On interactive proof-search for constructive modal necessity

Favio E. Miranda-Perea, Lourdes del Carmen González Huesca, and P. Selene Linares-Arévalo

Departamento de Matemáticas, Facultad de Ciencias,
Universidad Nacional Autónoma de México
{favio,luglzhuesca,selene_linares}@ciencias.unam.mx

Abstract. We present a dual sequent calculus for the necessity fragment of the constructive modal logic S4 and show its adequacy for proof-search in the style of modern interactive theorem provers. The main feature of dual systems is the use of two contexts to capture the notions of true and valid formulas, without using any formal semantics. This attribute allows us to give simple rules for the $\Box$ operator that, most of the time, grant the substitution of a strict modal reasoning by a pure propositional one, thus simplifying the proof-search process. Moreover, we introduce a formal notion of backward proof corresponding to a bottom-up construction of a derivation tree by means of a left-to-right depth-first proof-search.

Keywords: Constructive Modal logic · Proof-search · necessity · sequent calculus

1 Introduction

Modal logics play an important role in several areas of Computer Science and Philosophy. For instance, various type systems for concurrent and distributed computations [12,15] employ modalities and a modal lambda calculus has been proposed to model information flow in computer networks[4]. In Artificial Intelligence (AI) they are useful to model knowledge structures among agents, ontologies or the behaviour of computer systems. These and other applications oblige to design effective and simple to use procedures for proof search in modal logics. However, the existing processes are quite sophisticated, for instance [17,1], and machine-oriented in the sense that their inference rules deviate from the natural human reasoning, for they are or intend to be fully automated, like the bidirectional approach of [10]. This makes the modelling of actual arguments involving modal reasoning difficult. Such arguments arise for instance in AI, while verifying an agent-based computer system; or in Philosophy, in the representation of concrete cases in modal argumentation theory, where the $\Box$ operator usually represents the argument attack relation [3,8].

In this paper we walk in the opposite direction by proposing a sequent calculus adequate for proof-search but in an interactive style, as is understood and

implemented in modern proof-assistants like Coq. This means that the proof-search is driven by the human agent: the process starts with the desired sequent S; if it is suitable as a conclusion of some rule, then the process continues with the premises of this rule. If at the end all branches of this search end in an axiom or initial rule, the procedure is succesful. The purpose of this work is to give a precise formal definition of this process and to prove its equivalence with the usual forward proof construction in the case of constructive modal logic S4 for necessity.

2 A Dual Sequent Calculus for Necessity

The formal system $\mathcal{GS}4$ hereby considered is a sequent calculus whose foundations come from the analysis of [13]. Propositions are analysed judgmentally without any semantic label (worlds). In the specific case of modal connectives by means of judgments over propositions. The notion of so-called hypothetical judgments is extended to categorical judgment where a conclusion does not depend on hypotheses about the constructive truth of propositions. Hence, a distinction of two forms of primitive judgments is essential: 'A *true*' means that we know how to verify A under hypothetical judgments, whereas 'A *valid*' represents the fact that A is a proposition whose truth does not depend on any hypotheses, thus internalizing a categorical judgment as a proposition syntactically represented by the modal formula $\Box A$. The system is called dual for it handles sequents with two separate contexts of the form $\Delta|\Gamma \vdash A$. We chose to implement such contexts by means of two disjoint lists (instead of sets or multisets), Δ for valid and Γ for true hypotheses. This choice allows us to omit the labels *valid, true* in context formulae, contrary to [16]. A similar separation is present in several works: for instance the systems of [6] use global and local assumptions or the work of [2] which discusses modal logic encodings by using separate consequence relations (semantical and deductive) related to classical validity and truth. The idea behind the context separation in $\mathcal{GS}4$ is that formulae in Δ are modal (i.e. boxed formulae), whereas those in Γ are propositional. Nevertheless, like the intuitive semantic qualifiers, this idea does not represent a syntactic restriction, we can have arbitrary formulae in both contexts. Moreover, the succedent of a sequent only considers true propositions, there is no need to consider valid conclusions explicitly as they are represented by the formula $\Box A$. For the sake of self-containment let us briefly review the elements of our syntax. Modal formulae are generated by the following grammar:

$$A, B ::= p_n \mid A \to B \mid A \wedge B \mid A \vee B \mid \Box A$$

where p_n denotes an element taken from an infinite supply of propositional variables, indexed by a natural number. Let us also note that we do not consider neither negation nor the constant $\bot$. Therefore we will be dealing with minimal propositional logic [18] extended with the necessity operator. A particular modal logic is generated by adding suitable axioms. It is important to remark that

the logic here discussed is called constructive due to the approach of Martin-Löf but also to distinguish it from both, the classical modal logics where the $\Box$ operator obeys the axiom $\Box A \leftrightarrow \neg\Diamond\neg A$ and the intuitionistic modal logics which include the axiom $\neg\Diamond\bot$. In contrast, in the constructive modal logic S4 the above axioms are omitted and both modalities are primitive. Our work only considers the necessity modality $\Box$. Contexts are implemented by means of so-called *snoc* lists:

$$\Gamma, \Delta ::= \cdot \mid \Gamma, A$$

These are finite lists built from the empty list, denoted here by $\cdot$, and a constructor that generates a new list from a given one by adding a new element to its right-end. The constructor called *snoc* is denoted by a comma. Therefore Γ, A is the context obtained by adding A after the last element in Γ. This constructor is opposed to the traditional *cons* operation, predefined in functional languages, which adds elements at the left-end of the list [1]. The *snoc* lists provide a formalization akin to the usual practice in computer science logic where the last context formula is the one introduced or discharged to prove an implication. The concatenation operation is inductively defined in the obvious way and denoted by $\Gamma_1; \Gamma_2$. Also, whenever we write $\Gamma, A; \Gamma'$ it means $(\Gamma, A); \Gamma'$, that is, the formula A is associated to the left context. This should be clear, for the expression $\Gamma, (A; \Gamma')$ is ill-formed.

The system $\mathcal{GS}4$ is defined by the following inference rules:

– Starting rules

$$\frac{}{\Delta|\Gamma, A; \Gamma' \vdash A}\ (\textsc{Thyp}) \qquad \frac{}{\Delta, A; \Delta'|\Gamma \vdash A}\ (\textsc{Vhyp})$$

– Right rules

$$\frac{\Delta|\Gamma, A \vdash B}{\Delta|\Gamma \vdash A \to B}\ (\to \mathrm{R}) \qquad \frac{\Delta|\Gamma \vdash A \qquad \Delta|\Gamma \vdash B}{\Delta|\Gamma \vdash A \wedge B}\ (\wedge \mathrm{R})$$

$$\frac{\Delta|\Gamma \vdash A}{\Delta|\Gamma \vdash A \vee B}\ (\vee \mathrm{R}) \qquad \frac{\Delta|\Gamma \vdash B}{\Delta|\Gamma \vdash A \vee B}\ (\vee \mathrm{R})$$

$$\frac{\Delta|\cdot \vdash A}{\Delta|\Gamma \vdash \Box A}\ (\Box \mathrm{R})$$

– Left rules for truth contexts.

$$\frac{\Delta|\Gamma, A; \Gamma' \vdash C \qquad \Delta|\Gamma, B; \Gamma' \vdash C}{\Delta|\Gamma, A \vee B; \Gamma' \vdash C}\ (\vee \mathrm{L}) \qquad \frac{\Delta|\Gamma, A, B; \Gamma' \vdash C}{\Delta|\Gamma, A \wedge B; \Gamma' \vdash C}\ (\wedge \mathrm{L})$$

$$\frac{\Delta|\Gamma, A \to B; \Gamma' \vdash A}{\Delta|\Gamma, A \to B; \Gamma' \vdash B}\ (\to \mathrm{L}) \qquad \frac{\Delta, A|\Gamma; \Gamma' \vdash B}{\Delta|\Gamma, \Box A; \Gamma' \vdash B}\ (\Box \mathrm{L})$$

[1] The name *cons* is for constructor and note that *snoc* is the backward reading of *cons*.

– Left rules for valid contexts.

$$\frac{\Delta, A; \Delta' | \Gamma \vdash C \qquad \Delta, B; \Delta' | \Gamma \vdash C}{\Delta, A \vee B; \Delta' | \Gamma \vdash C} \ (\vee \text{LV}) \qquad \frac{\Delta, A, B; \Delta' | \Gamma \vdash C}{\Delta, A \wedge B; \Delta' | \Gamma \vdash C} \ (\wedge \text{LV})$$

$$\frac{\Delta, A \to B; \Delta' | \Gamma \vdash A}{\Delta, A \to B; \Delta' | \Gamma \vdash B} \ (\to \text{LV}) \qquad \frac{\Delta, A; \Delta' | \Gamma \vdash B}{\Delta, \Box A; \Delta' | \Gamma \vdash B} \ (\Box \text{LV})$$

– Cut rules

$$\frac{\Delta | \Gamma \vdash A \qquad \Delta | \Gamma, A \vdash B}{\Delta | \Gamma \vdash B} \ (\textsc{Cut}) \qquad \frac{\Delta | \cdot \vdash A \qquad \Delta, A | \Gamma \vdash B}{\Delta | \Gamma \vdash B} \ (\textsc{CutV})$$

We have two starting rules allowing to use an hypothesis present in any of the two contexts; the right rules for the connectives are the usual ones; the left rules come in two versions, one for each context. For the case of conjunction and disjunction these rules are also usual. The left rules for implication are new[2], to the best of our knowledge and capture the direct use of an implication to prove its consequent. The right rule for $\Box$ corresponds to the necessitation rule. Observe that we can introduce a formula $\Box A$ only in the case when we derive A without resorting to any true hypothesis. This formulation is the key to validate the deduction theorem in Hilbert axiomatic systems for modal logic. See for instance [9]. The left rule $\Box L$ is discussed in our previous work [5] and represents a transference principle between contexts: we can move a valid hypothesis A to the truth context by modalizing it. Finally $\Box LV$ says that to use an assumption $\Box A$ which is valid, it suffices to use only A. These left rules for necessity allow to replace a modal reasoning about $\Box A$, for a propositional reasoning about A.

Let us show a couple of derivations in our system.

Example 1. The following is a proof of

$$\cdot|\cdot \vdash \Box(\Box A \to \Box B) \to \Box\big(\Box A \to \Box(\Box C \to \Box B)\big)$$

(1)	$\Box A \to \Box B\,,\ A\|\cdot \vdash A$	VHYP
(2)	$\Box A \to \Box B\,,\ A\|\Box C \vdash \Box A$	$\Box$ R (1)
(3)	$\Box A \to \Box B\,,\ A\|\Box C \vdash \Box B$	$\to$ L (2)
(4)	$\Box A \to \Box B\,,\ A\|\cdot \vdash \Box C \to \Box B$	$\to$ R (3)
(5)	$\Box A \to \Box B\,,\ A\|\cdot \vdash \Box(\Box C \to \Box B)$	$\Box$ R (4)
(6)	$\Box A \to \Box B\|\Box A \vdash \Box(\Box C \to \Box B)$	$\Box$ L (5)
(7)	$\Box A \to \Box B\|\cdot \vdash \Box A \to \Box(\Box C \to \Box B)$	$\to$ R (6)
(8)	$\Box A \to \Box B\|\cdot \vdash \Box\big(\Box A \to \Box(\Box C \to \Box B)\big)$	$\Box$ R (7)
(9)	$\cdot\|\Box(\Box A \to \Box B) \vdash \Box\big(\Box A \to \Box(\Box C \to \Box B)\big)$	$\Box$ L (8)
(10)	$\cdot\|\cdot \vdash \Box(\Box A \to \Box B) \to \Box\big(\Box A \to \Box(\Box C \to \Box B)\big)$	$\to$ R (9)

[2] Except for a mention in our previous work [14]

Example 2. The following is a proof of

$$\Box\big((\Box B \to \Box A) \wedge (\Box C \to \Box A)\big)|\cdot \vdash \Box\big(\Box B \vee \Box C \to \Box A\big)$$

(**1**) $\Box B \to \Box A\,,\, \Box C \to \Box A|\Box B \vdash \Box B$	HYP
(**2**) $\Box B \to \Box A\,,\, \Box C \to \Box A|\Box B \vdash \Box A$	$\to$ LV (1)
(**3**) $\Box B \to \Box A\,,\, \Box C \to \Box A|\Box C \vdash \Box C$	HYP
(**4**) $\Box B \to \Box A\,,\, \Box C \to \Box A|\Box C \vdash \Box A$	$\to$ LV (3)
(**5**) $\Box B \to \Box A\,,\, \Box C \to \Box A|\Box B \vee \Box C \vdash \Box A$	$\vee$ L (2, 4)
(**6**) $\Box B \to \Box A\,,\, \Box C \to \Box A|\cdot \vdash \Box B \vee \Box C \to \Box A$	$\to$ R (5)
(**7**) $\Box B \to \Box A\,,\, \Box C \to \Box A|\cdot \vdash \Box(\Box B \vee \Box C \to \Box A)$	$\Box$ R (6)
(**8**) $(\Box B \to \Box A) \wedge (\Box C \to \Box A)|\cdot \vdash \Box(\Box B \vee \Box C \to \Box A)$	$\wedge$ L (7)
(**9**) $\Box((\Box B \to \Box A) \wedge (\Box C \to \Box A))|\cdot \vdash \Box(\Box B \vee \Box C \to \Box A)$	$\Box$ LV (8)

The above proofs can be read according to the usual definition of forward derivation, though actually they were gained from a bottom-up proof construction process, which corresponds to the below notion of derivation given originally by Kanger [11].

Definition 1 (Derivation à la Kanger). *A proof or derivation of $\Delta|\Gamma \vdash A$ is a finite sequence of sequents $\Pi = \langle \mathcal{J}_1, \ldots, \mathcal{J}_k \rangle$ such that $\mathcal{J}_1$ is $\Delta|\Gamma \vdash A$ and for every $1 \leq i \leq k$ one of the following conditions hold:*

- *$\mathcal{J}_i$ is an instance of the (*HYP*) or (*VHYP*) rules.*
- *For every $1 \leq i \leq k$, $\mathcal{J}_i$ is the conclusion of an instance of some inference rule with premise $\mathcal{J}_j$ with $j > i$, or premises $\mathcal{J}_j, \mathcal{J}_l$ with $j, l > i$.*

Thus, a derivation starts with the sequent sought after, whereas the usual notion of formal proof ends with it. Also, for any given sequent, the premisses that allow to conclude it appear later in the sequence. This gives an idea of backward proof. Regrettably, even when this intuitive notion permeates Kanger's work, the concept is not engaged, for the proofs there are written forwards. Moreover, like the examples above show, the mere sequence does not let us keep a trace of the proof-search process. Our purpose here is to give a formal notion of backward proof for modal logic that corresponds to the kind of techniques implemented in proof assistants, thus resolving the just mentioned issues. But first let us observe that $\mathcal{GS}4$ indeed captures the constructive modal logic $\mathsf{S4}$ for necessity.

Theorem 1. *The sequent calculus $\mathcal{GS}4$ exactly captures the constructive modal logic* $\mathsf{S4}$

Proof (Sketch). It is easy to show that the characteristic axioms of $\mathsf{S4}$, namely $\mathbb{K}, \mathbb{T}$ and **4** are derivable in $\mathcal{GS}4$. Moreover, by structural induction on the respective derivability relations, we can show that $\mathcal{GS}4$ is equivalent to the dual natural deduction system $\mathcal{N}_{\mathsf{S4}}$ of [5] (we omit the proof details due to lack of space), where we also give a detailed and formally verified proof of the equivalence of $\mathcal{N}_{\mathsf{S4}}$ with an axiomatic system for $\mathsf{S4}$.

3 Backward proofs in $\mathcal{GS}4$

To being able to formalize our notion of backward proof, let us first introduce a useful device to sequents, namely labelled hypotheses [7], which are pairs of the form $\mathcal{H} : A$ where $\mathcal{H}$ is a label or shortcut to refer to A. The set of labels is taken to be disjoint with the set of names in the current signature. A labelled context Γ is a set of labelled hypotheses $\Gamma = \{\mathcal{H}_1 : A_1, \ldots, \mathcal{H}_n : A_n\}$ where $\mathcal{H}_i \neq \mathcal{H}_j$, if $i \neq j$. Moreover, in a context of the form $\Gamma, \mathcal{H} : A; \Gamma'$ we assume that A is not in $\Gamma; \Gamma'$ and that $\mathcal{H}$ is a new label not used in $\Gamma; \Gamma'$. The use of labels does not contribute much to the usual system of forward derivations, but as we will see soon it is very useful to the backward approach. In the following we use labelled contexts where they account for simplicity. Otherwise, we use conventional contexts.

The process of backward proof construction of a given sequent S consists of searching for an inference rule $\mathcal{R}$ whose conclusion matches S and then to continue the proof-search with the premisses S or S_1, S_2 of $\mathcal{R}$. This backward reading of the inference rules is called a tactic. A backward proof will be a particular sequence of tactics. Let us formalise these concepts.

Definition 2. *A goal $\mathcal{G}$ is any sequent $\Delta|\Gamma \vdash A$. The set of finite sequences of goals GSeq is recursively defined as follows:*

$$\mathcal{S} ::= [\cdot] \mid (\mathcal{G} :: \mathcal{S})$$

where $[\cdot]$ denotes the empty goal sequence. Moreover, if $\mathcal{S}_1, \mathcal{S}_2 \in GSeq$ then by $\mathcal{S}_1; \mathcal{S}_2$ we mean the concatenation[3] of $\mathcal{S}_1$ with $\mathcal{S}_2$.

We define now a transition system of tactics corresponding to backward proof-search.

Definition 3. *The transition system of tactics for $\mathcal{GS}4$ is defined as follows:*

- *The non-empty set of states is the set of goal sequences GSeq.*
- *An initial state is a singleton sequence[4].*
- *$[\cdot]$ is the unique terminal state.*
- *The transition relation $\rhd \subseteq GSeq \times GSeq$ is inductively defined by the below axioms and inference rule, where a transition $\mathcal{S}_1 \rhd \mathcal{S}_2$ can be read as "to prove the sequents in $\mathcal{S}_1$ it suffices to prove the sequents in $\mathcal{S}_2$".*

  Conclusion analysis (right sequent rules):
  - `intro` $\mathcal{H}$: $\Delta|\Gamma \vdash A \to B \rhd \Delta|\Gamma, \mathcal{H} : A \vdash B$
  - `split`: $\Delta|\Gamma \vdash A \wedge B \rhd \Delta|\Gamma \vdash A; \Delta|\Gamma \vdash B$
  - `left`: $\Delta|\Gamma \vdash A \vee B \rhd \Delta|\Gamma \vdash A$
  - `right`: $\Delta|\Gamma \vdash A \vee B \rhd \Delta|\Gamma \vdash B$

[3] We use the ; for concatenation in both cases: contexts and goal sequences.

[4] For clarity, a singleton sequence is identified with its unique element. That is, we write $\Delta|\Gamma \vdash A$ instead of $(\Delta|\Gamma \vdash A :: [\cdot])$

- `necessitation`: $\Delta|\Gamma \vdash \Box A \ \rhd\ \Delta|\cdot \vdash A$

Premise analysis (left sequent rules):

- `apply` $\mathcal{H}$: $\Delta|\Gamma, \mathcal{H} : A \to B; \Gamma' \vdash B \ \rhd\ \Delta|\Gamma, \mathcal{H} : A \to B; \Gamma' \vdash A$
- `apply` $\mathcal{H}$: $\Delta, \mathcal{H} : A \to B; \Delta'|\Gamma \vdash B \ \rhd\ \Delta, \mathcal{H} : A \to B; \Delta'|\Gamma \vdash A$
- `destruct` $\mathcal{H}$: $\Delta|\Gamma, \mathcal{H} : A \wedge B; \Gamma' \vdash C \ \rhd\ \Delta|\Gamma, \mathcal{H}_1 : A, \mathcal{H}_2 : B; \Gamma' \vdash C$
- `destruct` $\mathcal{H}$: $\Delta, \mathcal{H} : A \wedge B; \Delta'|\Gamma \vdash C \ \rhd\ \Delta, \mathcal{H}_1 : A, \mathcal{H}_2 : B; \Delta'|\Gamma \vdash C$
- `destruct` $\mathcal{H}$: $\Delta|\Gamma, \mathcal{H} : A \vee B; \Gamma' \vdash C \ \rhd\ \Delta|\Gamma, \mathcal{H}_1 : A; \Gamma' \vdash C$; $\Delta|\Gamma, \mathcal{H}_2 : B; \Gamma' \vdash C$
- `destruct` $\mathcal{H}$: $\Delta, \mathcal{H} : A \vee B; \Delta'|\Gamma \vdash C \ \rhd\ \Delta, \mathcal{H}_1 : A; \Delta'|\Gamma \vdash C$; $\Delta, \mathcal{H}_2 : B; \Delta'|\Gamma \vdash C$
- `destruct` $\mathcal{H}$: $\Delta|\Gamma, \mathcal{H} : \Box A; \Gamma' \vdash B \ \rhd\ \Delta, \mathcal{H}_1 : A|\Gamma; \Gamma' \vdash B$
- `destruct` $\mathcal{H}$: $\Delta, \mathcal{H} : \Box A; \Delta'|\Gamma \vdash B \ \rhd\ \Delta, \mathcal{H} : A; \Delta'|\Gamma \vdash B$

Lemma Assertion:

- `assert A`: $\Delta|\Gamma \vdash C \ \rhd\ \Delta|\Gamma \vdash A; \Delta|\Gamma, \mathcal{H} : A \vdash C$
- `cut A`: $\Delta|\Gamma \vdash C \ \rhd\ \Delta|\Gamma, \mathcal{H} : A \vdash C; \Delta|\Gamma \vdash A$
- `vassert A`: $\Delta|\Gamma \vdash C \ \rhd\ \Delta|\cdot \vdash A; \Delta, \mathcal{H} : A|\Gamma \vdash C$
- `vcut A`: $\Delta|\Gamma \vdash C \ \rhd\ \Delta, \mathcal{H} : A|\Gamma \vdash C \ ;\ \Delta|\cdot \vdash A$

Discarding tactics:

- `assumption`: $\Delta|\Gamma, A; \Gamma' \vdash A \ \rhd\ [\cdot]$.
- `vassumption`: $\Delta, A; \Delta'|\Gamma \vdash A \ \rhd\ [\cdot]$.

Sequencing:

$$\frac{\mathcal{S}_1 \rhd \mathcal{S}_2}{\mathcal{S}_1; \mathcal{S} \rhd \mathcal{S}_2; \mathcal{S}}(\texttt{seq})$$

A basic transition transforms a singleton goal sequence into a, perhaps empty, sequence of subgoals dictated by the backwards reading of an inference rule of $\mathcal{GS}4$. The (`seq`) rule determines the order in which goals are solved, namely from the first (most left) goal in the current sequence of pending goals. Also observe that each cut rule has two corresponding tactics, namely cut and assert. The difference being only operational: either we first prove the lemma and then use it or viceversa.

Finally we can give the promised definition of a backward proof.

Definition 4. *A backward proof of* $\Delta|\Gamma \vdash A$ *is a finite sequence of states* $\mathcal{S}_1, \ldots, \mathcal{S}_k$ *such that*

- $\mathcal{S}_1$ *is* $\Delta|\Gamma \vdash A$
- *For every* $1 \leq i < k$, $\mathcal{S}_i \rhd \mathcal{S}_{i+1}$
- $\mathcal{S}_k$ *is* $[\cdot]$

Therefore a backward proof of a sequent $\Delta|\Gamma \vdash A$ is a finite sequence of tactics that ends in the empty sequence of goals $[\cdot]$, meaning that the proof-search of the original goal $\Delta|\Gamma \vdash A$ has no pending subgoals left to prove. This transition sequence of tactics corresponds to the bottom-up construction of a derivation tree by means of a left-to-right depth-first proof search.

Let us present now some particular examples of backward proofs that intend to show the adequacy of our secuent calculus $\mathcal{GS}4$ for interactive proof-search.

Example 3. The following is a backward proof of

$$\cdot|\mathcal{H}_1 : \Box\big((\Box P \to \Box Q) \lor \Box R\big) \vdash \Box\Big(\Box(\Box P \lor \Box R) \to \Box(\Box Q \lor \Box R)\Big)$$

$\cdot|\mathcal{H}_1 : \Box\big((\Box P \to \Box Q) \lor \Box R\big) \vdash \Box\Big(\Box(\Box P \lor \Box R) \to \Box(\Box Q \lor \Box R)\Big)$
$\triangleright$ `destruct` $\mathcal{H}_1$
$\mathcal{H}_1 : (\Box P \to \Box Q) \lor \Box R|\cdot \vdash \Box\Big(\Box(\Box P \lor \Box R) \to \Box(\Box Q \lor \Box R)\Big)$
$\triangleright$ `necessitation`
$\mathcal{H}_1 : (\Box P \to \Box Q) \lor \Box R|\cdot \vdash \Box(\Box P \lor \Box R) \to \Box(\Box Q \lor \Box R)$
$\triangleright$ `intro` $\mathcal{H}_2$
$\mathcal{H}_1 : (\Box P \to \Box Q) \lor \Box R|\mathcal{H}_2 : \Box(\Box P \lor \Box R) \vdash \Box(\Box Q \lor \Box R)$
$\triangleright$ `destruct` $\mathcal{H}_2$
$\mathcal{H}_1 : (\Box P \to \Box Q) \lor \Box R\,,\, \mathcal{H}_2 : \Box P \lor \Box R|\cdot \vdash \Box(\Box Q \lor \Box R)$
$\triangleright$ `necessitation`
$\mathcal{H}_1 : (\Box P \to \Box Q) \lor \Box R\,,\, \mathcal{H}_2 : \Box P \lor \Box R|\cdot \vdash \Box Q \lor \Box R$
$\triangleright$ `destruct` $\mathcal{H}_2$
$\mathcal{H}_1 : (\Box P \to \Box Q) \lor \Box R\,,\, \mathcal{H}_2 : \Box P|\cdot \vdash \Box Q \lor \Box R$;
$\mathcal{H}_1 : (\Box P \to \Box Q) \lor \Box R\,,\, \mathcal{H}_2 : \Box R|\cdot \vdash \Box Q \lor \Box R$
$\triangleright$ `destruct` $\mathcal{H}_1$
$\mathcal{H}_1 : \Box P \to \Box Q\,,\, \mathcal{H}_2 : \Box P|\cdot \vdash \Box Q \lor \Box R$;
$\mathcal{H}_1 : \Box R\,,\, \mathcal{H}_2 : \Box P|\cdot \vdash \Box Q \lor \Box R$;
$\mathcal{H}_1 : (\Box P \to \Box Q) \lor \Box R\,,\, \mathcal{H}_2 : \Box R|\cdot \vdash \Box Q \lor \Box R$
$\triangleright$ `left`
$\mathcal{H}_1 : \Box P \to \Box Q\,,\, \mathcal{H}_2 : \Box P|\cdot \vdash \Box Q$;
$\mathcal{H}_1 : \Box R\,,\, \mathcal{H}_2 : \Box P|\cdot \vdash \Box Q \lor \Box R$;
$\mathcal{H}_1 : (\Box P \to \Box Q) \lor \Box R\,,\, \mathcal{H}_2 : \Box R|\cdot \vdash \Box Q \lor \Box R$
$\triangleright$ `apply` $\mathcal{H}_1$
$\mathcal{H}_1 : \Box P \to \Box Q\,,\, \mathcal{H}_2 : \Box P|\cdot \vdash \Box P$;
$\mathcal{H}_1 : \Box R\,,\, \mathcal{H}_2 : \Box P|\cdot \vdash \Box Q \lor \Box R$;
$\mathcal{H}_1 : (\Box P \to \Box Q) \lor \Box R\,,\, \mathcal{H}_2 : \Box R|\cdot \vdash \Box Q \lor \Box R$
$\triangleright$ `vassumption`
$\mathcal{H}_1 : \Box R\,,\, \mathcal{H}_2 : \Box P|\cdot \vdash \Box Q \lor \Box R$;
$\mathcal{H}_1 : (\Box P \to \Box Q) \lor \Box R\,,\, \mathcal{H}_2 : \Box R|\cdot \vdash \Box Q \lor \Box R$
$\triangleright$ `right`
$\mathcal{H}_1 : \Box R\,,\, \mathcal{H}_2 : \Box P|\cdot \vdash \Box R$;
$\mathcal{H}_1 : (\Box P \to \Box Q) \lor \Box R\,,\, \mathcal{H}_2 : \Box R|\cdot \vdash \Box Q \lor \Box R$
$\triangleright$ `vassumption`
$\mathcal{H}_1 : (\Box P \to \Box Q) \lor \Box R\,,\, \mathcal{H}_2 : \Box R|\cdot \vdash \Box Q \lor \Box R$
$\triangleright$ `right`
$\mathcal{H}_1 : (\Box P \to \Box Q) \lor \Box R\,,\, \mathcal{H}_2 : \Box R|\cdot \vdash \Box R$
$\triangleright$ `vassumption`
$[\cdot]$

Example 4. The following is a proof of

$$\Delta|\cdot \vdash \Box(\Box A \wedge \Box D) \to \Box E$$

where $\Delta =_{def} \mathcal{H}_1 : \Box A \to \Box(\Box B \wedge \Box C),\ \mathcal{H}_2 : \Box(\Box D \wedge \Box C) \to \Box(\Box E \wedge \Box F)$.

$\Delta\|\cdot \vdash \Box(\Box A \wedge \Box D) \to \Box E$	
$\triangleright$	`intro` $\mathcal{H}_3$
$\Delta\|\mathcal{H}_3 : \Box(\Box A \wedge \Box D) \vdash \Box E$	
$\triangleright$	`destruct` $\mathcal{H}_3$
$\Delta\ ,\ \mathcal{H}_3 : \Box A \wedge \Box D\|\cdot \vdash \Box E$	
$\triangleright$	`destruct` $\mathcal{H}_3$
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\|\cdot \vdash \Box E$	
$\triangleright$	`vassert` $\Box(\Box B \wedge \Box C)$
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\|\cdot \vdash \Box(\Box B \wedge \Box C)$;	
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box(\Box B \wedge \Box C)\|\cdot \vdash \Box E$	
$\triangleright$	`apply` $\mathcal{H}_1$
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\|\cdot \vdash \Box A$;	
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box(\Box B \wedge \Box C)\|\cdot \vdash \Box E$	
$\triangleright$	`vassumption`
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box(\Box B \wedge \Box C)\|\cdot \vdash \Box E$	
$\triangleright$	`destruct` $\mathcal{H}_5$
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box B \wedge \Box C\|\cdot \vdash \Box E$	
$\triangleright$	`destruct` $\mathcal{H}_5$
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box B\ ,\ \mathcal{H}_6 : \Box C\|\cdot \vdash \Box E$	
$\triangleright$	`vassert` $\Box(\Box E \wedge \Box F)$
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box B\ ,\ \mathcal{H}_6 : \Box C\|\cdot \vdash \Box(\Box E \wedge \Box F)$;	
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box B\ ,\ \mathcal{H}_6 : \Box C\ ,\ \mathcal{H}_7 : \Box(\Box E \wedge \Box F)\|\cdot \vdash \Box E$	
$\triangleright$	`apply` $\mathcal{H}_2$
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box B\ ,\ \mathcal{H}_6 : \Box C\|\cdot \vdash \Box(\Box D \wedge \Box C)$;	
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box B\ ,\ \mathcal{H}_6 : \Box C\ ,\ \mathcal{H}_7 : \Box(\Box E \wedge \Box F)\|\cdot \vdash \Box E$	
$\triangleright$	`necessitation`
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box B\ ,\ \mathcal{H}_6 : \Box C\|\cdot \vdash \Box D \wedge \Box C$;	
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box B\ ,\ \mathcal{H}_6 : \Box C\ ,\ \mathcal{H}_7 : \Box(\Box E \wedge \Box F)\|\cdot \vdash \Box E$	
$\triangleright$	`split`
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box B\ ,\ \mathcal{H}_6 : \Box C\|\cdot \vdash \Box D$;	
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box B\ ,\ \mathcal{H}_6 : \Box C\|\cdot \vdash \Box C$;	
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box B\ ,\ \mathcal{H}_6 : \Box C\ ,\ \mathcal{H}_7 : \Box(\Box E \wedge \Box F)\|\cdot \vdash \Box E$	
$\triangleright$	`vassumption`
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box B\ ,\ \mathcal{H}_6 : \Box C\|\cdot \vdash \Box C$;	
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box B\ ,\ \mathcal{H}_6 : \Box C\ ,\ \mathcal{H}_7 : \Box(\Box E \wedge \Box F)\|\cdot \vdash \Box E$	
$\triangleright$	`vassumption`
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box B\ ,\ \mathcal{H}_6 : \Box C\ ,\ \mathcal{H}_7 : \Box(\Box E \wedge \Box F)\|\cdot \vdash \Box E$	
$\triangleright$	`destruct` $\mathcal{H}_7$
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box B\ ,\ \mathcal{H}_6 : \Box C\ ,\ \mathcal{H}_7 : \Box E \wedge \Box F\|\cdot \vdash \Box E$	
$\triangleright$	`destruct` $\mathcal{H}_7$
$\Delta\ ,\ \mathcal{H}_3 : \Box A\ ,\ \mathcal{H}_4 : \Box D\ ,\ \mathcal{H}_5 : \Box B\ ,\ \mathcal{H}_6 : \Box C\ ,\ \mathcal{H}_7 : \Box E\ ,\ \mathcal{H}_8 : \Box F\|\cdot \vdash \Box E$	
$\triangleright$	`vassumption`
$[\cdot]$	

The above example shows the utility of the cut rule with a valid hypothesis, implemented by means of the `vassert` tactic.

We discuss next the equivalence of backward and forward proofs.

4 Equivalence

In this section we formally prove that our backward approach is equivalent to the usual notion of (forward) derivation. Let us start by noting that, according to definition 4, a backward proof is a chaining sequence of particular instances of the relation $\rhd$. Therefore a backward proof of the sequent $\Delta|\Gamma \vdash A$ corresponds to an instance of the transitive closure of the $\rhd$ relation. For the proof of the equivalence, it will be useful to work directly with this relation.

Definition 5. *The transitive closure of the relation $\rhd$, denoted $\rhd^+$, is inductively defined by the following rules:*

$$\frac{\mathcal{S} \rhd \mathcal{S}'}{\mathcal{S} \rhd^+ \mathcal{S}'}(tc1) \qquad \frac{\mathcal{S} \rhd \mathcal{S}' \quad \mathcal{S}' \rhd^+ \mathcal{S}''}{\mathcal{S} \rhd^+ \mathcal{S}''}(tc2)$$

Lemma 1. *The following rule is admissible:*

$$\frac{\mathcal{S}_1 \rhd^+ \mathcal{S}_2}{\mathcal{S}_1; \mathcal{S} \rhd^+ \mathcal{S}_2; \mathcal{S}}(seq^+)$$

Proof. Induction on $\mathcal{S}_1 \rhd^+ \mathcal{S}_2$. If $\mathcal{S}_1 \rhd \mathcal{S}_2$, then the rule (seq) yields $\mathcal{S}_1; \mathcal{S} \rhd \mathcal{S}_2; \mathcal{S}$ and by rule (tc1) we get $\mathcal{S}_1; \mathcal{S} \rhd^+ \mathcal{S}_2; \mathcal{S}$. Assume now that $\mathcal{S}_1 \rhd \mathcal{S}'_1$ and $\mathcal{S}'_1 \rhd^+ \mathcal{S}_2$. By rule (seq) we have $\mathcal{S}_1; \mathcal{S} \rhd \mathcal{S}'_1; \mathcal{S}$ and the I.H. yields $\mathcal{S}'_1; \mathcal{S} \rhd^+ \mathcal{S}_2; \mathcal{S}$, therefore rule (tc2) allows us to conclude $\mathcal{S}_1; \mathcal{S} \rhd^+ \mathcal{S}_2; \mathcal{S}$. □

Definition 6. *We say that a goal sequence $\mathcal{S} =_{def} \mathcal{G}_1, \ldots, \mathcal{G}_k$ is solvable, if $\mathcal{G}_i$ is derivable for all $1 \leq i \leq k$.*

The next lemma shows that solvability of sequences is rearward preserved by the transition relation $\rhd$.

Lemma 2. *If $\mathcal{S}_1 \rhd \mathcal{S}_2$ and the sequence $\mathcal{S}_2$ is solvable then $\mathcal{S}_1$ is solvable.*

Proof. It is clear that the property holds for the basic transitions, since these correspond to the inference rules of our sequent calculus. The result follows by an easy induction on $\rhd$. □

The above property is easily lifted to the transitive closure $\rhd^+$.

Lemma 3. *If $\mathcal{S}_1 \rhd^+ \mathcal{S}_2$ and the sequence $\mathcal{S}_2$ is solvable then $\mathcal{S}_1$ is solvable.*

Proof. Induction on $\mathcal{S}_1 \rhd^+ \mathcal{S}_2$. □

We are now ready to prove the desired equivalence.

Theorem 2 (Equivalence of forward and backward proofs). *Let $\Delta|\Gamma \vdash A$ be any sequent. The following conditions are equivalent:*

- $\Delta|\Gamma \vdash A$ *is derivable*
- $\Delta|\Gamma \vdash A \rhd^+ [\cdot]$

Proof. The $\Leftarrow$) direction is immediate from lemma 3, since the sequence $[\cdot]$ is trivially solvable. We prove the $\Rightarrow$) direction by induction on the derivability of $\Delta|\Gamma \vdash A$.We show here only some cases leaving the others to the reader.

- Case (VHYP). The `vassumption` tactic yields $\Delta, \mathcal{H} : A; \Delta'|\Gamma \vdash A \rhd [\cdot]$.
- Case ($\wedge$ R). A backward derivation of $\Delta|\Gamma \vdash A \wedge B$ is the following:

  $$\Delta|\Gamma \vdash A \wedge B \rhd \Delta|\Gamma \vdash A; \Delta|\Gamma \vdash B \rhd^+ [\cdot]; \Delta|\Gamma \vdash B \rhd^+ [\cdot]; [\cdot] = [\cdot]$$

  The `split` tactic yields the first transition, after that, the second step is an application of the rule (seq^+) using the IH $\Delta|\Gamma \vdash A \rhd^+ [\cdot]$; finally, the last step is justified by the IH $\Delta|\Gamma \vdash B \rhd^+ [\cdot]$.
- Case ($\rightarrow$ LV). The backward derivation of $\Delta, A \rightarrow B; \Delta'|\Gamma \vdash B$ is

  $$\Delta, A \rightarrow B; \Delta'|\Gamma \vdash B \rhd \Delta, A \rightarrow B; \Delta'|\Gamma \vdash A \rhd^+ [\cdot]$$

  where the first step corresponds to the `apply` tactic and the second is given by the IH.
- Case (CUT). A backward derivation of $\Delta|\Gamma \vdash B$ is given by the following transition sequence:

  $$\Delta|\Gamma \vdash B \rhd \Delta|\Gamma \vdash A \ ; \ \Delta|\Gamma, A \vdash B \rhd^+ [\cdot]; \Delta|\Gamma, A \vdash B \rhd^+ [\cdot]; [\cdot] = [\cdot]$$

  where the first step is an instance of the `assert` A tactic, and the remaining are gained from the inductive hypotheses.

The above theorem guarantees the reliability of the proof-search process: if $\Delta|\Gamma \vdash A$ is derivable then the interactive proof-search process succeeds and viceversa. This finishes our exposition.

5 Final Remarks

In this paper we presented a dual sequent calculus $\mathcal{GS}4$ for the constructive modal logic S4 of necessity, and showed that it is adequate for interactive backward proof-search. The rules for handling the necessity operator are simple and intuitive due to the use of dual contexts, a feature that also let us define, in a simple way, a bottom-up construction process by means of a left-to-right depth-first proof-search. This procedure was captured with a formal notion of backward proof which results equivalent to the usual definition of forward proof. This work is the first step in a study of proof-search in modal logic whose natural continuation consists in the proof of the cut-elimination theorem in order to validate if $\mathcal{GS}4$ is loop-free and then can be also adequate for full automatization. As usual,

a proof of cut-elimination is not trivial. In our particular case we actually need to eliminate the two versions (Cut) and (CutV) of the cut-rule. However, the rules ($\Box$L) and ($\Box$LV) for the necessity operator allow us to prove the admissibility of (CutV) from that of (Cut) in a direct way. We are currently finding out if we can do the same for the left rules for valid contexts, a feature that would heavily simplify the proof of cut-elimination for the rule (Cut). Another part of this future inquiry is the extension of the current approach to the full modal logic S4, both constructively, where $\Diamond$ is not a dual of $\Box$, but also classically. In this last case additional research about proof-search with negation is required. The final purpose of this program is to show the utility of our deductive systems for actual case studies in some specific areas, like argumentation theory in the lines of [3,8] or proof-search in multi-agent dialogues related to the work of [19].

References

1. Julius Andrikonis. Loop-free calculus for modal logic s4. i. *Lithuanian Mathematical Journal*, 52(1):1–12, Jan 2012.
2. Arnon Avron, Furio Honsell, Marino Miculan, and Cristian Paravano. Encoding Modal Logics in Logical Frameworks. *Studia Logica*, 60(1):161–208, Jan 1998.
3. Guido Boella, Joris Hulstijn, and Leendert van der Torre. A logic of abstract argumentation. In Simon Parsons, Nicolas Maudet, Pavlos Moraitis, and Iyad Rahwan, editors, *Argumentation in Multi-Agent Systems*, pages 29–41, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
4. Tijn Borghuis and Loe Feijs. A constructive logic for services and information flow in computer networks. *The Computer Journal*, 43(4):274–289, 01 2000.
5. Lourdes del Carmen Gonzlez-Huesca, Favio E. Miranda-Perea, and P. Selene Linares-Arévalo. Axiomatic and dual systems for constructive necessity, a formally verified equivalence. *Journal of Applied Non-Classical Logics*, 29(3):255–287, 2019.
6. Melvin C. Fitting. *Proof Methods for Modal and Intuitionistic Logics*. Synthese Library. Springer Netherlands, 1983.
7. D. M. Gabbay. Introduction to labelled deductive systems. In *Handbook of Philosophical Logic*, pages 179–266. Springer Netherlands, 2014.
8. Davide Grossi. Argumentation in the view of modal logic. In Peter McBurney, Iyad Rahwan, and Simon Parsons, editors, *Argumentation in Multi-Agent Systems*, pages 190–208, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
9. Raul Hakli and Sara Negri. Does the deduction theorem fail for modal logic? *Synthese*, 187(3):849–867, 2012.
10. Samuli Heilala and Brigitte Pientka. Bidirectional decision procedures for the intuitionistic propositional modal logic is4. pages 116–131, 09 2007.
11. Stig Kanger. Provability in logic. In Ghita Holmström-Hintikka, Sten Lindström, and Rysiek Sliwinski, editors, *Collected Papers of Stig Kanger with Essays on His Life and Work*, volume 1 of *Synthese Library: Studies in Epistemology, Logic, Methodology, and Philosophy of Science*, pages 8–41. Kluwer Academic Publishers (Kluwer Academic Publishers Group), 2001. Originally published as Provability in Logic. Acta Universitatis Stockholrnensis, Stockholm Studies in Philosophy I. University of Stockholm 1957.

12. Pablo López, Frank Pfenning, Jeff Polakow, and Kevin Watkins. Monadic concurrent linear logic programming. In *Proceedings of the 7th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming, July 11-13 2005, Lisbon, Portugal*, pages 35–46, 2005.
13. Per Martin-Löf. On the meanings of the logical constants and the justifications of the logical laws. *Nordic J. Philos. Logic*, 1(1):11–60, 1996.
14. Favio E. Miranda-Perea, P. Selene Linares-Arévalo, and Atocha Aliseda-Llera. How to prove it in natural deduction: A tactical approach. *CoRR*, abs/1507.03678, 2015.
15. Tom Murphy VII, Karl Crary, Robert Harper, and Frank Pfenning. A symmetric modal lambda calculus for distributed computing. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science*, LICS '04, pages 286–295, Washington, DC, USA, 2004. IEEE Computer Society.
16. Frank Pfenning and Rowan Davies. A judgmental reconstruction of modal logic. *Mathematical Structures in Comp. Sci.*, 11(4):511–540, 2001.
17. Regimantas Pliukeviius and Aida Pliukeviien. A new method to obtain termination in backward proof search for modal logic s4. *Journal of Logic and Computation*, 20(1):353–379, 11 2008.
18. Morten Heine Sørensen and Pawel Urzyczyn. *Lectures on the Curry-Howard Isomorphism, Volume 149 (Studies in Logic and the Foundations of Mathematics)*. Elsevier Science Inc., New York, NY, USA, 2006.
19. Martin Sticht. *Proof Search in Multi-Agent Dialogues for Modal Logic*. PhD thesis, University of Bamberg Press, Universitätsbibliothek Bamberg, 2018. doctoralthesis.