

Nested Sequents for Intuitionistic Logics

Melvin Fitting

July 8, 2012

Abstract

Relatively recently nested sequent systems for modal logics have come to be seen as an attractive deep reasoning extension of familiar sequent calculi. In an earlier paper I showed there was a strong connection between modal nested sequents and modal prefixed tableaux. In this paper I show the connection continues to intuitionistic logic, both standard and constant domain, relating nested intuitionistic sequent calculi to intuitionistic prefixed tableaux. Modal nested sequent machinery generalizes one sided sequent calculi—intuitionistic nested sequents similarly generalize two sided sequents. It is noteworthy that the resulting system for constant domain intuitionistic logic is particularly simple. It amounts to a combination of intuitionistic propositional rules and classical quantifier rules, a combination that is known to be inadequate when conventional intuitionistic sequent systems are used.

Keywords: intuitionistic logic, constant domain logic, nested sequent, prefixed tableau

Classification Codes: 03B20, 03B44, 03B60, 68T15

1 Introduction

There has been much interest in deep reasoning systems. Recently a simple variety of such calculi has become rather popular. In 2009 Brünnler introduced *nested sequent* systems, [1, 2], and at the same time Poggiolesi introduced *tree hypersequents*, [18]. While they use different syntax, they are essentially the same, and are independent reinventions of the nested sequents of Kashima, [14], from 1994. These are one-sided sequent systems, but in which sequents can appear nested within sequents, and rules can be applied at any depth. Some modal logics that lack conventional sequent calculi have natural nested sequent calculi. Strong connections have been shown between nested sequents and prefixed tableaux, [10]. Prefixed tableaux themselves go back to 1972, [6]. Essentially, nested sequents are to prefixed tableau systems as conventional sequents are to conventional tableau systems,

The nesting paradigm has been extended to intuitionistic logic using two-sided sequents rather than one-sided as in the modal case. A version was introduced for propositional bi-intuitionistic logic, *BiInt*, in 2008, [11]. *BiInt* is an extension of intuitionistic logic with a connective *A excludes B*, whose Kripke semantics involves the converse of the standard reflexive and transitive accessibility relation of intuitionistic logic. This, of course, provides a nested sequent calculus for propositional intuitionistic logic too. An explicit cut-elimination procedure is given in [11], and issues of contraction are carefully considered. In 2009 these ideas were further refined, [19], and an implementation was given, <http://users.cecs.anu.edu.au/linda/>.

In 1983 a prefixed tableau system for propositional intuitionistic logic was sketched in [7, Chapter Nine, Section 5]. This converts to nested sequents much as happens in the modal setting, and the resulting rules are essentially those of [19]. Here we present both nested sequent rules and prefixed tableau rules, prove soundness and completeness directly for prefixed tableaux, and derive it for the nested sequent version via a translation procedure. Of course this gives a non-constructive proof of cut elimination—as noted above, a constructive version has already been shown in [11]—but the non-constructive version is easy to follow and quite intuitive.

We build on this by adding quantifier rules. As it turns out, the simplest first-order extension does not give intuitionistic logic proper, but the *constant domain* version of it. Constant domain intuitionistic logic was introduced by Grzegorzcyk in [13], axiomatized independently in [16] and in [12], and given a rather complex sequent calculus formulation in [15], with further developments in [4]. Constant domain

intuitionistic logic is an interesting logic, and the proof procedures we give here are remarkably simple and straightforward. The constant domain system, either in prefix tableau form or in nested sequent form, is probably the main contribution of this paper. After considering constant domain intuitionistic logic, we then sketch how to modify the formalism to get standard first-order intuitionistic logic.

I want to thank an anonymous referee for information about the history of nested sequent systems. The subject has had a much more complex development than I was previously aware of.

2 Syntax

The language of first-order intuitionistic logic is standard. Atomic formulas are relation symbols applied to the appropriate number of variables—we do not allow constant or function symbols here, though doing so would be straightforward. Then formulas are built up using \wedge , \vee , \neg , \supset , \forall and \exists in the usual way. We do not take \perp as primitive, though it would be simple to do so, making \neg defined. We begin our treatment with *propositional* intuitionistic logic, where we ignore the role of quantifiers.

Sequents, in the conventional sense, can be defined using sets, multisets, or sequences of formulas. Different choices depend on details of the intended applications. Simplicity appropriate to the present paper suggests the use of sets so that structural rules are not needed, or considered. Then a sequent is $\Gamma \rightarrow \Delta$, where Γ and Δ are sets of formulas. (Actually this is a bit too informal, since putting an arrow between sets of formulas is meaningless. Think of $\Gamma \rightarrow \Delta$ as an ordered pair of sets of formulas, with the arrow denoting a pairing function.) But sequents are not quite the subject here, instead it is the following generalization.

Definition 2.1 Nested sequents are defined recursively. A *nested sequent* is $\Gamma \rightarrow \Delta$ where Γ is a set of formulas and Δ is a set of formulas and nested sequents.

Note that nesting is only on the right side of the arrow. Some special notation, common in the nested sequent community, will be useful here. If $\Gamma \rightarrow \Delta$ occurs within another nested sequent we will write it as $[\Gamma \rightarrow \Delta]$ and refer to it as a *boxed sequent*. As is customary with sequents formulated using sets, the curly brackets for the set on the left and the set on the right of the arrow will be omitted. Here is an example:

$$A \rightarrow B, [C \rightarrow D, [E \rightarrow F]], [G \rightarrow H, [I \rightarrow J]]$$

is a nested sequent, where the letters stand for formulas. Written entirely in standard set notation (with arrow as a pairing function), this is the following.

$$\{A\} \rightarrow \{B, \{C\} \rightarrow \{D, \{E\} \rightarrow \{F\}\}, \{G\} \rightarrow \{H, \{I\} \rightarrow \{J\}\}\}$$

The set notation version is ‘official,’ and a translation procedure in Section 7 will be defined using it. The abbreviated notation with square brackets will be used whenever possible, in the interest of readability.

One final point—we do not allow the empty sequent, $\emptyset \rightarrow \emptyset$. It could be worked in, but it does not correspond to any prefixed tableau configuration and so is best omitted here.

There is an informal BHK (Brouwer, Heyting, Kolmogorov) style reading for nested sequents. Let us say we have a proof of a nested sequent $\Gamma \rightarrow \Delta$ if we have an algorithm that takes as input proofs of all members of Γ and outputs a proof of some member of Δ . For example, a proof of $A, B \rightarrow C, [D, E \rightarrow F]$ is an algorithm that converts proofs of A and B either into a proof of C , or into a proof of $D, E \rightarrow F$, which would be an algorithm that converts proofs of D and E into a proof of F .

3 Propositional Intuitionistic Nested Sequent Rules

Quantifier rules are postponed until Section 8. The essence of the present work is already apparent at the propositional level. In stating our formal system we use a simple subscript convention: $\Gamma_2, \Delta_2, \dots$ are sets of formulas and boxed sequents, and $\Gamma_1, \Delta_1, \dots$ are sets of formulas only—no boxed sequents. Thus set subscripts are either 1 or 2, and 1 means no boxed sequents are present. Suppose A is a single formula or a boxed sequent, and Σ is a set of formulas and boxed sequents. We write Σ, A as a more readable version of $\Sigma \cup \{A\}$. We do not assume $A \notin \Sigma$ —we are working with sets and do not track contraction applications here.

Axioms An *axiom* is defined recursively, as follows. First, $\Gamma_1, X \rightarrow \Gamma_2, X$ is an axiom, where X is a formula. (It is enough to restrict X to being atomic.) Second, $\Gamma_1 \rightarrow \Gamma_2, [S]$ is an axiom if S is an axiom.

For example, $A \rightarrow [B \rightarrow [C, D \rightarrow C, E]]$ is an axiom. Next are the rules of inference. The intention is that these rules apply not only at the top level, but also to any nested sequent below the top level as well. We begin with the *ordinary* rules. These are the same as classical, except for allowing the presence of boxed sequents on the right of arrows and, most importantly, being applicable at arbitrary nesting depth. That is, they may be applied to any nested sequent no matter how deep the nesting.

Ordinary Sequent Rules

$$\begin{array}{lcl}
L\wedge & \frac{\Gamma_1, A, B \rightarrow \Gamma_2}{\Gamma_1, A \wedge B \rightarrow \Gamma_2} & R\wedge \quad \frac{\Gamma_1 \rightarrow \Gamma_2, A \quad \Gamma_1 \rightarrow \Gamma_2, B}{\Gamma_1 \rightarrow \Gamma_2, A \wedge B} \\
L\vee & \frac{\Gamma_1, A \rightarrow \Gamma_2 \quad \Gamma_1, B \rightarrow \Gamma_2}{\Gamma_1, A \vee B \rightarrow \Gamma_2} & R\vee \quad \frac{\Gamma_1 \rightarrow \Gamma_2, A, B}{\Gamma_1 \rightarrow \Gamma_2, A \vee B} \\
L\neg & \frac{\Gamma_1 \rightarrow \Gamma_2, A}{\Gamma_1, \neg A \rightarrow \Gamma_2} & L\supset \quad \frac{\Gamma_1 \rightarrow \Gamma_2, A \quad \Gamma_1, B \rightarrow \Gamma_2}{\Gamma_1, A \supset B \rightarrow \Gamma_2}
\end{array}$$

Next we have the *special* rules, explicitly involving boxed sequents.

Special Sequent Rules

$$\begin{array}{lcl}
R\neg & \frac{\Gamma_1 \rightarrow \Gamma_2, [A \rightarrow]}{\Gamma_1 \rightarrow \Gamma_2, \neg A} & R\supset \quad \frac{\Gamma_1 \rightarrow \Gamma_2, [A \rightarrow B]}{\Gamma_1 \rightarrow \Gamma_2, A \supset B} \\
Lift & \frac{\Gamma_1 \rightarrow \Gamma_2, [\Delta_1, A \rightarrow \Delta_2]}{\Gamma_1, A \rightarrow \Gamma_2, [\Delta_1 \rightarrow \Delta_2]} &
\end{array}$$

Some comments about the special rules. Given the intuitive BHK reading of proof for nested sequents, the $R\supset$ rule simply reflects the usual understanding of implication, and similarly for $R\neg$. Something related to the $R\supset$ rule occurs in [20], where nested occurrences of \vdash are briefly considered. A rule stated there infers $\Gamma \vdash A \supset B$ from $\Gamma \vdash (A \vdash B)$, but this nesting is not pursued further in the paper. In particular, nesting of nested occurrences is not considered. A paper of Došen, [3], considers nesting of the consequence relation to arbitrary depths, but the exact relationship between this and nested sequents remains to be determined. The $R\supset$ rule as we gave it appears in [19] modulo some differences in notation.

The rule called *Lift* is somewhat akin to the *s-m-n* theorem. Suppose the sequent above the line has been verified, that is we have an algorithm, call it \mathcal{A} , that converts proofs of the members of Γ_1 into a proof of some member of Γ_2 or else into a proof of $\Delta_1, A \rightarrow \Delta_2$. We modify that algorithm into one we call \mathcal{A}' , to verify the sequent below the line. Here is the description of \mathcal{A}' . Suppose we are given proofs of the members of Γ_1 , along with a proof of A . Since we have proofs of the members of Γ_1 , we can feed these to \mathcal{A} . If the output is a proof of some member of Γ_2 , that is taken to be the output of \mathcal{A}' . Otherwise the output must be an algorithm verifying $\Delta_1, A \rightarrow \Delta_2$. Modify *that* algorithm by supplying it with the given proof of A , thus converting it into an algorithm that only needs as inputs proofs of the members of Δ_1 . This algorithm verifies $\Delta_1 \rightarrow \Delta_2$, and is taken to be the output of \mathcal{A}' .

Example 3.1 Here is a proof in this system, of the following formula.

$$(\neg\neg A \wedge \neg\neg(B \wedge D)) \supset \neg(\neg(A \vee C) \vee \neg B)$$

Reasons for each step are given within the displayed structure. Upper leaves are axioms.

$$\begin{array}{c}
\frac{\frac{\rightarrow [\rightarrow [\neg\neg(B \wedge D), [A \rightarrow A, C]]]}{\rightarrow [\rightarrow [\neg\neg(B \wedge D), [A \rightarrow A \vee C]]]} R\vee}{\rightarrow [\rightarrow [\neg\neg(B \wedge D), [\neg(A \vee C), A \rightarrow]]]} L\neg \\
\frac{\frac{\rightarrow [\rightarrow [\neg\neg(B \wedge D), \neg(A \vee C) \rightarrow [A \rightarrow]]]}{\rightarrow [\rightarrow [\neg\neg(B \wedge D), \neg(A \vee C) \rightarrow \neg A]]} R\neg \\
\frac{\rightarrow [\rightarrow [\neg\neg(B \wedge D), \neg(A \vee C) \rightarrow]}{\rightarrow [\rightarrow [\neg\neg A, \neg\neg(B \wedge D), \neg(A \vee C) \rightarrow]} L\neg \\
\frac{\frac{\frac{\frac{\rightarrow [\rightarrow [\neg\neg A, \neg\neg(B \wedge D), \neg(A \vee C) \vee \neg B \rightarrow]}{\rightarrow [\rightarrow [\neg\neg A \wedge \neg\neg(B \wedge D), \neg(A \vee C) \vee \neg B \rightarrow]} L\wedge}{\rightarrow [\neg\neg A \wedge \neg\neg(B \wedge D) \rightarrow [\neg(A \vee C) \vee \neg B \rightarrow]} Lift \\
\frac{\rightarrow [\neg\neg A \wedge \neg\neg(B \wedge D) \rightarrow \neg(\neg(A \vee C) \vee \neg B)]}{\rightarrow (\neg\neg A \wedge \neg\neg(B \wedge D)) \supset \neg(\neg(A \vee C) \vee \neg B)} R\neg \\
\frac{\rightarrow (\neg\neg A \wedge \neg\neg(B \wedge D)) \supset \neg(\neg(A \vee C) \vee \neg B)}{R\supset}
\end{array}$$

4 Sequent Soundness

We assume Kripke intuitionistic models are familiar items, and do not give a definition here. Soundness of the system with respect to Kripke models may be shown as follows. Let us say $\Gamma \rightarrow \Delta$ is true at a state of a Kripke model if one of the members of Γ is not forced at that state, or one of the members of Δ is. Equivalently, $\Gamma \rightarrow \Delta$ is true at a state provided, if all members of Γ are forced there, so is some member of Δ . And let us say a boxed sequent, $[S]$, is true at a state provided the sequent S is true at that state *and at every state accessible from it*.

It is easy to check that each of the rules, regular and special, preserves not just validity but truth at a state. Note that verifying this for *Lift* makes use of the fact that, in a Kripke intuitionistic model, any formula forced at a state is also forced at all accessible states. Since the axioms are valid sequents, it follows that only validities are provable.

5 Propositional Prefixed Intuitionistic Tableaus

In [10] it was shown that, for a variety of modal logics, *nested* sequent systems and *prefixed* tableaus are equivalent in the same way that Gentzen sequent systems and standard tableau systems are equivalent—one is the other ‘upsidedown.’ Nested sequent modal systems are described in [1, 2], and with different notation and terminology in [18]. Prefixed tableaus for modal logics are discussed in [9]. There is a similar relationship between the nested sequent system for intuitionistic logic given above, and a prefixed tableau system for intuitionistic logic. Such a prefixed tableau system was never published in detail, but there is a brief sketch of it in [7, Chapter Nine, Section 5]. Since completeness for the tableau system is rather easy to show, we describe the tableau version in some detail, sketch a completeness argument for it, and show how this implies completeness for the nested sequent system.

Occurring in prefixed tableau proofs are *prefixed*, *signed* formulas. Prefixed formulas originated in [5] for a natural deduction system. Think of a prefix as a name for a possible world or state; the syntax of these names will provide us with a representation of the accessibility relation for a Kripke intuitionistic model.

A *prefix* is a non-empty finite sequence of positive integers starting with 1, such as 1.3.2.1.4, which we write using periods as separators. If n is a positive integer and σ is a prefix, by $\sigma.n$ we mean the result of adjoining n to the end of σ . Think of prefixes as representing possible stages in the work of a mathematician—an intuitionistic mathematician say. Think of the actual world as 1; and $\sigma.n$ as a possible next stage to σ for the mathematician, it is one of the states that could immediately succeed σ , depending on what work the mathematician chooses to carry out.

A *signed formula* is TX or FX , where X is a formula. Think of TX as asserting that X has been proved, and FX as asserting that X has not yet been proved—an intuitionistic version of truth. A *prefixed signed formula* is of the form σTX or σFX where σ is a prefix and X is a formula. Think of σTX as saying X has been proved at state σ , and σFX as saying that X has not been proved at state σ . Of course

all this is quite informal, but it will serve as a guide. This intuition plays no direct role in the completeness argument of Section 6.

A tableau proof is a tree (written with the root at the top) meeting certain conditions, in which each node is labeled with a prefixed signed formula. We give rules for starting, continuing, and terminating a proof construction. The intuitive idea is that to prove X we suppose there is some state, call it 1, where X has not been proved, we derive a contradiction, and we conclude X . Informally if X is, in fact, provable by an intuitionistically oriented mathematician, at some point a proof may be discovered. Since we could take 1 to be that time point, we would have an impossible situation.

Formally, a tableau proof of X begins with the trivial tree with only a root node, labeled $1 \ F X$. Next we have the *branch extension rules*, for continuing the tableau construction. Just as with nested sequents, these divide into ordinary and special cases.

Ordinary Tableau Rules

$$\begin{array}{c}
 T\wedge \quad \frac{\sigma T A \wedge B}{\sigma T A} \\
 \sigma T B
 \end{array}
 \qquad
 \begin{array}{c}
 F\wedge \quad \frac{\sigma F A \wedge B}{\sigma F A \mid \sigma F B}
 \end{array}$$

$$\begin{array}{c}
 T\vee \quad \frac{\sigma T A \vee B}{\sigma T A \mid \sigma T B}
 \end{array}
 \qquad
 \begin{array}{c}
 F\vee \quad \frac{\sigma F A \vee B}{\sigma F A} \\
 \sigma F B
 \end{array}$$

$$\begin{array}{c}
 T\neg \quad \frac{\sigma T \neg A}{\sigma F A}
 \end{array}
 \qquad
 \begin{array}{c}
 T\supset \quad \frac{\sigma T A \supset B}{\sigma F A \mid \sigma T B}
 \end{array}$$

And then there are the special tableau rules.

Special Tableau Rules

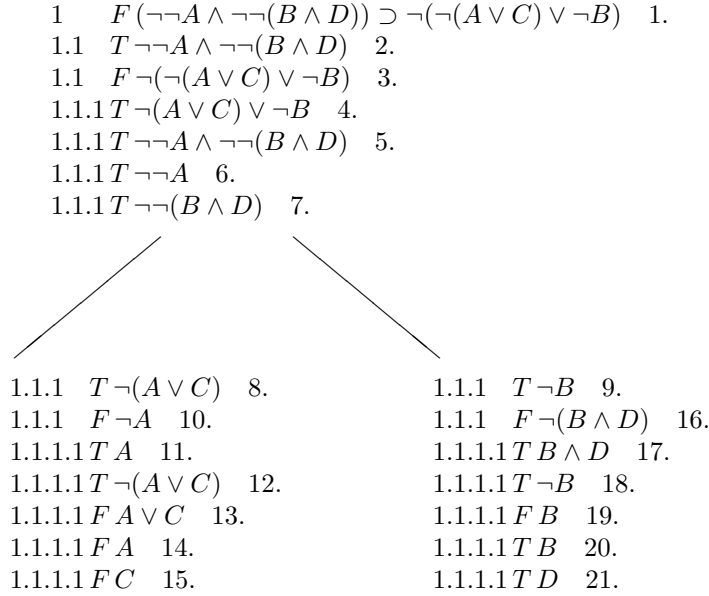
$$\begin{array}{c}
 F\neg \quad \frac{\sigma F \neg A}{\sigma.n T A} \\
 \sigma.n \text{ new}
 \end{array}
 \qquad
 \begin{array}{c}
 F\supset \quad \frac{\sigma F A \supset B}{\sigma.n T A} \\
 \sigma.n F B \\
 \sigma.n \text{ new}
 \end{array}$$

$$\begin{array}{c}
 \textit{Lower} \quad \frac{\sigma T A}{\sigma.n T A} \\
 \sigma.n \text{ not new}
 \end{array}$$

A prefix $\sigma.n$ is new if it does not occur as an initial segment, proper or not, of any prefix on the branch where the rule is being applied. Similarly, $\sigma.n$ is not new if it does already occur on the branch. Intuitive motivation is quite direct. For example, if $F A \supset B$ is the case at the world named by σ in a Kripke model, we understand that $A \supset B$ is not forced at that world. Then by the usual workings of Kripke models there must be a world accessible from that world, at which A is forced but B is not, that is, at which we have $T A$ and $F B$. We can assign this world a prefix name; $\sigma.n$ appropriately reflects the accessibility relation, but the name must be otherwise uncommitted, hence the newness condition in the $F \supset$ rule. Similar considerations apply to the other two special rules. *Lower* corresponds to *Lift* for the sequent system—it embodies the idea that formulas forced at a state are forced at all accessible states.

A *tableau branch* is *closed* if it contains both $\sigma T X$ and $\sigma F X$ for some formula X . A *tableau* is *closed* if each branch is closed. A closed tableau that starts with $1 \ F X$ is a *proof* of X .

Example 5.1 Here is a proof in the tableau system, of the same formula as in Example 3.1, $(\neg\neg A \wedge \neg\neg(B \wedge D)) \supset \neg(\neg(A \vee C) \vee \neg B)$.



Reasons for each step are as follows; formulas are numbered for this purpose. 2 and 3 are from 1 by $F \supset$ (note that 1.1 is new); 4 is from 3 by $F \neg$, again 1.1.1 is new; 5 is from 2 by *Lower*; 6 and 7 are from 5 by $T \wedge$; 8 and 9 are from 4 by $T \vee$; 10 is from 6 by $T \neg$; 11 is from 10 by $F \neg$; 12 is from 8 by *Lower*; 13 is from 12 by $T \neg$; 14 and 15 are from 13 by $F \vee$; 16 is from 7 by $T \neg$; 17 is from 16 by $F \neg$; 18 is from 9 by *Lower*; 19 is from 18 by $T \neg$; 20 and 21 are from 17 by $T \wedge$. Closure is by 11 and 14, and by 19 and 20.

There are some important restrictions that can be placed on rule usage, without compromising completeness.

Definition 5.2 A tableau branch is *atomically closed* if it contains $\sigma T A$ and $\sigma F A$, where A is atomic; a tableau is atomically closed if each branch is.

A tableau is *single-usage* if no $F \neg$ or $F \supset$ rule is applied to a prefixed signed formula occurrence more than once on a tableau branch.

It is the $F \neg$ and $F \supset$ rules that introduce new prefixes, so requiring single-usage is a restriction on prefix introduction. Imposing atomic closure and single-usage restrictions does not change the class of theorems; the completeness proof in Section 6 will be shown with these restrictions in place, soundness whether or not they are in place.

6 Propositional Tableau Soundness and Completeness

Soundness of the tableau system is by an argument that is standard for tableaux. We briefly sketch the ideas.

Call a set S of prefixed signed formulas *satisfiable* if there is a Kripke intuitionistic model \mathcal{M} and a mapping p from prefixes occurring in S to states in the model, meeting the following conditions. 1) if σ and $\sigma.n$ both occur in S , then $p(\sigma.n)$ is accessible from $p(\sigma)$. 2) if $\sigma T X \in S$ then X is forced at the state $p(\sigma)$. 3) if $\sigma F X \in S$ then X is not forced at the state $p(\sigma)$.

Call a tableau branch satisfiable if the set of prefixed signed formulas on it is satisfiable, and call a tableau satisfiable if one of its branches is. It is not hard to show that if a tableau rule is applied to a satisfiable tableau, the result is another satisfiable tableau.

Now, suppose X is not valid. It follows that $\{1 F X\}$ is a satisfiable set, so any attempt to prove X begins with a satisfiable tableau. Then all subsequent tableaux must be satisfiable. Since a satisfiable tableau cannot be closed, X does not have a proof. The tableau system is sound.

Completeness is more interesting. This can be shown using a systematic proof search method, which also yields propositional decidability. Instead we use a Lindenbaum/Henkin construction, which is easier to describe. It is along the lines of a similar completeness argument for prefixed modal tableaux in [9].

We now allow tableaux to start with a *finite set* of prefixed signed formulas instead of with a single one. Call a set S of prefixed signed formulas *consistent* if no tableau for any finite subset of S is closed, *where closure must be atomic and tableau construction is single-usage*. Call S *prefix complete* provided that first, $\sigma F A \supset B \in S$ implies that $\sigma.k T A, \sigma.k F B \in S$ for some integer k and second, $\sigma F \neg A \in S$ implies that $\sigma.k T A \in S$ for some integer k . (Prefix completeness is analogous to the existence of Henkin witnesses for existential statements in the common proof of completeness for first-order classical logic.) Every finite consistent set S of prefixed signed formulas can be extended to a set that is *maximally consistent* and *prefix complete*. This is the outcome of the following construction.

Propositional Extension Construction Suppose S is a finite consistent set of prefixed signed formulas.

Enumerate the (countably many) prefixed signed formulas of the language, $\sigma_1 \mathcal{X}_1, \sigma_2 \mathcal{X}_2, \dots$, and define the following sequence of sets. $\sigma_n.k$ *new* means the prefix does not occur in S_n .

$$S_0 = S$$

$$S_{n+1} = \begin{cases} S_n \cup \{\sigma_n \mathcal{X}_n\} & \text{if } S_n \cup \{\sigma_n \mathcal{X}_n\} \text{ is consistent} \\ & \text{and } \mathcal{X}_n \text{ is not} \\ & F A \supset B \text{ or } F \neg A \\ S_n \cup \{\sigma_n \mathcal{X}_n, \sigma_n.k T A, \sigma_n.k F B\} & \text{if } S_n \cup \{\sigma_n \mathcal{X}_n\} \text{ is consistent,} \\ & \mathcal{X}_n \text{ is } F A \supset B, \\ & \text{and } \sigma_n.k \text{ is new} \\ S_n \cup \{\sigma_n \mathcal{X}_n, \sigma_n.k T A\} & \text{if } S_n \cup \{\sigma_n \mathcal{X}_n\} \text{ is consistent,} \\ & \mathcal{X}_n \text{ is } F \neg A, \text{ and } \sigma_n.k \text{ is new} \\ S_n & \text{otherwise} \end{cases}$$

Informally, the construction really amounts to this. If $\sigma_n \mathcal{X}_n$ is consistent with S_n , add it, and throw in whatever else is required for prefix completeness, and otherwise $\sigma_n \mathcal{X}_n$ is not consistent with S_n , so don't add it.

In cases 2 and 3 of this construction some integer k is chosen so that $\sigma_n.k$ does not occur in S_n . If S is finite then each S_n will also be finite, hence the 'newness' requirements can always be met. In cases 1 and 4 it is obvious that S_{n+1} is consistent if S_n is, but this is not so clear with cases 2 and 3. We now verify it for case 3; case 2 is similar.

Suppose $S_n \cup \{\sigma_n F \neg A, \sigma_n.k T A\}$ is not consistent, where $\sigma_n.k$ does not occur in S_n . We show that $S_n \cup \{\sigma_n F \neg A\}$ is also not consistent, so case 3 does not apply. By assumption there is an atomically closed single-usage tableau \mathcal{T} for $S_n \cup \{\sigma_n F \neg A, \sigma_n.k T A\}$. Even though the set we are starting with already includes $\sigma_n.k T A$, the $F\neg$ rule still might have been applied to $\sigma_n F \neg A$ on one or more branches in tableau \mathcal{T} . Suppose that on some branch the rule was applied, to add $\sigma_n.m T A$, where $\sigma_n.m$ was new at this point of the tableau construction. In $\sigma_n.m T A$ and in its descendants, replace occurrences of $\sigma_n.m$ with occurrences of $\sigma_n.k$ (this introduces an extra copy of $\sigma_n.k T A$ which can be eliminated). It is not hard to show the resulting tableau is still correctly constructed. In a similar way, substitute away any other applications of the $F\neg$ rule to $\sigma_n F \neg A$ until none are left.

We now have an atomically closed tableau for the set $S_n \cup \{\sigma_n F \neg A, \sigma_n.k T A\}$ in which no tableau rule is applied to $\sigma_n F \neg A$, let us call the tableau \mathcal{T}' . We use it to construct a new tableau for the set $S_n \cup \{\sigma_n F \neg A\}$. Begin the tableau with these formulas, then use the $F\neg$ rule to add $\sigma_n.k T A$, and then continue exactly as in tableau \mathcal{T}' . Since no rule was applied to $\sigma_n F \neg A$ in \mathcal{T} , in the new tableau exactly one rule application is made to it. And otherwise the single-usage restriction is met in the new tableau as well, since it was in \mathcal{T} . We now have an atomically closed tableau for $S_n \cup \{\sigma_n F \neg A\}$, meeting the single-usage restriction, thus demonstrating inconsistency.

Let $M = \cup_n S_n$. It is straightforward that M is *maximally* consistent and prefix complete. This ends the discussion of the Propositional Extension Construction.

A maximally consistent and prefix complete set satisfies what we might call *downward saturation* properties. As one instance, if M is such a set and $\sigma T X \vee Y \in M$, then either $\sigma T X \in M$ or $\sigma T Y \in M$. Here is a sketch of the proof. Suppose $\sigma T X \vee Y \in M$ but $\sigma T X \notin M$ and $\sigma T Y \notin M$. Using maximal consistency, there must be finite subsets M_1, M_2 of M so that $M_1 \cup \{\sigma T X\}$ and $M_2 \cup \{\sigma T Y\}$ are inconsistent. Let $M_0 = M_1 \cup M_2 \cup \{\sigma T X \vee Y\} \subseteq M$. Then both $M_0 \cup \{\sigma T X\}$ and $M_0 \cup \{\sigma T Y\}$ are inconsistent, and so have closed tableaux. We use this to show M_0 has a closed tableau and hence M is inconsistent. Start a tableau with M_0 . Since M_0 contains $\sigma T X \vee Y$, immediately branch to $\sigma T X$ and $\sigma T Y$. On the branch containing $\sigma T X$ carry out the steps of the closed tableau for $M_0 \cup \{\sigma T X\}$, and on the branch containing $\sigma T Y$ carry out the steps of the closed tableau for $M_0 \cup \{\sigma T Y\}$. In this way we get a closed tableau for M_0 , contradicting the consistency of M .

By arguments like this one can show the following.

Lemma 6.1 (Downward Saturation) *Let M be a maximally consistent and prefix complete set. The following holds.*

1. $\sigma T X \wedge Y \in M \Rightarrow \sigma T X \in M$ and $\sigma T Y \in M$
2. $\sigma F X \wedge Y \in M \Rightarrow \sigma F X \in M$ or $\sigma F Y \in M$
3. $\sigma T X \vee Y \in M \Rightarrow \sigma T X \in M$ or $\sigma T Y \in M$
4. $\sigma F X \vee Y \in M \Rightarrow \sigma F X \in M$ and $\sigma F Y \in M$
5. $\sigma T X \supset Y \in M \Rightarrow \sigma F X \in M$ or $\sigma T Y \in M$
6. $\sigma F X \supset Y \in M \Rightarrow \sigma.k T X \in M$ and $\sigma.k F Y \in M$ for some k
7. $\sigma T \neg X \in M \Rightarrow \sigma F X \in M$
8. $\sigma F \neg X \in M \Rightarrow \sigma.k T X \in M$ for some k
9. $\sigma T X \in M \Rightarrow \sigma.n T X \in M$ for every $\sigma.n$ occurring in M

Now suppose X is not provable using the prefixed intuitionistic tableau rules, requiring atomic closure and single-usage. Then there is no closed tableau beginning with $1 F X$, so $\{1 F X\}$ is consistent, and obviously finite. Extend it to a maximally consistent, prefix complete set, M , using the Propositional Extension Construction above. We construct a Kripke intuitionistic model from M . Let the set of states be the set of prefixes that occur in M . For the accessibility relation, suppose τ and σ are states. These are also prefixes, and we set $\tau R \sigma$ if τ is an initial segment of σ , not necessarily proper. For an atomic formula P , let P be true at state σ if $\sigma T P \in M$. This gives us a Kripke intuitionistic model, let us call it \mathcal{M} .

Now a truth lemma is straightforward to prove using the Downward Saturation Lemma 6.1—details are omitted here. It says: for every formula X , the following is true.

$$\begin{aligned} \sigma T X \in M &\implies X \text{ is forced at world } \sigma \text{ in the model } \mathcal{M} \\ \sigma F X \in M &\implies X \text{ is not forced at world } \sigma \text{ in the model } \mathcal{M} \end{aligned}$$

Once we have this completeness is immediate. M extends $\{1 F X\}$ and so X is false at world 1 of the model \mathcal{M} .

7 Prefixed Tableaux to Nested Sequents

Tableau completeness has been proved, or at least sketched. Now completeness for the intuitionistic propositional nested sequent system follows once it has been shown that every tableau proof converts to a nested sequent proof. (Conversion in the other direction is also possible, but it is harder to describe and we omit it here.) For the conversion details we adapt the machinery of [10], where a similar result for modal logics was

shown. The ideas are simple. Formulas signed with T go on the left of an arrow, formulas signed with F go on the right. Signed formulas with the same prefix go together in a single nested sequent. Signed formulas prefixed with 1 become members of the ‘top level’ nested sequent. And otherwise, the nested sequent of formulas that were originally prefixed with $\sigma.n$ should appear as a boxed subsequent of the nested sequent of formulas that were originally prefixed with σ . In the formal statement of the translation we allow the empty prefix, and identify a signed formula having an empty prefix with the signed formula itself. In our tableau proofs all prefixed signed formulas have prefixes that start with 1, something that is assumed in the overall translation process, but during the course of the translation prefixes that start with something other than 1 are allowed. In the following we write \mathcal{X} for an arbitrary *signed* formula.

Definition 7.1 Let S be a finite set of prefixed signed formulas, allowing the empty prefix and prefixes starting with an integer that is not 1.

1. For each positive integer n let $S^n = \{\sigma \mathcal{X} \mid n.\sigma \mathcal{X} \in S\}$, where $n.\sigma$ is the prefix σ with n added at the beginning.
2. Let S° be the sequent $\{X \mid TX \in S\} \rightarrow \{Y \mid FY \in S\} \cup \{(S^n)^\circ \mid S^n \neq \emptyset\}$
(The condition $TX \in S$ implies that TX has the empty prefix; similarly for $FY \in S$.)

Now let S be a set of prefixed signed formulas where no prefix is empty and all prefixes start with 1. The *nested sequent translate* of S is $\mathcal{N}(S) =_{df} (S^1)^\circ$.

Note that in item 2 of the translation, on the left of the arrow only formulas appear, while on the right there are formulas and also sets of the form $(S^n)^\circ$. These sets are themselves nested sequents, so all nesting is thus on the right of the arrow.

Example 7.2 We show the conversion of the following set of prefixed signed formulas into a nested sequent.

$$S = \{1TA, 1FB, 1.1TC, 1.1FD, 1.1.1TE, 1.1.1FF, \\ 1.2TG, 1.2FH, 1.2.1TI, 1.2.1FJ\}$$

The conversion is as follows.

$$\begin{aligned} S^1 &= \{TA, FB, 1TC, 1FD, 1.1TE, 1.1FF, \\ &\quad 2TG, 2FH, 2.1TI, 2.1FJ\} \\ (S^1)^\circ &= \{A\} \rightarrow \{B, \{TC, FD, 1TE, 1FF\}^\circ, \\ &\quad \{TG, FH, 1TI, 1FJ\}^\circ\} \\ \{TC, FD, 1TE, 1FF\}^\circ &= \{C\} \rightarrow \{D, \{TE, FF\}^\circ\} \\ &= \{C\} \rightarrow \{D, \{E\} \rightarrow \{F\}\} \\ \{TG, FH, 1TI, 1FJ\}^\circ &= \{G\} \rightarrow \{H, \{TI, FJ\}^\circ\} \\ &= \{G\} \rightarrow \{H, \{I\} \rightarrow \{J\}\} \\ &\text{and so} \\ \mathcal{N}(S) = (S^1)^\circ &= \{A\} \rightarrow \{B, \{C\} \rightarrow \{D, \{E\} \rightarrow \{F\}\}, \\ &\quad \{G\} \rightarrow \{H, \{I\} \rightarrow \{J\}\}\} \end{aligned}$$

Written using boxed sequent notation, $\mathcal{N}(S)$ is the following.

$$A \rightarrow B, [C \rightarrow D, [E \rightarrow F]], [G \rightarrow H, [I \rightarrow J]]$$

This is the example nested sequent given in Section 2.

We have shown how to convert a finite set of prefixed signed formulas into a nested sequent. If we identify a tableau branch with the set of prefixed signed formulas on it, then a tableau branch converts to a nested sequent. If we identify a tableau with the set of its branches, an entire tableau converts to a set of

nested sequents. Going further, if we think of the process of constructing a tableau proof as the creation of a succession of tableaux, until we arrive at one that is closed, then such a succession of tableaux converts into a succession of sets of nested sequents. It is easy to see that a closed tableau branch converts to a nested sequent that is an axiom; hence a closed tableau converts to a set of nested sequent axioms. Finally, if a succession of nested sequents arising from the construction of a tableau proof is ‘turned over,’ the result will be a proof in the nested sequent intuitionistic calculus. Once this is checked, completeness of the nested sequent intuitionistic system follows from the completeness of the prefixed tableau calculus, which was shown in Section 6.

Basically, it is enough to show that each one of the tableau rules turns into a nested sequent rule under the convert-and-turn-over process described above. We show this for two representative rules and omit the rest, and our discussion for these cases is, essentially, just a sketch. We have chosen the two implication rules.

Representative Case, $T \supset$ The tableau rule is:

$$\frac{\sigma T A \supset B}{\sigma F A \mid \sigma T B}$$

All signed formulas mentioned in the rule have the same prefix, so under the tableau translation described above they all map to the same nested context. Suppose we translate the tableau branch before the rule is applied, and we just consider the nested sequent corresponding to formulas whose prefix begins with (or is) σ . We will get something of the form $\Gamma_1, A \supset B \rightarrow \Gamma_2$, where Γ_1 consists of formulas and Γ_2 consists of formulas and nested sequents. Likewise the translates of the branches after the tableau rule has been applied are: $\Gamma_1, A \supset B \rightarrow \Gamma_2, A$ and $\Gamma_1, A \supset B, B \rightarrow \Gamma_2$. When the rule application is ‘turned over,’ this becomes rule $L \supset$, one of the ordinary sequent rules. (Recall we discussed earlier that a formula like $A \supset B$ is allowed to be a member of Γ_1 , so displaying it as we have here, or not displaying it as we did when stating the rule makes no significant difference.)

Representative Case, $F \supset$ The tableau rule is:

$$\frac{\sigma F A \supset B}{\begin{array}{l} \sigma.n T A \\ \sigma.n F B \\ \sigma.n \text{ new} \end{array}}$$

As in the previous case, we translate the tableau branch before the rule is applied and consider only the nested sequent corresponding to formulas whose prefix begins with σ . We get something of the form $\Gamma_1 \rightarrow \Gamma_2, A \supset B$. When translating the tableau branch after the rule is applied, recall that formulas prefixed with $\sigma.n$ will appear in a boxed sequent nested inside the sequent corresponding to formulas prefixed with σ . Further, since $\sigma.n$ was required to be new *only* A and B can occur in the nested boxed subsequent. Carrying out the translation we get $\Gamma_1 \rightarrow \Gamma_2, A \supset B, [A \rightarrow B]$. Now, reversing the roles of rule premise and rule conclusion we get

$$\frac{\Gamma_1 \rightarrow \Gamma_2, A \supset B, [A \rightarrow B]}{\Gamma_1 \rightarrow \Gamma_2, A \supset B}$$

and this is special sequent rule $R \supset$ (understanding $A \supset B$ to be a member of Γ_2).

Each tableau rule converts to a sequent rule, hence tableau proofs convert to sequent proofs, and we have completeness of the sequent calculus.

As an illustration, the tableau proof in Example 5.1 converts into the sequent proof in Example 3.1.

8 Adding Quantifiers, Constant Domains

We have been discussing proof systems for propositional intuitionistic logic. Adding quantifiers to this is easy, and can be done so that two different logics result. Before getting into details, we note that we follow

a common convention and extend the language for the purpose of creating proofs. More specifically, when proving a first-order formula X we extend the language by adding an infinite set of new free variables, called *parameters*, which do not appear quantified. Parameters occur in proofs, but not in what is being proved. This avoids annoying problems with inadvertent capture of variables by quantifiers during an instantiation.

We begin, not with first-order intuitionistic logic, but with *constant domain* intuitionistic logic, which is captured quite easily using the present machinery. Semantically this logic is characterized by Kripke first-order intuitionistic models in which the domain of quantification is constant across all possible worlds. The logic was introduced by Grzegorzczuk in [13]. It is axiomatized by adding to a standard axiomatization of intuitionistic first-order logic the axiom schema $(\forall x)(A \vee B(x)) \supset (A \vee (\forall x)B(x))$, where x does not occur free in A . This axiomatization was proved complete independently in [16] and in [12]. There is a sequent calculus for the logic in [15], but it is somewhat complex; also see [4]. Producing a simple sequent calculus or a tableau calculus for the logic has been a mildly nagging problem for some time. See [8] for some personal history concerning this point.

If we take a standard sequent calculus for propositional intuitionistic logic, say the one of Beth, and add *classical* quantifier rules, we get a system that is sound for constant domain intuitionistic logic. But it is not complete or, putting it differently, cut elimination does not hold. As a simple example, $(\forall x)(A \vee B(x)) \supset (A \vee (C \supset (\forall x)B(x)))$ is valid in all constant domain models, but a simple examination of all cases shows it will not be provable in an intuitionistic propositional + classical quantifier sequent calculus.

Rather remarkably, adding classical quantifier rules to either our prefixed tableau or our nested sequent system for propositional intuitionistic logic produces a calculus that is sound and complete with respect to constant domain semantics. We begin with the prefixed tableau version. The following quantifier rules should be added to the propositional tableau rules given in Section 5. In stating the rules, a is a parameter, and it is called *new* if it does not occur on the tableau branch on which the rule is being applied.

Tableau Constant Domain Quantifier Rules

$$\begin{array}{cc}
 T\exists & \frac{\sigma T (\exists x)\varphi(x)}{\sigma T \varphi(a)} \\
 & a \text{ new} \\
 F\exists & \frac{\sigma F (\exists x)\varphi(x)}{\sigma F \varphi(a)} \\
 & \text{any } a \\
 T\forall & \frac{\sigma T (\forall x)\varphi(x)}{\sigma T \varphi(a)} \\
 & \text{any } a \\
 F\forall & \frac{\sigma F (\forall x)\varphi(x)}{\sigma F \varphi(a)} \\
 & a \text{ new}
 \end{array}$$

Next we have the nested sequent version, extending the propositional system of Section 3. As in the propositional case, these rules apply to nested sequents and not just at the top level. In the rules, a is a parameter, and it is said *not to occur in the conclusion* of a rule application if it does not appear in any formula of the conclusion. But keep in mind, the rules as stated are abbreviated. The premise and conclusion explicitly shown may occur nested within a larger sequent—that is, they may appear in a context. It is the *entire* conclusion that must be taken into account, the context as well, and not just the portion shown in the statement of the rule.

Nested Sequent Constant Domain Quantifier Rules

$$\begin{array}{cc}
 L\exists & \frac{\Gamma_1, \varphi(a) \rightarrow \Gamma_2}{\Gamma_1, (\exists x)\varphi(x) \rightarrow \Gamma_2} \\
 & a \text{ not in conclusion} \\
 R\exists & \frac{\Gamma_1 \rightarrow \Gamma_2, \varphi(a)}{\Gamma_1 \rightarrow \Gamma_2, (\exists x)\varphi(x)} \\
 & \text{any } a \\
 L\forall & \frac{\Gamma_1, \varphi(a) \rightarrow \Gamma_2}{\Gamma_1, (\forall x)\varphi(x) \rightarrow \Gamma_2} \\
 & \text{any } a \\
 R\forall & \frac{\Gamma_1 \rightarrow \Gamma_2, \varphi(a)}{\Gamma_1 \rightarrow \Gamma_2, (\forall x)\varphi(x)} \\
 & a \text{ not in conclusion}
 \end{array}$$

It should be clear that the sequent rules are the tableau rules inverted, as described in Section 7. Soundness for both tableaux and nested sequents is easy to show, and we say no more about it. Tableau completeness can be shown by adding ‘quantifier witness’ cases to the Propositional Extension Construction from Section 6. Now the construction should read as follows. In it, parameter a is *new* if it does not occur in S_n or in $\sigma_n \mathcal{X}_n$.

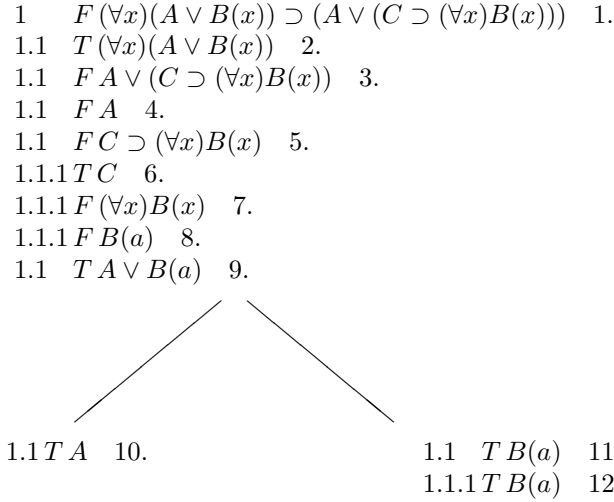
Constant Domain Extension Construction

$$\begin{array}{l}
 S_0 = S \\
 \\
 S_{n+1} = \left\{ \begin{array}{ll}
 S_n \cup \{\sigma_n \mathcal{X}_n\} & \text{if } S_n \cup \{\sigma_n \mathcal{X}_n\} \text{ is consistent} \\
 & \text{and } \mathcal{X}_n \text{ is not} \\
 & F A \supset B \text{ or } F \neg A \\
 \\
 S_n \cup \{\sigma_n \mathcal{X}_n, \sigma_n.k T A, \sigma_n.k F B\} & \text{if } S_n \cup \{\sigma_n \mathcal{X}_n\} \text{ is consistent,} \\
 & \mathcal{X}_n \text{ is } F A \supset B, \\
 & \text{and } \sigma_n.k \text{ is new} \\
 \\
 S_n \cup \{\sigma_n \mathcal{X}_n, \sigma_n.k T A\} & \text{if } S_n \cup \{\sigma_n \mathcal{X}_n\} \text{ is consistent,} \\
 & \mathcal{X}_n \text{ is } F \neg A, \text{ and } \sigma_n.k \text{ is new} \\
 \\
 S_n \cup \{\sigma_n \mathcal{X}_n, \sigma_n T A(a)\} & \text{if } S_n \cup \{\sigma_n \mathcal{X}_n\} \text{ is consistent,} \\
 & \mathcal{X}_n \text{ is } T(\exists x)A(x), \text{ and} \\
 & a \text{ is new} \\
 \\
 S_n \cup \{\sigma_n \mathcal{X}_n, \sigma_n F A(a)\} & \text{if } S_n \cup \{\sigma_n \mathcal{X}_n\} \text{ is consistent,} \\
 & \mathcal{X}_n \text{ is } F(\forall x)A(x), \text{ and} \\
 & a \text{ is new} \\
 \\
 S_n & \text{otherwise}
 \end{array} \right.
 \end{array}$$

As in the propositional case, one can show by this construction that any finite consistent set of prefixed formulas can be extended to one that is maximally consistent, prefix complete, and *E complete*, that is, it contains a witness for every true existential or false universal formula. Indeed, in the definition of consistency we can restrict rule applications of $T\exists$ and $F\forall$ to be single-usage, along the lines of Definition 5.2. One then constructs a model from a maximally consistent, prefix complete, *E complete* set, in which the quantifier domain is the collection of all parameters, and the possible worlds are the prefixes, as before. The details are straightforward and are omitted here, but a similar construction for modal logics is discussed in [9]. Completeness for the nested sequent calculus then follows by translation, just as in the propositional setting.

Here are some parallel examples: proofs in both systems of the formula $(\forall x)(A \vee B(x)) \supset (A \vee (C \supset (\forall x)B(x)))$.

Example 8.1 A tableau proof of $(\forall x)(A \vee B(x)) \supset (A \vee (C \supset (\forall x)B(x)))$. Items 2 and 3 are from 1 by $F \supset$; 4 and 5 are from 3 by $F\forall$; 6 and 7 are from 5 by $F \supset$; 8 is from 7 by $F\forall$, a is new to the branch at this point; 9 is from 2 by $T\forall$, which allows any parameter; 10 and 11 are from 9 by $T\vee$; 12 is from 11 by *Lower*.



Next is a nested sequent example, giving the tableau proof from Example 8.1 as converted to the sequent system. Note that in the application of the $R\forall$ rule, the parameter a does not occur in the conclusion. That is, it does not occur in $\rightarrow [(\forall x)(A \vee B(x)) \rightarrow A, [C \rightarrow (\forall x)B(x)]]$. It is not sufficient that a not occur in $C \rightarrow (\forall x)B(x)$, the *entire* conclusion must be taken into account.

Example 8.2 A nested sequent proof of $(\forall x)(A \vee B(x)) \supset (A \vee (C \supset (\forall x)B(x)))$.

$$\begin{array}{l}
\rightarrow [A \rightarrow A, [C \rightarrow B(a)]] \quad \frac{\rightarrow [A \rightarrow A, [B(a), C \rightarrow B(a)]]}{\rightarrow [B(a) \rightarrow A, [C \rightarrow B(a)]]} \text{Lift} \\
\hline
\rightarrow [A \vee B(a) \rightarrow A, [C \rightarrow B(a)]] \quad \text{LV} \\
\hline
\rightarrow [(\forall x)(A \vee B(x)) \rightarrow A, [C \rightarrow B(a)]] \quad \text{L}\forall \\
\hline
\rightarrow [(\forall x)(A \vee B(x)) \rightarrow A, [C \rightarrow (\forall x)B(x)]] \quad \text{R}\forall \\
\hline
\rightarrow [(\forall x)(A \vee B(x)) \rightarrow A, C \supset (\forall x)B(x)] \quad \text{R}\supset \\
\hline
\rightarrow [(\forall x)(A \vee B(x)) \rightarrow (A \vee (C \supset (\forall x)B(x)))] \quad \text{R}\forall \\
\hline
\rightarrow (\forall x)(A \vee B(x)) \supset (A \vee (C \supset (\forall x)B(x))) \quad \text{R}\supset
\end{array}$$

9 Intuitionistic Logic Proper

As we have seen, constant domain intuitionistic logic has a simple and natural formulation using prefixed tableaux or nested sequents. Similar formulations of intuitionistic logic proper, using this machinery, are a bit more complex. Of course intuitionistic logic has familiar conventional tableau and sequent systems of long standing. Still, formulating intuitionistic logic using nesting or prefixes is of some independent interest, and so we present it now.

There are, in fact, two different ways (at least) that intuitionistic first-order logic can be formalized using present machinery. Both have their merits and detriments. We treat them in separate subsections.

9.1 Adding an Existence Predicate

This approach simply embeds first-order intuitionistic logic into the constant domain version of Section 8. Suppose we set aside a one-place predicate symbol, say $E(x)$, and understand it as an *existence* predicate in the sense of [21]. For a formula X , let X^E be X with quantifiers relativized to $E(x)$, that is, $[(\forall x)\varphi]^E$ is $(\forall x)(E(x) \supset \varphi^E)$ and $[(\exists x)\varphi]^E$ is $(\exists x)(E(x) \wedge \varphi^E)$. Then, essentially, to prove X intuitionistically, we prove X^E in one of the constant domain systems of Section 8.

There is, however, one minor complication. We still need to ensure that the domain of each possible world is non-empty. Models constructed during the course of a completeness proof have a tree structure, so

by monotonicity it is enough to have the root world's domain be non-empty. This leads us to the following official version.

Intuitionistic First-Order Prefixed E Tableaux An intuitionistic proof of X is a closed constant domain tableau beginning with the *two* formulas $1T(\exists x)E(x)$ and $1FX^E$.

Intuitionistic First-Order E Nested Sequents An intuitionistic proof of X is a constant domain nested sequent proof of the sequent $(\exists x)E(x) \rightarrow X^E$.

Completeness is quite straightforward to show. Essentially, if X is not provable there is a constant domain countermodel for X^E . Convert this to a standard intuitionistic model by setting the domain of each possible world to be the set of things to which E applies at that world. The rules *Lift* for nested sequents and *Lower* for prefixed tableaux give us the monotonicity property for domains that intuitionistic semantics requires. And we have directly built in the requirement that domains be non-empty. We omit details.

The formula $(A \vee (\forall x)B(x)) \supset (\forall x)(A \vee B(x))$ is provable, when relativized to $E(x)$ and using the constant domain rules. It is illustrative to carry out proofs of the relativized version in both the tableau and the sequent systems.

9.2 Restricting Instantiation

Instead of introducing an existence predicate one can work with formulas as given, but one must complicate the universal instantiation rules from Section 8 a bit.

Definition 9.1 Let S be a set of prefixed signed formulas. Call a parameter a *available for prefix σ in S* if a occurs in the formula part of some member of S whose prefix is an initial segment of σ (the initial segment need not be proper). Also a is *new* in S if it does not occur in the formula part of any member of S .

For a tableau branch, parameter a is *available for σ on the branch* if it is available for σ in the set of prefixed signed formulas appearing on the branch. Similarly for a being *new* on a branch.

Now, here are the tableau quantifier rules. The terms *available* and *new* refer to the branch on which the rule is applied.

Direct Intuitionistic Tableau Quantifier Rules

$$\begin{array}{ccc}
 T\exists & \frac{\sigma T(\exists x)\varphi(x)}{\sigma T\varphi(a)} & F\exists & \frac{\sigma F(\exists x)\varphi(x)}{\sigma F\varphi(a)} \\
 & a \text{ new} & & a \text{ available for } \sigma \\
 & & & \text{or } a \text{ new} \\
 \\
 T\forall & \frac{\sigma T(\forall x)\varphi(x)}{\sigma T\varphi(a)} & F\forall & \frac{\sigma F(\forall x)\varphi(x)}{\sigma.n F\varphi(a)} \\
 & a \text{ available for } \sigma & & a \text{ new} \\
 & \text{or } a \text{ new} & & \text{and } \sigma.n \text{ new}
 \end{array}$$

Direct Intuitionistic First-Order Prefixed Tableaux A direct intuitionistic proof of X is a closed prefixed tableau beginning with $1FX$, using the Direct Intuitionistic Tableau Quantifier Rules given above and the propositional rules from Section 5.

Example 9.2 First we note that Example 8.1 is not a correct tableau under the present Direct rules, because the step introducing item 9 is no longer allowed. At this point in the tableau construction the parameter a is not new since it occurs in item 8, and a is not available at parameter 1.1 since it only occurs in a signed formula with prefix 1.1.1, which is not an initial segment of 1.1.

Second, here is an example of a tableau proof using the Direct quantifier rules. It is a proof of $(\forall x)A(x) \supset \neg(\exists x)\neg A(x)$.

- 1 $F (\forall x)A(x) \supset \neg(\exists x)\neg A(x)$ 1.
- 1.1 $T (\forall x)A(x)$ 2.
- 1.1 $F \neg(\exists x)\neg A(x)$ 3.
- 1.1.1 $T (\exists x)\neg A(x)$ 4.
- 1.1.1 $T \neg A(a)$ 5.
- 1.1.1 $F A(a)$ 6.
- 1.1.1 $T (\forall x)A(x)$ 7.
- 1.1.1 $T A(a)$ 8.

Items 2 and 3 are from 1 by $T \supset$; 4 is from 3 by $F \neg$; 5 is from 4 by $T \exists$ (a is new at this point); 6 is from 5 by $T \neg$. At this point we cannot apply Direct rule $T \forall$ to item 2 to add 1.1 $T A(a)$, since a is not an available parameter at prefix 1.1, but we can use *Lower* on 2 to add 7. Then since a is available at 1.1.1, we can use $T \forall$ on 7 to add 8. Closure is by 6 and 8.

We sketch an argument for tableau soundness. The argument for propositional soundness in Section 6 must be enhanced in order to take the quantificational machinery into account. Recall, Kripke first-order intuitionistic models associate a domain with each state, and add the requirement of monotonicity: if state w_2 is accessible from state w_1 then the domain associated with w_1 is a subset of that associated with w_2 . Quantification at each state is with respect to the domain associated with that state.

The definition of tableau satisfiability from Section 6 is modified as follows. Now call a set S of prefixed signed formulas *satisfiable* if there is a first-order Kripke intuitionistic model \mathcal{M} , a mapping p from prefixes occurring in S to states in the model, and a mapping d from parameters occurring in S to members occurring in domains of the model, all meeting the following conditions. 1) If σ and $\sigma.n$ both occur in S , then $p(\sigma.n)$ is accessible from $p(\sigma)$. 2) If $\sigma \mathcal{X}$ is in S and parameter a occurs in \mathcal{X} then $d(a)$ is in the quantifier domain the model associates with $p(\sigma)$. 3) If $\sigma T X \in S$ then X is forced at the state $p(\sigma)$. 4) if $\sigma F X \in S$ then X is not forced at the state $p(\sigma)$. As before, call a tableau branch satisfiable if the set of prefixed signed formulas on it is satisfiable, and call a tableau satisfiable if one of its branches is. It must be shown that if a tableau rule is applied to a satisfiable tableau, the result is another satisfiable tableau. We verify one rule, $T \forall$; the rest are similar or easier.

Suppose a tableau branch is satisfiable and $\sigma T (\forall x)A(x)$ occurs on the branch. We show the branch is still satisfiable after adding $\sigma T A(a)$ where a is available at σ on the branch, and also where a is new to the branch. Since the original branch is satisfiable, there is a Kripke model and mappings p and d meeting the conditions outlined above. Suppose first that a is available at σ on the branch. Then there is some prefixed signed formula $\tau \mathcal{X}$ already on the branch, with a occurring in \mathcal{X} and with τ being an initial segment of σ . Then $d(a)$ is in the domain associated with state $p(\tau)$. Since τ is an initial segment of σ , in the model $p(\sigma)$ is accessible from $p(\tau)$ and hence by monotonicity $d(a)$ is also in the domain associated with $p(\sigma)$. Since $\sigma T (\forall x)A(x)$ is on the branch, $(\forall x)A(x)$ is forced at state $p(\sigma)$. It now follows that $A(a)$ is also forced at state $p(\sigma)$, and hence the branch extended with $\sigma T A(a)$ is satisfiable. The second case, where a is new, is simpler. We can simply extend d by mapping a to any member of the domain associated with $p(\sigma)$. We omit details.

Next we turn to completeness. The proof is much like the one sketched in Section 8. We want to extend a consistent set to one that is maximally consistent, prefix complete, and E complete, as before, but there are two changes. First the definition of consistency, while still apparently reading the same, has changed its meaning. A set is consistent if no tableau for a finite subset closes, but the tableau now must be constructed in accordance with the Direct Intuitionistic Quantifier Rules—in particular, availability conditions are imposed. The second issue is that, if we construct a sequence of sets S_n as before, starting with a finite consistent set, as work continues more parameters can become available for a given prefix, so more applications of $T \forall$ and $F \exists$ become possible. This must be taken into account.

We use the definition of the S_n sequence as given in Section 8, understanding that consistency has a revised meaning. But we make an alteration.

Definition 9.3 For a set P of prefixed signed formulas, by the *universal saturation* of P we mean the result of: 1) adding to P all signed formulas $\sigma T A(a)$ such that $\sigma T (\forall x)A(x)$ is in P and a is a parameter that is

available for σ in P , and 2) adding to P all signed formulas $\sigma F A(a)$ such that $\sigma F (\exists x)A(x)$ is in P and a is a parameter that is available for σ in P .

The definition of the S_n sequence from Section 8 is modified so that, at each stage, S_{n+1} is enlarged to its universal saturation.

Let us write M for $\cup_n S_n$ and think of it as the limit of the S_n sequence. A first-order intuitionistic model can be constructed from M , and then a truth lemma can be proved. Here are the significant points. As usual, the states are the prefixes, and accessibility corresponds to initial-segment-of. For each prefix/state σ , the domain associated with that state consists of all parameters that are available for σ in M . Parameters designate themselves, as usual. This is familiar ground and we leave the remaining details to the reader.

Next, we convert these notions and rules to the nested sequent setting. We omit all discussion of soundness and completeness.

Definition 9.4 Let $\Gamma_1 \rightarrow \Gamma_2$ be a nested sequent. Any parameter a that occurs in some formula of Γ_1 or Γ_2 is said to be *available* in this sequent and in all its boxed subsequents (no matter how deeply nested).

For instance, consider the sequent $A \rightarrow B(a), [C \rightarrow D(b), [E \rightarrow F]]$. The parameter a is available in the entire sequent and in the nested subsequents $C \rightarrow D(b), [E \rightarrow F]$ and $E \rightarrow F$. The parameter b is available in $C \rightarrow D(b), [E \rightarrow F]$ and in $E \rightarrow F$.

Direct Intuitionistic Nested Sequent Quantifier Rules

$$\begin{array}{c}
 L\exists \quad \frac{\Gamma_1, \varphi(a) \rightarrow \Gamma_2}{\Gamma_1, (\exists x)\varphi(x) \rightarrow \Gamma_2} \\
 a \text{ not in conclusion}
 \end{array}
 \qquad
 \begin{array}{c}
 R\exists \quad \frac{\Gamma_1 \rightarrow \Gamma_2, \varphi(a)}{\Gamma_1 \rightarrow \Gamma_2, (\exists x)\varphi(x)} \\
 a \text{ available in conclusion} \\
 \text{or } a \text{ not in conclusion}
 \end{array}$$

$$\begin{array}{c}
 L\forall \quad \frac{\Gamma_1, \varphi(a) \rightarrow \Gamma_2}{\Gamma_1, (\forall x)\varphi(x) \rightarrow \Gamma_2} \\
 a \text{ available in conclusion} \\
 \text{or } a \text{ not in conclusion}
 \end{array}
 \qquad
 \begin{array}{c}
 R\forall \quad \frac{\Gamma_1 \rightarrow \Gamma_2, [\rightarrow \varphi(a)]}{\Gamma_1 \rightarrow \Gamma_2, (\forall x)\varphi(x)} \\
 a \text{ not in conclusion}
 \end{array}$$

Direct Intuitionistic First-Order Nested Sequents An intuitionistic proof of X is a nested sequent proof of the sequent $\rightarrow X$ using the Direct Intuitionistic Nested Sequent Quantifier Rules given above.

Example 9.5 Here is an intuitionistic nested sequent using the Direct quantifier rules, of $(\forall x)A(x) \supset \neg(\exists x)\neg A(x)$. It is translated from the tableau proof in Example 9.2.

$$\begin{array}{c}
 \rightarrow [\rightarrow [A(a) \rightarrow A(a)]] \\
 \hline
 \rightarrow [\rightarrow [(\forall x)A(x) \rightarrow A(a)]] \quad L\forall \\
 \hline
 \rightarrow [(\forall x)A(x) \rightarrow [\rightarrow A(a)]] \quad Lift \\
 \hline
 \rightarrow [(\forall x)A(x) \rightarrow [\neg A(a) \rightarrow]] \quad L\neg \\
 \hline
 \rightarrow [(\forall x)A(x) \rightarrow [(\exists x)\neg A(x) \rightarrow]] \quad L\exists \\
 \hline
 \rightarrow [(\forall x)A(x) \rightarrow \neg(\exists x)\neg A(x)] \quad R\neg \\
 \hline
 \rightarrow (\forall x)A(x) \supset \neg(\exists x)\neg A(x) \quad R\supset
 \end{array}$$

10 Conclusion

The family of logics having nested sequent systems is under much current investigation. Constant domain intuitionistic logic is now seen to be in this family. The nested sequent system is of a simple nature and,

rather surprisingly, amounts to a propositional intuitionistic nested sequent system, plus classical quantifier rules, a combination known not to work for conventional sequent systems. The connection between prefixed tableaux and nested sequents, known from modal logic, extends to the intuitionistic case too.

For many years the status of interpolation for constant domain intuitionistic logic was an open problem. It has recently been established that it *does not* hold, [17]. It has long been known to hold for intuitionistic logic itself. It would be interesting to see if the proof systems of this paper can help shed some light on this difference.

References

- [1] K. Brünnler. Deep sequent systems for modal logic. *Archive for Mathematical Logic*, 48(6):551–577, 2009.
- [2] K. Brünnler. *Nested Sequents*. Habilitation thesis, Institut für Informatik und angewandte Mathematik, Universität Bern, 2010.
- [3] K. Došen. Sequent-Systems for modal logic. *Journal of Symbolic Logic*, 50:149–168, 1985.
- [4] C. Fiorentini and P. Miglioli. A cut-free sequent calculus for the logic of constant domains with a limited amount of duplications. *Logic Journal of IGPL*, 7(6):733–753, 1999.
- [5] F. Fitch. Tree proofs in modal logic. *Journal of Symbolic Logic*, 31:152, 1966. (abstract).
- [6] M. C. Fitting. Tableau methods of proof for modal logics. *Notre Dame Journal of Formal Logic*, 13:237–247, 1972.
- [7] M. C. Fitting. *Proof Methods for Modal and Intuitionistic Logics*. D. Reidel Publishing Co., Dordrecht, 1983.
- [8] M. C. Fitting. A mistake on my part. In S. Artemov, H. Berringer, A. d’Avila Garcez, L. Lamb, and J. Woods, editors, *We Will Show Them! Essays in honour of Dov Gabbay*, volume 1, pages 665–669. College Publications, 2005.
- [9] M. C. Fitting. Modal proof theory. In P. Blackburn, J. van Benthem, and F. Wolter, editors, *Handbook of Modal Logic*, chapter 2, pages 85–138. Elsevier, 2007.
- [10] M. C. Fitting. Prefixed tableaux and nested sequents. *Annals of Pure and Applied Logic*, 163:291–313, 2012. Available on-line at <http://dx.doi.org/10.1016/j.apal.2011.09.004>.
- [11] R. Goré, L. Postniece, and A. Tiu. Cut-elimination and proof-search for bi-intuitionistic logic using nested sequents. In *Advances in Modal Logic*, volume 7, pages 43–66. College Publications, 2008.
- [12] S. Görneman. A logic stronger than intuitionism. *Journal of Symbolic Logic*, 36:249–261, 1971.
- [13] A. Grzegorzcyk. A philosophically plausible formal interpretation of intuitionistic logic. *Journal of Symbolic Logic*, 26:596–601, 1964.
- [14] R. Kashima. Cut-free sequent calculi for some tense logics. *Studia Logica*, 53:119–135, 1994.
- [15] R. Kashima and T. Shimura. Cut-elimination theorem for the logic of constant domains. *Mathematical Logic Quarterly*, 40:153–172, 1994.
- [16] D. Klemke. *Ein vollständiger Kalkül für die Folgerungsbeziehung der Grzegorzcyk-Semantik*. PhD thesis, Freiburg, 1969.
- [17] G. Mints, G. Olkhovikov, and A. Urquhart. Failure of interpolation in the intuitionistic logic of constant domains. 2012.

- [18] F. Poggiolesi. The tree-hypersequent method for modal propositional logic. In D. Makinson, J. Malinowski, and H. Wansing, editors, *Trends in Logic: Towards Mathematical Philosophy*, pages 31–51. Springer, 2009.
- [19] L. Postniece. Deep inference in bi-intuitionistic logic. In H. Ono, M. Kanazawa, and R. de Queiroz, editors, *WoLLIC*, volume 5514 of *Lecture Notes in Computer Science*, pages 320–334. 2009.
- [20] G. Sambin, G. Battilotti, and C. Faggian. Basic logic: Reflection, symmetry, visibility. *Journal of Symbolic Logic*, 65(3):979–1013, 2000.
- [21] D. Scott. Identity and existence in intuitionistic logic. In M. Fourman, C. Mulvey, and D. Scott, editors, *Applications of Sheaves*, volume 753 of *Lecture Notes in Mathematics*, pages 660–696. Springer Berlin / Heidelberg, 1979. 10.1007/BFb0061839.