Interactive Theorem Proving with Temporal Logic

Amy Felty, Laurent Théry

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Interactive Theorem Proving with Temporal Logic

Amy Felty, Laurent Théry

N° 2804

Février 1996

PROGRAMME 2

ISSN 0249-6399

Interactive Theorem Proving with Temporal Logic

Amy Felty*, Laurent Théry**

Programme 2 — Calcul symbolique, programmation et génie logiciel
Projet Croap

Rapport de recherche n° 2804 — Février 1996 — 41 pages

Abstract: In this paper, we present a theorem prover for linear temporal logic. Our goal is to extend the capabilities of existing interactive and automatic systems for verifying temporal properties of software and hardware systems. We focus on increasing the effectiveness of user interaction in such systems. In particular, we extend the techniques of *proof by pointing* and *point and shoot* for mouse-driven proof construction in first-order logic to temporal logic. In addition, we show how to generate text from proofs by extending a previously given translation for first-order logic to the temporal operators. Our theorem prover implements an inference system for temporal logic that we have defined. The inference rules of this system are more intuitive than the rules commonly given for temporal logics and thus they are better suited to our goals. We present this inference system and prove that it is sound and complete with respect to a known system.

Key-words: interactive theorem proving, linear temporal logic, proof by pointing, textual presentation

(Résumé : tsvp)

*Bell Laboratories, Lucent Technologies, 600 Mountain Ave., Murray Hill, NJ 07974 USA
felty@research.att.com
**Laurent.Thery@inria.fr

Unité de recherche INRIA Sophia-Antipolis
2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex (France)
Téléphone : (33) 93 65 77 77 – Télécopie : (33) 93 65 77 65

Preuves Interactives en Logique Temporelle

Résumé : Ce papier présente un démonstrateur de théorèmes pour la logique temporelle linéaire. Notre but est d'étendre les possibilités des démonstrateurs actuels pour vérifier les propriétés temporelles des systèmes logiciels et matériels. Nous nous intéressons ici à augmenter la pertinence de l'interaction homme-machine dans ces systèmes. En particulier nous étendons à la logique temporelle la technique de *preuve par sélection* développée pour la logique du premier ordre. Nous montrons aussi comment générer du texte à partir des preuves ainsi construites. Notre démonstrateur utilise un système de règles d'inférence que nous avons défini. Ces règles sont plus intuitives que celles usuellement proposées pour la logique temporelle et donc plus adaptées à nos objectifs. Nous présentons notre système de règles d'inférence et prouvons qu'il est correct et complet.

Mots-clé : preuve interactive, logique temporelle, preuve par sélection, présentation textuelle

1 Introduction

Temporal logics are widely used in verification of algorithms and systems in which reasoning about time is important for ensuring correctness. These logics are mainly used to formalize and express properties about future or possible behaviors in such systems. For example, *linear temporal logics* have been successfully used to express and prove properties of concurrent and reactive systems (*e.g.*, [10]). In this paper, we present a system that implements one such logic, the modal logic S4.3 with the two standard modal operators $\Box$ (always) and $\Diamond$ (eventually), whose semantics give a linear interpretation to time.

In order to formally verify large-scale complex systems, it will be important to have sophisticated verification tools that can integrate a variety of interactive and automatic techniques. In this paper, we concentrate on the interactive component of such verification systems. We show how techniques for interactive proof search developed for first-order logic can be extended to S4.3. We focus in particular on three aspects of effective interaction. First, the basic inference rules should correspond to intuitive proof steps. Second, it is important to provide simple operations (*e.g.* via mouse interaction) that have a direct and intuitive correspondence to the application of some combination of these basic inferences. For example, when there is an assumption of the form $A \vee B$, a mouse click on B might direct the system to break the proof into two cases, one with A as an assumption, and the other with B as an assumption, and in addition indicate that the second case should become the "current subgoal", *i.e.*, the one that all subsequent operations will be applied to unless otherwise specified. Third, it should be easy for the user to understand the proof at all points during and after its construction, and thus good proof presentation is crucial in such systems.

The theorem prover and graphical interface of our system are implemented as two separate components. The theorem prover uses the tactic-style theorem proving environment implemented in the higher-order logic programming language λProlog, as described in [6]. A simple tactic theorem prover for backward step-by-step proof construction is obtained from a direct specification of the inference rules of the desired logic. The inference system for S4.3 that we use is one that we have designed with our goals for effective interaction in mind. We begin with Gentzen's sequent calculus for first-order intuitionistic logic restricted to the propositional case presented in [13]. For classical logic, instead of using a multiple conclusion sequent calculus, we add a rule for excluded middle to the single conclusion system for intuitionistic logic. We then add rules for the temporal operators and show that the resulting system is sound and complete with respect to the multiple conclusion system given in [8].

We choose a sequent system since it is easy to map to interactive backward proof steps. We choose a single conclusion sequent calculus because proofs are generally more intuitive than those in multiple conclusion calculi.

The graphical interface of our system is implemented in Centaur [3]. In particular, we build on an existing interface for the theorem prover obtained from the first-order intuitionistic logic specification mentioned above. First, we extend the techniques of *proof by pointing* and *point and shoot* described in [2] to associate operations to mouse clicks on temporal formulas. Second, we extend techniques for generating textual explanations from proofs. To do so, we define a natural deduction inference system which is better-suited than the sequent calculus to the generation of readable text. We extend the mapping of natural deduction proofs to pseudo-English given in [4] for first-order logic by illustrating how to map the inference rules for the modal operators to fragments of text. Proof construction in our system proceeds by incrementally filling in such text.

In order to integrate both sequent and natural deduction proofs in the theorem prover, we show that our single conclusion sequent calculus has a direct mapping to our natural deduction system. We do so by introducing an intermediate inference system that builds fragments of natural deduction proofs within sequent proofs, and proving that both sequent and natural deduction proofs can easily be extracted. Our theorem prover is a direct implementation of this proof system and builds both kinds of proofs simultaneously. This implementation extends a similar one for first-order intuitionistic logic in [5].

In the next section, we present our sequent calculus for S4.3 and show that it is sound and complete with respect to Goré's system. In Section 3, we present a natural deduction system for S4.3 and show that it is sound and complete with respect to our sequent calculus. Section 4 illustrates proof construction in our system, and presents the extensions of proof by pointing and point and shoot to temporal logic. Section 5 shows how to map the temporal rules to text. In Section 6, we present a complete example illustrating interaction with the system, and in Section 7, we conclude. The proofs of the theorems in Sections 2 and 3 are given in the appendix.

2 A Sequent Calculus for S4.3

Figure 1 contains a complete set of inference rules for a sequent calculus for S4.3, which we call $\mathcal{S}$. In this system a sequent is written $\Gamma \vdash A$ where Γ is a *set* of formulas called the *assumptions* or *context*, and A is a formula. Following convention, we write A, Γ to denote the set $\Gamma \cup \{A\}$, and Γ, Γ' to denote the set $\Gamma \cup \Gamma'$. In addition, $\Box\Gamma$

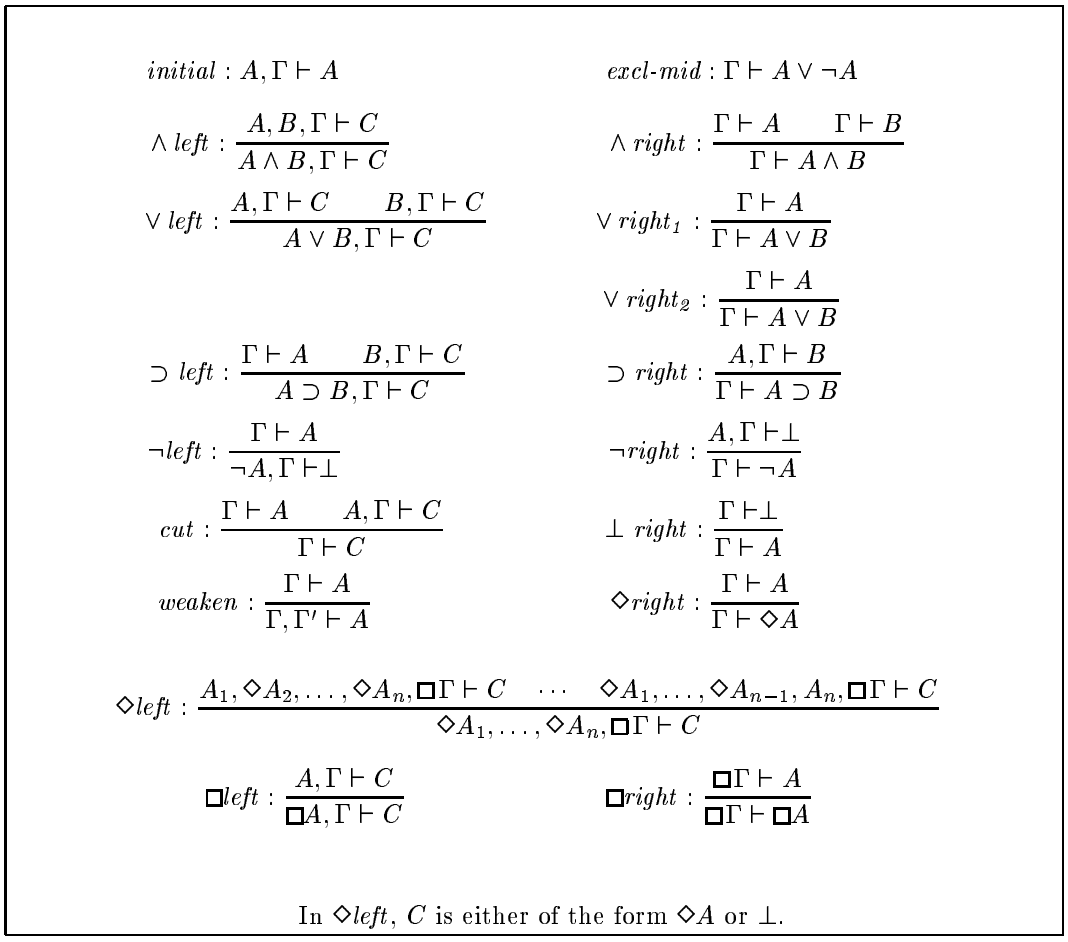

Figure 1: The $\mathcal{S}$ sequent calculus for S4.3

is a set of formulas such that each formula is of the form $\Box A$ or $\neg\Diamond\neg A$. In S4.3, for any formula A, the following dual equivalences hold: (1) $\Box A$ is equivalent to $\neg\Diamond\neg A$ and (2) $\Diamond A$ is equivalent to $\neg\Box\neg A$. We call a tree built from the rules in Figure 1 an *$\mathcal{S}$-proof.*

The formula $\bot$ has a special status in $\mathcal{S}$. This formula can only occur in proofs on the right in a sequent. Furthermore, it must occur alone; it must not be a subformula of any other formula.

Each of the rules of this sequent calculus can be given an intuitive reading. These readings will be reflected directly in the generation of text from proofs. Most of the propositional rules are straightforward. The $\wedge$ *right* rule for example states that if A and B each hold from the assumptions Γ, then we can conclude $A \wedge B$ holds under the same assumptions. Many of the readings of the *left* rules are given in the backwards direction and involve reasoning in a forward direction from the assumptions. The $\wedge$ *left* rule for example states that if we have as an assumption $A \wedge B$, then we can add to our assumptions both A and B separately. The $\vee$ *left* rule involves reasoning by cases; the formula C holds under the assumptions $A \vee B$ and Γ if it holds under the two cases: A and Γ, and B and Γ. We give interpretations to the modal rules that involve reasoning about time. If we interpret a sequent to mean that the conclusion holds from the assumptions at the present time, then the interpretations of $\Diamond$*right* and $\Box$*left* have simple readings. The $\Diamond$*right* rule states that if A holds now, then A eventually holds. The $\Box$*left* rule reads that from the assumption that A always holds, we can conclude that A holds now. The other two are slightly more complicated. The $\Box$*right* rule states that if A holds from a set of assumptions that all hold all the time, then A holds all the time. The $\Diamond$*left* rule involves reasoning by cases from a set of one or more assumptions that all eventually hold. The cases are broken down according to which one holds "first". In particular, there are n premises where $n \geq 1$, and for $i = 1, \ldots, n$, premise i is the case where A_i holds first. First here does not mean strictly before all others. There may be others that hold at the same time, though none can hold before. In addition, in order for this reasoning to be valid, all other assumptions used in the reasoning must hold all the time, and the conclusion must either be of the form $\Diamond A$ or $\bot$.

To show that this inference system is sound and complete, we show that the set of provable sequents is exactly those that are provable in the $\mathcal{S}'$ inference system in Figure 2. $\mathcal{S}'$ is a multiple-conclusion sequent calculus for S4.3 presented in [8]. A sequent is written $\Gamma \vdash \Delta$ where Γ and Δ are both sets of formulas. $\mathcal{S}'$ does not contain inference rules for the modal operator $\Diamond$. However, using the equivalence between the prefixes $\Diamond$ and $\neg\Box\neg$, we express and prove the correctness of $\mathcal{S}$ as follows.

Theorem 1 *Given a set of formulas Γ and a formula C, let Γ' and C' be Γ and C, respectively, with all occurrences of $\Diamond$ replaced by $\neg\Box\neg$. The sequent $\Gamma \vdash C$ is provable in $\mathcal{S}$ if and only if $\Gamma' \vdash C'$ (or $\Gamma' \vdash \emptyset$ when C is $\bot$) is provable in $\mathcal{S}'$.*

The $\mathcal{S}'$ system has the *cut-free property*, *i.e.*, any sequent provable in $\mathcal{S}'$ has a proof without any occurrences of the *cut* rule (see [8]). The cut-free property does not hold for the $\mathcal{S}$ system, but it can be shown that only limited use of the *cut* rule is needed, as expressed by the following theorem.

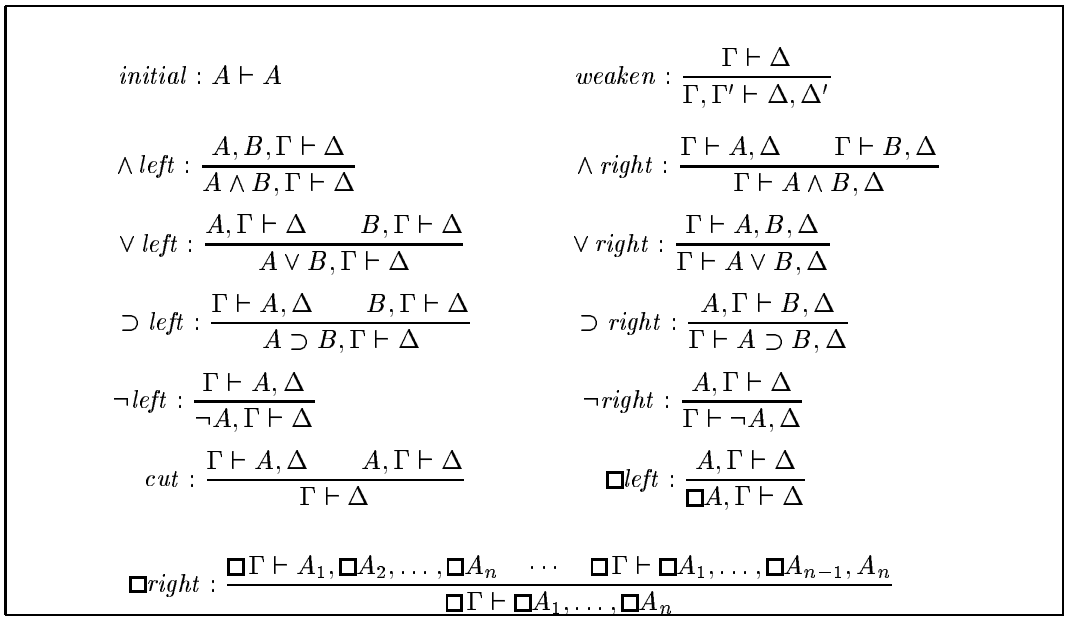

Figure 2: The $\mathcal{S}'$ multiple-conclusion sequent calculus for S4.3

Theorem 2 *Given a set of formulas Γ and a formula C, let Γ' and C' be Γ and C, respectively, with all occurrences of $\Diamond$ replaced by $\neg\Box\neg$. If the sequent $\Gamma' \vdash C'$ (or $\Gamma' \vdash \emptyset$ when C is $\bot$) is provable in $\mathcal{S}'$ without cut, then $\Gamma \vdash C$ has a proof in $\mathcal{S}$ such that in all occurrences of the cut rule, the left premise is a direct consequence of the excl-mid rule.*

3 A Natural Deduction Presentation of S4.3

Figure 3 contains a complete set of rules for a natural deduction inference system for S4.3, which we call $\mathcal{N}$. The rules are presented in the style of [12]. Formulas in parentheses are used to indicate the *discharge* of assumptions. In $\supset$ *intro*, for example, any occurrence of the formula A as a leaf in the tree above B is discharged by the application of the rule. The brackets in $\Box$*intro* and $\Diamond$ *elim* rules also denote discharge; all formulas in brackets are discharged by the rule application. In addition, the brackets denote a restriction on proofs: the formulas in brackets must be the

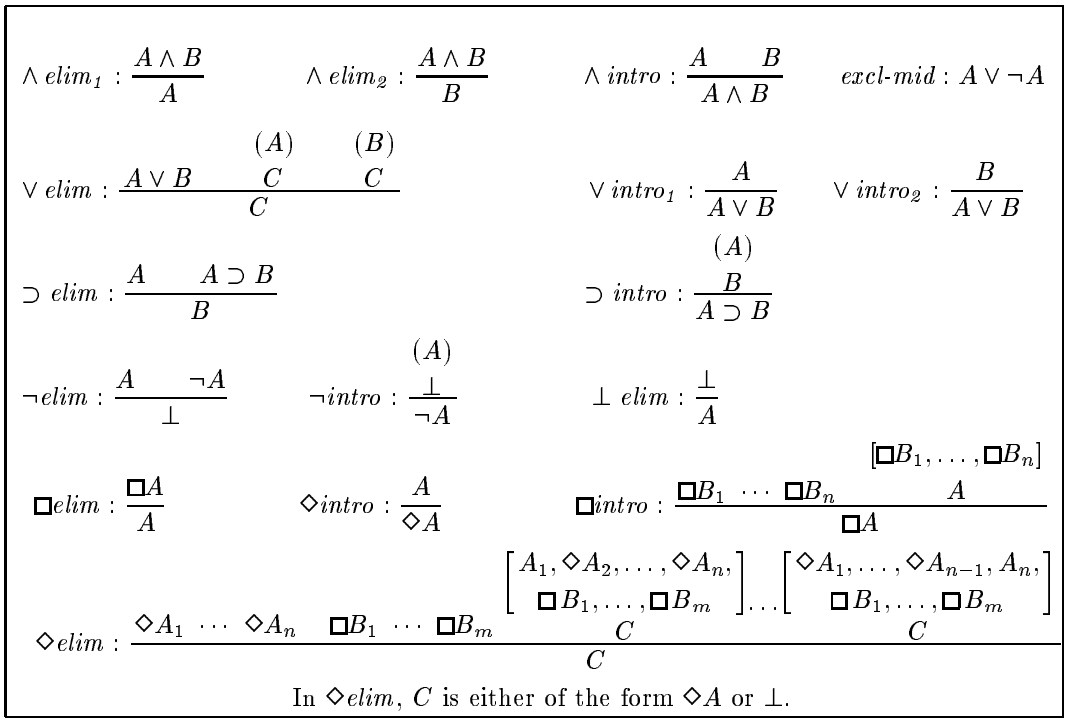

Figure 3: The natural deduction inference system $\mathcal{N}$ for S4.3

only formulas that occur as leaves in the subtree above the premise in which they occur. Note that in $\Box intro$ and $\Diamond elim$, $n > 0$ and $m \geq 0$. A proof in $\mathcal{N}$ of a formula C from a set of assumptions Γ is a tree constructed from the inference rules of Figure 3 such that every formula that occurs as a leaf and is not discharged by any rule application is either of the form $A \vee \neg A$ or occurs in Γ. We call such a proof an *$\mathcal{N}$-proof.*

The combination sequent and natural deduction inference system that we implement in our theorem prover is the system $\mathcal{M}$ given in Figure 4. It is the same as system $\mathcal{S}$ except that fragments of $\mathcal{N}$-proofs occur on the left in sequents. We denote such proofs using Π, possibly subscripted. To further denote that the formula that occurs at the root of such a proof is A, we write $\overset{\Pi}{A}$. For sets of such proofs, we write Ψ, again possibly subscripted. A formula alone on the left of a sequent denotes a one-node $\mathcal{N}$-proof. In contrast, only formulas, not proofs, occur on the

right in sequents of $\mathcal{M}$. We write $\Box\Psi$ to denote a set of proofs such that the root formula of each has prefix $\Box$ or $\neg\Diamond\neg$. Note for example, that the proof of B in the right premise of $\supset$ *left* is built from a one-node proof of A and the proof of $A \supset B$ that occurs in the conclusion. In the $\supset$ *right* rule, the formula $A \supset B$ occurs in the conclusion, while in the premise the one-node proof A occurs on the left and the formula B occurs on the right. In $\Diamond$*left*, arbitrary proofs occur on the left in the conclusion, while only one-node proofs appear in the premises. Note that in building a proof top-down, the $\mathcal{N}$-proofs in the conclusion appear to come out of nowhere. However, if we consider the bottom-up construction of proofs, the application of left rules can be viewed as the construction of new assumptions by forward reasoning from existing assumptions. The $\mathcal{N}$-proofs record the proofs of this forward reasoning. In the case of the $\Diamond$*left* rule, this record is dropped when continuing the proofs of the premises. Technically, these proofs are needed to define the function which extracts $\mathcal{N}$-proofs from $\mathcal{M}$-proofs, a function which is defined recursively over the structure of $\mathcal{M}$-proofs. For the $\Diamond$*left* rule, the $\mathcal{N}$-proofs occurring in the conclusion are not needed (and in fact must not be there) in order to extract $\mathcal{N}$-proofs from the premises. For the same reason, there are $\mathcal{N}$-proofs occurring in the conclusions of the $\neg$*left* and $\vee$ *left* rules which do not occur in the premises.

The correspondence between the sequent system $\mathcal{S}$ and the mixed system $\mathcal{M}$ can be made formal by defining two functions that take a proof in one to a proof in the other by simply replacing each rule application in one system by an application of the corresponding rule in the other system. The function $\hat{\mathcal{S}}$ from $\mathcal{M}$ to $\mathcal{S}$ can be defined to be the operation that simply erases $\mathcal{N}$-proofs on the left of sequents by replacing each proof with the formula at its root. The function $\overline{\mathcal{S}}$ from $\mathcal{S}$ to $\mathcal{M}$ can be defined by starting at the root and replacing each formula A on the left of the sequent with some $\mathcal{N}$-proof whose root is A and proceeding upward replacing formulas in the premises with the corresponding proofs built using the proof fragments in the conclusion. To make this definition precise, the set of $\mathcal{N}$-proofs replacing the formulas on the left of the root sequent is given as an explicit argument to $\overline{\mathcal{S}}$. As a result, the function application $\overline{\mathcal{S}}(\Sigma, \Psi)$ is only well-defined if the set of formulas on the left at the root of $\mathcal{S}$-proof Σ is the same as the set of formulas occurring at the roots of the $\mathcal{N}$-proofs Ψ. The following theorem is then proved by a simple inductive argument on proof trees.

Theorem 3 *Let Ψ be a set of $\mathcal{N}$-proofs and let Γ be the set of formulas that occur at the root of the proofs in Ψ. Let C be a formula. If Σ is a proof of $\Psi \vdash C$ in $\mathcal{M}$, then $\hat{\mathcal{S}}(\Sigma)$ is a proof of $\Gamma \vdash C$ in $\mathcal{S}$. Conversely, if Σ' is a proof of $\Gamma \vdash C$ in $\mathcal{S}$,*

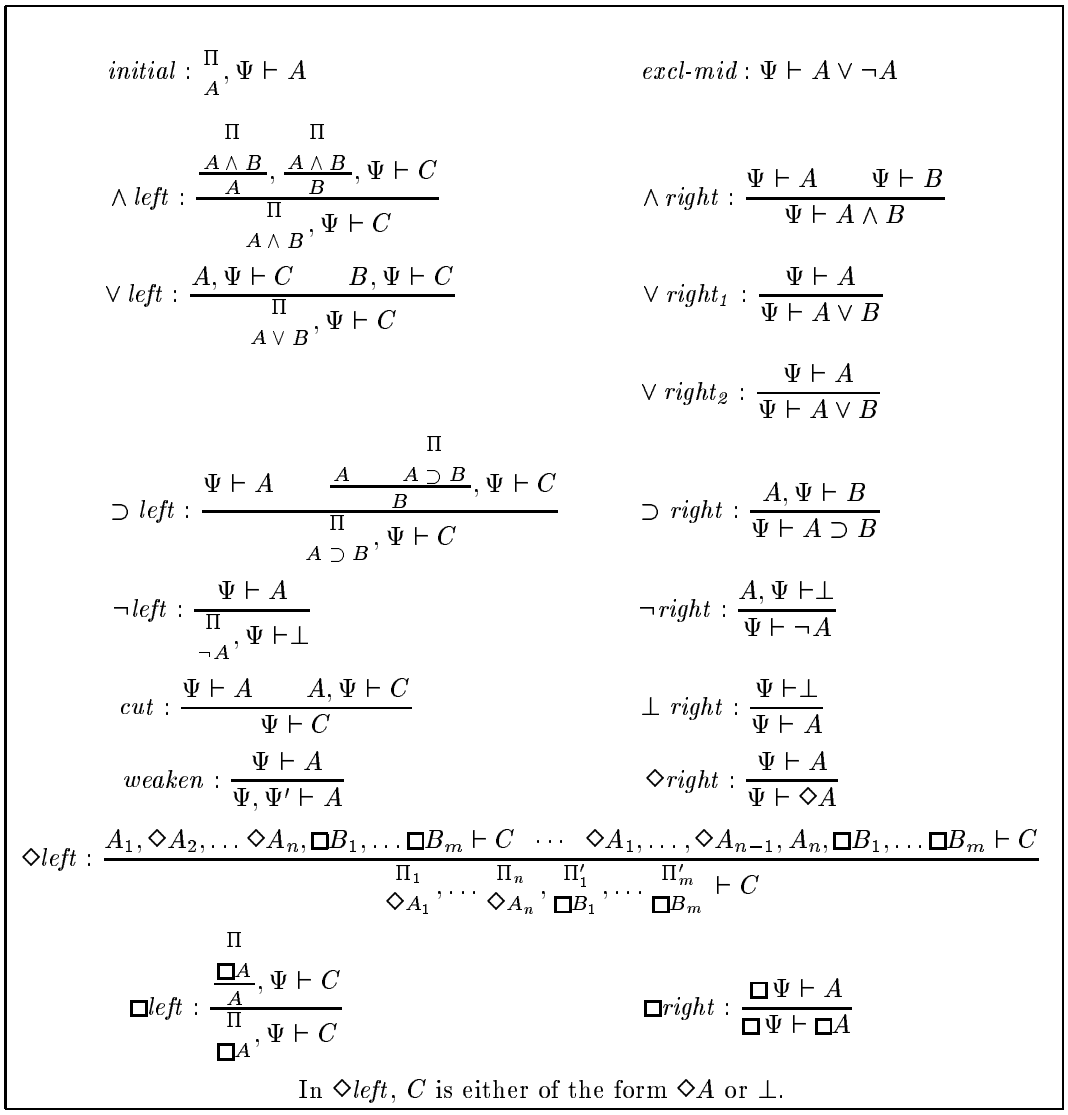

Figure 4: The mixed inference system $\mathcal{M}$: sequent rules with natural deduction fragments

then $\overline{\mathcal{S}}(\Sigma', \Psi)$ *is a proof of* $\Psi \vdash C$ *in* $\mathcal{M}$. *Furthermore, if* Ψ *contains only one-node proofs, then* $\overline{\mathcal{S}}(\hat{\mathcal{S}}(\Sigma), \Psi) = \Sigma$ *and* $\hat{\mathcal{S}}(\overline{\mathcal{S}}(\Sigma', \Psi) = \Sigma'$.

The soundness and completeness of $\mathcal{M}$ follow directly from this theorem.

The correspondence between $\mathcal{N}$ and $\mathcal{M}$ is not as direct. However, one direction—converting proofs in $\mathcal{M}$ to proofs in $\mathcal{N}$—is fairly direct. This is the direction we are interested in. In particular, our theorem prover builds proofs in $\mathcal{M}$ and we extract natural deduction proofs so that we can map them to text. We consider only proofs in $\mathcal{M}$ such that in all occurrences of the *cut* rule, the left premise is a direct consequence of the *excl-mid* rule. Let *assumps* be the function that maps an $\mathcal{N}$-proof Π to the set of formulas containing all formulas occurring as leaves in Π that are neither of the form $A \vee \neg A$ nor are discharged by any rule application. We extend this function to operate on sets of $\mathcal{N}$-proofs as follows: $\mathit{assumps}(\{\Pi_1, \ldots, \Pi_n\}) := \mathit{assumps}(\Pi_1) \cup \cdots \cup \mathit{assumps}(\Pi_n)$. We define the function $\hat{\mathcal{N}}$ that maps an $\mathcal{M}$-proof of $\Psi \vdash C$ to an $\mathcal{N}$-proof of C from $\mathit{assumps}(\Psi)$ recursively from the root upward, with a case for each inference rule. An $\mathcal{M}$-proof ending with $\wedge\, \mathit{right}$, for example, as shown on the left below is mapped to the $\mathcal{N}$-proof on the right below whose last inference is an application of $\wedge \mathit{intro}$.

$$\wedge\, \mathit{right} : \frac{\Sigma_1 \qquad \Sigma_2}{\Psi \vdash A \wedge B} \quad \Rightarrow \quad \frac{\mathcal{N}(\Sigma_1) \qquad \mathcal{N}(\Sigma_2)}{A \wedge B}$$

Here, Σ_1 is a proof of the sequent $\Psi \vdash A$ and Σ_2 is a proof of $\Psi \vdash B$. The cases for the other *right* rules are all defined by a similar recursion on the premises followed by an application of the corresponding *intro* rule. The one-node $\mathcal{M}$-proof of $\Psi \vdash A \vee \neg A$ is mapped directly to the one-node $\mathcal{N}$-proof $A \vee \neg A$. The remaining rules are slightly more complicated and the mapping is given in Figure 5. In this figure, Σ, Σ_1, etc., are assumed to be proofs of the premises of the specified rule. The sequents at the root of these proofs are assumed to be of the appropriate form (see Figure 4). The following theorem expresses the correctness of the translation of $\mathcal{M}$-proofs to $\mathcal{N}$-proofs as defined by the function $\hat{\mathcal{N}}$.

Theorem 4 *Let* C *be a formula and* Ψ *be a set of* $\mathcal{N}$*-proofs. If* Σ *is an* $\mathcal{M}$*-proof of* $\Psi \vdash C$, *then* $\hat{\mathcal{N}}(\Sigma)$ *is an* $\mathcal{N}$*-proof of* C *from* $\mathit{assumps}(\Psi)$.

Although arbitrary applications of *cut* are not necessary for the completeness of $\mathcal{M}$, it is important in practice to allow them in interactive theorem proving. Our implementation handles such applications by: (1) applying the function $\hat{\mathcal{N}}$ to the left premise $\Psi \vdash A$ to obtain an $\mathcal{N}$-proof Π of A from $\mathit{assumps}(\Psi)$, (2) modifying

$$initial : \overset{\Pi}{A}, \Psi \vdash A \quad \Rightarrow \quad \begin{matrix}\Pi \\ A\end{matrix}$$

$$\wedge\, left : \frac{\Sigma}{\overset{\Pi}{A \wedge B}, \Psi \vdash C} \quad \Rightarrow \quad \hat{\mathcal{N}}(\Sigma)$$

$$\vee\, left : \frac{\Sigma_1 \qquad \Sigma_2}{\overset{\Pi}{A \vee B}, \Psi \vdash C} \quad \Rightarrow \quad \frac{\begin{matrix}\Pi \\ A \vee B\end{matrix} \qquad \hat{\mathcal{N}}(\Sigma_1) \qquad \hat{\mathcal{N}}(\Sigma_2)}{C}$$

$$\supset\, left : \frac{\Sigma_1 \qquad \Sigma_2}{\overset{\Pi}{A \supset B}, \Psi \vdash C} \quad \Rightarrow \quad \frac{\hat{\mathcal{N}}(\Sigma_1) \qquad \begin{matrix}\Pi \\ A \supset B\end{matrix}}{B}$$

$$\neg left : \frac{\Sigma}{\overset{\Pi}{\neg A}, \Psi \vdash \bot} \quad \Rightarrow \quad \frac{\hat{\mathcal{N}}(\Sigma) \qquad \begin{matrix}\Pi \\ \neg A\end{matrix}}{\bot}$$

$$cut : \frac{\Psi \vdash A \vee \neg A \qquad \Sigma}{\Psi \vdash C} \quad \Rightarrow \quad \hat{\mathcal{N}}(\Sigma)$$

$$weaken : \frac{\Sigma}{\Psi, \Psi' \vdash A} \quad \Rightarrow \quad \hat{\mathcal{N}}(\Sigma)$$

$$\Diamond left : \frac{\Sigma_1 \quad \cdots \quad \Sigma_n}{\overset{\Pi_1}{\Diamond A_1}, \ldots, \overset{\Pi_n}{\Diamond A_n}, \overset{\Pi'_1}{\Box B_1}, \ldots, \overset{\Pi'_m}{\Box B_m} \vdash C} \quad \Rightarrow \quad \frac{\begin{matrix}\Pi_1 & & \Pi_n & \Pi'_1 & & \Pi'_m \\ \Diamond A_1 & \cdots & \Diamond A_n & \Box B_1 & \cdots & \Box B_m\end{matrix} \qquad \hat{\mathcal{N}}(\Sigma_1) \cdots \hat{\mathcal{N}}(\Sigma_n)}{C}$$

$$\Box left : \frac{\Sigma}{\overset{\Pi}{\Box A}, \Psi \vdash C} \quad \Rightarrow \quad \hat{\mathcal{N}}(\Sigma)$$

Figure 5: The function $\hat{\mathcal{N}}$ for transforming mixed-rule proofs to natural deduction proofs

the proof of the right premise $A, \Psi \vdash C$ by modifying the proofs of Ψ to replace all occurrences of A as a leaf with Π, (3) applying $\hat{\mathcal{N}}$ to the resulting $\mathcal{M}$-proof.

We do not consider a translation of $\mathcal{N}$-proofs to $\mathcal{M}$-proofs here. However, we note that it is possible to define a translation on *normal proofs* (see [12]) in the propositional intuitionistic fragment of $\mathcal{N}$ that does not use the *cut* rule, thereby illustrating the correspondence between cut-free sequent proofs and normal natural deduction proofs for this fragment.

4 Proof Construction

Interactive proof construction is most often done in a backward direction. The user sets a goal and then, applying the rules of the logic, tries to reduce it to already known theorems or axioms. The technique of *proof by pointing* described in [2] provides a means of giving proof directions by selecting subexpressions of goals. It has been proved sound and complete for classical logic. In what follows, we explain how the technique can be extended to our sequent system for S4.3, and we give some examples of proofs of temporal properties. We describe proof search using the $\mathcal{S}$ system, although as already stated, our theorem prover implements the $\mathcal{M}$ system which also builds $\mathcal{N}$-proofs of the assumptions.

4.1 Proof by Pointing

The main idea of proof by pointing is that each rule in the sequent presentation can be seen as a way of breaking down a term. The term to break is in the conclusion of the rule, either the conclusion of the sequent or one of its assumptions. The result is presented by the subterms of the term reappearing in the premises. For example, the $\wedge$*right* rule:

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$$

can be seen as breaking a conjunction, putting the left part in the first premise and the right one in the second premise. We can express this more graphically by the following two rules where in the first, the user has clicked on A, and in the second on B:

$$\frac{\Gamma \vdash \boxed{A} \qquad \Gamma \vdash B}{\Gamma \vdash \boxed{A} \wedge B}$$

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash \boxed{B}}{\Gamma \vdash A \wedge \boxed{B}}$$

Figure 6 presents the boxed rules for the usual connectives of propositional classical logic. Selecting a subexpression can be understood as a command to bring the

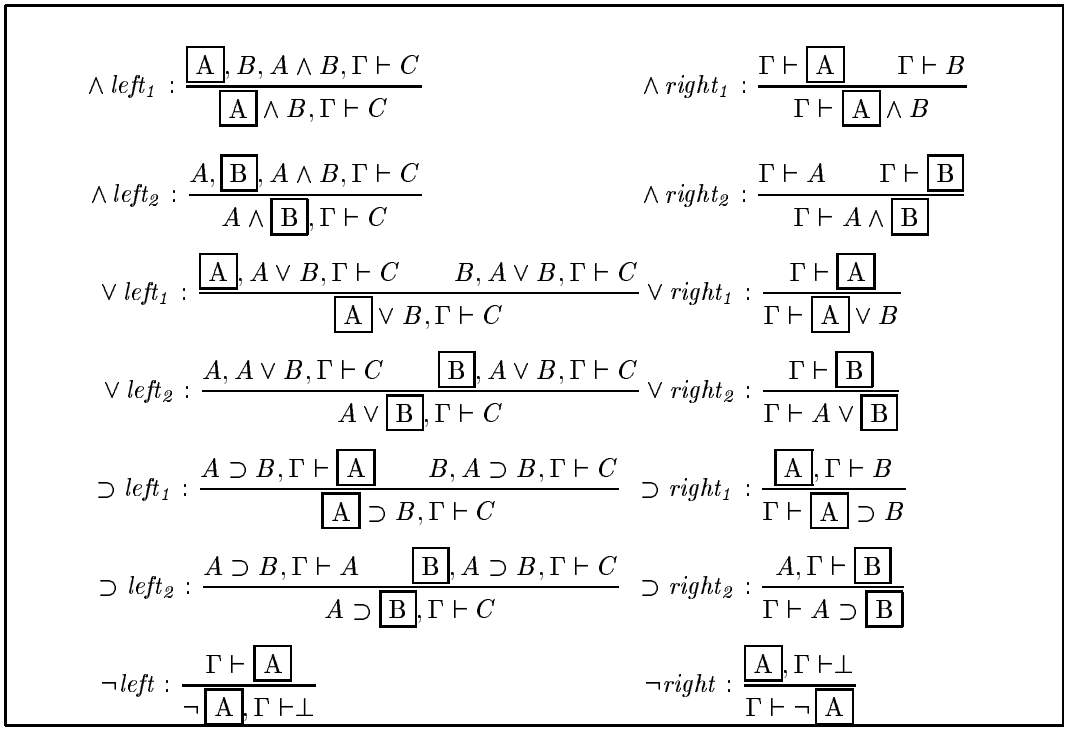

Figure 6: Proof by pointing rules for propositional classical connectives

subexpression to the surface of the sequent. It only makes sense if we have the two following properties:

1. *Well foundedness*: the box in the premises is more "immediate" than the formula it came from in the conclusion. This property ensures termination as the propagation of the selection moves toward the surface.

2. *Uniqueness*: given a goal and a selection, there is at most one rule that is applicable. This property ensures determinism.

These two properties hold for the rules of Figure 6. Here, more "immediate" means that the formula in the premises is smaller than the formula it came from in the

conclusion. Furthermore we have a property of *completeness*: any subformula can be reached by recursive application of the rules. Given a selection, we can then induce an algorithm that performs a series of rule applications. We display each step of the algorithm with a "$\rightarrow$" representing application of a rule from Figure 6. We do not indicate which rule since it will always be clear from context. An overline over a sequent is used to indicate that a branch of the proof has been completed using the *initial* rule. In addition, the propagation of the selection is displayed by underlining the selected subterm. As an example, a selection on z in the leftmost formula below gives:

$$\vdash x \wedge y \supset \underline{z} \vee t \;\rightarrow\; x \wedge y \vdash \underline{z} \vee t \;\rightarrow\; x \wedge y \vdash \underline{z}$$

which consists of an application of $\supset$ *right* followed by $\vee right_1$. All the rules of Figure 6 are instances of rules of $\mathcal{S}$ where the formula the rule is applied to in the *left* rules is repeated in the premises. To get proof by pointing in our system, we only have to give boxed versions for the remaining rules.

Always

Deriving the rule for proof by pointing for $\Box$ is straightforward:

$$\frac{\boxed{A}, \Box A, \Gamma \vdash C}{\Box\boxed{A}, \Gamma \vdash C}$$

We reach the formula A in $\Box A$ by selecting A. The right $\Box$ rule can also be boxed as follows:

$$\frac{\Box\Gamma \vdash \boxed{A}}{\Box\Gamma \vdash \Box\boxed{A}}$$

Because the context has to contain formulas of a certain form, by adding this proof by pointing rule, the properties of termination and determinism are preserved, but we lose the property that we can reach any formula by selecting it. To recapture this property, we use the fact that there exists a canonical way of transforming any context into a $\Box$ context by removing assumptions that don't have $\Box$ as their outermost operator with the *weaken* rule. For example:

$$A, \Box B, C, \Box D \vdash \Box\underline{E} \;\rightarrow\; \Box B, \Box D \vdash \Box\underline{E} \;\rightarrow\; \Box B, \Box D \vdash E$$

In the following we will represent the combination of *weaken* and the application of this rule as a single step of the algorithm:

$$A, \Box B, C, \Box D \vdash \Box\underline{E} \;\rightarrow\; \Box B, \Box D \vdash E$$

With these two rules, we can begin to prove some basic properties.

Example 1: $\vdash \Box x \supset x$
Proof: Click on x in the left part of the implication.

$$\vdash \Box\underline{x} \supset x \;\rightarrow\; \Box\underline{x} \vdash x \;\rightarrow\; \overline{x, \Box x \vdash x}$$

Example 2: $\vdash \Box x \supset \Box\Box x$
Proof: Click on $\Box x$ in the right part of the implication.

$$\vdash \Box x \supset \Box\Box\underline{x} \;\rightarrow\; \Box x \vdash \Box\Box\underline{x} \;\rightarrow\; \overline{\Box x \vdash \Box x}$$

Example 3: $\vdash \Box(x \wedge y) \supset \Box x \wedge \Box y$
Proof: Click on x in the right part of the implication.

$$\vdash \Box(x \wedge y) \supset \Box\underline{x} \wedge \Box y \;\rightarrow\; \Box(x \wedge y) \vdash \Box\underline{x} \wedge \Box y \;\rightarrow$$

$$\begin{array}{l} \Box(x \wedge y) \vdash \Box\underline{x} \\ \Box(x \wedge y) \vdash \Box y \end{array} \;\rightarrow\; \begin{array}{l} \Box(x \wedge y) \vdash x \\ \Box(x \wedge y) \vdash \Box y \end{array}$$

The first goal is solved by selecting the x of the assumption.

$$\Box(\underline{x} \wedge y) \vdash x \;\rightarrow\; \underline{x} \wedge y, \Box(x \wedge y) \vdash x \;\rightarrow\; \overline{x, y, x \wedge y, \Box(x \wedge y) \vdash x}$$

For the second goal we have to follow the same path, first select y in the goal:

$$\Box(x \wedge y) \vdash \Box\underline{y} \;\rightarrow\; \Box(x \wedge y) \vdash y$$

and then in the assumption.

$$\Box(x \wedge \underline{y}) \vdash y \;\rightarrow\; x \wedge \underline{y}, \Box(x \wedge y) \vdash y \;\rightarrow\; \overline{x, y, x \wedge y, \Box(x \wedge y) \vdash y}$$

Eventually

The $\Diamond$ left rule gives the following boxed rules.

$$\frac{A_1, \ldots, \Diamond A_n, \Box\Gamma \vdash \Diamond C \;\cdots\; \Diamond A_1, \ldots, \boxed{A_i}, \ldots, \Diamond A_n, \Box\Gamma \vdash \Diamond C \;\cdots\; \Diamond A_1, \ldots, A_n, \Box\Gamma \vdash \Diamond C}{\Diamond A_1, \ldots, \Diamond \boxed{A_i}, \ldots, \Diamond A_n, \Box\Gamma \vdash \Diamond C}$$

$$\frac{A_1, \ldots, \Diamond A_n, \Box\Gamma \vdash \bot \;\cdots\; \Diamond A_1, \ldots, \boxed{A_i}, \ldots, \Diamond A_n, \Box\Gamma \vdash \bot \;\cdots\; \Diamond A_1, \ldots, A_n, \Box\Gamma \vdash \bot}{\Diamond A_1, \ldots, \Diamond \boxed{A_i}, \ldots, \Diamond A_n, \Box\Gamma \vdash \bot}$$

With these two rules, it is easy to see that we can reach any subterm in an assumption with $\Diamond$ as its outermost operator. The problem of having a context with only $\Diamond$ and $\Box$

assumptions is solved by the *weaken* rule as before. In addition, when the conclusion of the sequent is not a $\Diamond$ formula we can always apply the $\bot$ *right* rule:

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash C}$$

Here is a simple example:

$$A, \Diamond \underline{B}, C, \Box D \vdash E \;\rightarrow\; \Diamond \underline{B}, \Box D \vdash E \;\rightarrow\; \Diamond \underline{B}, \Box D \vdash \bot \;\rightarrow\; B, \Box D \vdash \bot$$

As for $\Box$, in the following we will merge applications of *weaken*, $\bot$ *right*, and $\Diamond$*left* into a single step of the algorithm.

The $\Diamond$*right* rule has a direct correspondence to the boxed rule:

$$\frac{\Gamma \vdash \boxed{C}}{\Gamma \vdash \Diamond \boxed{C}}$$

With these rules, we can prove the temporal properties that are dual to those of the previous section.

Example 4: $\vdash x \supset \Diamond x$

Proof: Click on x in the right part of the implication.

$$\vdash x \supset \Diamond \underline{x} \;\rightarrow\; x \vdash \Diamond \underline{x} \;\rightarrow\; \overline{x \vdash x}$$

Example 5: $\vdash \Diamond\Diamond x \supset \Diamond x$

Proof: Click on $\Diamond x$ in the left part of the implication.

$$\vdash \Diamond\Diamond \underline{x} \supset \Diamond x \;\rightarrow\; \Diamond\Diamond \underline{x} \vdash \Diamond x \;\rightarrow\; \overline{\Diamond x \vdash \Diamond x}$$

Example 6: $\vdash \Diamond x \vee \Diamond y \supset \Diamond(x \vee y)$

Proof: Click on x in the left part of the implication.

$$\vdash \Diamond \underline{x} \vee \Diamond y \supset \Diamond(x \vee y) \;\rightarrow\; \Diamond \underline{x} \vee \Diamond y \vdash \Diamond(x \vee y) \;\rightarrow$$
$$\begin{array}{ll} \Diamond \underline{x}, \Diamond x \vee \Diamond y \vdash \Diamond(x \vee y) & \\ \Diamond y, \Diamond x \vee \Diamond y \vdash \Diamond(x \vee y) & \end{array} \rightarrow \begin{array}{l} x \vdash \Diamond(x \vee y) \\ \Diamond y, \Diamond x \vee \Diamond y \vdash \Diamond(x \vee y) \end{array}$$

Note that the extra assumptions disappear in the first goal of the last step due to an application of *weaken* before applying $\Diamond$*left*. The first goal is solved by a selection on the x of the conclusion.

$$x \vdash \Diamond(\underline{x} \vee y) \;\rightarrow\; x \vdash \underline{x} \vee y \;\rightarrow\; \overline{x \vdash x}$$

The second goal is solved by two selections on y, first the one of the first assumption:

$$\Diamond \underline{y} \vdash \Diamond(x \vee y) \;\rightarrow\; y \vdash \Diamond(x \vee y)$$

then the one of the conclusion.

$$y \vdash \Diamond(x \vee \underline{y}) \;\rightarrow\; y \vdash x \vee \underline{y} \;\rightarrow\; \overline{y \vdash y}$$

Conversions

In S4.3, for any formula A, $\neg\Diamond A$ is equivalent to $\Box\neg A$ and $\neg\Box A$ is equivalent to $\Diamond\neg A$. We have found it useful in practice to replace a formula of one of these forms with its equivalent during proof construction. In $\mathcal{S}$, it is possible to derive rules that perform this operation on a formula of any one of these four forms in the assumptions or on the right of a sequent. Two examples are as follows:

$$\frac{\Box\neg A, \Gamma \vdash C}{\neg\Diamond A, \Gamma \vdash C} \qquad \frac{\Diamond\neg A, \Gamma \vdash C}{\neg\Box A, \Gamma \vdash C}$$

These are examples of rules where it is impossible to directly use proof by pointing; they deal with *transforming* rather than breaking down. For this reason, we treat them as terminal rules (rules with no box in the premises) and use the principle of *point and shoot* to trigger them. Point and shoot simply allows multiple terminal rules by having multiple kind of selections. Graphically we differentiate terminal rules by indexing the box with a key. Operationally the user simultaneously selects the subterm and strikes the key. The key indicates what rule to apply, and possibly what rule to attempt after applying the desired rule (the *shoot* operation). In this case, the shoot rule is an attempt to apply *initial* to complete the proof:

$$\boxed{\text{A}}, \Gamma \vdash A \;\rightarrow\; \overline{A, \Gamma \vdash A}$$

$$A, \Gamma \vdash \boxed{\text{A}} \;\rightarrow\; \overline{A, \Gamma \vdash A}$$

We can now add our *shift* selection:

$$\boxed{\neg\Diamond A}^{s}, \Gamma \vdash C \;\rightarrow\; \Box\neg A, \Gamma \vdash C$$

$$\boxed{\neg\Box A}^{s}, \Gamma \vdash C \;\rightarrow\; \Diamond\neg A, \Gamma \vdash C$$

We illustrate these rules with two examples:
Example 7: $\vdash \neg\Diamond\neg x \supset \neg\neg\Box x$
Proof: Click on $\neg\Box x$ in the right part of the implication with the *shift* selection.

$$\vdash \neg\Diamond\neg x \supset \neg\underline{\neg\Box x}^{s} \;\rightarrow\; \neg\Diamond\neg x \vdash \neg\underline{\neg\Box x}^{s} \;\rightarrow\; \neg\Diamond\neg x, \underline{\neg\Box x}^{s} \vdash \bot \;\rightarrow\; \neg\Diamond\neg x, \Diamond\neg x \vdash \bot$$

Click on $\Diamond\neg x$ in the first assumption.

$$\neg\underline{\Diamond\neg x}, \Diamond\neg x \vdash \bot \;\rightarrow\; \overline{\neg\Diamond\neg x, \Diamond\neg x \vdash \Diamond\neg x}$$

Example 8: $\vdash \neg\Box\neg x \supset \neg\neg\Diamond x$

Proof: Click on $\neg\Diamond x$ in the right part of the implication with the *shift* selection.

$$\vdash \neg\Box\neg x \supset \neg\underline{\neg\Diamond x}^{s} \rightarrow \neg\Box\neg x \vdash \neg\underline{\neg\Diamond x}^{s} \rightarrow \neg\Box\neg x, \underline{\neg\Diamond x}^{s} \vdash \bot \rightarrow \neg\Box\neg x, \Box\neg x \vdash \bot$$

Click on the $\Box\neg x$ in the first assumption.

$$\neg\underline{\Box\neg x}, \Box\neg x \vdash \bot \rightarrow \overline{\neg\Box\neg x, \Box\neg x \vdash \Box\neg x}$$

Excluded Middle

The method proposed in [2] for introducing excluded middle into the point and shoot algorithm is by a higher-order theorem:

$$\forall P.\, P \vee \neg P$$

Then the *cut* rule is used to extend the possibility of adding a theorem to the assumptions. Operationally, the user selects some subformula A, and then clicks on the $\neg P$ of the *excl-mid* rule. $A \vee \neg A$ will be added as an assumption. Given a goal $\Gamma \vdash C$ and a theorem T we have:

$$\Gamma \vdash C \rightarrow \boxed{T}, \Gamma \vdash C$$

Using this rule, we can refine the two previous examples.

Example 9: $\neg\Diamond\neg x \vdash \Box x$

Proof: Click on the $\neg P$ of excluded middle with $\Box x$ as a witness and the *shift* selection:

$$\neg\Diamond\neg x \vdash \Box x \rightarrow \Box x \vee \underline{\neg\Box x}^{s}, \neg\Diamond\neg x \vdash \Box x \rightarrow$$

$$\begin{array}{r} \overline{\Box x, \Box x \vee \neg\Box x, \neg\Diamond\neg x \vdash \Box x} \\ \underline{\neg\Box x}^{s}, \Box x \vee \neg\Box x, \neg\Diamond\neg x \vdash \Box x \end{array} \rightarrow \Diamond\neg x, \Box x \vee \neg\Box x, \neg\Diamond\neg x \vdash \Box x$$

Click on the $\Diamond\neg x$ in the third assumption, which applies the $\bot$ *right* rule, followed by the $\neg$*left* rule:

$$\Diamond\neg x, \Box x \vee \neg\Box x, \neg\underline{\Diamond\neg x} \vdash \Box x \rightarrow \overline{\Diamond\neg x, \Box x \vee \neg\Box x, \neg\Diamond\neg x \vdash \Diamond\neg x}$$

Example 10: $\neg\Box\neg x \vdash \Diamond x$

Proof: Click on the $\neg P$ of excluded middle with $\Diamond x$ as a witness and the *shift* selection.

$$\neg\Box\neg x \vdash \Diamond x \rightarrow \Diamond x \vee \underline{\neg\Diamond x}^{s}, \neg\Box\neg x \vdash \Diamond x \rightarrow$$

$$\begin{array}{r}\overline{\Diamond x, \Diamond x \vee \neg\Diamond x, \neg\Box\neg x \vdash \Diamond x} \\ \underline{\neg\Diamond x}^{s}, \Diamond x \vee \neg\Diamond x, \neg\Box\neg x \vdash \Diamond x\end{array} \rightarrow \Box\neg x, \Diamond x \vee \neg\Diamond x, \neg\Box\neg x \vdash \Diamond x$$

Click on the $\Box\neg x$ in the third assumption.

$$\Box\neg x, \Diamond x \vee \neg\Diamond x, \neg\underline{\Box\neg x} \vdash \Diamond x \rightarrow \overline{\Box\neg x, \Diamond x \vee \neg\Diamond x, \neg\Box\neg x \vdash \Box\neg x}$$

Note that the last rule is $\neg$*right*. It includes an implicit application of $\bot$ *right*.

Weakening

Finally the last rule we add concerns *weaken*. In [2], there was no explicit way of applying this rule since it was always done implicitly just before completing the proof with *initial*. In our system, having an explicit *weaken* is important as the $\Diamond$ *left* rule generates as many subgoals as assumptions with $\Diamond$ as outermost operator. Applying *weaken* or not may change the structure of the proof. We simply implement *weaken* as a terminal rule:

$$\boxed{A}^{d}, \Gamma \vdash C \rightarrow \Gamma \vdash C$$

4.2 Examples

We have already given some examples in the previous section. We complement them with the proofs of two other classic properties.

Example 11: $\vdash \Box(x \supset y) \supset \Diamond x \supset \Diamond y$

Proof: Click on x in the right part of the top implication.

$$\vdash \Box(x \supset y) \supset \Diamond\underline{x} \supset \Diamond y \rightarrow \Box(x \supset y) \vdash \Diamond\underline{x} \supset \Diamond y \rightarrow$$

$$\Box(x \supset y), \Diamond\underline{x} \vdash \Diamond y \rightarrow \Box(x \supset y), x \vdash \Diamond y$$

Click on x in the right part of the first assumption.

$$\Box(\underline{x} \supset y), x \vdash \Diamond y \rightarrow \underline{x} \supset y, \Box(\underline{x} \supset y), x \vdash \Diamond y \rightarrow \begin{array}{l}\overline{x \supset y, \Box(x \supset y), x \vdash x} \\ y, x \supset y, \Box(x \supset y), x \vdash \Diamond y\end{array}$$

Only one subgoal is left, we can solve it by selecting the y in the conclusion.

$$y, x \supset y, \Box(x \supset y), x \vdash \Diamond\underline{y} \rightarrow \overline{y, x \supset y, \Box(x \supset y), x \vdash y}$$

Example 12: $\vdash \Diamond(x \vee y) \supset \Diamond x \vee \Diamond y$

Proof: Click on $\Diamond x \vee \Diamond y$ in the right part of the top implication.

$$\vdash \Diamond(x \vee y) \supset \underline{\Diamond x \vee \Diamond y} \rightarrow \Diamond(x \vee y) \vdash \Diamond x \vee \Diamond y$$

Click on the $\neg P$ of excluded middle with $\Diamond x$ as a witness and the *shift* selection.

$$\Diamond(x \vee y) \vdash \Diamond x \vee \Diamond y \rightarrow \Diamond x \vee \underline{\neg\Diamond x}^s, \Diamond(x \vee y) \vdash \Diamond x \vee \Diamond y \rightarrow$$

$$\begin{array}{l} \Diamond x, \Diamond x \vee \neg\Diamond x, \Diamond(x \vee y) \vdash \Diamond x \vee \Diamond y \\ \underline{\neg\Diamond x}^s, \Diamond x \vee \neg\Diamond x, \Diamond(x \vee y) \vdash \Diamond x \vee \Diamond y \end{array} \rightarrow \begin{array}{l} \Diamond x, \Diamond x \vee \neg\Diamond x, \Diamond(x \vee y) \vdash \Diamond x \vee \Diamond y \\ \Box\neg x, \Diamond x \vee \neg\Diamond x, \Diamond(x \vee y) \vdash \Diamond x \vee \Diamond y \end{array}$$

The proof of the first subgoal is trivial.

$$\Diamond x, \Diamond x \vee \neg\Diamond x, \Diamond(x \vee y) \vdash \underline{\Diamond x} \vee \Diamond y \rightarrow \overline{\Diamond x, \Diamond x \vee \neg\Diamond x, \Diamond(x \vee y) \vdash \Diamond x}$$

For the second goal, we first select $\Diamond y$.

$$\Box\neg x, \Diamond x \vee \neg\Diamond x, \Diamond(x \vee y) \vdash \Diamond x \vee \underline{\Diamond y} \rightarrow \Box\neg x, \Diamond x \vee \neg\Diamond x, \Diamond(x \vee y) \vdash \Diamond y$$

Now we do a case analysis selecting x of the third assumption.

$$\Box\neg x, \Diamond x \vee \neg\Diamond x, \Diamond(\underline{x} \vee y) \vdash \Diamond y \rightarrow \Box\neg x, \underline{x} \vee y \vdash \Diamond y \rightarrow \begin{array}{l} x, \Box\neg x, x \vee y \vdash \Diamond y \\ y, \Box\neg x, x \vee y \vdash \Diamond y \end{array}$$

Both cases are trivial:

$$x, \Box\neg\underline{x}, x \vee y \vdash \Diamond y \rightarrow \neg\underline{x}, x, \Box\neg x, x \vee y \vdash \Diamond y \rightarrow \overline{\neg x, x, \Box\neg x, x \vee y \vdash x}$$

$$y, \Box\neg x, x \vee y \vdash \Diamond\underline{y} \rightarrow \overline{y, \Box\neg x, x \vee y \vdash y}$$

5 Proof Presentation

In the previous section, proof by pointing illustrated a simple way of locally constructing a proof. In this section, we discuss the display of overall proofs. There are two important reasons to do so:

1. To present the final result. The final proof "explains" why the fact holds.
2. To aid in the construction of the proof. Displaying the incomplete proof gives a global view of the proof process.

A natural solution is to represent the sequent proof that we build as a tree. From our experiments we have found that displaying trees doesn't scale up. Proof trees rapidly become unmanageable not only because of their length but also because of their width. In [4], an alternative is described that proposes a textual presentation of natural deduction proofs in a pseudo-natural language. In natural deduction the combination of the inferences only deals with a single formula: the conclusion of the sequent. Generating a text in pseudo-natural language is then made easier.

The text is generated by translation. With each rule of the natural deduction system is associated a textual pattern. For example, the two translation rules for $\wedge intro$ and $\supset$ *intro* are the following:

$$\dfrac{\begin{array}{cc}\Pi_1 & \Pi_2 \\ A & B\end{array}}{A \wedge B} \quad \triangleright \quad \begin{array}{l} \text{- } \left[\begin{array}{c}\Pi_1 \\ A\end{array}\right] \\ \text{- } \left[\begin{array}{c}\Pi_2 \\ B\end{array}\right] \\ \text{Altogether we have } A \wedge B\end{array}$$

$$\dfrac{\begin{array}{c}\Pi \\ B\end{array}}{A \supset B} \quad \triangleright \quad \begin{array}{l} \text{Assume } A \text{ (i)} \\ \left[\begin{array}{c}\Pi \\ B\end{array}\right] \\ \text{We have proved } A \supset B\end{array}$$

where recursive calls are marked with square brackets. In addition to the direct application of such rules, a set of optimizations is performed on the text to remove irrelevant information such as unused assumption numbers or immediate references. As for proof by pointing, for $\mathcal{S}$ we have to extend the rules presented for the classical logic in [4] to the temporal rules.

Always

We first give the $\Box elim$ rule:

$$\dfrac{\begin{array}{c}\Pi \\ \Box A\end{array}}{A} \quad \triangleright \quad \begin{array}{l}\left[\begin{array}{c}\Pi \\ \Box A\end{array}\right] \\ \text{In particular } A\end{array}$$

The $\Box intro$ rule is a bit more complicated. The general layout is given in Figure 7. Special cases have been developed when n is 0 or 1:

$$\dfrac{\begin{array}{c}\Pi \\ A\end{array}}{\Box A} \quad \triangleright \quad \begin{array}{l}\left[\begin{array}{c}\Pi \\ A\end{array}\right] \\ \text{So } \Box\ A\end{array}$$

$$\frac{\begin{matrix}\Pi_1 & \Pi \\ \Box B_1 & A\end{matrix}}{\Box A} \quad \triangleright \quad \begin{matrix}\begin{bmatrix}\Pi_1 \\ \Box B_1\end{bmatrix} \\ \text{In the context: } \Box B_1(h_1) \\ \begin{bmatrix}\Pi \\ A\end{bmatrix} \\ \text{So we deduce } \Box A\end{matrix}$$

Example 1: $\vdash \Box x \supset x$

Proof: Assume $\Box x$ (1)
By (1) we have $\Box x$
In particular x

Example 2: $\vdash \Box x \supset \Box\Box x$

Proof: Assume $\Box x$
In the context $\Box x(1)$
By (1) we have $\Box x$
So we deduce $\Box\Box x$

Example 3: $\vdash \Box(x \wedge y) \supset \Box x \wedge \Box y$

Proof: Assume $\Box(x \wedge y)$
In the context $\Box (x \wedge y)$ (1)
By (1) we have $\Box (x \wedge y)$
In particular we have $x \wedge y$
We have x
-So we deduce $\Box x$
In the context $\Box (x \wedge y)$ (1)
By (1) we have $\Box (x \wedge y)$
In particular we have $x \wedge y$
We have y
-So we deduce $\Box y$
Altogether we have $\Box x \wedge \Box y$

Eventually

The general rule for $\Diamond$*elim* is given in Figure 7. Special cases can be easily derived. For example, when there is no $\Box$ assumption and only one $\Diamond$ assumption, we use:

$$\dfrac{\begin{array}{cc}\Pi_1 & \Pi_2\\ \Diamond A & \Diamond C\end{array}}{\Diamond C} \quad \triangleright \quad \begin{array}{l}\left[\begin{array}{c}\Pi_1\\ \Diamond A\end{array}\right]\\ \text{If we have } A \text{ (i)}\\ \left[\begin{array}{c}\Pi_2\\ \Diamond C\end{array}\right]\\ \text{So we deduce } \Diamond C\end{array}$$

For $\Diamond$*intro*, the rule is much simpler to explain:

$$\dfrac{\begin{array}{c}\Pi\\ A\end{array}}{\Diamond A} \quad \triangleright \quad \begin{array}{l}\left[\begin{array}{c}\Pi\\ A\end{array}\right]\\ \text{Obviously we have } \Diamond A\end{array}$$

Example 4: $\vdash x \supset \Diamond x$

Proof: Assume x (1)
By (1) we have x
Obviously we have $\Diamond x$

Example 5: $\vdash \Diamond\Diamond x \supset \Diamond x$

Proof: Assume $\Diamond\Diamond x$
If we have $\Diamond x$ (1)
By (1) we have $\Diamond x$
So we deduce $\Diamond x$

Example 6: $\vdash \Diamond x \vee \Diamond y \supset \Diamond(x \vee y)$

Proof: Assume $\Diamond x \vee \Diamond y$
So we have two cases
- Suppose $\Diamond x$
  If we have x (1)
  By (1) we have x
  Obviously we have $x \vee y$
  Obviously we have $\Diamond(x \vee y)$
  So we deduce $\Diamond(x \vee y)$
- Suppose $\Diamond y$
  If we have y (1)
  By (1) we have y
  Obviously we have $x \vee y$
  Obviously we have $\Diamond(x \vee y)$
  So we deduce $\Diamond(x \vee y)$

We have $\Diamond(x \vee y)$ in both cases, so $\Diamond(x \vee y)$

Note that the two subproofs in this example are similar. We could optimize the text to proof procedure so that it would note the similarity, and avoid writing out the details of the second case. Although this case is simple, the general problem of finding similarities while abstracting from differences is a difficult one, but one that must be addressed if larger proofs are to be readable. Note that it is often possible to replace duplicate proofs by lemmas.

Conversions

Conversions are handled by the simple concatenation of the converted term.

$$\frac{\begin{array}{c}\Pi\\ \neg\Box A\end{array}}{\Diamond\neg A} \quad \triangleright \quad \begin{bmatrix}\Pi\\ \neg\Box A\end{bmatrix}, \text{ so } \Diamond\neg A$$

$$\frac{\begin{array}{c}\Pi\\ \neg\Diamond\end{array}}{\Box\neg A} \quad \triangleright \quad \begin{bmatrix}\Pi\\ \neg\Diamond A\end{bmatrix}, \text{ so } \Box\neg A$$

6 An Example

Now that we have defined the two principles (proof by pointing and textual presentation), we are going to merge them into a single environment. Using the mixed system $\mathcal{M}$, we can simultaneous build the proof in sequent style while showing the natural deduction equivalent. We illustrate how the combination works with the proof of the property:

$$\Box(\Box x \supset y) \vee \Box(\Box y \supset x).$$

This proof is done by contradiction; as we don't have the DeMorgan laws we need to apply excluded middle to each of the two components of the disjunction.

Using *Excluded Middle*, $\Box(\Box x \supset y) \vee \neg\Box(\Box x \supset y)$
So we have two cases:

- Suppose $\Box(\Box x \supset y)$ (1)
  Obviously $\Box(\Box x \supset y) \vee \Box(\Box y \supset x)$
- Suppose $\boxed{\neg\Box(\Box x \supset y)}$ (2)
  Using *Excluded Middle*, $\Box(\Box y \supset x) \vee \neg\Box(\Box y \supset x)$
  So we have two cases:

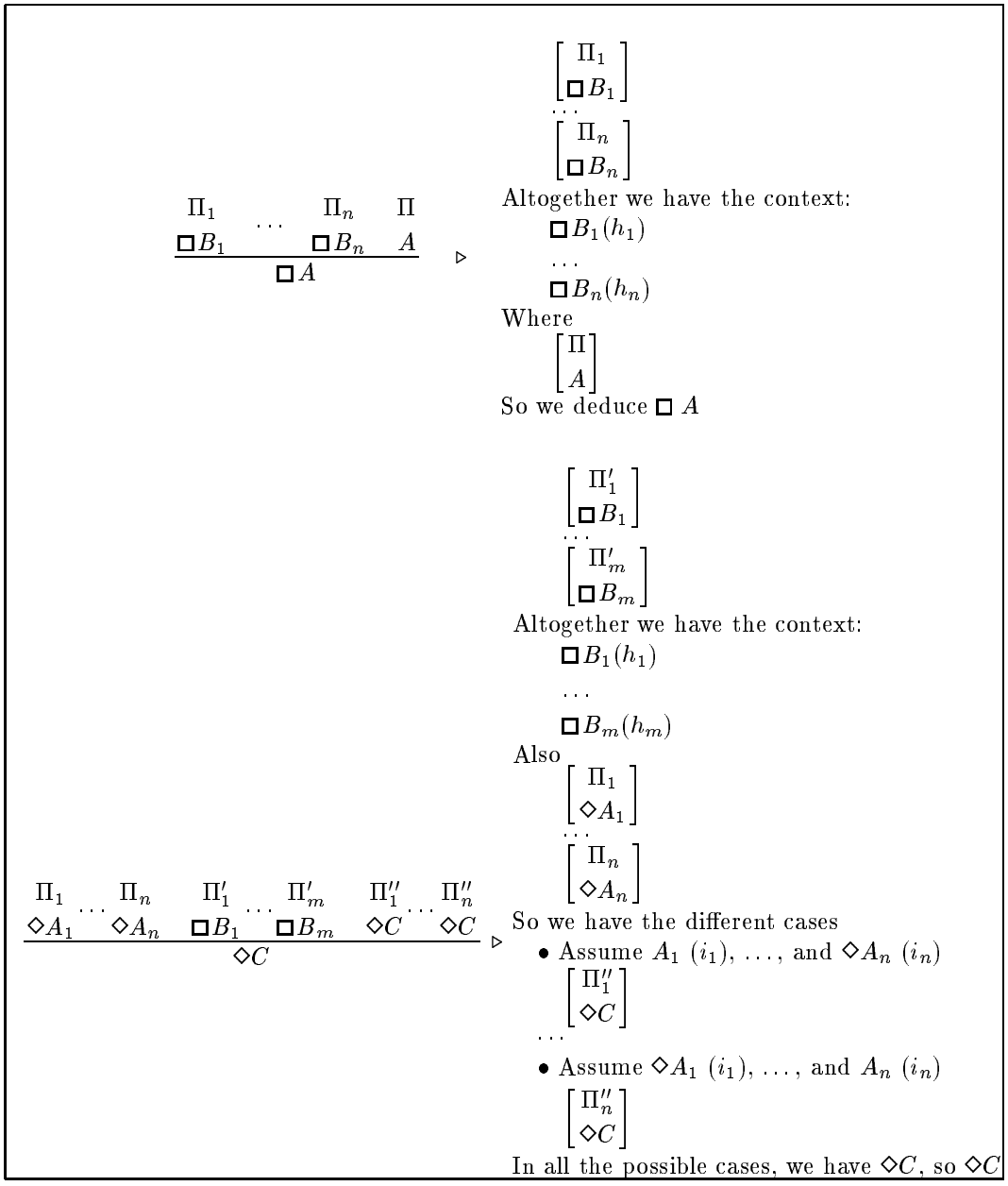

Figure 7: Textual rules for always introduction and eventually elimination

- Suppose $\Box(\Box y \supset x)$ (3)
  Obviously $\Box(\Box x \supset y) \vee \Box(\Box y \supset x)$
- Suppose $\boxed{\neg\Box(\Box y \supset x)}$ (4)
  Prove: $\boxed{\Box(\Box x \supset y) \vee \neg\Box(\Box x \supset y)}$

We have $\Box(\Box x \supset y) \vee \Box(\Box y \supset x)$ in both cases (3) and (4)
We have $\Box(\Box x \supset y) \vee \Box(\Box y \supset x)$ in both cases (1) and (2)

We are left with one goal, under the two assumptions (2) and (4). They are inside a box to show that the user can select inside them. The next step is to transform the assumptions into $\Diamond$ assumptions and apply the $\Diamond$*elim* rule.

Using *Excluded Middle*, $\Box(\Box x \supset y) \vee \neg\Box(\Box x \supset y)$
So we have two cases:

- Suppose $\Box(\Box x \supset y)$ (1)
  Obviously $\Box(\Box x \supset y) \vee \Box(\Box y \supset x)$
- Suppose $\neg\Box(\Box x \supset y)$ (2)
  Using *Excluded Middle*, $\Box(\Box y \supset x) \vee \neg\Box(\Box y \supset x)$
  So we have two cases:
  - Suppose $\Box(\Box y \supset x)$ (3)
    Obviously $\Box(\Box x \supset y) \vee \Box(\Box y \supset x)$
  - Suppose $\neg\Box(\Box y \supset x)$ (4)
    We have:
    By (2), $\neg\Box(\Box x \supset y)$, so $\Diamond\neg(\Box x \supset y)$
    By (3), $\neg\Box(\Box y \supset x)$, so $\Diamond\neg(\Box y \supset x)$
    So we have the different cases:
    - Assume $\boxed{\neg(\Box x \supset y)}$ (5) and $\boxed{\Diamond\neg(\Box y \supset x)}$ (6)
      Prove: $\boxed{\text{a contradiction}}$
    - Assume $\Diamond\neg(\Box x \supset y)$ (6) and $\neg(\Box y \supset x)$ (7)
      Prove: a contradiction

    In all possible cases, we have a contradiction, so $\Box(\Box x \supset y) \vee \Box(\Box y \supset x)$
  We have $\Box(\Box x \supset y) \vee \Box(\Box y \supset x)$ in both cases (3) and (4)

We have $\Box(\Box x \supset y) \vee \Box(\Box y \supset x)$ in both cases (1) and (2)

We now have two goals, that are symmetrical, so in the following we concentrate on the current one.

- Assume $\boxed{\neg(\Box x \supset y)}$ (5) and $\boxed{\Diamond\neg(\Box y \supset x)}$ (6)
  Prove: $\boxed{\text{a contradiction}}$

To prove a contradiction, we first select the y of the assumption 5.

- Assume $\boxed{\neg(\Box x \supset y)}$ (5) and $\boxed{\Diamond\neg(\Box y \supset x)}$ (6)
  Assume $\boxed{\Box x}$ (7)
  Prove $\boxed{y}$
  We have proved $\Box x \supset y$
  By (5), there is a contradiction

Assumptions 6 and 7 are clearly contradictory. To show the contradiction we select the x in assumption 6.

- Assume $\neg(\Box x \supset y)$ (5) and $\Diamond\neg(\Box y \supset x)$ (6)
  Assume $\Box x$ (7)
  In the context $\boxed{\Box x(8)}$
  If we have $\boxed{\neg(\Box y \supset x)(9)}$
  Assume $\boxed{\Box y}$ (10)
  Prove $\boxed{x}$
  We have proved $\Box y \supset x$
  By (9) there is a contradiction
  By (6) we deduce a contradiction, so y
  We have proved $\Box x \supset y$
  By (5), there is a contradiction

We end the proof by selecting the x of assumption 8.

- Assume $\neg(\Box x \supset y)$ (5) and $\Diamond\neg(\Box y \supset x)$ (6)
  Assume $\Box x$ (7)
  In the context $\Box x(8)$
  If we have $\neg(\Box y \supset x)(9)$
  Assume $\Box y$
  By (8) we have x
  We have proved $\Box y \supset x$

By (9) there is a contradiction
By (6) we deduce a contradiction, so y
We have proved $\Box x \supset y$
By (5), there is a contradiction

Figure 8 gives the overall proof that can be performed with 10 clicks.

7 Conclusion

The system we have described has a very simple and convivial user-interface with the following properties:

- The proof process is presented as the refinement of pseudo-English text.
- All the proof steps are input simply by using the mouse to make selections on formulas in this text.

The temporal calculus we use has been carefully designed to be as natural as possible. This calculus has been proved sound and complete. Our implementation has benefited from the use of a generic theorem prover. We were able to quickly and easily specify the inference rules and obtain a tactic-style theorem prover for goal-directed proof using these rules. The interface was also built with a generic toolkit in which all the features of the user-interface (window, layout, interaction) are handled by a separate process, and thus we were able to reuse a large part of an existing interface.

So far, we have considered S4.3. The techniques presented here should extend fairly directly to various related and more expressive logics. For example, extending the interface and explanation capabilities to logics with additional operators such as O (next) and U (until) should be straightforward.

In [1], the Isabelle theorem prover [11] is used to implement a class of modal logics that includes many logics similar to S4.3. The inference systems used in this work are natural deduction systems in which formulas are explicitly labelled with possible worlds, using a Kripke-style semantics. Isabelle contains a specification language that is essentially a subset of the higher-order logic implemented in λProlog. Thus, the Isabelle specifications of labelled deduction systems can be directly mapped to specifications in λProlog and used to implement a simple tactic theorem prover similar to the one presented in this paper. The techniques of proof by pointing and point and shoot could also be adapted to this kind of inference system. It would

be interesting to see if the possible world annotations could be used to improve or provide alternate explanations. Initial work towards this goal can be found in [7], where a labelled sequent inference system similar to the natural deduction systems in [1] is used, and a simple mapping of inference rules to text is given. Conversely, our S4.3 specification could be specified directly in Isabelle. In doing so, we would benefit from the built-in theorem proving support in the Isabelle system, which is more extensive than what is available in our tactic theorem proving environment in λProlog.

With our current system, we are still quite far from verifying algorithms. Thus far, the system has only been used to prove simple temporal properties. In that respect, the simplicity of the user-interface makes it an ideal tool to learn and experiment with temporal logics, aiding the user in both understanding temporal logic reasoning as well as understanding proofs as they are constructed. In order to tackle more realistic problems, a necessary step is to introduce some automation in our system so that users only concentrate on the general architecture of the proof while the system automatically proves details. Incorporating the extra theorem proving power of Isabelle, or building our interface on top of Isabelle instead of λProlog would provide an important step in that direction. However, for large algorithms more significant automation is needed. Incorporating more powerful decision procedures as well as model checking are both candidates in the future extension of the system.

Model checkers are fully automatic and effective for verifying finite state automata. Much work has gone into pushing the boundaries of the size of problems that can be handled, so that such techniques have been applied successfully to the automatic verification of a large class of systems and algorithms. However, although the boundaries continue to be pushed, there will always be a limit to the size of the problems that such methods can handle. In addition, they are limited to finite spaces. By integrating such techniques within a theorem proving environment, it should be possbile to increase the class of algorithms for which verification is practical, including for example those that are parameterized by the number of components or processors, or have infinite data domains. To do so, powerful and intuitive interaction is essential.

The Stanford Temporal Prover (STeP) [9] is one system that is working towards the goal of broadening the class of algorithms that can be verified. STeP integrates a variety of diverse components. Although they are not directly connected to each other, two such components include an interactive prover and a model checker. The techniques presented here could be integrated directly into the interactive prover.

References

[1] D. Basin, S. Matthews, and L. Viganò. A modular presentation of modal logics in a logical framework. In *Proceedings of the 1995 Isabelle Users Workshop*, 1995.

[2] Y. Bertot, G. Kahn, and L. Théry. Proof by pointing. In *Theoretical Aspects of Computer Software*, volume 789 of *Lecture Notes in Computer Science*, pages 141–160, 1994.

[3] P. Borras, D. Clément, T. Despeyroux, J. Incerpi, G. Kahn, B. Lang, and V. Pascual. Centaur: the system. In *Third Symposium on Software Development Environments*, 1988. (Also appears as INRIA Report no. 777).

[4] Y. Coscoy, G. Kahn, and L. Théry. Extracting text from proofs. In *Typed Lambda Calculus and its Applications*, 1995.

[5] A. Felty. A logic program for transforming sequent proofs to natural deduction proofs. In P. Schroeder-Heister, editor, *Proceedings of the 1989 International Workshop on Extensions of Logic Programming*, pages 157–178. Springer-Verlag LNCS, 1991.

[6] A. Felty. Implementing tactics and tacticals in a higher-order logic programming language. *Journal of Automated Reasoning*, 11(1):43–81, 1993.

[7] A. Felty and G. Hager. Explaining modal logic proofs. In *Proceedings of the IEEE 1988 International Conference on Systems, Man, and Cybernetics*, Aug. 1988.

[8] R. Goré. *Cut-free Tableau and Sequent Systems for Propositional Normal Modal Logics.* PhD thesis, University of Cambridge, 1992. (Also appears as Technical Report no. 257).

[9] Z. Manna et al. STeP: the Stanford Temporal Prover. Technical Report STAN-CS-TR-94-1518, Stanford University, 94.

[10] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification.* Springer-Verlag, 1992.

[11] L. C. Paulson. *Isabelle: A Generic Theorem Prover*, volume 828 of *Lecture Note in Computer Science.* Springer-Verlag, 1994.

[12] D. Prawitz. *Natural Deduction.* Almqvist & Wiksell, Uppsala, 1965.

[13] M. E. Szabo. *The Collected Papers of Gerhard Gentzen.* North-Holland, 1969.

Using *Excluded Middle*, $\Box(\Box x \supset y) \vee \neg\Box(\Box x \supset y)$
So we have two cases:
- Suppose $\Box(\Box x \supset y)$ (1)
  Obviously $\Box(\Box x \supset y) \vee \Box(\Box y \supset x)$
- Suppose $\neg\Box(\Box x \supset y)$ (2)
  Using *Excluded Middle*, $\Box(\Box y \supset x) \vee \neg\Box(\Box y \supset x)$
  So we have two cases:
  - Suppose $\Box(\Box y \supset x)$ (3)
    Obviously $\Box(\Box x \supset y) \vee \Box(\Box y \supset x)$
  - Suppose $\neg\Box(\Box y \supset x)$ (4)
    We have:
    By (2), $\neg\Box(\Box x \supset y)$, so $\Diamond\neg(\Box x \supset y)$
    By (3), $\neg\Box(\Box y \supset x)$, so $\Diamond\neg(\Box y \supset x)$
    So we have the different cases:
    - Assume $\neg(\Box x \supset y)$ (5) and $\Diamond\neg(\Box y \supset x)$ (6)
      Assume $\Box x$ (7)
      In the context $\Box x$ (8)
      If we have $\neg(\Box y \supset x)$ (9)
      Assume $\Box y$
      By (8) we have x
      We have proved $\Box y \supset x$
      By (9) there is a contradiction
      By (6) we deduce a contradiction, so y
      We have proved $\Box x \supset y$
      By (5), there is a contradiction
    - Assume $\Diamond\neg(\Box x \supset y)$ (10) and $\neg(\Box y \supset x)$ (11)
      Assume $\Box y$ (12)
      In the context $\Box y$ (13)
      If we have $\neg(\Box x \supset y)$ (14)
      Assume $\Box x$
      By (13) we have y
      We have proved $\Box x \supset y$
      By (14) there is a contradiction
      By (10) we deduce a contradiction, so x
      We have proved $\Box y \supset x$
      By (11), there is a contradiction
    In all possible cases, we have a contradiction, so $\Box(\Box x \supset y) \vee \Box(\Box y \supset x)$
  We have $\Box(\Box x \supset y) \vee \Box(\Box y \supset x)$ in both cases (3) and (4)
We have $\Box(\Box x \supset y) \vee \Box(\Box y \supset x)$ in both cases (1) and (2)

Figure 8: A complete example

A Proofs of Theorems 2.1, 2.2, and 3.2

THEOREM 2.1. *Given a set of formulas Γ and a formula C, let Γ' and C' be Γ and C, respectively, with all occurrences of $\Diamond$ replaced by $\neg\Box\neg$. The sequent $\Gamma \vdash C$ is provable in $\mathcal{S}$ if and only if $\Gamma' \vdash C'$ (or $\Gamma' \vdash \emptyset$ when C is $\bot$) is provable in $\mathcal{S}'$.*

PROOF. We prove the following more general theorem:

Given a set of formulas Γ and formulas $C_1, \ldots, C_n$, where $n \geq 0$, let $\Gamma', C'_1, \ldots, C'_n$ be $\Gamma, C_1, \ldots, C_n$, respectively, with all occurrences of $\Diamond$ replaced by $\neg\Box\neg$. The sequent $\Gamma \vdash C_1 \vee \cdots \vee C_n$ (or $\Gamma \vdash \bot$ when n is 0) is provable in $\mathcal{S}$ if and only if $\Gamma' \vdash C'_1, \ldots, C'_n$ is provable in $\mathcal{S}'$.

Theorem 1 expresses the special case when n is 0 or 1. We first consider the forward direction. The proof is by induction on the height of the $\mathcal{S}$-proof. Most cases follow directly by the induction hypothesis and an application of the corresponding rule in $\mathcal{S}'$. We consider the remaining cases below.

Case: *initial.* Since $\bot$ cannot occur on the left in an $\mathcal{S}$-proof, we need only consider the one-node proof $\Gamma \vdash C_1 \vee \cdots \vee C_n$ where $n > 0$. Since $C_1 \vee \cdots \vee C_n \in \Gamma$, we know that $C'_1 \vee \cdots \vee C'_n \in \Gamma'$. For $i = 1, \ldots n$, we have $C'_i \vdash C'_i$ in $\mathcal{S}'$ by *initial*, and $\Gamma', C'_i \vdash C'_1, \ldots, C'_n$ by *weaken*. Thus by a series of applications of $\vee$*left*, we can deduce $\Gamma', C'_1 \vee \cdots \vee C'_n \vdash C'_1, \ldots, C'_n$.

Case: *excl-mid.* We have the one-node $\mathcal{S}$-proof $\Gamma \vdash A \vee \neg A$. We have two cases: either $n = 2$, C_1 is A, and C_2 is $\neg A$; or $n = 1$ and C_1 is $A \vee \neg A$. In either case, we build the following proof in $\mathcal{S}'$.

$$\dfrac{\dfrac{A' \vdash A'}{\Gamma', A' \vdash A'}\, weaken}{\Gamma' \vdash A', \neg A'}\, \neg right$$

For the case when $n = 2$, we are done. For the case when $n = 1$, we apply the $\vee$*right* rule to obtain an $\mathcal{S}$-proof of $\Gamma' \vdash A' \vee \neg A'$.

Case: $\vee$ *right.* The premise is of the form $\Gamma \vdash C_1 \vee \ldots \vee C'_i$ for some i, $0 < i < n$. By the induction hypothesis, $\Gamma' \vdash C'_1, \ldots, C'_i$ is provable in $\mathcal{S}'$. By *weaken*, $\Gamma' \vdash C'_1, \ldots, C'_n$ is provable in $\mathcal{S}'$.

Case: $\bot$ *right.* By the induction hypothesis, $\Gamma' \vdash \emptyset$ is provable in $\mathcal{S}'$. By *weaken*, $\Gamma' \vdash C'_1, \ldots, C'_n$ is provable in $\mathcal{S}'$.

Case: $\Box$*right.* C_1 is $\Box A$ and $n = 1$. Γ has the form $\Box A_1, \ldots, \Box A_m, \neg\Diamond\neg B_1, \ldots, \neg\Diamond\neg B_p$ where $m, p \geq 0$. $\Box A'_1, \ldots, \Box A'_m, \neg\neg\Box\neg\neg B'_1, \ldots, \neg\neg\Box\neg\neg B'_p \vdash A'$ is provable in $\mathcal{S}'$ by

the induction hypothesis . By *weaken*, the following is provable:

$$\Box A'_1, \ldots, \Box A'_m, \neg\neg\Box\neg\neg B'_1, \ldots, \neg\neg\Box\neg\neg B'_p, \Box\neg\neg B'_1, \ldots, \Box\neg\neg B'_p \vdash A'.$$

For any formula B, the sequent $B \vdash \neg\neg B$ is provable by $\neg$*left* followed by $\neg$*right*. By taking B to be $\Box\neg\neg B'_1$ and applying *weaken*, the following holds:

$$\Box A'_1, \ldots, \Box A'_m, \neg\neg\Box\neg\neg B'_2, \ldots, \neg\neg\Box\neg\neg B'_p, \Box\neg\neg B'_1, \ldots, \Box\neg\neg B'_p \vdash \neg\neg\Box\neg\neg B'_1.$$

With this sequent as the left premise and the sequent above as the right, we can apply *cut* to get the following provable sequent:

$$\Box A'_1, \ldots, \Box A'_m, \neg\neg\Box\neg\neg B'_2, \ldots, \neg\neg\Box\neg\neg B'_p, \Box\neg\neg B'_1, \ldots, \Box\neg\neg B'_p \vdash A'.$$

By repeated applications of *weaken* and *cut*, the following sequent is provable:

$$\Box A'_1, \ldots, \Box A'_m, \Box\neg\neg B'_1, \ldots, \Box\neg\neg B'_p \vdash A'.$$

By applying $\Box$*right*, the following also holds:

$$\Box A'_1, \ldots, \Box A'_m, \Box\neg\neg B'_1, \ldots, \Box\neg\neg B'_p \vdash \Box A'.$$

Now, we can repeatedly apply $\neg$*right* and $\neg$*left* to obtain a proof of the following sequent.

$$\Box A'_1, \ldots, \Box A'_m, \neg\neg\Box\neg\neg B'_1, \ldots, \neg\neg\Box\neg\neg B'_p \vdash \Box A'.$$

Case: $\Diamond$*left*. The formula on the right of the sequent at the root is either $\bot$ (and thus $n = 0$) or $\Diamond C$ (and thus $n = 1$). We consider the latter case. The former is similar and slightly simpler. Γ has the form $\Diamond D_1, \ldots, \Diamond D_r, \Box A_1, \ldots, \Box A_m, \neg\Diamond\neg B_1, \ldots, \neg\Diamond\neg B_p$ where $r > 0$ and $m, p \geq 0$. By the induction hypothesis applied to the first premise, the following sequent is provable in $\mathcal{S}'$.

$$D'_1, \neg\Box\neg D'_2, \ldots, \neg\Box\neg D'_r, \Box A'_1, \ldots, \Box A'_m, \neg\neg\Box\neg\neg B'_1, \ldots, \neg\neg\Box\neg\neg B'_p \vdash \neg\Box\neg C'.$$

By repeatedly applying $\neg$*left* and $\neg$*right*, the following holds:

$$\neg\neg\Box\neg C', \Box A'_1, \ldots, \Box A'_m, \neg\neg\Box\neg\neg B'_1, \ldots, \neg\neg\Box\neg\neg B'_p \vdash \neg D'_1, \neg\neg\Box\neg D'_2, \ldots, \neg\neg\Box\neg D'_r.$$

By applying *weaken* and *cut* as in the previous case, the following can be shown to hold:

$$\Box\neg C', \Box A'_1, \ldots, \Box A'_m, \Box\neg\neg B'_1, \ldots, \Box\neg\neg B'_p \vdash \neg D'_1, \neg\neg\Box\neg D'_2, \ldots, \neg\neg\Box\neg D'_r.$$

For any formula B, the sequent $\neg\neg B \vdash B$ is also provable. Using this result, the above sequent, *weaken*, and *cut*, the following is provable:

$$\Box\neg C', \Box A'_1, \ldots, \Box A'_m, \Box\neg\neg B'_1, \ldots, \Box\neg\neg B'_p \vdash \neg D'_1, \Box\neg D'_2, \ldots, \Box\neg D'_r.$$

Similarly, by applying the induction hypothesis and the above reasoning to the other $r-1$ hypotheses, we can show that the following sequents hold:

$$\begin{array}{c}\Box\neg C', \Box A'_1, \ldots, \Box A'_m, \Box\neg\neg B'_1, \ldots, \Box\neg\neg B'_p \vdash \Box\neg D'_1, \neg D'_2, \Box\neg D'_3, \ldots, \Box\neg D'_r \\ \vdots \\ \Box\neg C', \Box A'_1, \ldots, \Box A'_m, \Box\neg\neg B'_1, \ldots, \Box\neg\neg B'_p \vdash \Box\neg D'_1, \ldots, \Box\neg D'_{n-1}, \neg D'_r\end{array}$$

We can now apply $\Box$*right* with these r sequents as premises to obtain:

$$\Box\neg C', \Box A'_1, \ldots, \Box A'_m, \Box\neg\neg B'_1, \ldots, \Box\neg\neg B'_p \vdash \Box\neg D'_1, \ldots, \Box\neg D'_r.$$

By $\neg$*left* and $\neg$*right*, we obtain the desired result:

$$\neg\Box\neg D'_1, \ldots, \neg\Box\neg D'_r, \Box A'_1, \ldots, \Box A'_m, \neg\neg\Box\neg\neg B'_1, \ldots, \neg\neg\Box\neg\neg B'_p \vdash \neg\Box\neg C'.$$

Case: $\Diamond$*right*. C_1 is $\Diamond A$ and $n = 1$. By the induction hypothesis $\Gamma' \vdash A'$ is provable in $\mathcal{S}'$. We build the following proof to obtain the desired result.

$$\dfrac{\dfrac{\dfrac{\Gamma' \vdash A'}{\neg A', \Gamma' \vdash}\,\neg\textit{left}}{\Box\neg A', \Gamma' \vdash}\,\Box\textit{left}}{\Gamma' \vdash \neg\Box\neg A'}\,\neg\textit{right}$$

We now consider the backward direction. We begin with an $\mathcal{S}'$-proof, and build the corresponding $\mathcal{S}$-proof. The proof is again by induction, in this case on the height of the $\mathcal{S}'$-proof. The first three cases below are fairly simple. As in the proof above several cases follow from the induction hypothesis followed by an application of the corresponding rule in $\mathcal{S}$. These include $\wedge$*left*, $\vee$*left*, and $\Box$*left*. The other cases all follow by slightly more complicated reasoning from sequents known to hold by the induction hypothesis, with additional assumptions of the form $A \vee \neg A$. These assumptions are then eliminated by applications of *cut* with an instance of *excl-mid* as the left premise. We show four such cases: $\neg$*left*, $\neg$*right*, *cut*, and $\Box$*right*. The $\neg$*left* and $\neg$*right* rules are the only cases that have two subcases.

Case: *initial*. Here, $n = 1$ and we have the one-node $\mathcal{S}'$-proof $C'_1 \vdash C'_1$. Clearly, $C_1 \vdash C_1$ is provable by *initial* in $\mathcal{S}$.

Case: *weaken*. Γ has the form Γ_1, Γ_2, and the premise of this application of *weaken* is $\Gamma_1' \vdash C_1', \ldots, C_i'$ where $0 \leq i \leq n$. By the induction hypothesis, $\Gamma_1 \vdash C_1 \vee \cdots \vee C_i$ is provable in $\mathcal{S}$. By *weaken* and repeated applications of $\vee right_2$, $\Gamma_1, \Gamma_2 \vdash C_1 \vee \cdots \vee C_n$ is provable in $\mathcal{S}$.

Case: $\vee$ *right*. C_1 has the form $A \vee B$. By the induction hypothesis, $\Gamma \vdash A \vee B \vee C_2 \vee \ldots \vee C_n$ is provable in $\mathcal{S}$ which is what we want to show.

Case: $\neg$*left*. Γ has either the form $\neg A, \Gamma_0$ or $\Diamond A, \Gamma_0$.

In the first case, by the induction hypothesis, $\Gamma_0 \vdash A \vee C_1 \vee \cdots \vee C_n$ is provable in $\mathcal{S}$. For any formulas A, C and set of formulas Γ, if $\Gamma \vdash A \vee C$ is provable in $\mathcal{S}$, it is straightforward to construct a proof of $(A \vee C) \vee \neg(A \vee C), \neg A, \Gamma \vdash C$ without using applications of *cut* (other than those already in the proof of $\Gamma \vdash A \vee C$). Then by *cut*, $\neg A, \Gamma \vdash C$ is provable. Taking C to be $C_1 \vee \cdots \vee C_n$ and Γ to be Γ_0, we get a proof of $\neg A, \Gamma_0 \vdash C_1 \vee \cdots \vee C_n$ from $\Gamma_0 \vdash A \vee C_1 \vee \cdots \vee C_n$.

In the second case, by the induction hypothesis, $\Gamma_0 \vdash \Box\neg A \vee C_1 \vee \cdots \vee C_n$ is provable in $\mathcal{S}$. For any formulas A, C and set of formulas Γ, if $\Gamma \vdash \Box\neg A \vee C$ is provable in $\mathcal{S}$, it is straightforward to construct a proof of $(\Box\neg A \vee C) \vee \neg(\Box\neg A \vee C), \Diamond A, \Gamma \vdash C$ without using applications of *cut* (other than those already in the proof of $\Gamma \vdash \Box\neg A \vee C$). Then by *cut*, $\Diamond A, \Gamma \vdash C$ is provable. Taking C to be $C_1 \vee \cdots \vee C_n$ and Γ to be Γ_0, we get a proof of $\Diamond A, \Gamma_0 \vdash C_1 \vee \cdots \vee C_n$ from $\Gamma_0 \vdash \Box\neg A \vee C_1 \vee \cdots \vee C_n$.

Case: $\neg$*right*. C_1 has the form $\neg A$ or $\Diamond A$.

In the first case, by the induction hypothesis, $A, \Gamma \vdash C_2 \vee \cdots \vee C_n$ is provable in $\mathcal{S}$. For any formulas A, C and set of formulas Γ, if $A, \Gamma \vdash C$ is provable in $\mathcal{S}$, it is straightforward to construct a proof of $A \vee \neg A, \Gamma \vdash \neg A \vee C$ without using applications of *cut* (other than those already in the proof of $A, \Gamma \vdash C$). Then by *cut*, $\Gamma \vdash \neg A \vee C$ is provable. Taking C to be $C_2 \vee \cdots \vee C_n$, we get a proof of $\Gamma \vdash \neg A \vee C_2 \vee \cdots \vee C_n$ from $A, \Gamma \vdash C_2 \vee \cdots \vee C_n$.

In the second case, by the induction hypothesis, $\Box\neg A, \Gamma \vdash C_2 \vee \cdots \vee C_n$ is provable in $\mathcal{S}$. For any formulas A, C and set of formulas Γ, if $\Box\neg A, \Gamma \vdash C$ is provable in $\mathcal{S}$, it is straightforward to construct a proof of $\Diamond\neg\neg A \vee \neg\Diamond\neg\neg A, \Box\neg A \vee \neg\Box\neg A, \Gamma \vdash \Diamond A \vee C$ without using applications of *cut* (other than those already in the proof of $\Box\neg A, \Gamma \vdash C$). Then by *cut*, $\Gamma \vdash \Diamond A \vee C$ is provable. Taking C to be $C_2 \vee \cdots \vee C_n$, we get a proof of $\Gamma \vdash \Diamond A \vee C_2 \vee \cdots \vee C_n$ from $\Box\neg A, \Gamma \vdash C_2 \vee \cdots \vee C_n$.

Case: *cut*. Let A be the cut formula. By the induction hypothesis, $\Gamma \vdash A \vee C_1 \vee \cdots \vee C_n$ and $A, \Gamma \vdash C_1 \vee \cdots \vee C_n$ are provable in $\mathcal{S}$. For any formulas A, C and set of formulas Γ, if $\Gamma \vdash A \vee C$ and $A, \Gamma \vdash C$ are provable in $\mathcal{S}$, it is straightforward to construct a proof of $A \vee \neg A, \Gamma \vdash C$ without using applications of *cut* (other than those already in the proofs of $\Gamma \vdash A \vee C$ and $A, \Gamma \vdash C$). Then by *cut*, $\Gamma \vdash C$ is

provable. Taking C to be $C_1 \vee \cdots \vee C_n$, we get a proof of $\Gamma \vdash C_1 \vee \cdots \vee C_n$ from $\Gamma \vdash A \vee C_1 \vee \cdots \vee C_n$ and $A, \Gamma \vdash C_1 \vee \cdots \vee C_n$.

Case: $\Box$*right*. For $i = 1, \ldots, n$, C_i has the form $\Box A_i$. By the induction hypothesis, the following sequents are provable in $\mathcal{S}$:

$$\begin{array}{ll} \Box\Gamma \vdash A_1 \vee \Box A_2 \vee \cdots \vee \Box A_n & (a_1) \\ \vdots & \\ \Box\Gamma \vdash \Box A_1 \vee \cdots \vee \Box A_{n-1} \vee A_n & (a_n) \end{array}$$

We want to show that $\Box\Gamma \vdash \Box A_1 \vee \cdots \vee \Box A_n$ is provable in $\mathcal{S}$. For any formula A, we can build the following proof in $\mathcal{S}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{\Box A \vdash A}\,\Box left}{\neg A, \Box A \vdash \bot}\,\neg left}{\Diamond\neg A, \Box A \vdash \bot}\,\Diamond left}$$

By applying *weaken* followed by $n-2$ applications of $\vee$*left* to $n-1$ copies of the above proof with $A_2, \ldots, A_n$ as A, we get an $\mathcal{S}$-proof of the sequent on the top right below, which we then build on to get a proof of sequent (b).

$$\dfrac{\dfrac{A_1 \vdash A_1}{\neg A_1, A_1 \vdash \bot}\,\neg left \qquad \Diamond\neg A_2, \ldots, \Diamond\neg A_n, \Box A_2 \vee \cdots \vee \Box A_n, \Box\Gamma \vdash \bot}{\neg A_1, \Diamond\neg A_2, \ldots, \Diamond\neg A_n, A_1 \vee \Box A_2 \vee \cdots \vee \Box A_n, \Box\Gamma \vdash \bot \quad (b)}\,\vee left$$

In this proof and what follows, instances of *weaken* are left implicit. In the proof above for example, both premises of $\vee$*left* must be followed by *weaken* before $\vee$*left* can be applied. By $\neg$*left* from sequent (a_1), $\neg(A_1 \vee \Box A_2 \vee \cdots \vee \Box A_n), \Box\Gamma \vdash \bot$ holds. Then by $\vee$*left* from this sequent and (b), the following holds:

$$\neg A_1, \Diamond\neg A_2, \ldots, \Diamond\neg A_n, (A_1 \vee \Box A_2 \vee \cdots \vee \Box A_n) \vee \neg(A_1 \vee \Box A_2 \vee \cdots \vee \Box A_n), \Box\Gamma \vdash \bot\,.$$

Then by *excl-mid* and *cut*, we have $\neg A_1, \Diamond\neg A_2, \ldots, \Diamond\neg A_n, \Box\Gamma \vdash \bot$ (c_1). By similar reasoning from sequents (a_2) to (a_n), the following sequents are all provable in $\mathcal{S}$.

$$\begin{array}{ll} \neg A_1, \Diamond\neg A_2, \ldots, \Diamond\neg A_n, \Box\Gamma \vdash \bot & (c_1) \\ \vdots & \\ \Diamond\neg A_1, \ldots, \Diamond\neg A_{n-1}, \neg A_n, \Box\Gamma \vdash \bot & (c_n) \end{array}$$

We can now apply $\Diamond$*left* to get $\Diamond\neg A_1, \ldots, \Diamond\neg A_n, \Box\Gamma \vdash \bot$, followed by $\bot$ *right* to get $\Diamond\neg A_1, \ldots, \Diamond\neg A_n, \Box\Gamma \vdash \Box A_1 \vee \cdots \vee \Box A_n$ (d).

For any formula A, we can build the $\mathcal{S}$-proof below. In addition to leaving *weaken* implicit, we omit the left premise of the *cut* rule. It is always the consequence of *excl-mid*.

$$\dfrac{\dfrac{A \vdash A \qquad \dfrac{\dfrac{\dfrac{\dfrac{\neg A \vdash \neg A}{\neg A \vdash \Diamond\neg A}\Diamond right}{\neg\Diamond\neg A, \neg A \vdash \bot}\neg left}{\neg\Diamond\neg A, \neg A \vdash A}\bot\ right}{\neg\Diamond\neg A, A \vee \neg A \vdash A}\vee left}{\neg\Diamond\neg A \vdash A}\ cut}{\neg\Diamond\neg A \vdash \Box A \quad (e)}\Box right$$

We now build the following $\mathcal{S}$-proof. The sequent on the top right is obtained by taking A to be A_1 in the proof of (e).

$$\dfrac{\dfrac{\Diamond\neg A_1, \ldots, \Diamond\neg A_n, \Box\Gamma \vdash \Box A_1 \vee \cdots \vee \Box A_n \quad (d) \qquad \dfrac{\neg\Diamond\neg A_1 \vdash \Box A_1}{\neg\Diamond\neg A_1 \vdash \Box A_1 \vee \cdots \vee \Box A_n}\vee right}{\Diamond\neg A_1 \vee \neg\Diamond\neg A_1, \Diamond\neg A_2, \ldots, \Diamond\neg A_n, \Box\Gamma \vdash \Box A_1 \vee \cdots \vee \Box A_n}\vee left}{\Diamond\neg A_2, \ldots, \Diamond\neg A_n, \Box\Gamma \vdash \Box A_1 \vee \cdots \vee \Box A_n}\ cut$$

By similar reasoning, taking A to be A_2 through A_n in the proof of (e), we get an $\mathcal{S}$-proof of $\Box\Gamma \vdash \Box A_1 \vee \cdots \vee \Box A_n$.

THEOREM 2.2. *Given a set of formulas Γ and a formula C, let Γ' and C' be Γ and C, respectively, with all occurrences of $\Diamond$ replaced by $\neg\Box\neg$. If the sequent $\Gamma' \vdash C'$ (or $\Gamma' \vdash \emptyset$ when C is $\bot$) is provable in $\mathcal{S}'$ without cut, then $\Gamma \vdash C$ has a proof in $\mathcal{S}$ such that in all occurrences of the cut rule, the left premise is a direct consequence of the excl-mid rule.*

PROOF. The proof proceeds exactly as the proof of the converse of Theorem 1 above without the case for the *cut* rule. In all other cases in this proof, no applications of *cut* other than those whose left premise is the consequence of *excl-mid* are introduced in the $\mathcal{S}$-proof.

THEOREM 3.2. *Let C be a formula and Ψ be a set of $\mathcal{N}$-proofs. If Σ is an $\mathcal{M}$-proof of $\Psi \vdash C$, then $\hat{\mathcal{N}}(\Sigma)$ is an $\mathcal{N}$-proof of C from assumps(Ψ).*

PROOF. The proof is a fairly simple induction on the height of Σ. All of the cases for the *right* rules follow directly from the induction hypothesis and the corresponding *intro* rule of $\mathcal{N}$. We show the $\supset$ *right* rule to illustrate. We present the remaining cases except for $\neg$*left* since it is similar to $\supset$ *left* and $\Box$*left* since it is similar to $\wedge$*left*.

Case: *initial.* We have $\overset{\Pi}{C} \in \Psi$. Π is a proof of C from $assumps(\Pi)$. Since $assumps(\Pi) \subseteq assumps(\Psi)$, we have our result.

Case: *excl-mid.* C has the form $B \vee \neg B$ which is provable in $\mathcal{N}$ from any set of assumptions.

Case: $\supset$ *right.* C has the form $A \supset B$. Let Σ' be the $\mathcal{M}$-proof of the premise. By the induction hypothesis $\hat{\mathcal{N}}(\Sigma')$ is an $\mathcal{N}$-proof of B from $\{A\} \cup assumps(\Psi)$. By $\supset$ *intro*, we obtain a proof of $A \supset B$ from $assumps(\Psi)$.

Case: $\wedge$*left.* Ψ has the form $\overset{\Pi}{A \wedge B}, \Psi_0$. Let Σ' be the proof of the premise. Let Ψ' be the set of $\mathcal{N}$-proofs on the left of the sequent at the root of Σ'. Then Ψ' is $\frac{\overset{\Pi}{A \wedge B}}{A}, \frac{\overset{\Pi}{A \wedge B}}{B}, \Psi_0$. By the induction hypothesis, $\hat{\mathcal{N}}(\Sigma')$ is an $\mathcal{N}$-proof of C from $assumps(\Psi')$. Clearly $assumps(\Psi')$ is the same set of formulas as $assumps(\Psi)$, and thus we have our result.

Case: $\vee$*left.* Ψ has the form $\overset{\Pi}{A \vee B}, \Psi_0$. Let Σ_1 and Σ_2 be the proofs of the premises. Since $assumps(\Pi) \subseteq assumps(\Psi)$, we know that Π is a proof of $A \vee B$ from $assumps(\Psi)$. By the induction hypothesis $\hat{\mathcal{N}}(\Sigma_1)$ is an $\mathcal{N}$-proof of C from $\{A\} \cup assumps(\Psi_0)$ and $\hat{\mathcal{N}}(\Sigma_2)$ is an $\mathcal{N}$-proof of C from $\{B\} \cup assumps(\Psi_0)$. Since $\Psi_0 \subseteq \Psi$, $\hat{\mathcal{N}}(\Sigma_1)$ is a proof of C from $\{A\} \cup assumps(\Psi)$ and $\hat{\mathcal{N}}(\Sigma_2)$ is a proof of C from $\{B\} \cup assumps(\Psi)$. By an application of $\vee$*left*, we obtain a proof of C from $assumps(\Psi)$.

Case: $\supset$ *left.* Ψ has the form $\overset{\Pi}{A \supset B}, \Psi_0$. Let Σ_1 be the proof of the left premise. Since $assumps(\Pi) \subseteq assumps(\Psi)$, we know that Π is a proof of $A \supset B$ from $assumps(\Psi)$. By the induction hypothesis, $\hat{\mathcal{N}}(\Sigma_1)$ is an $\mathcal{N}$-proof of A from $assumps(\Psi_0)$. Since $\Psi_0 \subseteq \Psi$, $\hat{\mathcal{N}}(\Sigma_1)$ is a proof of A from $assumps(\Psi)$. By an application of $\supset$ *elim*, we obtain a proof of C from $assumps(\Psi)$.

Case: *cut.* Let $A \vee \neg A$ be the formula on the right of the sequent in the left premise and let Σ_2 be the proof of the right premise. By the induction hypothesis, $\hat{\mathcal{N}}(\Sigma_2)$ is an $\mathcal{N}$-proof of C from $assumps(A \vee \neg A) \cup assumps(\Psi)$. By definition, $assumps(A \vee \neg A) = \emptyset$. Thus, $\hat{\mathcal{N}}(\Sigma_2)$ is an $\mathcal{N}$-proof of C from $assumps(\Psi)$.

Case: *weaken.* Ψ has the form Ψ_1, Ψ_2. Let Σ' be the proof of the premise. By the induction hypothesis, $\hat{\mathcal{N}}(\Sigma')$ is an $\mathcal{N}$-proof of C from $assumps(\Psi_1)$. Since $\Psi_1 \subseteq \Psi$, $\hat{\mathcal{N}}(\Sigma')$ is a proof of C from $assumps(\Psi)$.

Case: $\Diamond$*left.* Ψ has the form $\overset{\Pi_1}{\Diamond A_1}, \ldots, \overset{\Pi_n}{\Diamond A_n}, \overset{\Pi'_1}{\Box B_1}, \ldots, \overset{\Pi'_m}{\Box B_m}$. Let $\Sigma_1, \ldots, \Sigma_n$ be the proofs of the premises. The sets of formulas

$$assumps(\Pi_1), \ldots, assumps(\Pi_n), assumps(\Pi'_1), \ldots, assumps(\Pi'_m)$$

are all contained in $assumps(\Psi)$. Thus we know that for $i = 1, \ldots, n$, Π_i is an $\mathcal{N}$-proof of $\Diamond A_i$ from $assumps(\Psi)$ and for $i = 1, \ldots, m$, Π'_i is an $\mathcal{N}$-proof of $\Box B_i$ from $assumps(\Psi)$. By the induction hypothesis, we know the following:

$$\begin{array}{c} \hat{\mathcal{N}}(\Sigma_1) \text{ is an } \mathcal{N}\text{-proof of } C \text{ from } A_1, \Diamond A_2, \ldots, \Diamond A_n, \Box B_1, \ldots, \Box B_m \\ \vdots \\ \hat{\mathcal{N}}(\Sigma_n) \text{ is an } \mathcal{N}\text{-proof of } C \text{ from } \Diamond A_1, \ldots, \Diamond A_{n-1}, A_n, \Box B_1, \ldots, \Box B_m \end{array}$$

Thus by an application of $\Diamond elim$, we obtain a proof of C from $assumps(\Psi)$.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399