Pc$. c is the only object in the model. Thus, the countermodel demonstrates that $\exists x(Px \vee \neg Px)$ is invalid.

L. Open tableau completed using Proposed False Particular Instantiation Rule (20.4.2):

$$\exists x(Px \vee \neg Px)$$

$$\begin{gathered}
\mathfrak{E}c,\ +0 \\
\exists x(Px \vee \neg Px),\ -0 \\
0r0 \\
Pc \vee \neg Pc,\ -0 \\
Pc,\ -0 \\
\neg Pc,\ -0 \\
0r1,\ 1r1 \\
Pc,\ +1 \\
\mathfrak{E}c,\ +1 \\
\uparrow
\end{gathered}$$

The countermodel is generated following the same procedure as above and may be verified in the same way.

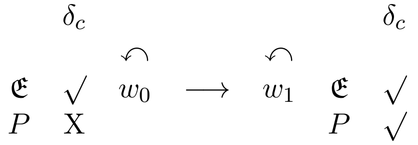


Tableau K has 9 lines and branches once; Tableau L has 9 lines and does not branch. Both executions arrived at the same countermodel. As is demonstrated by this example executed two different ways, the proposed rules can simplify tableaux in Intuitionist Logic and eliminate the need to have two types of rules in the chapter. The proposed rules result in Intuitionistic Logic tableaux that are both perspicuous and not unwieldy.

4.4 Soundness and Completeness in Intuitionist Logic

The Locality Lemma (20.9.2) and Denotation Lemma (20.9.3) are unchanged.

Theorem: *The tableaux of Intuitionist Logic are sound with respect to their semantics.*

The proof is as in 20.9.5. The only differences in the Soundness Lemma are in the cases for truth in Universal Instantiation and falsity in Particular Instantiation.

Soundness is proven by demonstrating that, if we assume that on some interpretation, I, everything on the branch thus far is true, application of the rule in question maintains truth on at least one of the branches, on some interpretation, I'.

We have two rules to consider. For truth in the case of Universal Instantiation we have the following:

Let f be a function that shows interpretation I to be *faithful* to the branch, that is, f is a function from numbers to worlds that indicates which world in an interpretation each number is referring to (2.9.2). Consider an application of the rule:

$$\begin{array}{cc} (i) & \forall xA,\ +i \\ & irj \\ & \mathfrak{E}a,\ +j \\ & \downarrow \\ & A_x(a),\ +j \end{array}$$

Assume that in I $\forall xA$ is true at $f(i)$ and $\mathfrak{E}a$ true at $f(j)$ when $f(i)Rf(j)$. This means that for $\forall d \in D_{f(j)}$ $A_x(k_d)$ is true at $f(j)$. Let d be such that $v(a) = v(k_d)$. Because $\mathfrak{E}a$ is true at $f(j)$ this means $d \in D_{f(j)}$. This means $A_x(k_d)$ is true at $f(j)$. By the Denotation Lemma (20.9.3) $A_x(a)$ is true at $f(j)$. Hence, I is faithful to the branch, and we can take I' to be I.

For the falsity case of Particular Instantiation we have the following:

Let f be a function that shows interpretation I to be faithful to the branch. Consider an application of the rule:

$$\begin{array}{cc} (ii) & \exists xA,\ -i \\ & \mathfrak{E}a,\ +i \\ & \downarrow \\ & A_x(a),\ -i \end{array}$$

Assume that in I $\exists xA$ is false at $f(i)$ and $\mathfrak{E}a$ true at $f(i)$. This means that for $\forall d \in D,\ f(i)$ $A_x(k_d)$ is false at $f(i)$. Let d be such that $v(a) = v(k_d)$. Because $\mathfrak{E}a$ true at $f(i)$ this means $d \in D_{f(i)}$. This means $A_x(k_d)$ is false at $f(i)$. By the Denotation Lemma (20.9.3) $A_x(a)$ is false at $f(i)$. Hence, I is faithful to the branch, and we can take I' to be I.

Theorem: *The tableaux of Intuitionist Logic are complete with respect to their semantics.*

Completeness is proven by demonstrating that for every formula, A, on an open branch, B

If $A,\ +i$ is on B then $v_{w_i}(A) = 1$, and
If $A,\ -i$ is on B then $v_{w_i}(A) = 0$.

The open branch, B, induces an interpretation $< D, W, R, v >$ that is defined as follows. Let C be the set of all constants on the branch. $W = \{w_i :\ i$ occurs on B$\}$. w_iRw_j if and only if irj occurs on B. $D = \{\delta_a : a \in C\}$. For all constants, a, on B, $v(a) = \delta_a$. For every n-place predicate on B (including $\mathfrak{E}$), $< \delta_{a_1}, ..., \delta_{a_n} >\in vw_i$ if and only if $Pa_1...a_n,\ +i$ is on B. $D_{w_i} = v(w_i) = v_{w_i}(\mathfrak{E}) = \{\delta_a : \mathfrak{E}a,\ i$ occurs on B$\}$. This means $\delta_a \in D_{w_i}$ if and only if $\mathfrak{E}a,\ i$ is on the branch.

The argument is as in 20.9.8. There are only two differences: the plus case for Universal Instantiation and minus case for Existential Instantiation.

For the first of these, we have the following: suppose that $\forall xA,\ +i$ is on the branch. We must show that $\forall xA$ is true at w_i in the induced interpretation. This means that for all j such that irj $d \in D_{w_j}$ $A_x(k_d)$ is true at w_j. Suppose that $d \in D_{wj}$. Let c denote d. Because $d \in D_{w_j}$ this means that $\mathfrak{E}c,\ +j$, is on B. So, we have applied the modified universal instantiation rule and $A_x(c),\ +j$, is on B. By Induction Hypothesis, $A_x(c)$ is true at w_j and $A_x(k_d)$ is true at w_j by the Denotation Lemma. Thus $v_{w_i}(\forall xA) = 1$.

For the second case: suppose that $\exists xA,\ -i$ is on the branch. We must show that $\exists xA$ is false at w_i in the induced interpretation. This means that for every c such that $d \in D_{w_i}$ $A_x(k_d)$ is false at w_i. Suppose that $d \in D_{w_i}$. Let c denote d. Because $d \in D_{w_i}$ this means that $\mathfrak{E}c,\ +i$, is on B. So, we have applied the modified particular instantiation rule and $A_x(c),\ -i$, is on B. By Induction Hypothesis, $A_x(c)$ is false at w_i and $A_x(k_d)$ is false at w_i by the Denotation Lemma. Thus $v_{w_i}(\exists xA) = 0$.

The tableaux of Intuitionist Logic are sound and complete with respect to their semantics with the proposed True Universal Instantiation and False Particular Instantiation rules replacing Priest's True Universal Instantiation and False Particular Instantiation rules.

5. Concluding Remarks

In the 12 examples provided, the proposed rule reduced the total lines of proofs by 6 and the total number of branches by 18. With Priest's rule, it was often the case that a left branch was created only to be quickly closed. This is remedied by applying the proposed rule instead: the universal quantifier is instantiated only if an existent object is present in the relevant world, if applicable.

To conclude I will point out a way in which the proposed rule is similar in principle to a more familiar rule. In Priest's Universal Instantiation Rule there is a disanalogy between $\Box Px$ and $\forall x$; this analogy is restored with the proposed rules. There are a number of parallels between the rules for $\Box Px$ and $\Diamond Px$ and for $\forall x$ and $\exists x$, respectively. For $\Diamond Px$ and $\exists x$ something new is introduced when the rule is applied: if $\Diamond Px$ is on the branch, we introduce a world; if $\exists x$ is on the branch, we introduce a constant. If $\Box Px$ is on the branch, we are not at liberty to introduce a world at which this holds because $\Box Px$ is "trivially" true when there is no world; the rule is applied only to worlds already present on the branch. In Priest's rule for $\forall xPx$, we were at liberty to introduce a constant which made Px true even when there was no existing constant present on the branch. Now, for $\Box Px$ and $\forall x$, application of the rule is limited to what we already have: we apply $\Box Px$ to every world on the branch; we apply $\forall xPx$ to every constant on the branch. If there is no existent object, $\forall xPx$ is "trivially" true. Requiring an existent constant to be on the branch before application of the $\forall x$ rule is similar in principle to requiring a world to exist before application of the $\Box Px$ rule. For purposes of reading a countermodel off the branch, it is not assumed that there is an accessible world beyond w_0 if it is not explicitly stated on the branch. Similarly, with the proposed rules, it is not assumed that the object in the domain exists if it is not explicitly stated to exist on the branch. In these ways, the similarity in treatment and explanation of $\forall x$ and $\Box Px$ is restored.

The proposed rules can greatly simplify tableaux and could be used beneficially in place of the rules for universal instantiation introduced by Priest in the chapters on Free Logic and Variable Domain Modal Logic, and for truth of universal instantiation and falsity of particular instantiation in the chapter on Intuitionist Logic.

Acknowledgements

Thanks to Graham Priest for encouragement and guidance on this idea from its inception to completion, and to Stephen Neale and Nick Pappas for their comments. Thanks also to Ed Mares at AJL for his efficient work preparing the paper for its final published format. A version of this paper was presented at the Saul Kripke Center at the City University of New York in December 2013.

Reference

Priest, Graham. (2008). *An Introduction to Non-Classical Logic: From If to Is* (2nd ed.). Cambridge: Cambridge University Press