

On the Correspondence between Nested Calculi and Semantic Systems for Intuitionistic Logics

Tim Lyon

Institut für Logic and Computation, Technische Universität Wien, 1040 Wien, Austria
lyon@logic.at

Abstract. This paper studies the relationship between labelled and nested calculi for propositional intuitionistic logic, first-order intuitionistic logic with non-constant domains, and first-order intuitionistic logic with constant domains. It is shown that Fitting’s nested calculi naturally arise from their corresponding labelled calculi—for each of the aforementioned logics—via the elimination of structural rules in labelled derivations. The translational correspondence between the two types of systems is leveraged to show that the nested calculi inherit proof-theoretic properties from their associated labelled calculi, such as completeness, invertibility of rules, and cut-admissibility. Since labelled calculi are easily obtained via a logic’s semantics, the method presented in this paper can be seen as one whereby refined versions of labelled calculi (containing nested calculi as fragments) with favorable properties are derived directly from a logic’s semantics.

Keywords: First-order · Intuitionistic logic · Kripke semantics · Labelled calculi · Nested calculi · Proof theory · Propositional · Refinement

1 Introduction

In his seminal work [12], Gentzen introduced the *sequent calculus* framework for classical and intuitionistic logic, and proved the celebrated *Hauptsatz*, i.e. *cut-elimination theorem*, for the systems. As a corollary of his theorem, it can be observed that any formula provable in one of Gentzen’s systems, is provable with an *analytic derivation*, that is, a derivation where all formulae used to *reach* the conclusion of the derivation, *occur* in the conclusion of the derivation. This method of proof happens to be of practical consequence, and as such, many variations and extensions of Gentzen’s sequent calculi have been assembled and proposed—examples include tableaux calculi [7,9], display calculi [2,18,28], hypersequent calculi [1,22], labelled calculi [11,24,27], and nested calculi [3,4,14]. Such calculi have been applied to provide decision algorithms [12,22], to automate the extraction of interpolants [19], and to automated counter-model extraction [17,25].

In this paper, we focus entirely on the labelled and nested proof-theoretic paradigms. The labelled paradigm may be qualified as *semantic* as calculi are built by transforming the semantic clauses and Kripke-frame properties of a logic

into inference rules [24,27]. Despite some drawbacks and criticisms of this approach [1], the labelled paradigm offers many advantages. First, it is relatively straightforward to transform the semantics of a logic into a calculus; in fact, this process has been shown to be *automatable* [5]. Second, the approach is exceptionally modular—allowing for the addition or deletion of rules to immediately obtain calculi for weaker or stronger logics—and is applicable to a wide variety of logics [6,17,24,27]. Last, labelled calculi consistently possess fundamental proof-theoretic properties such as invertibility of rules, admissibility of structural rules, and cut-admissibility—with fairly general results provided for large classes of modal, intuitionistic, and related logics [6,17,24,27]. Although these characteristics are certainly favorable, a drawback of labelled calculi is that they typically involve a complicated syntax (which incorporates a large amount of semantic information), the sequents utilized in proofs encode general graphs, and inference rules often violate the subformula property (i.e. labelled calculi are not usually analytic). Such properties cause an unnecessary increase in the size of sequents/proofs, and a decrease in the efficiency of associated automated reasoning algorithms.

In contrast to the data structures (called, *labelled sequents*) employed in labelled calculi—which can be viewed as general graphs—the nested paradigm employs treelike data structures (called, *nested sequents*) which manipulate logical information and are used to derive theorems. The inception of the paradigm is often attributed to Bull [4] and Kashima [14], though it should be noted that nested calculi can be considered ‘upside down’ versions of prefixed tableaux calculi, which were introduced much earlier in 1972 by Fitting [7]. (NB. See [9] for a discussion on the relationship between nested and tableaux systems). A strength of the nested paradigm is that the nested sequents employed reduce the bureaucracy and syntactic structures appearing in proofs, showing the nested formalism to be more parsimonious than the labelled formalism. Also, the nested formalism has proven itself useful in applications such as constructing analytic calculi [3,23], developing automated reasoning methods [25], and verifying interpolation [19]. Still, in spite of these advantages, the construction of nested calculi and the confirmation of their proof-theoretic properties is often done on a case-by-case basis. That is to say, the nested paradigm lacks general results—like those of the labelled paradigm—regarding the immediate construction of calculi in possession of fundamental properties.

Due to the fact that the labelled formalism is well-suited for constructing calculi possessing essential proof-theoretic properties, and the nested formalism is more refined and better suited for a variety of applications, a method of extracting nested calculi from labelled calculi—with the properties of the latter preserved—is highly desirable. This *refinement* process allows us to capture the best of both worlds: we invoke the general results of the labelled setting to obtain satisfactory labelled calculi for a class of logics, and via refinement, transform the systems into nested calculi better suited for applications. Similar ideas and relationships have been discussed in the literature [13,17,16,20,21], where refined calculi (which can be considered nested calculi) were derived from labelled calculi

for modal, intuitionistic, and related logics. (NB. The paper [21] mentions results strongly related to Sect. 5. Although the results presented here were discovered independently, the work of Sect. 5 can be seen as a detailed explication and expansion of the work presented in [21]. Moreover, [21] does not consider the non-trivial and interesting first-order cases considered here.)

In this paper we advance our understanding of the aforementioned method, and show how to extract *slight* variants of Fitting’s nested calculi for propositional intuitionistic logic, first-order intuitionistic logic with non-constant domains, and first-order intuitionistic logic with constant domains from the labelled calculi for these logics. Additionally, we demonstrate the converse translation—showing how to transform each considered nested calculus into its associated labelled calculus. These translations are worthwhile in that they show how each nested calculus inherits properties from its corresponding labelled calculus, and also shed light on how the semantics of each logic affects the shape of rules and syntactic structures inherent in nested derivations (via the labelled calculi).

The organization of this paper is as follows: Sect. 2 introduces the semantics and axiomatizations for the intuitionistic logics we will consider. Sect. 3 introduces the labelled and nested calculi for these logics. Sect. 4 introduces preliminary definitions and lemmata sufficient to translate intuitionistic labelled calculi into nested calculi, and vice-versa. Sect. 5, Sect. 6, and Sect. 7 show how to refine each labelled calculus through structural rule elimination, allowing for the extraction of the nested calculus from the labelled calculus for propositional intuitionistic logic, first-order intuitionistic logic with non-constant domains, and first-order intuitionistic logic with constant domains, respectively. Sect. 8 shows how to translate each nested calculus into its associated labelled calculus, and discusses corollaries of the translational correspondence between the calculi—primarily focusing on the properties inherited by each nested calculus from its corresponding labelled calculus. The last section, Sect. 9, concludes.

This paper serves as an enhanced and revised version of the conference paper [16]. Most significantly, the nested-to-labelled translation of Sect. 4, the content of Sect. 6, and the majority of content in Sect. 8 are entirely new. Beyond this, the paper was written with an increased focus and more detailed exposition on *how* the labelled calculi are refined in order to extract each nested calculus—this provides the paper with more explanatory force than [16]. Also, it should be noted that this paper corrects an error that occurs in the conference version. In [16], the labelled calculus $\mathbf{G3IntQC}$ is lacking a structural rule (called *ihd* in this paper) corresponding to the condition that domains in first-order intuitionistic models are *non-empty*, or *inhabited*. Without the inclusion of this rule, $\mathbf{G3IntQC}$ is incomplete as modus ponens cannot be simulated in the calculus (see Thm. 3 and Appendix A for details). We have included this rule here and adjusted the content of [16] regarding $\mathbf{G3IntQC}$ accordingly. Furthermore, to increase the flow and readability of the paper, some results outside the scope of, or auxiliary to, our main focus (i.e. the translational correspondence between labelled and nested calculi for intuitionistic logics) have been moved to a technical appendix (Appendix A) for the interested reader.

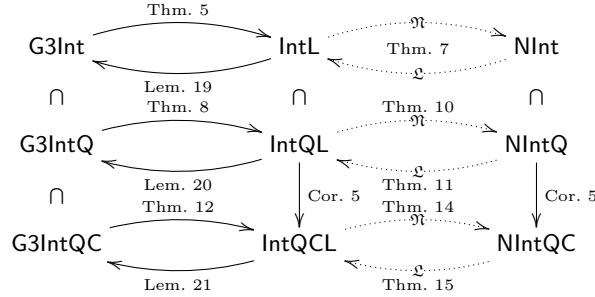


Fig. 1. Transformations and translations between the intuitionistic calculi considered.

Last, Fig. 1 shows the transformations (indicated by a solid arrow) and translations (indicated by a dotted arrow) between the various intuitionistic calculi considered. Transformations indicate that derivations in one system (or in a fragment of the system) are effectively (i.e. algorithmically) transformable to derivations in another system, and translations indicate a transformation along with a change in notation. The symbols \mathfrak{N} and \mathfrak{L} represent a change from labelled to nested notation, and nested to labelled notation, respectively. The inclusion sign \subset is taken to mean that one calculus consists of a proper subset of the rules in another calculus.

2 Logical Preliminaries

In this section, we introduce the language, semantics, and axiomatizations for propositional intuitionistic logic Int , first-order intuitionistic logic with non-constant domains IntQ , and first-order intuitionistic logic with constant domains IntQC . The first subsection will discuss the propositional setting, whereas the second subsection will discuss the first-order setting.

2.1 Propositional Intuitionistic Logic

The propositional language \mathcal{L} is defined via the BNF grammar shown below:

$$A ::= p \mid \perp \mid (A \vee A) \mid (A \wedge A) \mid (A \supset A)$$

where p is among a denumerable set of *propositional variables* $\text{Prop} = \{p, q, r, \dots\}$. As usual, we define intuitionistic negation as $\neg A := A \supset \perp$. Moreover, the language admits a relational (or, Kripke-style) semantics as defined below (cf. [15]).

Definition 1 (Int-Frame, Int-Model). *An Int-frame is an ordered pair $F = (W, \leq)$ such that:*

- ▶ W is a non-empty set of worlds $\{w, u, v, \dots\}$;

► $\leq \subseteq W \times W$ is a reflexive and transitive binary relation on W .¹

An **Int**-model is an ordered pair $M = (F, V)$ where F is an **Int**-frame and $V : \text{Prop} \mapsto 2^W$ is a valuation function mapping propositional variables to subsets of W satisfying the following monotonicity condition:

(M) If $w \in V(p)$ and $w \leq v$, then $v \in V(p)$.

Definition 2 (Propositional Semantic Clauses). Let $M = (W, \leq, V)$ be an **Int**-model with $w \in W$. The satisfaction relation $M, w \Vdash A$ between $w \in W$ and a formula A from \mathcal{L} is inductively defined as follows:

- $M, w \not\Vdash \perp$;
- $M, w \Vdash p$ iff $w \in V(p)$;
- $M, w \Vdash A \vee B$ iff $M, w \Vdash A$ or $M, w \Vdash B$;
- $M, w \Vdash A \wedge B$ iff $M, w \Vdash A$ and $M, w \Vdash B$;
- $M, w \Vdash A \supset B$ iff for all $u \in W$, if $w \leq u$ and $M, u \Vdash A$, then $M, u \Vdash B$.

We say that a formula A is globally true on M , written $M \Vdash A$, iff $M, u \Vdash A$ for all worlds $u \in W$. A formula A is **Int**-valid, written $\Vdash_{\text{Int}} A$, iff it is globally true on all **Int**-models. Last, we say that a set Φ of formulae semantically implies a formula A , written $\Phi \Vdash A$, iff for all intuitionistic models M with $w \in W$, if $M, w \Vdash B$ for all $B \in \Phi$, then $M, w \Vdash A$.²

The monotonicity condition (M), together with the intuitionistic semantics defined above, necessitates a general form of monotonicity as detailed below:

Lemma 1 (General Monotonicity). Let M be a model with $w, v \in W$ of M . If $M, w \Vdash A$ and $w \leq v$, then $M, v \Vdash A$.

Proof. See [10, Lem. 3.2.16] for details. □

Additionally, propositional intuitionistic logic **Int** is finitely axiomatizable. The axioms and inference rule syntactically characterizing **Int** are given in Fig. 2. We define an **Int**-derivation (relative to the axiomatization of **Int**) from a set of formulae Φ (written $\Phi \vdash A$) in the usual way (cf. [10,26]). It is well-known that the notion of semantic consequence for **Int** is equivalent to the syntactic notion of an **Int**-derivation from a set of formulae:

Theorem 1 (Soundness and Completeness [10]). For any $A \in \mathcal{L}$, $\Phi \Vdash A$ iff $\Phi \vdash A$.

¹The properties imposed on \leq are defined as follows: (reflexivity) for all $w \in W$, $w \leq w$, and (transitivity) for all $w, u, v \in W$, if $w \leq v$ and $v \leq u$, then $w \leq u$.

²We note that we could define a *global* version of semantic implication as follows: A set of formulae Φ *globally implies* a formula A iff for all intuitionistic models M , if $M \Vdash B$ for all $B \in \Phi$, then $M \Vdash A$. We make use of the local version in Def. 2 however, because the axiomatization we use for **Int** (shown in Fig. 2) is sound and complete relative to the local version of semantic implication (Thm. 1) [10].

$$\begin{array}{l}
A \supset (B \supset A) \quad (A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C)) \quad A \supset (B \supset (A \wedge B)) \\
(A \wedge B) \supset A \quad (A \wedge B) \supset B \quad A \supset (A \vee B) \quad \frac{A \quad A \supset B}{B} \text{ (mp)} \\
B \supset (A \vee B) \quad \perp \supset A \quad (A \supset C) \supset ((B \supset C) \supset ((A \vee B) \supset C))
\end{array}$$

Fig. 2. An axiomatization of propositional intuitionistic logic Int [10].

2.2 First-Order Intuitionistic Logics

The language \mathcal{Q} for our first-order logics is defined via the BNF grammar below:

$$A ::= p(x_1, \dots, x_n) \mid \perp \mid (A \vee A) \mid (A \wedge A) \mid (A \supset A) \mid (\forall x)A \mid (\exists x)A$$

where p is among a denumerable set of n -ary *predicate symbols* $\text{Pred} = \{p, q, r, \dots\}$ and x_1, \dots, x_n, x are *variables* (with $n \in \mathbb{N}$). We refer to formulae of the form $p(x_1, \dots, x_n)$ as *atomic formulae* when $n > 0$, and refer to formulae of the form p as *propositional variables* when $n = 0$ (i.e. a 0-ary predicate p is a propositional variable). Furthermore, we define a variable x to be a *free variable* in A iff it is not within the scope of a quantifier $\forall x$ or $\exists x$, and to be a *bound variable* iff it is within the scope of a quantifier. Last, as in the propositional case, our language admits a relational semantics, defined below.

Definition 3 (IntQ-Frames, IntQ-Models [10]). We define an IntQ-frame to be a tuple $F = (W, \leq, D)$ such that:

- ▶ W is a non-empty set of worlds $\{w, u, v, \dots\}$;
- ▶ $\leq \subseteq W \times W$ is a reflexive and transitive binary relation on W ;
- ▶ D is a domain function mapping a world $w \in W$ to a non-empty set D_w of objects $\{a, b, c, \dots\}$ satisfying the nested domain condition shown below:

(ND) If $a \in D_w$ and $w \leq v$, then $a \in D_v$.

A IntQC-frame is an IntQ-frame that additionally satisfies the following constant domain condition shown below:

(CD) If $a \in D_v$ and $w \leq v$, then $a \in D_w$.

An IntQ-model (IntQC-model) M is an ordered pair (F, V) where F is an IntQ-frame (IntQC-frame) and V is a valuation function such that $V(p, w) \subseteq (D_w)^n$ (with $n \in \mathbb{N}$) satisfying the following monotonicity condition:

- (M) If $w \in V(p, w)$ and $w \leq v$, then $v \in V(p, v)$ (if p is of arity 0);
If $w \leq v$, then $V(p, w) \subseteq V(p, v)$ (if p is of arity $n > 0$).

We uphold the convention in [10] and assume that for each world $w \in W$, $(D_w)^0 = \{w\}$, so $V(p, w) = \{w\}$ or $V(p, w) = \emptyset$, for a propositional variable p .

As in [10], we forgo the direct interpretation of formulae from \mathcal{Q} on relational models, and instead, introduce D_w -sentences (Def. 4) to be interpreted on such models. Defining satisfaction relative to D_w -sentences gives rise to a notion of validity for formulae in \mathcal{Q} (Def. 5). However, this notion of validity also depends on the *universal closure* of a formula: given that $A \in \mathcal{Q}$ contains only x_1, \dots, x_n as free variables, the universal closure $\forall \vec{x}A$ is taken to be the formula $\forall x_1 \dots \forall x_n A$.

Definition 4 (D_w -Sentence). *Let $M = (W, \leq, D, V)$ be an IntQ-model with $w \in W$. We define $\mathcal{Q}(D_w)$ to be the language \mathcal{Q} expanded with parameters $\underline{a}, \underline{b}, \underline{c}, \dots$ corresponding to the objects in the set $D_w = \{a, b, c, \dots\}$. We define a D_w -formula to be a formula in $\mathcal{Q}(D_w)$, and we define a D_w -sentence to be a D_w -formula that does not contain any free variables.*

Definition 5 (First-Order Semantic Clauses). *Let $M = (W, \leq, D, V)$ be an IntQ- or IntQC-model with $w \in W$. The satisfaction relation $M, w \Vdash A$ between w and a D_w -sentence A is inductively defined as follows:*

- ▶ *If p is a propositional variable, then $M, w \Vdash p$ iff $w \in V(p, w)$;*
- ▶ *If p is an n -ary predicate symbol (with $n > 0$), then $M, w \Vdash p(\underline{a}_1, \dots, \underline{a}_n)$ iff $(a_1, \dots, a_n) \in V(p, w)$;*
- ▶ *$M, w \Vdash \forall x A$ iff for all $v \in W$ and all $a \in D_v$, if $w \leq v$, then $M, v \Vdash A(\underline{a}/x)$;*
- ▶ *$M, w \Vdash \exists x A$ iff there exists an $a \in D_w$ such that $M, w \Vdash A(\underline{a}/x)$.*

The clauses for the \vee , \wedge , and \supset connectives are the same as in Def. 2. We say that a formula A is globally true on M , written $M \Vdash A$, iff $M, u \Vdash \forall \vec{x}A$ for all worlds $u \in W$. A formula A is IntQ-valid (IntQC-valid), written $\Vdash_{\text{IntQ}} A$ ($\Vdash_{\text{IntQC}} A$, resp.), iff it is globally true on all IntQ-models (IntQC-models).

Similar to the propositional case, the monotonicity condition imposed on atomic formulae in models generalizes:

Lemma 2 (General Monotonicity). *Let M be an IntQ- or IntQC-model with $w, v \in W$ of M . For any D_w -sentence A , if $M, w \Vdash A$ and $w \leq v$, then $M, v \Vdash A$.*

Proof. See [10, Lem. 3.2.16] for details. \square

Sound and complete axiomatizations for our first-order intuitionistic logics (viz. IntQ and IntQC) are provided in Fig. 3. We define the *substitution* (y/x) of the variable y for the free variable x on a formula A in the standard way as the replacement of all free occurrences of x in A with y . Last, the side condition *y is free for x* (see Fig. 3) is taken to mean that y does not become bound by a quantifier if substituted for x .

The logics IntQ and IntQC are defined to be the smallest set of formulae from \mathcal{Q} closed under substitutions of the axioms and applications of the inference rules in their axiomatizations. The sole difference between the axiomatizations for IntQ and IntQC is that the former omits the *constant domain axiom* $\forall x(A \vee B) \supset \forall x A \vee B$ (with $x \notin B$) whereas the latter includes it. We write $\vdash_{\text{IntQ}} A$ ($\vdash_{\text{IntQC}} A$) to denote that A is an element, or *theorem*, of IntQ (IntQC, resp.).

Theorem 2 (Soundness and Completeness [10]). *For any $A \in \mathcal{Q}$, $\Vdash_{\text{IntQ}} A$ ($\Vdash_{\text{IntQC}} A$) iff $\vdash_{\text{IntQ}} A$ ($\vdash_{\text{IntQC}} A$, resp.).*

$$\begin{array}{l} \forall x(B \supset A) \supset (B \supset \forall xA) \text{ with } x \notin B \quad \forall x(A \supset B) \supset (\exists xA \supset B) \text{ with } x \notin B \\ \\ \forall xA \supset A(y/x) \text{ } y \text{ free for } x \quad A(y/x) \supset \exists xA \text{ } y \text{ free for } x \quad \frac{A}{\forall xA} \text{ gen} \\ \\ \forall x(A \vee B) \supset \forall xA \vee B \text{ with } x \notin B \end{array}$$

Fig. 3. The axiomatization for the logic **IntQ** is given by extending the axiomatization of **Int** with the first two rows. The axiomatization for the logic **IntQC** is given by extending the axiomatization of **Int** with all three rows. Both axiomatizations can be found in [10].

3 Proof Calculi for Intuitionistic Logics

In this section, we introduce the labelled and nested proof systems for the intuitionistic logics **Int**, **IntQ**, and **IntQC**. The first subsection presents the labelled system **G3Int** for propositional intuitionistic logic from [6] as well as the first-order extensions of this calculus, that makes use of quantifier and structural rules motivated by those (viz. $\forall L$, $\forall R$, id , and dd) given in [27, Ch. 6]. The second subsection introduces (slight variants of) Fitting’s nested calculi for intuitionistic logics from [9].

3.1 The Labelled Calculi **G3Int**, **G3IntQ**, and **G3IntQC**

We define *propositional (first-order) labelled sequents* to be syntactic objects of the form $L_1 \Rightarrow L_2$ ($L'_1 \Rightarrow L'_2$, resp.), where L_1 and L_2 (L'_1 and L'_2 , resp.) are formulae defined via the BNF grammar below top (below bottom, resp.).

$$\begin{array}{l} L_1 ::= \varepsilon \mid w : A \mid w \leq v \mid L_1, L_1 \quad L_2 ::= \varepsilon \mid w : A \mid L_2, L_2 \\ \\ L'_1 ::= \varepsilon \mid w : A \mid \underline{a} \in D_w \mid w \leq v \mid L'_1, L'_1 \quad L'_2 ::= \varepsilon \mid w : A \mid L'_2, L'_2 \end{array}$$

In the propositional case, A is in the language \mathcal{L} and w is among a denumerable set of labels $\{w, v, u, \dots\}$. In the first-order case, A is in the language \mathcal{Q} , \underline{a} is among a denumerable set of *parameters* $\{\underline{a}, \underline{b}, \underline{c}, \dots\}$, and w is among a denumerable set of labels $\{w, v, u, \dots\}$. We refer to formulae of the forms $w \leq v$ and $\underline{a} \in D_w$ as *relational atoms* (with formulae of the form $\underline{a} \in D_w$ sometimes referred to as *domain atoms*, more specifically) and refer to formulae of the form $w : A$ as *labelled formulae*. Due to the two types of formulae occurring in a labelled sequent, we often use \mathcal{R} to denote relational atoms, and Γ and Δ to denote labelled formulae, thus distinguishing between the two. Labelled sequents are therefore written in a general form as $\mathcal{R}, \Gamma \Rightarrow \Delta$.

Moreover, we take the comma operator to be commutative and associative; for example, we identify the labelled sequent $w \leq u, w : A, \underline{a} \in D_w \Rightarrow v : C, u : B$ with $\underline{a} \in D_w, w \leq u, w : A \Rightarrow u : B, v : C$. This interpretation of comma

$$\begin{array}{c}
 \frac{}{\mathcal{R}, w \leq v, w : p, \Gamma \Rightarrow \Delta, v : p} (id) \quad \frac{\mathcal{R}, w \leq v, v : A, \Gamma \Rightarrow \Delta, v : B}{\mathcal{R}, \Gamma \Rightarrow \Delta, w : A \supset B} (\supset_r)^{\dagger_1} \\
 \\
 \frac{\mathcal{R}, w : A, w : B, \Gamma \Rightarrow \Delta}{\mathcal{R}, w : A \wedge B, \Gamma \Rightarrow \Delta} (\wedge_l) \quad \frac{\mathcal{R}, \Gamma \Rightarrow \Delta, w : A \quad \mathcal{R}, \Gamma \Rightarrow \Delta, w : B}{\mathcal{R}, \Gamma \Rightarrow \Delta, w : A \wedge B} (\wedge_r) \\
 \\
 \frac{\mathcal{R}, w : A, \Gamma \Rightarrow \Delta \quad \mathcal{R}, w : B, \Gamma \Rightarrow \Delta}{\mathcal{R}, w : A \vee B, \Gamma \Rightarrow \Delta} (\vee_l) \quad \frac{\mathcal{R}, w \leq v, v \leq u, w \leq u, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq v, v \leq u, \Gamma \Rightarrow \Delta} (tra) \\
 \\
 \frac{\mathcal{R}, w \leq v, w : A \supset B, \Gamma \Rightarrow \Delta, v : A \quad \mathcal{R}, w \leq v, w : A \supset B, v : B, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq v, w : A \supset B, \Gamma \Rightarrow \Delta} (\supset_l) \\
 \\
 \frac{\mathcal{R}, w \leq w, \Gamma \Rightarrow \Delta}{\mathcal{R}, \Gamma \Rightarrow \Delta} (ref) \quad \frac{\mathcal{R}, \Gamma \Rightarrow \Delta, w : A, w : B}{\mathcal{R}, \Gamma \Rightarrow \Delta, w : A \vee B} (\vee_r) \quad \frac{}{\mathcal{R}, w : \perp, \Gamma \Rightarrow \Delta} (\perp_l) \\
 \\
 \frac{}{\mathcal{R}, w \leq v, \vec{a} \in D_w, w : p(\vec{a}), \Gamma \Rightarrow \Delta, v : p(\vec{a})} (id_q) \quad \frac{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta}{\mathcal{R}, \Gamma \Rightarrow \Delta} (ihd)^{\dagger_2} \\
 \\
 \frac{\mathcal{R}, w \leq v, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, v : A(\underline{a}/x)}{\mathcal{R}, \Gamma \Rightarrow \Delta, w : \forall x A} (\forall_r)^{\dagger_3} \quad \frac{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists x A}{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta, w : \exists x A} (\exists_r) \\
 \\
 \frac{\mathcal{R}, \underline{a} \in D_w, w : A(\underline{a}/x), \Gamma \Rightarrow \Delta}{\mathcal{R}, w : \exists x A, \Gamma \Rightarrow \Delta} (\exists_l)^{\dagger_2} \quad \frac{\mathcal{R}, w \leq v, \underline{a} \in D_v, v : A(\underline{a}/x), w : \forall x A, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq v, \underline{a} \in D_v, w : \forall x A, \Gamma \Rightarrow \Delta} (\forall_l) \\
 \\
 \frac{\mathcal{R}, w \leq v, \underline{a} \in D_w, \underline{a} \in D_v, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq v, \underline{a} \in D_w, \Gamma \Rightarrow \Delta} (nd) \quad \frac{\mathcal{R}, w \leq v, \underline{a} \in D_v, \underline{a} \in D_w, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq v, \underline{a} \in D_v, \Gamma \Rightarrow \Delta} (cd)
 \end{array}$$

Fig. 4. The labelled calculus $\mathbf{G3Int}$ for \mathbf{Int} consists of (id) , (\perp_l) , (\wedge_l) , (\wedge_r) , (\vee_l) , (\vee_r) , (\supset_l) , (\supset_r) , (ref) , and (tra) (see [6]). The labelled calculus $\mathbf{G3IntQ}$ for \mathbf{IntQ} consists of all rules minus the (cd) rule, and all rules give the calculus $\mathbf{G3IntQC}$ for \mathbf{IntQC} . The side condition \dagger_1 states that the variable v does not occur in the conclusion, \dagger_2 states that \underline{a} does not occur in the conclusion, and \dagger_3 states that neither \underline{a} nor v occurs in the conclusion. Labels and parameters restricted from occurring in the conclusion of an inference are called *eigenvariables*.

lets us view \mathcal{R}, Γ (the *antecedent*) and Δ (the *succedent*) of a labelled sequent $\mathcal{R}, \Gamma \Rightarrow \Delta$ as multisets. Also, we use ε to denote the *empty string* which acts as the identity element for the comma operator (e.g. we identify $w \leq v, \varepsilon, v : B$ with $w \leq v, v : B$), and we stipulate that if ε is the antecedent or succedent of a sequent, then the antecedent or succedent is left empty. Therefore, ε will be implicit in labelled sequents, but will never explicitly appear.

In the first-order setting, we syntactically distinguish between *bound variables* $\{x, y, z, \dots\}$ and *free variables*, which are replaced with *parameters* $\{\underline{a}, \underline{b}, \underline{c}, \dots\}$, to avoid clashes between the two categories (cf. [9, Sect. 8]). Therefore, instead of using formulae directly from the first-order language, we use formulae from the first-order language where each freely occurring variable x has been replaced by a distinct parameter \underline{a} . For example, we would make use of the labelled

$$\begin{array}{c}
\frac{\mathcal{R}, \Gamma \Rightarrow \Delta}{\mathcal{R}(w/v), \Gamma(w/v) \Rightarrow \Delta(w/v)} \text{ (lsub)} \quad \frac{\mathcal{R}, \Gamma \Rightarrow \Delta}{\mathcal{R}(\underline{a}/\underline{b}), \Gamma(\underline{a}/\underline{b}) \Rightarrow \Delta(\underline{a}/\underline{b})} \text{ (psub)} \\
\frac{\mathcal{R}, \Gamma \Rightarrow \Delta}{\mathcal{R}', \mathcal{R}, \Gamma', \Gamma \Rightarrow \Delta', \Delta} \text{ (wk)} \quad \frac{\mathcal{R}, \mathcal{R}', \mathcal{R}', \Gamma \Rightarrow \Delta}{\mathcal{R}, \mathcal{R}', \Gamma \Rightarrow \Delta} \text{ (ctr}_{\mathcal{R}})} \quad \frac{\mathcal{R}, \Gamma', \Gamma', \Gamma \Rightarrow \Delta}{\mathcal{R}, \Gamma', \Gamma \Rightarrow \Delta} \text{ (ctr}_l)} \\
\frac{\mathcal{R}, \Gamma \Rightarrow \Delta, \Delta', \Delta'}{\mathcal{R}, \Gamma \Rightarrow \Delta, \Delta'} \text{ (ctr}_r)} \quad \frac{\mathcal{R}, \Gamma \Rightarrow \Delta, w : A \quad \mathcal{R}, w : A, \Gamma \Rightarrow \Delta}{\mathcal{R}, \Gamma \Rightarrow \Delta} \text{ (cut)}
\end{array}$$

Fig. 5. The set LR of admissible labelled rules consists of all rules shown above.

formula $w : (\forall x)p(\underline{a}, x) \vee q(\underline{a}, \underline{b})$ instead of $w : (\forall x)p(y, x) \vee q(y, z)$ in a first-order sequent. Last, we use the notation $A(\underline{a}_1, \dots, \underline{a}_n)$, with $n \in \mathbb{N}$, to denote that the parameters $\underline{a}_1, \dots, \underline{a}_n$ are all parameters occurring in the formula A . (NB. We will occasionally abuse notation and write $A(\underline{a})$ to indicate that the formula A contains a parameter \underline{a} in which we are interested; however, when using this notation we leave open the possibility that A may contain other parameters as well.) We write $A(\vec{\underline{a}})$ as shorthand for $A(\underline{a}_1, \dots, \underline{a}_n)$ and $\vec{\underline{a}} \in D_w$ as shorthand for $\underline{a}_1 \in D_w, \dots, \underline{a}_n \in D_w$. The labelled calculi are given in Fig. 4.

Each labelled calculus is obtained from the models and semantic clauses of **Int**, **IntQ**, and **IntQC**. We note that the labelled calculus **G3Int** for **Int** is presented in [6], whereas the labelled calculi **G3IntQ** and **G3IntQC** for **IntQ** and **IntQC** (respectively) are new. The rules (\perp_l) , (\wedge_l) , (\wedge_r) , (\vee_l) , (\vee_r) , (\supset_l) , (\supset_r) , (\exists_l) , (\exists_r) , (\forall_l) , and (\forall_r) are rule representations of the semantic clauses given in Def. 2 and Def. 5. The *(ref)* and *(tra)* rules allow inferences arising from the fact that frames are reflexive and transitive, whereas *(ihd)*, *(nd)*, and *(cd)* allow inferences based on the fact that the domains of frames are always inhabited (i.e. non-empty), are nested, and are constant, respectively. The rules *(id)* and *(id_q)* encode the monotonicity condition imposed on models; note that *(id)* is an instance of *(id_q)* (the same holds for the corresponding rules in the nested setting, which are introduced in the next subsection). Last, we write $\vdash_{\mathbf{G3Int}} A$, $\vdash_{\mathbf{G3IntQ}} A$, and $\vdash_{\mathbf{G3IntQC}} A$ to indicate that the labelled sequent A is derivable in **G3Int**, **G3IntQ**, and **G3IntQC**, respectively—this notation extends straightforwardly to the other calculi we will consider.

We define a *label substitution* (w/v) on a multiset of labelled formulae or relational atoms in the usual way as the replacement of all labels v occurring in the multiset with the label w . Similarly, we define a *parameter substitution* $(\underline{a}/\underline{b})$ on a multiset of labelled formulae or relational atoms as the replacement of all parameters \underline{b} occurring in the multiset with the parameter \underline{a} .

Our labelled calculi possess desirable proof-theoretic properties such as the *height-preserving (hp-) admissibility* of substitutions and structural rules (e.g. *(psub)* and *(wk)*), the *height-preserving (hp-) invertibility* of all rules, and admissibility of *(cut)*. These properties are detailed in the theorem below (Thm. 3) with the (hp-)admissible rules shown in Fig. 5.

Theorem 3. *Let $\text{G3X} \in \{\text{G3IntQ}, \text{G3IntQC}\}$. The calculi G3Int , G3IntQ , and G3IntQC have the following properties:*

- (i) (a) For all $A \in \mathcal{L}$, $\vdash_{\text{G3Int}} \mathcal{R}, w \leq v, w : A, \Gamma \Rightarrow v : A, \Delta$;
- (b) For all $A \in \mathcal{L}$, $\vdash_{\text{G3Int}} \mathcal{R}, w : A, \Gamma \Rightarrow \Delta, w : A$;
- (c) For all $A \in \mathcal{Q}$, $\vdash_{\text{G3X}} \mathcal{R}, w \leq v, \vec{a} \in D_w, w : A(\vec{a}), \Gamma \Rightarrow v : A(\vec{a}), \Delta$;
- (d) For all $A \in \mathcal{Q}$, $\vdash_{\text{G3X}} \mathcal{R}, \vec{a} \in D_w, w : A(\vec{a}), \Gamma \Rightarrow \Delta, w : A(\vec{a})$;
- (ii) All rules in $\text{LR} - \{\text{cut}\}$ are hp-admissible;
- (iii) All rules are hp-invertible;
- (iv) The (cut) rule is admissible;
- (v) G3Int , G3IntQ , and G3IntQC are sound and complete for Int , IntQ , and IntQC , respectively.

Proof. We refer the reader to [6] for proofs of properties (i)–(v) for G3Int ; note that hp-admissibility of (*psub*) is trivial in the propositional setting since formulae do not contain parameters. The proofs of properties (i)–(v) for G3IntQ and G3IntQC are similar to those for G3Int and can be found in Appendix A. \square

3.2 The Nested Calculi NInt , NIntQ , and NIntQC

We define a propositional (or, first-order) nested sequent Σ to be a syntactic object defined via the following BNF grammars:

$$X ::= \varepsilon \mid A \mid X, X \quad \Sigma ::= X \rightarrow X \mid X \rightarrow X, [\Sigma], \dots, [\Sigma]$$

where A is in the propositional language \mathcal{L} (first-order language \mathcal{Q} , resp.). As in the previous section, we take the comma operator to be commutative and associative, allowing us to view syntactic entities X as multisets, and we let ε represent the empty string which—as in the labelled setting—occurs implicitly, but not explicitly, in sequents.

In the first-order setting, we syntactically distinguish between bound variables and free variables in first-order formulae, using $\{x, y, z, \dots\}$ for bound variables and replacing the occurrence of free variables in formulae with parameters $\{\underline{a}, \underline{b}, \underline{c}, \dots\}$. For example, we would use $p(\underline{a}) \rightarrow p(\underline{b}), [\perp \rightarrow \forall xq(x, \underline{b})]$ instead of the sequent $p(x) \rightarrow p(y), [\perp \rightarrow \forall xq(x, y)]$ in a nested derivation (where the free variable x has been replaced by the parameter \underline{a} and y has been replaced by \underline{b}).

Nested sequents are often written as $\Sigma\{X \rightarrow Y, [\Sigma_1], \dots, [\Sigma_n]\}$, which indicates that $X \rightarrow Y, [\Sigma_1], \dots, [\Sigma_n]$ occurs at some depth in the nestings of the sequent Σ . For example, if Σ is taken to be $p(\underline{a}) \rightarrow [\perp \rightarrow \forall xq(x, \underline{b}), [\rightarrow \top]]$, then $\Sigma\{\perp \rightarrow \forall xq(x, \underline{b})\}$, $\Sigma\{\perp \rightarrow \forall xq(x, \underline{b}), [\rightarrow \top]\}$, and $\Sigma\{\rightarrow \top\}$ are correct representations of Σ in our notation.

The nested calculi are given in Fig. 6 and are *slight* variants of the calculi presented by Fitting in [9]. The only difference is that the rules (\neg_l) , (*lift*), (\supset_l) , (\exists_r) , and (\forall_l) preserve copies of the principal formula in the premise of the rule. Since we will be extracting Fitting’s nested calculi from our labelled calculi, it will be seen that these copies of principal formulae are residua of this process, which is based on the fact that our labelled calculi preserve copies of principal

$$\begin{array}{c}
\frac{}{\Sigma\{X, p \rightarrow p, Y\}} (id) \quad \frac{\Sigma\{X, A, B \rightarrow Y\}}{\Sigma\{X, A \wedge B \rightarrow Y\}} (\wedge_i) \quad \frac{\Sigma\{X, A \rightarrow Y, [X', A \rightarrow Y']\}}{\Sigma\{X, A \rightarrow Y, [X' \rightarrow Y']\}} (lift) \\
\\
\frac{\Sigma\{X, A \rightarrow Y\} \quad \Sigma\{X, B \rightarrow Y\}}{\Sigma\{X, A \vee B \rightarrow Y\}} (\vee_i) \quad \frac{\Sigma\{X \rightarrow A, Y\} \quad \Sigma\{X \rightarrow B, Y\}}{\Sigma\{X \rightarrow A \wedge B, Y\}} (\wedge_r) \\
\\
\frac{\Sigma\{X \rightarrow Y, [A \rightarrow]\}}{\Sigma\{X \rightarrow Y, \neg A\}} (\neg_r) \quad \frac{\Sigma\{X, \neg A \rightarrow A, Y\}}{\Sigma\{X, \neg A \rightarrow Y\}} (\neg_l) \quad \frac{\Sigma\{X \rightarrow A, B, Y\}}{\Sigma\{X \rightarrow A \vee B, Y\}} (\vee_r) \\
\\
\frac{\Sigma\{X \rightarrow Y, [A \rightarrow B]\}}{\Sigma\{X \rightarrow A \supset B, Y\}} (\supset_r) \quad \frac{\Sigma\{X, A \supset B \rightarrow A, Y\} \quad \Sigma\{X, A \supset B, B \rightarrow Y\}}{\Sigma\{X, A \supset B \rightarrow Y\}} (\supset_l) \\
\\
\frac{}{\Sigma\{X, p(\underline{a}) \rightarrow p(\underline{a}), Y\}} (id_q) \quad \frac{\Sigma\{X \rightarrow A(\underline{a}/x), \exists x A, Y\}}{\Sigma\{X \rightarrow \exists x A, Y\}} (\exists_r)^{\dagger_1} \\
\\
\frac{\Sigma\{X, A(\underline{a}/x) \rightarrow Y\}}{\Sigma\{X, \exists x A \rightarrow Y\}} (\exists_l)^{\dagger_2} \quad \frac{\Sigma\{X \rightarrow Y, [\rightarrow A(\underline{a}/x)]\}}{\Sigma\{X \rightarrow \forall x A, Y\}} (\forall_r^n)^{\dagger_2} \\
\\
\frac{\Sigma\{X \rightarrow A(\underline{a}/x), Y\}}{\Sigma\{X \rightarrow \forall x A, Y\}} (\forall_r^c)^{\dagger_2} \quad \frac{\Sigma\{X, \forall x A, A(\underline{a}/x) \rightarrow Y\}}{\Sigma\{X, \forall x A \rightarrow Y\}} (\forall_l)^{\dagger_1}
\end{array}$$

Fig. 6. The nested calculus **NInt** for propositional intuitionistic logic consists of (id) , (\wedge_l) , (\wedge_r) , (\vee_r) , (\vee_l) , (\neg_l) , (\neg_r) , (\supset_r) , (\supset_l) , and $(lift)$. The nested calculus **NIntQ** extends **NInt** with (id_q) , (\exists_l) , (\exists_r) , (\forall_l) , and (\forall_r^n) . The nested calculus **NIntQC** extends **NInt** with (id_q) , (\exists_l) , (\exists_r) , (\forall_l) , and (\forall_r^c) and omits the side condition \dagger_1 on the (\exists_r) and (\forall_l) rules [9]. The side condition \dagger_1 states that the parameter \underline{a} is either available or is an eigenvariable, and \dagger_2 states that \underline{a} is an eigenvariable.

formulae in the premise(s) of certain rules. Howbeit, it is easy to confirm that the calculi **NInt**, **NIntQ**, and **NIntQC** are equivalent to Fitting's calculi from [9], as it is straightforward to transform a derivation from one calculus into a derivation in the other, and vice-versa.

The two distinguishing factors between the nested calculi **NIntQ** and **NIntQC** are that (i) the former uses the (\forall_r^n) rule whereas the latter uses the (\forall_r^c) rule, and (ii) the former imposes a side condition (see Fig. 6) on the (\exists_r) and (\forall_l) rules, whereas the latter does not. This side condition ensures the soundness of the (\exists_r) and (\forall_l) rules with respect to **IntQ**-models, that is, with respect to models where domains are *not necessarily constant* (cf. [9]). The side condition relies on the notion of an *available parameter*, which we define below:

Definition 6 (Available Parameter [9]). *Let $\Sigma\{X \rightarrow Y, [\Sigma_1], \dots, [\Sigma_n]\}$ be a nested sequent. If there exists a formula $A(\underline{a}) \in X, Y$, then the parameter \underline{a} is available in $X \rightarrow Y$ and in all boxed subsequents Σ_i (with $i \in \{1, \dots, n\}$).*

As explained in the following section, a nested sequent can be seen as a tree of sequents of the form $X \rightarrow Y$. Intuitively, if a parameter is available at some

point $X \rightarrow Y$ in the tree, then it will be available at all points in the future of that point, i.e. in the subtree with $X \rightarrow Y$ as its root. Since a nested sequent can be seen as an abstraction of an IntQ -model encoding the model's underlying treelike structure, the notion of an available parameter corresponds to the fact that if a parameter denotes an object in the domain of a world, then—due to the nested domain condition (Def. 3)—that parameter denotes that object at all future worlds and is *available* for use in sentences at such worlds.

A feature which distinguishes the nested formalism from the labelled is that nested sequents may readily be converted into an *equivalent formula*, that is, nested sequents allow for a straightforward formula translation (see Def. 7 below). As a consequence of our refinement procedure, we will see in Sect. 8 (viz. Cor. 4) that a nested sequent Σ is provable if and only if its formula interpretation is provable.

Definition 7 (Formula Interpretation). *The formula interpretation of a nested sequent is defined inductively as follows:*

$$\begin{aligned} \iota(X \rightarrow Y) &:= \bigwedge X \supset \bigvee Y \\ \iota(X \rightarrow Y, [\Sigma_1], \dots, [\Sigma_n]) &:= \bigwedge X \supset \left(\bigvee Y \vee \bigvee_{1 \leq i \leq n} \iota(\Sigma_i) \right) \end{aligned}$$

In the propositional setting, $\iota(\Sigma)$ is a formula in \mathcal{L} . In the first-order setting, we interpret a nested sequent Σ as a formula in \mathcal{Q} by taking the universal closure $\forall \vec{x} \iota(\Sigma)$ of $\iota(\Sigma)$. Also, as usual $\bigwedge \varepsilon := \top$ and $\bigvee \varepsilon := \perp$, i.e. empty antecedents translate to \top and empty succedents translate to \perp .

Proof-theoretic properties of NInt , NIntQ , and NIntQC will be discussed in Sect. 8, though we do state the essential properties of soundness and completeness below, which were proven by Fitting in [9]. By *strongly* sound and complete, we mean that a nested sequent Σ is valid (relative to an Int -, IntQ -, or IntQC -model) iff it is derivable (in NInt , NIntQ , or NIntQC , respectively).

Theorem 4 (Soundness and Completeness). *NInt , NIntQ , and NIntQC are strongly sound and complete for Int , IntQ , and IntQC , respectively.*

Interestingly, the correspondence we establish between G3Int and NInt , and G3IntQC and NIntQC , can be leveraged to show that NInt and NIntQC inherit soundness and completeness from their associated labelled calculi. The correspondence between NIntQ and G3IntQ cannot be leveraged to conclude the completeness of NIntQ however, because of the use of a lemma (Lem. 14 in Sect. 6) that invokes the strong completeness of NIntQ (i.e. it invokes Thm. 4 above). Still, we note in that section how the lemma may be proved independent of the completeness of NIntQ , thus allowing for NIntQ to inherit the completeness of G3IntQ via our refinement and translation processes. Last, in Sect. 8, we will show that our nested calculi inherit additional properties from their corresponding labelled calculi such as the admissibility of certain structural rules (Cor. 2) and the invertibility of all rules (Cor. 3).

4 Fundamentals for Establishing Correspondence

This section consists of two parts: in the first subsection, we define translation functions that transform labelled sequents into nested sequents and vice-versa, as well as define classes of labelled sequents that are fruitful for carrying out our proof-theoretic transformation and translation work. In the second subsection, we establish preliminary results that are convenient for refining our labelled calculi, that is, such results will assist us in eliminating the structural rules (e.g. (*ref*) and (*nd*)) from **G3Int**, **G3IntQ**, and **G3IntQC** (which will ultimately yield systems that are close variants of Fitting's nested systems).

4.1 Translating Notation: Labelled and Nested

It is instructive to observe that both nested and labelled sequents can be viewed as graphs (with the former restricted to trees and the latter more general). Graphs of sequents are significant for two reasons: the first (technical) reason is that graphs can be leveraged to switch from labelled to nested notation; thus, graphs will play a role in deriving our nested calculi from our labelled calculi, and vice-versa. Second, graphs offer insight into *why* structural rule elimination yields nested systems, which will be discussed in the next section (Sect. 5).

It is straightforward to define the graph of each type of sequent. To do this, we first introduce a bit of notation and define the multiset $\Gamma \uparrow w := \{A \mid w : A \in \Gamma\}$. For a labelled sequent $\Lambda := \mathcal{R}, \Gamma \Rightarrow \Delta$, the *graph* $G(\Lambda)$ is the tuple (V, E, λ) , where (i) $V := \{w \mid w \text{ is a label in } \Lambda\}$, (ii) $(w, v) \in E$ iff $w \leq v \in \mathcal{R}$, and

$$(iii) \quad \lambda := \{(w, \Gamma' \Rightarrow \Delta') \mid \Gamma' = \Gamma \uparrow w, \Delta' = \Delta \uparrow w, \text{ and } w \in V\}.$$

For a nested sequent, the graph is defined inductively on the structure of the nestings; we use strings σ of natural numbers to denote vertices in the graph, similar to the prefixes used in prefixed tableaux [7,8,9].

Base case. Let our nested sequent be of the form $X \rightarrow Y$ with X and Y multisets of formulae. Then, $G_\sigma(X \rightarrow Y) := (V_\sigma, E_\sigma, \lambda_\sigma)$, where (i) $V_\sigma := \{\sigma\}$, (ii) $E_\sigma := \emptyset$, and (iii) $\lambda_\sigma := \{(\sigma, X \rightarrow Y)\}$.

Inductive step. Suppose our nested sequent is of the form $X \rightarrow Y, [\Sigma_1], \dots, [\Sigma_n]$. We assume that each $G_{\sigma.i}(\Sigma_i) = (V_{\sigma.i}, E_{\sigma.i}, \lambda_{\sigma.i})$ (with $i \in \{1, \dots, n\}$) is already defined, and define $G_\sigma(X \rightarrow Y, [\Sigma_1], \dots, [\Sigma_n]) := (V_\sigma, E_\sigma, \lambda_\sigma)$ as follows:

$$(i) \quad V_\sigma := \{\sigma\} \cup \bigcup_{1 \leq i \leq n} V_{\sigma.i} \quad (ii) \quad E_\sigma := \{(\sigma, \sigma.i) \mid 1 \leq i \leq n\} \cup \bigcup_{1 \leq i \leq n} E_{\sigma.i}$$

$$(iii) \quad \lambda_\sigma := \{(\sigma, X \rightarrow Y)\} \cup \bigcup_{1 \leq i \leq n} \lambda_{\sigma.i}$$

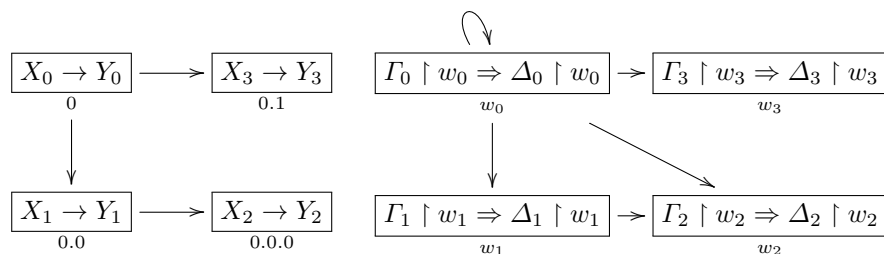
We will occasionally refer to a sequent of the form $X \rightarrow Y$ in a nested sequent Σ such that $(\sigma, X \rightarrow Y) \in \lambda_0$ (with $G_0(\Sigma) = (V_0, E_0, \lambda_0)$) as a *component* of the nested sequent Σ .

Definition 8. Let $G_0 = (V_0, E_0, \lambda_0)$ and $G_1 = (V_1, E_1, \lambda_1)$ be two graphs. We define an isomorphism $f : V_0 \mapsto V_1$ between G_0 and G_1 to be a function such that: (i) f is bijective, (ii) $(x, y) \in E_0$ iff $(fx, fy) \in E_1$, (iii) $\lambda_0(x) = \lambda_1(fx)$. We say G_0 and G_1 are isomorphic iff there exists an isomorphism between them.

Although the formal definitions above may appear somewhat cumbersome, the example below shows that transforming a sequent into its graph—or conversely, obtaining the sequent from its graph—is relatively straightforward.

Example 1. The nested sequent Σ is given below with its corresponding graph $G_0(\Sigma)$ shown on the left, and the labelled sequent Λ is given below with its corresponding graph $G(\Lambda)$ on the right. Regarding the labelled sequent, we assume that Γ_i and Δ_i consist solely of formulae labelled with w_i (for $i \in \{0, 1, 2, 3\}$).

$$\Sigma = X_0 \rightarrow Y_0, [X_1 \rightarrow Y_1, [X_2 \rightarrow Y_2]], [X_3 \rightarrow Y_3]$$



$$\Lambda = w_0 \leq w_0, w_0 \leq w_1, w_1 \leq w_2, w_0 \leq w_2, w_0 \leq w_3, \Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3 \Rightarrow \Delta_0, \Delta_1, \Delta_2, \Delta_3$$

In the above example there is a loop from w_0 to itself in the graph of the labelled sequent; furthermore, there is an undirected cycle occurring between w_0 , w_1 , and w_2 . As will be explained in the next section (specifically, Thm. 6), the (*ref*) and (*tra*) rules allow for such structures to appear in the graphs of labelled sequents used to derive theorems; however, the elimination of these rules in our labelled calculi has the effect that such structures *can no longer* occur in the labelled derivation of a theorem. Consequently, it will be seen that eliminating such rules yields a labelled derivation where every sequent has a purely *treelike* structure (cf. [13]). This implies that each labelled sequent in the derivation has a graph isomorphic to the graph of a nested sequent. It is this idea which ultimately permits the extraction of our nested calculi from our labelled calculi.

Definition 9. Let Λ be a labelled sequent and $G(\Lambda) = (V, E, \lambda)$. We say that Λ is *treelike* iff there exists a unique vertex $w \in V$, called the root, such that there exists a unique path from w to every other vertex $v \in V$.³ We say that a labelled derivation is *treelike* iff every labelled sequent in the derivation is *treelike*.

If we take the graph of a treelike labelled sequent, then it can be viewed as the graph of a nested sequent, as the example below demonstrates:

³Treelike sequents are equivalently characterized as sequents with graphs that are: (i) connected, (ii) acyclic, and (iii) contain no backwards branching.

Example 2. The treelike labelled sequent Λ' and its graph are given below. We assume that Γ_i and Δ_i contain only formulae labelled with w_i (for $i \in \{0, 1, 2, 3\}$).

$$\Lambda' = w_0 \leq w_1, w_1 \leq w_2, w_0 \leq w_3, \Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3 \Rightarrow \Delta_0, \Delta_1, \Delta_2, \Delta_3$$

$$\boxed{\Gamma'_2 \Rightarrow \Delta'_2}_{w_2} \longleftarrow \boxed{\Gamma'_1 \Rightarrow \Delta'_1}_{w_1} \longleftarrow \boxed{\Gamma'_0 \Rightarrow \Delta'_0}_{w_0} \longrightarrow \boxed{\Gamma'_3 \Rightarrow \Delta'_3}_{w_3}$$

Also, if we assume that $\Gamma'_i = \Gamma_i \upharpoonright w_i = X_i$ and $\Delta'_i = \Delta_i \upharpoonright w_i = Y_i$ (for $i \in \{0, 1, 2, 3\}$), then the above graph is isomorphic to the graph of the nested sequent in Example 1, meaning that Λ can be translated as that nested sequent.

To make the above translation precise, we introduce the *w-downward closure* of the graph of a labelled sequent. We may utilize this notion to explicitly define the *translation* \mathfrak{N} (see Def. 11) that translates labelled sequents into nested sequents. This translation will prove itself useful in extracting Fitting's nested calculi from our labelled calculi.

Definition 10 (Downward Closure). *Let Λ be a treelike labelled sequent with graph $G(\Lambda) = (V, E, \lambda)$ and $w \in V$. We define the w-downward closure $G_w(\Lambda) = (V', E', \lambda')$ to be the smallest subgraph of $G(\Lambda)$ such that $w \in V'$ and*

- ▶ if $v \in V'$ and $(v, u) \in E$, then $u \in V'$;
- ▶ $E' = \{(u, v) \mid u, v \in V'\}$;
- ▶ $\lambda' = \{(v, \Gamma' \Rightarrow \Delta') \mid v \in V' \text{ and } (v, \Gamma' \Rightarrow \Delta') \in \lambda\}$.

Definition 11 (The Translation \mathfrak{N}). *Let Λ be a treelike labelled sequent with root w . We inductively define $\mathfrak{N}(\Lambda) := \mathfrak{N}(G_w(\Lambda))$ to be the nested sequent obtained from the graph $G_w(\Lambda)$ as follows:*

- ▶ If $G_u(\Lambda) = (V, E, \lambda)$, where $V = \{u\}$, $E = \emptyset$, and $\lambda = \{(u, \Gamma' \Rightarrow \Delta')\}$, then

$$\mathfrak{N}(G_u(\Lambda)) := \Gamma' \rightarrow \Delta'.$$

- ▶ If $G_u(\Lambda) = (V, E, \lambda)$, where $u, v_1, \dots, v_n \in V$, $(u, v_1), \dots, (u, v_n) \in E$, and $(u, \Gamma' \Rightarrow \Delta') \in \lambda$, then

$$\mathfrak{N}(G_u(\Lambda)) := \Gamma' \rightarrow \Delta', [\mathfrak{N}(G_{v_1}(\Lambda))], \dots, [\mathfrak{N}(G_{v_n}(\Lambda))].$$

We also define a converse translation \mathfrak{L} that translates nested sequents into labelled sequents. Whereas the former translation \mathfrak{N} is useful in extracting our nested calculi from our labelled calculi, the translation \mathfrak{L} is useful in transferring proof-theoretic properties of the latter to the former as well as establishing the converse translation (see Sect. 8). In order to define this translation, we first define two operations: (i) If X is a multiset of formulae from \mathcal{L} or \mathcal{Q} , then $w : X := \{w : A \mid A \in X\}$, that is, $w : X$ is the multiset of all formulae from X labelled with w , and (ii) If $\Lambda_1 = \mathcal{R}_1, \Gamma_1 \Rightarrow \Delta_1$ and $\Lambda_2 = \mathcal{R}_2, \Gamma_2 \Rightarrow \Delta_2$, then we define the *sequent composition* $\Lambda_1 \oplus \Lambda_2 := \mathcal{R}_1, \mathcal{R}_2, \Gamma_1, \Gamma_2 \Rightarrow \Delta_1, \Delta_2$.

Definition 12 (The Translation \mathfrak{L}). Let Σ be a nested sequent. We inductively define $\mathfrak{L}(\Sigma) := \mathfrak{L}(G_0(\Sigma))$ to be the labelled sequent obtained from the graph $G_0(\Sigma)$ as follows:

- If $G_\sigma(\Sigma) = (V_\sigma, E_\sigma, \lambda_\sigma)$ with $V = \{\sigma\}$, $E = \emptyset$, $\lambda = \{(\sigma, X \rightarrow Y)\}$, and \vec{a} are all parameters occurring in X, Y , then

$$\mathfrak{L}(G_\sigma(\Sigma)) := \vec{a} \in D_\sigma, \sigma : X \Rightarrow \sigma : Y.$$

- If $G_\sigma(\Sigma) = (V_\sigma, E_\sigma, \lambda_\sigma)$ with $\sigma, \sigma.1, \dots, \sigma.n \in V$, $(\sigma, \sigma.1), \dots, (\sigma, \sigma.n) \in E$, $(\sigma, X \rightarrow Y) \in \lambda$, and \vec{a} are all parameters occurring in X, Y , then

$$\begin{aligned} \mathfrak{L}(G_\sigma(\Sigma)) := & \left(\sigma \leq \sigma.1, \dots, \sigma \leq \sigma.n, \vec{a} \in D_\sigma, \sigma : X \Rightarrow \sigma : Y \right) \\ & \oplus \left(\mathfrak{L}(G_{\sigma.1}(\Sigma)) \right) \oplus \dots \oplus \left(\mathfrak{L}(G_{\sigma.n}(\Sigma)) \right). \end{aligned}$$

To simplify notation when making use of the \mathfrak{L} translation, we will use the same labels w, u, v, \dots employed in our labelled calculi in place of strings σ of natural numbers.

Before finishing this subsection, we introduce a fundamental notion useful for completing our translation from labelled to nested, which is based on the above definition—the notion of a *nestedlike* labelled sequent:

Definition 13 (Nestedlike). We say that a labelled sequent Λ is nestedlike iff there exists a nested sequent Σ such that $\mathfrak{L}(\Sigma) = \Lambda$.

It is not difficult to see that the following lemma holds:

Lemma 3. Let $\Lambda := \mathcal{R}, \Gamma \Rightarrow \Delta$ be a nestedlike labelled sequent. Then, (i) Λ is treelike, (ii) $\mathfrak{L}(\mathfrak{N}(\Lambda)) = \Lambda$ (up to a change in labels), and (iii) there exists a domain atom $\underline{a} \in D_w \in \mathcal{R}$ iff there exists a labelled formula $w : A(\underline{a}) \in \Gamma, \Delta$.

4.2 Setting the Stage for Refinement: Preliminary Results

In order to extract Fitting’s nested calculi from our labelled calculi, we will expand our labelled calculi with rules sufficient for the elimination of certain structural rules. Therefore, we first inflate our calculi to a large collection of rules, and use the new additions to systematically eliminate certain rules from each calculus, thus obtaining a refined variant of each calculus. To avoid lengthy names for our inflated and refined calculi, we will use the following abbreviations throughout the remainder of the paper (see Fig. 7 for newly introduced rules):

Definition 14. The inflated (G3Int^* , G3IntQ^* , and G3IntQC^*) and refined (IntL , IntQL , and IntQCL) labelled calculi are defined as follows:

$$\begin{aligned} \text{G3Int}^* &:= \text{G3Int} + \{(id^*), (\neg_l), (\neg_r), (\supset_l^*), (lift)\} \\ \text{G3IntQ}^* &:= \text{G3IntQ} + \{(id_q^n), (\neg_l), (\neg_r), (\supset_l^*), (lift), (\exists_r^n), (\forall_l^n), (\exists_r^{in}), (\forall_l^{in})\} \\ \text{G3IntQC}^* &:= \text{G3IntQC} + \{(id_q^c), (\neg_l), (\neg_r), (\supset_l^*), (lift), (\exists_r^c), (\forall_l^c), (\exists_r^{ic}), (\forall_l^{ic}), (\forall_r^c)\} \end{aligned}$$

$$\begin{array}{c}
\frac{}{\mathcal{R}, \Gamma, w : p \Rightarrow w : p, \Delta} (id^*) \frac{}{\mathcal{R}, \underline{a}_1 \in D_{v_1}, \dots, \underline{a}_n \in D_{v_n}, \Gamma, w : p(\vec{\underline{a}}) \Rightarrow w : p(\vec{\underline{a}}), \Delta} (id_q^n)^{\dagger_1} \\
\frac{}{\mathcal{R}, \underline{a}_1 \in D_{v_1}, \dots, \underline{a}_n \in D_{v_n}, \Gamma, w : p(\vec{\underline{a}}) \Rightarrow w : p(\vec{\underline{a}}), \Delta} (id_q^c)^{\dagger_2} \\
\frac{\mathcal{R}, w \leq v, v : A, \Gamma \Rightarrow \Delta}{\mathcal{R}, \Gamma \Rightarrow \Delta, w : \neg A} (\neg_r)^{\dagger_3} \quad \frac{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists x A}{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \exists x A} (\exists_r^n)^{\dagger_4} \\
\frac{\mathcal{R}, w : \neg A, \Gamma \Rightarrow w : A, \Delta}{\mathcal{R}, w : \neg A, \Gamma \Rightarrow \Delta} (\neg_l) \quad \frac{\mathcal{R}, \underline{a} \in D_v, w : A(\underline{a}/x), w : \forall x A, \Gamma \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_v, w : \forall x A, \Gamma \Rightarrow \Delta} (\forall_l^n)^{\dagger_4} \\
\frac{\mathcal{R}, \underline{a} \in D_v, w : A(\underline{a}/x), w : \forall x A, \Gamma \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_v, w : \forall x A, \Gamma \Rightarrow \Delta} (\forall_l^c)^{\dagger_5} \quad \frac{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow w : A(\underline{a}/x), \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \forall x A, \Delta} (\forall_r^c)^{\dagger_6} \\
\frac{\mathcal{R}, w \leq u, w : A, u : A, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq u, w : A, \Gamma \Rightarrow \Delta} (lift) \quad \frac{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists x A}{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \exists x A} (\exists_r^c)^{\dagger_5} \\
\frac{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists x A, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \exists x A, \Delta} (\exists_r^{in})^{\dagger_7} \quad \frac{\mathcal{R}, \underline{a} \in D_v, \Gamma, w : \forall x A, w : A(\underline{a}/x) \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \forall x A, \Rightarrow \Delta} (\forall_l^{in})^{\dagger_7} \\
\frac{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists x A, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \exists x A, \Delta} (\exists_r^{ic})^{\dagger_8} \quad \frac{\mathcal{R}, \underline{a} \in D_v, \Gamma, w : \forall x A, w : A(\underline{a}/x) \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \forall x A, \Rightarrow \Delta} (\forall_l^{ic})^{\dagger_8} \\
\frac{\mathcal{R}, w : A \supset B, \Gamma \Rightarrow \Delta, w : A}{\mathcal{R}, w : A \supset B, \Gamma \Rightarrow \Delta} (\supset_i^*)
\end{array}$$

Fig. 7. Rules used to derive NInt, NIntQ, and NIntQC from G3Int, G3IntQ, and G3IntQC, respectively. Let \mathcal{R} be the relational atoms occurring in the relevant rule; the side condition \dagger_1 states that $v_i \rightsquigarrow^{\mathcal{R}} w$ for each $i \in \{1, \dots, n\}$, \dagger_2 states that $v_i \sim^{\mathcal{R}} w$ for each $i \in \{1, \dots, n\}$, \dagger_3 states that v is an eigenvariable, \dagger_4 states that $v \rightsquigarrow^{\mathcal{R}} w$, \dagger_5 states that $v \sim^{\mathcal{R}} w$, \dagger_6 states that \underline{a} is an eigenvariable, \dagger_7 states that $v \rightsquigarrow^{\mathcal{R}} w$ and \underline{a} is an eigenvariable, and \dagger_8 states that $v \sim^{\mathcal{R}} w$ and \underline{a} is an eigenvariable.

$$\text{IntL} := \text{G3Int}^* - \{(id), (\perp_l), (\supset_l), (ref), (tra)\}$$

$$\text{IntQL} := \text{G3IntQ}^* - \{(id_q), (\perp_l), (\supset_l), (ref), (tra), (\exists_r), (\forall_l), (nd), (ihd)\}$$

$$\text{IntQCL} :=$$

$$\text{G3IntQC}^* - \{(id_q), (\perp_l), (\supset_l), (ref), (tra), (\exists_r), (\forall_l), (\forall_r), (nd), (cd), (ihd)\}$$

The next three sections will focus on the non-trivial refinement and translation of our labelled calculi into our nested calculi. As such, we present two useful lemmata below which will aid us in the aforementioned endeavor and allow for us to focus our attention on the refinement and translation from labelled to nested, thus freeing us from being sidetracked by too many auxiliary details. Moreover, we present the rules (mentioned in the abbreviations above and lemmata below) that we expand each labelled calculus with in Fig. 7 below. Some of these rules depend on the notion of a *directed path* or *undirected path*, whose definitions are as follows:

Definition 15 (Directed Path). Let $\Lambda = \mathcal{R}, \Gamma \Rightarrow \Delta$ be a labelled sequent. We say that there exists a directed path from w to u in \mathcal{R} (written $w \rightsquigarrow^{\mathcal{R}} u$) iff $w = u$, or there exist worlds v_i (with $i \in \{1, \dots, n\}$) such that $w \leq v_1, \dots, v_n \leq u \in \mathcal{R}$ holds (we stipulate that $w \leq u \in \mathcal{R}$ when $n = 0$).

Definition 16 (Undirected Path). Let $\Lambda = \mathcal{R}, \Gamma \Rightarrow \Delta$ be a labelled sequent and $w \sim v \in \{w \leq u, u \leq w\}$. We say that there exists an undirected path from w to u in \mathcal{R} (written $w \sim^{\mathcal{R}} u$) iff $w = u$, or there exist worlds v_i (with $i \in \{1, \dots, n\}$) such that $w \sim v_1, \dots, v_n \sim u \in \mathcal{R}$ holds (we stipulate that $w \sim u \in \mathcal{R}$ when $n = 0$).

The following lemma confirms that the expansion of our labelled calculi **G3Int**, **G3IntQ**, and **G3IntQC** with rules from Fig. 7 preserves (to a degree) favorable proof-theoretic properties:

Lemma 4. Let $\text{G3X}^* \in \{\text{G3IntQ}^*, \text{G3IntQC}^*\}$. The calculi **G3Int**^{*}, **G3IntQ**^{*}, and **G3IntQC**^{*} have the following properties:

- (i) (a) For all $A \in \mathcal{L}$, $\vdash_{\text{G3Int}^*} \mathcal{R}, w \leq v, w : A, \Gamma \Rightarrow v : A, \Delta$;
- (b) For all $A \in \mathcal{L}$, $\vdash_{\text{G3Int}^*} \mathcal{R}, w : A, \Gamma \Rightarrow \Delta, w : A$;
- (c) For all $A \in \mathcal{Q}$, $\vdash_{\text{G3X}^*} \mathcal{R}, w \leq v, \vec{a} \in D_w, w : A(\vec{a}), \Gamma \Rightarrow v : A(\vec{a}), \Delta$;
- (d) For all $A \in \mathcal{Q}$, $\vdash_{\text{G3X}^*} \mathcal{R}, \vec{a} \in D_w, w : A(\vec{a}), \Gamma \Rightarrow \Delta, w : A(\vec{a})$;
- (ii) The rules $\{(lsub), (psub), (wk), (ctr_{\mathcal{R}}), (ctr_r)\}$ are hp-admissible;
- (iii) With the exception of $\{(\wedge_l), (\exists_l)\}$, all rules are hp-invertible;
- (iv) The rules $\{(\wedge_l), (\exists_l)\}$ are invertible;
- (v) The rule (ctr_l) is admissible.

Proof. See Appendix A for details. □

The following lemma is useful in that it explicates what rules freely permute with each other, thus letting us focus only on the non-trivial cases in the sequel.

Lemma 5. The following hold in **G3Int**^{*}, **G3IntQ**^{*}, and **G3IntQC**^{*}:

- (i) The (ref) and (tra) rules can be permuted above (id^*) , (\perp_l) , (\wedge_l) , (\wedge_r) , (\vee_l) , (\vee_r) , (\supset_l^*) , (\supset_r) , (\neg_l) , (\neg_r) , (ihd) , (\exists_l) , (\exists_r) , and (\forall_r^c) .
- (ii) The (nd) , (cd) , and (ihd) rules can be permuted above (id) , (id^*) , (\perp_l) , (\wedge_l) , (\wedge_r) , (\vee_l) , (\vee_r) , (\supset_l) , (\supset_l^*) , (\supset_r) , (\neg_l) , (\neg_r) , $(lift)$, (ihd) , (\exists_l) , (\exists_r^{in}) , (\exists_r^{ic}) , (\forall_l^{in}) , (\forall_l^{ic}) , (\forall_r) , and (\forall_r^c) .

Proof. The claim of (i) follows from the fact that none of the rules mentioned have active relational atoms of the form $w \leq u$ in the conclusion, and so, (ref) and (tra) may be freely permuted above each rule. The claim of (ii) follows from the fact that none of the rules mentioned contain active domain atoms in the conclusion, allowing for (nd) , (cd) , and (ihd) to be permuted above each rule. □

5 Deriving the Calculus NInt from G3Int

Deriving the calculus NInt from G3Int depends on a crucial observation concerning labelled derivations: *rules such as (ref) and (tra) allow for theorems to be derived in proofs containing non-treelike labelled sequents*. To demonstrate this fact, observe the following derivation in G3Int:

$$\frac{\frac{w \leq v, v \leq v, v : p \Rightarrow v : p}{w \leq v, v : p \Rightarrow v : p} (ref)}{\Rightarrow w : p \supset p} (\supset_r)$$

The initial sequent is non-treelike due to the presence of the $v \leq v$ relational atom; however, the application of *(ref)* deletes this structure from the initial sequent and produces a treelike sequent as the conclusion.

In fact, it is true in general that every labelled derivation ending with a treelike sequent can be partitioned into a top derivation consisting of non-treelike sequents, and a bottom derivation consisting of treelike sequents. Note that if a derivation ends with a treelike sequent, then the derivation must necessarily contain a bottom treelike fragment. By contrast, the top non-treelike fragment of the derivation may be empty (e.g. the minimal derivation of $\Rightarrow w : \perp \supset p$).

To demonstrate why the aforementioned partition always exists, suppose you are given a labelled derivation of a treelike sequent and consider the derivation in a bottom-up manner. Observe the each bottom-up application of a rule in G3Int—with the exception of *(ref)* and *(tra)*—will produce a treelike sequent (see Thm. 6 for auxiliary details). If, however, at some point in the derivation *(ref)* or *(tra)* is applied, then all sequents above the inference will inherit the (un)directed cycle produced by the rule, thus producing the non-treelike fragment of the proof.

One can therefore imagine that permuting instances of the *(ref)* and *(tra)* rules upward in a given derivation would potentially increase the bottom treelike fragment of the derivation and decrease the top non-treelike fragment. As it so happens, this intuition is correct so long as we choose *adequate* rules—that bottom-up preserve the treelike structure of sequents—to replace certain instances of the *(ref)* and *(tra)* rules in a derivation, when necessary. We will first examine permuting instances of the *(ref)* rule, and motivate which adequate rules we ought to add to our calculus in order to achieve the complete elimination of *(ref)*. After, we will turn our attention toward eliminating the *(tra)* rule, and conclude the section by leveraging our results to show that NInt can be derived from G3Int.

Let us first observe an application of *(ref)* to an initial sequent obtained via the *(id)* rule. There are two possible cases to consider: either the relational atom principal in the initial sequent is active in the *(ref)* inference (shown below left), or it is not (shown below right):

$$\frac{\frac{\mathcal{R}, w \leq w, w : p, \Gamma \Rightarrow \Delta, w : p}{\mathcal{R}, w : p, \Gamma \Rightarrow \Delta, w : p} (id)}{\mathcal{R}, w : p, \Gamma \Rightarrow \Delta, w : p} (ref) \quad \frac{\frac{\mathcal{R}, u \leq u, w \leq v, w : p, \Gamma \Rightarrow \Delta, v : p}{\mathcal{R}, w \leq v, w : p, \Gamma \Rightarrow \Delta, v : p} (id)}{\mathcal{R}, w \leq v, w : p, \Gamma \Rightarrow \Delta, v : p} (ref)}$$

In the case shown above right, the end sequent is an instance of the (id) rule, regardless of if $u = w$, $u = v$, or u is distinct from w and v . The case shown above left however, indicates that we ought to add the (id^*) rule (see Fig. 7) to our calculus if we aim to eliminate (ref) from any given derivation. These facts, coupled with Lem. 5, imply that any application of (ref) to an initial sequent, produces an initial sequent.

Concerning the remaining rules of $\mathbf{G3Int}$, we need only investigate the permutation of (ref) above the (\supset_l) rule, if we rely on Lem. 5. There are two cases: either the relational atom principal in the (\supset_l) inference is active in the (ref) inference, or it is not. The latter case is easily resolved, so we observe the former:

$$\frac{\mathcal{R}, w \leq w, w : A \supset B, \Gamma \Rightarrow \Delta, w : A \quad \mathcal{R}, w \leq w, w : A \supset B, w : B, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq w, w : A \supset B, \Gamma \Rightarrow \Delta} (\supset_l) \quad \frac{\mathcal{R}, w \leq w, w : A \supset B, \Gamma \Rightarrow \Delta}{\mathcal{R}, w : A \supset B, \Gamma \Rightarrow \Delta} (ref)$$

Applying (ref) to each premise of the (\supset_l) inference yields the following:

$$\frac{\mathcal{R}, w \leq w, w : A \supset B, \Gamma \Rightarrow \Delta, w : A}{\mathcal{R}, w : A \supset B, \Gamma \Rightarrow \Delta, w : A} (ref) \quad \frac{\mathcal{R}, w \leq w, w : A \supset B, w : B, \Gamma \Rightarrow \Delta}{\mathcal{R}, w : A \supset B, w : B, \Gamma \Rightarrow \Delta} (ref)$$

The above observation suggests that we ought to add the (\supset_l^*) rule (see Fig. 7) to our calculus if we wish to permute (ref) above the (\supset_l) rule; a single application of the (\supset_l^*) rule to the end sequents above gives the desired conclusion. With the (\supset_l^*) rule added to our calculus, we may freely permute the (ref) rule above any (\supset_l) inference.

On the basis of our investigation, together with Lem. 5, we may conclude the following lemma:

Lemma 6. *The (ref) rule is eliminable in $\mathbf{G3Int} + \{(id^*), (\supset_l^*)\} - (tra)$.*

Let us turn our attention toward eliminating the (tra) rule from a labelled derivation. Since our aim is to eliminate *both* (ref) and (tra) from any derivation, we assume that the rules $\{(id^*), (\supset_l^*)\}$ have been added to our calculus.

By Lem. 5, (tra) permutes with (\perp_l) and (id^*) , so we only consider the (id) case. As with the (ref) rule, there are two cases to consider when permuting (tra) above an (id) inference: either, the active formula of (tra) is principal in (id) , or it is not. In the latter case, the result of the (tra) rule is an initial sequent, implying that the (tra) rule may be eliminated from the derivation. The former case proves trickier and is explicitly given below:

$$\frac{\mathcal{R}, w \leq u, u \leq v, w \leq v, w : p, \Gamma \Rightarrow \Delta, v : p}{\mathcal{R}, w \leq u, u \leq v, w : p, \Gamma \Rightarrow \Delta, v : p} \begin{array}{l} (id) \\ (tra) \end{array}$$

Observe that the end sequent is not an initial sequent as it is not obtainable from an (id) , (id^*) , or (\perp_l) rule. The issue is solved by considering the $(lift)$ rule (see Fig. 7), which allows us to obtain the desired end sequent without the use of (tra) , as the following derivation demonstrates:

$$\frac{\frac{\mathcal{R}, w \leq u, u \leq v, w : p, u : p, v : p, \Gamma \Rightarrow v : p, \Delta}{\mathcal{R}, w \leq u, u \leq v, w : p, u : p, \Gamma \Rightarrow v : p, \Delta} (lift) \quad \frac{\mathcal{R}, w \leq u, u \leq v, w : p, u : p, v : p, \Gamma \Rightarrow v : p, \Delta}{\mathcal{R}, w \leq u, u \leq v, w : p, \Gamma \Rightarrow v : p, \Delta} (id^*)}{\mathcal{R}, w \leq u, u \leq v, w : p, \Gamma \Rightarrow v : p, \Delta} (lift)$$

Thus, the addition of $(lift)$ to our calculus resolves the issue of permuting (tra) above any initial sequent. Nevertheless, by Lem. 5, we still need to consider the permutation of (tra) above the (\supset_l) and $(lift)$ rules. However, due to the following lemma, we may omit consideration of the (\supset_l) case.

Lemma 7. *The rule (\supset_l) is admissible in $\mathbf{G3Int} + \{(id^*), (\supset_l^*), (lift)\}$.*

Proof. The derivation below proves the admissibility of the rule:

$$\frac{\frac{\frac{\mathcal{R}, x \leq y, x : A \supset B, \Gamma \Rightarrow \Delta, y : A}{\mathcal{R}, x \leq y, x : A \supset B, y : A \supset B, \Gamma \Rightarrow \Delta, y : A} (wk)}{\mathcal{R}, x \leq y, x : A \supset B, y : A \supset B, \Gamma \Rightarrow \Delta} (lift)}{\frac{\frac{\frac{\mathcal{R}, x \leq y, x : A \supset B, y : B, \Gamma \Rightarrow \Delta}{\mathcal{R}, x \leq y, x : A \supset B, y : A \supset B, y : B, \Gamma \Rightarrow \Delta} (wk)}{\mathcal{R}, x \leq y, x : A \supset B, y : A \supset B, \Gamma \Rightarrow \Delta} (\supset_l^*)}{\mathcal{R}, x \leq y, x : A \supset B, \Gamma \Rightarrow \Delta} (lift)} \quad \square$$

Last, the (tra) rule is permutable with the $(lift)$ rule. In the case where the principal relational atom of $(lift)$ is not active in the ensuing (tra) application, the two rules freely permute. The alternative case is resolved as shown below:

$$\frac{\frac{\mathcal{R}, w \leq u, u \leq v, w \leq v, w : A, v : A, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq u, u \leq v, w \leq v, w : A, \Gamma \Rightarrow \Delta} (lift)}{\mathcal{R}, w \leq u, u \leq v, w : A, \Gamma \Rightarrow \Delta} (tra)}{\frac{\frac{\mathcal{R}, w \leq u, u \leq v, w \leq v, w : A, v : A, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq u, u \leq v, w : A, v : A, \Gamma \Rightarrow \Delta} (wk)}{\frac{\frac{\mathcal{R}, w \leq u, u \leq v, w \leq v, w : A, v : A, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq u, u \leq v, w : A, u : A, v : A, \Gamma \Rightarrow \Delta} (lift)}{\mathcal{R}, w \leq u, u \leq v, w : A, u : A, \Gamma \Rightarrow \Delta} (lift)}{\mathcal{R}, w \leq u, u \leq v, w : A, \Gamma \Rightarrow \Delta} (tra)}$$

Hence, we obtain the following:

Lemma 8. *The (tra) rule is eliminable in $\mathbf{G3Int} + \{(id^*), (\supset_l^*), (lift)\} - (ref)$.*

Enough groundwork has been laid to state one of our main lemmata, which is also a consequence of the work in [21].

Lemma 9. *The (ref) and (tra) rules are admissible in the calculus $\mathbf{G3Int} + \{(id^*), (\supset_l^*), (lift)\}$.*

Proof. Suppose we are given a derivation in $\mathbf{G3Int} + \{(id^*), (\supset_l^*), (lift)\}$, and consider the topmost occurrence of either (ref) or (tra) . If we can show that the (ref) rule permutes above the $(lift)$ rule, then we may invoke Lem. 6 and Lem. 8 to conclude that each topmost occurrence of (ref) and (tra) can be eliminated from the given derivation in succession. This yields a (ref) - and (tra) -free proof of the end sequent and establishes the claim. Thus, we prove that the (ref) rule permutes above the $(lift)$ rule.

In the case where the relational atom active in (ref) is not principal in the $(lift)$ inference, the two rules may be permuted; the alternative case is resolved as shown below:

$$\frac{\frac{\mathcal{R}, w \leq w, w : A, w : A, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq w, w : A, \Gamma \Rightarrow \Delta} (lift) \quad \frac{\mathcal{R}, w : A, w : A, \Gamma \Rightarrow \Delta}{\mathcal{R}, w : A, \Gamma \Rightarrow \Delta} (ref)}{\mathcal{R}, w : A, \Gamma \Rightarrow \Delta} \quad \frac{\text{IH}}{\mathcal{R}, w : A, \Gamma \Rightarrow \Delta} (ctrl)$$

□

The addition of the rules $\{(id^*), (\supset_l^*), (lift)\}$ to our calculus and the above admissibility results demonstrate that we are readily advancing toward our goal of deriving **NInt**. However, our labelled calculus is still distinct since it makes use of the logical signature $\{\perp, \wedge, \vee, \supset\}$, whereas **NInt** uses the signature $\{\neg, \wedge, \vee, \supset\}$. Therefore, we need to show that (\perp_l) (we define $\perp := p \wedge \neg p$) is admissible in the presence of (labelled versions of) the (\neg_r) and (\neg_l) rules (see Fig. 7). This admissibility result is explained in the theorem below.

Theorem 5. *The rules $\{(id), (\perp_l), (\supset_l), (ref), (tra)\}$ are admissible in **IntL**.*

Proof. Follows from Lem. 5, Lem. 9, the fact that (id) is derivable using (id^*) and $(lift)$, and the fact that (\perp_l) is derivable from (\neg_l) and (\wedge_l) . Also, the admissibility of (\supset_l) is shown as in Lem. 7. □

Although the above theorem is sufficient to conclude the completeness of **IntL**, we obtain a stronger result. As shown in the theorem below, we may conclude that **IntL** is *complete relative to treelike derivations* possessing the *fixed root property*; i.e. for every **Int**-valid formula $A \in \mathcal{L}$, (i) $\Rightarrow w : A$ is derivable in **IntL** with a treelike derivation, and (ii) the label w is the root of each treelike labelled sequent in the derivation.

Theorem 6. *(i) The calculus **IntL** is complete relative to treelike derivations with the fixed root property. (ii) Every derivation in **IntL** of a treelike labelled sequent is a treelike derivation with the fixed root property.*

Proof. We prove (i), as (ii) is proven similarly. By Thm. 5 above, we know that **IntL** is complete. To prove claim (i) then, we assume that we are given a derivation in **IntL** of a labelled theorem, i.e. the end sequent is of the form $\Rightarrow w : A$. It is not difficult to see that the derivation of $\Rightarrow w : A$ must be treelike with the fixed root property since applying inference rules from the calculus bottom-up to $\Rightarrow w : A$ either preserves relational structure or adds forward relational structure (e.g. (\supset_r) and (\neg_r)), thus constructing a tree emanating from the root w . □

We may leverage the above theorem and the translation \mathfrak{N} (from Def. 11) to complete the extraction of the nested calculus **NInt** from the labelled calculus **G3Int** (via the calculus **IntL**). The theorem below shows that (the treelike fragment of) **IntL** and **NInt** are notational variants of one another:

Theorem 7. *(i) Every derivation in **IntL** of a nestedlike labelled sequent A is effectively translatable to a derivation of the nested sequent $\mathfrak{N}(A)$ in **NInt**. (ii) Every derivation in **NInt** of a nested sequent Σ is effectively translatable to a derivation of the labelled sequent $\mathfrak{L}(\Sigma)$ in **IntL**.*

Proof. By Lem. 3, we know that Λ is treelike, implying that our derivation of Λ is treelike by Thm. 6, which further implies that the translation function \mathfrak{N} is defined for each labelled sequent in the derivation. It is straightforward to verify that each rule in IntL translates to an instance of the corresponding rule in NInt under the \mathfrak{N} translation, and every rule of NInt translates to an instance of the corresponding rule in IntL under the \mathfrak{L} translation. \square

6 Deriving the Calculus NIntQ from G3IntQ

In the preceding section, we observed that the elimination of the (ref) and (tra) rules yielded derivations where every labelled sequent was of a treelike shape. In this section (and the succeeding section) the same effect obtains via the elimination of these two structural rules. Therefore, the majority of this section is dedicated to investigating the effects of eliminating the (nd) and (ihd) structural rules instead.

Similar to our analysis of eliminating the (ref) and (tra) rules, we motivate and explain the *adequate* rules we ought to add to G3IntQ in order to obtain the complete elimination of (nd) and (ihd) . Although the reader may refer to Fig. 7 to witness what rules are needed for the admissibility of (nd) and (ihd) , a central aim of this section is to demonstrate *how* such rules emerge from our analysis when attempting to eliminate (nd) and (ihd) from a derivation—this begets a deeper understanding of how the semantics of IntQ affects the shape of rules in NIntQ (via the labelled calculus G3IntQ). Ultimately, our admissibility results will permit the extraction of NIntQ from G3IntQ .

Lemma 10. *The (ref) and (tra) rules are admissible in the calculus G3IntQ^* .*

Proof. We prove the result by induction on the height of the given derivation and assume w.l.o.g. that the derivation ends with (ref) or (tra) , and that no other instances of (ref) or (tra) occur in the derivation. The general result follows by considering a topmost occurrence of (ref) or (tra) and eliminating each topmost occurrence in succession until a proof free of (ref) and (tra) inferences is obtained.

By Lem. 5, we need only show that (ref) and (tra) permute above the rules (id_q) , (id_q^n) , (\supset_i) , $(lift)$, (\exists_r^n) , (\exists_r^{in}) , (\forall_l) , (\forall_l^n) , (\forall_l^{in}) , and (nd) . The cases of permuting (ref) and (tra) above $(lift)$ are similar as for Thm. 5 and Lem. 8, respectively. Also, (\supset_i) is admissible in the presence of (\supset_i^*) (similar to Lem. 7), (id_q) is an instance of (id_q^n) (by Lem. 11 below), and (\forall_l) is an instance of (\forall_l^n) (also by Lem. 11 below), so these cases may be omitted. Hence, we focus only on the nontrivial cases involving the (id_q^n) , (\exists_r^n) , (\exists_r^{in}) , (\forall_l^n) , (\forall_l^{in}) , and (nd) rules, which are shown below to the left and resolved as shown below on the right. We prove the elimination of (ref) and refer the reader to Appendix A for the similar proof of (tra) elimination.

Base case. We let $\mathcal{R}' := \mathcal{R}, \underline{a}_1 \in D_{v_1}, \dots, \underline{a}_n \in D_{v_n}$ in the (id_q^n) case below. Observe that the end sequent of the derivation below left is an instance of (id_q^n) since the side condition that $v_i \rightsquigarrow^{\mathcal{R}'} w$ for $i \in \{1, \dots, n\}$ holds: if none of the

directed paths from v_i to w contain $u \leq u$, then the paths are present in \mathcal{R}' , and if a directed path from v_i to w does contain $u \leq u$, then deleting each occurrence of $u \leq u$ from the directed path gives a new path from v_i to w in \mathcal{R}' .

$$\frac{\frac{\mathcal{R}', u \leq u, \Gamma, w : p(\vec{a}) \Rightarrow w : p(\vec{a}), \Delta}{\mathcal{R}', \Gamma, w : p(\vec{a}) \Rightarrow w : p(\vec{a}), \Delta} (id_q^n)}{\mathcal{R}', \Gamma, w : p(\vec{a}) \Rightarrow w : p(\vec{a}), \Delta} (ref)} \quad \frac{}{\mathcal{R}', \Gamma, w : p(\vec{a}) \Rightarrow w : p(\vec{a}), \Delta} (id_q^n)$$

Inductive step. For the inductive step, we consider the non-trivial (\exists_r^n) , (\exists_r^{in}) , (\forall_l^n) , (\forall_l^{in}) , and (nd) cases.

$$\frac{\frac{\mathcal{R}, u \leq u, \underline{a} \in D_v, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta}{\mathcal{R}, u \leq u, \underline{a} \in D_v, \Gamma \Rightarrow w : \exists xA, \Delta} (\exists_r^n)}{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow w : \exists xA, \Delta} (ref)} \quad \frac{\frac{}{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta} \text{IH}}{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow w : \exists xA, \Delta} (\exists_r^n)}$$

$$\frac{\frac{\mathcal{R}, u \leq u, \underline{a} \in D_v, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta}{\mathcal{R}, u \leq u, \Gamma \Rightarrow w : \exists xA, \Delta} (\exists_r^{in})}{\mathcal{R}, \Gamma \Rightarrow w : \exists xA, \Delta} (ref)} \quad \frac{\frac{}{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta} \text{IH}}{\mathcal{R}, \Gamma \Rightarrow w : \exists xA, \Delta} (\exists_r^{in})}$$

$$\frac{\frac{\mathcal{R}, u \leq u, \underline{a} \in D_v, w : A(\underline{a}/x), w : \forall xA, \Gamma \Rightarrow \Delta}{\mathcal{R}, u \leq u, \underline{a} \in D_v, w : \forall xA, \Gamma \Rightarrow \Delta} (\forall_l^n)}{\mathcal{R}, \underline{a} \in D_v, w : \forall xA, \Gamma \Rightarrow \Delta} (ref)} \quad \frac{\frac{}{\mathcal{R}, \underline{a} \in D_v, w : A(\underline{a}/x), w : \forall xA, \Gamma \Rightarrow \Delta} \text{IH}}{\mathcal{R}, \underline{a} \in D_v, w : \forall xA, \Gamma \Rightarrow \Delta} (\forall_l^n)}$$

$$\frac{\frac{\mathcal{R}, u \leq u, \underline{a} \in D_v, \Gamma, w : A(\underline{a}/x), w : \forall xA \Rightarrow \Delta}{\mathcal{R}, u \leq u, \Gamma, w : \forall xA \Rightarrow \Delta} (\forall_l^{in})}{\mathcal{R}, w : \forall xA, \Gamma \Rightarrow \Delta} (ref)} \quad \frac{\frac{}{\mathcal{R}, \underline{a} \in D_v, \Gamma, w : A(\underline{a}/x), w : \forall xA, \Gamma \Rightarrow \Delta} \text{IH}}{\mathcal{R}, w : \forall xA, \Gamma \Rightarrow \Delta} (\forall_l^{in})}$$

$$\frac{\frac{\mathcal{R}, w \leq w, \underline{a} \in D_w, \underline{a} \in D_w, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq w, \underline{a} \in D_w, \Gamma \Rightarrow \Delta} (nd)}{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta} (ref)} \quad \frac{\frac{}{\mathcal{R}, w \leq w, \underline{a} \in D_w, \underline{a} \in D_w, \Gamma \Rightarrow \Delta} \text{IH}}{\mathcal{R}, w \leq w, \underline{a} \in D_w, \Gamma \Rightarrow \Delta} (ctr_{\mathcal{R}})}$$

We now argue that the side condition $(v \rightsquigarrow^{\mathcal{R}} w)$ of (\exists_r^n) , (\exists_r^{in}) , (\forall_l^n) , and (\forall_l^{in}) continues to hold after applying IH above. If the directed path from v to w does not go through u , then the side condition trivially holds. Alternatively, if the directed path from v to w does go through u , then by deleting each occurrence of $u \leq u$ from the path, we obtain a new directed path from v to w , which continues to be present after the invocation of IH. \square

We now analyze the admissibility of (nd) in $\text{G3IntQ}^* - \{\text{ref}, \text{tra}\}$, and afterward, analyze the admissibility of the (ihd) rule. It will be instructive to first consider the case of permuting (nd) above the (\exists_r) rule solely. It will be seen that in order to permute (nd) above (\exists_r) , it is sufficient to strengthen the (\exists_r) rule through the addition of a side condition; this motivates a similar side condition that we will impose on (id_q) (yielding the rule (id_q^n)) and on (\forall_l) (yielding the rule (\forall_l^n)).

To begin our analysis, observe (1) a non-trivial application of (nd) after an (\exists_r) inference, and (2) the result of applying (nd) to the top sequent in (1).

$$\frac{\frac{\mathcal{R}, v \leq w, \underline{a} \in D_v, \underline{a} \in D_w, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists x A}{\mathcal{R}, v \leq w, \underline{a} \in D_v, \underline{a} \in D_w, \Gamma \Rightarrow \Delta, w : \exists x A} (\exists_r)}{\mathcal{R}, v \leq w, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \exists x A} (nd) \quad (1)$$

$$\frac{\mathcal{R}, v \leq w, \underline{a} \in D_v, \underline{a} \in D_w, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists x A}{\mathcal{R}, v \leq w, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists x A} (nd) \quad (2)$$

In (2), the application of (nd) has generated a labelled sequent where the domain atom (i.e. $\underline{a} \in D_v$) is associated with the label v *one step in the past* of w (due to the presence of the $v \leq w$ relational atom). One possible solution to overcome this obstacle, and to allow for the permutation of (nd) above (\exists_r) , would be to add the following rule to our calculus:

$$\frac{\mathcal{R}, v \leq w, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists x A}{\mathcal{R}, v \leq w, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \exists x A} (\exists_r)'$$

In the rule above, the domain atom $\underline{a} \in D_v$ is associated with the label v , which is *one step in the past* of w . Although the addition of the above rule solves the problem of permuting (nd) above the (\exists_r) rule, its inclusion into our calculus would inevitably place us in a similar, undesirable circumstance:

$$\frac{\frac{\mathcal{R}, u \leq v, v \leq w, \underline{a} \in D_u, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists x A}{\mathcal{R}, u \leq v, v \leq w, \underline{a} \in D_u, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \exists x A} (\exists_r)'}{\mathcal{R}, u \leq v, v \leq w, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \exists x A} (nd)$$

$$\frac{\mathcal{R}, u \leq v, v \leq w, \underline{a} \in D_u, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists x A}{\mathcal{R}, u \leq v, v \leq w, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists x A} (nd)$$

It appears that now we would need to add a new version of the (\exists_r) rule allowing us to delete $w : A(\underline{a})$ when the required relational atom $\underline{a} \in D_u$ is associated with a label (viz. u) that is *two steps in the past* of w (due to the occurrence

of $u \leq v, v \leq w$). Including such a rule however, would inevitably force us to add another existential rule where the required relational atom is associated with a label *three steps in the past*—such rule inclusions would then continue ad infinitum. To resolve the problem, and capture all such possibilities, we add the following rule to our calculus, which has the side condition $v \rightsquigarrow^{\mathcal{R}} w$ (see Def. 15 and Fig. 7):

$$\frac{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists x A}{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \exists x A} (\exists_r^n)$$

This rule generalizes the shifting behaviour of the domain atom witnessed above to include all possible *shifts*. Furthermore, due to the fact that the labelled sequents we are considering encode an **IntQ**-model—which satisfies the *nested domain* condition—it is appropriate to impose the above side condition, which essentially states that *if an object is an element of some domain D_v in the past of w , then it is an element of the domain D_w* . It is straightforward to verify that the (\exists_r^n) rule is sound relative to **IntQ**-models.

If one analyzes the cases of permuting (nd) above the (id_q) and (\forall_l) rules, then they will observe the same shifting behavior of domain atoms as in the (\exists_r) case. We therefore impose a similar side condition on (id_q) and (\forall_l) to obtain the (id_q^n) and (\forall_l^n) rules (see Fig. 7). It is not difficult to see that the following holds:

Lemma 11. *The (id_q) , (\exists_r) , and (\forall_l) rules are instances of the (id_q^n) , (\exists_r^n) , and (\forall_l^n) rules, respectively.*

The lemma below shows that the (id_q^n) , (\forall_l^n) , and (\exists_r^n) rules are sufficient to prove the admissibility of (nd) :

Lemma 12. *The rule (nd) is admissible in the calculus $\mathbf{G3IntQ}^* - \{(ref), (tra)\}$.*

Proof. We prove the result by induction on the height of the given derivation. By Lem. 5 and Lem. 10, we need only consider the non-trivial (id_q^n) , (\exists_r^n) , and (\forall_l^n) cases.

Base case. We let $\mathcal{R}' := \mathcal{R}, u \leq v, \underline{a}_1 \in D_{v_1}, \dots, \underline{a}_n \in D_{v_n}$ and assume that both $v \rightsquigarrow^{\mathcal{R}'} w$ and $v_i \rightsquigarrow^{\mathcal{R}'} w$ (for $i \in \{1, \dots, n\}$) hold. Observe that the end sequent is an instance of (id_q^n) since (i) the paths from v_i to w are still present and (ii) there is a directed path from u to w composed of the relational atom $u \leq v$ and the direct path from v to w (i.e. $u \rightsquigarrow^{\mathcal{R}'} w$ holds).

$$\frac{\mathcal{R}', \underline{b} \in D_u, \underline{b} \in D_v, \Gamma, w : p(\vec{\underline{a}}, \underline{b}) \Rightarrow w : p(\vec{\underline{a}}, \underline{b}), \Delta}{\mathcal{R}', \underline{b} \in D_u, \Gamma, w : p(\vec{\underline{a}}, \underline{b}) \Rightarrow w : p(\vec{\underline{a}}, \underline{b}), \Delta} \begin{matrix} (id_q^n) \\ (nd) \end{matrix}$$

Inductive step. We provide each proof below showing that (nd) can be permuted above (\exists_r^n) and (\forall_l^n) (we consider only the non-trivial cases). Also, we assume that there is a directed path from v to w , i.e. the side condition $v \rightsquigarrow^{\mathcal{R}} w$ holds.

$$\begin{array}{c}
\frac{\mathcal{R}, u \leq v, \underline{a} \in D_u, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists xA}{\frac{\mathcal{R}, u \leq v, \underline{a} \in D_u, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \exists xA}{\mathcal{R}, u \leq v, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \exists xA} \text{ (nd)}} (\exists_r^n) \\
\\
\frac{\frac{\mathcal{R}, u \leq v, \underline{a} \in D_u, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists xA}{\mathcal{R}, u \leq v, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \exists xA} \text{ (nd)}}{\mathcal{R}, u \leq v, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \exists xA} \text{ IH}} (\exists_r^n) \\
\\
\frac{\mathcal{R}, u \leq v, \underline{a} \in D_u, \underline{a} \in D_v, \Gamma, w : \forall xA, w : A(\underline{a}/x) \Rightarrow \Delta}{\frac{\mathcal{R}, u \leq v, \underline{a} \in D_u, \underline{a} \in D_v, \Gamma, w : \forall xA \Rightarrow \Delta}{\mathcal{R}, u \leq v, \underline{a} \in D_u, \Gamma, w : \forall xA \Rightarrow \Delta} \text{ (nd)}} (\forall_l^n) \\
\\
\frac{\frac{\mathcal{R}, u \leq v, \underline{a} \in D_u, \underline{a} \in D_v, \Gamma, w : \forall xA, w : A(\underline{a}/x) \Rightarrow \Delta}{\mathcal{R}, u \leq v, \underline{a} \in D_u, \Gamma, w : \forall xA \Rightarrow \Delta} \text{ (nd)}}{\mathcal{R}, u \leq v, \underline{a} \in D_u, \Gamma, w : \forall xA \Rightarrow \Delta} \text{ IH}} (\forall_l^n)
\end{array}$$

By assumption, there is a directed path from v to w in \mathcal{R} . Observe that the side condition is still satisfied after the invocation of IH since there is a directed path from u to w composed of the relational atom $u \leq v$ and the directed path from v to w in \mathcal{R} . \square

We now turn our attention toward analyzing the admissibility of (ihd) in the calculus $\text{G3IntQ}^* - \{(ref), (tra), (nd)\}$. We will see that sufficient modifications must be made to the (\exists_r^n) and (\forall_l^n) rules to necessitate the elimination of (ihd) from a given derivation. (NB. Due to Lem. 11, we may omit consideration of permuting (ihd) above (\exists_r) and (\forall_l) .) Let us consider non-trivial applications of (ihd) after an (\exists_r^n) and (\forall_l^n) inference:

$$\frac{\frac{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta}{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow w : \exists xA, \Delta} (\exists_r^n) \frac{\mathcal{R}, \underline{a} \in D_v, \Gamma, w : \forall xA, w : A(\underline{a}/x) \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \forall xA, \Rightarrow \Delta} (\forall_l^n)}{\mathcal{R}, \Gamma \Rightarrow w : \exists xA, \Delta} (ihd)$$

Observe that the (ihd) rule cannot be permuted above the (\exists_r^n) or (\forall_l^n) rules due to the existence of the labelled formula $w : A(\underline{a}/x)$ in the top sequent which violates the side condition on (ihd) stating that \underline{a} must be an eigenvariable. Our solution to this issue is to *absorb* the (ihd) rule into the (\exists_r^n) and (\forall_l^n) rules yielding the new versions (\exists_r^{in}) and (\forall_l^{in}) :

$$\frac{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \exists xA, \Delta} (\exists_r^{in}) \frac{\mathcal{R}, \underline{a} \in D_v, \Gamma, w : \forall xA, w : A(\underline{a}/x) \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \forall xA, \Rightarrow \Delta} (\forall_l^{in})$$

where we impose the side condition that $v \rightsquigarrow_{\mathcal{R}} w$ and \underline{a} is an eigenvariable. This side condition essentially states that *there exists an element of D_w , which is also in a domain D_v in the past of w* . It can be shown that each rule is sound relative to IntQ-models, and relies both on the nested domain condition and the fact that such domains are *non-empty*, i.e. *inhabited*.

Lemma 13. *The rule (ihd) is admissible in $\text{G3IntQ}^* - \{(ref), (tra), (nd)\}$.*

Proof. We prove the result by induction on the height of the given derivation. The base case is simple as any application of (ihd) to (id_q^n) or (\perp_l) yields another instance of the rule. By Lem. 5 and Lem. 11, the only cases we need to consider in the inductive step are the non-trivial (\exists_r^n) and (\forall_l^n) cases; however, as we saw above, the non-trivial cases can be handled by applying (\exists_r^{in}) or (\forall_l^{in}) , respectively. \square

Theorem 8. *The rules $\{(id_q), (\perp_l), (\supset_l), (ref), (tra), (\exists_r), (\forall_l), (nd), (ihd)\}$ are admissible in the calculus $\mathbf{G3IntQ}^*$.*

Proof. The admissibility of $\{(ref), (tra)\}$, $\{(id_q), (\exists_r), (\forall_l)\}$, (nd) , and (ihd) follow from Lem. 10, Lem. 11, Lem. 12, and Lem. 13, respectively. Admissibility of (\supset_l) is similar to Lem. 7 and admissibility of (\perp_l) follows from the fact that the rule can be derived using (\neg_l) and (\wedge_l) . \square

Theorem 9. *(i) The calculus \mathbf{IntQL} is complete relative to treelike derivations with the fixed root property. (ii) Every derivation in \mathbf{IntQL} of a treelike labelled sequent is a treelike derivation with the fixed root property.*

Proof. Similar to Thm. 6. \square

Although the above theorem is fundamental for translating derivations in \mathbf{IntQL} into derivations in \mathbf{NIntQ} , additional work needs to be done to ensure that the (\exists_r^n) and (\forall_l^n) rules properly translate to instances of the (\exists_r) and (\forall_l) rules. The side condition imposed on (\exists_r^n) and (\forall_l^n) relies on the existence of a domain atom containing a parameter \underline{a} , whereas the side condition imposed on the nested (\exists_r) and (\forall_l) rules stipulates that the parameter \underline{a} must either be an eigenvariable or available. The question arises: how do we reconcile these seemingly distinct side conditions?

Interestingly, our process of refinement has offered us an unadulterated perspective of the nested (\exists_r) and (\forall_l) rules. As will be argued below (Thm. 10), the rules (\exists_r^{in}) and (\forall_l^{in}) in \mathbf{IntQL} translate to instances of the nested (\exists_r) and (\forall_l) rules where the relevant parameter of the inference is an *eigenvariable*, and the rules (\exists_r^n) and (\forall_l^n) translate to instances of the nested (\exists_r) and (\forall_l) rules where the relevant parameter of the inference is *available*. In other words, what are single rules with a disjunctive side condition in the nested setting have bifurcated into distinct sets of rules with atomic side conditions in the labelled setting.

In order to ensure that instances of (\exists_r^n) and (\forall_l^n) properly translate to instances of the nested (\exists_r) and (\forall_l) rules then, we must show that if the side condition of the former rules holds, then the side condition of the latter rules holds. This motivates the definition of a *nested form* derivation:

Definition 17 (Nested Form). *Let Π be a derivation of a labelled sequent Λ in \mathbf{IntQL} . We define Π to be in nested form iff (i) Λ is nestedlike, and (ii) for each instance of (\exists_r^n) and (\forall_l^n) in Π :*

$$\frac{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists xA}{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \exists xA} (\exists_r^n) \frac{\mathcal{R}, \underline{a} \in D_v, w : A(\underline{a}/x), w : \forall xA, \Gamma \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_v, w : \forall xA, \Gamma \Rightarrow \Delta} (\forall_l^n)$$

there exists a formula $u : B(\underline{a})$ such that $u \rightsquigarrow^{\mathcal{R}} w$ in the conclusion of the inference.

Assuming we are given a nested form derivation in **IntQL**, condition (ii) of the above definition ensures that instances of (\exists_r^n) and (\forall_l^n) will properly translate into instances of (\exists_r) and (\forall_l) in **NIntQ**. Additionally, nested form proofs are significant because one can view the space of nested form proofs in **IntQL** as being equivalent to the space of proofs generated by **NIntQ**, as confirmed by Thm. 10 and 11 below. Before we establish this correspondence however, the following lemma is needed for the proof of Thm. 10, that is, for showing that if a nestedlike labelled sequent A is provable in **IntQL**, then $\mathfrak{N}(A)$ is provable in **NIntQ**.

Lemma 14. *If a nestedlike labelled sequent A is derivable in **IntQL**, then A has a nested form derivation in **IntQL**.*

The truth of the above lemma can be seen to follow from Thm. 11 below (which states that each derivation in **NIntQ** of a nested sequent Σ can be translated into a nested form derivation of $\mathfrak{L}(\Sigma)$ in **IntQL**) in conjunction with a few facts: (i) **IntQL** is sound, (ii) one can show by semantic arguments that if a nestedlike labelled sequent A is valid, then so is $\mathfrak{N}(A)$, (iii) by Fitting’s work in [9], we know that **NIntQ** is strongly complete (Thm. 4), and (iv) for any nestedlike labelled sequent A , $\mathfrak{L}(\mathfrak{N}(A)) = A$ (Lem. 3).⁴ Hence, given a derivable nestedlike labelled sequent A , which is therefore valid since **IntQL** is sound by (i), it follows that $\mathfrak{N}(A)$ is valid by (ii), meaning that $\mathfrak{N}(A)$ is derivable by (iii), and so by Thm. 11 below, $\mathfrak{L}(\mathfrak{N}(A))$ has a nested form derivation, i.e. A has a nested form derivation by (iv). Although this argument serves as a proof of Lem. 14, it is not entirely satisfactory.

The above argument relies on the completeness of **NIntQ**, and so, if our aim were to extract **NIntQ** from **G3IntQ** without any prior knowledge of **NIntQ** or its properties, or to show **NIntQ** complete via this extraction, then since the above argument relies on the completeness of **NIntQ** (i.e. Thm. 4), we would fail to reach our aim. The author sees at least two possible strategies for confirming the above lemma without relying on the completeness of **NIntQ**. In the first method, one could write a (potentially non-terminating) proof-search procedure for **IntQL** that takes a nestedlike labelled sequent A as input and attempts to construct a nested form proof of A , showing that a counter-model for A can be constructed given that the procedure does not halt with a nested form proof of the input. Such a method of proof would effectively prove the completeness of **IntQL** relative to valid nestedlike labelled sequents while showing that every such sequent possesses a nested form proof. The second method consists of showing how to algorithmically transform any derivation of a nestedlike labelled sequent

⁴The soundness of **IntQL** is established by Thm. 16 in Sect. 8, where it is shown that every derivation in **IntQL** can be effectively transformed into a derivation of the same labelled sequent in **G3IntQ**. Since **G3IntQ** is sound by Thm. 3, this establishes the soundness of **IntQL**.

Λ in IntQL into a nested form proof of Λ . Such methods of proof are desirable as they would effectively demonstrate that NIntQ inherits completeness from its associated labelled calculus G3IntQ , however, we leave such results to future work, and continue our investigation of the correspondence between semantic and nested systems.

Theorem 10. *If a nestedlike labelled sequent Λ is derivable in IntQL , then $\mathfrak{N}(\Lambda)$ is derivable in NIntQ .*

Proof. Since Λ is nestedlike and derivable by assumption, we know that Λ is treelike (by Lem. 3) and has a nested form derivation Π (by Lem. 14), implying that Π is treelike by Thm. 9. Hence, the translation function \mathfrak{N} is defined for each labelled sequent in Π . It is straightforward to verify that each rule in IntQL —with the exception of (\exists_r^n) , (\exists_r^{in}) , (\forall_l^n) , and (\forall_l^{in}) —translates to an instance of the corresponding rule in NIntQ under the \mathfrak{N} translation. We show the translation of the (\exists_r^n) and (\exists_r^{in}) inferences below; the (\forall_l^n) and (\forall_l^{in}) inferences are similar.

In the (\exists_r^n) inference top-left below, we know that there exists a labelled formula $u : B(\underline{a}) \in \Gamma, \Delta$ where $u \rightsquigarrow^{\mathcal{R}} w$ since our given derivation is in nested form (Def. 17). Hence, after applying \mathfrak{N} to the premise of the (\exists_r^n) inference, the parameter \underline{a} will occur in the formula $B(\underline{a})$ that is either in the component $\Gamma \upharpoonright w \rightarrow A(\underline{a}/x), \exists xA, \Delta \upharpoonright w$ (and is different than the occurrence of $A(\underline{a}/x)$) or is in the past of the component in the nested sequent $\Sigma\{\Gamma \upharpoonright w \rightarrow A(\underline{a}/x), \exists xA, \Delta \upharpoonright w\}$. Hence, the parameter \underline{a} is available, and so, the necessary side condition of the (\exists_r) inference is met. In the (\exists_r^{in}) inference below, the parameter \underline{a} is an eigenvariable and continues to be so after applying the translation \mathfrak{N} , implying that the nested (\exists_r) inference is valid.

$$\frac{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists xA}{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \exists xA} (\exists_r^n)$$

$$\begin{aligned} & \frac{\mathfrak{N}(\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists xA)}{\Sigma\{\Gamma \upharpoonright w \rightarrow A(\underline{a}/x), \exists xA, \Delta \upharpoonright w\}} (\exists_r) = \\ & \frac{\Sigma\{\Gamma \upharpoonright w \rightarrow \exists xA, \Delta \upharpoonright w\}}{\mathfrak{N}(\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \exists xA)} = \end{aligned}$$

$$\frac{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists xA}{\mathcal{R}, \Gamma \Rightarrow \Delta, w : \exists xA} (\exists_r^{in})$$

$$\begin{aligned} & \frac{\mathfrak{N}(\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists xA)}{\Sigma\{\Gamma \upharpoonright w \rightarrow A(\underline{a}/x), \exists xA, \Delta \upharpoonright w\}} (\exists_r) = \\ & \frac{\Sigma\{\Gamma \upharpoonright w \rightarrow \exists xA, \Delta \upharpoonright w\}}{\mathfrak{N}(\mathcal{R}, \Gamma \Rightarrow \Delta, w : \exists xA)} = \end{aligned}$$

□

Theorem 11. *Every derivation in NIntQ of a nested sequent Σ is effectively translatable to a nested form derivation of the labelled sequent $\mathfrak{L}(\Sigma)$ in IntQL .*

Proof. We show how to translate the (\forall_l) , $(lift)$, (\supset_r) and (\exists_r) rules as all other cases are simple or similar. Last, we argue that the derivation obtained in IntQL is in nested form.

(\forall_l) . Our given (\forall_l) inference is shown below top. Let $\vec{a} := \underline{a}_1, \dots, \underline{a}_n$ be all free parameters in A, X, Y that do not occur in B, X, Y , and let $\vec{b} := \underline{b}_1, \dots, \underline{b}_k$ be all free parameters in B, X, Y that do not occur in A, X, Y . To obtain the desired conclusion, we first apply the hp-admissibility of (wk) (Lem. 4) in order to ensure that the contexts of the premises match, followed by an application of the (\forall_l) rule. We assume that w is the label assigned to the formulae of the components $X, A \rightarrow Y$ and $X, B \rightarrow Y$ in $\Sigma\{X, A \rightarrow Y\}$ and $\Sigma\{X, B \rightarrow Y\}$, respectively.

$$\frac{\Sigma\{X, A \rightarrow Y\} \quad \Sigma\{X, B \rightarrow Y\}}{\Sigma\{X, A \vee B \rightarrow Y\}} (\forall_l)$$

$$\frac{\frac{\frac{\mathfrak{L}(\Sigma\{X, A \rightarrow Y\})}{\mathcal{R}, \vec{a} \in D_w, w : A, \Gamma \Rightarrow \Delta} = \quad \frac{\mathfrak{L}(\Sigma\{X, B \rightarrow Y\})}{\mathcal{R}, \vec{b} \in D_w, w : B, \Gamma \Rightarrow \Delta} =}{\mathcal{R}, \vec{a} \in D_w, \vec{b} \in D_w, w : A, \Gamma \Rightarrow \Delta} (wk) \quad \frac{\mathcal{R}, \vec{a} \in D_w, \vec{b} \in D_w, w : B, \Gamma \Rightarrow \Delta}{\mathcal{R}, \vec{a} \in D_w, \vec{b} \in D_w, w : B, \Gamma \Rightarrow \Delta} (wk)}{\mathcal{R}, \vec{a} \in D_w, \vec{b} \in D_w, w : A \vee B, \Gamma \Rightarrow \Delta} (\forall_l)}{\mathfrak{L}(\Sigma\{X, A \vee B \rightarrow Y\})} =$$

$(lift)$. Our given $(lift)$ inference is shown below top-left. Let $\vec{a} := \underline{a}_1, \dots, \underline{a}_n$ be all free parameters occurring in A that do not occur in X', Y' , w be the label assigned to the formulae of the component $X, A \rightarrow Y$, and u be the label assigned to the formulae of the component $X', A \rightarrow Y'$. If we invoke the admissibility of (nd) (Thm. 8) n times to delete all domain atoms $\vec{a} \in D_u$, then after an application of $(lift)$, we obtain the desired conclusion.

$$\frac{\Sigma\{X, A \rightarrow Y, [X', A \rightarrow Y']\}}{\Sigma\{X, A \rightarrow Y, [X' \rightarrow Y']\}} (lift)$$

$$\frac{\frac{\frac{\mathfrak{L}(\Sigma\{X, A \rightarrow Y, [X', A \rightarrow Y']\})}{\mathcal{R}, \vec{a} \in D_w, \vec{a} \in D_u, w \leq u, w : A, u : A, \Gamma \Rightarrow \Delta} =}{\mathcal{R}, \vec{a} \in D_w, w \leq u, w : A, u : A, \Gamma \Rightarrow \Delta} (nd) \times n}{\mathcal{R}, \vec{a} \in D_w, w \leq u, w : A, \Gamma \Rightarrow \Delta} (lift)}{\mathfrak{L}(\Sigma\{X, A \rightarrow Y, [X' \rightarrow Y']\})} =$$

(\supset_r) . Let $\vec{a} := \underline{a}_1, \dots, \underline{a}_n$ be the free parameters occurring in the active formulae A and B (shown in the (\supset_r) inference below). Let w be the label assigned to the formulae of the component $X \rightarrow Y$ and v be the label assigned to the formulae of the component $A \rightarrow B$ after applying the translation function \mathfrak{L} . In order to obtain the desired conclusion, we first apply the hp-admissibility of (wk) (Lem. 4), which introduces the domain atoms $\vec{a}' \in D_w := \underline{a}_{i_1} \in D_w, \dots, \underline{a}_{i_k} \in D_w$ such that $\underline{a}_{i_j} \in D_w \notin \mathcal{R}$ for $i_j \in \{1, \dots, n\}$ and $j \in \{1, \dots, k\}$, i.e. for each parameter \underline{a}_{i_j} in \vec{a} we introduce the domain atom $\underline{a}_{i_j} \in D_w$ given that it does not already occur in \mathcal{R} . Next, we apply admissibility of (nd) (Thm. 8) n times

to delete all domain atoms $\underline{a} \in D_v$, making v an eigenvariable, and permitting (\supset_r) to be applied.

$$\frac{\Sigma\{X \rightarrow Y, [A \rightarrow B]\}}{\Sigma\{X \rightarrow A \supset B, Y\}} (\supset_r)$$

$$\begin{array}{c} \mathfrak{L}(\Sigma\{X \rightarrow Y, [A \rightarrow B]\}) \\ \dots\dots\dots = \\ \mathcal{R}, \underline{a} \in D_v, w \leq v, v : A, \Gamma \Rightarrow \Delta, v : B \quad (wk) \\ \hline \mathcal{R}, \underline{a}' \in D_w, \underline{a} \in D_v, w \leq v, v : A, \Gamma \Rightarrow \Delta, v : B \quad (nd) \times n \\ \hline \mathcal{R}, \underline{a}' \in D_w, w \leq v, v : A, \Gamma \Rightarrow \Delta, v : B \quad (\supset_r) \\ \hline \mathcal{R}, \underline{a}' \in D_w, \Gamma \Rightarrow \Delta, w : A \supset B \\ \dots\dots\dots = \\ \mathfrak{L}(\Sigma\{X \rightarrow A \supset B, Y\}) \end{array}$$

(\exists_r) . If the relevant parameter \underline{a} (in the active formula $A(\underline{a}/x)$) is an *eigenvariable* in the (\exists_r) inference below, then the inference translates to an instance of (\exists_r^{in}) . If the relevant parameter is *available* in the inference, then the translation requires more care, so we focus on showing how to translate this case below.

$$\frac{\Sigma\{X \rightarrow A(\underline{a}/x), \exists xA, Y\}}{\Sigma\{X \rightarrow \exists xA, Y\}} (\exists_r)$$

Since the parameter \underline{a} is available in (\exists_r) , this implies that there exists some formula $B(\underline{a})$ (different than the occurrence of $A(\underline{a}/x)$) occurring in, or in the past of, the displayed component $X \rightarrow A(\underline{a}/x), \exists xA, Y$ of the nested sequent $\Sigma\{X \rightarrow A(\underline{a}/x), \exists xA, Y\}$ (i.e. the premise of the (\exists_r) inference above). We assume that the translation \mathfrak{L} assigns the label u to all formulae in the component where $B(\underline{a})$ occurs, and note that $u : B(\underline{a}) \in \Gamma, \Delta$ below. By the definition of \mathfrak{L} (Def. 12), this implies the existence of a domain atom $\underline{a} \in D_u$ in the image of $\Sigma\{X \rightarrow A(\underline{a}/x), \exists xA, Y\}$ under \mathfrak{L} (which we have explicitly included in the derivation below). Because $B(\underline{a})$ is available and is associated with the label u , this implies that there is a directed path $u \leq v_1, \dots, v_n \leq w$ from u to w in \mathcal{R} , i.e. $u \rightsquigarrow^{\mathcal{R}} w$. (NB. If $u = w$, then the translation is straightforward, so we omit consideration of that case.) If we weaken in domain atoms of the form $\underline{a} \in D_{v_i}$ (with $i \in \{1, \dots, n\}$) along this directed path, invoke the admissibility of (nd) (Thm. 8), and then apply (\exists_r^n) , we obtain the image of the conclusion of the (\exists_r) inference above under the \mathfrak{L} translation. This entire process is succinctly demonstrated in the derivation below:

$$\begin{array}{c} \mathfrak{L}(\Sigma\{X \rightarrow A(\underline{a}/x), \exists xA, Y\}) \\ \dots\dots\dots = \\ \mathcal{R}, \underline{a} \in D_u, \underline{a} \in D_w, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta \quad (wk) \\ \hline \mathcal{R}, \underline{a} \in D_u, \underline{a} \in D_{v_1}, \dots, \underline{a} \in D_{v_n}, \underline{a} \in D_w, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta \quad (nd) \times (n+1) \\ \hline \mathcal{R}, \underline{a} \in D_u, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta \quad (\exists_r^n) \\ \hline \mathcal{R}, \underline{a} \in D_u, \Gamma \Rightarrow w : \exists xA, \Delta \\ \dots\dots\dots = \\ \mathfrak{L}(\Sigma\{X \rightarrow \exists xA, Y\}) \end{array}$$

To see why the output derivation is in nested form, first observe that the conclusion of the derivation will be $\mathfrak{L}(\Sigma)$, which, *ipso facto*, makes the conclusion a nestedlike labelled sequent; hence, the derivation satisfies property (i) of Def. 17. Second, observe that the output derivation in **IntQL** satisfies property (ii) of Def. 17 since each (\exists_r^n) and (\forall_r^n) inference is obtained from an (\exists_r) or (\forall_l) inference in the input derivation, ensuring the existence of a labelled formula $u : B(\underline{a})$ in the conclusion of any (\exists_r^n) or (\forall_r^n) inference such that there is a directed path from u to the label w prefixing the principal formula. \square

7 Deriving the Calculus **NIntQC** from **G3IntQC***

In the previous section, we saw that strengthening the (\exists_r) and (\forall_l) rules allowed for the rules (nd) and (ihd) to be eliminated from any given derivation. In this section, we will focus on how to strengthen these rules even further to additionally obtain the elimination of (cd) from any given derivation. We begin our analysis after first stating the following admissibility result:

Lemma 15. *The (ref) and (tra) rules are admissible in the calculus **G3IntQC****.

Proof. The proof is similar to Lem. 10 and is detailed in Appendix A for the interested reader. \square

The exclusion of (ref) and (tra) implies the completeness of **G3IntQC*** – $\{(ref), (tra)\}$ relative to treelike derivations, similar to Lem. 9. We therefore assume for the rest of the section that all labelled sequents we consider are treelike—this assumption will ease our admissibility analyses of (nd) , (cd) , and (ihd) , though note that the assumption can be omitted, if desired.

Previously, we saw that permutations of the (nd) rule above the (id_q) , (\exists_r) , and (\forall_l) rules shifted domain atoms to the past. We now investigate how permutations of the (cd) rule affect domain atoms in (id_q) , (\exists_r) , and (\forall_l) . As before, we first concentrate on permuting (cd) above the (\exists_r) rule solely. Observing the effect of permuting (cd) above (\exists_r) will motivate how we strengthen the rule—and the (id_q) and (\forall_l) rules—to obtain the complete elimination of (cd) from any given derivation.

To begin our analysis, observe (3) a non-trivial application of (cd) after an (\exists_r) inference, and (4) the result of applying (cd) to the top sequent in (3).

$$\frac{\frac{\mathcal{R}, w \leq v, \underline{a} \in D_w, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists x A}{\mathcal{R}, w \leq v, \underline{a} \in D_w, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \exists x A} (\exists_r)}{\mathcal{R}, w \leq v, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \exists x A} (cd) \quad (3)$$

$$\frac{\mathcal{R}, w \leq v, \underline{a} \in D_w, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists x A}{\mathcal{R}, w \leq v, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists x A} (cd) \quad (4)$$

The application of (cd) has generated a labelled sequent where the domain atom (i.e. $\underline{a} \in D_v$) is associated with the label v *one step in the future* of w (due to the presence of the $w \leq v$ relational atom). By analogy to what we saw in

the last section (in the case of permuting (nd) above (\exists_r)), it is not hard to see that successive (non-trivial) permutations of (cd) above an (\exists_r) inference will continually shift the domain atom to the future. If—at the same time—we also consider (non-trivial) permutations of the (nd) rule above the (\exists_r) rule, then the domain atom can shift either to the future or the past.

Recall that we have restricted our analysis to only consider derivations in $\text{G3IntQC}^* - \{(ref), (tra)\}$ that employ treelike labelled sequents. Observe that *any node* in the graph of a treelike labelled sequent is reachable from any other node by a sequence of forward and backwards edges. This property of treelike labelled sequents, accompanied by the witnessed shifting behavior discussed in the previous paragraph, suggests that we ought to add the following (\exists_r^c) rule (see Fig. 7) to our calculus:

$$\frac{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists x A}{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \exists x A} (\exists_r^c)$$

with the side condition that $v \sim^{\mathcal{R}} w$, i.e. there is an undirected path from the label v to the label w . Observe that this side condition generalizes the shifting behaviour of the domain atom to include all possible forward and backward *shifts*. Intuitively, this side condition states that *if an object is an element of some domain D_v , then it is an element of the domain D_w* , which always holds in the *constant domain* setting.

The same shifting behavior can be witnessed by considering the permutation of (nd) and (cd) above the (id_q) and (\forall_i) rules. Thus, we strengthen those rules in a similar way to obtain the (id_q^c) and (\forall_i^c) rules (see Fig. 7).

Lemma 16. *The (id_q) , (\exists_r) , and (\forall_i) rules are instances of the (id_q^c) , (\exists_r^c) , and (\forall_i^c) rules, respectively.*

The lemma below confirms that the strengthened versions of (id_q) , (\exists_r) , and (\forall_i) are sufficient to prove the admissibility of the (nd) and (cd) rules in the calculus $\text{G3IntQC}^* - \{(ref), (tra)\}$:

Lemma 17. *The rules (nd) and (cd) are admissible in the calculus $\text{G3IntQC}^* - \{(ref), (tra)\}$.*

Proof. The proof is similar to the proof of Lem. 12, but where undirected paths (Def. 16) are used instead of directed paths (Def. 15). \square

In the previous section, we saw that permutations of (ihd) above the (\exists_r^n) and (\forall_i^n) rules could not be performed in certain cases. To resolve this issue, we introduced versions of the rules that absorbed the (ihd) rule. The same phenomenon occurs in the case of the (\exists_r^c) and (\forall_i^c) rules, implying that we need to add (\exists_r^{ic}) and (\forall_i^{ic}) to our calculus (see Fig. 7).

Lemma 18. *The rule (ihd) is admissible in $\text{G3IntQC}^* - \{(ref), (tra), (nd), (cd)\}$.*

Proof. Similar to the proof of Lem. 13. \square

Theorem 12. *The rules (id_q) , (\perp_l) , (\supset_l) , (ref) , (tra) , (\exists_r) , (\forall_l) , (\forall_r) , (nd) , (cd) , and (ihd) are admissible in the calculus IntQCL .*

Proof. The admissibility of $\{(ref), (tra)\}$, $\{(id_q), (\exists_r), (\forall_l)\}$, $\{(nd), (cd)\}$, and (ihd) follow from Lem. 15, Lem. 16, Lem. 17, and Lem. 18, respectively. Admissibility of (\supset_l) and (\perp_l) is shown as stated in Thm. 8. Admissibility of (\forall_r) in the presence of (\forall_r^c) is shown below:

$$\frac{\frac{\mathcal{R}, w \leq v, \underline{a} \in D_v, \Gamma \Rightarrow v : A(\underline{a}/x), \Delta}{\mathcal{R}, w \leq w, \underline{a} \in D_w, \Gamma \Rightarrow w : A(\underline{a}/x), \Delta} (lsub)}{\frac{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow w : A(\underline{a}/x), \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \forall x A, \Delta} (ref)} (\forall_r^c)$$

□

Theorem 13. *(i) The calculus IntQCL is complete relative to treelike derivations with the fixed root property. (ii) Every derivation in IntQCL of a treelike labelled sequent is a treelike derivation with the fixed root property.*

Proof. Similar to Thm. 6. □

Although we relied on the notion of a nested form derivation (Def. 17) in the previous section to translate proofs from the refined labelled calculus IntQL into proofs in the nested calculus NIntQ , we note that this notion, or an analog thereof, is unnecessary to translate derivations from IntQCL to NIntQC . This follows from the fact that side conditions are not imposed on (\exists_r) and (\forall_l) in NIntQC —contrary to what is the case in NIntQ . Hence, applications of (\exists_r^c) and (\forall_l^c) in IntQCL need not satisfy certain condition in order to be translated, thus simplifying the translation from labelled to nested in the constant domain setting.

Theorem 14. *Every derivation in IntQCL of a nestedlike labelled sequent Λ is effectively translatable to a derivation of the nested sequent $\mathfrak{N}(\Lambda)$ in NIntQC .*

Proof. By Lem. 3, we know that Λ is treelike, implying that the derivation of Λ is treelike by Thm. 13, and so, the translation function \mathfrak{N} is defined. When translating from labelled to nested, the existential rules (\exists_r^c) and (\exists_r^{ic}) , and the universal rules (\forall_l^c) and (\forall_l^{ic}) , translate to instances of (\exists_r) and (\forall_l) , respectively. All remaining rules in IntQCL translate to their corresponding version in the nested calculus. □

Theorem 15. *Every derivation in NIntQC of a nested sequent Σ is effectively translatable to a derivation of the labelled sequent $\mathfrak{L}(\Sigma)$ in IntQCL .*

Proof. We show how to translate an instance of (\exists_r) ; the remaining cases are similar to those in the proof of Thm. 10. Suppose we aim to translate an (\exists_r) inference occurring in our input derivation:

$$\frac{\Sigma\{X \rightarrow A(\underline{a}/x), \exists x A, Y\}}{\Sigma\{X \rightarrow \exists x A, Y\}} (\exists_r)$$

If the parameter \underline{a} in the above inference is an eigenvariable, then we translate the inference to an instance of (\exists_r^{in}) in IntQCL . However, if the parameter occurs in a side formula, then the translation is non-trivial. Suppose then, that there exists some formula $B(\underline{a})$ (different than the occurrence of $A(\underline{a}/x)$) occurring in the premise of the (\exists_r) inference above. Suppose that the translation \mathfrak{L} assigns the label u to all formulae in the component where $B(\underline{a})$ occurs, and note that $u : B(\underline{a}) \in \Gamma, \Delta$ below. This, in conjunction with the definition of \mathfrak{L} (Def. 12), implies the existence of a domain atom $\underline{a} \in D_u$ in the image of $\Sigma\{X \rightarrow A(\underline{a}/x), \exists xA, Y\}$ under \mathfrak{L} (which is explicitly included in the derivation below). Since $\mathfrak{L}(\Sigma\{X \rightarrow A(\underline{a}/x), \exists xA, Y\})$ is a treelike labelled sequent, there must exist an undirected path $u \sim v_1, \dots, v_n \sim w$ from u to w in \mathcal{R} , i.e. $u \sim^{\mathcal{R}} w$. (NB. If $u = w$, then the translation is straightforward, so we omit consideration of that case.) If we weaken in domain atoms of the form $\underline{a} \in D_{v_i}$ (with $i \in \{1, \dots, n\}$) along this undirected path, invoke the admissibility of (nd) and (cd) (Thm. 12), and then apply (\exists_r^c) , we obtain the image of the conclusion of the (\exists_r) inference above under the \mathfrak{L} translation. This entire process is succinctly demonstrated in the derivation below:

$$\begin{array}{c}
 \mathfrak{L}(\Sigma\{X \rightarrow A(\underline{a}/x), \exists xA, Y\}) \\
 \dots\dots\dots = \\
 \mathcal{R}, \underline{a} \in D_u, \underline{a} \in D_w, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta \quad (wk) \\
 \hline
 \mathcal{R}, \underline{a} \in D_u, \underline{a} \in D_{v_1}, \dots, \underline{a} \in D_{v_n}, \underline{a} \in D_w, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta \quad (nd) \times k_1 + (cd) \times k_2 \\
 \hline
 \mathcal{R}, \underline{a} \in D_u, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta \quad (\exists_r^c) \\
 \hline
 \mathcal{R}, \underline{a} \in D_u, \Gamma \Rightarrow w : \exists xA, \Delta \\
 \dots\dots\dots = \\
 \mathfrak{L}(\Sigma\{X \rightarrow \exists xA, Y\})
 \end{array}$$

Note that $k_1 + k_2 = n + 1$, i.e. $k_1 + k_2$ is the number of labels occurring in the undirected path from u to w occurring after u and including w . \square

8 Reverse Translations, Corollaries, and Inheritance

The previous three sections were dedicated to refining the labelled calculi G3Int , G3IntQ , and G3IntQC to obtain refined labelled variants that were translatable into nested calculi. In this section, we analyze and provide the reverse translations from NInt , NIntQ , and NIntQC to G3Int , G3IntQ , and G3IntQC , respectively. These reverse translations are interesting in that they demonstrate how to add further syntactic structures and semantic information in order to obtain the more complicated semantic systems from the more refined nested systems. Additionally, these reverse translations can be leveraged to show that each nested calculus inherits proof theoretic properties from each labelled calculus, which we discuss in the second subsection.

8.1 Reverse Translations

When translating from labelled to nested, we discovered what new rules and what strengthened versions of rules were needed in order to allow the elimination of certain structural rules (e.g. (ref) and (nd)) from a derivation. Obtaining

the reverse translation is much simpler, as we already know what rules need to be shown admissible. Lem. 19–21 below show how to effectively transform derivations from IntL , IntQL , and IntQCL into G3Int , G3IntQ , and G3IntQC , respectively. These transformations, coupled with the translations from Thm. 7, 11, and 15, yield effective translations from our nested calculi to the G3-style labelled calculi.

Lemma 19 ([21]). *The rules (id^*) , (\supset_i^*) , and $(lift)$ are admissible in G3Int .*

Proof. The proof of the admissibility of each rule in G3Int is shown below:

$$\frac{\frac{\mathcal{R}, w \leq w, \Gamma, w : p \Rightarrow w : p, \Delta}{\mathcal{R}, \Gamma, w : p \Rightarrow w : p, \Delta} \begin{array}{l} (id) \\ (ref) \end{array}}{\mathcal{R}, w \leq w, \Gamma, w : p \Rightarrow w : p, \Delta} \frac{\frac{\mathcal{R}, w : A \supset B, \Gamma \Rightarrow \Delta, w : A}{\mathcal{R}, w \leq w, w : A \supset B, \Gamma \Rightarrow \Delta, w : A} (wk) \quad \frac{\mathcal{R}, w : A \supset B, w : B, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq w, w : A \supset B, w : B, \Gamma \Rightarrow \Delta} (wk)}{\frac{\mathcal{R}, w \leq w, w : A \supset B, \Gamma \Rightarrow \Delta}{\mathcal{R}, w : A \supset B, \Gamma \Rightarrow \Delta} (ref)} (\supset_i)$$

$$\frac{\frac{\mathcal{R}, w \leq v, \Gamma, w : A \Rightarrow v : A, \Delta}{\mathcal{R}, w \leq v, \Gamma, w : A \Rightarrow v : A, \Delta} \text{Thm. 3-(i)-(a)} \quad \frac{\mathcal{R}, w \leq v, \Gamma, w : A, v : A \Rightarrow \Delta}{\mathcal{R}, w \leq v, \Gamma, w : A, v : A \Rightarrow \Delta} (cut)}{\mathcal{R}, w \leq v, \Gamma, w : A \Rightarrow v : A, \Delta} (cut) \quad \square$$

Lemma 20. *The rules (id_q^n) , (\neg_l) , (\neg_r) , (\supset_i^*) , $(lift)$, (\exists_r^n) , (\forall_l^n) , (\exists_r^{in}) , and (\forall_l^{in}) are admissible in G3IntQ .*

Proof. The (id_q^n) rule can be derived by using (id_q) , (ref) , and (nd) . The admissibility of (\supset_i^*) and $(lift)$ are shown similarly as in Lem. 19. Recall that $\neg A := A \supset \perp$; using this fact, the proofs of the admissibility of (\neg_l) and (\neg_r) are as shown below:

$$\frac{\frac{\mathcal{R}, w : A \supset \perp, \Gamma \Rightarrow \Delta, w : A}{\mathcal{R}, w \leq w, w : A \supset \perp, \Gamma \Rightarrow \Delta, w : A} (wk) \quad \frac{\mathcal{R}, w : A \supset \perp, w : \perp, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq w, w : A \supset \perp, w : \perp, \Gamma \Rightarrow \Delta} (\perp_l)}{\frac{\mathcal{R}, w \leq w, w : A \supset \perp, \Gamma \Rightarrow \Delta}{\mathcal{R}, w : A \supset \perp, \Gamma \Rightarrow \Delta} (ref)} (\supset_i)$$

$$\frac{\frac{\mathcal{R}, w \leq v, \Gamma, v : A \Rightarrow \Delta}{\mathcal{R}, w \leq v, \Gamma, v : A \Rightarrow v : \perp, \Delta} (wk)}{\mathcal{R}, \Gamma \Rightarrow w : A \supset \perp, \Delta} (\supset_r)$$

The admissibility of (\exists_r^n) and (\forall_l^n) are shown below. We assume that there exists a directed path from v to w of a length greater than 0 in \mathcal{R} , i.e. $v \leq u_1, \dots, u_n \leq w \in \mathcal{R}$. The cases where the directed path from v to w is of length 0 (i.e. $v = w$) are simple and are handled similarly.

$$\frac{\frac{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists x A}{\mathcal{R}, \underline{a} \in D_v, \underline{a} \in D_{u_1}, \dots, \underline{a} \in D_{u_n}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists x A} (wk)}{\frac{\mathcal{R}, \underline{a} \in D_v, \underline{a} \in D_{u_1}, \dots, \underline{a} \in D_{u_n}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta, w : \exists x A}{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \exists x A} (nd) \times (n+1)}$$

$$\begin{array}{c}
 \frac{\mathcal{R}, \underline{a} \in D_v, \Gamma, w : A(\underline{a}/x), w : \forall xA \Rightarrow \Delta}{\mathcal{R}, w \leq w, \underline{a} \in D_v, \underline{a} \in D_{u_1}, \dots, \underline{a} \in D_{u_n}, \underline{a} \in D_w, \Gamma, w : A(\underline{a}/x), w : \forall xA \Rightarrow \Delta} \text{ (wk)} \\
 \frac{\mathcal{R}, w \leq w, \underline{a} \in D_v, \underline{a} \in D_{u_1}, \dots, \underline{a} \in D_{u_n}, \underline{a} \in D_w, \Gamma, w : \forall xA \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_v, \underline{a} \in D_{u_1}, \dots, \underline{a} \in D_{u_n}, \underline{a} \in D_w, \Gamma, w : \forall xA \Rightarrow \Delta} \text{ (}\forall\text{)} \\
 \frac{\mathcal{R}, \underline{a} \in D_v, \underline{a} \in D_{u_1}, \dots, \underline{a} \in D_{u_n}, \underline{a} \in D_w, \Gamma, w : \forall xA \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_v, \Gamma, w : \forall xA \Rightarrow \Delta} \text{ (ref)} \\
 \frac{\mathcal{R}, \underline{a} \in D_v, \Gamma, w : \forall xA \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_v, \Gamma, w : \forall xA \Rightarrow \Delta} \text{ (nd)} \times (n+1)
 \end{array}$$

Similar to the previous two proofs, to show the admissibility of (\exists_r^{in}) and (\forall_l^{in}) , we assume that $v \leq u_1, \dots, u_n \leq w \in \mathcal{R}$, i.e. there exists a directed path of length greater than 0 in \mathcal{R} . The cases where the directed path from v to w is of length 0 (i.e. $v = w$) are simple and are handled similarly.

$$\begin{array}{c}
 \frac{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists xA}{\mathcal{R}, \underline{a} \in D_v, \underline{a} \in D_{u_1}, \dots, \underline{a} \in D_{u_n}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists xA} \text{ (wk)} \\
 \frac{\mathcal{R}, \underline{a} \in D_v, \underline{a} \in D_{u_1}, \dots, \underline{a} \in D_{u_n}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta, w : \exists xA}{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \exists xA} \text{ (}\exists\text{)} \\
 \frac{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \exists xA}{\mathcal{R}, \Gamma \Rightarrow \Delta, w : \exists xA} \text{ (nd)} \times (n+1) \\
 \frac{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \exists xA}{\mathcal{R}, \Gamma \Rightarrow \Delta, w : \exists xA} \text{ (ihd)}
 \end{array}$$

$$\begin{array}{c}
 \frac{\mathcal{R}, \underline{a} \in D_v, \Gamma, w : A(\underline{a}/x), w : \forall xA \Rightarrow \Delta}{\mathcal{R}, w \leq w, \underline{a} \in D_v, \underline{a} \in D_{u_1}, \dots, \underline{a} \in D_{u_n}, \underline{a} \in D_w, \Gamma, w : A(\underline{a}/x), w : \forall xA \Rightarrow \Delta} \text{ (wk)} \\
 \frac{\mathcal{R}, w \leq w, \underline{a} \in D_v, \underline{a} \in D_{u_1}, \dots, \underline{a} \in D_{u_n}, \underline{a} \in D_w, \Gamma, w : \forall xA \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_v, \underline{a} \in D_{u_1}, \dots, \underline{a} \in D_{u_n}, \underline{a} \in D_w, \Gamma, w : \forall xA \Rightarrow \Delta} \text{ (}\forall\text{)} \\
 \frac{\mathcal{R}, \underline{a} \in D_v, \underline{a} \in D_{u_1}, \dots, \underline{a} \in D_{u_n}, \underline{a} \in D_w, \Gamma, w : \forall xA \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_v, \Gamma, w : \forall xA \Rightarrow \Delta} \text{ (ref)} \\
 \frac{\mathcal{R}, \underline{a} \in D_v, \Gamma, w : \forall xA \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \forall xA \Rightarrow \Delta} \text{ (nd)} \times (n+1) \\
 \frac{\mathcal{R}, \underline{a} \in D_v, \Gamma, w : \forall xA \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \forall xA \Rightarrow \Delta} \text{ (ihd)}
 \end{array}$$

□

Lemma 21. *The rules (id_q^c) , (\neg_l) , (\neg_r) , (\supset_l^*) , $(lift)$, (\forall_r^c) , (\exists_r^c) , (\forall_l^c) , (\exists_r^{ic}) , and (\forall_l^{ic}) are admissible in G3IntQC.*

Proof. The admissibility of (id_q^c) , (\neg_l) , (\neg_r) , (\supset_l^*) , $(lift)$, (\exists_r^c) , (\forall_l^c) , (\exists_r^{ic}) , and (\forall_l^{ic}) are shown similarly to the admissibility of the corresponding rules in G3IntQ (Lem. 20 above). The admissibility of (\forall_r^c) is shown below; we assume that v is a fresh label.

$$\begin{array}{c}
 \frac{\frac{\frac{\Pi_1}{\mathcal{R}, w \leq v, \underline{a} \in D_w, \underline{a} \in D_v, \Gamma \Rightarrow v : A(\underline{a}/x), \Delta} \text{ (cut)}}{\mathcal{R}, w \leq v, \underline{a} \in D_w, \underline{a} \in D_v, \Gamma \Rightarrow v : A(\underline{a}/x), \Delta} \text{ (cd)}}{\mathcal{R}, \Gamma \Rightarrow w : \forall xA, \Delta} \text{ (}\forall\text{)} \\
 \Pi_1 = \left\{ \frac{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow w : A(\underline{a}/x), \Delta}{\mathcal{R}, w \leq v, \underline{a} \in D_w, \underline{a} \in D_v, \Gamma \Rightarrow w : A(\underline{a}/x), v : A(\underline{a}/x), \Delta} \text{ (wk)} \right. \\
 \left. \Pi_2 = \left\{ \frac{\mathcal{R}, w \leq v, \underline{a} \in D_w, \underline{a} \in D_v, \Gamma, w : A(\underline{a}/x) \Rightarrow v : A(\underline{a}/x), \Delta}{\mathcal{R}, w \leq v, \underline{a} \in D_w, \underline{a} \in D_v, \Gamma, w : A(\underline{a}/x) \Rightarrow v : A(\underline{a}/x), \Delta} \text{ Thm. 3-(i)-(c)} \right. \right. \\
 \left. \left. \right. \right.
 \end{array}$$

□

Theorem 16. *Every derivation of a labelled sequent Λ in IntL, IntQL, and IntQCL can be effectively transformed into a derivation of Λ in G3Int, G3IntQ, and G3IntQC, respectively, and vice-versa.*

Proof. The forward directions follow from Thm. 5, 8, and 12. The backward directions follow from Lem. 19, 20, and 21, respectively. \square

Theorem 17. (i) *If a nestedlike labelled sequent A is derivable in $\mathbf{G3Int}$, $\mathbf{G3IntQ}$, and $\mathbf{G3IntQC}$, then $\mathfrak{N}(A)$ is derivable in \mathbf{NInt} , \mathbf{NIntQ} , and \mathbf{NIntQC} , respectively.*
(ii) *Every derivation in \mathbf{NInt} , \mathbf{NIntQ} , and \mathbf{NIntQC} of a nested sequent Σ can be effectively translated into a derivation of the labelled sequent $\mathfrak{L}(\Sigma)$ in $\mathbf{G3Int}$, $\mathbf{G3IntQ}$, and $\mathbf{G3IntQC}$, respectively.*

Proof. Follows from Thm. 7, 10, 11, 14, 15, and 16. \square

8.2 Corollaries and Inheritance

An appealing consequence of our refinement and translation procedures is that our nested calculi inherit desirable proof-theoretic properties from their associated labelled calculi. Such properties are stated in Cor. 1–4 below.

As mentioned near the end of Sect. 6, the translation of proofs from \mathbf{IntQL} into proofs in \mathbf{NIntQ} (Thm. 10) relies on the completeness of the latter (Thm. 4); hence, we are barred from leveraging this proof-theoretic translation to conclude that \mathbf{NIntQ} ultimately inherits completeness from $\mathbf{G3IntQ}$. (NB. In Sect. 6, we also discussed methods of proof that would free us from relying on the completeness of \mathbf{NIntQ} , thus allowing for this inheritance result to go through.) Nevertheless, this issue does not arise in our work concerning propositional intuitionistic logic \mathbf{Int} and first-order intuitionistic logic with constant domains \mathbf{IntQC} , and so, we may conclude the following corollary:

Corollary 1. *The calculi \mathbf{NInt} and \mathbf{NIntQC} have inherited soundness and completeness from $\mathbf{G3Int}$ and $\mathbf{G3IntQC}$, respectively.*

Proof. Soundness follows from Thm. 3 and 17. Regarding completeness, observe that if A is a theorem of \mathbf{Int} or \mathbf{IntQC} , then $\vec{a} \in D_w \Rightarrow w : A$ is derivable in $\mathbf{G3Int}$ or $\mathbf{G3IntQC}$ (respectively) by Thm. 3. By Thm. 5 and 7, or Thm. 12 and 14, the derivation of $\vec{a} \in D_w \Rightarrow w : A$ may be transformed and translated into a derivation of A in \mathbf{NInt} or \mathbf{NIntQC} , respectively. \square

Furthermore, we obtain the admissibility of the set of structural rules presented in Fig. 8. This set of structural rules was primarily identified by investigating the variety of ways in which labelled structural rules (e.g. (wk) , (ctr_r) , etc.) can be applied to manipulate treelike labelled sequents. Therefore, such rules offer a more fine-grained view of the types of operations inherent in the more general functionality of the aforementioned labelled structural rules. Although the admissibility of the rules in Fig. 8 can be established by confirming the soundness of each rule, and recognizing that each nested calculus is complete (Thm. 4), we provide proof-theoretic arguments which demonstrate how such rules are obtained from their associated labelled structural rules. The proof of this admissibility result (Cor. 2 below) can be found in Appendix A for the interested reader.

$$\begin{array}{c}
 \frac{\Sigma\{X \rightarrow Y\}}{\Sigma\{X, Z \rightarrow Y\}} (wk_l) \quad \frac{\Sigma\{X \rightarrow Y\}}{\Sigma\{X \rightarrow Y, Z\}} (wk_r) \quad \frac{\Sigma}{\Sigma(\underline{a}/\underline{b})} (psub) \\
 \\
 \frac{\Sigma\{X, Z, Z \rightarrow Y\}}{\Sigma\{X, Z \rightarrow Y\}} (ctr_l) \quad \frac{\Sigma\{X \rightarrow Z, Z, Y\}}{\Sigma\{X \rightarrow Z, Y\}} (ctr_r) \quad \frac{\Sigma}{\rightarrow [\Sigma]} (nr) \\
 \\
 \frac{\Sigma\{X \rightarrow Y, [\Sigma'], [\Sigma']\}}{\Sigma\{X \rightarrow Y, [\Sigma']\}} (mrg_1) \quad \frac{\Sigma\{X_1 \rightarrow Y_1, [X_2 \rightarrow Y_2, [\Sigma_1], \dots, [\Sigma_n]]\}}{\Sigma\{X_1, X_2 \rightarrow Y_1, Y_2, [\Sigma_1], \dots, [\Sigma_n]\}} (mrg_2) \\
 \\
 \frac{\Sigma\{X \rightarrow Y\}}{\Sigma\{X \rightarrow Y, [\Sigma']\}} (ew_1) \quad \frac{\Sigma\{X_1 \rightarrow Y_1, [X_2 \rightarrow Y_2, [\Sigma_1], \dots, [\Sigma_n]]\}}{\Sigma\{X_1 \rightarrow Y_1, [\rightarrow [X_2 \rightarrow Y_2, [\Sigma_1], \dots, [\Sigma_n]]]\}} (ew_2) \\
 \\
 \frac{\Sigma\{X_1 \rightarrow A, Y_1, [X_2 \rightarrow A, Y_2]\}}{\Sigma\{X_1 \rightarrow Y_1, [X_2 \rightarrow A, Y_2]\}} (lwr) \quad \frac{\Sigma\{X \rightarrow A, Y\} \quad \Sigma\{X, A \rightarrow Y\}}{\Sigma\{X \rightarrow Y\}} (cut)
 \end{array}$$

Fig. 8. The set NR of admissible rules in NInt, NIntQ, and NIntQC.

Corollary 2. *All rules in NR are admissible in NInt, NIntQ and NIntQC.*

Additionally, we obtain the invertibility of the rules in our nested calculi:

Corollary 3. *All rules in NInt, NIntQ and NIntQC are invertible.*

Proof. Follows from Lem. 4 and Thm. 17. □

By making use of the previous two corollaries, we can prove that derivability of a nested sequent is equivalent to the derivability of its formula interpretation (Def. 7). Note that in the propositional case the universal closure $\forall \vec{x}$ is vacuous, and so, $\rightarrow \forall \vec{x} \iota(\Sigma)$ is identical to $\rightarrow \iota(\Sigma)$.

Corollary 4. *The nested sequent Σ is derivable in NInt, NIntQ and NIntQC iff the nested sequent $\rightarrow \forall \vec{x} \iota(\Sigma)$ is derivable in NInt, NIntQ and NIntQC, respectively.*

Proof. We begin with the forward direction, and assume that Σ is derivable in one of our nested systems. By the admissibility of (nr) (Cor. 2), we know that $\rightarrow [\Sigma]$ is derivable. Starting at the leaves of the nested sequent and continuously applying (\wedge_l) , (\vee_r) , and (\supset_r) , we eventually obtain a derivation of $\rightarrow \iota(\Sigma)$. From here, we have three cases to consider: (i) If we are working in the nested system NInt, then we are done. (ii) If we are working in the nested system NIntQ, and $\rightarrow \iota(\Sigma)$ contains no free variables (i.e. parameters), then we are done. Otherwise, if the free variables (i.e. parameters) $\underline{a}_1, \dots, \underline{a}_n$ occur in $\iota(\Sigma)$, then we apply the (nr) rule n times to obtain:

$$\underbrace{\rightarrow [\dots [\rightarrow \iota(\Sigma)] \dots]}_n$$

Applying the (\forall_r^n) rule n times gives a derivation of $\rightarrow \forall \vec{x} \iota(\Sigma)$. (iii) If we are working in the nested system \mathbf{NIntQC} and $\rightarrow \iota(\Sigma)$ does not contain any occurrences of free variables (i.e. parameters), then we are done. Otherwise, we repeatedly apply (\forall_r^c) to obtain a derivation of the desired conclusion.

For the backwards direction, we invoke the invertibility of (\wedge_l) , (\forall_r) , (\supset_r) , (\forall_r^n) , and (\forall_r^c) (Cor. 3), as well as the admissibility of (mrg_2) (Cor. 3), to obtain a derivation of Σ . \square

Last, by making use of the admissibility of (mrg_2) , we can show that each derivation in \mathbf{NIntQ} can be transformed into a derivation in \mathbf{NIntQC} . We also show that the same embedding holds between \mathbf{IntQL} and \mathbf{IntQCL} to complete our mapping of the relationships mentioned in Fig. 1 of Sect. 1.

Corollary 5. (i) *Every derivation in \mathbf{IntQL} is transformable into a derivation in \mathbf{IntQCL} .* (ii) *Every derivation in \mathbf{NIntQ} is transformable into a derivation in \mathbf{NIntQC} .*

Proof. Observe that the (\exists_r^n) , (\exists_r^{in}) , (\forall_l^n) , and (\forall_l^{in}) rules in \mathbf{IntQL} are instances of the (\exists_r^c) , (\exists_r^{ic}) , (\forall_l^c) , and (\forall_l^{ic}) rules in \mathbf{IntQCL} . Additionally, the same relationship holds true of the (\exists_r) and (\forall_l) rules in \mathbf{NIntQ} and \mathbf{NIntQC} . Thus, the only rules that differ between the systems \mathbf{IntQL} and \mathbf{IntQCL} are the (\forall_r) and (\forall_r^c) rules, and the only rules that differ between the systems \mathbf{NIntQ} and \mathbf{NIntQC} are the (\forall_r^n) and (\forall_r^c) rules. Therefore, we show below that (\forall_r) and (\forall_r^n) are admissible in \mathbf{IntQCL} and \mathbf{NIntQC} , respectively, which proves the claim since all remaining cases are straightforward.

$$\frac{\frac{\mathcal{R}, w \leq v, \underline{a} \in D_v, \Gamma \Rightarrow v : A(\underline{a}/x), \Delta}{\mathcal{R}, w \leq w, \underline{a} \in D_w, \Gamma \Rightarrow w : A(\underline{a}/x), \Delta} \text{ (lsub)} \quad \frac{\Sigma\{X \rightarrow Y, [\rightarrow A(\underline{a}/x)]\}}{\Sigma\{X \rightarrow A(\underline{a}/x), Y\}} \text{ (mrg}_2)}{\frac{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow w : A(\underline{a}/x), \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \forall x A, \Delta} \text{ (}\forall_r^c\text{)} \quad \frac{\Sigma\{X \rightarrow A(\underline{a}/x), Y\}}{\Sigma\{X \rightarrow \forall x A, Y\}} \text{ (}\forall_r^c\text{)}}{\square}$$

9 Conclusion

In this paper, we mapped out the interrelationships between semantic systems and Fitting's nested calculi for intuitionistic logics. We focused extensively on *how* to eliminate the structural rules of the labelled calculi $\mathbf{G3Int}$, $\mathbf{G3IntQ}$, and $\mathbf{G3IntQC}$ from derivations, by considering how to strengthen their logical rules, or via the addition of new rules. We referred to this process as the method of *refinement* since it ultimately yielded the more refined (sound and complete) calculi \mathbf{IntL} , \mathbf{IntQL} , and \mathbf{IntQCL} which omitted the use of semantically explicit structural rules (e.g. *ref*) and (*nd*) and only required treelike labelled sequents in derivations. We saw that fragments of these refined labelled calculi were equivalent to Fitting's nested calculi [9]. Moreover, we also observed that Fitting's nested calculi inherited proof-theoretic properties from their associated labelled calculi; e.g. soundness and completeness, invertibility of rules, and admissibility of (*cut*).

Such results suggest that structural rule elimination is a natural method for connecting the relational semantics of a logic to a nested calculus for the logic (via the labelled sequent paradigm).

The method of refining labelled calculi into nested calculi for intuitionistic logics is part of a larger project that seeks to investigate the extent to which such refinements can be performed. Due to general results concerning the construction of labelled calculi, their modularity, and the confirmation of their proof-theoretic properties [5,6,11,24], the exposition of a method for deriving refined (or, nested) calculi—which utilize sequents with a simpler underlying data structure (e.g. trees) and have proven to be well-suited for certain automated reasoning tasks (e.g. proof-search [25], counter-model extraction [17,25], and effective interpolation [19])—from semantic systems could prove to be of practical consequence. Moreover, since labelled calculi are easily obtained from a logic’s semantics, the method of refinement provides a means by which (nested) proof calculi with favorable properties may be obtained directly from a logic’s semantics.

Acknowledgments. Work funded by FWF projects I2982 and W1255-N23.

References

1. Avron, A.: The method of hypersequents in the proof theory of propositional non-classical logics. In: W. Hodges, M. Hyland, C. Steinhorn, J. Truss (eds.) *Logic: From Foundations to Applications: European Logic Colloquium*, p. 1–32. Clarendon Press, USA (1996)
2. Belnap, N.D.: Display logic. *Journal of philosophical logic* **11**(4), 375–417 (1982)
3. Brünnler, K.: Deep sequent systems for modal logic. *Arch. Math. Log.* **48**(6), 551–577 (2009). DOI 10.1007/s00153-009-0137-3. URL <https://doi.org/10.1007/s00153-009-0137-3>
4. Bull, R.A.: Cut elimination for propositional dynamic logic without *. *Z. Math. Logik Grundlag. Math.* **38**(2), 85–100 (1992)
5. Ciabattoni, A., Maffezioli, P., Spindler, L.: Hypersequent and labelled calculi for intermediate logics. In: D. Galmiche, D. Larchey-Wendling (eds.) *Automated Reasoning with Analytic Tableaux and Related Methods, Lecture Notes in Computer Science*, vol. 8123, pp. 81–96. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
6. Dyckhoff, R., Negri, S.: Proof analysis in intermediate logics. *Archive for Mathematical Logic* **51**(1-2), 71–92 (2012)
7. Fitting, M.: Tableau methods of proof for modal logics. *Notre Dame Journal of Formal Logic* **13**(2), 237–247 (1972)
8. Fitting, M.: Prefixed tableaux and nested sequents. *Annals of Pure and Applied Logic* **163**(3), 291 – 313 (2012). DOI <https://doi.org/10.1016/j.apal.2011.09.004>
9. Fitting, M.: Nested sequents for intuitionistic logics. *Notre Dame Journal of Formal Logic* **55**(1), 41–61 (2014)
10. Gabbay, D., Shehtman, V., Skvortsov, D.: *Quantification in Non-classical Logics. Studies in Logic and Foundations of Mathematics*. Elsevier (2009)
11. Gabbay, D.M.: *Labelled deductive systems, Oxford Logic guides*, vol. 33. Clarendon Press/Oxford Science Publications (1996)
12. Gentzen, G.: Untersuchungen über das logische schliessen. *Mathematische Zeitschrift* **39**(3), 405–431 (1935)

13. Goré, R., Ramanayake, R.: Labelled tree sequents, tree hypersequents and nested (deep) sequents. In: T. Bolander, T. Braüner, S. Ghilardi, L.S. Moss (eds.) *Advances in Modal Logic 9*, papers from the ninth conference on "Advances in Modal Logic," held in Copenhagen, Denmark, 22-25 August 2012, pp. 279–299. College Publications (2012). URL <http://www.aiml.net/volumes/volume9/Gore-Ramanayake.pdf>
14. Kashima, R.: Cut-free sequent calculi for some tense logics. *Studia Logica* **53**(1), 119–135 (1994)
15. Kripke, S.A.: Semantical analysis of intuitionistic logic i. In: J. Crossley, M. Dummett (eds.) *Formal Systems and Recursive Functions, Studies in Logic and the Foundations of Mathematics*, vol. 40, pp. 92 – 130. Elsevier (1965). DOI [https://doi.org/10.1016/S0049-237X\(08\)71685-9](https://doi.org/10.1016/S0049-237X(08)71685-9). URL <http://www.sciencedirect.com/science/article/pii/S0049237X08716859>
16. Lyon, T.: On deriving nested calculi for intuitionistic logics from semantic systems. In: S.N. Artëmov, A. Nerode (eds.) *Logical Foundations of Computer Science - International Symposium, LFCS 2020*, Deerfield Beach, FL, USA, January 4-7, 2020, Proceedings, *Lecture Notes in Computer Science*, vol. 11972, pp. 177–194. Springer (2020). DOI [10.1007/978-3-030-36755-8_12](https://doi.org/10.1007/978-3-030-36755-8_12). URL https://doi.org/10.1007/978-3-030-36755-8_12
17. Lyon, T., van Berkel, K.: Automating agential reasoning: Proof-calculi and syntactic decidability for stit logics. In: M. Baldoni, M. Dastani, B. Liao, Y. Sakurai, R. Zalila Wenkstern (eds.) *PRIMA 2019: Principles and Practice of Multi-Agent Systems - 22nd International Conference*, Turin, Italy, October 28-31, 2019, Proceedings, *Lecture Notes in Computer Science*, vol. 11873, pp. 202–218. Springer International Publishing, Cham (2019)
18. Lyon, T., Ittner, C., Eckhardt, T., Gratzl, N.: The basics of display calculi. *Kriterion – Journal of Philosophy* **31**(2), 55–100 (2017)
19. Lyon, T., Tiu, A., Goré, R., Clouston, R.: Syntactic interpolation for tense logics and bi-intuitionistic logic via nested sequents. In: M. Fernández, A. Muscholl (eds.) *28th EACSL Annual Conference on Computer Science Logic, CSL 2020*, January 13-16, 2020, Barcelona, Spain, *LIPICs*, vol. 152, pp. 28:1–28:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020). DOI [10.4230/LIPICs.CSL.2020.28](https://doi.org/10.4230/LIPICs.CSL.2020.28). URL <https://doi.org/10.4230/LIPICs.CSL.2020.28>
20. Minari, P.: Labeled sequent calculi for modal logics and implicit contractions. *Arch. Math. Log.* **52**(7-8), 881–907 (2013). DOI [10.1007/s00153-013-0350-y](https://doi.org/10.1007/s00153-013-0350-y)
21. Pimentel, E.: A semantical view of proof systems. In: L.S. Moss, R.J.G.B. de Queiroz, M. Martinez (eds.) *Logic, Language, Information, and Computation - 25th International Workshop, WoLLIC 2018*, Bogota, Colombia, July 24-27, 2018, Proceedings, *Lecture Notes in Computer Science*, vol. 10944, pp. 61–76. Springer (2018). DOI [10.1007/978-3-662-57669-4_3](https://doi.org/10.1007/978-3-662-57669-4_3). URL https://doi.org/10.1007/978-3-662-57669-4_3
22. Poggiolesi, F.: A cut-free simple sequent calculus for modal logic $s5$. *The Review of Symbolic Logic* **1**(1), 3–15 (2008)
23. Poggiolesi, F.: The method of tree-hypersequents for modal propositional logic. In: D. Makinson, J. Malinowski, H. Wansing (eds.) *Towards Mathematical Philosophy, Trends in logic*, vol. 28, pp. 31–51. Springer (2009). DOI [10.1007/978-1-4020-9084-4_3](https://doi.org/10.1007/978-1-4020-9084-4_3). URL https://doi.org/10.1007/978-1-4020-9084-4_3
24. Simpson, A.K.: The proof theory and semantics of intuitionistic modal logic. Ph.D. thesis, University of Edinburgh. College of Science and Engineering. School of Informatics (1994)

25. Tiu, A., Ianovski, E., Goré, R.: Grammar logics in nested sequent calculus: Proof theory and decision procedures. In: T. Bolander, T. Braüner, S. Ghilardi, L.S. Moss (eds.) *Advances in Modal Logic 9*, papers from the ninth conference on "Advances in Modal Logic," held in Copenhagen, Denmark, 22-25 August 2012, pp. 516–537. College Publications (2012). URL <http://www.aiml.net/volumes/volume9/Tiu-Ianovski-Gore.pdf>
26. Troelstra, A., van Dalen, D.: *Constructivism in Mathematics*, vol. 1. Elsevier Science (1988)
27. Viganò, L.: *Labelled Non-Classical Logics*. Springer Science & Business Media (2000)
28. Wansing, H.: Sequent calculi for normal modal propositional logics. *Journal of Logic and Computation* **4**(2), 125–142 (1994)

A Proofs

Theorem 3. *Let $G3X \in \{G3IntQ, G3IntQC\}$. The calculi $G3Int$, $G3IntQ$, and $G3IntQC$ have the following properties:*

- (i) (a) For all $A \in \mathcal{L}$, $\vdash_{G3Int} \mathcal{R}, w \leq v, w : A, \Gamma \Rightarrow v : A, \Delta$;
 (b) For all $A \in \mathcal{L}$, $\vdash_{G3Int} \mathcal{R}, w : A, \Gamma \Rightarrow \Delta, w : A$;
 (c) For all $A \in \mathcal{Q}$, $\vdash_{G3X} \mathcal{R}, w \leq v, \vec{a} \in D_w, w : A(\vec{a}), \Gamma \Rightarrow v : A(\vec{a}), \Delta$;
 (d) For all $A \in \mathcal{Q}$, $\vdash_{G3X} \mathcal{R}, \vec{a} \in D_w, w : A(\vec{a}), \Gamma \Rightarrow \Delta, w : A(\vec{a})$;
- (ii) All rules in $LR - \{\text{cut}\}$ are hp-admissible;
- (iii) All rules are hp-invertible;
- (iv) The (cut) rule is admissible;
- (v) $G3Int$, $G3IntQ$, and $G3IntQC$ are sound and complete for Int , $IntQ$, and $IntQC$, respectively.

Proof. We argue that properties (i)–(v) obtain for $G3IntQC$; by omitting consideration of the (cd) rule, we obtain proofs of these properties for $G3IntQ$ as well.

Claim (i). Claims (a) and (b) are shown in [6, Lem. 1], so we focus on proving (c) and (d). We prove claims (c) and (d) together by mutual induction on the complexity of A .

Claim (i)-(c). The base case is resolved using the (id_q) rule for atomic formulae and the (\perp_l) rule for \perp . We provide the cases for \supset , \exists , and \forall for the inductive step; the \vee and \wedge cases are simple to verify. Note that for the \supset case the parameters \vec{a}_1 and \vec{a}_2 are all and only those parameters that occur in \vec{a} , with \vec{a}_1 and \vec{a}_2 potentially intersecting. Also, the \supset and \forall cases rely on claim (d) (shown below), and we occasionally use primes (e.g. Γ' and Δ') to indicate that formulae explicitly occurring lower in the proof have been absorbed into the context in order to save space. Last, we let $\vec{a} := a_1, \dots, a_n$.

$$\frac{\frac{\frac{\Pi_1 \quad \Pi_2}{\mathcal{R}, w \leq v, v \leq u, w \leq u, \vec{a} \in D_w, \vec{a} \in D_u, w : A(\vec{a}_1) \supset B(\vec{a}_2), u : A(\vec{a}_1), \Gamma \Rightarrow u : B(\vec{a}_2), \Delta} (\supset_l)}{\mathcal{R}, w \leq v, v \leq u, w \leq u, \vec{a} \in D_w, w : A(\vec{a}_1) \supset B(\vec{a}_2), u : A(\vec{a}_1), \Gamma \Rightarrow u : B(\vec{a}_2), \Delta} (nd) \times n}{\mathcal{R}, w \leq v, v \leq u, \vec{a} \in D_w, w : A(\vec{a}_1) \supset B(\vec{a}_2), u : A(\vec{a}_1), \Gamma \Rightarrow u : B(\vec{a}_2), \Delta} (tra)}{\mathcal{R}, w \leq v, \vec{a} \in D_w, w : A(\vec{a}_1) \supset B(\vec{a}_2), \Gamma \Rightarrow v : A(\vec{a}_1) \supset B(\vec{a}_2), \Delta} (\supset_r)$$

$$\begin{aligned}
 \Pi_1 &= \left\{ \frac{\overline{\mathcal{R}'}, \overline{\vec{a}} \in D_w, \overline{\vec{a}} \in D_u, u : A(\overline{\vec{a}_1}), \Gamma' \Rightarrow u : A(\overline{\vec{a}_1}), \Delta'}{\overline{\mathcal{R}'}, \overline{\vec{a}} \in D_w, \overline{\vec{a}} \in D_u, u : A(\overline{\vec{a}_1}), \Gamma' \Rightarrow u : A(\overline{\vec{a}_1}), \Delta'} \text{ (d)} \right. \\
 \Pi_2 &= \left\{ \frac{\overline{\mathcal{R}'}, \overline{\vec{a}} \in D_w, \overline{\vec{a}} \in D_u, u : B(\overline{\vec{a}_2}), \Gamma' \Rightarrow u : B(\overline{\vec{a}_2}), \Delta'}{\overline{\mathcal{R}'}, \overline{\vec{a}} \in D_w, \overline{\vec{a}} \in D_u, u : B(\overline{\vec{a}_2}), \Gamma' \Rightarrow u : B(\overline{\vec{a}_2}), \Delta'} \text{ (d)} \right. \\
 &\frac{\overline{\mathcal{R}, w \leq v, \vec{b} \in D_w, \overline{\vec{a}} \in D_w, w : A(\overline{\vec{a}})(\vec{b}/x), \Gamma \Rightarrow v : A(\overline{\vec{a}})(\vec{b}/x), \Delta'}{\overline{\mathcal{R}, w \leq v, \vec{b} \in D_w, \overline{\vec{a}} \in D_w, w : A(\overline{\vec{a}})(\vec{b}/x), \Gamma \Rightarrow v : \exists x A(\overline{\vec{a}}), \Delta}} \text{ IH} \text{ } (\exists_r) \\
 &\frac{\overline{\mathcal{R}, w \leq v, \vec{a} \in D_w, w : \exists x A(\overline{\vec{a}}), \Gamma \Rightarrow v : \exists x A(\overline{\vec{a}}), \Delta}}{\overline{\mathcal{R}, w \leq v, \vec{a} \in D_w, w : \exists x A(\overline{\vec{a}}), \Gamma \Rightarrow v : \exists x A(\overline{\vec{a}}), \Delta}} \text{ } (\exists_l) \\
 &\frac{\overline{\mathcal{R}', w \leq u, \overline{\vec{a}} \in D_u, \vec{b} \in D_u, u : A(\overline{\vec{a}})(\vec{b}/x), w : \forall x A(\overline{\vec{a}}), \Gamma \Rightarrow u : A(\overline{\vec{a}})(\vec{b}/x), \Delta}}{\overline{\mathcal{R}', w \leq u, \overline{\vec{a}} \in D_u, \vec{b} \in D_u, u : \forall x A(\overline{\vec{a}}), \Gamma \Rightarrow u : A(\overline{\vec{a}})(\vec{b}/x), \Delta}} \text{ IH} \text{ } (\forall_l) \\
 &\frac{\overline{\mathcal{R}', w \leq u, \vec{a} \in D_u, \vec{b} \in D_u, w : \forall x A(\overline{\vec{a}}), \Gamma \Rightarrow u : A(\overline{\vec{a}})(\vec{b}/x), \Delta}}{\overline{\mathcal{R}', w \leq u, \vec{a} \in D_u, \vec{b} \in D_u, w : \forall x A(\overline{\vec{a}}), \Gamma \Rightarrow u : A(\overline{\vec{a}})(\vec{b}/x), \Delta}} \text{ (nd)} \times n \\
 &\frac{\overline{\mathcal{R}', w \leq u, \vec{b} \in D_u, w : \forall x A(\overline{\vec{a}}), \Gamma \Rightarrow u : A(\overline{\vec{a}})(\vec{b}/x), \Delta}}{\overline{\mathcal{R}', \vec{b} \in D_u, w : \forall x A(\overline{\vec{a}}), \Gamma \Rightarrow u : A(\overline{\vec{a}})(\vec{b}/x), \Delta}} \text{ (tra)} \\
 &\frac{\overline{\mathcal{R}, w \leq v, \vec{a} \in D_w, w : \forall x A(\overline{\vec{a}}), \Gamma \Rightarrow v : \forall x A(\overline{\vec{a}}), \Delta}}{\overline{\mathcal{R}, w \leq v, \vec{a} \in D_w, w : \forall x A(\overline{\vec{a}}), \Gamma \Rightarrow v : \forall x A(\overline{\vec{a}}), \Delta}} \text{ } (\forall_r)
 \end{aligned}$$

In the \forall case above, we let $\mathcal{R}' := \mathcal{R}, w \leq v, v \leq u, \vec{a} \in D_w$.

Claim (i)-(d). The base case for atomic formulae is shown below; the case for \perp is omitted as it is simple to verify using the (\perp_l) rule. We provide the \supset , \exists , and \forall cases as the cases for the other connectives are straightforward to prove.

$$\begin{aligned}
 &\frac{\overline{\mathcal{R}, w \leq w, \vec{a} \in D_w, w : p(\vec{a}), \Gamma \Rightarrow w : p(\vec{a}), \Delta}}{\overline{\mathcal{R}, \vec{a} \in D_w, w : p(\vec{a}), \Gamma \Rightarrow w : p(\vec{a}), \Delta}} \text{ (id}_q) \text{ } (ref) \\
 &\frac{\overline{\mathcal{R}, w \leq v, \overline{\vec{a}} \in D_w, \overline{\vec{a}} \in D_v, w : A(\overline{\vec{a}_1}) \supset B(\overline{\vec{a}_2}), v : A(\overline{\vec{a}_1}), \Gamma \Rightarrow v : B(\overline{\vec{a}_2}), \Delta}}{\overline{\mathcal{R}, w \leq v, \overline{\vec{a}} \in D_w, w : A(\overline{\vec{a}_1}) \supset B(\overline{\vec{a}_2}), v : A(\overline{\vec{a}_1}), \Gamma \Rightarrow v : B(\overline{\vec{a}_2}), \Delta}} \text{ } (\supset_l) \\
 &\frac{\overline{\mathcal{R}, w \leq v, \overline{\vec{a}} \in D_w, w : A(\overline{\vec{a}_1}) \supset B(\overline{\vec{a}_2}), v : A(\overline{\vec{a}_1}), \Gamma \Rightarrow v : B(\overline{\vec{a}_2}), \Delta}}{\overline{\mathcal{R}, \overline{\vec{a}} \in D_w, w : A(\overline{\vec{a}_1}) \supset B(\overline{\vec{a}_2}), \Gamma \Rightarrow w : A(\overline{\vec{a}_1}) \supset B(\overline{\vec{a}_2}), \Delta}} \text{ } (\supset_r) \\
 \Pi_1 &= \left\{ \frac{\overline{\mathcal{R}, w \leq v, \overline{\vec{a}} \in D_w, \overline{\vec{a}} \in D_v, v : A(\overline{\vec{a}_1}), \Gamma' \Rightarrow v : A(\overline{\vec{a}_1}), \Delta'}{\overline{\mathcal{R}, w \leq v, \overline{\vec{a}} \in D_w, \overline{\vec{a}} \in D_v, v : A(\overline{\vec{a}_1}), \Gamma' \Rightarrow v : A(\overline{\vec{a}_1}), \Delta'} \text{ IH} \right. \\
 \Pi_2 &= \left\{ \frac{\overline{\mathcal{R}, w \leq v, \overline{\vec{a}} \in D_w, \overline{\vec{a}} \in D_v, v : B(\overline{\vec{a}_2}), \Gamma' \Rightarrow v : B(\overline{\vec{a}_2}), \Delta'}{\overline{\mathcal{R}, w \leq v, \overline{\vec{a}} \in D_w, \overline{\vec{a}} \in D_v, v : B(\overline{\vec{a}_2}), \Gamma' \Rightarrow v : B(\overline{\vec{a}_2}), \Delta'} \text{ IH} \right. \\
 &\frac{\overline{\mathcal{R}, \vec{b} \in D_w, \overline{\vec{a}} \in D_w, w : A(\overline{\vec{a}})(\vec{b}/x), \Gamma \Rightarrow w : A(\overline{\vec{a}})(\vec{b}/x), \Delta'}{\overline{\mathcal{R}, \vec{b} \in D_w, \overline{\vec{a}} \in D_w, w : A(\overline{\vec{a}})(\vec{b}/x), \Gamma \Rightarrow w : \exists x A(\overline{\vec{a}}), \Delta}} \text{ IH} \text{ } (\exists_r) \\
 &\frac{\overline{\mathcal{R}, \vec{b} \in D_w, \overline{\vec{a}} \in D_w, w : A(\overline{\vec{a}})(\vec{b}/x), \Gamma \Rightarrow w : \exists x A(\overline{\vec{a}}), \Delta}}{\overline{\mathcal{R}, \overline{\vec{a}} \in D_w, w : \exists x A(\overline{\vec{a}}), \Gamma \Rightarrow w : \exists x A(\overline{\vec{a}}), \Delta}} \text{ } (\exists_l) \\
 &\frac{\overline{\mathcal{R}, w \leq v, \vec{b} \in D_u, \overline{\vec{a}} \in D_w, \overline{\vec{a}} \in D_v, v : A(\overline{\vec{a}})(\vec{b}/x), w : \forall x A(\overline{\vec{a}}), \Gamma \Rightarrow v : A(\overline{\vec{a}})(\vec{b}/x), \Delta}}{\overline{\mathcal{R}, w \leq v, \vec{b} \in D_u, \overline{\vec{a}} \in D_w, \overline{\vec{a}} \in D_v, v : \forall x A(\overline{\vec{a}}), \Gamma \Rightarrow v : A(\overline{\vec{a}})(\vec{b}/x), \Delta}} \text{ IH} \text{ } (\forall_l) \\
 &\frac{\overline{\mathcal{R}, w \leq v, \vec{b} \in D_u, \overline{\vec{a}} \in D_w, w : \forall x A(\overline{\vec{a}}), \Gamma \Rightarrow v : A(\overline{\vec{a}})(\vec{b}/x), \Delta}}{\overline{\mathcal{R}, w \leq v, \vec{b} \in D_u, \overline{\vec{a}} \in D_w, w : \forall x A(\overline{\vec{a}}), \Gamma \Rightarrow v : A(\overline{\vec{a}})(\vec{b}/x), \Delta}} \text{ (nd)} \times n \\
 &\frac{\overline{\mathcal{R}, w \leq v, \vec{b} \in D_u, \overline{\vec{a}} \in D_w, w : \forall x A(\overline{\vec{a}}), \Gamma \Rightarrow v : A(\overline{\vec{a}})(\vec{b}/x), \Delta}}{\overline{\mathcal{R}, \overline{\vec{a}} \in D_w, w : \forall x A(\overline{\vec{a}}), \Gamma \Rightarrow w : \forall x A(\overline{\vec{a}}), \Delta}} \text{ } (\forall_r)
 \end{aligned}$$

Claim (ii). We consider two sets of rules from LR in turn.

The $\{(lsub), (psub)\}$ rules. Hp-admissibility of $(lsub)$ and $(psub)$ are proven by induction on the height of the given derivation; both are similar to [6, Lem. 3].

The only non-trivial cases for the former are the (\supset_r) and (\forall_r) rules of the inductive step, and the non-trivial cases for the latter are the (\exists_l) , (\forall_r) , and (ihd) rules of the inductive step. In such cases, the side condition must be preserved if the rule is to be applied. Nevertheless, this can be ensured in the usual way (cf. [6, Lem. 5.1]) by invoking IH twice: first, IH replaces the eigenvariable of the (\supset_r) , (\exists_l) , (\forall_r) , or (ihd) rule with a fresh one, and second, the substitution of $(lsub)$ or $(psub)$ is performed, after which, the corresponding rule may be applied to derive the desired conclusion.

The $\{(wk), (ctr_{\mathcal{R}}), (ctr_l), (ctr_r)\}$ rules. The hp-admissibility of (wk) , $(ctr_{\mathcal{R}})$, (ctr_l) , and (ctr_r) is shown by induction on the height of the given derivation. Hp-admissibility of (wk) is relatively straightforward; the only non-trivial cases are the (\supset_r) , (\exists_l) , (\forall_r) , and (ihd) rules of the inductive step due to the eigenvariable condition. Such cases are resolved, however, by potentially applying hp-admissibility of $(lsub)$ or $(psub)$ (argued above), then IH, and last the corresponding rule. Hp-admissibility of $(ctr_{\mathcal{R}})$ is simple; any application of the rule to an initial sequent yields an initial sequent, and every case in the inductive step is resolved by applying IH followed by the corresponding rule. The (ctr_l) and (ctr_r) cases are also quite straightforward. The only non-trivial cases occur when a principal occurrence of an (\supset_r) , (\exists_l) , or (\forall_r) rule is contracted. In such cases, the inductive step is solved by invoking the hp-invertibility of each rule (property (iii) below), followed by a potential application of $(lsub)$ or $(psub)$, an application of IH, and finally, the hp-admissibility of $(ctr_{\mathcal{R}})$ if needed.

Claim (iii). The cases for the propositional rules are straightforward to check, so we omit them. By hp-admissibility of (wk) , we know that (\exists_r) , (\forall_l) , (ihd) , (nd) , and (cd) are hp-invertible (note that the proof of (wk) admissibility does not depend on the hp-invertibility of rules holding, so we may invoke it). We therefore only need to check that (\forall_r) and (\exists_l) are hp-invertible. We prove the claim by induction on the height of the given derivation of $\mathcal{R}, \Gamma \Rightarrow w : \forall xA, \Delta$ for the (\forall_r) rule; the proof for (\exists_l) is similar.

Base case. If the height of the derivation is 0, then $\mathcal{R}, w \leq v, a \in D_v, \Gamma \Rightarrow v : A(\underline{a}/x), \Delta$ is either an instance of (\perp_l) or (id_q) .

Inductive step. If the last rule applied in the derivation is (\wedge_l) , (\wedge_r) , (\vee_l) , (\vee_r) , (\supset_l) , (ref) , (tra) , (\exists_r) , (\forall_l) , (nd) , or (cd) , then the conclusion follows by applying IH followed by an application of the associated rule. If the last rule applied is (\supset_r) , (\exists_l) , (\forall_r) , or (ihd) (where the principal formula of (\forall_r) is in Δ), then we potentially apply hp-admissibility of $(lsub)$ or $(psub)$ (property (ii) above), followed by IH, and then an application of the corresponding rule. If the last rule applied is (\forall_r) , with $w : \forall xA$ the principal formula, then the premise of the inference gives the desired result.

Claim (iv). We prove the admissibility of (cut) by induction on the lexicographic ordering of tuples $(|A|, h_1, h_2)$, where $|A|$ is the complexity of the cut formula A , h_1 is the height of the left premise of (cut) , and h_2 is the height of the right premise of (cut) . We only consider the cases where the cut formula is principal in both premises of (cut) ; the other cases where the cut formula is not principal in one premise, or in both premises, are relatively straightforward to verify, though the large number of cases makes the proof tedious. By the cut admissibility theorem for G3Int [6, Thm. 5.6], we need only verify the cases where the cut formula is of the form $\forall xB$ or $\exists xB$. We show each case in turn below:

$$\frac{\Pi_1 \quad \Pi_2}{\mathcal{R}, w \leq v, \underline{a} \in D_v, \Gamma \Rightarrow \Delta} (cut)$$

$$\Pi_1 = \left\{ \frac{\mathcal{R}, w \leq v, w \leq u, \underline{a} \in D_v, \underline{b} \in D_u, \Gamma \Rightarrow \Delta, u : B(\underline{b}/x)}{\mathcal{R}, w \leq v, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \forall xB} (\forall_r) \right.$$

$$\Pi_2 = \left\{ \frac{\mathcal{R}, w \leq v, \underline{a} \in D_v, v : B(\underline{a}/x), w : \forall xB, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq v, \underline{a} \in D_v, w : \forall xB, \Gamma \Rightarrow \Delta} (\forall_i) \right.$$

The case is resolved as follows:

$$\frac{\Pi'_1 \quad \Pi'_2}{\mathcal{R}, w \leq v, \underline{a} \in D_v, \Gamma \Rightarrow \Delta} (cut)$$

$$\Pi'_1 = \left\{ \frac{\mathcal{R}, w \leq u, \underline{b} \in D_u, \Gamma \Rightarrow \Delta, u : B(\underline{b}/x)}{\mathcal{R}, w \leq v, \underline{b} \in D_v, \Gamma \Rightarrow \Delta, v : B(\underline{b}/x)} (lsub) \right.$$

$$\left. \frac{\mathcal{R}, w \leq v, \underline{b} \in D_v, \Gamma \Rightarrow \Delta, v : B(\underline{b}/x)}{\mathcal{R}, w \leq v, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, v : B(\underline{a}/x)} (psub) \right.$$

$$\Pi'_2 = \left\{ \frac{\mathcal{R}, w \leq v, w \leq u, \underline{a} \in D_v, \underline{b} \in D_u, \Gamma \Rightarrow \Delta, u : B(\underline{b}/x)}{\mathcal{R}, w \leq v, w \leq u, \underline{a} \in D_v, \underline{b} \in D_u, v : B(\underline{a}/x), \Gamma \Rightarrow \Delta, u : B(\underline{b}/x)} (wk) \right.$$

$$\left. \frac{\mathcal{R}, w \leq v, \underline{a} \in D_v, v : B(\underline{a}/x), \Gamma \Rightarrow \Delta, w : \forall xB}{\mathcal{R}, w \leq v, \underline{a} \in D_v, v : B(\underline{a}/x), \Gamma \Rightarrow \Delta} \Lambda (cut) \right.$$

$$\Lambda = \mathcal{R}, w \leq v, \underline{a} \in D_v, v : B(\underline{a}/x), w : \forall xB, \Gamma \Rightarrow \Delta$$

Observe that the (cut) in Π'_2 has a height h_2 that is one less than the original (cut) , and the second (cut) is on a formula $B(\underline{a}/x)$ that is of less complexity than the original (cut) . Let us examine the $\exists xB$ case.

$$\frac{\frac{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta, w : B(\underline{a}/x), w : \exists xB}{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta, w : \exists xB} (\exists_r) \quad \frac{\mathcal{R}, \underline{a} \in D_w, \underline{b} \in D_w, w : B(\underline{b}/x), \Gamma \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_w, w : \exists xB, \Gamma \Rightarrow \Delta} (\exists_l)}{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta} (cut)$$

The case is resolved as follows:

$$\frac{\Pi'_1 \quad \Pi'_2}{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta} (cut)$$

$$\Pi'_1 = \left\{ \begin{array}{c} \frac{\frac{\mathcal{R}, \underline{a} \in D_w, \underline{b} \in D_w, w : B(\underline{b}/x), \Gamma \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_w, \underline{b} \in D_w, w : B(\underline{b}/x), \Gamma \Rightarrow w : B(\underline{a}/x), \Delta} \quad (wk)}{\frac{\mathcal{R}, \underline{a} \in D_w, w : \exists x B, \Gamma \Rightarrow w : B(\underline{a}/x), \Delta}{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow w : B(\underline{a}/x), \Delta}} \quad (\exists_l)} \\ A \\ \frac{}{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow w : B(\underline{a}/x), \Delta} \quad (cut) \end{array} \right.$$

$$\Lambda = \mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta, w : B(\underline{a}/x), w : \exists x B$$

$$\Pi'_2 = \left\{ \begin{array}{c} \frac{\mathcal{R}, \underline{a} \in D_w, \underline{b} \in D_w, w : B(\underline{b}/x), \Gamma \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_w, \underline{a} \in D_w, w : B(\underline{a}/x), \Gamma \Rightarrow \Delta} \quad (psub) \\ \frac{}{\mathcal{R}, \underline{a} \in D_w, w : B(\underline{a}/x), \Gamma \Rightarrow \Delta} \quad (ctr_{\mathcal{R}}) \end{array} \right.$$

Observe that the (*cut*) in the Π'_1 inference has a height h_1 that is one less than the original (*cut*), and the second (*cut*) is on a formula of smaller complexity.

Claim (v). Soundness is straightforward and is shown by interpreting labelled sequents on IntQC -models (or IntQ -models in the case of G3IntQ) and proving that validity is preserved from premise to conclusion. By [6], we know that G3Int is complete relative to Int , so we need only show that G3IntQC can derive quantifier axioms and simulate the inference rules of the axiomatization for IntQC (see [10, Ch. 2.6]); we assume that none of the quantifiers occurring in formulae are vacuous, and note that the vacuous cases can be derived in a similar fashion. We say that a formula $A(\vec{a})$ is derivable in G3IntQC if and only if $\vec{a} \in D_w \Rightarrow w : A(\vec{a})$ is derivable in G3IntQC . Also, we use the notation $(nd) \times n$ to indicate that the rule (*nd*) was applied some number of times to shift a group of domain atoms forward (which will be made clear by the context).

$$\frac{\frac{\frac{w \leq u, u \leq u, \vec{a} \in D_w, \underline{a} \in D_w, \underline{a} \in D_u, \vec{a} \in D_u, u : \forall x A(\vec{a}, x), u : A(\vec{a}, \underline{a}) \Rightarrow u : A(\vec{a}, \underline{a})}{w \leq u, u \leq u, \vec{a} \in D_w, \underline{a} \in D_w, \underline{a} \in D_u, \vec{a} \in D_u, u : \forall x A(\vec{a}, x) \Rightarrow u : A(\vec{a}, \underline{a})} \quad (\forall_l)}{\frac{w \leq u, u \leq u, \vec{a} \in D_w, \underline{a} \in D_w, u : \forall x A(\vec{a}, x) \Rightarrow u : A(\vec{a}, \underline{a})}{w \leq u, \vec{a} \in D_w, \underline{a} \in D_w, u : \forall x A(\vec{a}, x) \Rightarrow u : A(\vec{a}, \underline{a})} \quad (ref)}{\frac{\vec{a} \in D_w, \underline{a} \in D_w \Rightarrow w : \forall x A(\vec{a}, x) \supset A(\vec{a}, \underline{a})}{\vec{a} \in D_w, \underline{a} \in D_w \Rightarrow w : \forall x A(\vec{a}, x) \supset A(\vec{a}, \underline{a})} \quad (\supset_r)} \quad (nd) \times n \quad \text{Prop. (i)}$$

$$\frac{\frac{\frac{w \leq u, \vec{a} \in D_w, \underline{a} \in D_w, \vec{a} \in D_u, \underline{a} \in D_u, u : A(\vec{a}, \underline{a}) \Rightarrow u : A(\vec{a}, \underline{a}), u : \exists x A(\vec{a}, x)}{\frac{w \leq u, \vec{a} \in D_w, \underline{a} \in D_w, \vec{a} \in D_u, \underline{a} \in D_u, u : A(\vec{a}, \underline{a}) \Rightarrow u : \exists x A(\vec{a}, x)}{\frac{w \leq u, \vec{a} \in D_w, \underline{a} \in D_w, u : A(\vec{a}, \underline{a}) \Rightarrow u : \exists x A(\vec{a}, x)}{\vec{a} \in D_w, \underline{a} \in D_w \Rightarrow w : A(\vec{a}, \underline{a}) \supset \exists x A(\vec{a}, x)} \quad (\supset_r)} \quad (nd) \times n}{\vec{a} \in D_w, \underline{a} \in D_w \Rightarrow w : A(\vec{a}, \underline{a}) \supset \exists x A(\vec{a}, x)} \quad (\supset_r)} \quad (\exists_r) \quad \text{Prop. (i)}$$

$$\frac{\frac{\frac{\frac{\Pi_1 \quad \Pi_2}{\mathcal{R}, v \leq z, z \leq z, z : B(\vec{b}) \supset A(\vec{a}, \underline{a}), v : \forall x (B(\vec{b}) \supset A(\vec{a}, x)), u : B(\vec{b}) \Rightarrow z : A(\vec{a}, \underline{a})} \quad (ref)}{\frac{\mathcal{R}, v \leq z, z : B(\vec{b}) \supset A(\vec{a}, \underline{a}), v : \forall x (B(\vec{b}) \supset A(\vec{a}, x)), u : B(\vec{b}) \Rightarrow z : A(\vec{a}, \underline{a})}{\mathcal{R}, v \leq z, v : \forall x (B(\vec{b}) \supset A(\vec{a}, x)), u : B(\vec{b}) \Rightarrow z : A(\vec{a}, \underline{a})} \quad (\forall_l)}{\mathcal{R}, v : \forall x (B(\vec{b}) \supset A(\vec{a}, x)), u : B(\vec{b}) \Rightarrow z : A(\vec{a}, \underline{a})} \quad (tra)}{\frac{\mathcal{R}, v : \forall x (B(\vec{b}) \supset A(\vec{a}, x)), u : B(\vec{b}) \Rightarrow z : A(\vec{a}, \underline{a})}{w \leq v, v \leq u, \vec{a} \in D_w, \vec{b} \in D_w, v : \forall x (B(\vec{b}) \supset A(\vec{a}, x)), u : B(\vec{b}) \Rightarrow u : \forall x A(\vec{a}, x)} \quad (\forall_r)}{\frac{w \leq v, \vec{a} \in D_w, \vec{b} \in D_w, v : \forall x (B(\vec{b}) \supset A(\vec{a}, x)) \Rightarrow v : B(\vec{b}) \supset \forall x A(\vec{a}, x)}{\vec{a} \in D_w, \vec{b} \in D_w \Rightarrow w : \forall x (B(\vec{b}) \supset A(\vec{a}, x)) \supset (B(\vec{b}) \supset \forall x A(\vec{a}, x))} \quad (\supset_r)} \quad (\supset_r)$$

To save space, we let $\mathcal{R} := w \leq v, v \leq u, u \leq z, \vec{a} \in D_w, \vec{b} \in D_w, \underline{a} \in D_z$ and $\Gamma := z : B(\vec{b}) \supset A(\vec{a}, \underline{a}), v : \forall x(B(\vec{b}) \supset A(\vec{a}, x))$. The proofs Π_1 and Π_2 are as follows:

$$\Pi_1 = \left\{ \frac{\text{Prop. (i)}}{\frac{\mathcal{R}, v \leq z, z \leq z, \vec{b} \in D_v, \vec{b} \in D_u, \Gamma, u : B(\vec{b}) \Rightarrow z : B(\vec{b}), z : A(\vec{a}, \underline{a})}{\mathcal{R}, v \leq z, z \leq z, \Gamma, u : B(\vec{b}) \Rightarrow z : B(\vec{b}), z : A(\vec{a}, \underline{a})}} (nd) \times k_1} \right.$$

$$\Pi_2 = \left\{ \frac{\text{Prop. (i)}}{\frac{\mathcal{R}, v \leq z, z \leq z, \vec{a} \in D_v, \vec{a} \in D_z, \Gamma, u : B(\vec{b}), z : A(\vec{a}, \underline{a}) \Rightarrow z : A(\vec{a}, \underline{a})}{\mathcal{R}, v \leq z, z \leq z, \Gamma, u : B(\vec{b}), z : A(\vec{a}, \underline{a}) \Rightarrow z : A(\vec{a}, \underline{a})}} (nd) \times k_2} \right.$$

The proof of the axiom $\forall x(A(x) \supset B) \supset (\exists x A(x) \supset B)$ is similar to the previous proof. The generalization rule is simulated as shown below:

$$\frac{\frac{\frac{\vec{a} \in D_w, \underline{a} \in D_w \Rightarrow w : A(\vec{a}, \underline{a})}{u \leq w, \vec{a} \in D_u, \vec{a} \in D_w, \underline{a} \in D_w \Rightarrow w : A(\vec{a}, \underline{a})} (wk)}{u \leq w, \vec{a} \in D_u, \underline{a} \in D_w \Rightarrow w : A(\vec{a}, \underline{a})} (\forall_r)}{\frac{\vec{a} \in D_u \Rightarrow u : \forall x A(\vec{a}, x)}{\vec{a} \in D_w \Rightarrow w : \forall x A(\vec{a}, x)} (lsub)}} (nd) \times n$$

$$\frac{\frac{\frac{\frac{\Pi_1 \quad \Pi_2}{\mathcal{R}, v \leq v, \underline{a} \in D_v, v : A(\vec{a}, \underline{a}) \vee B(\vec{b}), v : \forall x(A(\vec{a}, x) \vee B(\vec{b})) \Rightarrow u : A(\vec{a}, \underline{a}), v : B(\vec{b})} (\forall_l)}{\mathcal{R}, v \leq v, \underline{a} \in D_v, v : \forall x(A(\vec{a}, x) \vee B(\vec{b})) \Rightarrow u : A(\vec{a}, \underline{a}), v : B(\vec{b})} (\forall_l)} (cd)}{\mathcal{R}, v \leq v, v : \forall x(A(\vec{a}, x) \vee B(\vec{b})) \Rightarrow u : A(\vec{a}, \underline{a}), v : B(\vec{b})} (ref)}} (\forall_r)$$

$$\frac{\frac{\frac{\mathcal{R}, v : \forall x(A(\vec{a}, x) \vee B(\vec{b})) \Rightarrow u : A(\vec{a}, \underline{a}), v : B(\vec{b})}{w \leq v, \vec{a} \in D_w, \vec{b} \in D_w, v : \forall x(A(\vec{a}, x) \vee B(\vec{b})) \Rightarrow v : \forall x A(\vec{a}, x), v : B(\vec{b})} (\forall_r)}{w \leq v, \vec{a} \in D_w, \vec{b} \in D_w, v : \forall x(A(\vec{a}, x) \vee B(\vec{b})) \Rightarrow v : \forall x A(\vec{a}, x) \vee B(\vec{b})} (\forall_r)}{\vec{a} \in D_w, \vec{b} \in D_w \Rightarrow w : \forall x(A(\vec{a}, x) \vee B(\vec{b})) \supset \forall x A(\vec{a}, x) \vee B(\vec{b})} (\supset_r)}$$

to save space, we let $\mathcal{R} := w \leq v, v \leq u, \vec{a} \in D_w, \vec{b} \in D_w, \underline{a} \in D_u$. The proofs Π_1 and Π_2 are as follows (resp.).

$$\frac{\text{Prop. (i)}}{\frac{\mathcal{R}, v \leq v, \vec{a} \in D_v, \underline{a} \in D_v, v : A(\vec{a}, \underline{a}), v : \forall x(A(\vec{a}, x) \vee B(\vec{b})) \Rightarrow u : A(\vec{a}, \underline{a}), v : B(\vec{b})}{\mathcal{R}, v \leq v, \underline{a} \in D_v, v : A(\vec{a}, \underline{a}), v : \forall x(A(\vec{a}, x) \vee B(\vec{b})) \Rightarrow u : A(\vec{a}, \underline{a}), v : B(\vec{b})}} (nd) \times k_1}$$

$$\frac{\text{Prop. (i)}}{\frac{\mathcal{R}, v \leq v, \underline{a} \in D_v, \vec{b} \in D_v, v : B(\vec{b}), v : \forall x(A(\vec{a}, x) \vee B(\vec{b})) \Rightarrow u : A(\vec{a}, \underline{a}), v : B(\vec{b})}{\mathcal{R}, v \leq v, \underline{a} \in D_v, v : B(\vec{b}), v : \forall x(A(\vec{a}, x) \vee B(\vec{b})) \Rightarrow u : A(\vec{a}, \underline{a}), v : B(\vec{b})}} (nd) \times k_2}$$

To show that modus ponens can be simulated, we let \vec{a} be all parameters occurring in A , \vec{b} be all parameters occurring in B , and let \vec{c} consist of all parameters occurring in B , but not A . The last inference consists of a sequence of k (*ihd*) applications that delete all domains atoms containing parameters from A , but not B .

$$\frac{\frac{\frac{\vec{a} \in D_w, \vec{c} \in D_w \Rightarrow w : A(\vec{a}) \supset B(\vec{b})}{w \leq u, \vec{a} \in D_w, \vec{c} \in D_w, u : A(\vec{a}) \Rightarrow u : B(\vec{b})} \text{ (Prop. (iii))}}{w \leq w, \vec{a} \in D_w, \vec{c} \in D_w, w : A(\vec{a}) \Rightarrow w : B(\vec{b})} \text{ (ref)}}{\frac{\vec{a} \in D_w, \vec{c} \in D_w \Rightarrow w : A(\vec{a})}{\vec{a} \in D_w, \vec{c} \in D_w \Rightarrow w : A(\vec{a})} \text{ (wk)} \quad \frac{\vec{a} \in D_w, \vec{c} \in D_w, w : A(\vec{a}) \Rightarrow w : B(\vec{b})}{\vec{a} \in D_w, \vec{c} \in D_w, w : A(\vec{a}) \Rightarrow w : B(\vec{b})} \text{ (cut)}}{\frac{\vec{a} \in D_w, \vec{c} \in D_w \Rightarrow w : B(\vec{b})}{\vec{b} \in D_w \Rightarrow w : B(\vec{b})} \text{ (ihd)} \times k}$$

□

Lemma 4. *Let $\text{G3X}^* \in \{\text{G3IntQ}^*, \text{G3IntQC}^*\}$. The calculi G3Int^* , G3IntQ^* , and G3IntQC^* have the following properties:*

- (i) (a) For all $A \in \mathcal{L}$, $\vdash_{\text{G3Int}^*} \mathcal{R}, w \leq v, w : A, \Gamma \Rightarrow v : A, \Delta$;
- (b) For all $A \in \mathcal{L}$, $\vdash_{\text{G3Int}^*} \mathcal{R}, w : A, \Gamma \Rightarrow \Delta, w : A$;
- (c) For all $A \in \mathcal{Q}$, $\vdash_{\text{G3X}^*} \mathcal{R}, w \leq v, \vec{a} \in D_w, w : A(\vec{a}), \Gamma \Rightarrow v : A(\vec{a}), \Delta$;
- (d) For all $A \in \mathcal{Q}$, $\vdash_{\text{G3X}^*} \mathcal{R}, \vec{a} \in D_w, w : A(\vec{a}), \Gamma \Rightarrow \Delta, w : A(\vec{a})$;
- (ii) The rules $\{(lsub), (psub), (wk), (ctr_{\mathcal{R}}), (ctr_r)\}$ are hp-admissible;
- (iii) With the exception of $\{(\wedge_l), (\exists_l)\}$, all rules are hp-invertible;
- (iv) The rules $\{(\wedge_l), (\exists_l)\}$ are invertible;
- (v) The rule (ctr_l) is admissible.

Proof. We show that the results hold when all rules from G3Int^* , G3IntQ^* , and G3IntQC^* are considered.

Claim (i). Similar to Thm. 3-(i).

Claim (ii). By induction on the height of the given derivation—similar to the proofs given for Thm. 3-(ii). Note that the hp-admissibility of (ctr_r) invokes property (iii) below.

Claim (iii). Hp-invertibility of (\supset_l) , (\supset_l^*) , (\neg_l) , (ref) , (tra) , $(lift)$, (ihd) , (\exists_r) , (\exists_r^n) , (\exists_r^{in}) , (\exists_r^c) , (\exists_r^{ic}) , (\forall_l) , (\forall_l^n) , (\forall_l^{in}) , (\forall_l^c) , (\forall_l^{ic}) , (nd) , (cd) follow from hp-admissibility of (wk) (property (ii) above). Hence, we need only prove invertibility of the (\supset_r) , (\wedge_r) , (\vee_l) , (\vee_r) , (\neg_r) , (\forall_r) , and (\forall_r^c) rules. The result is shown by induction on the height of the given derivation and is similar to Thm. 3-(iii).

Claim (iv). The addition of $(lift)$ and (nd) to our calculus breaks the *height preserving* invertibility of (\wedge_l) and (\exists_l) . Nevertheless, it is worthwhile to note that the rule $(lift')$ (shown below) ought to allow for hp-invertibility of (\wedge_l) in the propositional calculus, and (nd') (shown below) ought to allow for the hp-invertibility of (\wedge_l) and (\exists_l) in the first-order calculi (which would have the effect that (ctr_l) is hp-admissible in both calculi) while retaining the soundness and cut-free completeness of each system. The notation $v : \Gamma'$ is used to represent multisets of formulae labelled with v .

$$\frac{\mathcal{R}, w \leq u, \Gamma, w : \Gamma', u : \Gamma' \Rightarrow \Delta}{\mathcal{R}, w \leq u, \Gamma, w : \Gamma' \Rightarrow \Delta} \text{ (lift')} \quad \frac{\mathcal{R}, w \leq u, \vec{a} \in D_w, \vec{a} \in D_u, \Gamma, w : \Gamma', u : \Gamma' \Rightarrow \Delta}{\mathcal{R}, w \leq u, \vec{a} \in D_w, \Gamma, w : \Gamma' \Rightarrow \Delta} \text{ (nd')}$$

Despite this shortcoming, the rules are still invertible. To show this, we prove the following two claims by induction on the height of the given derivation.

- (a) If $\mathcal{R}, w_1 : A \wedge B, \dots, w_n : A \wedge B, \Gamma \Rightarrow \Delta$ is provable, then so is the sequent $\mathcal{R}, w_1 : A, w_1 : B, \dots, w_n : A, w_n : B, \Gamma \Rightarrow \Delta$.
- (b) If $\mathcal{R}, w_1 : \exists x A, \dots, w_n : \exists x A, \Gamma \Rightarrow \Delta$ is provable, then so is the sequent $\mathcal{R}, \underline{a}_1 \in D_{w_1}, \dots, \underline{a}_n \in D_{w_n}, w_1 : A(\underline{a}_1/x), \dots, w_n : A(\underline{a}_n/x), \Gamma \Rightarrow \Delta$.

Claim (a). The base case is trivial, so we move on to the inductive step.

Inductive step. Excluding the case of (*lift*), the result follows by applying IH followed by the relevant rule, or in the case where one of the conjunction formulae is principal in a (\wedge_r) inference, apply IH to the premise (if needed) to obtain the desired result. If none of the conjunctions are active in a (*lift*) inference, then apply IH followed by an application of (*lift*). If one of the conjunctions is principal in an application of (*lift*) (as shown below top), then resolve the case as shown below bottom; we assume w.l.o.g. that the label of the principal formula is w_1 .

$$\frac{\mathcal{R}, u \leq w_1, u : A \wedge B, w_1 : A \wedge B, \dots, w_n : A \wedge B, \Gamma \Rightarrow \Delta}{\mathcal{R}, u \leq w_1, u : A \wedge B, \dots, w_n : A \wedge B, \Gamma \Rightarrow \Delta} \text{ (lift)}$$

$$\frac{\frac{\mathcal{R}, u \leq w_1, u : A, u : B, w_1 : A, w_1 : B, \dots, w_n : A, w_n : B, \Gamma \Rightarrow \Delta}{\mathcal{R}, u \leq w_1, u : A, u : B, w_1 : B, \dots, w_n : A, w_n : B, \Gamma \Rightarrow \Delta} \text{ IH}}{\mathcal{R}, u \leq w_1, u : A, u : B, \dots, w_n : A, w_n : B, \Gamma \Rightarrow \Delta} \text{ (lift)}$$

Notice that the two applications of (*lift*) needed to derive the desired conclusion break the hp-invertibility of the (\wedge_l) rule.

Claim (b). The base case is trivial, so we move on to the inductive step.

Inductive step. All cases, with the exception of the one given below top (where one of our existential formulae is principal in an application of (*lift*)), are either resolved by applying IH followed by the corresponding rule, or if one of the existential formulae is principal in an (\exists_r) inference, then apply IH to the premise (if needed) to obtain the desired result. The non-trivial case given below top is resolved as shown below bottom; we assume w.l.o.g. that the label of the principal formula is w_1 .

$$\frac{\mathcal{R}, u \leq w_1, u : \exists x A, w_1 : \exists x A, \dots, w_n : \exists x A, \Gamma \Rightarrow \Delta}{\mathcal{R}, u \leq w_1, u : \exists x A, \dots, w_n : \exists x A, \Gamma \Rightarrow \Delta} \text{ (lift)}$$

$$\frac{\frac{\mathcal{R}, u \leq w_1, \underline{a} \in D_u, \underline{a}_1 \in D_{w_1}, \dots, \underline{a}_n \in D_{w_n}, u : A(\underline{a}/x), w_1 : A(\underline{a}_1/x), \dots, w_n : A(\underline{a}_n/x), \Gamma \Rightarrow \Delta}{\mathcal{R}, u \leq w_1, \underline{a} \in D_u, \underline{a} \in D_{w_1}, \dots, \underline{a}_n \in D_{w_n}, u : A(\underline{a}/x), w_1 : A(\underline{a}/x), \dots, w_n : A(\underline{a}_n/x), \Gamma \Rightarrow \Delta} \text{ IH (psub)}}{\frac{\mathcal{R}, u \leq w_1, \underline{a} \in D_u, \underline{a} \in D_{w_1}, \dots, \underline{a}_n \in D_{w_n}, u : A(\underline{a}/x), \dots, w_n : A(\underline{a}_n/x), \Gamma \Rightarrow \Delta}{\mathcal{R}, u \leq w_1, \underline{a} \in D_u, \dots, \underline{a}_n \in D_{w_n}, u : A(\underline{a}/x), \dots, w_n : A(\underline{a}_n/x), \Gamma \Rightarrow \Delta} \text{ (lift) (nd)}}$$

Observe that the use of (*lift*) and (*nd*) breaks height-preserving invertibility of the rule.

Propositions (a) and (b) imply the invertibility of (\wedge_l) and (\exists_l).

Claim (v). By induction on pairs of the form $(|A|, h)$ where $|A|$ is the complexity of the contraction formula A and h is the height of the derivation. The proof makes use of properties (ii)–(iv). \square

Lemma 10. *The (ref) and (tra) rules are admissible in the calculus G3IntQ*.*

Proof. The (tra) elimination cases are given below with each non-trivial case being followed by a proof of how it is resolved. In the (id_q^n) case, We let $\mathcal{R}' := \mathcal{R}, \underline{a}_1 \in D_{v_1}, \dots, \underline{a}_n \in D_{v_n}$.

$$\frac{\frac{\mathcal{R}', u \leq z, z \leq z', u \leq z', \Gamma, w : p(\vec{a}) \Rightarrow w : p(\vec{a}), \Delta}{\mathcal{R}', u \leq z, z \leq z', \Gamma, w : p(\vec{a}) \Rightarrow w : p(\vec{a}), \Delta} (id_q^n)}{\mathcal{R}', u \leq z, z \leq z', \Gamma, w : p(\vec{a}) \Rightarrow w : p(\vec{a}), \Delta} (tra)$$

$$\frac{\mathcal{R}', u \leq z, z \leq z', \Gamma, w : p(\vec{a}) \Rightarrow w : p(\vec{a}), \Delta}{\mathcal{R}', u \leq z, z \leq z', \Gamma, w : p(\vec{a}) \Rightarrow w : p(\vec{a}), \Delta} (id_q^n)$$

$$\frac{\frac{\mathcal{R}, u \leq z, z \leq z', u \leq z', \underline{a} \in D_v, \Gamma, w : A(\underline{a}/x), w : \forall xA \Rightarrow \Delta}{\mathcal{R}, u \leq z, z \leq z', u \leq z', \underline{a} \in D_v, \Gamma, w : \forall xA \Rightarrow \Delta} (\forall_l^n)}{\mathcal{R}, u \leq z, z \leq z', \underline{a} \in D_v, \Gamma, w : \forall xA \Rightarrow \Delta} (tra)$$

$$\frac{\frac{\mathcal{R}, u \leq z, z \leq z', \underline{a} \in D_v, w : A(\underline{a}/x), w : \forall xA, \Gamma \Rightarrow \Delta}{\mathcal{R}, u \leq z, z \leq z', \underline{a} \in D_v, w : \forall xA, \Gamma \Rightarrow \Delta} \text{IH}}{\mathcal{R}, u \leq z, z \leq z', \underline{a} \in D_v, w : \forall xA, \Gamma \Rightarrow \Delta} (\forall_l^{in})$$

$$\frac{\frac{\mathcal{R}, u \leq z, z \leq z', u \leq z', \underline{a} \in D_v, \Gamma, w : A(\underline{a}/x), w : \forall xA \Rightarrow \Delta}{\mathcal{R}, u \leq z, z \leq z', u \leq z', \Gamma, w : \forall xA \Rightarrow \Delta} (\forall_l^{in})}{\mathcal{R}, u \leq z, z \leq z', \Gamma, w : \forall xA \Rightarrow \Delta} (tra)$$

$$\frac{\frac{\mathcal{R}, u \leq z, z \leq z', \underline{a} \in D_v, w : A(\underline{a}/x), w : \forall xA, \Gamma \Rightarrow \Delta}{\mathcal{R}, u \leq z, z \leq z', \underline{a} \in D_v, w : \forall xA, \Gamma \Rightarrow \Delta} \text{IH}}{\mathcal{R}, u \leq z, z \leq z', w : \forall xA, \Gamma \Rightarrow \Delta} (\forall_l^{in})$$

$$\frac{\frac{\mathcal{R}, u \leq z, z \leq z', u \leq z', \underline{a} \in D_v, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta}{\mathcal{R}, u \leq z, z \leq z', u \leq z', \underline{a} \in D_v, \Gamma \Rightarrow w : \exists xA, \Delta} (\exists_r^n)}{\mathcal{R}, u \leq z, z \leq z', \underline{a} \in D_v, \Gamma \Rightarrow w : \exists xA, \Delta} (tra)$$

$$\frac{\frac{\mathcal{R}, u \leq z, z \leq z', \underline{a} \in D_v, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta}{\mathcal{R}, u \leq z, z \leq z', \underline{a} \in D_v, \Gamma \Rightarrow w : \exists xA, \Delta} \text{IH}}{\mathcal{R}, u \leq z, z \leq z', \underline{a} \in D_v, \Gamma \Rightarrow w : \exists xA, \Delta} (\exists_r^n)$$

$$\frac{\frac{\mathcal{R}, u \leq z, z \leq z', u \leq z', \Gamma, \underline{a} \in D_v \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta}{\mathcal{R}, u \leq z, z \leq z', u \leq z', \Gamma \Rightarrow w : \exists xA, \Delta} (\exists_r^{in})}{\mathcal{R}, u \leq z, z \leq z', \Gamma \Rightarrow w : \exists xA, \Delta} (tra)$$

$$\frac{\frac{\mathcal{R}, u \leq z, z \leq z', \underline{a} \in D_v, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta}{\mathcal{R}, u \leq z, z \leq z', \Gamma \Rightarrow w : \exists xA, \Delta} \text{IH}}{\mathcal{R}, u \leq z, z \leq z', \Gamma \Rightarrow w : \exists xA, \Delta} (\exists_r^{in})$$

$$\frac{\frac{\mathcal{R}, w \leq u, u \leq v, w \leq v, \underline{a} \in D_w, \underline{a} \in D_v, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq u, u \leq v, w \leq v, \underline{a} \in D_w, \Gamma \Rightarrow \Delta} (nd)}{\mathcal{R}, w \leq u, u \leq v, \underline{a} \in D_w, \Gamma \Rightarrow \Delta} (tra)$$

$$\begin{array}{c}
\frac{\mathcal{R}, w \leq u, u \leq v, w \leq v, \underline{a} \in D_w, \underline{a} \in D_v, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq u, u \leq v, w \leq v, \underline{a} \in D_w, \underline{a} \in D_u, \underline{a} \in D_v, \Gamma \Rightarrow \Delta} \text{ (wk)} \\
\frac{\mathcal{R}, w \leq u, u \leq v, \underline{a} \in D_w, \underline{a} \in D_u, \underline{a} \in D_v, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq u, u \leq v, \underline{a} \in D_w, \underline{a} \in D_u, \underline{a} \in D_v, \Gamma \Rightarrow \Delta} \text{ IH} \\
\frac{\mathcal{R}, w \leq u, u \leq v, \underline{a} \in D_w, \underline{a} \in D_u, \underline{a} \in D_v, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq u, u \leq v, \underline{a} \in D_w, \underline{a} \in D_u, \underline{a} \in D_v, \Gamma \Rightarrow \Delta} \text{ (nd)}
\end{array}$$

As with the admissibility proof for (ref) , the side condition of the (id_q^n) rule still holds after $u \leq z'$ is deleted, and the side condition of the (\exists_r^n) , (\exists_r^{in}) , (\forall_l^n) , and (\forall_l^{in}) cases continues to hold after the invocation of IH. If a directed path traverses the relational atom $u \leq z'$ in the (id_q^n) case, a directed path still exists by replacing $u \leq z'$ with $u \leq z, z \leq z'$ in the given path. In the other cases, if the directed path traverses the $u \leq z'$ relational atom, then the relational atoms $u \leq z, z \leq z'$ are still present in the sequent after invoking IH, ensuring that a path between v and w continues to exist. \square

Lemma 15. *The (ref) and (tra) rules are admissible in the calculus G3IntQC*.*

Proof. We prove the result by induction on the height of the given derivation. We assume w.l.o.g. that the last inference in the given derivation is an instance of (ref) or (tra) , and that this is the only instance of the rule in the given derivation. The general result follows by successively eliminating topmost occurrences of (ref) or (tra) from a given derivation until it is free of such inferences. By Lem. 5 and Lem. 16, we need only consider the non-trivial (id_q^c) , (\exists_r^c) , (\exists_r^{ic}) , (\forall_l^c) , (\forall_l^{ic}) , (nd) , and (cd) cases.

Base case. We let $\mathcal{R}' := \mathcal{R}, \underline{a}_1 \in D_{v_1}, \dots, \underline{a}_n \in D_{v_n}$ in the (id_q^n) cases below. In both the (ref) and (tra) cases, the side condition that there is an undirected path from v_i to w , for each $i \in \{1, \dots, n\}$, holds in the end sequent. If none of the undirected paths from v_i to w contain $u \leq u$ ($u \leq z'$), then the paths are present in \mathcal{R}' ($\mathcal{R}', u \leq z, z \leq z'$, resp.), and if an undirected path from v_i to w contains $u \leq u$ ($u \leq z'$, resp.), then deleting each occurrence of $u \leq u$ from the undirected path (replacing each occurrence of $u \leq z'$ with $u \leq z, z \leq z'$, resp.) gives a new path from v_i to w .

$$\begin{array}{c}
\frac{\mathcal{R}', u \leq u, \Gamma, w : p(\vec{a}) \Rightarrow w : p(\vec{a}), \Delta}{\mathcal{R}', \Gamma, w : p(\vec{a}) \Rightarrow w : p(\vec{a}), \Delta} \text{ (ref)} \frac{\mathcal{R}', \Gamma, w : p(\vec{a}) \Rightarrow w : p(\vec{a}), \Delta}{\mathcal{R}', \Gamma, w : p(\vec{a}) \Rightarrow w : p(\vec{a}), \Delta} \text{ (id}_q^n) \\
\frac{\mathcal{R}', u \leq z, z \leq z', u \leq z', \Gamma, w : p(\vec{a}) \Rightarrow w : p(\vec{a}), \Delta}{\mathcal{R}', u \leq z, z \leq z', \Gamma, w : p(\vec{a}) \Rightarrow w : p(\vec{a}), \Delta} \text{ (tra)} \text{ (id}_q^c) \\
\frac{\mathcal{R}', u \leq z, z \leq z', \Gamma, w : p(\vec{a}) \Rightarrow w : p(\vec{a}), \Delta}{\mathcal{R}', u \leq z, z \leq z', \Gamma, w : p(\vec{a}) \Rightarrow w : p(\vec{a}), \Delta} \text{ (id}_q^c)
\end{array}$$

Inductive step. For the inductive step, we consider the non-trivial (\exists_r^c) , (\exists_r^{ic}) , (\forall_l^c) , (\forall_l^{ic}) , (nd) , and (cd) cases. We first show how to permute (ref) above each of these rules, and then focus on the permutation of (tra) above each considered rule.

$$\begin{array}{c}
 \frac{\mathcal{R}, u \leq u, \underline{a} \in D_v, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta}{\frac{\mathcal{R}, u \leq u, \underline{a} \in D_v, \Gamma \Rightarrow w : \exists xA, \Delta}{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow w : \exists xA, \Delta} \text{ (ref)}} (\exists_r^c) \\
 \frac{\frac{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta}{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow w : \exists xA, \Delta} \text{ IH}}{} (\exists_r^{ic}) \\
 \frac{\mathcal{R}, u \leq u, \underline{a} \in D_v, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta}{\frac{\mathcal{R}, u \leq u, \Gamma \Rightarrow w : \exists xA, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \exists xA, \Delta} \text{ (ref)}} (\exists_r^{ic}) \\
 \frac{\frac{\mathcal{R}, \underline{a} \in D_v, \Gamma \Rightarrow w : A(\underline{a}/x), w : \exists xA, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \exists xA, \Delta} \text{ IH}}{} (\exists_r^{ic}) \\
 \frac{\mathcal{R}, u \leq u, \underline{a} \in D_v, w : A(\underline{a}/x), w : \forall xA, \Gamma \Rightarrow \Delta}{\frac{\mathcal{R}, u \leq u, \underline{a} \in D_v, w : \forall xA, \Gamma \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_v, w : \forall xA, \Gamma \Rightarrow \Delta} \text{ (ref)}} (\forall_l^c) \\
 \frac{\frac{\mathcal{R}, \underline{a} \in D_v, w : A(\underline{a}/x), w : \forall xA, \Gamma \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_v, w : \forall xA, \Gamma \Rightarrow \Delta} \text{ IH}}{} (\forall_l^{ic}) \\
 \frac{\mathcal{R}, u \leq u, \underline{a} \in D_v, \Gamma, w : A(\underline{a}/x), w : \forall xA \Rightarrow \Delta}{\frac{\mathcal{R}, u \leq u, \Gamma, w : \forall xA \Rightarrow \Delta}{\mathcal{R}, w : \forall xA, \Gamma \Rightarrow \Delta} \text{ (ref)}} (\forall_l^{ic}) \\
 \frac{\frac{\mathcal{R}, \underline{a} \in D_v, \Gamma, w : A(\underline{a}/x), w : \forall xA, \Gamma \Rightarrow \Delta}{\mathcal{R}, w : \forall xA, \Gamma \Rightarrow \Delta} \text{ IH}}{} (\forall_l^{ic}) \\
 \frac{\mathcal{R}, w \leq w, \underline{a} \in D_w, \underline{a} \in D_w, \Gamma \Rightarrow \Delta}{\frac{\mathcal{R}, w \leq w, \underline{a} \in D_w, \Gamma \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta} \text{ (ref)}} \text{ (nd)} \\
 \frac{\frac{\mathcal{R}, w \leq w, \underline{a} \in D_w, \underline{a} \in D_w, \Gamma \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta} \text{ IH}}{} \text{ (ctr}_{\mathcal{R}}) \\
 \frac{\mathcal{R}, w \leq w, \underline{a} \in D_w, \underline{a} \in D_w, \Gamma \Rightarrow \Delta}{\frac{\mathcal{R}, w \leq w, \underline{a} \in D_w, \Gamma \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta} \text{ (ref)}} \text{ (cd)} \\
 \frac{\frac{\mathcal{R}, w \leq w, \underline{a} \in D_w, \underline{a} \in D_w, \Gamma \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta} \text{ IH}}{} \text{ (ctr}_{\mathcal{R}})
 \end{array}$$

In the (\exists_r^c) , (\exists_r^{ic}) , (\forall_l^c) , and (\forall_l^{ic}) cases above, observe that each rule can be applied after invoking IH for the following reason: if the undirected path from v to w does not go through u , then the path is still present after applying IH, and if the undirected path contains $u \leq u$, then deleting all occurrences of $u \leq u$ from the undirected path yields another undirected path between v and w . The (tra) cases are considered below:

We now argue that the side condition $v \sim^{\mathcal{R}} w$ of (\exists_l^c) , (\exists_r^{ic}) , (\forall_l^c) , and (\forall_l^{ic}) continues to hold after applying IH. If the undirected path from v to w does not contain $u \leq z'$, then the side condition trivially holds. If, on the other hand, the undirected path from v to w contains $u \leq z'$, then another undirected path from v to w is found by replacing all occurrences of $u \leq z'$ with $u \leq z, z \leq z'$, which continues to be present after the invocation of IH. \square

Corollary 2. *All rules in NR are admissible in NInt, NIntQ and NIntQC.*

Proof. We prove the admissibility of each rule of NR in turn. The first three cases we consider are the (wk_l) , (wk_r) , and $(psub)$ cases, though we omit the (wk_r) case since it is similar to the (wk_l) case.

$$\begin{array}{ccc} \frac{\Sigma\{X \rightarrow Y\}}{\mathfrak{L}(\Sigma\{X \rightarrow Y\})} \text{ Thm. 17} & & \frac{\Sigma}{\mathfrak{L}(\Sigma)} \text{ Thm. 17} \\ \dots\dots\dots = & & \dots\dots\dots = \\ \frac{\mathcal{R}, \Gamma, w : X \Rightarrow w : Y, \Delta}{\mathfrak{N}(\mathcal{R}, \Gamma, w : X, w : Z \Rightarrow w : Y, \Delta)} \text{ Thm. 17} & (wk) & \frac{\Lambda}{\Lambda(\underline{a}/\underline{b})} \text{ (psub)} \\ \dots\dots\dots = & & \dots\dots\dots = \\ \frac{\mathfrak{N}(\mathcal{R}, \Gamma, w : X, w : Z \Rightarrow w : Y, \Delta)}{\Sigma\{X, Z \rightarrow Y\}} \text{ Thm. 17} & & \frac{\mathfrak{N}(\Lambda(\underline{a}/\underline{b}))}{\Sigma(\underline{a}/\underline{b})} \text{ Thm. 17} \end{array}$$

We assume that $\mathcal{R}' := \mathcal{R}, \vec{a} \in D_w$ where \vec{a} are all parameters occurring in the labelled formulae of $w : Z$. This ensures that the labelled sequent $\mathcal{R}', \Gamma, w : X, w : Z \Rightarrow w : Y, \Delta$ is nestedlike (cf. Def. 12 and 13). Also, the labelled sequent $\Lambda(\underline{a}/\underline{b})$ in the $(psub)$ derivation is trivially nestedlike since $\mathfrak{L}(\Sigma(\underline{a}/\underline{b})) = \Lambda(\underline{a}/\underline{b})$ (up to a change of labels). This implies that the second invocation of Thm. 17 in both cases is valid.

We next consider the (ctr_l) , (ctr_r) , (nr) , and (mrg_1) cases, but omit the proof of (ctr_r) since it is similar to (ctr_l) . In the top right (i.e. (nr)) derivation, we assume that v is the root of $G(\mathfrak{L}(\Sigma))$ and that w is fresh, meaning we effectively change the root from v to w after applying (wk) .

$$\begin{array}{ccc} \frac{\Sigma\{X, Z, Z \rightarrow Y\}}{\mathfrak{L}(\Sigma\{X, Z, Z \rightarrow Y\})} \text{ Thm. 17} & & \frac{\Sigma}{\mathfrak{L}(\Sigma)} \text{ Thm. 17} \\ \dots\dots\dots = & & \dots\dots\dots = \\ \frac{\mathcal{R}, \Gamma, w : X, w : Z, w : Z \Rightarrow w : Y, \Delta}{\mathfrak{N}(\mathcal{R}, \Gamma, w : X, w : Z \Rightarrow w : Y, \Delta)} \text{ Thm. 17} & (ctr_l) + (ctr_{\mathcal{R}}) & \frac{\mathcal{R}, \Gamma \Rightarrow \Delta}{\mathfrak{N}(\mathcal{R}, w \leq v, \Gamma \Rightarrow \Delta)} \text{ Thm. 17} \\ \dots\dots\dots = & & \dots\dots\dots = \\ \frac{\mathfrak{N}(\mathcal{R}, \Gamma, w : X, w : Z \Rightarrow w : Y, \Delta)}{\Sigma\{X, Z \rightarrow Y\}} \text{ Thm. 17} & & \frac{\mathfrak{N}(\mathcal{R}, w \leq v, \Gamma \Rightarrow \Delta)}{\rightarrow [\Sigma]} \text{ Thm. 17} \end{array}$$

$$\begin{array}{ccc} \frac{\Sigma\{X \rightarrow Y, [\Sigma'], [\Sigma']\}}{\mathfrak{L}(\Sigma\{X \rightarrow Y, [\Sigma'], [\Sigma']\})} \text{ Thm. 17} & & \\ \dots\dots\dots = & & \\ \frac{\mathcal{R}, \mathcal{R}', \mathcal{R}'', \Gamma, \Gamma', \Gamma'' \Rightarrow \Delta', \Delta'', \Delta}{\mathfrak{N}(\mathcal{R}, \mathcal{R}', \mathcal{R}'', \Gamma, \Gamma', \Gamma'' \Rightarrow \Delta', \Delta', \Delta)} \text{ Thm. 17} & (lsub) \times n & \\ \dots\dots\dots = & & \\ \frac{\mathcal{R}, \mathcal{R}', \Gamma, \Gamma' \Rightarrow \Delta', \Delta}{\mathfrak{N}(\mathcal{R}, \mathcal{R}', \Gamma, \Gamma' \Rightarrow \Delta', \Delta)} \text{ Thm. 17} & (ctr_{\mathcal{R}}) + (ctr_l) + (ctr_r) & \\ \dots\dots\dots = & & \\ \frac{\mathfrak{N}(\mathcal{R}, \mathcal{R}', \Gamma, \Gamma' \Rightarrow \Delta', \Delta)}{\Sigma\{X \rightarrow Y, [\Sigma']\}} \text{ Thm. 17} & & \end{array}$$

The labelled sequents $\mathcal{R}, \Gamma, w : X, w : Z \Rightarrow w : Y, \Delta$, $\mathcal{R}, w \leq v, \Gamma \Rightarrow \Delta$, and $\mathcal{R}, \mathcal{R}', \Gamma, \Gamma' \Rightarrow \Delta', \Delta$ are nestedlike, as the conclusion of each derivation is the nested sequent that when input into \mathfrak{L} outputs each such labelled sequent (up to a change of labels). In the bottom (i.e. (mrg_1)) derivation, we assume that the first copy of Σ' is translated to $\mathcal{R}', \Gamma' \Rightarrow \Delta'$ under \mathfrak{L} and that the second copy of Σ' is translated to $\mathcal{R}'', \Gamma'' \Rightarrow \Delta''$ under \mathfrak{L} . Therefore, the graphs $G(\mathcal{R}', \Gamma' \Rightarrow \Delta')$ and $G(\mathcal{R}'', \Gamma'' \Rightarrow \Delta'')$ are isomorphic, and so, the n applications of $(lsub)$ correspond to the replacement of labels in $\mathcal{R}'', \Gamma'' \Rightarrow \Delta''$ with labels in $\mathcal{R}', \Gamma' \Rightarrow \Delta'$ to make these two *identical*. Then, using $(ctr_{\mathcal{R}})$, (ctr_l) , and (ctr_r) , we contract all duplicate copies of relational atoms and labelled formulae to obtain $\mathcal{R}, \mathcal{R}', \Gamma, \Gamma' \Rightarrow \Delta', \Delta$.

We prove the admissibility of the (mrg_2) rule below. We assume that the label w is associated with the component $X_1 \rightarrow Y_1$ and the label v is associated with the component $X_2 \rightarrow Y_2$. As before, the conclusion of the derivation serves as the nested sequent that when input into \mathfrak{L} yields $\mathcal{R}(w/v), \Gamma(w/v), w : X_1, w : X_2 \Rightarrow w : Y_1, w : Y_2, \Delta(w/v)$ (up to a change of labels), thus showing the validity of our second use of Thm. 17.

$$\begin{array}{c}
\frac{\Sigma\{X_1 \rightarrow Y_1, [X_2 \rightarrow Y_2, [\Sigma_1], \dots, [\Sigma_n]]\}}{\mathfrak{L}(\Sigma\{X_1 \rightarrow Y_1, [X_2 \rightarrow Y_2, [\Sigma_1], \dots, [\Sigma_n]]\})} \text{Thm. 17} \\
\frac{\mathcal{R}, w \leq v, \Gamma, w : X_1, v : X_2 \Rightarrow w : Y_1, v : Y_2, \Delta}{\mathcal{R}(w/v), w \leq w, \Gamma(w/v), w : X_1, w : X_2 \Rightarrow w : Y_1, w : Y_2, \Delta(w/v)} (lsub) \\
\frac{\mathcal{R}(w/v), \Gamma(w/v), w : X_1, w : X_2 \Rightarrow w : Y_1, w : Y_2, \Delta(w/v)}{\mathfrak{N}(\mathcal{R}(w/v), \Gamma(w/v), w : X_1, w : X_2 \Rightarrow w : Y_1, w : Y_2, \Delta(w/v))} (ref) \\
\frac{\mathfrak{N}(\mathcal{R}(w/v), \Gamma(w/v), w : X_1, w : X_2 \Rightarrow w : Y_1, w : Y_2, \Delta(w/v))}{\Sigma\{X_1, X_2 \rightarrow Y_1, Y_2, [\Sigma_1], \dots, [\Sigma_n]\}} \text{Thm. 17}
\end{array}$$

Next we consider the (ew_1) case. We assume that $\mathfrak{L}(\Sigma') = \mathcal{R}', \Gamma' \Rightarrow \Delta'$, which shows that the conclusion of the derivation serves as the nested sequent, which when input into \mathfrak{L} outputs $\mathcal{R}, \mathcal{R}', \Gamma, \Gamma' \Rightarrow \Delta', \Delta$ (up to a change of labels). Therefore, the second use of Thm. 17 is valid.

$$\begin{array}{c}
\frac{\Sigma\{X \rightarrow Y\}}{\mathfrak{L}(\Sigma\{X \rightarrow Y\})} \text{Thm. 17} \\
\frac{\mathcal{R}, \Gamma \Rightarrow \Delta}{\mathcal{R}, \mathcal{R}', \Gamma, \Gamma' \Rightarrow \Delta', \Delta} (wk) \\
\frac{\mathfrak{N}(\mathcal{R}, \mathcal{R}', \Gamma, \Gamma' \Rightarrow \Delta', \Delta)}{\Sigma\{X \rightarrow Y, [\Sigma']\}} \text{Thm. 17}
\end{array}$$

In the proof of the admissibility of (ew_2) below, we assume that the label w is associated with the component $X_1 \rightarrow Y_1$, the label v is associated with the component $X_2 \rightarrow Y_2$, and that the label u is fresh. Up to a change of labels, the conclusion of the derivation is the nested sequent that when input into \mathfrak{L} outputs the labelled sequent $\mathcal{R}, w \leq u, u \leq v, \Gamma, w : X_1, v : X_2 \Rightarrow w : Y_1, v : Y_2, \Delta$, justifying our second use of Thm. 17.

$$\begin{array}{c}
\frac{\Sigma\{X_1 \rightarrow Y_1, [X_2 \rightarrow Y_2, [\Sigma_1], \dots, [\Sigma_n]]\}}{\mathfrak{L}(\Sigma\{X_1 \rightarrow Y_1, [X_2 \rightarrow Y_2, [\Sigma_1], \dots, [\Sigma_n]]\})} \text{Thm. 17} \\
\frac{\dots}{\mathcal{R}, w \leq v, \Gamma, w : X_1, v : X_2 \Rightarrow w : Y_1, v : Y_2, \Delta} = \\
\frac{\mathcal{R}, w \leq u, u \leq v, w \leq v, \Gamma, w : X_1, v : X_2 \Rightarrow w : Y_1, v : Y_2, \Delta}{\mathcal{R}, w \leq u, u \leq v, \Gamma, w : X_1, v : X_2 \Rightarrow w : Y_1, v : Y_2, \Delta} \begin{array}{l} (wk) \\ (tra) \end{array} \\
\frac{\mathfrak{N}(\mathcal{R}, w \leq u, u \leq v, \Gamma, w : X_1, v : X_2 \Rightarrow w : Y_1, v : Y_2, \Delta)}{\Sigma\{X_1 \rightarrow Y_1, [\rightarrow [X_2 \rightarrow Y_2, [\Sigma_1], \dots, [\Sigma_n]]]\}} \text{Thm. 17} \\
\frac{\dots}{\dots} =
\end{array}$$

Last, we show the admissibility of (lwr) and (cut) . In the proof of the admissibility of (lwr) below, we assume that the label w is associated with the component $X_1 \rightarrow A, Y_1$ and that the label v is associated with the component $X_2 \rightarrow A, Y_2$. In the proof of the admissibility of (cut) , we assume that the label w is associated with $X \rightarrow A, Y$ and $X, A \rightarrow Y$. As in all previous cases, the end sequent of each derivation serves as the nested sequent (up to a change of labels) that when input into \mathfrak{L} yields the labelled sequent before the (second) invocation of Thm. 17, thus showing that our use of the theorem is valid.

$$\begin{array}{c}
\frac{\Sigma\{X_1 \rightarrow A, Y_1, [X_2 \rightarrow A, Y_2]\}}{\mathfrak{L}(\Sigma\{X_1 \rightarrow A, Y_1, [X_2 \rightarrow A, Y_2]\})} \text{Thm. 17} \\
\frac{\dots}{\mathcal{R}, w \leq v, \Gamma \Rightarrow w : A, v : A, \Delta} = \frac{\mathcal{R}, w \leq v, \Gamma, w : A \Rightarrow v : A, \Delta}{\mathcal{R}, w \leq v, \Gamma \Rightarrow v : A \Rightarrow \Delta} \begin{array}{l} \text{Thm. 3-(i)-(c)} \\ (cut) \end{array} \\
\frac{\mathfrak{N}(\mathcal{R}, w \leq v, \Gamma \Rightarrow v : A, \Delta)}{\Sigma\{X_1 \rightarrow Y_1, [X_2 \rightarrow A, Y_2]\}} \text{Thm. 17} \\
\frac{\dots}{\dots} =
\end{array}$$

$$\begin{array}{c}
\frac{\Sigma\{X \rightarrow A, Y\}}{\mathfrak{L}(\Sigma\{X \rightarrow A, Y\})} \text{Thm. 17} \quad \frac{\Sigma\{X, A \rightarrow Y\}}{\mathfrak{L}(\Sigma\{X, A \rightarrow Y\})} \text{Thm. 17} \\
\frac{\dots}{\mathcal{R}, \Gamma \Rightarrow w : A, \Delta} = \frac{\dots}{\mathcal{R}, \Gamma, w : A \Rightarrow \Delta} = \frac{\dots}{\mathcal{R}, \Gamma \Rightarrow \Delta} \begin{array}{l} \\ (cut) \end{array} \\
\frac{\mathfrak{N}(\mathcal{R}, \Gamma \Rightarrow \Delta)}{\Sigma\{X \rightarrow Y\}} \text{Thm. 17} \\
\frac{\dots}{\dots} =
\end{array}$$

□