Proof Search in Nested Sequent Calculi

Björn Lellmann[1]⋆ and Elaine Pimentel[2]⋆⋆

[1] Department of Computer Languages, TU Wien, Austria
[2] Department of Mathematics, UFRN, Brazil

Abstract. We propose a notion of focusing for nested sequent calculi for modal logics which brings down the complexity of proof search to that of the corresponding sequent calculi. The resulting systems are amenable to specifications in linear logic. Examples include modal logic K, a simply dependent bimodal logic and the standard non-normal modal logics. As byproduct we obtain the first nested sequent calculi for the considered non-normal modal logics.

1 Introduction

A main concern in proof theory for modal logics is the development of philosophically and, at the same time, computationally satisfying frameworks to capture large classes of logics in a uniform and systematic way. Unfortunately the standard sequent framework satisfies these desiderata only partly. Undoubtedly, there are sequent calculi for a number of modal logics exhibiting many good properties (such as analyticity), which can be used in complexity-optimal decision procedures. However, their construction often seems ad-hoc, they are usually not modular, and they mostly lack philosophically relevant properties such as separate left and right introduction rules for the modalities. These problems are often connected to the fact that the modal rules in such calculi usually introduce more than one connective at a time. For example, in the rule

$$\frac{\Gamma \vdash A}{\Gamma', \Box\Gamma \vdash \Box A, \Delta}\ \mathsf{k}$$

for modal logic K [3], the context Γ could contain an arbitrary finite number of formulae. Hence this rule can also be seen as an infinite set of rules

$$\left\{ \frac{B_1, \dots, B_n \vdash A}{\Gamma', \Box B_1, \dots, \Box B_n \vdash \Box A, \Delta}\ \mathsf{k}_n \mid n \geq 0 \right\}$$

each with a fixed number of principal formulae. Both of these perspectives are somewhat dissatisfying: the first since it necessitates modifying the context, and the second since it explicitly discards the distinction between left and right rules for the modal connective.

One way to approach this problem is to consider extensions of the sequent framework that are expressive enough for capturing these modalities using separate left and right introduction rules. This is possible e.g. in the frameworks of *labelled sequents* [11]

⋆ Funded by the EU under Marie Skłodowska-Curie grant agreement No. 660047
⋆⋆ Funded by CNPq

or in that of *nested sequents* or *tree-hypersequents* [2,14,15]. Intuitively, in the latter framework a single sequent is replaced with a tree of sequents, where successors of a sequent are interpreted under a modality. The modal rules of these calculi govern the transfer of (modal) formulae between the different sequents, and it can be shown that it is sufficient to transfer only one formula at a time. However, the price to pay for this added expressivity is that the obvious proof search procedure is of suboptimal complexity since it constructs potentially exponentially large nested sequents [2].

In this work, we reconcile the added superior expressiveness and modularity of nested sequents with the computational behaviour of the standard sequent framework by proposing a *focusing discipline* for *linear nested sequents* [7], a restricted form of nested sequents where the tree-structure is restricted to that of a line. The result is a notion of normal derivations in the linear nested setting, which directly correspond to derivations in the standard sequent setting. Moreover, the resulting calculi lend themselves to specification and implementation in linear logic following the approach in [10]. Since we are interested in the connections to the standard sequent framework, we concentrate on logics which have a standard sequent calculus, with examples including normal modal logic K and simple extensions, the exemplary *simply dependent bimodal logic* $\mathsf{KT} \oplus_{\subseteq} \mathsf{S4}$ [4], but also several non-normal modal logics, i.e., standard extensions of *classical modal logic* [3]. As a side effect, we obtain to the best of our knowledge the first nested sequent calculi for all the considered non-normal modal logics.

2 Linear nested sequent systems

We briefly recall the basic notions of the linear nested sequent framework [7], essentially a reformulation of Masini's 2-sequents [9] in the nested sequent framework. In the following we consider a *sequent* to be a pair $\Gamma \vdash \Delta$ of multisets and adopt the standard conventions and notations (see e.g. [11]). In the linear nested sequent framework, the tree structure of nested sequents is restricted to a line, i.e., a linear nested sequent is simply a finite list of sequents. This data structure matches exactly the *history* in a backwards proof search in an ordinary sequent calculus, a fact we will heavily use in what follows.

Definition 1. *The set* LNS *of* linear nested sequents *is given recursively by:*

1. *if $\Gamma \vdash \Delta$ is a sequent then $\Gamma \vdash \Delta \in \mathsf{LNS}$*
2. *if $\Gamma \vdash \Delta$ is a sequent and $\mathcal{G} \in \mathsf{LNS}$ then $\Gamma \vdash \Delta \mathbin{/\!/} \mathcal{G} \in \mathsf{LNS}$.*

We will write $\mathcal{S}\{\Gamma \vdash \Delta\}$ for denoting a context *$\mathcal{G} \mathbin{/\!/} \Gamma \vdash \Delta \mathbin{/\!/} \mathcal{H}$ where $\mathcal{G}, \mathcal{H} \in \mathsf{LNS}$ or $\mathcal{G}, \mathcal{H} = \emptyset$. We call each sequent in a linear nested sequent a* component *and slightly abuse notation and abbreviate "linear nested sequent" to* LNS.

In this work we consider only modal logics based on classical propositional logic, and we take the system $\mathsf{LNS_G}$ (Fig. 1) as our base calculus. Note that the initial sequents are atomic, contraction, weakening and cut are admissible and all rules are invertible.

Fig. 2 presents the modal rules for the linear nested sequent calculus $\mathsf{LNS_K}$ for K, essentially a linear version of the standard nested sequent calculus from [2,14]. Conceptually, the main point is that the sequent rule k is split into the two rules $\Box_L$ and $\Box_R$, which permit to simulate the sequent rule treating one formula at a time. Completeness of

$$\frac{}{\mathcal{S}\{\Gamma, p \vdash p, \Delta\}}\ \mathsf{init} \qquad \frac{\mathcal{S}\{\Gamma, A, B \vdash \Delta\}}{\mathcal{S}\{\Gamma, A \wedge B \vdash \Delta\}}\ \wedge_L \qquad \frac{\mathcal{S}\{\Gamma \vdash A, \Delta\} \quad \mathcal{S}\{\Gamma \vdash B, \Delta\}}{\mathcal{S}\{\Gamma \vdash A \wedge B, \Delta\}}\ \wedge_R$$

$$\frac{}{\mathcal{S}\{\Gamma, \bot \vdash \Delta\}}\ \bot_L \qquad \frac{\mathcal{S}\{\Gamma, B \vdash \Delta\} \quad \mathcal{S}\{\Gamma \vdash A, \Delta\}}{\mathcal{S}\{\Gamma, A \supset B \vdash \Delta\}}\ \supset_L \qquad \frac{\mathcal{S}\{\Gamma, A \vdash B, \Delta\}}{\mathcal{S}\{\Gamma \vdash A \supset B, \Delta\}}\ \supset_R$$

Fig. 1. System $\mathsf{LNS_G}$ for classical propositional logic. In the init rule, p is atomic.

$$\frac{\mathcal{S}\{\Gamma \vdash \Delta \mathbin{/\!\!/} \Sigma, A \vdash \Pi\}}{\mathcal{S}\{\Gamma, \Box A \vdash \Delta \mathbin{/\!\!/} \Sigma \vdash \Pi\}}\ \Box_L \qquad \frac{\mathcal{G} \mathbin{/\!\!/} \Gamma \vdash \Delta \mathbin{/\!\!/} \ \vdash A}{\mathcal{G} \mathbin{/\!\!/} \Gamma \vdash \Delta, \Box A}\ \Box_R$$

Fig. 2. The modal rules of the linear nested sequent calculus $\mathsf{LNS_K}$ for K.

$\mathsf{LNS_K}$ w.r.t. modal logic K is shown by simulating a sequent derivation bottom-up in the last two components of the linear nested sequents, marking applications of transitional rules by the nesting $/\!\!/$ and simulating the k-rule by a block of $\Box_L$ and $\Box_R$ rules [7]. E.g., an application of k on a branch with history captured by the LNS $\mathcal{G}$ is simulated by:

$$\begin{array}{c}\dfrac{\Gamma \vdash A}{\Gamma', \Box\Gamma \vdash \Box A, \Delta}\ \mathbf{k} \\ \vdots\ \mathcal{G}\end{array} \quad \rightsquigarrow \quad \dfrac{\dfrac{\mathcal{G} \mathbin{/\!\!/} \Gamma' \vdash \Delta \mathbin{/\!\!/} \Gamma \vdash A}{\mathcal{G} \mathbin{/\!\!/} \Gamma', \Box\Gamma \vdash \Delta \mathbin{/\!\!/} \vdash A}\ \Box_L}{\mathcal{G} \mathbin{/\!\!/} \Gamma', \Box\Gamma \vdash \Box A, \Delta}\ \Box_R$$

where the double line indicates multiple rule applications. Observe that this method relies on the view of linear nested sequents as histories in proof search. It also simulates the propositional sequent rules in the *rightmost* component of the linear nested sequents. This gives a different way of looking at system K, where formulas in the context can be handled separately. However, the modal rules do not need to occur in a block corresponding to one application of the sequent rule anymore. For instance, one way of deriving the instance $\Box(p \supset q) \supset (\Box p \supset \Box q)$ of the normality axiom for modal logic K is as follows.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Box p \vdash \mathbin{/\!\!/} q \vdash q}\ \mathsf{init} \quad \dfrac{\dfrac{}{\vdash \mathbin{/\!\!/} p \vdash p, q}\ \mathsf{init}}{\Box p \vdash \mathbin{/\!\!/} \ \vdash p, q}\ \Box_L}{\Box p \vdash \mathbin{/\!\!/} p \supset q \vdash q}\ \supset_L}{\Box(p \supset q), \Box p \vdash \Box q}\ \Box_R, \Box_L}{\vdash \Box(p \supset q) \supset (\Box p \supset \Box q)}\ \supset_R$$

Note that the propositional rule $\supset_L$ is applied between two modal rules. Hence there are many derivations in $\mathsf{LNS_K}$ which are not the result of simulating a derivation of the sequent calculus for K. Thus, while the linear nested sequent calculus $\mathsf{LNS_K}$ has conceptual advantages over the standard sequent calculus for K, its behaviour in terms of proof search is worse: there are many more possible derivations with the same conclusion, when compared to the sequent calculus. We will address this issue by proposing a *focusing discipline* [1] to restrict proof search to a smaller class of derivations, while retaining the conceptual advantages of the framework.

$$\frac{}{\mathcal{R}, X, x : p \vdash x : p, Y}\ \mathsf{init} \qquad \frac{\mathcal{R}, X, x : A, x : B \vdash Y}{\mathcal{R}, X, x : A \wedge B \vdash Y}\ \wedge_L \qquad \frac{\mathcal{R}, X \vdash x : A, Y \quad \mathcal{R}, X \vdash x : B, Y}{\mathcal{R}, X \vdash x : A \wedge B, Y}\ \wedge_R$$

$$\frac{}{\mathcal{R}, X, x : \bot \vdash Y}\ \bot_L \qquad \frac{\mathcal{R}, X \vdash Y, x : A \quad \mathcal{R}, X, x : B \vdash Y}{\mathcal{R}, X, x : A \supset B \vdash Y}\ \supset_L \qquad \frac{\mathcal{R}, X, x : A \vdash Y, x : B}{\mathcal{R}, X \vdash Y, x : A \supset B}\ \supset_R$$

Fig. 3. Labelled line sequent calculus $\mathsf{LLS_G}$.

3 Labelled line sequent systems

For simplifying the notation of the focused systems and also for encoding linear nested sequent calculi in linear logic (see Section 6), we follow the correspondence between nested sequents and labelled tree sequents given in [5], and consider the *labelled sequents* [11] corresponding to linear nested sequents. Intuitively, the components of a LNS are labelled with variables and their order is encoded in a relation.

More formally, a (possibly empty) set of relations terms (*i.e.* terms of the form xRy) is called a *relation set*. For a relation set $\mathcal{R}$, the *frame* $Fr(\mathcal{R})$ *defined by* $\mathcal{R}$ is given by $(|\mathcal{R}|, \mathcal{R})$ where $|\mathcal{R}| = \{x \mid xRv \in \mathcal{R}$ or $vRx \in \mathcal{R}$ for some state $v\}$. We say that a relation set $\mathcal{R}$ is *treelike* if the frame defined by $\mathcal{R}$ is a tree or $\mathcal{R}$ is empty. A treelike relation set $\mathcal{R}$ is called *linelike* if each node in $\mathcal{R}$ has at most one child.

Definition 2. *A* labelled line sequent LLS *is a labelled sequent* $\mathcal{R}, X \vdash Y$ *where*

1. *$\mathcal{R}$ is linelike;*
2. *if $\mathcal{R} = \emptyset$ then X has the form $x : A_1, \ldots, x : A_n$ and Y has the form $x : B_1, \ldots, x : B_m$ for some state variable x;*
3. *if $\mathcal{R} \neq \emptyset$ then every state variable x that occurs in either X or Y also occurs in $\mathcal{R}$.*

Observe that, in LLS, if $xRy \in \mathcal{R}$ then $uRy \notin \mathcal{R}$ and $xRv \notin \mathcal{R}$ for any $u \neq x$ and $v \neq y$.

Definition 3. *A* labelled line sequent calculus *is a labelled sequent calculus whose initial sequents and inference rules are constructed from* LLS.

In Fig. 3 we present the rules for the labelled line classical calculus $\mathsf{LLS_G}$.

Since linear nested sequents form a particular case of nested sequents, the algorithm given in [5] can be used for generating LLS from LNS, and vice versa. However, one has to keep the linearity property invariant through inference rules. For example, the rule

$$\frac{\mathcal{R}, xRy, X \vdash Y, y : A}{\mathcal{R}, X, \vdash Y, x : \Box A}\ \Box_R'$$

where y is fresh, considered more generally as a *labelled* sequent rule is not adequate w.r.t. the system $\mathsf{LNS_K}$, since there may exist $z \in |\mathcal{R}|$ such that $xRz \in \mathcal{R}$. That is, for labelled sequents in general, freshness alone is not enough for guaranteeing unicity of x in $\mathcal{R}$. And it does not seem to be trivial to assure this unicity by using logical rules without side conditions. To avoid this problem, we slightly modify the framework by restricting $\mathcal{R}$ to singletons, that is, $\mathcal{R} = \{xRy\}$ will record only the two last components, in

$$\frac{}{zRx, X, x : p \vdash x : p, Y}\ \mathsf{init} \quad \frac{zRx, X, x : A, x : B \vdash Y}{zRx, X, x : A \wedge B \vdash Y}\ \wedge_L \quad \frac{zRx, X \vdash x : A, Y \quad zRx, X \vdash x : B, Y}{zRx, X \vdash x : A \wedge B, Y}\ \wedge_R$$

$$\frac{}{zRx, X, x : \bot \vdash Y}\ \bot_L \quad \frac{zRx, X \vdash Y, x : A \quad zRx, X, x : B \vdash Y}{zRx, X, x : A \supset B \vdash Y}\ \supset_L \quad \frac{zRx, X, x : A \vdash Y, x : B}{zRx, X \vdash Y, x : A \supset B}\ \supset_R$$

Fig. 4. The end-active version of $\mathsf{LLS_G}$. In rule init, p is atomic.

this case labelled by x and y, and by adding a base case $\mathcal{R} = \{y_0 R x_0\}$ for x_0, y_0 different state variables when there are no nested components. The rule for introducing $\Box_R$ then is

$$\frac{xRy, X \vdash Y, y : A}{zRx, X, \vdash Y, x : \Box A}\ \Box_R$$

with y fresh. Note that this solution corresponds to recording the history of the proof search up to the last two steps. We adopt the following terminology for calculi where this restriction is possible.

Definition 4. *A* LNS *calculus is* end-active *if in all its rules the rightmost components of the premisses are active and the only active components (in premisses and conclusion) are the two rightmost ones. An* end-active LLS *is a singleton relation set* $\mathcal{R}$ *together with a sequent* $X \vdash Y$ *of labelled formulae, written* $\mathcal{R}, X \vdash Y$*. The rules of an* end-active LLS calculus *are constructed from end-active labelled line sequents such that the active formulae in a premiss* $xRy, X \vdash Y$ *are labelled with* y *and the labels of all active formulae in the conclusion are in its relation set.*

Observe that the completeness proof for $\mathsf{LNS_K}$ via simulating a sequent derivation in the last component actually shows that the end-active version of the calculus $\mathsf{LNS_K}$ is complete for K [7]. From now on, we will use the end-active version of the propositional rules (see Fig. 4). Note that, in an end-active LLS, state variables might occur in the sequent and not in the relation set. Such formulae will remain inactive towards the leaves of the derivation. In fact, a key property of end-active LNS calculi is that rules can only move formulas "forward", that is, either an active formula produces other formulae in the same component or in the next one. Hence one can automatically generate LLS from LNS. In the following we write $x : \Gamma$ if the label of every labelled formula in Γ is x.

Definition 5. *For a state variable x, define the mapping* $\mathbb{TL}_x$ *from* LNS *to end-active* LLS *as follows*

$$\begin{array}{ll} \mathbb{TL}_{x_0}(\Gamma_0 \vdash \Delta_0) & = y_0 R x_0, x_0 : \Gamma_0 \vdash x_0 : \Delta_0 \\ \mathbb{TL}_{x_n}(\Gamma_0 \vdash \Delta_0 /\!\!/ \ldots /\!\!/ \Gamma_n \vdash \Delta_n) & = x_{n-1} R x_n, x_0 : \Gamma_0, \ldots, x_n : \Gamma_n \vdash x_0 : \Delta_0, \ldots, x_n : \Delta_n \quad n > 0 \end{array}$$

with all state variables pairwise distinct.

It is straightforward to use $\mathbb{TL}_x$ in order to construct a LLS inference rule from an inference rule of an end-active LNS calculus. The procedure, that can be automatised, is the same as the one presented in [5], as we shall illustrate it here.

$$\frac{xRy, X, y : A \vdash Y}{xRy, X, x : \Box A \vdash Y}\ \Box_L \qquad \frac{xRy, X \vdash Y, y : A}{zRx, X \vdash Y, x : \Box A}\ \Box_R \ \text{(y is a fresh variable)}$$

Fig. 5. The modal rules of $\mathsf{LLS_K}$.

Example 6. Consider the following application of the rule $\Box_R$ of Fig. 2:

$$\frac{\Gamma_0 \vdash \Delta_0 /\!\!/ \ldots /\!\!/ \Gamma_{n-1} \vdash \Delta_{n-1} /\!\!/ \Gamma_n \vdash \Delta_n /\!\!/ \ \vdash A}{\Gamma_0 \vdash \Delta_0 /\!\!/ \ldots /\!\!/ \Gamma_{n-1} \vdash \Delta_{n-1} /\!\!/ \Gamma_n \vdash \Delta_n, \Box A}\ \Box_R$$

Applying $\mathbb{TL}_x$ to the conclusion we obtain $x_{n-1}Rx_n, X \vdash Y, x_n : \Box A$, where $X = x_1 : \Gamma_1, \ldots, x_n : \Gamma_n$ and $Y = x_1 : \Delta_1, \ldots, x_n : \Delta_n$. Applying $\mathbb{TL}_x$ to the premise we obtain $x_nRx_{n+1}, X \vdash Y, x_{n+1} : A$. We thus obtain an application of the LLS rule

$$\frac{x_nRx_{n+1}, X \vdash Y, x_{n+1} : A}{x_{n-1}Rx_n, X \vdash Y, x_n : \Box A}\ \mathbb{TL}_x(\Box_R)$$

which is the rule $\Box_R$ presented in Fig. 5.

The following result follows readily by transforming derivations bottom-up.

Theorem 7. *$\Gamma \vdash \Delta$ is provable in a certain end-active LNS calculus if and only if $\mathbb{TL}_{x_0}(\Gamma \vdash \Delta)$ is provable in the corresponding end-active LLS calculus.* □

The end-active labelled line sequent calculus $\mathsf{LLS_K}$ for K is given in Fig. 5. The following is immediate from completeness of the end-active version of $\mathsf{LNS_K}$.

Corollary 8. *A sequent $\Gamma \vdash \Delta$ has a proof in $\mathsf{LNS_K}$ if and only if $yRx, x : \Gamma \vdash x : \Delta$ has a proof in $\mathsf{LLS_K}$ for some different state variables x, y.* □

4 Focused labelled line sequent systems

Although adding labels and restricting systems to their end-active form enhance proof search a little, this is still not enough for guaranteeing that modal rules occur in a block.

In [1], Andreoli introduced a notion of normal form for cut-free proofs in linear logic. This normal form is given by a *focused* proof system organised around two "phases" of proof construction: the *negative* phase for invertible inference rules and the *positive* phase for non-necessarily-invertible inference rules. Observe that a similar organisation is adopted when moving from $\mathsf{LNS_K}$ to $\mathsf{LLS_K}$: invertible rules are done eagerly while the non invertible ones ($\Box_R + \Box_L$) are done only in the last two components.

We will now define $\mathsf{FLLS_K}$, a focused system for $\mathsf{LLS_K}$. Depending on the focusing, the sequents manipulated in $\mathsf{FLLS_K}$ have one of the following three shapes:

1. $zRx : \Gamma; X \Rightarrow Y; \Delta$ is an unfocused sequent, where Γ contains only modal formulae and Δ contains only modal or atomic formulae.
2. $zR[x] : \Gamma; X \rightarrow \cdot; \Delta$ is a sequent focused on a boxed or atomic formula.
3. $[x]Ry : \Gamma; X \rightarrow Y; \Delta$ is a sequent focused on a boxed formula.

$$\frac{}{zRx : \Gamma; X, x : \bot \Rightarrow Y; \Delta}\ \bot_L \qquad \frac{zRx : \Gamma; X, x : A, x : B \Rightarrow Y; \Delta}{zRx : \Gamma; X, x : A \wedge B \Rightarrow Y; \Delta}\ \wedge_L$$

$$\frac{zRx : \Gamma; X \Rightarrow x : A, Y; \Delta \quad zRx : \Gamma; X \Rightarrow x : B, Y; \Delta}{zRx : \Gamma; X \Rightarrow x : A \wedge B, Y; \Delta}\ \wedge_R$$

$$\frac{zRx : \Gamma, x : B_b; X \Rightarrow Y; \Delta}{zRx : \Gamma; X, x : B_b \Rightarrow Y; \Delta}\ \mathtt{store}_L \qquad \frac{zRx : \Gamma; X \Rightarrow Y; \Delta, x : A_b}{zRx : \Gamma; X \Rightarrow Y, x : A_b; \Delta}\ \mathtt{store}_R$$

$$\frac{}{zR[x] : \Gamma; X, x : A \rightarrow \cdot; \Delta, x : A}\ \mathsf{init} \quad \frac{zR[x] : \Gamma; X \rightarrow \cdot; \Delta}{zRx : \Gamma; X \Rightarrow \cdot; \Delta}\ D \quad \frac{xRy : \cdot; X \Rightarrow Y; \Delta}{[x]Ry : \cdot; X \rightarrow Y; \Delta}\ R$$

$$\frac{[x]Ry : \Gamma; X \rightarrow y : A; \Delta}{zR[x] : \Gamma; X \rightarrow \cdot; \Delta, x : \Box A}\ \Box_R \qquad \frac{[x]Ry : \Gamma; X, y : A \rightarrow Y; \Delta}{[x]Ry : \Gamma, x : \Box A; X \rightarrow Y; \Delta}\ \Box_L$$

Fig. 6. Some rules of the focused labelled line sequent calculus $\mathsf{FLLS_K}$ for K. A_b is atomic or a boxed formula, B_b is a boxed formula.

In the *negative phase*, sequents have the shape (1) above and all invertible propositional or modal rules are applied eagerly on formulae labelled with the variable x until there are only atomic or boxed formulae left. Some of those are moved to special contexts Γ, Δ using `store` rules. These contexts store the formulae that can be chosen for focusing. When this process terminates, the *positive phase* starts by deciding on one of the formulae in Δ, indicated by a sequent of the form (2). If this formula is an atom, then the proof should terminate. Otherwise, the focusing is over a modal formula, and the rule $\Box_R$ creates a fresh label y and moves the unboxed part of the formula to this new label, resulting in a sequent of the form (3). The positive phase then continues by possibly moving boxed formulae in Γ, labelled with x, to the label y. Finally, focusing is lost and we come back to the negative phase, now inside the component labelled by y.

The rules for $\mathsf{FLLS_K}$ are presented in Figure 6. Note that the rule $\mathtt{store}_R$ systematically moves all atomic and boxed formulae from Y to Δ, and hence Y will be eventually empty. This is the trigger for switching from the negative to the positive phase. Note also that the contexts may carry some "garbage", i.e., formulae which will never be principal. In fact, since the calculus is end-active, only formulae in one of the two last components can be principal. Similar to standard systems where weakening is admissible, these formulae are then absorbed by the initial sequents init. Since the focusing procedure described above is just a systematic organisation of proofs, soundness and completeness proofs are often straightforward permutation-of-rules arguments.

Theorem 9. *The system* $\mathsf{FLLS_K}$ *is sound and complete w.r.t. modal logic* K*, i.e., a formula A is a theorem of* K *iff the sequent* $zRx : \cdot; \cdot \Rightarrow x : A; \cdot$ *is derivable in* $\mathsf{FLLS_K}$.

Proof. Observe that propositional rules permute up over the $\Box_L$ rule. Hence all the applications of $\Box_L$ can be done in sequence, just after the $\Box_R$ rule. □

Example 10. The normality axiom is derived as shown in Fig. 7. Note that the modal rules occur in a block corresponding to an application of the sequent rule k. That is, focusing effectively blocks derivations where propositional rules are applied between modal ones.

$$\dfrac{\dfrac{xR[y] : \cdot; y : q, y : p \to \cdot; y : q}{xRy : \cdot; y : q, y : p \Rightarrow y : q; \cdot}\ \mathsf{store_R}, D \qquad \dfrac{xR[y] : \cdot; y : p \to y : q; y : p}{xRy : \cdot; y : p \Rightarrow y : p, y : q; \cdot}\ \mathsf{store_R}, D}{\dfrac{xRy : \cdot; y : p \supset q, y : p \Rightarrow y : q; \cdot}{\dfrac{[x]Ry : \cdot; y : p \supset q, y : p \to y : q; \cdot}{\dfrac{[x]Ry : x : \Box(p \supset q), x : \Box p; \cdot \to y : q; \cdot}{\dfrac{zR[x] : x : \Box(p \supset q), x : \Box p; \cdot \to \cdot; x : \Box q}{\dfrac{zRx : x : \Box(p \supset q), x : \Box p; \cdot \Rightarrow \cdot; x : \Box q}{\dfrac{zRx : \cdot; x : \Box(p \supset q), x : \Box p \Rightarrow x : \Box q; \cdot}{zRx : \cdot; \cdot \Rightarrow x : \Box(p \supset q) \supset (\Box p \supset \Box q); \cdot}\ \supset_R}\ \mathsf{store_L}, \mathsf{store_R}}\ D}\ \Box_R}\ \Box_L}\ R}}\ \supset_L$$

Fig. 7. The derivation of the normality axiom in $\mathsf{FLLS_K}$

$$\heartsuit A \to \Box A \qquad \mathsf{k}_\Box\ \Box(A \to B) \to (\Box A \to \Box B) \qquad \mathsf{t}_\Box\ \Box A \to A \qquad \dfrac{\vdash A}{\vdash \Box A}\ \mathsf{nec}_\Box$$

$$\mathsf{k}_\heartsuit\ \heartsuit(A \to B) \to (\heartsuit A \to \heartsuit B) \qquad \mathsf{t}_\heartsuit\ \heartsuit A \to A \qquad \mathsf{4}_\heartsuit\ \heartsuit A \to \heartsuit\heartsuit A \qquad \dfrac{\vdash A}{\vdash \heartsuit A}\ \mathsf{nec}_\heartsuit$$

Fig. 8. The modal axioms for logic $\mathsf{KT} \oplus_\subseteq \mathsf{S4}$.

5 Some more involved examples

It is straightforward to see that the method described above apply to any sequent calculus which can be written as an end-variant linear nested sequent calculus, in particular to extensions of K with combinations of the axioms $\mathsf{D}, \mathsf{T}, \mathsf{4}$ or to the multi-succedent calculus for intuitionistic logic [7]. We now consider some less trivial examples.

5.1 Simply dependent bimodal logics

As a first example, we consider a bimodal logic with a simple interaction between the modalities. While we only treat one example, our method is readily adapted to other such logics. The language of *simply dependent bimodal logic* $\mathsf{KT} \oplus_\subseteq \mathsf{S4}$ from [4] contains two modalities $\Box$ and $\heartsuit$, and the axioms are the KT axioms for $\Box$ together with the $\mathsf{S4}$ axioms for $\heartsuit$ and the *interaction* axiom $\heartsuit A \to \Box A$ (Fig. 8). Using the methods in [8], these axioms are easily converted into the sequent system $\mathsf{G}_{\mathsf{KT} \oplus_\subseteq \mathsf{S4}}$ extending the standard propositional rules with the modal rules of Fig. 9. It is straightforward to check that these rules satisfy the criteria for cut elimination from [8], and hence $\mathsf{G}_{\mathsf{KT} \oplus_\subseteq \mathsf{S4}}$ is cut-free.

To obtain a focused system, we again convert the sequent calculus into a LNS calculus. However, since now we have two different non-invertible right rules ($\Box_R$ and $\heartsuit_R$), we need to modify the linear nested setting slightly, introducing the two different nesting operators $/\!\!/^\Box$ and $/\!\!/^\heartsuit$ for the rules $\Box_R$ resp. $\heartsuit_R$. The intended interpretation is

$$\iota(\Gamma \vdash \Delta) := \bigwedge \Gamma \to \bigvee \Delta$$
$$\iota(\Gamma \vdash \Delta /\!\!/^\Box \mathcal{H}) := \bigwedge \Gamma \to \bigvee \Delta \vee \Box \iota(\mathcal{H})$$
$$\iota(\Gamma \vdash \Delta /\!\!/^\heartsuit \mathcal{H}) := \bigwedge \Gamma \to \bigvee \Delta \vee \heartsuit \iota(\mathcal{H})$$

$$\frac{\Gamma, \heartsuit\Sigma, \Sigma, \Box\Theta, \Theta \vdash \Delta}{\Gamma, \heartsuit\Sigma, \Box\Theta \vdash \Delta}\ \mathsf{T} \qquad \frac{\heartsuit\Gamma, \heartsuit\Sigma, \Sigma, \Theta \vdash A}{\Omega, \heartsuit\Gamma, \heartsuit\Sigma, \Box\Theta \vdash \Box A, \Xi}\ \Box_R \qquad \frac{\heartsuit\Gamma \vdash A}{\Omega, \heartsuit\Gamma \vdash \heartsuit A, \Xi}\ \heartsuit_R$$

Fig. 9. The modal rules of the sequent calculus $\mathsf{G}_{\mathsf{KT}\oplus_{\subseteq}\mathsf{S4}}$ for $\mathsf{KT} \oplus_{\subseteq} \mathsf{S4}$

$$\frac{\mathcal{G} /\!\!/^{*} \Gamma \vdash \Delta /\!\!/^{\Box} \vdash A}{\mathcal{G} /\!\!/^{*} \Gamma \vdash \Delta, \Box A}\ \Box_{R\Box} \qquad \frac{\mathcal{S}\{\Gamma \vdash \Delta /\!\!/^{\Box} \Sigma, A \vdash \Pi\}}{\mathcal{S}\{\Gamma, \Box A \vdash \Delta /\!\!/^{\Box} \Sigma \vdash \Pi\}}\ \Box_L \qquad \frac{\mathcal{S}\{\Gamma \vdash \Delta /\!\!/^{\Box} \Sigma, \heartsuit A \vdash \Pi\}}{\mathcal{S}\{\Gamma, \heartsuit A \vdash \Delta /\!\!/^{\Box} \Sigma \vdash \Pi\}}\ \heartsuit_{L\Box}$$

$$\frac{\mathcal{G} /\!\!/^{*} \Gamma \vdash \Delta /\!\!/^{\heartsuit} \vdash A}{\mathcal{G} /\!\!/^{*} \Gamma \vdash \Delta, \heartsuit A}\ \heartsuit_{R\heartsuit} \qquad \frac{\mathcal{S}\{\Gamma \vdash \Delta /\!\!/^{\heartsuit} \Sigma, \heartsuit A \vdash \Pi\}}{\mathcal{S}\{\Gamma, \heartsuit A \vdash \Delta /\!\!/^{\heartsuit} \Sigma \vdash \Pi\}}\ \heartsuit_{L\heartsuit} \qquad \frac{\mathcal{S}\{\Gamma, \Box A, A \vdash \Delta\}}{\mathcal{S}\{\Gamma, \Box A \vdash \Delta\}}\ \mathsf{t}_{\Box} \qquad \frac{\mathcal{S}\{\Gamma, \heartsuit A, A \vdash \Delta\}}{\mathcal{S}\{\Gamma, \heartsuit A \vdash \Delta\}}\ \mathsf{t}_{\heartsuit}$$

Fig. 10. The modal linear nested sequent rules for $\mathsf{KT} \oplus_{\subseteq} \mathsf{S4}$. Here $* \in \{\Box, \heartsuit\}$.

The modal sequent rules are then converted into the rules of Fig. 10. The propositional rules are those of $\mathsf{LNS_G}$ (Fig. 1). Cut-free completeness of (the end-active variant of) this calculus again follows from simulating sequent derivations in the rightmost two components.

Lemma 11 (Soundness). *The rules of* $\mathsf{LNS}_{\mathsf{KT}\oplus_{\subseteq}\mathsf{S4}}$ *preserve validity of the formula interpretation of the sequents with respect to* $\mathsf{KT} \oplus_{\subseteq} \mathsf{S4}$ *frames.*

Proof. By showing that if the negation of the interpretation of the conclusion of a rule is satisfiable in a $\mathsf{KT} \oplus_{\subseteq} \mathsf{S4}$ frame, then so is its conclusion, using that in such frames the accessibility relation $R_{\Box}$ for $\Box$ is contained in the accessibility relation $R_{\heartsuit}$ for $\heartsuit$. □

Note that this also shows that the obvious adaption of this calculus to the full nested sequent setting is sound and cut-free complete for $\mathsf{KT} \oplus_{\subseteq} \mathsf{S4}$. For proposing a focused version for the linear nested sequent rules we essentially follow the method given in Section 4, adapting the framework slightly to the multimodal setting by introducing two different kinds of relation terms $xR_{\Box}y$ and $xR_{\heartsuit}y$ corresponding to the accessibility relations of the modalities $\Box$ and $\heartsuit$ respectively. The frame $Fr(\mathcal{R})$ is defined as $(|R_{\Box} \cup R_{\heartsuit}|, R_{\Box} \cup R_{\heartsuit})$ and linelike relation sets are defined using this definition. The FLLS rules then are defined straightforwardly (Fig. 11). Soundness and completeness of the resulting system $\mathsf{FLLS}_{\mathsf{KT}\oplus_{\subseteq}\mathsf{S4}}$ follow as above. Summing up we have:

Theorem 12. $\mathsf{LNS}_{\mathsf{KT}\oplus_{\subseteq}\mathsf{S4}}$ *and* $\mathsf{FLLS}_{\mathsf{KT}\oplus_{\subseteq}\mathsf{S4}}$ *are sound and complete for* $\mathsf{KT} \oplus_{\subseteq} \mathsf{S4}$. □

5.2 Non-normal modal logics

The same ideas also yield LNS calculi and their focused versions for some *non-normal* modal logics, i.e., modal logics that are not extensions of modal logic K (see [3] for an introduction). The calculi themselves are of independent interest since, to the best of our knowledge, nested sequent calculi for the logics below have not been considered before in the literature. The most basic non-normal logic, *classical modal logic* E, is given Hilbert-style by stipulating only the rule (E) (or *congruence rule*) for the connective $\Box$

$$\frac{A \supset B \quad B \supset A}{\Box A \supset \Box B}\ (\mathsf{E})$$

$$\frac{[x]R_{\Box}y : \Gamma; X \to y : A; \Delta}{zR_{*}[x] : \Gamma; X \to \cdot; \Delta, x : \Box A}\ \Box_{R\Box} \qquad \frac{[x]R_{\heartsuit}y : \Gamma; X \to y : A; \Delta}{zR_{*}[x] : \Gamma; X \to \cdot; \Delta, x : \heartsuit A}\ \heartsuit_{R\heartsuit}$$

$$\frac{[x]R_{\Box}y : \Gamma, y : \heartsuit A; X \to Y; \Delta}{[x]R_{\Box}y : \Gamma, x : \heartsuit A; X \to Y; \Delta}\ \heartsuit_{L\Box} \qquad \frac{[x]R_{\Box}y : \Gamma; X, y : A \to Y; \Delta}{[x]R_{\Box}y : \Gamma, x : \Box A; X \to Y; \Delta}\ \Box_{L}$$

$$\frac{[x]R_{\heartsuit}y : \Gamma, y : \heartsuit A; X \to Y; \Delta}{[x]R_{\heartsuit}y : \Gamma, x : \heartsuit A; X \to Y; \Delta}\ \heartsuit_{L\heartsuit} \qquad \frac{zR_{*}x : \Gamma, x : \Box A; X, x : A \Rightarrow Y; \Delta}{zR_{*}x : \Gamma; X, x : \Box A \Rightarrow Y; \Delta}\ t_{\Box}$$

Fig. 11. The modal rules of $\mathsf{FLLS}_{\mathsf{KT}\oplus_{\subseteq}\mathsf{S4}}$. Here $* \in \{\Box, \heartsuit\}$ and y is fresh in $\Box_{R\Box}$ and $\Box_{R\heartsuit}$. The rule $t_{\heartsuit}$ is analogous to $t_{\Box}$ and is omitted. The propositional rules are as in Fig. 4 with R_{*} instead of R.

$$\frac{A \vdash B \quad B \vdash A}{\Gamma, \Box A \vdash \Box B, \Delta}\ (\mathsf{E}) \qquad \frac{A \vdash B}{\Gamma, \Box A \vdash \Box B, \Delta}\ (\mathsf{M}) \qquad \frac{\vdash A}{\Gamma \vdash \Box A, \Delta}\ (\mathsf{N})$$

$$\frac{A_1, \dots, A_n \vdash B \quad B \vdash A_1 \quad \cdots \quad B \vdash A_n}{\Gamma, \Box A_1, \dots, \Box A_n \vdash \Box B, \Delta}\ (\mathsf{E}n) \qquad \frac{A_1, \dots, A_n \vdash B}{\Gamma, \Box A_1, \dots, \Box A_n \vdash \Box B, \Delta}\ (\mathsf{M}n)$$

$\mathsf{G_E}$	$\{\ (\mathsf{E})\ \}$	$\mathsf{G_{EC}}$	$\{\ (\mathsf{E}n) : n \geq 1\ \}$	$\mathsf{G_{MN}}$	$\{\ (\mathsf{M}), (\mathsf{N})\ \}$
$\mathsf{G_M}$	$\{\ (\mathsf{M})\ \}$	$\mathsf{G_{MC}}$	$\{\ (\mathsf{M}n) : n \geq 1\ \}$	$\mathsf{G_{MCN}}$	$\{\ (\mathsf{M}n) : n \geq 0\ \}$

Fig. 12. Sequent rules and calculi for some non-normal modal logics

which allows exchanging logically equivalent formulae under the modality. Some of the better known extensions of this logic are formulated by the addition of axioms from

$$\mathsf{M} \quad \Box(A \wedge B) \to (\Box A \wedge \Box B) \qquad \mathsf{C} \quad (\Box A \wedge \Box B) \to \Box(A \wedge B) \qquad \mathsf{N} \quad \Box\top$$

Fig. 12 shows the modal rules of the standard cut-free sequent calculi for these logics [6], where in addition weakening is embedded in the conclusion. Extensions of E are written by concatenating the names of the axioms, and in presence of the monotonicity axiom M, the initial E is dropped. E.g., the logic MC is the extension of E with axioms M and C. Its sequent calculus $\mathsf{G_{MC}}$ is given by the standard propositional and structural rules together with the rule (E) as well as the rules $(\mathsf{M}n)$ for $n \geq 1$.

We first consider *monotone logics*, i.e., extensions of M. To simulate the rules from Fig. 12 in the linear nested setting we introduce an auxiliary nesting operator $/\!\!/^{\mathsf{m}}$ to capture a state where a sequent rule has been partly processed. In contrast, the intuition for the original nesting $/\!\!/$ is that the simulation of a rule is finished. In view of end-active systems, we restrict the occurrences of $/\!\!/^{\mathsf{m}}$ to the end of the structures. Linear nested sequents for monotonic non-normal modal logics then are given by:

$$\mathsf{LNS_m} ::= \Gamma \vdash \Delta \ \mid\ \Gamma \vdash \Delta /\!\!/^{\mathsf{m}} \Sigma \vdash \Pi \ \mid\ \Gamma \vdash \Delta /\!\!/ \mathsf{LNS_m}$$

The modal linear nested sequent rules are given in Fig. 13. The propositional rules are those of the end-active version of $\mathsf{LNS_G}$ (Fig. 1) with the restriction that they cannot be applied inside $/\!\!/^{\mathsf{m}}$. The sequent rule $(\mathsf{M}n)$ is then simulated by the following derivation

$$\dfrac{A_1, \dots, A_n \vdash B}{\Box A_1, \dots, \Box A_n \vdash \Box B}\ (\mathsf{M}n) \quad \begin{matrix}\vdots\\ \mathcal{G}\end{matrix} \qquad \rightsquigarrow \qquad \dfrac{\dfrac{\dfrac{\mathcal{G} /\!\!/ \vdash /\!\!/ A_1, \dots, A_{n-1}, A_n \vdash B}{\mathcal{G} /\!\!/ \vdash /\!\!/^{\mathsf{m}} A_1, \dots, A_{n-1} \vdash B}\ \Box_L^{\mathsf{m}}}{\mathcal{G} /\!\!/ \Box A_1, \dots, \Box A_n \vdash /\!\!/^{\mathsf{m}} \vdash B}\ \Box_L^{\mathsf{c}}}{\mathcal{G} /\!\!/ \Box A_1, \dots, \Box A_n \vdash \Box B}\ \Box_R^{\mathsf{m}}$$

$$\frac{\mathcal{G} \sslash \Gamma \vdash \Delta \sslash^{\mathsf{m}} \vdash B}{\mathcal{G} \sslash \Gamma \vdash \Box B, \Delta}\ \Box_R^{\mathsf{m}} \qquad \frac{\mathcal{G} \sslash \Gamma \vdash \Delta \sslash \Sigma, A \vdash \Pi}{\mathcal{G} \sslash \Gamma, \Box A \vdash \Delta \sslash^{\mathsf{m}} \Sigma \vdash \Pi}\ \Box_L^{\mathsf{m}}$$

$$\frac{\mathcal{G} \sslash \Gamma \vdash \Delta \sslash^{\mathsf{m}} \Sigma, A \vdash \Pi}{\mathcal{G} \sslash \Gamma, \Box A \vdash \Delta \sslash^{\mathsf{m}} \Sigma \vdash \Pi}\ \Box_L^{\mathsf{c}} \qquad \frac{\mathcal{G} \sslash \Gamma \vdash \Delta \sslash \ \vdash B}{\mathcal{G} \sslash \Gamma \vdash \Box B, \Delta}\ \Box_R^{\mathsf{n}}$$

$\mathsf{LNS_M}\ \{ \Box_R^{\mathsf{m}}, \Box_L^{\mathsf{m}} \}$ $\mathsf{LNS_{MC}}\ \{ \Box_R^{\mathsf{m}}, \Box_L^{\mathsf{m}}, \Box_L^{\mathsf{c}} \}$ $\mathsf{LNS_{MN}}\ \{ \Box_R^{\mathsf{m}}, \Box_L^{\mathsf{m}}, \Box_R^{\mathsf{n}} \}$ $\mathsf{LNS_{MCN}}\ \{ \Box_R^{\mathsf{m}}, \Box_L^{\mathsf{m}}, \Box_L^{\mathsf{c}}, \Box_R^{\mathsf{n}} \}$

Fig. 13. Modal linear nested sequent rules for some monotone non-normal modal logics.

$$\frac{\mathcal{G} \sslash \Gamma \vdash \Delta \sslash^{\mathsf{e}} (\vdash B; B \vdash)}{\mathcal{G} \sslash \Gamma \vdash \Box B, \Delta}\ \Box_R^{\mathsf{e}} \qquad \frac{\mathcal{G} \sslash \Gamma \vdash \Delta \sslash \Sigma, A \vdash \Pi \quad \mathcal{G} \sslash \Gamma \vdash \Delta \sslash \Omega \vdash A, \Theta}{\mathcal{G} \sslash \Gamma, \Box A \vdash \Delta \sslash^{\mathsf{e}} (\Sigma \vdash \Pi; \Omega \vdash \Theta)}\ \Box_L^{\mathsf{e}}$$

$$\frac{\mathcal{G} \sslash \Gamma \vdash \Delta \sslash^{\mathsf{e}} (\Sigma, A \vdash \Pi; \Omega \vdash \Theta) \quad \mathcal{G} \sslash \Gamma \vdash \Delta \sslash \Omega \vdash A, \Theta}{\mathcal{G} \sslash \Gamma, \Box A \vdash \Delta \sslash^{\mathsf{e}} (\Sigma \vdash \Pi; \Omega \vdash \Theta)}\ \Box_L^{\mathsf{ec}}$$

$\mathsf{LNS_E}\ \{ \Box_R^{\mathsf{e}}, \Box_L^{\mathsf{e}} \}$ $\mathsf{LNS_{EC}}\ \{ \Box_R^{\mathsf{e}}, \Box_L^{\mathsf{e}}, \Box_L^{\mathsf{ec}} \}$

Fig. 14. Modal linear nested sequent rules for some non-monotone non-normal modal logics

For extensions of classical modal logic E not containing the monotonicity axiom M we need to store more information about the unfinished premisses. Thus instead of $\sslash^{\mathsf{m}}$ we introduce a *binary* nesting operator $\sslash^{\mathsf{e}}(.;.)$. Linear nested sequents then are given by

$$\mathsf{LNS_e} ::= \Gamma \vdash \Delta \ \mid \ \Gamma \vdash \Delta \sslash^{\mathsf{e}} (\Sigma \vdash \Pi; \Omega \vdash \Theta) \ \mid \ \Gamma \vdash \Delta \sslash \mathsf{LNS_e}$$

Fig. 14 shows the modal rules for these logics, where again the propositional rules are those of end-active $\mathsf{LNS_G}$ (Fig. 1) with the restriction that they are not applied inside the nesting $\sslash^{\mathsf{e}}$. The derivation simulating the rule ($\mathsf{E}n$) then is

$$\cfrac{\cfrac{\cfrac{\mathcal{G} \sslash \Gamma \vdash \Delta \sslash A_1, \dots, A_n \vdash B \quad \mathcal{G} \sslash \Gamma \vdash \Delta \sslash B \vdash A_n}{\mathcal{G} \sslash \Gamma, \Box A_n \vdash \Delta \sslash^{\mathsf{e}} (A_1, \dots, A_{n-1} \vdash B; B \vdash)}\ \Box_L^{\mathsf{e}} \\ \vdots \\ \mathcal{G} \sslash \Gamma, \Box A_2, \dots, \Box A_n \vdash \Delta \sslash^{\mathsf{e}} (A_1 \vdash B; B \vdash) \qquad \mathcal{G} \sslash \Gamma, \Box A_2, \dots, \Box A_n \vdash \Delta \sslash B \vdash A_1}{\mathcal{G} \sslash \Gamma, \Box A_1, \dots, \Box A_n \vdash \Delta \sslash^{\mathsf{e}} (\vdash B; B \vdash)}\ \Box_L^{\mathsf{ec}}}{\mathcal{G} \sslash \Gamma, \Box A_1, \dots, \Box A_n \vdash \Box B, \Delta}\ \Box_R^{\mathsf{e}}$$

Theorem 13 (Completeness). *The linear nested sequent calculi of Fig. 13 and Fig. 14 are complete w.r.t. the corresponding logics.* □

For showing soundness of such calculi we need a different method, though. This is due to the fact that, unlike for normal modal logics, there is no clear formula interpretation for linear nested sequents for non-normal modal logics. However, since the propositional rules cannot be applied inside the auxiliary nestings $\sslash^{\mathsf{m}}$ resp. $\sslash^{\mathsf{e}}$, the modal rules can only occur in blocks. Together with the fact that the (end-variant) propositional rules can only be applied in the last component this means that we can straightforwardly translate LNS derivations back into sequent derivations.

$$\frac{yR_{\mathsf{m}}[z,w] : \Gamma; X, y : A \to z : A, \Delta}{xR[y] : \Gamma; X \to \cdot; \Delta, y : \Box A}\ \Box_R^e \qquad \frac{yRz : \Gamma; X, z : A \Rightarrow Y; \Delta \quad yRw : \Gamma; X, w : A \Rightarrow Y; \Delta}{yR_{\mathsf{m}}[z,w] : \Gamma, y : \Box A; X \to Y, \Delta}\ \Box_L^e$$

$$\frac{yR_{\mathsf{m}}[z,w] : \Gamma; X, z : A \to Y, \Delta \qquad yRw : \Gamma; X, w : A \Rightarrow Y; \Delta}{yR_{\mathsf{m}}[z,w] : \Gamma, y : \Box A; X \to Y, \Delta}\ \Box_L^{ec}$$

$$\mathsf{FLLS_E} \quad \{ \Box_R^e, \Box_L^e \} \qquad \mathsf{FLLS_{EC}} \quad \{ \Box_R^e, \Box_L^e, \Box_L^{ec} \}$$

Fig. 15. The modal FLLS rules for non-monotone non-normal modal logics

Theorem 14 (Soundness). *If a sequent $\Gamma \vdash \Delta$ is derivable in $\mathsf{LNS}_{\mathcal{L}}$ for $\mathcal{L}$ one of the logics of this section, then it is derivable in the corresponding sequent calculus.*

Proof. By translating a $\mathsf{LNS}_{\mathcal{L}}$ derivation into a $\mathsf{G}_{\mathcal{L}}$ derivation, discarding everything apart from the last component of the linear nested sequents, and translating blocks of modal rules into the corresponding modal sequent rules. E.g., a block consisting of an application of $\Box_L^{\mathsf{m}}$ followed by n applications of $\Box_L^c$ and an application of $\Box_R^{\mathsf{m}}$ is translated into an application of the rule ($\mathsf{M}n$). The propositional rules only work on the last component and never inside the nesting $/\!\!/^{\mathsf{m}}$ resp. $/\!\!/^{\mathsf{e}}$ and are translated easily by the corresponding sequent rules. □

Remark 15. It is possible to consider linear nested sequent calculi for these non-normal modal logics in which the propositional rules are not restricted to their end-active versions. In this case, soundness can be shown by a permutation-of-rules argument, similar to the argument for *levelled derivations* in [9], using "levelling-preserving" invertibility of the propositional rules.

The modal FLLS rules for the non-monotone non-normal modal logics are given in Fig. 15, writing R_{e} for the relation corresponding to $/\!\!/^{\mathsf{e}}$. The propositional rules are those of $\mathsf{FLLS_K}$ (Fig. 6). The systems for monotone logics are constructed similarly.

6 Automatic proof search in linear nested sequents

The method for constructing focused systems from Section 4 generates *optimal* systems, in the sense that proof search complexity matches exactly that of the original sequent calculi. We will now go one step further and exploit the fact that these calculi sport separate left and right introduction rules for the modalities to present a systematic way of encoding labelled line nested sequents in linear logic. This enables us to both: (i) use the rich linear logic meta-level theory in order to reason about the specified systems; and (ii) use a linear logic prover in order to do automatic proof search in those systems.

Observe that, while the goal in (ii) is also achieved by implementing the focused versions of the various systems case by case, using a meta-level framework like linear logic allows the use of *a single* prover for various logics: all one has to do is to change the theory, *i.e.*, the specified introduction clauses. Some encodings are presented in Appendix C and the implementation of the specified systems is available online at `http://subsell.logic.at/nestLL/`.

6.1 From sequent rules to linear logic clauses

We now consider *focused linear logic* (LLF) as a "meta-logic" and the formulae of a labelled modal logic as the "object-logic" and then illustrate how sets of bipoles in linear logic can be used to specify sequent calculi for the object-logic. Since we follow mostly the procedure of [10], here we only give a general idea, leaving the details to Appendix A.

Specifying sequents Let *obj* be the type of object-level formulae and let $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ be two meta-level predicates on these, i.e., both of type $obj \to o$. Object-level sequents of the form $B_1, \ldots, B_n \vdash C_1, \ldots, C_m$ (where $n, m \geq 0$) are specified as the multiset $\lfloor B_1\rfloor, \ldots, \lfloor B_n\rfloor, \lceil C_1\rceil, \ldots, \lceil C_m\rceil$ within the LLF proof system. The $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ predicates identify which object-level formulas appear on which side of the sequent – brackets down for left (useful mnemonic: $\lfloor$ for "left") and brackets up for right. Finally, binary relations R are specified by a meta-level atomic formula of the form $R(\cdot, \cdot)$.

Specifying inference rules Inference rules are specified by a re-writing clause that replaces the active formulae in the conclusion by the active formulae in the premises. The linear logic connectives indicate how these object level formulae are connected: contexts are copied (&) or split ($\otimes$), in different inference rules ($\oplus$) or in the same sequent ($⅋$). For example, the specification of (a representative sample of) the rules of $\mathsf{LLS_K}$ are

$$
\begin{array}{ll}
(\mathsf{init}) & \exists A.\lfloor x : A\rfloor^\perp \otimes \lceil x : A\rceil^\perp \otimes \mathtt{atomic}(A) \\
(\wedge_l) & \exists A, B.\lfloor x : A \wedge B\rfloor^\perp \otimes \lfloor x : A\rfloor ⅋ \lfloor x : B\rfloor \\
(\wedge_r) & \exists A, B.\lceil x : A \wedge B\rceil^\perp \otimes \lceil x : A\rceil \ \& \lceil x : B\rceil \\
(\Box_R) & \exists A, B.\lceil x : \Box A\rceil^\perp \otimes \forall y.(\lceil y : A\rceil ⅋ R(x, y)) \otimes \exists z.R(z, x)^\perp \\
(\Box_L) & \exists A, B.\lfloor x : \Box A\rfloor^\perp \otimes \exists y.(\lfloor y : A\rfloor ⅋ R(x, y)) \otimes R(x, y)^\perp
\end{array}
$$

The correspondence between focusing on a formula and an induced big-step inference rule is particularly interesting when the focused formula is a *bipole*. Roughly speaking, bipoles are positive formulae in which no positive connective can be in the scope of a negative one (see Def. 18 or [10, Def. 3]). Focusing on such a formula will produce a single positive and a single negative phase. This two-phase decomposition enables the adequate capturing of the application of an object-level inference rule by the meta-level logic. For example, focusing on the bipole clause ($\Box_R$) will produce the derivation

$$
\dfrac{\pi_1 \qquad \dfrac{\Psi; \Delta', \lceil y : A\rceil, R(x, y)) \Uparrow}{\Psi; \Delta' \Downarrow \forall y.(\lceil y : A\rceil ⅋ R(x, y))}\ [R \Downarrow, \forall, ⅋, R \Uparrow] \qquad \pi_2}{\Psi; \Delta \Downarrow \exists A, B.\lceil x : \Box A\rceil^\perp \otimes \forall y.(\lceil y : A\rceil ⅋ R(x, y)) \otimes \exists z.R(z, x)^\perp}\ [\exists, \otimes]
$$

where $\Delta = \lceil x : \Box A\rceil \cup R(z, x) \cup \Delta'$, and π_1 and π_2 are, respectively,

$$
\dfrac{}{\Psi; \lceil x : \Box A\rceil \Downarrow \lceil x : \Box A\rceil^\perp}\ I_1 \qquad\qquad \dfrac{}{\Psi; R(z, x) \Downarrow \exists z.R(z, x)^\perp}\ [\exists, I_1]
$$

This one-step focused derivation will: (a) consume $\lceil x : \Box A\rceil$ and $R(z, x)$; (b) create a fresh label y; and (c) add $\lceil y : A\rceil$ and $R(x, y)$ to the context. Observe that this matches *exactly* the application of the object-level rule $\Box_R$.

When specifying a system (logical, computational, etc) into a meta level framework, it is desirable and often mandatory that the specification is *faithful*, that is, one step of computation on the object level should correspond to one step of logical reasoning in the meta level. This is what is called *adequacy* [12].

Definition 16. *A specification of an object sequent system is* proof-adequate *if provability is preserved by the specification. If the adequacy can be shown for (open) derivations (such as inference rules themselves), then we call the specification* adequate*.*

Fig. 19 shows adequate specifications in LLF of the labelled systems for the logic EC. These specifications can be used for automatic proof search as illustrated by the following theorem which is shown readily using the methods of [10].

Theorem 17. *Let L be a* LLS *system and let $\mathcal{L}$ be the theory given by the clauses of an adequate specification of the inference rules of L. A sequent $\mathcal{R}, \Gamma \vdash \Delta$ is provable in L if and only if $\mathcal{L}; \mathcal{R} \Uparrow \lfloor \Gamma \rfloor, \lceil \Delta \rceil$ is provable in LLF.* □

Specifying modalities The reason why the specifications in LLF and the construction of focused systems for LLS systems work rather well is the fact that the LNS modal rules only manipulate a fixed number of principal formulae, i.e., one can choose some formulae and replace them with some other formulae. If there are no principal formulae, or if the object rule is context dependent, then proposing such encodings or a neat notion of focusing becomes tricky, as it is often the case with sequent systems for modal logics. In [13] linear logic with *subexponentials* (SELL) was used as a framework for specifying a number of modal logics. Unfortunately, the encodings are far from natural, and cannot be automated. Thus, in our opinion, the use of linear nested systems constitutes a significant step towards defining efficient methods for proof search, but also the construction of automatic provers for modal logics.

7 Concluding remarks and future work

In this work we used the correspondence between linear nested sequents and labelled line sequents to (a) propose focused nested sequent systems for a number of modal logics (including a non-trivial bimodal logic and non-normal logics) which match the complexity of existing sequent calculi; and (b) specify the labelled systems in linear logic, thereby obtaining automatic provers for all of them. This not only constitues a significant step towards a better understanding of proof theory for modal logics in general, but also opens an avenue for research in proof search for a broad set of systems (not only modal).

One natural line of investigation concerns the applicability of this approach to logics based on non-classical propositional logic such as constructive modal logics. Moreover, we would like to understand whether our methods work for “proper” nested sequent calculi, i.e., calculi for logics which are not based on a cut-free sequent calculus, such as the calculi for K5 or KB [2]. Finally, it might be possible to automatically extract focused systems from LLF specifications (while not explicitly mentioned, there is an attempt to do so in [10]). It would be rather interesting to compare these systems with ours. Moreover, this would provide a total mechanisation of proof systems for end-active LNS systems, in the sense that the map $\mathbb{TL}_x$ automatically generates LLS systems and the encoding into LLF also can be automatised. Hence one could choose to either use an existing LLF prover to do proof search in these systems or generate a specific prover automatically, based on a focused version of the system.

References

1. Andreoli, J.M.: Logic programming with focusing proofs in linear logic. J. of Logic and Computation 2(3), 297–347 (1992)
2. Brünnler, K.: Deep sequent systems for modal logic. Arch. Math. Log. 48, 551–577 (2009)
3. Chellas, B.F.: Modal Logic. Cambridge University Press (1980)
4. Demri, S.: Complexity of simple dependent bimodal logics. In: Dyckhoff, R. (ed.) TABLEAUX 2000, LNCS, vol. 1847, pp. 190–204. Springer (2000)
5. Goré, R., Ramanayake, R.: Labelled tree sequents, tree hypersequents and nested (deep) sequents. In: AiML 9. pp. 279–299 (2012)
6. Lavendhomme, R., Lucas, T.: Sequent calculi and decision procedures for weak modal systems. Studia Logica 65, 121–145 (2000)
7. Lellmann, B.: Linear nested sequents, 2-sequents and hypersequents (2015), accepted for publication, TABLEAUX 2015
8. Lellmann, B., Pattinson, D.: Constructing cut free sequent systems with context restrictions based on classical or intuitionistic logic. In: ICLA 2013, LNCS, vol. 7750, pp. 148–160. Springer (2013)
9. Masini, A.: 2-sequent calculus: a proof theory of modalities. Ann. Pure Appl. Logic 58, 229–246 (1992)
10. Miller, D., Pimentel, E.: A formal framework for specifying sequent calculus proof systems. Theor. Comput. Sci. 474, 98–116 (2013)
11. Negri, S., van Plato, J.: Proof Analysis: A Contribution to Hilbert's Last Problem. Cambridge University Press (2011)
12. Nigam, V., Miller, D.: A framework for proof systems. J. of Automated Reasoning 45(2), 157–188 (2010)
13. Nigam, V., Pimentel, E., Reis, G.: An extended framework for specifying and reasoning about proof systems. J. of Logic and Computation (2014)
14. Poggiolesi, F.: The method of tree-hypersequents for modal propositional logic. In: Towards Mathematical Philosophy, Trends In Logic, vol. 28, pp. 31–51. Springer (2009)
15. Straßburger, L.: Cut elimination in nested sequents for intuitionistic modal logics. In: Pfenning, F. (ed.) FOSSACS 2013, LNCS, vol. 7794, pp. 209–224. Springer (2013)

A Focused linear logic

The connectives of linear logic can be divided into two classes. The *negative* connectives have invertible introduction rules: these connectives are $\parr$, $\bot$, &, $\top$, $\forall$, and ?. The *positive* connectives $\oplus$, 0, $\otimes$, 1, $\exists$, and ! are the de Morgan duals of the negative connectives. A formula is *positive* if it is a negated atom or its top-level logical connective is positive. Similarly, a formula is *negative* if it is an atom or its top-level logical connective is negative.

The one-sided version of the focused proof system LLF is given in Figure 16 (the variable y in the [$\forall$] rule is restricted so that it is not free in any formula of its conclusion). A *literal* is either an atomic formula or a negated atomic formula. In LLF, there are two kinds of sequents: $\Psi;\Delta \Uparrow L$ and $\Psi;\Delta \Downarrow F$, where Ψ is a set of formulas, Δ is a multiset of formulas, L is a list of formulas, and F is a formula. The inference rules with $\Uparrow$ in the premises and conclusion are the invertible rules. A sequence of these rules, reading them bottom-up, deals with the "don't-care non-determinism" of proof search: in this *negative* phase of proof construction, no backtracking on the selection of inference rules is necessary. The inference rules with $\Downarrow$ in the conclusion are the non-invertible rules. A sequence of these rules, reading them bottom-up, deals with the "don't-know non-determinism" of proof search: in this *positive* phase of proof construction, choices within inference rules can lead to failures for which one may need to backtrack. The negative phase ends (reading proofs bottom-up) when $\top \in L$ or when all formulas in L have been "processed": that is, when L is the empty list. The positive phase begins by choosing (via one of the decide rules [D_1] or [D_2]) a formula F on which to focus. Positive rules are applied to F until either 1 or a negated atom is encountered (and the proof must end using the introduction rule [1] or an initial rule [I_1] or [I_2] respectively), the promotion rule [!] is applied or a negative subformula is encountered (and the proof switches to the negative phase). This means that focused proofs can be seen (bottom-up) as a sequence of alternations between negative and positive phases.

A.1 Bipoles

Definition 18. *A* monopole *formula is a linear logic formula that is built up from atoms and occurrences of the negative connectives, with the restriction that* ? *has atomic scope. A* bipole *is a positive formula built from monopoles and negated atoms using only positive connectives, with the additional restriction that* ! *can only be applied to a monopole.*

Using the linear logic distributive properties, monopoles are equivalent to formulas of the form

$$\forall x_1 \ldots \forall x_p[\&_{i=1,\ldots,n} \parr_{j=1,\ldots,m_i} B_{i,j}],$$

where the $B_{i,j}$ are either atoms or the result of applying ? to an atomic formula. Similarly, bipoles can be rewritten as formulas of the form

$$\exists x_1 \ldots \exists x_p[\oplus_{i=1,\ldots,n} \otimes_{j=1,\ldots,m_i} C_{i,j}],$$

where $C_{i,j}$ are either negated atoms, monopole formulas, or the result of applying ! to a monopole formula. Notice that the units $\top$, 0, $\bot$, and 1 are 0-ary versions of &, $\oplus$, $\parr$, *Negative rules*

$$\frac{\Psi;\Delta \Uparrow L}{\Psi;\Delta \Uparrow \bot, L}\,[\bot] \qquad \frac{\Psi;\Delta \Uparrow F, G, L}{\Psi;\Delta \Uparrow F \parr G, L}\,[\parr] \qquad \frac{\Psi, F;\Delta \Uparrow L}{\Psi;\Delta \Uparrow ?F, L}\,[?]$$

$$\frac{}{\Psi;\Delta \Uparrow \top, L}\,[\top] \qquad \frac{\Psi;\Delta \Uparrow F, L \quad \Psi;\Delta \Uparrow G, L}{\Psi;\Delta \Uparrow F \,\&\, G, L}\,[\&] \qquad \frac{\Psi;\Delta \Uparrow F[y/x], L}{\Psi;\Delta \Uparrow \forall x.F, L}\,[\forall]$$

Positive rules

$$\frac{}{\Psi;\cdot \Downarrow 1}\,[1] \qquad \frac{\Psi;\Delta_1 \Downarrow F \quad \Psi;\Delta_2 \Downarrow G}{\Psi;\Delta_1,\Delta_2 \Downarrow F \otimes G}\,[\otimes] \qquad \frac{\Psi;\cdot \Uparrow F}{\Psi;\cdot \Downarrow !F}\,[!]$$

$$\frac{\Psi;\Delta \Downarrow F_1}{\Psi;\Delta \Downarrow F_1 \oplus F_2}\,[\oplus_l] \qquad \frac{\Psi;\Delta \Downarrow F_2}{\Psi;\Delta \Downarrow F_1 \oplus F_2}\,[\oplus_r] \qquad \frac{\Psi;\Delta \Downarrow F[t/x]}{\Psi;\Delta \Downarrow \exists x.F}\,[\exists]$$

Identity, Decide, and Reaction rules

$$\frac{}{\Psi;A \Downarrow A^\perp}\,[I_1] \qquad \frac{}{\Psi, A;\cdot \Downarrow A^\perp}\,[I_2] \qquad \frac{\Psi;\Delta \Downarrow F}{\Psi;\Delta, F \Uparrow \cdot}\,[D_1] \qquad \frac{\Psi, F;\Delta \Downarrow F}{\Psi, F;\Delta \Uparrow \cdot}\,[D_2]$$

In $[I_1]$ and $[I_2]$, A is atomic; in $[D_1]$ and $[D_2]$, F is not an atom.

$$\frac{\Psi;\Delta, F \Uparrow L}{\Psi;\Delta \Uparrow F, L}\,[R\Uparrow] \qquad \text{provided that } F \text{ is positive or an atom}$$

$$\frac{\Psi;\Delta \Uparrow F}{\Psi;\Delta \Downarrow F}\,[R\Downarrow] \qquad \text{provided that } F \text{ is negative}$$

Fig. 16. Focused proof search in linear logic LLF.

and $\otimes$, respectively. Given this normal representation of bipoles and according to the focusing discipline, it turns out that, once introduced, a bipole is completely decomposed into its atomic subformulas, a fact illustrated by the following bipole derivation.

$$\dfrac{\cdots \quad \dfrac{\dfrac{\cdots \quad \dfrac{\Psi';\Gamma' \Uparrow \cdot}{\Psi;\Gamma' \Uparrow \parr_{j=1,\ldots,m_i} ?A_{i,j}}\,[\parr, ?] \quad \cdots}{\Psi;\Gamma' \Uparrow \forall x_1 \ldots \forall x_p[\&_{i=1,\ldots,n} \parr_{j=1,\ldots,m_i} ?A_{i,j}]}\,[\forall, \&]}{\Psi;\Gamma' \Downarrow !\forall x_1 \ldots \forall x_p[\&_{i=1,\ldots,n} \parr_{j=1,\ldots,m_i} ?A_{i,j}]}\,[!] \quad \cdots}{\Psi;\Gamma \Downarrow \exists x_1 \ldots \exists x_t[\oplus_{i=1,\ldots,k} \otimes_{j=1,\ldots,q_i} C_{i,j}]}\,[\exists, \oplus, \otimes]$$

Here $A_{i,j}$ is atomic for all i, j. If the connective ! is not present, then the rule ! is replaced by the rule $R\Downarrow$.

Definition 19. *Let* Q *be the set* $\{\lfloor\cdot\rfloor, \lceil\cdot\rceil\}$. *An* introduction clause *is a closed bipole formula of the form*

$$\exists x_1 \ldots \exists x_n[(q(\diamond(x_1, \ldots, x_n)))^\perp \otimes F]$$

where $\diamond$ *is an object-level connective of arity* n ($n \geq 0$) *and* $q \in Q$*. Furthermore,* F *does not contain negated atoms and an atom occurring in* F *is either of the form* $p(x_i)$ *or* $p(x_i(y))$ *where* $p \in Q$ *and* $1 \leq i \leq n$*. In the first case,* x_i *has type* obj *while in the* *second case x_i has type $d \to obj$ and y is a variable (of type d) quantified (universally or existentially) in F (in particular, y is not in $\{x_1, \ldots, x_n\}$).*

Focusing on an introduction clause replaces an atom $q(\diamond(t_1, \ldots, t_n))$ with the formula $F[t_1/x_1, \ldots, t_n/x_n]$. Since this formula is a bipole, it will be immediately decomposed into its atomic subformulas, hence capturing in one meta-level step of derivation the one object-level step of applying an inference rule.

B Some end-active labelled systems

$$\frac{}{zRx, X, x : p \vdash x : p, Y}\ \mathsf{init} \qquad \frac{zRx, X, x : A, x : B \vdash Y}{zRx, X, x : A \wedge B \vdash Y}\ \wedge_L \qquad \frac{zRx, X \vdash x : A, Y \quad zRx, X \vdash x : B, Y}{zRx, X \vdash x : A \wedge B, Y}\ \wedge_R$$

$$\frac{}{zRx, X, x : \bot \vdash Y}\ \bot_L \qquad \frac{zRx, X \vdash Y, x : A \quad zRx, X, x : B \vdash Y}{zRx, X, x : A \supset B \vdash Y}\ \supset_L \qquad \frac{zRx, X, x : A \vdash Y, x : B}{zRx, X \vdash Y, x : A \supset B}\ \supset_R$$

$$\frac{xRy, X, y : A \vdash Y}{xRy, X, x : \Box A \vdash Y}\ \Box_L \qquad \frac{xRy, X \vdash Y, y : A}{zRx, X \vdash Y, x : \Box A}\ \Box_R$$

Fig. 17. End-active $\mathsf{LLS_K}$. In rule init, p is atomic.

$$\frac{xR_\Box y, X \vdash Y, y : A}{zR_* x, X \vdash Y, x : \Box A}\ \Box_{R\Box} \qquad \frac{xR_\heartsuit y, X \vdash Y, y : A}{zR_* x, X \vdash Y, x : \heartsuit A}\ \heartsuit_{R\heartsuit}$$

$$\frac{xR_\Box y, X, y : \heartsuit A \vdash Y}{xR_\Box y, X, x : \heartsuit A \vdash Y}\ \heartsuit_{L\Box} \qquad \frac{xR_\Box y, X, y : A \vdash Y}{xR_\Box y, X, x : \Box A \vdash Y}\ \Box_L$$

$$\frac{xR_\heartsuit y, X, y : \heartsuit A \vdash Y}{xR_\heartsuit y, X, x : \heartsuit A \vdash Y}\ \heartsuit_{L\heartsuit} \qquad \frac{zR_* x, X, x : \Box A, x : A \vdash Y}{zR_* x, X, x : \Box A \vdash Y}\ \mathsf{t}_\Box$$

$$\frac{zR_* x, X, x : \heartsuit A, x : A \vdash Y}{zR_* x, X, x : \heartsuit A \vdash Y}\ \mathsf{t}_\heartsuit$$

Fig. 18. End-active $\mathsf{KT} \oplus_\subseteq \mathsf{S4}$. Here $* \in \{\Box, \heartsuit\}$ and y is a fresh variable in $\Box_{R\Box}$ and $\Box_{R\heartsuit}$.

C Some specifications in LLF

$$
\begin{array}{ll}
(\Box_R^{\mathsf{e}}) & \lceil x:\Box B\rceil^\perp \otimes \forall y \forall z.(\lceil y:B\rceil \parr \lfloor z:B\rfloor \parr R_{\mathsf{e}}(x,y,z)) \otimes \exists z.R(z,x)^\perp \\
(\Box_L^{\mathsf{e}}) & \lfloor x:\Box A\rfloor^\perp \otimes \exists y \exists z.((\lfloor y:\Box A\rfloor \parr R(x,y)) \otimes (\lceil z:\Box A\rceil \parr R(x,z)) \otimes R_{\mathsf{e}}(x,y,z)^\perp) \\
(\Box_L^{\mathsf{ec}}) & \lfloor x:\Box A\rfloor^\perp \otimes \exists y \exists z.((\lfloor y:\Box A\rfloor \parr R_{\mathsf{e}}(x,y,z)) \otimes (\lceil z:\Box A\rceil \parr R(x,z)) \otimes R_{\mathsf{e}}(x,y,z)^\perp)
\end{array}
$$

Fig. 19. The LLF specification of the modal rules of $\mathsf{LLS}_{\mathsf{EC}}$ for the logic EC from Sec. 5.2.

$$
\begin{array}{ll}
(\Box_R) & \lceil x:\Box A\rceil^\perp \otimes \forall y.(\lceil y:A\rceil \parr R_\Box(x,y)) \otimes \exists z.R_*(z,x)^\perp \\
 & \lfloor x:\Box A\rfloor^\perp \otimes \exists y.((\lfloor y:A\rfloor \parr R_\Box(x,y)) \otimes R_\Box(x,y)^\perp) \\
 & \lfloor x:\heartsuit A\rfloor^\perp \otimes \exists y.((\lfloor y:A\rfloor \parr R_\Box(x,y)) \otimes R_\Box(x,y)^\perp) \\
 & \lfloor x:\heartsuit A\rfloor^\perp \otimes \exists y.((\lfloor y:\heartsuit A\rfloor \parr R_\Box(x,y)) \otimes R_\Box(x,y)^\perp) \\
(\heartsuit_R) & \lceil x:\heartsuit A\rceil^\perp \otimes \forall y.(\lceil y:A\rceil \parr R_\heartsuit(x,y)) \otimes \exists z.R_*(z,x)^\perp \\
 & \lfloor x:\heartsuit A\rfloor^\perp \otimes \exists y.((\lfloor y:\heartsuit A\rfloor \parr R_\heartsuit(x,y)) \otimes R_\heartsuit(x,y)^\perp) \\
(\mathsf{T}_\Box) & \lfloor x:\Box A\rfloor^\perp \otimes \lfloor x:A\rfloor \\
(\mathsf{T}_\heartsuit) & \lfloor x:\heartsuit A\rfloor^\perp \otimes \lfloor x:A\rfloor
\end{array}
$$

Fig. 20. The specification of the modal rules from the labelled-linear sequent calculus for $\mathsf{KT} \oplus_\subseteq \mathsf{S4}$ in LLF. Here $* \in \{\Box, \heartsuit\}$.

$$
\begin{array}{ll}
(\Box_R^{\mathsf{m}}) & \lceil x:\Box B\rceil^\perp \otimes \forall y.(\lceil y:B\rceil \parr R_{\mathsf{m}}(x,y)) \otimes \exists z.R(z,x)^\perp \\
(\Box_L^{\mathsf{m}}) & \lfloor x:\Box A\rfloor^\perp \otimes \exists y.((\lfloor y:\Box A\rfloor \parr R(x,y)) \otimes R_{\mathsf{m}}(x,y)^\perp) \\
(\Box_L^{\mathsf{c}}) & \lfloor x:\Box A\rfloor^\perp \otimes \exists y.((\lfloor y:\Box A\rfloor \parr R_{\mathsf{m}}(x,y)) \otimes R_{\mathsf{m}}(x,y)^\perp) \\
(\Box_R^{\mathsf{n}}) & \lceil x:\Box B\rceil^\perp \otimes \forall y.(\lceil y:B\rceil \parr R(x,y)) \otimes \exists z.R(z,x)^\perp
\end{array}
$$

Fig. 21. Specification of the systems for monotone non-normal modal logics.