Display to Labeled Proofs and Back Again for Tense Logics

AGATA CIABATTONI, Technische Universität Wien, Austria
TIM S. LYON, Technische Universität Dresden, Germany
REVANTHA RAMANAYAKE, University of Groningen, Netherlands
ALWEN TIU, The Australian National University, Australia

We introduce translations between display calculus proofs and labeled calculus proofs in the context of tense logics. First, we show that every derivation in the display calculus for the minimal tense logic Kt extended with general path axioms can be effectively transformed into a derivation in the corresponding labeled calculus. Concerning the converse translation, we show that for Kt extended with path axioms, every derivation in the corresponding labeled calculus can be put into a special form that is translatable to a derivation in the associated display calculus. A key insight in this converse translation is a canonical representation of display sequents as labeled polytrees. Labeled polytrees, which represent equivalence classes of display sequents modulo display postulates, also shed light on related correspondence results for tense logics.

CCS Concepts: • **Theory of computation** → **Proof theory**; **Modal and temporal logics**.

Additional Key Words and Phrases: Nested calculus, Labeled calculus, Display calculus, Effective translations, Tense logic, Modal logic

1 INTRODUCTION

A crucial question for any logic is if it possesses an analytic calculus. An analytic calculus consists of rules that decompose a formula of the logic in a stepwise manner, and can be exploited to prove certain metalogical properties as well as develop automated reasoning methods. Since its introduction in the 1930's, Gentzen's sequent calculus (and equivalently, the tableaux calculus) has been a preferred formalism for constructing analytic calculi due to its simplicity. Unfortunately, this simplicity is also an obstacle: the formalism is not expressive enough to present many logics of interest. In response, many proof-theoretic formalisms extending the syntactic elements of the sequent calculus have been introduced over the last 30 years. Of particular interest in this paper are the formalisms of the labeled calculus [15, 34, 35], nested calculus [6, 23, 28], and display calculus [1, 24]. Each formalism extends the sequent calculus in a seemingly unique way, suggesting distinct

111

strengths, weaknesses, and expressive powers. There are trade-offs in employing one formalism as opposed to another, motivating a study of the interrelationships between the current patchwork (see, e.g. [32]) of proof systems.

In this paper, we consider proof calculi for a special class of multi-modal logics: extensions of the *minimal tense logic* Kt with *general path axioms* $\Pi A \to \Sigma A$ ($\Pi, \Sigma \in \{\diamond, \blacklozenge\}^*$). Tense logics incorporate modalities that reference what is true in successor ($\diamond$) and predecessor states ($\blacklozenge$). Such logics are used to model temporal notions having to do with future and past states of affairs. This class of logics provides a good case study for our proof-theoretic investigations since it includes many interesting/well-known logics and possesses a diverse proof theory.

Numerous analytic proof calculi have been presented for extensions of Kt such as labeled calculi [3, 4], nested calculi [19], and display calculi [23, 24, 36]. Since the term *nested sequent* has been used in the literature to refer to slightly different objects, this is a good time to clarify our terminology. In this paper:

> **Nested sequent:** Any term generated via the BNF grammar $X ::= \varepsilon \mid A \mid X, X \mid \circ\{X\} \mid \bullet\{X\}$ where A is a tense formula.[1] Note that this *extends* the typical definition of a nested sequent in the proof theory literature for modal (rather than tense) logics that uses a single nesting operator (e.g., the grammar for traditional nested sequents is usually given by the following BNF grammar: $X ::= \varepsilon \mid A \mid X, X \mid [X]$).
>
> **Shallow nested calculus** (used here *interchangeably*[2] with **display calculus**) A proof calculus built from nested sequents in the sense above, where *display rules* are used to unpack ('display') a formula nested under $\circ$ and $\bullet$ to bring it to the top-level, where the inference rules operate.
>
> **Deep nested calculus:** A proof calculus built from nested sequents in the sense above where the display rules are dispensed with, and the inference rules can apply inside arbitrary nestings of $\circ$ and $\bullet$ (i.e. *deep inference* is implemented).

Deep nested calculi are better suited than shallow nested calculi for proving e.g. decidability [5, 19] and interpolation [26], due to the absence of the hard-to-control display rules that expand the proof-search space. Both shallow and deep nested calculi are typically *internal* in the sense that each sequent in a proof can be interpreted as a formula of the logic, whereas labeled calculi often appear to be *external* in the sense that the sequents cannot generally be interpreted as a formula of the logic (and use a language that explicitly encodes the semantics).

An effective way to relate calculi is by defining *translations*, i.e. functions that algorithmically transform any proof in a calculus into a proof of the same formula in another calculus. A crucial feature of such functions is that the structural properties of the derivation are preserved in the translation. Such embeddings permit the transfer of certain proof theoretic results, thus alleviating the need for independent proofs in each system, e.g. [14, 20, 27]. Moreover, they shed light on the role of certain syntactic features in proof calculi, and on the general problem of characterizing the relationships between different syntactic and semantic presentations of a logic [31].

In [9] we obtained translations from shallow nested calculi to labeled calculi for Scott-Lemmon axiomatic extensions ($\blacklozenge^h \diamond^i A \to \diamond^j \blacklozenge^k A$ with $h, i, j, k \in \mathbb{N}$) of Kt. This paper extends these results to a larger set of tense logics, and answers an open question posed in that paper regarding the existence of labeled to nested translations for extensions of Kt.

We first show how to translate derivations in shallow nested calculi into derivations in labeled calculi for all general path extensions of Kt. The reverse translation—from labeled to shallow

---

[1]We use ε as the empty string, which in this context denotes the empty sequent.

[2]The alternative term *shallow nested sequent* for *display calculus* is due to [19] whose motivation was to contrast the shallow inference rules of the display calculus with a proof calculus that uses deep inference instead.

nested—employs more sophisticated techniques and is only obtained for *path axiom*—$\Pi A \rightarrow \langle ? \rangle A$ ($\Pi \in \{\diamond, \blacklozenge\}^*$ and $\langle ? \rangle \in \{\diamond, \blacklozenge\}$)—extensions of Kt. Our proof strategy, described in the following paragraphs, ensures that each labeled sequent occurring in the derivation of a theorem of a path extension of Kt is interpretable as a nested sequent. This permits a translation from labeled to shallow nested sequent proofs. This translation witnesses a relation between the relational semantics and algebraic semantics (see e.g. [2, 18]) for tense logics: the labeled calculi are clearly underpinned by the relational semantics; the shallow nested calculi, on the other hand, employ display rules that encode the algebraic residuation property between $\blacklozenge$ (and $\diamond$) in the antecedent and $\square$ (and $\blacksquare$, resp.) in the succedent of an implication. Indeed, the display rules have *no* analog in the labeled calculi since the premise and conclusion translate to the same labeled sequent (see Lemma 3.9).

The ability to display any formula nested under structural connectives using the display rules is a crucial part in Belnap's [1] proof of cut-elimination for arbitrary display calculi. However, the display rules greatly expand the proof search space, in particular when these rules interact with other structural rules (e.g. contraction) or structural rules that capture the modal/tense axioms of the formalized logic. In [19], the authors show how to translate display calculi to deep nested calculi, eliminating the display rules by employing deep inference. In our translation from display calculi to labeled calculi, display rules are not translated to inference rules; rather, they are dealt with by changing the representation of the nested sequent. The key idea is that a nested sequent can naturally be interpreted as a labeled sequent whose binary relation between labels forms a *polytree* (i.e. a directed graph whose underlying undirected graph is a tree). The polytree interpretation of a nested sequent has the crucial property of being invariant under display rules—applications of display rules to a nested sequent do not change its labeled polytree translation. Thus, display-equivalent nested sequents have a canonical representation as a labeled polytree sequent. This representation also sheds light on the correspondence results between shallow and deep nested calculi for tense logics [19]. In particular, we show that the admissibility of display rules is independent from the admissibility of structural rules capturing the path axioms in tense logics, something that was not observed in [19]. This polytree representation also significantly simplifies the proof of Craig interpolation for the class of path extensions of Kt [26].

Given that labeled polytree sequents correspond closely to nested sequents, one strategy to translate a labeled calculus to a shallow nested calculus is to translate a subset of the labeled calculus where all sequents are polytree sequents, and then show that the latter is complete, i.e. that it proves the same set of theorems as the unrestricted labeled calculi. One issue with this approach is that the property of being a polytree is not closed under some structural rules in labeled calculi, i.e. there could be instances of a rule where one of the premises is not a polytree but the conclusion is. To get around this issue, when translating from labeled to shallow nested, we first put our given derivation into a special form that makes use of so-called *propagation rules* [7, 19, 27, 34]. Such rules allow us to eliminate certain structural rules from our labeled calculi and their derivations; this results in an *internal* or *refined* variant of the labeled calculus that—interestingly—inherits the nice properties of the original *external* calculus. This methodology of eliminating structural rules to obtain refined calculi is of practical value in its own right [25]. In this paper, the methodology is used to provide a translation from labeled to shallow nested; however, this method is also useful in that it yields calculi suitable for proof-search and proving interpolation [26, 27]. Furthermore, this new form of the derivation permits an *effective* (i.e. algorithmic) translation into a derivation of a deep nested calculus, that is to say, a step-by-step procedure defined at the level of the proof rules in the derivation; methods from [19] are then applied to the deep nested derivation to further translate the proof into a proof of the corresponding shallow nested calculus. We note that this entire translation process is effective in the above sense, showing that the output derivation

may be obtained algorithmically from local transformation on the input derivation. Our proof of admissibility of structural rules, in favor of propagation rules, for path axioms follows a similar methodology to that used in [19], with one notable difference: in [19], the admissibility of display rules needs to be proved for *every* extension with path axioms, whereas in our case, admissibility of display rules is independent of the extensions, since the polytree representation makes the display rules superfluous. Our result thus suggests that perhaps display rules should be viewed as structural properties of sequents rather than as structural properties of proofs. This is analogous to, for example, internalizing the exchange rule as a property of sequents (i.e. commutativity and associativity of *comma* in the sequent).

The paper is structured as follows: Section 2 introduces the class of tense logics considered along with their associated shallow nested, labeled, and deep nested calculi. Section 3 presents labeled polytrees which are used to give the translation from nested proof systems to labeled proof systems as well as the reverse. In Section 4, we provide an effective translation from shallow nested proofs to labeled proofs for all *general path* extensions of Kt. Section 5 gives the reverse translation from labeled proofs to shallow nested proofs for *path* extensions of Kt. Section 6 discusses consequences and potential applications.

We summarize below the calculi considered in this paper and illustrate the effective proof-transformations (which transform the shape of a derivation and preserve the language of the calculus; indicated by a dotted arrow) and translations (which not only transform the shape of the derivation, but translate the language of the calculus; indicated by solid arrow) obtained in this paper.

Base Calculi and Extensions (GP general path axioms, P path axioms):

Base Calc.	Type	Gen. Path Str. Rules	Path Str. Rules	Propagation Rules
G3Kt [3, 9]	labeled	*LabSt(GP)*	*LabSt(P)*	*LabPr(P)*
SKT [19]	Shal. Nes.	*NestSt(GP)*	*NestSt(P)*	
DKT [19]	Deep Nes.			*DeepPr(P)*

Effective Transformations/Translations:

G3Kt + *LabSt(GP)*   G3Kt + *LabSt(P)* ·····*Lem. 5.10*····➤ G3Kt + *LabPr(P)*

Thm. 4.3 ↑   *Thm.* 5.15 ( ) *Thm.* 4.3   *Lem.* 5.13 ↓

SKT + *NestSt(GP)*   SKT + *NestSt(P)* ◄·············· DKT + *DeepPr(P)*
                                          Lem. 2.21

2 NESTED AND LABELED CALCULI FOR TENSE LOGICS

For convenience, we take the language $\mathcal{L}_{Kt}$ as consisting of formulae in negation normal form. In particular, formulae are built from the literals p and $\overline{p}$ using the $\wedge$, $\vee$, $\Diamond$, $\Box$, $\blacklozenge$, and $\blacksquare$ operators. Note that all results hold also for the full language where the $\neg$, $\rightarrow$, and $\leftrightarrow$ operators are taken as primitive. The language $\mathcal{L}_{Kt}$ is explicitly defined via the following BNF grammar:

$$A ::= p \mid \overline{p} \mid A \wedge A \mid A \vee A \mid \Box A \mid \Diamond A \mid \blacksquare A \mid \blacklozenge A$$

For an introduction to tense logics and their accompanying Kripke semantics, we refer the reader to the following references: [2, 16].

Intuitively, we interpret $\Box A$ as claiming that the formula A holds at every point in the immediate future, whereas $\blacksquare A$ is interpreted as claiming that A holds at every point in the immediate past. Similarly, we interpret the formula $\Diamond A$ as claiming that A holds at some point in the immediate future, while $\blacklozenge A$ intuitively means that A holds at some point in the immediate past.

Define $\overline{A}$ inductively as follows.

(1) If $A = p$, then $\overline{A} = \overline{p}$;

(2) If $A = \overline{p}$, then $\overline{A} = p$;

(3) If $A = B \wedge C$, then $\overline{A} = \overline{B} \vee \overline{C}$;

(4) If $A = B \vee C$, then $\overline{A} = \overline{B} \wedge \overline{C}$;

(5) If $A = \Box B$, then $\overline{A} = \Diamond \overline{B}$;

(6) If $A = \Diamond B$, then $\overline{A} = \Box \overline{B}$;

(7) If $A = \blacksquare B$, then $\overline{A} = \blacklozenge \overline{B}$;

(8) If $A = \blacklozenge B$, then $\overline{A} = \blacksquare \overline{B}$.

We define the negation $\neg A$ of formula A as $\overline{A}$, the conditional $A \to B$ as $\overline{A} \vee B$, and the biconditional $A \leftrightarrow B$ as $A \to B \wedge B \to A$.

The tense logic Kt—a conservative extension of the normal modal logic K—is typically axiomatized as shown below (see, e.g. [2, 8]).

$$A \to (B \to A) \quad (\neg B \to \neg A) \to (A \to B) \quad (A \to (B \to C)) \to ((A \to B) \to (A \to C))$$

$$A \to \Box\blacklozenge A \qquad A \to \blacksquare\Diamond A \qquad \Box A \leftrightarrow \neg\Diamond\neg A \qquad \blacksquare A \leftrightarrow \neg\blacklozenge\neg A \qquad \dfrac{A}{\Box A} \qquad \dfrac{A}{\blacksquare A}$$

$$\Box(A \to B) \to (\Box A \to \Box B) \qquad \blacksquare(A \to B) \to (\blacksquare A \to \blacksquare B) \qquad \dfrac{A \qquad A \to B}{B}$$

As mentioned previously, the logics we consider in this paper are extensions of Kt with *general path axioms* of the form $\langle ?\rangle_1...\langle ?\rangle_n A \to \langle ?\rangle_{n+1}...\langle ?\rangle_{n+m} A$ where each $\langle ?\rangle_j$ is either $\Diamond$ or $\blacklozenge$. Occasionally, we may use $\langle F\rangle, \langle G\rangle, \ldots$ to represent either a $\Diamond$ or a $\blacklozenge$. Also, note that when $n = 0$, the antecedent of the path axiom is free of diamonds (i.e. it is of the form $A \to \langle ?\rangle_1...\langle ?\rangle_m A$), and when $m = 0$, the consequent is free of diamonds (i.e. it is of the form $\langle ?\rangle_1...\langle ?\rangle_n A \to A$). We will use the notation $\Pi A \to \Sigma A$ to represent such axioms. This class of axioms contains many well-known axioms such as reflexivity $A \to \Diamond A$, confluence $\blacklozenge\Diamond A \to \Diamond\blacklozenge A$, and partial-functionality $\blacklozenge\Diamond A \to A$. We will use GP to denote an arbitrary set of general path axioms and write $Kt + GP$ to mean the minimal tense logic Kt extended with the axioms from GP; note that this notation extends straightforwardly to any set S of formulae, i.e. $Kt + S$ will be used to represent extensions of Kt with the formulae from S, as well as the corresponding logic (i.e. the set of theorems). Last, we let $Kt + S \vdash A$ denote that A is a theorem of the logic $Kt + S$.

Path axioms are general path axioms where the consequent of the axiom is restricted to a single-diamond formula, *i.e.* any formula of the form $\langle ?\rangle_1...\langle ?\rangle_n A \to \langle ?\rangle_{n+1} A$ is a path axiom. We focus on this class of axioms because the translation methods presented in this paper only allow us to translate derivations from labeled to nested for the logics $Kt + P$, where P is an arbitrary set of path axioms. Nevertheless, this class of axioms still contains well-known axioms such as transitivity $\Diamond\Diamond A \to \Diamond A$, symmetry $\blacklozenge A \to \Diamond A$, and Euclideanity $\blacklozenge\Diamond A \to \Diamond A$.

2.1 Shallow Nested (Display) Calculi for Tense Logics

We will present Goré *et al.*'s [19] shallow nested calculus SKT for Kt. This calculus can be seen as a one-sided version of Kracht's [24] display calculus for Kt, and also as a variant of Kashima's [23] calculus.

The shallow nested calculus is modular in the sense that certain axiomatic extensions of Kt can be captured by adding equivalent structural rules to SKT. Moreover, SKT allows for a uniform proof of cut-elimination where cut is eliminable from any derivation of SKT extended with any number of *substitution-closed linear structural rules* (see [19] for details). This makes the shallow nested calculus a good candidate for capturing large classes of tense logics in a unified, cut-free manner. The nested sequents of SKT are generated by the following grammar where A is a tense formula in $\mathcal{L}_{Kt}$.

$$X ::= \varepsilon \mid A \mid X, X \mid \circ\{X\} \mid \bullet\{X\}$$

We assume comma to commute and associate, meaning, for example, that we may freely re-write a nested sequent of the form X, Y, Z as Z, X, Y when performing derivations in SKT. Also, ε represents the *empty string* or *empty sequent*, which acts as an identity element for comma (e.g. we identify X, ε with X), and so, ε will be implicit in nested sequents, but not explicitly appear.

A characteristic of nested sequents is that each can be translated into an equivalent formula in the language $\mathcal{L}_{Kt}$, that is, each connective introduced in the language of nested sequents acts as a *proxy* for a logical connective (cf. [1, 19, 24]). The interpretation $\mathcal{I}$ of a nested sequent as a tense formula is defined as follows:

(1) $\mathcal{I}(\varepsilon) = \top$ (3) $\mathcal{I}(X, Y) = \mathcal{I}(X) \vee \mathcal{I}(Y)$ (5) $\mathcal{I}(\bullet\{X\}) = \blacksquare \mathcal{I}(X)$

(2) $\mathcal{I}(A) = A$ for $A \in \mathcal{L}_{Kt}$ (4) $\mathcal{I}(\circ\{X\}) = \square \mathcal{I}(X)$

It will occasionally be useful to refer to the *substructures* of a nested sequent X. We say that a sequent Y is a *substructure* of X if and only if $Y \in \mathfrak{S}(X)$, where the set of *substructures* of X, written $\mathfrak{S}(X)$, is inductively defined as follows:

(1) $\mathfrak{S}(\varepsilon) = \emptyset$ (3) $\mathfrak{S}(X) = \{X\} \cup \mathfrak{S}(Y) \cup \mathfrak{S}(Z)$, if $X = Y, Z$

(2) $\mathfrak{S}(A) = \{A\}$ for $A \in \mathcal{L}_{Kt}$ (4) $\mathfrak{S}(X) = \{X\} \cup \mathfrak{S}(Y)$, if $X = \circ\{Y\}$ or $\bullet\{Y\}$

Definition 2.1 (The Calculus SKT [19]).

$$\frac{}{X, p, \bar{p}} \text{ (id)} \qquad \frac{X, A, B}{X, A \vee B} \text{ ($\vee$)} \qquad \frac{X, A \qquad X, B}{X, A \wedge B} \text{ ($\wedge$)}$$

$$\frac{X, Y, Y}{X, Y} \text{ (c)} \qquad \frac{X}{X, Y} \text{ (w)} \qquad \frac{X, \circ\{Y\}}{\bullet\{X\}, Y} \text{ (rf)} \qquad \frac{X, \bullet\{Y\}}{\circ\{X\}, Y} \text{ (rp)}$$

$$\frac{X, \bullet\{A\}}{X, \blacksquare A} \text{ ($\blacksquare$)} \qquad \frac{X, \circ\{A\}}{X, \square A} \text{ ($\square$)} \qquad \frac{X, \bullet\{Y, A\}, \blacklozenge A}{X, \bullet\{Y\}, \blacklozenge A} \text{ ($\blacklozenge$)} \qquad \frac{X, \circ\{Y, A\}, \lozenge A}{X, \circ\{Y\}, \lozenge A} \text{ ($\lozenge$)}$$

SKT is referred to as a shallow nested sequent calculus because (i) the $\circ\{\cdot\}$ and $\bullet\{\cdot\}$ provide (two types of) nestings and (ii) all the rules are shallow in the sense that they operate at the *root* or *top-level* of the sequent (i.e. rules are only applied to formulae or structures that do not occur within nestings).

Definition 2.2 (Display Property). A calculus has the *display property* if it contains a set of rules (called *display rules*) such that for any sequent X containing a substructure Y, there exists a sequent Z such that Y, Z is derivable from X using only the display rules.

The display property states that any substructure in X can be brought to the *top level* using the display rules. The calculus SKT has the display property when $\{(rp), (rf)\}$ is chosen to be the set of display rules, i.e., the residuation rules (rp) and (rf) serve as the display rules in SKT. A pair of nested sequents are *display equivalent* when they are mutually derivable using only the display rules. The display property is significant since it is a crucial component in the proof of cut-elimination (see [1]).

A modular method of obtaining a cut-free extension of the base calculus for Kt by a large class of axioms inclusive of the general path axioms was introduced in [24] (see also [10]). Following [24], we present the rule (GP) corresponding to a general path axiom $\langle?\rangle_1...\langle?\rangle_n A \rightarrow \langle?\rangle_{n+1}...\langle?\rangle_{n+m} A$:

$$\frac{X, \star_{n+1}\{... \star_{n+m} \{Y\}...\}}{X, \star_1\{... \star_n \{Y\}...\}} \text{ (GP)}$$

Here if $\langle?\rangle_j = \lozenge$ then $\star_j = \circ$, and if $\langle?\rangle_j = \blacklozenge$ then $\star_j = \bullet$.

Since path axioms form a proper subclass of the general path axioms, the rule (GP) can be specialized to the rule (Path) for any given path axiom $\langle?\rangle_1...\langle?\rangle_n A \rightarrow \langle?\rangle_{n+1} A$:

$$\frac{X, \star_{n+1}\{Y\}}{X, \star_1\{\dots \star_n \{Y\}\dots\}} \ (\text{Path})$$

THEOREM 2.3 ([19]). *The* (*cut*) *rule*

$$\frac{X, A \qquad \overline{A}, Y}{X, Y} \ (\textit{cut})$$

is admissible in SKT + NestSt(GP).

THEOREM 2.4 ([19]). *Kt + GP ⊢ A iff A is derivable in SKT + NestSt(GP).*

2.2 Labeled Calculi for Tense Logics

Labeled sequents [13, 29] generalize Gentzen sequents by prefixing *state variables* to formulae occurring in the sequent and by making the relational semantics explicit in the syntax. Labeled sequents are defined via the BNF grammar below:

$$\Lambda ::= \varepsilon \mid x : A \mid \Lambda, \Lambda \mid Rxy, \Lambda$$

where $A \in \mathcal{L}_{\mathsf{Kt}}$, and x and y are among a denumerable set $x, y, z, \dots$ of labels. We often write a labeled sequent Λ as $\mathcal{R}, \Gamma$ where $\mathcal{R}$ consists of the *relational atoms* of the form Rxy occurring in Λ and Γ consists of the *labeled formulae* of the form $x : A$ occurring in Λ. Additionally, characters such as $\mathcal{R}, \mathcal{Q}, \dots$ will be used to denote *sets* of relational atoms and Greek letters such as $\Gamma, \Delta, \dots$ will be used to denote *multisets* of labeled formulae. As in the case of nested sequents, we assume that comma commutes and associates, meaning that each labeled sequent Λ can indeed be written in the form above, and also assume that ε represents the *empty string* or *empty sequent*, which acts as an identity element for comma and occurs only implicitly in labeled sequents.

A labeled sequent can be viewed as a directed graph (defined using $\mathcal{R}$) with formulae decorating each node [9]. Note that in a labeled sequent $\Lambda = \mathcal{R}, \Gamma$ commas between relational atoms are interpreted conjunctively, the comma between $\mathcal{R}$ and Γ is interpreted as an implication, and the commas between the labeled formulae in Γ are interpreted disjunctively.

Viganò [35] constructed labeled sequent calculi for non-classical logics whose semantics are defined by Horn formulae. Negri [30] extended the method to generate cut-free and contraction-free labeled sequent calculi for the large family of modal logics whose Kripke semantics are defined by geometric (first-order) formulae, similar to what had been achieved by Simpson for intuitionistic modal logics in his dissertation [34]. The proof of cut-elimination is general in the sense that it applies uniformly to every modal logic defined by geometric formulae; this result has been extended to intermediate and other non-classical logics [3, 11] and to arbitrary first-order formulae [12].

We begin by extending in the natural way the usual labeled sequent calculus for K to a labeled sequent calculus for Kt.

Definition 2.5 (The labeled sequent calculus G3Kt [3, 9]).

$$\frac{}{\mathcal{R}, x : p, x : \overline{p}, \Gamma} \ (\text{id}) \qquad \frac{\mathcal{R}, x : A, x : B, \Gamma}{\mathcal{R}, x : A \vee B, \Gamma} \ (\vee) \qquad \frac{\mathcal{R}, x : A, \Gamma \qquad \mathcal{R}, x : B, \Gamma}{\mathcal{R}, x : A \wedge B, \Gamma} \ (\wedge)$$

$$\frac{\mathcal{R}, Rxy, y : A, \Gamma}{\mathcal{R}, x : \square A, \Gamma} \ (\square)^* \qquad \frac{\mathcal{R}, Rxy, y : A, x : \Diamond A, \Gamma}{\mathcal{R}, Rxy, x : \Diamond A, \Gamma} \ (\Diamond)$$

$$\frac{\mathcal{R}, Ryx, y : A, \Gamma}{\mathcal{R}, x : \blacksquare A, \Gamma} \ (\blacksquare)^* \qquad \frac{\mathcal{R}, Ryx, y : A, x : \blacklozenge A, \Gamma}{\mathcal{R}, Ryx, x : \blacklozenge A, \Gamma} \ (\blacklozenge)$$

The ($\square$) and ($\blacksquare$) rules have a side condition: (*) the variable y does not occur in the conclusion. When a variable is not allowed to occur in the conclusion of an inference, we refer to it as an *eigenvariable*.

A general path axiom is a Sahlqvist formula, and hence it has a first-order frame correspondent which can be computed—even in the case of tense logics (see [2]). Following the method in [30], the labeled structural rule (GP) corresponding to a general path axiom $\Pi A \to \Sigma A$ is obtained below. We let $R_\Diamond xy := Rxy$, and $R_\blacklozenge xy := Ryx$, and use $R_{\langle ? \rangle} xy \in \{R_\Diamond xy, R_\blacklozenge xy\}$, that is, $R_{\langle ? \rangle} xy$ may represent either Rxy or Ryx. Furthermore, we use the following notation: $\mathcal{R}_\Pi xy := R_{\langle ? \rangle_1} xy_1, ..., R_{\langle ? \rangle_m} y_m y$ for $\Pi = \langle ? \rangle_1 ... \langle ? \rangle_m$ and $\mathcal{R}_\Sigma xy := R_{\langle ? \rangle_1} xz_1, ..., R_{\langle ? \rangle_n} y_n y$ for $\Sigma = \langle ? \rangle_1 ... \langle ? \rangle_n$.

$$\frac{\mathcal{R}, \mathcal{R}_\Pi xy, \mathcal{R}_\Sigma xy, \Gamma}{\mathcal{R}, \mathcal{R}_\Pi xy, \Gamma} \ (\text{GP})^*$$

This rule also has a side condition: $(*)$ all variables occurring in the relational atoms $\mathcal{R}_\Sigma xy$ with the exception of x and y are eigenvariables.

REMARK 2.6. *In the rule above, some care is needed in the boundary cases when Π or Σ are empty strings of diamonds. The table below specifies the instances of the rule depending on whether the string is non-empty (marked with +), or empty (marked with ϵ):*

Π	Σ	Premise	Conclusion
+	+	$\mathcal{R}, R_\Pi xy, R_\Sigma xy, \Gamma$	$\mathcal{R}, R_\Pi xy, \Gamma$
+	ϵ	$\mathcal{R}, R_\Pi xy, x = y, \Gamma$	$\mathcal{R}, R_\Pi xy, \Gamma$
ϵ	+	$\mathcal{R}, R_\Sigma xx, \Gamma$	$\mathcal{R}, \Gamma$
ϵ	ϵ	$\mathcal{R}, \Gamma$	$\mathcal{R}, \Gamma$

Note that $[x/y]$ stands for the substitution of the label y by the label x. Also, when $\Pi = \epsilon$ or $\Sigma = \epsilon$, $R_\Pi xy$ and $R_\Sigma xy$ are taken to be $x = y$. For the second entry of the table, we must extend our labelled sequents to include equations of the form $x = y$ as relational atoms and extend our calculus with equality and substitution rules (see [30]). All results of the paper continue to hold with respect to the addition of such rules; however, we omit their presentation for simplicity. For the third and fourth entries in the table, the equality symbols that arise have been eliminated through substitutions and suitable argumentation. This argumentation can be formalized using the equality and substitution rules specified in [30].

Since particular attention will be paid to the class of path axioms (specifically in section 5.2), we also explicitly give the structural rule (Path) which is an instance of (GP) and corresponds to a path axiom $\Pi A \to \langle ? \rangle A$:

$$\frac{\mathcal{R}, R_\Pi xy, R_{\langle ? \rangle} xy, \Gamma}{\mathcal{R}, R_\Pi xy, \Gamma} \ (\text{Path})$$

We use the name *LabSt(GP)* to represent the set of labeled structural rules corresponding to a set *GP* of general path axioms and the name *LabSt(P)* to refer to the set of labeled structural rules corresponding to a set *P* of path axioms.

It is straightforward to apply the arguments and methods concerning labeled calculi for modal and tense logics, presented in [3, 30], to conclude the following:

LEMMA 2.7. *Let $\mathcal{R}$ and Q be sets of relational atoms with $\mathcal{R} \cap Q = \emptyset$, and Γ and Δ be multisets of labeled formulae. The following rules are admissible in G3Kt + LabSt(GP):*

$$\frac{\mathcal{R}, \Gamma}{\mathcal{R}[x/y], \Gamma[x/y]} \ [\text{s}] \qquad \frac{\mathcal{R}, \Gamma, \Delta, \Delta}{\mathcal{R}, \Gamma, \Delta} \ (c) \qquad \frac{\mathcal{R}, \Gamma}{\mathcal{R}, Q, \Gamma, \Delta} \ (w) \qquad \frac{\mathcal{R}, \Gamma, x : A \qquad \mathcal{R}, \Gamma, x : \overline{A}}{\mathcal{R}, \Gamma} \ (cut)$$

We assume that duplicates of relational atoms are removed from $\mathcal{R}[x/y]$ in the conclusion of the $[\text{s}]$ rule.

THEOREM 2.8. *$Kt + GP \vdash A$ iff $x : A$ is derivable in G3Kt + LabSt(GP).*

2.3 Deep Nested Calculi for Tense Logics

In this section we present Goré *et al.*'s [19] deep nested calculus DKT for Kt, as well as extensions of DKT with inference rules—referred to as *propagation rules*—that correspond to the class of *path axioms*. Although we will show how to translate shallow nested derivations into labeled derivations for the logics Kt + GP, we consider path axioms here because the reverse translation from labeled proofs to shallow nested proofs is only known for the smaller class of logics Kt + P. The deep nested calculi presented here will be used to facilitate and simplify the reverse translation.

Our calculi make use of nested sequents from the same language as SKT. Every nested sequent $X := Y, \circ\{Z_1\}, ..., \circ\{Z_n\}, \bullet\{W_1\}, ..., \bullet\{W_m\}$ (Y contains no nesting) may be represented as a tree with two types of edges [19, 23]. The tree of X, denoted $tree(X)$, is shown below:
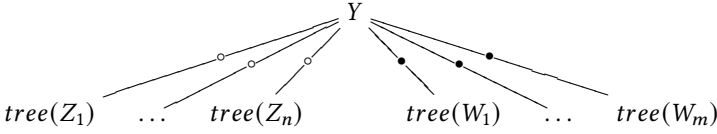
A nested sequent that contains holes in place of formulae is called a *context*. Like nested sequents, contexts may be represented as trees, but where nodes are additionally labeled with holes. A context with a single hole is written as $X[]$ and a context with multiple holes is written as $X[] \cdots []$. We may compose a context with sequents to obtain a sequent (e.g. $X[Y_1] \cdots [Y_n]$ is a sequent where $X[] \cdots []$ is a multi-hole context and $Y_1, ..., Y_n$ are sequents); graphically, this corresponds to fusing the root of the tree of each sequent with the node in the context where the associated hole occurs. Note that this notation is the opposite of what is often used for nested sequent calculi for *modal* logics in the literature, though is consistent with the notation used in the literature for nested sequent calculi for tense logics (cf. [19]).

When representing a context graphically, each hole will label a unique node in the corresponding tree. For a single-hole context we write $X[]_i$ to indicate the node i where the hole occurs, and for a multi-hole context we write $X[]_{i_1} \cdots []_{i_n}$ to indicate the unique nodes in the tree that correspond to each hole.

Definition 2.9 (The Calculus DKT [19][3]).

$$\frac{}{X[p, \overline{p}]} \text{ (id)} \qquad \frac{X[A, Y] \qquad X[B, Y]}{X[A \wedge B, Y]} \text{ ($\wedge$)} \qquad \frac{X[A, B, Y]}{X[A \vee B, Y]} \text{ ($\vee$)}$$

$$\frac{X[\blacksquare A, \bullet\{A\}]}{X[\blacksquare A]} \text{ ($\blacksquare$)} \qquad \frac{X[\bullet\{Y, A\}, \blacklozenge A]}{X[\bullet\{Y\}, \blacklozenge A]} \text{ ($\blacklozenge_1$)} \qquad \frac{X[\circ\{Y, \blacklozenge A\}, A]}{X[\circ\{Y, \blacklozenge A\}]} \text{ ($\blacklozenge_2$)}$$

$$\frac{X[\square A, \circ\{A\}]}{X[\square A]} \text{ ($\square$)} \qquad \frac{X[\circ\{Y, A\}, \lozenge A]}{X[\circ\{Y\}, \lozenge A]} \text{ ($\lozenge_1$)} \qquad \frac{X[\bullet\{Y, \lozenge A\}, A]}{X[\bullet\{Y, \lozenge A\}]} \text{ ($\lozenge_2$)}$$

We now aim to define propagation rules for deep nested calculi. To do this, we follow the work in [19] and first introduce path axiom inverses, compositions of path axioms, and the completion of a set of path axioms in order to define the corresponding set of equivalent propagation rules. Additions of these propagation rules to DKT will yield cut-free, sound, and complete deep nested calculi for logics Kt + P. Note that we define $\langle ? \rangle^{-1} = \lozenge$ if $\langle ? \rangle = \blacklozenge$, and $\langle ? \rangle^{-1} = \blacklozenge$, if $\langle ? \rangle = \lozenge$.

Definition 2.10 (Path Axiom Inverse [19]). If F is a path axiom of the form $\langle F_1 \rangle \cdots \langle F_n \rangle A \rightarrow \langle F \rangle A$, then we define the *inverse of F* to be

---

[3]As shown in [19], copying the principal formula in the ($\square$) and ($\blacksquare$) rules is useful when performing proof-search, despite being unnecessary for completeness of the calculus. Still, we make use of the same rules here since we will leverage methods presented in [19] that make use of the calculus DKT in the form above.

$$I(F) := \langle F_n \rangle^{-1} \cdots \langle F_1 \rangle^{-1} A \to \langle F \rangle^{-1} A$$

Given a set of path axioms P, we define the *set of inverses* to be the set $I(P) := \{I(F) \mid F \in P\}$.

Definition 2.11 (Composition of Path Axioms [19]). Given two path axioms

$$F := \langle F_1 \rangle \cdots \langle F_n \rangle A \to \langle F \rangle A \text{ and } G := \langle G_1 \rangle \cdots \langle G_m \rangle A \to \langle G \rangle A$$

we say *F is composable with G at i* iff $\langle F \rangle = \langle G_i \rangle$. We define the *composition*

$$F \triangleright^i G := \langle G_1 \rangle \cdots \langle G_{i-1} \rangle \langle F_1 \rangle \cdots \langle F_n \rangle \langle G_{i+1} \rangle \cdots \langle G_m \rangle A \to \langle G \rangle A$$

when F is composable with G at i.

Using these individual compositions, we define the following *set of compositions*:

$$F \triangleright G := \{F \triangleright^i G \mid F \text{ is composable with } G \text{ at } i\}$$

Example 2.12. As an example, we can compose the axiom $\diamond\diamond A \to \blacklozenge A$ with $\blacklozenge \diamond A \to \diamond A$ to obtain $\diamond\diamond\diamond A \to \diamond A$.

Definition 2.13 (Completion [19]). The *completion* of a set P of path axioms, written P^*, is the smallest set of path axioms containing P such that

(1) $\diamond A \to \diamond A, \blacklozenge A \to \blacklozenge A \in P^*$

(2) If $F, G \in P^*$ and F is composable with G, then $F \triangleright G \subseteq P^*$.

After introducing further notions necessary to define the propagation rules, we will give examples showing the significance of defining the rules relative to the *completion* of a set of path axioms, rather than defining the rules relative to just the given set of path axioms. As will be shown, without defining the rules relative to the completion, the corresponding set of rules would not be enough to achieve completeness of the resulting calculus.

Let us now recall the notion of a propagation graph and the notion of a path in a propagation graph from [19]. We introduce these concepts using the diamond rules of DKT as an example. The diamond rules $(\diamond_1)$, $(\diamond_2)$, $(\blacklozenge_1)$, $(\blacklozenge_2)$ can be read bottom-up as propagating formulae to nodes in the tree of a sequent.

For example, the $(\diamond_1)$ rule propagates an A to a node along a $\circ$-edge, whereas the $(\diamond_2)$ rule propagates an A backward along a $\bullet$-edge. Similarly, the $(\blacklozenge_1)$ rule propagates an A forward to a node along a $\bullet$-edge, and the $(\blacklozenge_2)$ rule propagates an A backward along a $\circ$-edge. These movements are represented in the diagram below:
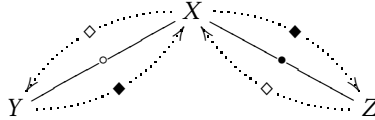
This understanding of how formulae are propagated is crucial to define the propagation rules for deep nested calculi. In fact, as will be explained below, each path axiom can be read as an instruction that expresses how to propagate a formula along some path. We therefore give a precise definition of the *propagation graph* of a sequent, which explicitly specifies how formulae may move when being propagated throughout the tree of a sequent.

Definition 2.14 (Propagation Graph [19]). Let X be a nested sequent where N is the set of nodes in $tree(X)$. We define the *propagation graph* $PG(X) = (N, E)$ *of X* to be the directed graph with set of nodes N and set of edges E each labeled with either a $\diamond$ or $\blacklozenge$ as follows:

(1) For every node $n \in N$ and $\circ$-child m of n, we have a labeled edge $(n, m, \diamond) \in E$ and a labeled edge $(m, n, \blacklozenge) \in E$.

(2) For every node $n \in N$ and $\bullet$-child m of n, we have a labeled edge $(n, m, \blacklozenge) \in E$ and a labeled edge $(m, n, \diamondsuit) \in E$.

LEMMA 2.15. *Suppose that X and Y are display equivalent nested sequents. Then, $PG(X) = PG(Y)$.*

PROOF. We prove the result by induction on the minimum number of display inferences needed to derive Y from X.

Base case. Assume w.l.o.g. that $X = Z, \circ\{W\}$ and $Y = \bullet\{Z\}, W$ so that Y is derivable from X with a single application of a display rule. Let $PG(Z) = (N_1, E_1)$ and $PG(W) = (N_2, E_2)$ with n_1 the root of $tree(Z)$ and n_2 the root of $tree(W)$. Observe that $PG(X) = (N, E)$, where $N = N_1 \cup N_2$ and $E = E_1 \cup E_2 \cup \{(n_1, n_2, \diamondsuit), (n_2, n_1, \blacklozenge)\}$, which is identical to $PG(Y)$ by definition.

Inductive step. Suppose that $n + 1$ is the minimum number of display inferences needed to derive Y from X. It follows that there exists a nested sequent Z such that Z is derivable from X with one display inference, and Y is derivable from Z with n applications of the display rules. By the base case we know that $PG(X) = PG(Z)$, and by the inductive hypothesis, $PG(Z) = PG(Y)$. □

Definition 2.16 (Path [19]). A *path* is a sequence of nodes and diamonds (labeling edges) of the form:

$$n_1, \langle? \rangle_1, n_2, \langle? \rangle_2, ..., \langle? \rangle_{k-1}, n_k$$

in the propagation graph $PG(X)$ such that n_i is connected to n_{i+1} by an edge labeled with $\langle? \rangle_i$. Note that we allow repetitions of nodes along a path (e.g. $n, \diamondsuit, m, \blacklozenge, n$ is a path). For a given path $\pi = n_1, \langle? \rangle_1, n_2, \langle? \rangle_2, ...\langle? \rangle_{k-1}, n_k$, we define the *string of π* to be the string of diamonds $\Pi = \langle? \rangle_1 \langle? \rangle_2 ... \langle? \rangle_{k-1}$.

Definition 2.17 (Deep Nested Propagation Rules [19]). Let P be a set of path axioms. The set of propagation rules *DeepPr(P)* contains all rules of the form:

$$\frac{X[\langle? \rangle A]_i [A]_j}{X[\langle? \rangle A]_i [\emptyset]_j} \text{ (dp)}$$

where there is a path π from i to j in the propagation graph of the premise and $\Pi A \to \langle? \rangle A \in (P \cup I(P))^*$ with Π the string of π.

It should be noted that two different sets P and P' of path axioms can generate the same set of propagation rules, i.e. $(P \cup I(P))^* = (P' \cup I(P'))^*$. For example, both $\{A \to \diamondsuit A, \blacklozenge \diamondsuit A \to \diamondsuit A\}$ and $\{A \to \diamondsuit A, \blacklozenge A \to \diamondsuit A, \diamondsuit \diamondsuit A \to \diamondsuit A\}$ yield the same set of propagation rules, which would provide a deep nested calculus for tense S5.

Example 2.18 (Necessity of Inverses). Let us now demonstrate why inverses must be taken into account when defining propagation rules. Suppose that we did not define the set of propagation rules relative to the set $(\{\diamondsuit\diamondsuit A \to \diamondsuit A\} \cup \{\blacklozenge\blacklozenge A \to \blacklozenge A\})^*$, but rather, we defined the set of propagation rules relative to the set $\{\diamondsuit\diamondsuit A \to \diamondsuit A\}^*$. All propagation rules in this restricted set are of the form below (where there is a path of the form $i, \diamondsuit, \ldots, \diamondsuit, j$ of length $n \geq 1$ from i to j):

$$\frac{X[\diamondsuit A]_i [A]_j}{X[\diamondsuit A]_i [\emptyset]_j} \text{ (dp)}$$

We now explain why this restricted set of propagation rules–that does not take inverses into account—leads to an incomplete calculus. Below, we attempt to give a root-first derivation of $I(\diamondsuit\diamondsuit p \to \diamondsuit p) = \blacklozenge\blacklozenge p \to \blacklozenge p$, which is a theorem of the logic Kt + $\diamondsuit\diamondsuit A \to \diamondsuit A$ and should therefore be derivable:

$$\frac{\bullet\{\bullet\{\overline{p}\}\}, \blacklozenge p}{\bullet\{\blacksquare\overline{p}\}, \blacklozenge p} \ (\blacksquare)$$

$$\frac{\bullet\{\blacksquare\overline{p}\}, \blacklozenge p}{\blacksquare\blacksquare\overline{p}, \blacklozenge p} \ (\blacksquare)$$

$$\frac{\blacksquare\blacksquare\overline{p}, \blacklozenge p}{\blacksquare\blacksquare\overline{p} \vee \blacklozenge p} \ (\vee)$$

$$\cdots\cdots\cdots\cdots\cdots$$

$$\blacklozenge\blacklozenge p \rightarrow \blacklozenge p \ =$$

Observe that no propagation rule from the restricted set is applicable to the top sequent of the derivation because no propagation rule acts along a path of the form $i, \blacklozenge, \ldots, \blacklozenge, j$. However, if we allow ourselves to define the propagation rules relative to the set $(\{\Diamond\Diamond A \rightarrow \Diamond A\} \cup \{\blacklozenge\blacklozenge A \rightarrow \blacklozenge A\})^*$, then we also have the following rules in our calculus (where there is a path of the form $i, \blacklozenge, \ldots, \blacklozenge, j$ of length $n \geq 1$ from i to j):

$$\frac{X[\blacklozenge A]_i[A]_j}{X[\blacklozenge A]_i[\emptyset]_j} \ (\mathsf{dp})$$

Using this rule we can complete the derivation by deriving the top sequent of the above derivation from the initial sequent $\bullet\{\bullet\{\overline{p}, p\}\}, \blacklozenge p$:

$$\frac{}{\bullet\{\bullet\{\overline{p}, p\}\}, \blacklozenge p} \ (\mathsf{id})$$

$$\frac{\bullet\{\bullet\{\overline{p}, p\}\}, \blacklozenge p}{\bullet\{\bullet\{\overline{p}\}\}, \blacklozenge p} \ (\mathsf{dp})$$

Example 2.19 (Necessity of Compositions). Suppose we are given the set $P = \{\Diamond\blacklozenge\Diamond A \rightarrow \Diamond A, \Diamond\Diamond A \rightarrow \blacklozenge A\}$. One of the composition formulae derivable in the logic $\mathsf{Kt} + P$ is $\Diamond\Diamond\Diamond\Diamond A \rightarrow \Diamond A$. Our example below demonstrates the necessity of defining *DeepPr(P)* relative to the completion $(P \cup I(P))^*$ (which takes into account compositions) instead of just P.

If we define our propagation rules relative to just P, then we will have the following two propagation rules in our calculus:

$$\frac{X[\Diamond A]_i[A]_j}{X[\Diamond A]_i[\emptyset]_j} \ (\mathsf{dp}) \qquad\qquad \frac{X[\blacklozenge A]_k[A]_n}{X[\blacklozenge A]_k[\emptyset]_n} \ (\mathsf{dp})$$

The left rule is applicable when there is a path of the form $i, \Diamond, n_1, \blacklozenge, n_2, \Diamond, j$ from node i to j, and the right rule is applicable when there is a path of the form $k, \Diamond, n_1, \Diamond, n$ from k to n in the respective propagation graphs.

We now attempt to derive $\Diamond\Diamond\Diamond\Diamond p \rightarrow \Diamond p$, and show that no sequence of rules applied backward can give a proof of the formula:

$$\frac{\circ\{\circ\{\circ\{\circ\{\overline{p}\}\}\}\}, \Diamond p}{\square\square\square\square\overline{p}, \Diamond p} \ (\square) \times 4$$

$$\frac{\square\square\square\square\overline{p}, \Diamond p}{\square\square\square\square\overline{p} \vee \Diamond p} \ (\vee)$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\Diamond\Diamond\Diamond\Diamond p \rightarrow \Diamond p \ =$$

None of the rules in DKT or in the restricted set of propagation rules are bottom-up applicable to the top sequent. However, since $\Diamond\Diamond\Diamond\Diamond A \rightarrow \Diamond A \in (P \cup I(P))^*$, if we allow the addition of propagation rules to correspond to axioms in $(P \cup I(P))^*$ rather than just P, then we have the following rule in our calculus (where there is a path of the form $c, \Diamond, n_1, \Diamond, n_2, \Diamond, n_3, \Diamond, p$ from c to p):

$$\frac{X[\Diamond A]_c[A]_p}{X[\Diamond A]_c[\emptyset]_p} \ (\mathsf{dp})$$

This can be used to prove the formula $\Diamond\Diamond\Diamond\Diamond p \rightarrow \Diamond p$ by deriving the top sequent in the above derivation from the initial sequent $\circ\{\circ\{\circ\{\circ\{\overline{p}, p\}\}\}\}, \Diamond p$:

$$\frac{}{\circ\{\circ\{\circ\{\circ\{\overline{p}, p\}\}\}\}, \Diamond p} \ (\mathsf{id})$$

$$\frac{\circ\{\circ\{\circ\{\circ\{\overline{p}, p\}\}\}\}, \Diamond p}{\circ\{\circ\{\circ\{\circ\{\overline{p}\}\}\}\}, \Diamond p} \ (\mathsf{dp})$$

Lemma 2.20 ([19]). *The following rules are admissible in DKT + DeepPr(P):*

$$\frac{X[Y]}{X[Y,Z]} \ (w) \quad \frac{X[Y,Y]}{X[Y]} \ (c) \quad \frac{X,\circ\{Y\}}{\bullet\{X\},Y} \ (rf) \quad \frac{X,\bullet\{Y\}}{\circ\{X\},Y} \ (rp)$$

Lemma 2.21 ([19]). *Let P be a set of path axioms. Every derivation in SKT + NestSt(P) of a sequent X is transformable [effectively relatable] to a derivation in DKT + DeepPr(P), and vice-versa.*

We have inserted the term "effectively relatable" to emphasize that the proof in [19] is via a local transformation (i.e. a step-by-step procedure defined at the level of the proof rules in the derivation). Indeed, the forward direction of the above lemma is shown by induction on the height of the given derivation ([19, Lem. 6.13]), and implies that we can effectively transform a derivation $\mathcal{D}$ in DKT + *DeepPr(P)* into a derivation $\mathcal{D}'$ of the same end sequent in SKT + *NestSt(P)*. The reverse direction follows from the fact that SKT + *NestSt(P)* can mimic propagation rules ([19, Lem. 6.12]). Also, observe that the above lemma implies cut-free completeness for each deep nested calculus DKT + *DeepPr(P)*.

Theorem 2.22 ([19]). *Let P be a set of path axioms. Kt + P ⊢ A iff A is cut-free derivable in DKT + DeepPr(P).*

3 NESTED SEQUENTS AND LABELED POLYTREES

In this section we show how to translate (back and forth) a nested sequent into a labeled polytree (called a *labeled UT* in [9]). These graphical structures facilitate the translations between nested and labeled proofs.

We write $V = V_1 \sqcup V_2$ to mean that $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$. The multiset union of multisets M_1 and M_2 is denoted $M_1 \uplus M_2$. A *labeling function L* is a map from a set V to a multiset of tense formulae. For labeling functions L_1 and L_2 on the sets V_1 and V_2 respectively, let $L_1 \cup L_2$ be the labeling function on $V_1 \cup V_2$ defined as follows:

$$(L_1 \cup L_2)(x) = \begin{cases} L_1(x) & x \in V_1, x \notin V_2 \\ L_2(x) & x \notin V_1, x \in V_2 \\ L_1(x) \uplus L_2(x) & x \in V_1, x \in V_2 \end{cases}$$

A *labeled graph* (V, E, L) is a directed graph (V, E) equipped with a labeling function L on V.

Definition 3.1 (Labeled Graph Isomorphism). We say that two labeled graphs $G_1 = (V_1, E_1, L_1)$ and $G_2 = (V_2, E_2, L_2)$ are isomorphic (written $G_1 \cong G_2$) if and only if there is a function $f : V_1 \to V_2$ such that:

(i) f is bijective;
(ii) for every $x, y \in V_1$, $(x, y) \in E_1$ iff $(fx, fy) \in E_2$;
(iii) for every $x \in V_1$, $L_1(x) = L_2(fx)$.

Definition 3.2 (Labeled Polytree). A *labeled polytree* is a labeled graph whose underlying (i.e. undirected) graph is a tree, *i.e.* there exists exactly one path of undirected edges between every pair of distinct nodes.

Example 3.3. The following two graphs represent labeled polytrees, where each node is decorated with a multiset M_i of formulae:
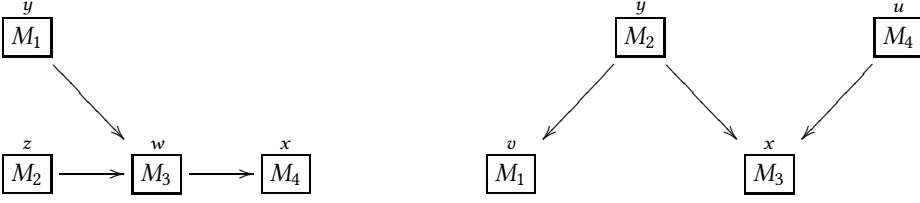
Polytrees have been discussed in the graph theory literature and have also found applications in computer science [22, 33].

3.1 Interpreting a Nested Sequent as a Labeled Polytree

Every nested sequent has a natural interpretation as a labeled tree with two types of directed edges: $\overset{\circ}{\to}$ and $\overset{\bullet}{\to}$ [19, 23]. If we interpret every directed edge $\alpha \overset{\bullet}{\to} \beta$ as the directed edge $\alpha \overset{\circ}{\leftarrow} \beta$, we can then interpret every nested sequent as a connected labeled graph with a *single* type of directed edge (so we can drop the ∘ symbol altogether). Moreover, it is easy to see that its underlying graph (*i.e.* the undirected graph obtained by treating all edges as undirected) is a tree, and that every nested sequent can be interpreted naturally as a labeled polytree.

Example 3.4 (Transforming a Nested Sequent into a Labeled Sequent). First, interpret the nested sequent $A, \circ\{B, \bullet\{\varepsilon\}\}, \bullet\{D, E, \bullet\{F\}, \circ\{G\}\}$ as the labeled tree with two types of directed edges, below left. Next, convert the labeled tree to a labeled polytree (with directed edges of a single type) by reading each $\alpha \overset{\bullet}{\to} \beta$ as $\alpha \leftarrow \beta$ (below right) and remove the ∘-typing from the remaining edges.
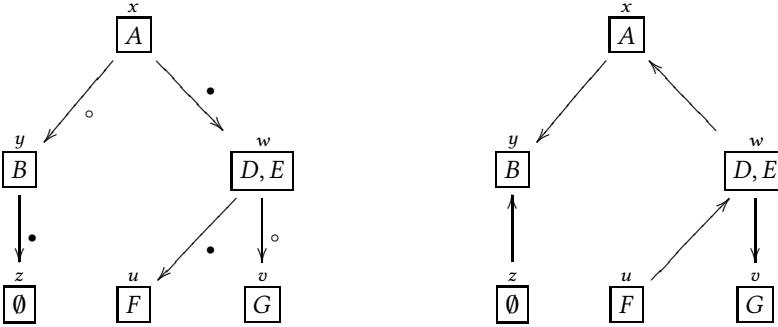
For concreteness let us formally define the map $\mathfrak{L}$ from a nested sequent to a labeled polytree.

Definition 3.5 (The Translation $\mathfrak{L}$). We define the map $\mathfrak{L}_x(X)$ recursively on the structure of the input nested sequent X as follows:[4]

$$
\begin{aligned}
\mathfrak{L}_x(\varepsilon) &:= (\emptyset, \emptyset, \emptyset) \\
\mathfrak{L}_x(A) &:= (\{x\}, \emptyset, \{(x, A)\}) \\
\mathfrak{L}_x(X_1, X_2) &:= (V_1 \cup V_2, E_1 \cup E_2, L_1 \cup L_2) \text{ where } \mathfrak{L}_x(X_i) = (V_i, E_i, L_i) \\
\mathfrak{L}_x(\circ\{X\}) &:= (V \cup \{x\}, E \cup \{(x, y)\}, L) \text{ where } \mathfrak{L}_y(X) = (V, E, L) \text{ and } y \text{ is fresh.} \\
\mathfrak{L}_x(\bullet\{X\}) &:= (V \cup \{x\}, E \cup \{(y, x)\}, L) \text{ where } \mathfrak{L}_y(X) = (V, E, L) \text{ and } y \text{ is fresh.}
\end{aligned}
$$

Example 3.6. The labeled polytree $\mathfrak{L}_x(X) := (V, E, L)$ corresponding to the nested sequent $X :=
A, \circ\{B, \bullet\{C\}\}, \bullet\{D\}$ is shown below:

---

[4]Thank you to the anonymous reviewer who suggested this definition.
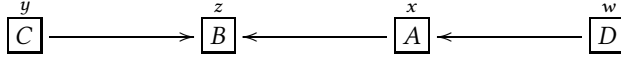
*Definition 3.7 (Labeled Polytree Merge and Subgraph).* Let $G \oplus_x H$ denote the labeled polytree obtained as the graph union of labeled polytrees G and H that have a single vertex x in common, such that the label of x in $G \oplus_x H$ (i.e. the multiset of tense formulae that x maps to under the labeling function of $G \oplus_x H$) is the union of the labels of the vertex x in G and in H. We refer to $G \oplus_x H$ as the *merge* of two polytrees.

We say that a H is a *labeled polytree subgraph* of a labeled polytree G if and only if there exists a labeled polytree H' and a vertex x in G such that $G = H' \oplus_x H$. We use $G[H]_x$ both as a name for the labeled polytree G and to denote that H is a labeled polytree subgraph of G.

Example 3.8. The labeled polytree $G[H]_x = H' \oplus_x H$, where x is the common vertex between H' and H, is shown below left. The top labeled polytree below right is H' and the other is H.
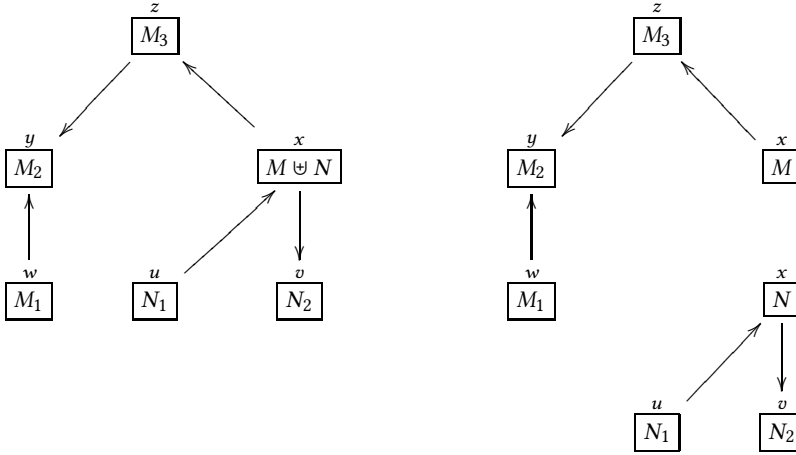
For any labeled polytree (V, E, L) with $x \in V$ there exist partitions $V = V_1 \sqcup \{x\} \sqcup V_2$, $E = E_1 \sqcup E_2$, and $L = L_1 \cup L_2$ such that $G[H]_x = H' \oplus_x H = (V, E, L)$ with $H' = (V_1 \sqcup \{x\}, E_1, L_1)$ and $H = (V_2 \sqcup \{x\}, E_2, L_2)$. Clearly, $L(x) = L_1(x) \uplus L_2(x)$, and H' and H are labeled polytrees. In other words, we view H in $G[H]_x = H' \oplus_x H$ as the redex and H' as the context.

Since nested sequents may be interpreted as trees with two types of edges ($\circ$-edges and $\bullet$-edges), they possess a root node, whereas labeled polytrees do not possess a root in general. Nevertheless, the underlying tree structure of a labeled polytree permits us to view any node as the root, and the lemma below ensures that we obtain isomorphic labeled polytrees via the display rules regardless of the node where we begin the translation.

Note that the label x in $\mathfrak{L}_x$ simply denotes the name of the starting vertex of the translation so $\mathfrak{L}_x(X) \cong \mathfrak{L}_y(X)$ for all labels x and y, and all nested sequents X.

LEMMA 3.9. *For every label x, and any nested sequents X and Y: $\mathfrak{L}_x(X, \circ\{Y\}) \cong \mathfrak{L}_x(\bullet\{X\}, Y)$.*

PROOF. Observe that $\mathfrak{L}_x(X, \circ\{Y\})$ is isomorphic to the labeled polytree obtained from the disjoint union of $\mathfrak{L}_x(X)$ and $\mathfrak{L}_y(Y)$ by the addition of an edge (x, y). Meanwhile $\mathfrak{L}_x(\bullet\{X\}, Y)$ is isomorphic to the labeled polytree obtained from the disjoint union of $\mathfrak{L}_y(X)$ and $\mathfrak{L}_x(Y)$ by the addition of an edge (y, x). The result follows because $\mathfrak{L}_x(X) \cong \mathfrak{L}_y(X)$ and $\mathfrak{L}_y(Y) \cong \mathfrak{L}_x(Y)$.   □

COROLLARY 3.10. *For all labels x and y, and nested sequents X and Y, if X and Y are display equivalent, then $\mathfrak{L}_x(X) \cong \mathfrak{L}_y(Y)$.*

PROOF. By repeated application of Lemma 3.9. □

Henceforth we write $\mathfrak{L}$ instead of $\mathfrak{L}_x$ to reduce clutter when the name of the starting vertex is not important.

3.2 Interpreting a Labeled Polytree as a Nested Sequent

Given a labeled polytree $G = (V, E, L)$ we first pick a vertex $x \in V$ to compute the nested sequent $\mathfrak{N}_x(G)$. If $E = \emptyset$, then $\mathfrak{N}_x(G) = L(x)$ is the desired nested sequent. Otherwise, for all n forward looking edges $(x, y_i) \in E$ (with $1 \leq i \leq n$) where $G[H_i]_{y_i}$, and for all k backward looking edges $(z_j, x) \in E$ (with $1 \leq j \leq k$) where $G[H'_j]_{z_j}$, we define the image of $\mathfrak{N}_x(G)$ as the nested sequent

$$L(x), \circ\{\mathfrak{N}_{y_1}(H_1)\}, \ldots, \circ\{\mathfrak{N}_{y_n}(H_n)\}, \bullet\{\mathfrak{N}_{z_1}(H'_1)\}, \ldots, \bullet\{\mathfrak{N}_{z_k}(H'_k)\}$$

Since the labeled polytrees $H_1, \ldots, H_n, H'_1, \ldots, H'_k$ are smaller than G, the recursive definition of $\mathfrak{N}$ is well-founded.

LEMMA 3.11. *For any labeled polytree $G = (V, E, L)$, and for any vertices $x, y \in V$, the nested sequent $\mathfrak{N}_x(G)$ is derivable from $\mathfrak{N}_y(G)$ via the display rules (rf) and (rp).*

PROOF. We prove the result by induction on the length of the (unique) path $dist(x, y)$ between x and y. When $dist(x, y) = 0$ we have $x = y$ and the claim holds.

Base case. Suppose that $dist(x, y) = 1$. There are two cases to consider: either there is a forward edge from x to y, or there is a backward edge from x to y. Without loss of generality, we consider only the first case. It follows that if there is a forward edge connecting x to y, then since $\mathfrak{N}_x(G)$ is of the form $X, \circ\{Y\}$, then $\mathfrak{N}_y(G) = \bullet\{X\}, Y$. It is easy to see that both sequents are display equivalent.

Inductive step. Suppose that $dist(x, y) = n + 1$. Let z represent the node one edge away from x along the $n + 1$ path to y. By the base case, $\mathfrak{N}_x(G)$ and $\mathfrak{N}_z(G)$ are display equivalent, and since the distance from z to y is n, we have that $\mathfrak{N}_z(G)$ is also display equivalent to $\mathfrak{N}_y(G)$ by the induction hypothesis. Hence, $\mathfrak{N}_x(G)$ is display equivalent to $\mathfrak{N}_y(G)$. □

When translating a labeled polytree we must choose a vertex as the starting point of our translation. The above lemma states that all nested sequents obtained from choosing a different vertex to translate from are mutually derivable from one another, i.e. they are derivable from each other by use of the display rules (rp) and (rf) only (hence, they are display equivalent). Due to this fact, we will omit the subscript when contextually permissible and simply write $\mathfrak{N}$ as the translation function.

To clarify the translation procedure, we provide an example below of the various nested sequents obtained from translating at a different initial vertex.

Example 3.12. Suppose we are given the labeled polytree $G = (V, E, L)$ where $V = \{x, y, z\}$, $E = \{(x, y), (z, x)\}$, $L(x) = \{A\}$, $L(y) = \{B, C\}$, and $L(z) = \{D\}$. A pictorial representation of the labeled polytree G is given on the left with the corresponding nested sequent translations on the right:

$$
\begin{array}{ll}
\overset{y}{\boxed{B, C}} \longleftarrow \overset{x}{\boxed{A}} \longleftarrow \overset{z}{\boxed{D}} &
\begin{aligned}
\mathfrak{N}_x(G) &= A, \circ\{B, C\}, \bullet\{D\} \\
\mathfrak{N}_y(G) &= B, C, \bullet\{A, \bullet\{D\}\} \\
\mathfrak{N}_z(G) &= D, \circ\{A, \circ\{B, C\}\}
\end{aligned}
\end{array}
$$

The following lemma ensures that the pieces X and Y of the nested sequent $\mathfrak{N}_x(G[H]_x) = \mathfrak{N}_x(H' \oplus_x H) = X, Y$ and the pieces H and H' of the labeled polytree $\mathfrak{L}_x(X, Y) = G[H]_x = H' \oplus_x H$ correctly map to each other under our translation functions.

LEMMA 3.13. *(i) For every X and Y, $\mathfrak{L}_x(X, Y)$ is the labeled polytree $G[H]_x = H' \oplus_x H$ where H' is the labeled polytree $\mathfrak{L}_x(X)$ and H is the labeled polytree $\mathfrak{L}_x(Y)$.*

(ii) For every labeled polytree $G[H]_x = H' \oplus_x H$, $\mathfrak{N}_x(G[H]_x)$ is a nested sequent of the form X, Y where $X = \mathfrak{N}_x(H')$ and $Y = \mathfrak{N}_x(H)$.

PROOF. By construction of $\mathfrak{L}$ and $\mathfrak{N}$. □

Last, we note that due to the correspondence between nested sequents and labeled polytrees, the latter are easily translated to formulas in $\mathcal{L}_{Kt}$ via the function $\mathfrak{N}$ and the function $\mathcal{I}$ from Section 2.1.

4 FROM SHALLOW NESTED TO LABELED CALCULI

We answer the following question: given a derivation $\mathcal{D}$ of A in SKT + *NestSt(GP)*, is there a derivation $\mathcal{D}'$ of $x : A$ in G3Kt + *LabSt(GP)* that is *effectively relatable* to $\mathcal{D}$? As a consequence, the structure of the output derivation is related to the structure of the input. The constraint that the new derivation be *effectively relatable* is crucial, for otherwise one could trivially relate $\mathcal{D}'$ with the derivation $\mathcal{D}$ as obtained from the following equivalences:

$$\exists \mathcal{D}(\vdash^{\mathcal{D}}_{SKT+NestSt(GP)} A) \quad \text{iff} \quad Kt + GP \vdash A \quad \text{iff} \quad \exists \mathcal{D}'(\vdash^{\mathcal{D}'}_{G3Kt+LabSt(GP)} x : A)$$

where the notation $\exists \mathcal{D}(\vdash^{\mathcal{D}}_{SKT+NestSt(GP)} A)$ and $\exists \mathcal{D}'(\vdash^{\mathcal{D}'}_{G3Kt+LabSt(GP)} x : A)$ is taken to mean that there exists a derivation $\mathcal{D}$ and a derivation $\mathcal{D}'$ such that A and $x : A$ are derivable in SKT + *NestSt(GP)* and G3Kt + *LabSt(GP)*, respectively.

The point is that the derivations in SKT + *NestSt(GP)* and G3Kt + *LabSt(GP)* obtained solely from the above equivalences would *not* be defined via local transformations. Indeed, due to the existential operators, the structure of the two derivations would not be related in any meaningful way.

4.1 Transforming a Labeled Graph $G = (V, E, L)$ into a Labeled Sequent $\mathcal{R}, \Gamma$

Define $\mathcal{R} = \{Rxy \mid (x, y) \in E\}$ and

$$\Gamma = \biguplus_{x \in V} x : L(x)$$

where $x : L(x)$ represents the multiset $L(x)$ with each formula prepended with a label x.

Example 4.1. The labeled graph $G = (V, E, L)$ where $V = \{x, y, z\}$, $E = \{(x, y), (z, x)\}$, $L(x) = \{A\}$, $L(y) = \{B\}$, and $L(z) = \{C\}$ corresponds to the labeled sequent $Rxy, Rzx, x : A, y : B, z : C$.

4.2 Transforming a Labeled Sequent $\mathcal{R}, \Gamma$ into a Labeled Graph (V, E, L)

Let V be the set of all labels occurring in $\mathcal{R}, \Gamma$. Define

$$E = \{(x, y) \mid Rxy \in \mathcal{R}\} \qquad\qquad L(x) = \{A \mid x : A \in \Gamma\}$$

Example 4.2. The labeled sequent $Rxy, Ryz, Rux, x : A, z : B, z : C, u : D$ becomes the labeled graph $G = (V, E, L)$ where $V = \{x, y, z, u\}$, $E = \{(x, y), (y, z), (u, x)\}$, $L(x) = \{A\}$, $L(y) = \emptyset$, $L(z) = \{B, C\}$ and $L(u) = \{D\}$.

The reader will observe that the translations are obtained rather directly. This is because the main difference between a labeled graph and a labeled sequent is notation. Therefore, for a given nested sequent X, we let $\mathfrak{L}(X)$ also represent the labeled sequent obtained from the labeled polytree of X. We follow this convention for the remainder of the paper and let $\mathfrak{L}(X)$ represent a labeled sequent.

Combining the previous results we obtain:

THEOREM 4.3. *Let GP be a set of general path axioms. If $\mathcal{D}$ is a derivation of X in SKT + NestSt(GP),
then there is an effective translation of $\mathcal{D}$ to a derivation $\mathcal{D}'$ of $\mathfrak{L}(X)$ in G3Kt + LabSt(GP).*

PROOF. We prove the result by induction on the height of the given derivation.

Base case. The translation of an initial sequent $Y, p, \overline{p}$ in SKT + *NestSt(GP)* gives the initial
sequent $\mathfrak{L}_x(Y), x : p, x : \overline{p}$ in G3Kt + *LabSt(GP)*, which proves the base case.

Inductive step. We show the inductive step for the rules $(\vee)$, $(\blacksquare)$, $(\Diamond)$, (rp), and (GP). We note
that weakening and contraction are admissible in $\mathcal{D}'$ due to Lemma 2.7.

$$\frac{Y, A, B}{Y, A \vee B} \; (\vee) \qquad \frac{\mathfrak{L}_x(Y), x : A, x : B}{\mathfrak{L}_x(Y), x : A \vee B} \; (\vee)$$

$$\frac{Y, \bullet\{A\}}{Y, \blacksquare A} \; (\blacksquare) \qquad \frac{\mathfrak{L}_x(Y), Ryx, y : A}{\mathfrak{L}_x(Y), x : \blacksquare A} \; (\blacksquare)$$

$$\frac{Y, \circ\{Z, A\}, \Diamond A}{Y, \circ\{Z\}, \Diamond A} \; (\Diamond) \qquad \frac{\mathfrak{L}_x(Y), \mathfrak{L}_y(Z), Rxy, x : \Diamond A, y : A}{\mathfrak{L}_x(Y), \mathfrak{L}_y(Z), Rxy, x : \Diamond A} \; (\Diamond)$$

$$\frac{Y, \bullet\{Z\}}{\circ\{Y\}, Z} \; (\text{rp}) \qquad \frac{\mathfrak{L}_y(Y, \bullet\{Z\})}{\mathfrak{L}_z(\circ\{Y\}, Z)} \; \text{Lem. 3.9}$$

$$\frac{Y, \star_{n+1}\{... \star_{n+m} \{Z\}...\}}{Y, \star_1\{... \star_n \{Z\}...\}} \; (\text{GP}) \qquad \frac{\dfrac{\mathfrak{L}_x(Y), \mathcal{R}_\Sigma xy, \mathfrak{L}_y(Z)}{\mathfrak{L}_x(Y), \mathcal{R}_\Pi xy, \mathcal{R}_\Sigma xy, \mathfrak{L}_y(Z)} \; \text{Lem. 2.7}}{\mathfrak{L}_x(Y), \mathcal{R}_\Pi xy, \mathfrak{L}_y(Z)} \; (\text{GP})$$

Because $\mathfrak{L}_y(Y, \bullet\{Z\})$ and $\mathfrak{L}_z(\circ\{Y\}, Z)$ are isomorphic, the premise and conclusion of (rp) can
be mapped to the same labeled sequent (thus, the two will be identical), and hence no rule is used
for translating (rp). In the above, this is denoted by the dotted line.   □

Example 4.4. We translate a derivation of $\blacklozenge\Diamond p \rightarrow \Diamond\blacklozenge p$ in SKT + *NestSt*($\{\blacklozenge\Diamond p \rightarrow \Diamond\blacklozenge p\}$) to
a derivation in G3Kt + *LabSt*($\{\blacklozenge\Diamond p \rightarrow \Diamond\blacklozenge p\}$).

$$\frac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\blacklozenge p, \bullet\{p, \bar{p}\}, \bullet\{\Diamond\blacklozenge p\}}{\blacklozenge p, \bullet\{\bar{p}\}, \bullet\{\Diamond\blacklozenge p\}} (\blacklozenge)}{\circ\{\blacklozenge p, \bullet\{\bar{p}\}\}, \Diamond\blacklozenge p} (\text{rp})}{\circ\{\bullet\{\bar{p}\}\}, \Diamond\blacklozenge p} (\Diamond)}{\bullet\{\circ\{\bar{p}\}\}, \Diamond\blacklozenge p} (\text{GP})}{\circ\{\bar{p}\}, \circ\{\Diamond\blacklozenge p\}} (\text{rp})}{\Box\bar{p}, \circ\{\Diamond\blacklozenge p\}} (\Box)}{\bullet\{\Box\bar{p}\}, \Diamond\blacklozenge p} (\text{rf})}{\frac{\blacksquare\Box\bar{p}, \Diamond\blacklozenge p}{\blacksquare\Box\bar{p} \vee \Diamond\blacklozenge p} (\vee)}$$
$$\cdots\cdots\cdots\cdots\cdots = $$
$$\blacklozenge\Diamond p \rightarrow \Diamond\blacklozenge p$$

$$\frac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{Rxu, Rzu, Ryx, Ryz, z : \bar{p}, x : \Diamond\blacklozenge p, u : \blacklozenge p, z : p}{Rxu, Rzu, Ryx, Ryz, z : \bar{p}, x : \Diamond\blacklozenge p, u : \blacklozenge p} (\blacklozenge)}{Rxu, Rzu, Ryx, Ryz, z : \bar{p}, x : \Diamond\blacklozenge p} (\Diamond)}{Ryx, Ryz, z : \bar{p}, x : \Diamond\blacklozenge p} (\text{GP})}{Ryx, y : \Box\bar{p}, x : \Diamond\blacklozenge p} (\Box)}{x : \blacksquare\Box\bar{p}, x : \Diamond\blacklozenge p} (\blacksquare)}{x : \blacksquare\Box\bar{p} \vee \Diamond\blacklozenge p} (\vee)}$$
$$\cdots\cdots\cdots\cdots\cdots = $$
$$\blacklozenge\Diamond p \rightarrow \Diamond\blacklozenge p$$

COROLLARY 4.5. *Let $M \subseteq \{\Pi A \rightarrow \Sigma A \mid \Pi, \Sigma \in \{\Diamond\}^*\}$ be a set of modal general path axioms.
Every derivation in the shallow nested calculus $SKT - \{(\blacksquare), (\blacklozenge)\} + NestSt(M)$ is translatable to a
derivation in the labeled calculus $G3Kt - \{(\blacksquare), (\blacklozenge)\} + LabSt(M).*

The above corollary shows that our translations also hold for the modal (non-tense) fragments
of the logics we consider. As detailed in the conclusion, this is useful since one can prove con-
servativity of the display calculi SKT $- \{(\blacksquare), (\blacklozenge)\} + NestSt(M)$ over their modal fragments by

translating derivations into G3Kt − {(■), (◆)} + *LabSt(M)* and invoking the soundness of the labeled calculus.

5 FROM LABELED TO SHALLOW NESTED CALCULI

In this section, we address the converse question: translating labeled proofs into shallow nested proofs, which will be achieved by translating through the deep nested calculi DKT + *DeepPr(P)*. In the base case for Kt when $GP = \emptyset$, i.e. for the calculus G3Kt, it is fairly straightforward to effectively translate labeled derivations into nested derivations. As will be argued in Lemma 5.2, every derivation in G3Kt which proves a labeled theorem of the form $x : A$, consists solely of labeled sequents which are translatable into nested notation. After providing the translation from G3Kt to SKT, we explain a problem that arises when attempting to translate derivations from extensions of G3Kt to extensions of SKT, and how we solve this problem for path extensions of Kt.

The central issue complicating the reverse translation from labeled to nested for general path extensions of Kt is that structural rule extensions of G3Kt allow for labeled sequents to occur in derivations that are *not* labeled polytree sequents. In other words, general path structural rules allow one to derive theorems with labeled sequents not in the domain of the translation function given in Section 3.2.[5] This complication arises since our translation is only defined for labeled polytree sequents, and not for labeled sequents in general. Nevertheless, we can overcome this obstacle by considering labeled calculi for Kt extended with propagation rules for path axioms since every derivation can be transformed into one containing only (translatable) labeled sequents, i.e. labeled polytree sequents. In Section 5.2, we explain this proof transformation procedure, followed by the translation from G3Kt + *LabSt(P)* to SKT + *NestSt(P)* that leverages DKT + *DeepPr(P)* to facilitate the translation. Note that although the translation presented here takes a detour through a deep nested calculus, a direct translation from labeled to shallow nested could be provided; still, we opt for the latter approach since it allows us to exploit results from [19] that simplify our work.

5.1 Translating the Base Calculus

We first consider the converse translation for the base calculus G3Kt.

Definition 5.1 (Labeled Polytree Sequent). A *labeled polytree sequent* is a labeled sequent whose graph is a labeled polytree.

Lemma 5.2. *Every labeled derivation in G3Kt of a labeled polytree sequent $\mathcal{R}, \Gamma$ consists solely of labeled polytree sequents.*

Proof. We argue by contradiction. Let $\mathcal{D}$ be a derivation of $\mathcal{R}, \Gamma$ in G3Kt and suppose there is a labeled sequent $\mathcal{R}', \Gamma'$ in $\mathcal{D}$ that is not a labeled polytree sequent. By definition, the underlying undirected graph of the graph of $\mathcal{R}', \Gamma'$ is not a tree. It follows that the graph of $\mathcal{R}', \Gamma'$ is not connected or contains an undirected cycle.

If the graph of $\mathcal{R}', \Gamma'$ is not connected, then by inspection of the rules of G3Kt, the graph of every sequent below $\mathcal{R}', \Gamma'$ in $\mathcal{D}$ is disconnected, including the graph of $\mathcal{R}, \Gamma$, which is a contradiction. On the other hand, if it is connected, then the graph of $\mathcal{R}', \Gamma'$ must contain an undirected cycle. Since the derivation ends with a labeled polytree sequent $\mathcal{R}, \Gamma$, it must be the case that every undirected cycle is deleted ultimately. The only rules that delete relational atoms in G3Kt are (□) and (■). However, the eigenvariable conditions in these rules are not satisfied for labels occurring in an undirected cycle, so the undirected cycle cannot be eliminated, giving a contradiction.

---

[5]Note that the translation function $\mathfrak{R}$ is only defined for labeled polytree sequents.

Hence, every sequent occurring in a G3Kt derivation of a labeled polytree sequent is a labeled polytree sequent.   □

The observation that G3Kt is complete relative to derivations consisting solely of labeled polytree sequents is useful for our translation work. Recognizing that this fact generalizes to G3Kt extended with propagation rules allows us to easily translate our labeled derivations into deep nested derivations, and then leverage Lemma 2.21 to complete the effective translation from labeled to shallow nested.

LEMMA 5.3. *Every derivation in G3Kt consisting solely of labeled polytree sequents can be effectively translated to a derivation in DKT.*

PROOF. We prove this by induction on the height of the given derivation.

Base case. The translation of an initial sequent $\mathcal{R}, x : p, x : \overline{p}, \Gamma$ in G3Kt gives an initial sequent $\mathfrak{N}(\mathcal{R}, x : p, x : \overline{p}, \Gamma) = X[p, \overline{p}]$ in DKT which proves the base case.

Inductive step. We show the inductive step for the rules $(\vee)$, $(\blacksquare)$, and $(\Diamond)$; all remaining cases are similar.

$$
\frac{\mathcal{R}, \Gamma, x : A, x : B}{\mathcal{R}, \Gamma, x : A \vee B} \ (\vee)
\qquad
\begin{array}{c}
\mathfrak{N}(\mathcal{R}, \Gamma, x : A, x : B) \\
\cdots\cdots\cdots\cdots\cdots\cdots = \\
\dfrac{X[A, B]}{X[A \vee B]} \ (\vee) \\
\cdots\cdots\cdots\cdots\cdots\cdots = \\
\mathfrak{N}(\mathcal{R}, \Gamma, x : A \vee B)
\end{array}
$$

$$
\frac{\mathcal{R}, Ryx, y : A, \Gamma}{\mathcal{R}, x : \blacksquare A, \Gamma} \ (\blacksquare)
\qquad
\begin{array}{c}
\mathfrak{N}(\mathcal{R}, Ryx, y : A, \Gamma) \\
\cdots\cdots\cdots\cdots\cdots\cdots = \\
\dfrac{X[\bullet\{A\}]}{\dfrac{X[\blacksquare A, \bullet\{A\}]}{X[\blacksquare A]}} \ \begin{array}{l}\text{Lem. 2.20}\\ (\blacksquare)\end{array} \\
\cdots\cdots\cdots\cdots\cdots = \\
\mathfrak{N}(\mathcal{R}, x : \blacksquare A, \Gamma)
\end{array}
$$

We note that for the $(\blacksquare)$ case above, we choose to translate from a label $z \neq y$ to ensure that the $(\blacksquare)$ rule may be immediately applied. We are permitted to translate from whatever label we please due to Lemma 2.20, which states the admissibility of the (rf) and (rp) rules. For example, if we consider the translation $\mathfrak{N}_y(\mathcal{R}, Ryx, y : A, \Gamma)$ starting at y, then the output nested sequent would be of the form $\circ\{X\}, A$. Applying the admissibility of (rf) yields a proof of the nested sequent $X, \bullet\{A\}$, which the $(\blacksquare)$ rule may be applied to. By restricting our translation to start translating from a label $z \neq y$ however, we ensure that the $(\blacksquare)$ may be immediately applied. Translating $(\Box)$ inferences is performed in a similar fashion.

For the $(\Diamond)$ case, there are two possible inferences in DKT depending on whether the unique undirected path from the node z (that we translate from in the premise of the last inference in the G3Kt derivation) to x encounters y (below rightmost) or not (below center). Note that below center Y stands for all formulae in Γ labeled with y, and below rightmost X stands for all formulae from Γ labeled with x.

$$
\frac{\mathcal{R}, Rxy, x : \Diamond A, y : A, \Gamma}{\mathcal{R}, Rxy, x : \Diamond A, \Gamma} \ (\Diamond)
\qquad
\begin{array}{c}
\mathfrak{N}_z(\mathcal{R}, Rxy, x : \Diamond A, y : A, \Gamma) \\
\cdots\cdots\cdots\cdots\cdots\cdots = \\
\dfrac{Z[\circ\{Y, A\}, \Diamond A]}{Z[\circ\{Y\}, \Diamond A]} \ (\Diamond_1) \\
\cdots\cdots\cdots\cdots\cdots = \\
\mathfrak{N}_z(\mathcal{R}, Rxy, x : \Diamond A, \Gamma)
\end{array}
\qquad
\begin{array}{c}
\mathfrak{N}_z(\mathcal{R}, Rxy, x : \Diamond A, y : A, \Gamma) \\
\cdots\cdots\cdots\cdots\cdots\cdots = \\
\dfrac{Z[\bullet\{X, \Diamond A\}, A]}{Z[\bullet\{X, \Diamond A\}]} \ (\Diamond_2) \\
\cdots\cdots\cdots\cdots\cdots = \\
\mathfrak{N}_z(\mathcal{R}, Rxy, x : \Diamond A, \Gamma)
\end{array}
$$

Consider the directed labeled graph associated with the premise of ($\diamond$). It contains a directed edge from x to y. Since the undirected graph is a polytree by assumption, there is a unique undirected path from z to x. If (i) y is not on this path, then the translation starting at z encounters x first. Since x to y is a forward edge (i.e. it is in the same direction as the directed edge) it will be translated as $\circ$; in this case apply the ($\diamond_1$) rule. Otherwise it must be that (ii) y is on the path. This means that the translation starting at z encounters y first. Since y to x is a backward edge (i.e. its direction is the reverse of the directed edge) it will be translated as $\bullet$; in this case apply the ($\diamond_2$) rule. □

Theorem 5.4. *Every derivation in G3Kt of a labeled polytree sequent $\mathcal{R}, \Gamma$ is effectively relatable to a derivation of $\mathfrak{N}(\mathcal{R}, \Gamma)$ in SKT.*

Proof. Let $\mathcal{D}$ be a derivation in G3Kt of a labeled polytree sequent $\mathcal{R}, \Gamma$. By Lemma 5.2, $\mathcal{D}$ consists solely of labeled polytree sequents. Hence, by Lemma 5.3 we can effectively transform $\mathcal{D}$ into a derivation $\mathcal{D}'$ in DKT, and so, by Lemma 2.21 we can effectively transform $\mathcal{D}'$ into a derivation of $\mathfrak{N}(\mathcal{R}, \Gamma)$ in SKT. The composition of these two effective transformations gives the desired effective transformation. □

The above argument does not always work for *extensions of G3Kt* because additional structural rules may be capable of removing cycles in the following sense: the underlying undirected graph of the premise might have a cycle, yet the underlying undirected graph of the conclusion might not (this was not the case for any rule in G3Kt). For example, let us consider the rule (Conf) for the confluence axiom $\blacklozenge\diamond A \rightarrow \diamond\blacklozenge A$:

$$\frac{\mathcal{R}, Rxu, Rzu, Ryx, Ryz, \Gamma}{\mathcal{R}, Ryx, Ryz, \Gamma} \text{ (Conf)}$$
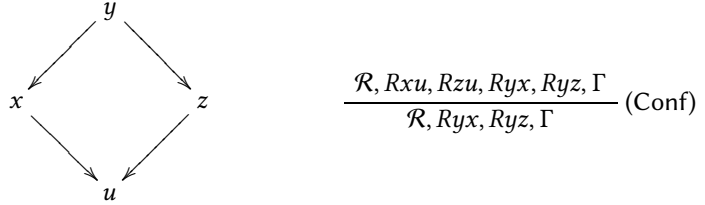
In a rule instance of (Conf), the underlying undirected graph of the premise necessarily contains a cycle of the form shown above left. However, it need not be the case that the underlying undirected graph of the conclusion contains a cycle, since the edges from x to u and z to u (corresponding to the relational atoms Rxu and Rzu) are deleted. As a consequence, a labeled derivation of a labeled formula $x : A$ in G3Kt + (Conf) may contain labeled sequents that are not labeled polytree sequents. Therefore, such a derivation is not immediately translatable to a derivation in SKT + (Conf) via our methods because the derivation may contain sequents that are not in the domain of our translation.

For all general path extensions of Kt, every shallow nested derivation can be translated into a labeled derivation; this fact implies that the space of shallow nested derivations corresponds to a subspace of the space of labeled derivations. Derivations of theorems in our labeled calculi may contain labeled sequents that are not labeled polytree sequents, showing that labeled derivations contain structures that go beyond those of the nested formalism. Nevertheless, we may invoke techniques present in [19, 27] to pre-process each labeled derivation (in a labeled calculus for Kt extended with path axioms P) in such a way that each is translatable to a shallow nested derivation, thus answering an open question in [9].

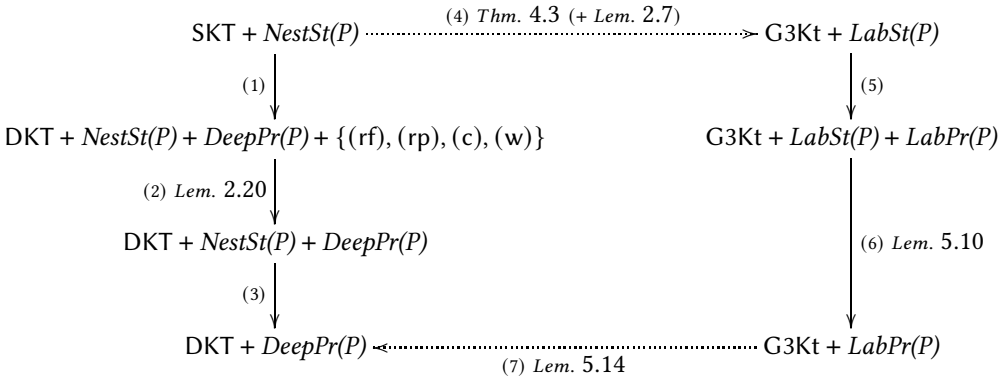## 5.2 Translating the Path Axiom Extension

We now show that the labeled calculus can be *internalized* (also referred to as *refinement* in [27]) for Kt+P (where P represents a set of path axioms), meaning that we can effectively transform any

G3Kt + *LabSt(P)* derivation of a labeled formula into one where every sequent is a labeled polytree sequent. This internalization of proofs is interesting in its own right, and is also helpful in that the resulting labeled derivation is easily translatable into a derivation in DKT + *DeepPr(P)*. From there, we can invoke Lemma 2.21 to conclude the existence of an effective translation from G3Kt+*LabSt(P)* derivations to SKT + *NestSt(P)* derivations (since composing two effective procedures gives an effective procedure).

The method of transforming every derivation in G3Kt + *LabSt(P)* into a derivation consisting solely of labeled polytree sequents relies on the addition of propagation rules *LabPr(P)* to the calculus (cf. [19, 27, 34]). Such propagation rules simulate the (Path) rules, preserve disconnected and cyclic structures downward in a derivation, and, equivalently, preserve labeled polytree structure bottom-up in a derivation. The latter properties are significant because they allow us to make an argument similar to the one made in Lemma 5.2, where we argue by contradiction that every labeled sequent occurring in a given derivation of a labeled formula $x : A$ must be a labeled polytree sequent.

The main technical lemma in this section is Lemma 5.8, where we show that in the presence of propagation rules *LabPr(P)*, the structural rules *LabSt(P)* in G3Kt + *LabSt(P)* can be eliminated from any proof. This allows for the effective transformation of any proof in an (unrestricted) labeled calculus G3Kt + *LabSt(P)* into a proof in the associated internal labeled calculus G3Kt + *LabPr(P)* (Lemma 5.10). Proofs in the internal calculi G3Kt + *LabPr(P)* can then be effectively translated into derivations in DKT + *DeepPr(P)*. Once we prove these claims, we obtain an effective translation from the labeled calculus G3Kt + *LabSt(P)* to the nested calculus SKT + *NestSt(P)* via Lemma 2.21.

The proof of admissibility of structural rules *LabSt(P)* in the presence of propagation rules *LabPr(P)* (Lemma 5.8) bears some resemblance to the proof of admissibility of structural rules *NestSt(P)* for *DeepPr(P)* in the deep nested calculi of [19]. There is, however, a crucial difference in our result compared to that of [19]. In [19], an additional admissibility result needs to be proved *for every path axiom extension*: the admissibility of all display rules. By contrast, this additional admissibility result does not need to be proved in the labeled setting—display rules are all absent in the labeled calculi. This mismatch results is an interesting observation regarding Goré *et al.*'s translation from SKT + *NestSt(P)* to DKT + *DeepPr(P)*. Consider the following transformations of a proof of a nested sequent in SKT + *NestSt(P)* to a proof of the same sequent in DKT + *DeepPr(P)*: one done directly in a nested calculus, the other through a detour in the associated labeled calculus. Note that step (3) is given by [19, Lem. 6.14] and step (5) is trivial as any derivation in G3Kt + *LabSt(P)* is a derivation in G3Kt + *LabSt(P)* + *LabPr(P)*.

$$
\begin{array}{ccc}
\text{SKT} + \textit{NestSt(P)} & \xdashrightarrow{\;(4)\ \textit{Thm. } 4.3\ (+\ \textit{Lem. } 2.7)\;} & \text{G3Kt} + \textit{LabSt(P)} \\[2pt]
{\scriptstyle(1)}\big\downarrow & & \big\downarrow{\scriptstyle(5)} \\[4pt]
\text{DKT} + \textit{NestSt(P)} + \textit{DeepPr(P)} + \{(\text{rf}), (\text{rp}), (\text{c}), (\text{w})\} & & \text{G3Kt} + \textit{LabSt(P)} + \textit{LabPr(P)} \\[2pt]
{\scriptstyle(2)\ \textit{Lem. } 2.20}\big\downarrow & & \\[4pt]
\text{DKT} + \textit{NestSt(P)} + \textit{DeepPr(P)} & & \big\downarrow{\scriptstyle(6)\ \textit{Lem. } 5.10} \\[2pt]
{\scriptstyle(3)}\big\downarrow & & \\[4pt]
\text{DKT} + \textit{DeepPr(P)} & \xdashleftarrow[\;(7)\ \textit{Lem. } 5.14\;]{} & \text{G3Kt} + \textit{LabPr(P)}
\end{array}
$$

The direct translation from SKT + *NestSt(P)* to DKT + *DeepPr(P)* in [19] is described on the left path in the above diagram; it starts with the trivial observation (1) that DKT + *NestSt(P)* + *DeepPr(P)* +

$\{(rf), (rp), (c), (w)\}$ subsumes SKT + *NestSt(P)*; followed by (2) the admissibility of display rules, contraction (c), and weakening (w); and finally, (3) the admissibility of structural rules for path axioms. The detour through labeled calculus takes care of the display rules and the (c) and (w) structural rules at step (4), where the admissibility of display rules is built into the canonical representation of nested sequents as polytrees (Corollary 3.10) and is completely independent of any extension with path axioms. This independence is not obviously observed in the transformation through the nested calculi. In fact, the designs of the propagation rules in the deep nested calculi in [19] take into account all possible interactions between display postulates and the path axioms and that leads to a proliferation of inference rules, e.g., for every propagation rule going downward in the syntax tree, there needs to be a symmetric version that propagates upward the tree. The proofs of admissibility of display rules in [19] in DKT and its extensions then need to consider all these cases, each of which is essentially the same. Moving to the labeled polytree sequent representation cuts the propagation rules by a half, and brings out the essence of a proof more clearly. These observations suggest that the syntax of the nested calculi is unnecessarily *bureaucratic* in the sense that the syntactic structures of nested sequents obscure certain identities on proofs.[6]

For another demonstration of bureaucracy of nested sequent proofs (in comparison to labeled polytree sequent proofs): take a proof $\mathcal{D}$ of the nested sequent $\circ\{\Gamma\}, \Delta$. In proving admissibility of display postulates for DKT, Gore *et al.* applied a transformation (see the proof of Lemma 4.3 in [19]) to $\mathcal{D}$ to obtain another proof $\mathcal{D}'$ of $\Gamma, \bullet\{\Delta\}$. Clearly $\mathcal{D}$ and $\mathcal{D}'$ are distinct proofs in any extension of DKT, as they have distinct end sequents. But it can be shown that they both map to the same proof in the polytree representation (i.e., by simply replacing $\diamond_1$ and $\diamond_2$ rules in DKT with $\diamond$ rule in labeled sequent calculus, and $\blacklozenge_1$ and $\blacklozenge_2$ with $\blacklozenge$). The distinction in the nested sequent proofs $\mathcal{D}$ and $\mathcal{D}'$ arises from the choice of which node in the nested sequent tree is to be designated as the root node; in the polytree representation this distinction does not arise, as there is no special node to be designated as the root node.

Let us now define the labeled propagation rules.

Definition 5.5 (Propagation Graph of a Labeled Sequent). Let $\mathcal{R}, \Gamma$ be a labeled sequent where N is the set of labels occurring in the sequent. We define the *propagation graph* $PG(\mathcal{R}, \Gamma) = (N, E)$ to be the directed graph with the set of nodes N and where E is a set of labeled edges that are labeled with either a $\diamond$ or $\blacklozenge$ as follows: For every $Rxy \in \mathcal{R}$, we have a labeled edge $(x, y, \diamond) \in E$ and a labeled edge $(y, x, \blacklozenge) \in E$. Given that $PG(\mathcal{R}, \Gamma) = (N, E)$, we will often write $x \in PG(\mathcal{R}, \Gamma)$ to mean $x \in N$, and $(x, y, \diamond) \in PG(\mathcal{R}, \Gamma)$ or $(y, x, \blacklozenge) \in PG(\mathcal{R}, \Gamma)$ to mean $(x, y, \diamond) \in E$ or $(y, x, \blacklozenge) \in E$, respectively.

Definition 5.6 (Labeled Propagation Rules). Let P be a set of path axioms. The set of propagation rules *LabPr(P)* contains all rules of the form:

$$\frac{\mathcal{R}, x : \langle?\rangle A, y : A, \Gamma}{\mathcal{R}, x : \langle?\rangle A, \Gamma} \text{ (Prop)}$$

where there is a path π from x to y in the propagation graph of the premise and $\Pi A \to \langle?\rangle A \in (P \cup I(P))^*$ with Π the string of π.[7]

We now aim to prove that we can effectively transform any derivation in G3Kt + *LabSt(P)* + *LabPr(P)* into a derivation in G3Kt + *LabPr(P)*. This inevitably yields an effective transformation from proofs in G3Kt + *LabSt(P)* to proofs in G3Kt + *LabPr(P)* (and eventually to SKT + *NestSt(P)*) in the following way: Given a derivation in G3Kt + *LabSt(P)*, we show that we can permute the topmost inference of a labeled structural rule (Path) upwards into the initial sequents to eliminate

---

[6]See e.g., [17] on the broader context of the use of the phrase "bureaucracy" in proof theory.

[7]Note that *path* and *string* are defined the same here as for nested sequents.

the use of the rule. This provides a proof in G3Kt + *LabSt(P)* + *LabPr(P)* since the *LabPr(P)* rules may be used in the permutation process to simulate the eliminated *LabSt(P)* rule. By permuting away all labeled structural rules (Path) ∈ *LabSt(P)* from the derivation, we effectively obtain a proof in G3Kt + *LabPr(P)*, which (as we will show) contains exclusively labeled polytree sequents when the end sequent is a labeled polytree sequent. The last thing that we will show in this section is how to effectively translate G3Kt + *LabPr(P)* derivations into DKt + *DeepPr(P)* derivations; this result, in conjunction with Lemma 2.21, gives the desired effective translation and result.

Lemma 5.7. *For any structural rule* (Path) *defined relative to a path axiom* $\Pi A \to \langle ? \rangle A$:

$$\frac{\mathcal{R}, \mathcal{R}_\Pi xy, R_{\langle ? \rangle} xy, \Gamma}{\mathcal{R}, \mathcal{R}_\Pi xy, \Gamma} \text{ (Path)}$$

there exists a path π *in* $PG(\mathcal{R}, \mathcal{R}_\Pi xy, R_{\langle ? \rangle} xy, \Gamma)$ *from* x *to* y *whose string is* Π *as well as a path from* x *to* y *whose string is* $\langle ? \rangle$.

Proof. Follows from the definition of (Path) and the definition of a propagation graph of a labeled sequent.   □

We now show that every structural rule (Path) can be permuted above a propagation rule (Prop). This lemma is vital for transforming any G3Kt + *LabSt(P)* derivation into a G3Kt + *LabPr(P)* derivation, as it shows that all structural rules can be eliminated from a given derivation, so long as propagation rules are properly introduced. Before moving on to prove this fact however, we recall notation that was given in the introduction to Section 2. To differentiate between different path axioms in the following lemma, and make associations of paths explicit with path axioms, we use the notation $\langle G \rangle$ and $\langle F \rangle$ (potentially annotated) to represent either a white diamond $\diamond$ or black diamond $\blacklozenge$, and forgo the use of $\langle ? \rangle$ for the scope of the following lemma.

Lemma 5.8. *Let P be a set of path axioms,* (Path) ∈ *LabSt(P),* (Prop) ∈ *LabPr(P), and* $\mathcal{R}_\Pi uv :=$ $R_{\langle G_1 \rangle} uz_1, \ldots, R_{\langle G_n \rangle} z_n v$. *Suppose we are given a derivation that ends with:*

$$\frac{\dfrac{\mathcal{R}, \mathcal{R}_\Pi uv, R_{\langle G \rangle} uv, x : \langle F \rangle A, y : A, \Gamma}{\mathcal{R}, \mathcal{R}_\Pi uv, R_{\langle G \rangle} uv, x : \langle F \rangle A, \Gamma} \text{ (Prop)}}{\mathcal{R}, \mathcal{R}_\Pi uv, x : \langle F \rangle A, \Gamma} \text{ (Path)}$$

where $\mathcal{R}_\Pi uv$ *is active in the* (Path) *inference. Then, there exists a propagation rule* (Prop)′ ∈ *LabPr(P)* *such that the* (Path) *rule may be permuted upwards followed by an instance of* (Prop)′ *to derive the same end sequent:*

$$\frac{\dfrac{\mathcal{R}, \mathcal{R}_\Pi uv, R_{\langle G \rangle} uv, x : \langle F \rangle A, y : A, \Gamma}{\mathcal{R}, \mathcal{R}_\Pi uv, x : \langle F \rangle A, y : A, \Gamma} \text{ (Path)}}{\mathcal{R}, \mathcal{R}_\Pi uv, x : \langle F \rangle A, \Gamma} \text{ (Prop)}′$$

*Note that* (Prop) *and* (Path) *may correspond to different path axioms.*

Proof. Suppose we are given a derivation ending with a (Prop) inference followed by a (Path) inference and let $\mathcal{R}' = \mathcal{R}, R_{\langle G_1 \rangle} uz_1, \ldots, R_{\langle G_n \rangle} z_n v$. Moreover, due to the application of (Prop), there exists a path π' of the form $x, \langle F_1 \rangle, \ldots, \langle F_n \rangle, y$ from x to y in $PG(\mathcal{R}', R_{\langle G \rangle} uv, x : \langle F \rangle A, y : A, \Gamma)$. In the case where the relational atom $R_{\langle G \rangle} uv$ principal in (Path) *does not* lay along the path π' used in applying (Prop), the two rules may be freely permuted since there is no interaction between the two.

We therefore assume that the relational atom $R_{\langle G \rangle} uv$ lies along the path π' from x to y. By this assumption, we know that there exists an axiom $F = \langle F_1 \rangle \cdots \langle F_m \rangle A \to \langle F \rangle A = \Pi' A \to \langle F \rangle A \in$ $(P \cup I(P))^*$, where $\Pi' = \langle F_1 \rangle \cdots \langle F_m \rangle$ is the string of the path π', corresponding to the application of (Prop). Moreover, by our assumption that (Path) deletes the relational atom $R_{\langle G \rangle} uv$ that occurs

along the path π', the structural rule (Path) corresponds to a path axiom $G = \langle G_1 \rangle \cdots \langle G_n \rangle A \to \langle G \rangle A$ where $\langle G \rangle = \langle F_i \rangle$ for some $i \in \{1, \dots, m\}$. To prove the claim we must show that there exists a path σ from x to y in $PG(\mathcal{R}', x : \langle F \rangle A, y : A, \Gamma)$ such that $\Sigma p \to \langle F \rangle p \in (P \cup I(P))^*$ with Σ the string of the path σ. We construct the path σ as follows: (i) replace each $u, \langle G \rangle, v$ in π' with $u, \langle G_1 \rangle, z_1, \dots, z_n, \langle G_n \rangle, v$, and (ii) replace each $v, \langle G \rangle^{-1}, u$ in π' with $v, \langle G_n \rangle^{-1}, z_n, \dots, z_1, \langle G_1 \rangle^{-1}, u$. Taking Σ to be the string of σ, we know that $\Sigma A \to \langle F \rangle A \in (P \cup I(P))^*$ since the operations performed in steps (i) and (ii) above correspond to compositions of the axioms G and $I(G)$ with F. Let (Prop)$'$ be the propagation rule corresponding to the path axiom $\Sigma A \to \langle F \rangle A$. Since the path σ only relies on relational atoms in $\mathcal{R}'$, the rule (Prop)$'$ may be applied after (Path). $\qquad\square$

Example 5.9. We give an example of permuting a structural rule (Path) above a propagation rule (Prop). Let $P := \{F, G\}$ with $F := \Diamond \blacklozenge A \to \Diamond A$ and $G := \blacklozenge \Diamond \Diamond A \to \blacklozenge A$, where our propagation rules are defined relative to $(P \cup I(P))^*$. Let the application of (Prop) correspond to the axiom F and the application of (Path) correspond to G. Our derivation is given below left with the propagation graph of the initial sequent below right:

$$\dfrac{\dfrac{\rule{0pt}{1.2em}}{Rxv, Rxz, Rzy, Ryv, x : \Diamond p, y : p, y : \overline{p}} \text{ (id)}}{\dfrac{Rxv, Rxz, Rzy, Ryv, x : \Diamond p, y : \overline{p}}{Rxv, Rxz, Rzy, x : \Diamond p, y : \overline{p}} \text{ (Path)}} \text{ (Prop)}$$
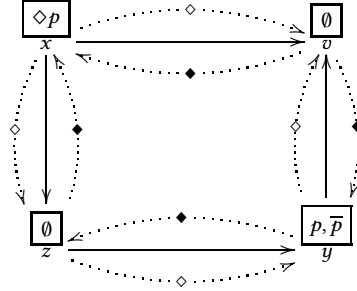
The (Prop) rule is applicable to the top sequent above because of the path $x, \Diamond, v, \blacklozenge, y$ whose string is $\Diamond \blacklozenge$, which occurs in the antecedent of F. However, we can see that the structural rule (Path) deletes the relational atom Ryv that gives rise to this path. If we were to apply the (Path) rule first (as shown below left), the conclusion would have the propagation graph shown below right:

$$\dfrac{\dfrac{\rule{0pt}{1.2em}}{Rxv, Rxz, Rzy, Ryv, x : \Diamond p, y : p, y : \overline{p}} \text{ (id)}}{Rxv, Rxz, Rzy, x : \Diamond p, y : p, y : \overline{p}} \text{ (Path)}$$
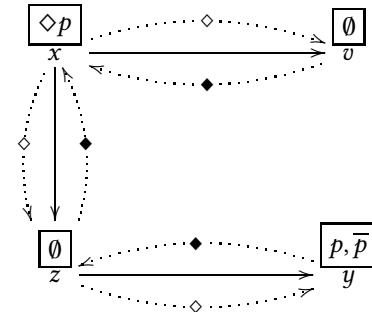
We construct a new path from x to y following the procedure explained in Lemma 5.8 by replacing $v, \blacklozenge, y$ with $v, \blacklozenge, x, \Diamond, z, \Diamond, y$ to obtain the path $x, \Diamond, v, \blacklozenge, x, \Diamond, z, \Diamond, y$. Observe that the axiom $G \triangleright^2 F = \Diamond \blacklozenge \Diamond \Diamond A \to \Diamond A$ is an element of the completion $(P \cup I(P))^*$. Thus, there exists a propagation rule (Prop)$'$ corresponding to $\Diamond \blacklozenge \Diamond \Diamond A \to \Diamond A$ which may be applied to the end sequent above to obtain the desired conclusion.

LEMMA 5.10. *Every derivation in G3Kt + LabSt(P) + LabPr(P) is effectively relatable to a derivation in G3Kt + LabPr(P).*

PROOF. We prove the result by induction on the height of the given derivation in G3Kt+*LabSt(P)*+*LabPr(P)*; we consider the topmost application of (Path) ∈ *LabSt(P)* (the general result where there are n rules of *LabSt(P)* in our derivation is immediately obtained by applying the given procedure and successively deleting the topmost occurrences). That the output derivation is effectively relatable to the input is an immediate consequence of the rule-by-rule transformations that we use.

Base case. Suppose the rule (Path) is used on an axiom in G3Kt + *LabSt(P)* + *LabPr(P)*:

$$\frac{\overline{\phantom{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, z : p, z : \overline{p}, \Gamma}} \; \text{(id)}}{\mathcal{R}, \mathcal{R}_\Pi xy, z : p, z : \overline{p}, \Gamma} \; \text{(Path)}$$

Then, it is easy to see that the conclusion is an axiom as well regardless of whether $z = x$, $z = y$, or $x \neq z \neq y$.

Inductive step. We show that (Path) ∈ *LabSt(P)* can be permuted upward with each rule in G3Kt + *LabPr(P)*:

(i) Permuting (∨) with (Path):

$$\frac{\dfrac{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, , z : A, z : B, \Gamma}{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, z : A \vee B, \Gamma} \; (\vee)}{\mathcal{R}, \mathcal{R}_\Pi xy, z : A \vee B, \Gamma} \; \text{(Path)} \qquad \frac{\dfrac{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, z : A, z : B, \Gamma}{\mathcal{R}, \mathcal{R}_\Pi xy, z : A, z : B, \Gamma} \; \text{(Path)}}{\mathcal{R}, \mathcal{R}_\Pi xy, z : A \vee B, \Gamma} \; (\vee)$$

(ii) Permuting (∧) with (Path):

$$\frac{\dfrac{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, x : A, \Gamma \qquad \mathcal{R}, \mathcal{R}_\Pi xy, Rxy, x : B, \Gamma}{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, x : A \wedge B, \Gamma} \; (\wedge)}{\mathcal{R}, \mathcal{R}_\Pi xy, x : A \wedge B, \Gamma} \; \text{(Path)}$$

$$\frac{\dfrac{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, z : A, \Gamma}{\mathcal{R}, \mathcal{R}_\Pi xy, z : A, \Gamma} \; \text{(Path)} \qquad \dfrac{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, z : B, \Gamma}{\mathcal{R}, \mathcal{R}_\Pi xy, z : B, \Gamma} \; \text{(Path)}}{\mathcal{R}, \mathcal{R}_\Pi xy, z : A \wedge B, \Gamma} \; (\wedge)$$

(iii) Permuting (■) with (Path):

$$\frac{\dfrac{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, Rvu, v : A, \Gamma}{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, u : \blacksquare A, \Gamma} \; (\blacksquare)}{\mathcal{R}, \mathcal{R}_\Pi xy, u : \blacksquare A, \Gamma} \; \text{(Path)} \qquad \frac{\dfrac{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, Rvu, v : A, \Gamma}{\mathcal{R}, \mathcal{R}_\Pi xy, Rvu, v : A, \Gamma} \; \text{(Path)}}{\mathcal{R}, \mathcal{R}_\Pi xy, u : \blacksquare A, \Gamma} \; (\blacksquare)$$

(iv) Permuting (□) with (Path):

$$\frac{\dfrac{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, Ruv, v : A, \Gamma}{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, u : \square A, \Gamma} \; (\square)}{\mathcal{R}, \mathcal{R}_\Pi xy, u : \square A, \Gamma} \; \text{(Path)} \qquad \frac{\dfrac{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, Ruv, v : A, \Gamma}{\mathcal{R}, \mathcal{R}_\Pi xy, Ruv, v : A, \Gamma} \; \text{(Path)}}{\mathcal{R}, \mathcal{R}_\Pi xy, u : \square A, \Gamma} \; (\square)$$

(v) Permuting (◆) with (Path): We consider the case where Rxy is used in both rules; the other cases are easily shown.

$$\frac{\dfrac{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, x : A, y : \blacklozenge A, \Gamma}{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, y : \blacklozenge A, \Gamma} \ (\blacklozenge)}{\mathcal{R}, \mathcal{R}_\Pi xy, y : \blacklozenge A, \Gamma} \ (\text{Path}) \qquad \frac{\dfrac{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, x : A, y : \blacklozenge A, \Gamma}{\mathcal{R}, \mathcal{R}_\Pi xy, x : A, y : \blacklozenge A, \Gamma} \ (\text{Path})}{\mathcal{R}, \mathcal{R}_\Pi xy, y : \blacklozenge A, \Gamma} \ (\text{Prop})$$

(vi) Permuting ($\diamondsuit$) with (Path): Similar to the last case we only consider when Rxy is used in both rules.

$$\frac{\dfrac{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, y : A, x : \diamondsuit A, \Gamma}{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, x : \diamondsuit A, \Gamma} \ (\diamondsuit)}{\mathcal{R}, \mathcal{R}_\Pi xy, x : \diamondsuit A, \Gamma} \ (\text{Path}) \qquad \frac{\dfrac{\mathcal{R}, \mathcal{R}_\Pi xy, Rxy, y : A, x : \diamondsuit A, \Gamma}{\mathcal{R}, \mathcal{R}_\Pi xy, y : A, x : \diamondsuit A, \Gamma} \ (\text{Path})}{\mathcal{R}, \mathcal{R}_\Pi xy, x : \diamondsuit A, \Gamma} \ (\text{Prop})$$

(vii) Permuting (Prop) with (Path): Follows from Lemma 5.8. □

The previous lemma demonstrates that propagation rules are sufficient to eliminate each structural rule (Path) from a given derivation of G3Kt + *LabSt(P)*. The consequence of this elimination procedure is that each derivation of G3Kt + *LabSt(P)* may be effectively transformed into a derivation in G3Kt + *LabPr(P)*. Our goal now is to prove that every derivation in G3Kt + *LabPr(P)* (of a labeled polytree sequent) contains nothing but labeled polytree sequents, which implies that each labeled sequent in such a derivation is in the domain of our translation function $\mathfrak{N}$. This fact proves that the effective transformation of the previous lemma has gotten us closer to completing the full translation of proofs from G3Kt + *LabSt(P)* to SKT + *NestSt(P)* since our derivations contain labeled sequents that are readily convertible to nested sequents via the function $\mathfrak{N}$.

LEMMA 5.11. *Let P be a set of path axioms. Every G3Kt + LabPr(P) proof of a labeled polytree sequent consists solely of labeled polytree sequents.*

PROOF. Similar to the proof of Lemma 5.2. Observe that all rules of G3Kt + *LabPr(P)* preserve disconnectivity and cycles downward in an inference. □

Before showing that every derivation in G3Kt + *LabPr(P)* is translatable to a derivation in DKT + *DeepPr(P)*, we must ensure that applications of propagation rules can be properly translated from the labeled to the deep nested setting. The following lemma proves that every path in a labeled polytree sequent is present in its translatee under $\mathfrak{N}$, that is, translating a labeled polytree sequent into a nested sequent preserves paths. We use this fact (Lemma 5.12) in the sequel to effectively translate instances of propagation rules.

LEMMA 5.12. *For any labeled polytree sequent $\mathcal{R}, \Gamma$ with a path π from a label x to a label y in its propagation graph, the path π exists in $PG(\mathfrak{N}_z(\mathcal{R}, \Gamma))$ from x to y (where z is an arbitrary node in $\mathcal{R}, \Gamma$).*

PROOF. Let $\mathcal{R}, \Gamma$ be a labeled polytree sequent with a path π from x to y in its propagation graph. We translate $\mathcal{R}, \Gamma$ to a nested sequent relative to the node x (as opposed to an arbitrary node z) and let the nodes in $PG(\mathfrak{N}_x(\mathcal{R}, \Gamma))$ be the same as those in the given labeled polytree sequent. Note that by Lemma 3.11, translating $\mathcal{R}, \Gamma$ relative to any label yields a display equivalent sequent, and by Lemma 2.15 the propagation graphs of all such sequents are identical. Therefore, the claim will hold regardless of the node chosen to translate from, meaning we are permitted to translate from x. We now prove the claim by induction on the length of the path connecting x and y.

Base case. The case when the path from x to y is of length 0 is trivial, so we show the case when the path from x to y is of length 1. Suppose that there is a forward edge from x to y, that is, $\pi = x, \diamondsuit, y$ (the case when there is a backward edge from x to y is similar). Then, $\mathfrak{N}_x(\mathcal{R}, \Gamma)$ will be a nested sequent with a $\circ$-edge from x to y, and so the labeled edge $(x, y, \diamondsuit)$ is in the propagation graph.

Inductive step. Suppose there is a path $x, ..., z, \langle ? \rangle, y$ from x to y of length $n+1$. Therefore, there is a path of length n from x to z, and a path of length 1 from z to y in $PG(\mathcal{R}, \Gamma)$. By the inductive hypothesis, the path from x to z occurs in $\mathfrak{N}_x(\mathcal{R}, \Gamma)$. By the base case, the path $z, \langle ? \rangle, y$ also occurs in $\mathfrak{N}_x(\mathcal{R}, \Gamma)$. Therefore, the path $x, ..., z, \langle ? \rangle, y$ is in $\mathfrak{N}_x(\mathcal{R}, \Gamma)$.   □

Lemma 5.13. *Every derivation of a sequent $\mathcal{R}, \Gamma$ in G3Kt + LabPr(P) consisting solely of labeled polytree sequents is effectively relatable to a derivation of $\mathfrak{N}(\mathcal{R}, \Gamma)$ in DKT + DeepPr(P).*

Proof. We extend the proof of Lemma 5.3 and include the inductive case for translating propagation inferences.

If we assume that a labeled propagation rule is used last in the given derivation, then there must be a corresponding axiom $\Pi p \to \langle ? \rangle p \in (P \cup I(P))^*$ whose antecedent allows for an application of the rule. This axiom will also define a deep nested propagation rule:

$$\frac{\mathcal{R}, x : \langle ? \rangle A, y : A, \Gamma}{\mathcal{R}, x : \langle ? \rangle A, \Gamma} \text{ (Prop)} \qquad \frac{X[\langle ? \rangle A]_x [A]_y}{X[\langle ? \rangle A]_x [\emptyset]_y} \text{ (Prop)}$$

By Lemma 5.12, the propagation rule may be applied in the deep nested proof because the path π from x to y (whose string is Π) exists in the propagation graph of the premise $\mathfrak{N}(\mathcal{R}, x : \langle ? \rangle A, y : A, \Gamma) = X[\langle ? \rangle A]_x [A]_y$.   □

Lemma 5.14. *Every derivation of a labeled polytree sequent $\mathcal{R}, \Gamma$ in G3Kt + LabPr(P) is effectively relatable to a derivation of $\mathfrak{N}(\mathcal{R}, \Gamma)$ in DKT + DeepPr(P).*

Proof. Let $\mathcal{D}$ be our derivation of $\mathcal{R}, \Gamma$ in G3Kt + *LabPr(P)*. By Lemma 5.11, we know that every sequent occurring in $\mathcal{D}$ will be a labeled polytree sequent. By the previous lemma, we may effectively translate this derivation into a derivation in DKT + *DeepPr(P)*.   □

Theorem 5.15. *Every derivation of a labeled polytree sequent $\mathcal{R}, \Gamma$ in G3Kt + LabSt(P) is effectively relatable to a derivation of $\mathfrak{N}(\mathcal{R}, \Gamma)$ in SKT + NestSt(P).*

Proof. By Lemma 5.10 we know that every derivation $\mathcal{D}$ of a labeled polytree sequent $\mathcal{R}, \Gamma$ in G3Kt + *LabSt(P)* is effectively transformable to a derivation $\mathcal{D}'$ of $\mathcal{R}, \Gamma$ in G3Kt + *LabPr(P)*. By Lemma 5.14, there is an effective translation of $\mathcal{D}'$ to a proof $\mathcal{D}''$ of $\mathfrak{N}(\mathcal{R}, \Gamma)$ in DKT + *DeepPr(P)*. Lemma 2.21 implies that we can effectively transform $\mathcal{D}''$ in DKT+*DeepPr(P)* into a derivation $\mathcal{D}'''$ of $\mathfrak{N}(\mathcal{R}, \Gamma)$ in SKT+*NestSt(P)*. The composition of effective procedures gives an effective procedure, which gives the result.   □

Note that the application of Lemma 2.21 in the above theorem is a rather heavy proof-theoretic transformation since it invokes cut-elimination. Nevertheless, the output derivation is still effectively relatable since cut-elimination is defined locally, at the level of the proof rules.

6 CONCLUDING REMARKS

One consequence of our work is a methodology for proving the conservativity of shallow nested (i.e. display) calculi under the deletion of certain logical rules. For example, if SKT + $NestSt(A \to \Diamond A)$ is a (sound and complete) shallow nested calculus for the logic Kt + $A \to \Diamond A$, is SKT − $\{(\blacksquare), (\blacklozenge)\} + NestSt(A \to \Diamond A)$ a (sound and complete) shallow nested calculus for K + $A \to \Diamond A$? Notice that a derivation in the latter calculus may contain a sequent with the structural connective $\bullet\{\cdot\}$ (which could be introduced via the residuation rule (rf)) or even though the corresponding logical connective $\blacksquare$ is not an operator in the ($\blacksquare, \blacklozenge$-free) language of K + $A \to \Diamond A$ (meaning that a sequent such as $\circ\{\bullet\{p\}, \bullet\{q\}\}$ cannot be interpreted as a formula). Therefore, care must be taken when attempting to identify the logic obtained under the deletion of logical rules for connectives

$\heartsuit_1, \ldots, \heartsuit_n$, since structural connectives that act as proxies for $\heartsuit_1, \ldots, \heartsuit_n$ will still be present in sequents and therefore may give the calculus increased expressive power.

A general solution which establishes the conservativity of display calculi for tense logics over their modal fragments, by making use of algebraic semantics, has been presented in [21]. Our work obtains this result *syntactically* in the context of tense logics with *modal general path axioms* by exploiting the translations developed in the previous sections (Corollary 4.5). This subsumes the conservativity results in [19], for the more restricted set of *modal path axioms*.

Another interesting consequence of our work is the suggestion of a potential methodology for constructing labeled calculi suitable for proof-search and for proving decidability of the associated logics. The labeled calculus formalism offers a uniform method for obtaining cut-, contraction-, and weakening-admissible calculi for a large class of logics [30, 34, 35]. The drawback of such calculi is that they contain structural rules which are not immediately well-suited for proof-search; if the rules are applied naïvely bottom-up, then proof-search may not terminate. Therefore, auxiliary results concerning a bound on the number of times a rule needs to be applied is required to ensure termination, see, e.g. [30]. Nevertheless, the method presented here of internalizing labeled calculi for path extensions of Kt shows that such structural rules can be eliminated from a labeled derivation in the presence of appropriate, auxiliary inference rules. This opens up an avenue for future research and gives rise to new questions: for what other logics can labeled structural rules be eliminated in favor of rules better adapted for proof-search? Is there an effective procedure for determining such rules? Note that this procedure has been investigated in [25, 27] and has shown that the method of refining labeled calculi is applicable to a variety of logics.

Moreover, the obtained internalized labeled calculi lend themselves nicely to uniformly proving interpolation for the class of path extensions of Kt [26]. As explained in Section 5.2, labeled polytrees provide a canonical representation of nested sequents that encode the polytree structure in the set $\mathcal{R}$ of relational atoms, and the decorations of the nodes as the labeled formulae in Γ. Such a representation makes it easier to define a generalized notion of interpolant, and to observe useful relationships between such interpolants (e.g. a generalized notion of duality via the (cut) rule) [26].

The relationship between Kripke frames and the algebraic semantics for modal logics is well-studied (see e.g. [2]). Because labeled calculi are based on the former, and shallow nested (display) calculi on the latter, the bi-directional translations established in this work can be interpreted as demonstrating this relationship concretely, at the level of an inference rule.

Lastly, let us briefly discuss the issues with translating labeled proofs for all general path extensions of Kt into shallow nested proofs. As explained in Section 5, when translating from labeled to shallow nested, we first pre-process our labeled derivation by eliminating all *path* structural rules from the given derivation, trading such inferences for propagation rules. This methodology does not immediately appear applicable to the class of *general path* structural rules, since it is not clear what propagation rules (if any) should be added to a labeled calculus to allow for complete structural rule elimination. Therefore, it appears that an alternative methodology is required to carry out the translation of labeled proofs for all general path extensions of Kt, which we defer to future work.

ACKNOWLEDGMENTS

REFERENCES

[1] Nuel D. Belnap, Jr. 1982. Display logic. *J. Philos. Logic* 11, 4 (1982), 375–417.

[2] Patrick Blackburn, Maarten de Rijke, and Yde Venema. 2001. *Modal Logic.* Cambridge Tracts in Theoretical Computer Science, Vol. 53. Cambridge University Press. https://doi.org/10.1017/CBO9781107050884

[3] Bianca Boretti. 2008. *Proof Analysis in Temporal Logic.* Ph.D. Dissertation. University of Milan.

[4] Bianca Boretti and Sara Negri. 2009. On the finitization of Priorean linear time. In *SILFS 2007 (Proceedings of the International Conference of the Italian Society for Logic and Philosophy of Science).* College Publications.

[5] Kai Brünnler. 2006. Deep sequent systems for modal logic. In *Advances in modal logic. Vol. 6.* Coll. Publ., London, 107–119.

[6] Robert A. Bull. 1992. Cut elimination for propositional dynamic logic without *. *Z. Math. Logik Grundlag. Math.* 38, 2 (1992), 85–100.

[7] Marcos A. Castilho, Luis Fariñas del Cerro, Olivier Gasquet, and Andreas Herzig. 1997. Modal Tableaux with Propagation Rules and Structural Rules. *Fundam. Inform.* 32, 3-4 (1997), 281–297. https://doi.org/10.3233/FI-1997-323404

[8] Alexander Chagrov and Michael Zakharyashchev. 1992. Modal companions of intermediate propositional logics. *Studia Logica* 51, 1 (1992), 49–82.

[9] Agata Ciabattoni, Tim Lyon, and Revantha Ramanayake. 2018. From Display to Labelled Proofs for Tense Logics. In *Logical Foundations of Computer Science - International Symposium, LFCS 2018, Deerfield Beach, FL, USA, January 8-11, 2018, Proceedings (Lecture Notes in Computer Science)*, Sergei N. Artëmov and Anil Nerode (Eds.), Vol. 10703. Springer, 120–139. https://doi.org/10.1007/978-3-319-72056-2_8

[10] A. Ciabattoni and R. Ramanayake. 2016. Power and Limits of Structural Display Rules. *ACM Trans. Comput. Logic* 17, 3, Article 17 (Feb. 2016), 39 pages. https://doi.org/10.1145/2874775

[11] Roy Dyckhoff and Sara Negri. 2012. Proof analysis in intermediate logics. *Arch. Math. Log.* 51, 1-2 (2012), 71–92.

[12] Roy Dyckhoff and Sara Negri. 2015. Geometrization of first-order logic. *The Bulletin of Symbolic Logic* 21 (2015), 123–163.

[13] Melvin Fitting. 1983. *Proof methods for modal and intuitionistic logics.* Synthese Library, Vol. 169. D. Reidel Publishing Co., Dordrecht. viii+555 pages.

[14] Melvin Fitting. 2012. Prefixed tableaus and nested sequents. *Ann. Pure Appl. Logic* 163, 3 (2012), 291–313.

[15] Dov M Gabbay. 1996. *Labelled deductive systems.* Oxford Logic guides, Vol. 33. Clarendon Press/Oxford Science Publications.

[16] James W. Garson. 2013. *Modal Logic for Philosophers* (2 ed.). Cambridge University Press. https://doi.org/10.1017/CBO9781139342117

[17] Jean-Yves Girard. 1989. Towards a Geometry of Interaction. In *Categories in Computer Science and Logic.* AMS, 69–108.

[18] Rajeev Goré. 1998. Substructural logics on display. *Log. J. IGPL* 6, 3 (1998), 451–504.

[19] Rajeev Goré, Linda Postniece, and Alwen Tiu. 2011. On the correspondence between display postulates and deep inference in nested sequent calculi for tense logics. *Log. Methods Comput. Sci.* 7, 2 (2011), 2:8, 38.

[20] Rajeev Goré and Revantha Ramanayake. 2012. Labelled tree sequents, Tree hypersequents and Nested (Deep) Sequents. In *Advances in modal logic, Vol. 9.* College Publications.

[21] Giuseppe Greco, Minghui Ma, Alessandra Palmigiano, Apostolos Tzimoulis, and Zhiguang Zhao. 2016. Unified correspondence as a proof-theoretic tool. *Journal of Logic and Computation* (2016).

[22] Juan F Huete and Luis M de Campos. 1993. Learning causal polytrees. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty.* Springer, 180–185.

[23] Ryo Kashima. 1994. Cut-free sequent calculi for some tense logics. *Studia Logica* 53, 1 (1994), 119–135.

[24] Marcus Kracht. 1996. Power and weakness of the modal display calculus. In *Proof theory of modal logic (Hamburg, 1993).* Appl. Log. Ser., Vol. 2. Kluwer Acad. Publ., Dordrecht, 93–121.

[25] Tim Lyon. 2020. On the correspondence between nested calculi and semantic systems for intuitionistic logics. *Journal of Logic and Computation* 31, 1 (12 2020), 213–265. https://doi.org/10.1093/logcom/exaa078 arXiv:https://academic.oup.com/logcom/article-pdf/31/1/213/36677523/exaa078.pdf

[26] Tim Lyon, Alwen Tiu, Rajeev Goré, and Ranald Clouston. 2020. Syntactic Interpolation for Tense Logics and Bi-Intuitionistic Logic via Nested Sequents. In *28th EACSL Annual Conference on Computer Science Logic, CSL 2020, January 13-16, 2020, Barcelona, Spain (LIPIcs)*, Maribel Fernández and Anca Muscholl (Eds.), Vol. 152. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 28:1–28:16. https://doi.org/10.4230/LIPIcs.CSL.2020.28

[27] Tim Lyon and Kees van Berkel. 2019. Automating Agential Reasoning: Proof-Calculi and Syntactic Decidability for STIT Logics. In *PRIMA 2019: Principles and Practice of Multi-Agent Systems - 22nd International Conference, Turin, Italy,*

October 28-31, 2019, Proceedings (Lecture Notes in Computer Science), Matteo Baldoni, Mehdi Dastani, Beishui Liao, Yuko Sakurai, and Rym Zalila Wenkstern (Eds.), Vol. 11873. Springer International Publishing, Cham, 202–218.

[28] S. Marin and L. Strassburger. 2014. Label-free Modular Systems for Classical and Intuitionistic Modal Logics. In *AiML 2014 (LNCS)*. Springer.

[29] Grigori Mints. 1997. Indexed systems of sequents and cut-elimination. *J. Philos. Logic* 26, 6 (1997), 671–696.

[30] Sara Negri. 2005. Proof analysis in modal logic. *J. Philos. Logic* 34, 5-6 (2005), 507–544.

[31] Elaine Pimentel. 2018. A Semantical View of Proof Systems. In *Logic, Language, Information, and Computation*, Lawrence S. Moss, Ruy de Queiroz, and Maricarmen Martinez (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 61–76.

[32] Francesca Poggiolesi. 2010. Display calculi and other modal calculi: a comparison. *Synthese* 173, 3 (01 Apr 2010), 259–279. https://doi.org/10.1007/s11229-008-9425-4

[33] George Rebane and Judea Pearl. 2013. The recovery of causal poly-trees from statistical data. *arXiv preprint arXiv:1304.2736* (2013).

[34] Alex Simpson. 1994. *The Proof Theory and Semantics of Intuitionistic Modal Logic*. Ph.D. Dissertation. University of Edinburgh.

[35] Luca Viganò. 2000. *Labelled non-classical logics*. Kluwer Academic Publishers, Dordrecht. xiv+291 pages. With a foreword by Dov M. Gabbay.

[36] Heinrich Wansing. 1998. *Displaying modal logic*. Trends in Logic—Studia Logica Library, Vol. 3. Kluwer Academic Publishers, Dordrecht. xii+252 pages.