Automating Reasoning with Standpoint Logic via Nested Sequents

Tim S. Lyon , **Lucía Gómez Álvarez**
Computational Logic Group, Faculty of Computer Science, TU Dresden
{timothy_stephen.lyon, lucia.gomez_alvarez}@tu-dresden.de

Abstract

Standpoint logic is a recently proposed formalism in the context of knowledge integration, which advocates a multi-perspective approach permitting reasoning with a selection of diverse and possibly conflicting standpoints rather than forcing their unification. In this paper, we introduce nested sequent calculi for propositional standpoint logics—proof systems that manipulate trees whose nodes are multisets of formulae—and show how to automate standpoint reasoning by means of non-deterministic proof-search algorithms. To obtain worst-case complexity-optimal proof-search, we introduce a novel technique in the context of nested sequents, referred to as *coloring*, which consists of taking a formula as input, guessing a certain coloring of its subformulae, and then running proof-search in a nested sequent calculus on the colored input. Our technique lets us decide the validity of standpoint formulae in CoNP since proof-search only produces a *partial* proof relative to each permitted coloring of the input. We show how all partial proofs can be fused together to construct a complete proof when the input is valid, and how certain partial proofs can be transformed into a counter-model when the input is invalid. These "certificates" (i.e. proofs and counter-models) serve as explanations of the (in)validity of the input.

1 Introduction

Standpoint Logic. The fact that knowledge bases (KBs) encode the standpoints of their creators (e.g. in the form of viewpoints, contextual factors or semantic commitments) is the source of well-known challenges in the area of knowledge integration. Since semantic heterogeneity between the sources is to be expected, inconsistencies may arise if we attempt to combine them into a single conflict-free conceptual model. To illustrate this, consider three KBs: C, a 'common-sense' representation of colours; H, a KB by a *house painting* business that reuses and extends C; and R, a KB that formalizes the *RYB color model*, from the fine arts tradition.

Example 1. *According to* C*, basic colours such as* `Blue` *and* `Green` *are disjoint.* H *complies with* C *and further specifies that* `Teal` *is* `Green`*. In contrast, according to* R *it is unequivocal that* `Teal` *is both* `Green` *and* `Blue`*. Generally, it is conceivable that* `Blue` *holds.*

These sources cannot be merged without the undesired effect of inconsistency, and circumventing it requires either knowledge weakening or duplication (Pesquita et al. 2013). Instead, one may wish to jointly reason with the KBs, treating them as alternative standpoints on a domain.

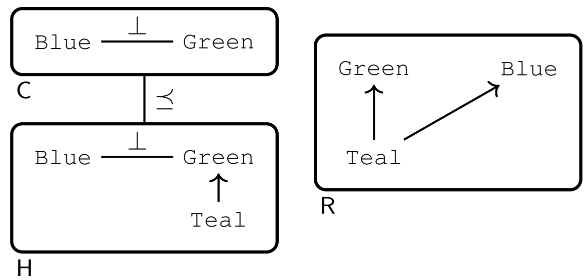


Figure 1: Diagrams of C, H and R. $\preceq$ indicates that H extends C.

Standpoint logic (Gómez Álvarez and Rudolph 2021) is a simple multi-modal logic intended for the representation of knowledge relative to different, possibly conflicting, perspectives. The framework introduces the labeled modalities $\Box_s$ and $\Diamond_s$ for each standpoint s, where $\Box_s \varphi$ is read as "according to s, it is *unequivocal* that φ" and $\Diamond_s \varphi$ as "according to s, it is *conceivable* that φ". In addition, $s \preceq s'$ indicates that the standpoint s is *sharper* than s', that is, s complies with s' and further specifies it.

(F1) $\Box_{\mathsf{C}} \neg(\texttt{Blue} \land \texttt{Green}) \land \Box_{\mathsf{R}}(\texttt{Teal} \rightarrow (\texttt{Blue} \land \texttt{Green}))$

(F2) $(\mathsf{H} \preceq \mathsf{C})$ (F3) $\Box_{\mathsf{H}}(\texttt{Teal} \rightarrow \texttt{Green})$ (F4) $\Diamond_{*}(\texttt{Blue})$

The formulae (F1)-(F4) formalize Example 1, (illustrated in Figure 1) in propositional standpoint logic. (F1) encodes that `Blue` and `Green` are unequivocally disjoint according to standpoint C, while according to standpoint R it is unequivocal that `Teal` implies both `Blue` and `Green`. (F2) encodes that H includes the knowledge of C, and (F3) that `Teal` is `Green` according to H. Last, (F4) encodes that `Blue` holds under some interpretations by using the universal standpoint $*$, which sits atop any hierarchy of standpoints and is used to reference knowledge that is unequivocally true or conceivable among all perspectives.

In addition to representing unequivocal and conceivable facts (e.g. (F3) and (F4)), which may be relative to standpoints (e.g. (F1)), hold universally (e.g. (F4)), or establish a hierarchy of standpoints (e.g. (F2)), one may also express (in)determinate knowledge by means of the (definable) dual operators $\mathcal{I}_s$ and $\mathcal{D}_s$. The indeterminacy operator

$\mathcal{I}_s\varphi := \Diamond_s \varphi \land \Diamond_s \neg\varphi$ makes explicit that both φ and $\neg\varphi$ are conceivable in the context of s, thus making φ inherently indeterminate. Finally, the framework can be used to establish correspondences or bridges between the standpoints themselves. For instance, (F5) encodes that if something is $\texttt{Teal}$ according to R, then it is $\texttt{Green}$ for C and R.

(F5) $\Box_{\mathsf{R}}(\texttt{Teal}) \to (\Box_{\mathsf{C}}\, \texttt{Green} \land \Box_{\mathsf{R}}\, \texttt{Green})$

Natural reasoning tasks over multi-standpoint specifications include gathering unequivocal or undisputed knowledge, determining knowledge that is relative to a standpoint or a set of them, and contrasting the knowledge that can be inferred from different standpoints. To illustrate, let us assume $\Box_* \,\texttt{Teal}$ and examine some inferences that we can draw from this in the setting of Example 1. On the one hand, from (F5), (F3), and $\Box_* \,\texttt{Teal}$ we obtain that green is unequivocal for the three standpoints: $\Box_{\mathsf{C}}\,\texttt{Green}$, $\Box_{\mathsf{H}}\,\texttt{Green}$ and $\Box_{\mathsf{R}}\,\texttt{Green}$. On the other hand, we can infer the global indeterminacy of blue $\mathcal{I}_*\texttt{Blue}$, because (i) $\texttt{Teal}$ holds universally, (ii) it is unequivocal for R that $\texttt{Teal}$ implies $\texttt{Blue}$ (F1), hence $\Diamond_* \,\texttt{Blue}$, and (iii) we know $\Box_{\mathsf{C}}\,\texttt{Green}$, which together with (F1) implies $\Box_{\mathsf{C}}\, \neg\texttt{Blue}$ and thus $\Diamond_* \,\neg\texttt{Blue}$.

Conveniently, the satisfiability problem in propositional standpoint logic is known to be NP-complete (Gómez Álvarez and Rudolph 2021), in pleasant contrast to the PSPACE-completeness normally exhibited by multimodal epistemic logics, such as the closely related $\mathsf{KD45}_n$.[1] This result, obtained via a translation to one-variable first-order logic, makes the framework attractive in applied scenarios, and prompts our work to provide a suitable proof-theory for standpoint logic. Not only can our proof systems be leveraged to provide a proof-search procedure deciding the validity of standpoint formulae, but our proof-theoretic approach yields *witnesses*, that is, proofs of valid formulae and counter-models of invalid formulae. Such "certificates" (i.e. proofs and counter-models) possess explanatory value, and may be used, for instance, to trace the standpoints involved in a certain inference; e.g. when a global indeterminacy such as $\mathcal{I}_*\texttt{Blue}$ is inferred from a large collection of standpoints, we may want to gather the standpoints that hold contrasting views (in this case R and $\{\mathsf{H}, \mathsf{C}\}$, which can be easily extracted from a proof). Thus, our reliance on proof theory provides essential information that may be used to answer "why" a certain piece of information holds while still allowing "low" complexity reasoning.

Nested Sequents and Proof Theory. Since their inception, sequent systems—which consist of inference rules that syntactically manipulate pairs of multisets of formulae—have proven themselves fruitful in writing decision algorithms for logics (Dyckhoff 1992; Gentzen 1935a; Gentzen 1935b; Slaney 1997). A crucial feature of such systems, and their use in decidability, is the so-called *subformula property*, which a sequent system has *iff* the premise(s) of each inference rule only contain subformulae of the conclusion of the rule. (NB. Systems with the subformula property are also referred to as *analytic*.) With the goal of securing this property for proof systems for theories *beyond classical propositional logic* (e.g. the modal logics Kt and $\mathsf{S5}$), more sophisticated sequent systems extending Gentzen's original formalism were eventually proposed; e.g., see (Belnap 1982; Simpson 1994; Wansing 2002). In this paper, we employ one such extended formalism, viz. the *nested sequent formalism* (Brünnler 2009; Bull 1992; Kashima 1994; Poggiolesi 2009), which utilizes trees of multisets of formulae in deriving theorems. Such systems have proven well-suited for automated reasoning with modal and related logics, being used (for instance) in the writing of decision/proof-search algorithms (Brünnler 2009; Tiu, Ianovski, and Goré 2012) and the extraction of interpolants (Fitting and Kuznets 2015; Lyon et al. 2020).

Drawing on ideas from the *structural refinement* methodology, detailed in (Lyon 2021a) and used to provide nested sequent systems for diverse classes of modal and constructive logics (see (Lyon and van Berkel 2019; Lyon 2020; Lyon 2021b)), our first contribution in this paper is the introduction of analytic nested sequent systems (each dubbed $\mathsf{NS}(\mathcal{V})$ with $\mathcal{V}$ a certain parameter) for propositional standpoint logics (Gómez Álvarez and Rudolph 2021). For our second contribution, we exploit our nested systems to write concrete, *worst-case complexity-optimal* proof-search algorithms (deciding the validity of propositional standpoint formulae in CoNP), which apply inference rules from $\mathsf{NS}(\mathcal{V})$ *in reverse* on an input formula with the goal of building a proof thereof. Whereas typical proof-search algorithms operate deterministically and attempt to build a *complete* proof of the input, we introduce a novel technique (our third contribution) referred to as *coloring*, which performs proof-search *non-deterministically* and which only constructs a *partial* proof of the input relative to each non-deterministic choice. The technique of coloring involves first guessing a particular labeling of the subformulae of an input formula with *active* $\circ$ and *inactive* $\bullet$ labels, with the proof-search algorithm subsequently only processing data deemed active. An interesting consequence of this technique is the attainment of a CoNP proof-search algorithm as the partial proofs constructed during proof-search are at most polynomially larger than the input and only require polynomial time to compute. Moreover, in the instance where the input formula is invalid, we show how to construct a counter-model from failed proof-search, and in the instance where our input formula is valid, we provide a procedure that generates a *complete* proof witnessing the validity of the input formula by patching together all *partial* proofs (our fourth contribution).

Organization of Paper. Our paper is organized as follows: Sect. 2 presents the syntax and semantics of propositional standpoint logic. In Sect. 3, we introduce our nested sequent systems for propositional standpoint logics, proving such systems sound and concluding their completeness. In the penultimate section (Sect. 4), we introduce the method of coloring and show how to automate reasoning with standpoint logics, that is, we provide a (worst-case complexity-optimal) proof-search algorithm deciding the validity of propositional standpoint formulae in CoNP. The final section (Sect. 5) concludes the paper and discusses future work.

[1] Standpoint logic introduces sharpenings and stronger interaction axioms than $\mathsf{KD45}_n$, as discussed in (Gómez Álvarez 2020).

2 Standpoint Logic

Let us now specify the syntax of propositional standpoint logic (SL), denoted by $\mathbb{S}$.

Definition 1 (Syntax of Standpoint Logic)**.** *Let $\mathcal{V} = \langle \mathcal{P}, \mathcal{S} \rangle$ be a* vocabulary *where $\mathcal{P}$ is a non-empty set of propositional variables and $\mathcal{S}$ is a set of standpoint symbols containing the distinguished symbol $*$, i.e. the* universal standpoint*. We define the language $\mathcal{L}_{\preceq} := \{s \preceq s' \mid s, s' \in \mathcal{S}\}$, and refer to formulae in $\mathcal{L}_{\preceq}$ as* sharpening statements*. The language $\mathcal{L}_{\mathcal{V}}$ is defined via the following grammar in BNF:*

$$\varphi ::= p \mid \neg p \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid \Box_s \varphi \mid \Diamond_s \varphi$$

where $p \in \mathcal{P}$ and $s \in \mathcal{S}$. We also use $\top$ and $\bot$ as shorthands with the usual definitions.

Last, for $\Gamma \subseteq \mathcal{L}_{\preceq}$ and $\varphi \in \mathcal{L}_{\mathcal{V}}$, we define a standpoint implication *to be a formula of the form $\bigwedge \Gamma \to \varphi$, where $\bigwedge \Gamma$ is a conjunction of all elements of Γ, which equals $\top$ when Γ is empty.*

We make use of formulae in negation normal form as this will simplify the structures present in our nested systems and enhance the readability of our proof theory. To further simplify, we also assume w.l.o.g. that sets of sharpening statements are (1) free of cycles $s_1 \preceq s_2, \ldots, s_n \preceq s_1$ and (2) omit occurrences of $*$. Assumption (1) is permitted since any standpoint implication containing a cycle $s_1 \preceq s_2, \ldots, s_n \preceq s_1$ of standpoints is equivalent to one where the cycle is deleted and all occurrences of $s_1, \ldots, s_{n-1}$ are replaced by s_n in the formula. Regarding assumption (2), any sharpening statement with $*$ is either of the form $s \preceq *$, and is thus trivial (see Def. 4 below), or is of the form $* \preceq s$, in which case s can be systematically replaced by $*$ in a standpoint implication to obtain an equivalent one.

Definition 2 (Subformula and Size)**.** *We define the set of* subformulae *of φ, denoted $\mathrm{sufo}(\varphi)$, recursively as follows:*

- $\mathrm{sufo}(p) := \{p\}$ *and* $\mathrm{sufo}(\neg p) := \{\neg p\}$*;*
- $\mathrm{sufo}(\heartsuit\psi) := \{\heartsuit\psi\} \cup \mathrm{sufo}(\psi)$*;*
- $\mathrm{sufo}(\psi \otimes \chi) := \{\psi \otimes \chi\} \cup \mathrm{sufo}(\psi) \cup \mathrm{sufo}(\chi)$.

with $\heartsuit \in \{\Diamond_s, \Box_s \mid s \in \mathcal{S}\}$ and $\otimes \in \{\vee, \wedge\}$. We say that ψ is a subformula *of φ iff $\psi \in \mathrm{sufo}(\varphi)$, and define the* size *of a formula φ in $\mathcal{L}_{\mathcal{V}}$, denoted $|\varphi|$, to be equal to $|\mathrm{sufo}(\varphi)|$, i.e. to the number of its subformulae.*

In what follows, we introduce the semantics of SL, defined over a structure of precisifications, which is akin to the usual structure of possible worlds. A *precisification* is a complete and consistent way in which the state of affairs can be described with a given vocabulary, and standpoints are modeled as sets of precisifications considered admissible. This strategy of modelling the variability of natural language as hyper-ambiguity is based on the theory supervaluationism (Fine 1975; Keefe and Smith 1997), which standpoint logic draws from (Gómez Álvarez and Bennett 2018; Gómez Álvarez, Bennett, and Richard-Bollans 2017).

Definition 3 (Standpoint Model)**.** *Given a vocabulary $\mathcal{V}$, a model $\mathcal{M}$ (over $\mathcal{V}$) is a triple $\langle \Pi, \sigma, \delta \rangle$, where Π is a non-empty set of precisifications, $\sigma : \mathcal{S} \to 2^{\Pi}$, and $\delta : \mathcal{P} \to 2^{\Pi}$ with $\sigma(s) \neq \emptyset$ for all $s \in \mathcal{S}$ and $\sigma(*) = \Pi$. The set of all such models is denoted by $\mathfrak{M}_{\mathbb{S}}$.*

Definition 4 (Semantic Clauses)**.** *Let $\Gamma \subseteq \mathcal{L}_{\preceq}$ and $\varphi, \psi \in \mathcal{L}_{\mathcal{V}}$. Moreover, let $\mathcal{M} = \langle \Pi, \sigma, \delta \rangle$ be a standpoint model with $\pi \in \Pi$. We recursively define the satisfaction of a formula on $\mathcal{M}$ at π accordingly:*

- $\mathcal{M}, \pi \models p$ *iff* $\pi \in \delta(p)$*;*
- $\mathcal{M}, \pi \models \neg p$ *iff* $\pi \notin \delta(p)$*;*
- $\mathcal{M}, \pi \models \varphi \wedge \psi$ *iff* $\mathcal{M}, \pi \models \varphi$ *and* $\mathcal{M}, \pi \models \psi$*;*
- $\mathcal{M}, \pi \models \varphi \vee \psi$ *iff* $\mathcal{M}, \pi \models \varphi$ *or* $\mathcal{M}, \pi \models \psi$*;*
- $\mathcal{M}, \pi \models \Diamond_s \varphi$ *iff for some* $\pi' \in \sigma(s)$, $\mathcal{M}, \pi' \models \varphi$*;*
- $\mathcal{M}, \pi \models \Box_s \varphi$ *iff for all* $\pi' \in \sigma(s)$, $\mathcal{M}, \pi' \models \varphi$*;*
- $\mathcal{M}, \pi \models s \preceq s'$ *iff* $\sigma(s) \subseteq \sigma(s')$*;*
- $\mathcal{M}, \pi \models \bigwedge \Gamma$ *iff* $\mathcal{M}, \pi \models s \preceq s'$ *for all* $s \preceq s' \in \Gamma$*;*
- $\mathcal{M}, \pi \models \bigwedge \Gamma \to \varphi$ *iff* $\mathcal{M}, \pi \models \bigwedge \Gamma$ *implies* $\mathcal{M}, \pi \models \varphi$*;*
- $\mathcal{M} \models \bigwedge \Gamma \to \varphi$ *iff* $\mathcal{M}, \pi \models \bigwedge \Gamma \to \varphi$ *for all* $\pi \in \Pi$.

A standpoint implication $\bigwedge \Gamma \to \varphi$ is defined to be valid *(relative to a vocabulary $\mathcal{V}$) iff it is true on each model $\mathcal{M} \in \mathfrak{M}_{\mathbb{S}}$; it is defined to be* invalid *(relative to $\mathcal{V}$) otherwise.*

For a vocabulary $\mathcal{V}$, the standpoint logic $\mathbb{S}(\mathcal{V})$ *is the set of all valid standpoint implications $\bigwedge \Gamma \to \varphi$ over $\mathfrak{M}_{\mathbb{S}}$.*

It is worth remarking that the specification of sharpening statements in a separate language (viz. $\mathcal{L}_{\preceq}$) and the above definition of satisfiability and validity contrast with the original presentation in (Gómez Álvarez and Rudolph 2021). However, this specification simplifies our treatment of sharpening statements, which previously served as atomic propositions in the language $\mathcal{L}_{\mathcal{V}}$. In fact, these statements are obsolete in extensions of the language allowing set theoretical combinations of standpoints in modalities (which is the object of current research). Moreover, in these extensions, the natural requirement of inner consistency (i.e. the non-emptiness of $\sigma(s)$, for each $s \in \mathcal{S}$) of standpoints is relaxed, which can be easily reflected in our nested sequent systems by dropping the (n_s) rule (see Fig. 2 in Section 3).

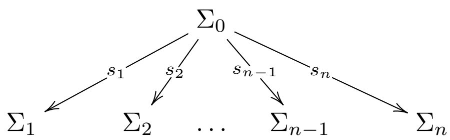
3 Nested Sequent Systems

We define a *nested sequent* (which we will also refer to as a *sequent*) to be a formula of the form $\Gamma \vdash \Delta$ with Γ and Δ defined via the following grammars in BNF:

$$\Gamma ::= s \preceq s' \mid \emptyset \mid \Gamma, \Gamma \quad \Delta ::= \Sigma \mid \Delta, (s)[\Sigma]_{\pi}$$

$$\Sigma ::= \varphi \mid \emptyset \mid \Sigma, \Sigma$$

where $s, s' \in \mathcal{S} \setminus \{*\}$, $\varphi \in \mathcal{L}_{\mathcal{V}}$, and π is among a countably infinite set of labels $\{\pi_i \mid i \in \mathbb{N} \setminus \{0\}\}$. We use Φ and Ψ (occasionally annotated) to denote nested sequents and note that we employ the use of labels as this proves useful in extracting a counter-model from failed proof-search (see Thm. 3). Moreover, each nested sequent $\Gamma \vdash \Delta$ with $\Delta = \Sigma_0, (s_1)[\Sigma_1]_{\pi_1}, \ldots, (s_n)[\Sigma_n]_{\pi_n}$ possesses a special structure; namely, the *antecedent* Γ is a set of sharpening statements of the form $s \preceq s'$, and the *consequent* Δ is a multiset encoding a tree of depth 1 whose nodes are multisets of formulae from $\mathcal{L}_{\mathcal{V}}$. The consequent Δ can be expressed graphically as follows:



We refer to a multiset Σ_i occurring in the consequent of a nested sequent as a *component*, and note that components (along with the antecedent and consequent) are permitted to be empty $\emptyset$. Intuitively, components correspond to precisifications in a standpoint model. It is also worthwhile to define the relation $\preceq_\Gamma^*$ on standpoints as this will be used as a side condition dictating applications of certain inference rules:

Definition 5. *For a nested sequent $\Gamma \vdash \Delta$, let $\preceq_\Gamma^* \subseteq \mathcal{S} \times \mathcal{S}$ be the minimal reflexive and transitive relation such that*

- *$s \preceq_\Gamma^* s'$ for every $s \preceq s' \in \Gamma$, and*
- *$s \preceq_\Gamma^* *$ for every $s \in \mathcal{S}$.*

A nice feature of nested sequents is that such objects typically permit a formula translation, e.g. (Brünnler 2009; Bull 1992; Kashima 1994; Poggiolesi 2009), meaning that our logical semantics can be lifted to the language of our proof systems without introducing an extended semantics for nested sequents.

Definition 6 (Formula Interpretation)**.** *We define the* formula interpretation *of a nested sequent $\Gamma \vdash \Delta$ with $\Delta = \Sigma_0, (s_1)[\Sigma_1]_{\pi_1}, \ldots, (s_n)[\Sigma_n]_{\pi_n}$ as follows:*

$$\iota(\Gamma \vdash \Delta) := \bigwedge \Gamma \rightarrow \bigvee \Sigma_0 \vee \bigvee_{1 \leq i \leq n} \Box_{s_i}(\bigvee \Sigma_i)$$

We define $\Gamma \vdash \Delta$ to be valid iff $\iota(\Gamma \vdash \Delta)$ is valid. Also, we note that $\bigwedge \emptyset = \top$ and $\bigvee \emptyset = \bot$, as usual.

A uniform presentation of our nested calculi is given in Fig. 2. We let $\mathsf{NS}(\mathcal{V})$ denote the corresponding nested sequent calculus over a vocabulary $\mathcal{V}$. Our inference rules make use of the brackets '{' and '}' in the consequent of a nested sequent to indicate that the displayed formula(e) occur in some component. In particular, given a nested sequent $\Gamma \vdash \Delta$, where Δ is of the form $\Sigma_0, (s_1)[\Sigma_1]_{\pi_1}, \ldots, (s_i)[\Sigma_i]_{\pi_i}, \ldots, (s_n)[\Sigma_n]_{\pi_n}$, the notation $\Gamma \vdash \Delta\{\varphi\}_{\pi_i}$ indicates that φ occurs in Σ_i; additionally, we use $\Gamma \vdash \Delta\{\varphi\}_{\pi_0}$ to indicate that φ occurs in Σ_0, i.e. the label π_0 is used to reference the multiset Σ_0 serving as the root of the tree encoded by the consequent.

To make the functionality of each rule in $\mathsf{NS}(\mathcal{V})$ precise, we explicitly state the operation performed by each rule. With the exception of the premise-free (id) rule, we explain for each rule how the *premise(s)* (the nested sequent(s) occurring above the horizontal inference line) are obtained from the *conclusion* (the nested sequent occurring below the horizontal inference line). This explanation is consistent with how the rules are applied (bottom-up) during proof-search as described in the following section. Also, in accordance with standard proof-theoretic terminology (Buss 1998; Takeuti 2013), we refer to the formula that is explicitly displayed in the conclusion of a rule as *principal*, and indicate the principal formulae in our explanation of the rules below to make this precise for the reader.

(id) A nested sequent is *initial*, and may be used to begin a derivation, so long as some component contains both p and $\neg p$ (the principal formulae);

$(\vee)$ If a component Σ_i of the conclusion contains $\varphi \vee \psi$ (the principal formula), then adding φ and ψ to Σ_i yields the premise;

$(\wedge)$ If a component Σ_i of the conclusion contains $\varphi \wedge \psi$ (the principal formula), then adding φ to Σ_i yields the left premise and adding ψ to Σ_i yields the right premise;

$(\Box_s)$ For any $s \in \mathcal{S}$, if a component Σ_i of the conclusion contains $\Box_s \varphi$ (the principal formula), then appending the consequent Δ with $(s)[\varphi]_{\pi'}$, where π' is fresh (i.e. it does not occur in the conclusion), yields the premise;

(n_s) For any $s \in \mathcal{S}$, we may append the consequent of the conclusion with $(s)[\emptyset]_{\pi'}$ to obtain the premise so long as π' is fresh;

$(\Diamond_s^1)$ For any $s \in \mathcal{S}$, if a component Σ_i of the conclusion contains $\Diamond_s \varphi$ (the principal formula), the consequent contains a nesting $(s')[\Sigma]_{\pi'}$, and $s' \preceq_\Gamma^* s$, then adding φ to the nesting $(s')[\Sigma]_{\pi'}$ yields the premise;

$(\Diamond_s^2)$ For any $s \in \mathcal{S}$, if the consequent contains a nesting of the form $(s')[\Diamond_s \varphi, \Sigma]_{\pi'}$ with $\Diamond_s \varphi$ the principal formula, and $s' \preceq_\Gamma^* s$, then adding φ to the nesting $(s')[\Diamond_s \varphi, \Sigma]_{\pi'}$ yields the premise;

$(\Diamond_*)$ If a component Σ_i of the conclusion contains $\Diamond_* \varphi$ (the principal formula), then prepending the consequent Δ with φ (i.e. adding φ to Σ_0) yields the premise.

Example 2. *Below, we provide an example of a nested sequent derivation. To minimize the width of the proof, we let φ denote $\Diamond_s \Diamond_* \neg p \vee \Box_{s'} p$.*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{s' \preceq s \vdash \varphi, \Diamond_s \Diamond_* \neg p, \Box_{s'} p, (s')[\Diamond_* \neg p, \neg p, p]}\,(id)}{s' \preceq s \vdash \varphi, \Diamond_s \Diamond_* \neg p, \Box_{s'} p, (s')[\Diamond_* \neg p, p]}\,(\Diamond_*^2)}{s' \preceq s \vdash \varphi, \Diamond_s \Diamond_* \neg p, \Box_{s'} p, (s')[p]}\,(\Diamond_s^1)}{s' \preceq s \vdash \varphi, \Diamond_s \Diamond_* \neg p, \Box_{s'} p}\,(\Box_{s'})}{s' \preceq s \vdash \varphi}\,(\vee)$$

Observe that $(\Diamond_s^1)$ is applicable as $s' \preceq_\Gamma^* s$ holds due to the antecedent, and $(\Diamond_*^2)$ is applicable as $s' \preceq_\Gamma^* *$ holds by definition (see Def. 5).

We now prove that our calculi are sound (Thm. 1, building on Lem. 1), that is, that every nested sequent derivable in $\mathsf{NS}(\mathcal{V})$ is valid. We then state our completeness theorem, which is a consequence of the work in Sect. 4.

Lemma 1. *Let $\Gamma \vdash \Delta$ be a sequent, $\mathcal{M} = \langle \Pi, \sigma, \delta \rangle$ be a model with $\pi \in \Pi$, and $s, s' \in \mathcal{S}$. If $\mathcal{M}, \pi \models \bigwedge \Gamma$ and $s' \preceq_\Gamma^* s$, then $\mathcal{M}, \pi \models s' \preceq s$.*

Proof. Assume that $\mathcal{M}, \pi \models \bigwedge \Gamma$ and $s' \preceq_\Gamma^* s$ for some $s', s \in \mathcal{S}$. There are four cases to consider:

(1) $s = *$. The result is immediate as $\sigma(*) = \Pi$, and therefore, $\mathcal{M}, \pi \models s' \preceq *$ for every $s' \in \mathcal{S}$ by Def. 4.

(2) $s' \preceq s \in \Gamma$. From the assumption that $\mathcal{M}, \pi \models \bigwedge \Gamma$ it follows that $\mathcal{M}, \pi \models s' \preceq s$.

$$\frac{}{\Gamma \vdash \Delta\{p, \neg p\}_\pi} (id) \quad \frac{\Gamma \vdash \Delta\{\varphi, \psi\}_\pi}{\Gamma \vdash \Delta\{\varphi \lor \psi\}_\pi} (\lor) \quad \frac{\Gamma \vdash \Delta\{\varphi\}_\pi \qquad \Gamma \vdash \Delta\{\psi\}_\pi}{\Gamma \vdash \Delta\{\varphi \land \psi\}_\pi} (\land)$$

$$\frac{\Gamma \vdash \Delta\{\Box_s \varphi\}_\pi, (s)[\varphi]_{\pi'}}{\Gamma \vdash \Delta\{\Box_s \varphi\}_\pi} (\Box_s)^{\dagger_1} \quad \frac{\Gamma \vdash \Delta, (s)[\emptyset]_{\pi'}}{\Gamma \vdash \Delta} (n_s)^{\dagger_1}$$

$$\frac{\Gamma \vdash \Delta\{\Diamond_s \varphi\}_\pi, (s')[\Sigma, \varphi]_{\pi'}}{\Gamma \vdash \Delta\{\Diamond_s \varphi\}_\pi, (s')[\Sigma]_{\pi'}} (\Diamond_s^1)^{\dagger_2} \quad \frac{\Gamma \vdash \Delta, (s')[\Diamond_s \varphi, \varphi, \Sigma]_\pi}{\Gamma \vdash \Delta, (s')[\Diamond_s \varphi, \Sigma]_\pi} (\Diamond_s^2)^{\dagger_2} \quad \frac{\Gamma \vdash \varphi, \Delta\{\Diamond_* \varphi\}_\pi}{\Gamma \vdash \Delta\{\Diamond_* \varphi\}_\pi} (\Diamond_*)$$

Figure 2: The nested calculus $\mathsf{NS}(\mathcal{V})$ with $\mathcal{V} = \langle \mathcal{P}, \mathcal{S} \rangle$ a vocabulary. We note that π is permitted to be any label from $\{\pi_i \mid i \in \mathbb{N} \setminus \{0\}\}$ and that $\mathsf{NS}(\mathcal{V})$ contains a copy of $(\Box_s)$, (n_s), $(\Diamond_s^1)$, and $(\Diamond_s^2)$ for each $s \in \mathcal{S}$. The side condition $\dagger_1$ stipulates that the rule is applicable only if the label π' is fresh and $\dagger_2$ stipulates that the rule is applicable only if $s' \preceq^*_\Gamma s$.

(3) $s = s'$. Then, it is trivially implied that $\mathcal{M}, \pi \models s' \preceq s$ since $\sigma(s') = \sigma(s) \subseteq \sigma(s)$ by Def. 4.

(4) There are some $s_1, \ldots, s_n \in \mathcal{S}$ such that $s' \preceq s_1, \in \Gamma$, $s_i \preceq s_{i+1} \in \Gamma$ for every $1 \leq i \leq n-1$, and $s_n \preceq s \in \Gamma$, that is, $s' \preceq^*_\Gamma s$ is obtained by transitivity on a path in $\mathcal{S}$. From this, together with the assumption that $\mathcal{M}, \pi \models \bigwedge \Gamma$, it directly follows that $\mathcal{M}, \pi \models s' \preceq s$ by Def. 4. □

Theorem 1 (Soundness)**.** *If* $\Gamma \vdash \Delta$ *is derivable in* $\mathsf{NS}(\mathcal{V})$*, then* $\Gamma \vdash \Delta$ *is valid.*

Proof. We prove the result by induction on the number of inferences in a given derivation, and assume that Δ is of the form $\Sigma_0, (s_1)[\Sigma_1], \ldots, (s_n)[\Sigma_n]$.

Base case. In the base case, our derivation consists of a single application of the (id) rule. Hence,

$$\iota(\Gamma \vdash \Delta\{p, \neg p\}_{\pi_i}) := \bigwedge \Gamma \to \bigvee \Sigma_0 \lor \bigvee_{1 \leq i \leq n} \Box_{s_i}(\bigvee \Sigma_i)$$

where $p, \neg p \in \Sigma_i$, for some $0 \leq i \leq n$. Regardless, the consequent of the implication above will be satisfied in any model $\mathcal{M}$, implying that the above implication is valid.

Inductive step. We make a case distinction based on the last rule applied, and show that if the conclusion of the rule is invalid, then at least one of the premises of the rule is invalid, that is to say, we show by contraposition that if the premise(s) is (are) valid, then the conclusion is valid. We only show the $(\Box_s)$ and $(\Diamond_s^1)$ cases as the remaining cases are simple or argued in a similar manner.

$(\Box_s)$. We assume that Σ_1 is of the form $\Box_s \varphi, \Sigma'_1$ with $\Box_s \varphi$ principal; all remaining cases are similar. Furthermore, let us suppose that $\iota(\Gamma \vdash \Delta\{\Box_s \varphi\}_{\pi_1}) :=$

$$\bigwedge \Gamma \to \bigvee \Sigma_0 \lor \Box_{s_1}(\Box_s \varphi \lor \bigvee \Sigma'_1) \lor \bigvee_{2 \leq i \leq n} \Box_{s_i}(\bigvee \Sigma_i)$$

is invalid. Then, $\mathcal{M}, \pi \not\models \Box_{s_1}(\Box_s \varphi \lor \bigvee \Sigma'_1)$ for some standpoint model $\mathcal{M} := \langle \Pi, \sigma, \delta \rangle$ with a precisification π. Hence, there exists a precisification $\pi' \in \sigma(s_1)$ such that $\mathcal{M}, \pi' \not\models \Box_s \varphi$, implying that there exists a precisification $\pi'' \in \sigma(s)$ such that $\mathcal{M}, \pi'' \not\models \varphi$. It thus follows that $\mathcal{M}, \pi \not\models \Box_s \varphi$, showing that the premise of $(\Box_s)$ is invalid.

$(\Diamond_s^1)$. Suppose that Σ_1 is of the form $\Diamond_s \varphi, \Sigma'_1$ with $\Diamond_s \varphi$ principal; all remaining cases are argued in a similar fashion. Assume that $s' \preceq^*_\Gamma s$ holds and that the following is invalid:

$$\iota(\Gamma \vdash \Delta\{\Diamond_s \varphi\}_\pi, (s')[\Sigma]_{\pi'}) := \bigwedge \Gamma \to \bigvee \Sigma_0 \lor$$
$$\Box_{s_1}(\Diamond_s \varphi \lor \bigvee \Sigma'_1) \lor \Big(\bigvee_{2 \leq i \leq n} \Box_{s_i} (\bigvee \Sigma_i) \Big) \lor \Box_{s'} (\bigvee \Sigma)$$

Therefore, there is a standpoint model $\mathcal{M} := \langle \Pi, \sigma, \delta \rangle$ with $\pi \in \Pi$ such that $\mathcal{M}, \pi \not\models \Box_{s'}(\bigvee \Sigma)$ and $\mathcal{M}, \pi \not\models \Box_{s_1}(\Diamond_s \varphi \lor \bigvee \Sigma'_1)$, and such that $\sigma(s') \subseteq \sigma(s)$, by $s' \preceq^*_\Gamma s$ and Lem. 1. This entails that there exists a precisification $\pi' \in \sigma(s')$ such that $\mathcal{M}, \pi' \not\models \bigvee \Sigma$, and that there exists a precisification $\pi_1 \in \sigma(s_1)$ such that $\mathcal{M}, \pi_1 \not\models \Diamond_s \varphi$. The latter further implies that for every precisification in $\sigma(s)$, and thus for π' (since $\pi' \in \sigma(s') \subseteq \sigma(s)$), that $\mathcal{M}, \pi' \not\models \varphi$. Thus, the premise has been shown invalid. □

Theorem 2 (Completeness)**.** *If* $\Gamma \vdash \Delta$ *is valid, then* $\Gamma \vdash \Delta$ *is provable in* $\mathsf{NS}(\mathcal{V})$*.*

Proof. The theorem follows from the correct (Thm. 3) and terminating (Thm. 4) proof-search procedure given in the subsequent section (Sect. 4). □

4 Automating Standpoint Logic via Proof-Search

We now employ our nested calculi in an algorithm that decides the validity of formulae for propositional standpoint logics. In particular, we design a proof-search algorithm (see Alg. 1 below) which takes a vocabulary $\mathcal{V}$ as a parameter and bottom-up applies rules from $\mathsf{NS}(\mathcal{V})$ in attempt to construct a proof of a given input sequent $\Gamma \vdash \varphi$. We may make the simplifying assumption that our proof-search algorithm only receives inputs of the form $\Gamma \vdash \varphi$ as any nested sequent $\Gamma \vdash \Delta$ with $\Delta = \Sigma_0, (s_1)[\Sigma_1], \ldots, (s_n)[\Sigma_n]$ is valid *iff* $\iota(\Gamma \vdash \Delta)$ is valid *iff* $\Gamma \vdash \varphi$ is valid, where $\varphi = \bigvee \Sigma_0 \lor \Box_{s_1}(\bigvee \Sigma_1) \lor \cdots \lor \Box_{s_n}(\bigvee \Sigma_n)$.

To decrease the complexity of proof-search and obtain (worst-case) complexity-optimality, we introduce a new technique in the context of nested sequents which we refer to as *coloring*. In essence, given the input $\Gamma \vdash \varphi$, the first step of proof-search guesses a *proper coloring* of the formula φ, that is, it labels the formula's subformulae with either an *active* label $\circ$ or an *inactive* label $\bullet$ in a particular manner. Recall that, due to the $(\land)$ rule, a proof in $\mathsf{NS}(\mathcal{V})$ has the structure of a binary tree, thus giving rise to the possibility

Algorithm 1: $\texttt{Prove}_{\mathcal{V}}$

Input: A Nested Sequent: $\Gamma \vdash \varphi$
Output: A Boolean: $\texttt{True}$, $\texttt{False}$
Choose a proper coloring $\widetilde{\varphi}$ of φ;
return $\texttt{ProofSearch}_{\mathcal{V}}(\Gamma \vdash \widetilde{\varphi})$;

Algorithm 2: $\texttt{ProofSearch}_{\mathcal{V}}$ (Part I)

Input: A Colored Nested Sequent:
$\Phi := \Gamma \vdash \Sigma_0, (s_1)[\Sigma_1]_{\pi_1}, \ldots, (s_n)[\Sigma_n]_{\pi_n}$
Output: A Boolean: $\texttt{True}$, $\texttt{False}$
1 **if** *for some* $0 \leq i \leq n$, $p^\circ, \neg p^\circ \in \Sigma_i$ **then**
2 | **return** $\texttt{True}$;
3 **end**
4 **if** Σ *is saturated* **then**
5 | **return** $\texttt{False}$;
6 **end**
7 **if** *for some* $0 \leq i \leq n$, $(\widetilde{\varphi} \vee \widetilde{\psi})^\circ \in \Sigma_i$, *but* $\widetilde{\varphi}, \widetilde{\psi} \notin \Sigma_i$ **then**
8 | Let $\Sigma'_i := \Sigma_i, \widetilde{\varphi}, \widetilde{\psi}$;
9 | Let $\Phi' := \Gamma \vdash \Sigma_0, \ldots, (s_i)[\Sigma'_i]_{\pi_i}, \ldots, (s_n)[\Sigma_n]_{\pi_n}$;
| // Replace Σ_i by Σ'_i to obtain Φ'.
10 | **return** $\texttt{Prove}_{\mathcal{V}}(\Phi')$;
11 **end**
12 **if** *for some* $0 \leq i \leq n$, $(\widetilde{\varphi}^\circ \wedge \widetilde{\psi}^\bullet)^\circ \in \Sigma_i$, *but* $\widetilde{\varphi}^\circ \notin \Sigma_i$ **then**
13 | Let $\Sigma'_i := \Sigma_i, \widetilde{\varphi}^\circ$;
14 | Let $\Phi' := \Gamma \vdash \Sigma_0, \ldots, (s_i)[\Sigma'_i]_{\pi_i}, \ldots, (s_n)[\Sigma_n]_{\pi_n}$;
| // Replace Σ_i by Σ'_i to obtain Φ'.
15 | **return** $\texttt{Prove}_{\mathcal{V}}(\Phi')$
16 **end**
17 **if** *for some* $0 \leq i \leq n$, $(\widetilde{\varphi}^\bullet \wedge \widetilde{\psi}^\circ)^\circ \in \Sigma_i$, *but* $\widetilde{\psi}^\circ \notin \Sigma_i$ **then**
18 | Let $\Sigma'_i := \Sigma_i, \widetilde{\psi}^\circ$;
19 | Let $\Phi' := \Gamma \vdash \Sigma_0, \ldots, (s_i)[\Sigma'_i]_{\pi_i}, \ldots, (s_n)[\Sigma_n]_{\pi_n}$;
| // Replace Σ_i by Σ'_i to obtain Φ'.
20 | **return** $\texttt{Prove}_{\mathcal{V}}(\Phi')$
21 **end**

that proof-search is exponential; therefore, our proof-search algorithm uses the aforementioned labels to only generate a single path in this binary tree relative to each coloring, which yields a worst-case complexity-optimal proof-search procedure in CoNP (for the validity problem of $\mathbb{S}(\mathcal{V})$).

Definition 7 (Coloring)**.** *We define a* colored formula *to be a formula generated via the following grammar in BNF:*

$$\widetilde{\varphi} ::= p^* \mid \neg p^* \mid (\widetilde{\varphi} \vee \widetilde{\varphi})^* \mid (\widetilde{\varphi} \wedge \widetilde{\varphi})^* \mid (\Diamond_s \widetilde{\varphi})^* \mid (\Box_s \widetilde{\varphi})^*$$

with $* \in \{\circ, \bullet\}$*. For any colored formula* $\widetilde{\varphi}$*, we let* φ *be the formula in* $\mathcal{L}_{\mathcal{V}}$ *obtained by removing all labels* $\circ$ *and* $\bullet$ *from* $\widetilde{\varphi}$*. A formula* $\widetilde{\varphi}$ *is* properly colored *iff* $\widetilde{\varphi} = f_\circ(\varphi)$*, where the non-deterministic* coloring function $f_\circ$ *and* $f_\bullet$ *are defined accordingly with* $* \in \{\circ, \bullet\}$*:*

- $f_*(p) = p^*$
- $f_*(\neg p) = \neg p^*$
- $f_*(\varphi \vee \psi) = (f_*(\varphi) \vee f_*(\psi))^*$
- $f_\circ(\varphi \wedge \psi) \in \{(f_\circ(\varphi) \wedge f_\bullet(\psi))^\circ, (f_\bullet(\varphi) \wedge f_\circ(\psi))^\circ\}$
- $f_\bullet(\varphi \wedge \psi) = (f_\bullet(\varphi) \wedge f_\bullet(\psi))^\bullet$
- $f_*(\Diamond_s \varphi) = (\Diamond_s f_*(\varphi))^*$
- $f_*(\Box_s \varphi) = (\Box_s f_*(\varphi))^*$

We define $\mathrm{pcs}(\varphi)$ *to be the set of all proper colorings of* φ*, and define a* colored nested sequent *to be a nested sequent that uses colored formulae as opposed to formulae from* $\mathcal{L}_{\mathcal{V}}$*.*

We now stipulate our *saturation conditions*. When such conditions are unsatisfied during proof-search it signals that certain inference rules still need to be applied bottom-up. Alternatively, once all such conditions are satisfied this signals that proof-search ought to terminate.

Definition 8 (Saturation Conditions)**.** *A colored nested sequent* $\Gamma \vdash \Sigma_0, (s_1)[\Sigma_1]_{\pi_1}, \ldots, (s_n)[\Sigma_1]_{\pi_n}$ *is* saturated *iff for every* $i \in \{0, \ldots, n\}$ *it satisfies the following conditions:*

id *If* $p^\circ \in \Sigma_i$*, then* $\neg p^\circ \notin \Sigma_i$*;*

$\vee$ *if* $(\widetilde{\varphi} \vee \widetilde{\psi})^\circ \in \Sigma_i$*, then* $\widetilde{\varphi}^\circ, \widetilde{\psi}^\circ \in \Sigma_i$*;*

$\wedge$ *if* $(\widetilde{\varphi} \wedge \widetilde{\psi})^\circ \in \Sigma_i$*, then either* $\widetilde{\varphi}^\circ \in \Sigma_i$ *or* $\widetilde{\psi}^\circ \in \Sigma_i$*;*

$\Diamond_s$ *if* $(\Diamond_s \widetilde{\varphi})^\circ \in \Sigma_i$ *and* $s' \preceq^*_\Gamma s$*, then for each* $j \in \{1, \ldots, n\}$ *such that* $s_j = s'$*,* $\widetilde{\varphi}^\circ \in \Sigma_j$*;*

$\Diamond_*$ *if* $(\Diamond_* \widetilde{\varphi})^\circ \in \Sigma_i$*, then* $\widetilde{\varphi}^\circ \in \Sigma_0$*;*

$\Box_s$ *if* $(\Box_s \widetilde{\varphi})^\circ \in \Sigma_i$*, then for some* $j \in \{1, \ldots, n\}$*,* $s_j = s$*, and* $\widetilde{\varphi}^\circ \in \Sigma_j$*;*

n_s *for each* $s \in \mathcal{S}$*, there exists a* $j \in \{1, \ldots, n\}$ *such that* $s_j = s$*.*

Let us comment on the functionality of our (non-deterministic) proof-search algorithm $\texttt{Prove}_{\mathcal{V}}$ (Alg. 1), which takes $\texttt{ProofSearch}_{\mathcal{V}}$ (Alg. 2) as a subroutine. (NB. Alg. 2 is split between this page and the next due to its length.) As mentioned above, given an input $\Gamma \vdash \varphi$, the algorithm $\texttt{Prove}_{\mathcal{V}}$ guesses a proper coloring $\widetilde{\varphi}$ of φ, and then returns the value of $\texttt{ProofSearch}_{\mathcal{V}}(\Gamma \vdash \widetilde{\varphi})$. We note that $\texttt{ProofSearch}_{\mathcal{V}}$ applies the rules from $\mathsf{NS}(\mathcal{V})$ in a bottom-up manner (each corresponding to a recursive call of the algorithm with the exception of (id)). The application of each rule is as follows: (id) corresponds to lines 1–3, $(\vee)$ to lines 7–11, $(\wedge)$ to lines 12–16 and 17–21, that respectively yields the left and right premises of $(\wedge)$, $(\Diamond^1_s)$ and $(\Diamond^2_s)$ to lines 22–25, $(\Diamond_*)$ to lines 26–29, $(\Box_s)$ to lines 30–33, and (n_s) to lines 34–37.

Moreover, $\texttt{ProofSearch}_{\mathcal{V}}$ contrasts with typical proof-search algorithms in that it utilizes the active and inactive labels $\circ$ and $\bullet$ in $\widetilde{\varphi}$ to guide its computation and only constructs *a single thread* of the proof.[2] In other words, if a nested sequent $\Gamma \vdash \varphi$ is derivable in $\mathsf{NS}(\mathcal{V})$, then the sequent has a proof in $\mathsf{NS}(\mathcal{V})$ such that $\texttt{ProofSearch}_{\mathcal{V}}$ generates each thread of the proof relative to each proper coloring of φ; as argued in the lemma below, all such threads may be 'zipped' together to reconstruct a full proof of $\Gamma \vdash \varphi$ in

[2] A *thread* in a proof is defined in the usual fashion as a path of sequents from the conclusion of the proof to an initial sequent (cf. (Takeuti 2013, p. 14)).

$\mathsf{NS}(\mathcal{V})$. In this way, our proof-search algorithm may be used to construct certificates witnessing the validity (by means of a proof in $\mathsf{NS}(\mathcal{V})$) or invalidity (by means of a counter-model) of any input $\Gamma \vdash \varphi$ (see Thm. 3 below for details).

Lemma 2. *Let $\Gamma \vdash \varphi$ be a sequent and $\mathrm{pcs}(\varphi)$ the (finite) set of proper colorings of φ. If $\texttt{ProofSearch}_{\mathcal{V}}(\Gamma \vdash \widetilde{\varphi}) = \texttt{True}$ for all $\widetilde{\varphi} \in \mathrm{pcs}(\varphi)$, then there is a proof of $\Gamma \vdash \varphi$ in $\mathsf{NS}(\mathcal{V})$.*

Proof. Assume that $\texttt{ProofSearch}_{\mathcal{V}}(\Gamma \vdash \widetilde{\varphi}) = \texttt{True}$ for every $\widetilde{\varphi} \in \mathrm{pcs}(\varphi)$, and the following thread of colored nested sequents is generated during its execution:

$$T(\widetilde{\varphi}) := \Gamma \vdash \Delta_0, \ldots, \Gamma \vdash \Delta_h$$

such that $\Delta_0 = \widetilde{\varphi}$ and $\Gamma \vdash \Delta_h$ is an instance of (id) (by lines 1–3). Let $\mathcal{T}$ be the set of all such threads.

For a thread $\Gamma \vdash \Delta_0, \ldots, \Gamma \vdash \Delta_h \in \mathcal{T}$ and $0 \leq k \leq h$, the colored nested sequent $\Gamma \vdash \Delta_k$ is:

- *left conjunctive iff* $\Gamma \vdash \Delta_{k+1}$ is obtained from $\Gamma \vdash \Delta_k$ by applying $(\wedge)$ yielding the left premise (lines 12–16), and
- *right conjunctive iff* $\Gamma \vdash \Delta_{k+1}$ is obtained from $\Gamma \vdash \Delta_k$ by applying $(\wedge)$ yielding the right premise (lines 17–21).

A colored nested sequent is *conjunctive iff* it is left or right conjunctive. We now explain how our threads may be transformed into a proof of $\Gamma \vdash \varphi$ in $\mathsf{NS}(\mathcal{V})$.

We assume w.l.o.g. that the initial segments of all threads up to and including the first conjunctive sequent $\Gamma \vdash \Delta_k$ are identical (i.e. we assume that the subroutine $\texttt{ProofSearch}_{\mathcal{V}}$ executes in a deterministic fashion). Hence, we may form the 'pseudo-derivation' shown below left, by making use of the first k sequents of any given thread, where the rules $(r_1), \ldots, (r_{k-1})$ are determined on the basis of which lines of $\texttt{ProofSearch}_{\mathcal{V}}$ were executed.

$$\dfrac{\dfrac{\Gamma \vdash \Delta_k}{\vdots}\,(r_{k-1})}{\Gamma \vdash \varphi}\,(r_1) \qquad \dfrac{\dfrac{\dfrac{\Gamma \vdash \Delta^l_{k+1} \qquad \Gamma \vdash \Delta^r_{k+1}}{\Gamma \vdash \Delta_k}\,(\wedge)}{\vdots}\,(r_{k-1})}{\Gamma \vdash \varphi}\,(r_1)$$

Let us define $T^k(\widetilde{\varphi}) := \Gamma \vdash \Delta_{k+1}, \ldots, \Gamma \vdash \Delta_h$ to be the tail of a thread $T(\widetilde{\varphi})$ starting from $k+1$. Since the k^{th} colored nested sequent of every thread is conjunctive, we may generate two sets of threads $\mathcal{T}_L$ and $\mathcal{T}_R$ from $\mathcal{T}$:

$$\mathcal{T}_L := \{T^k(\widetilde{\varphi}) \mid \Gamma \vdash \Delta_k \in T(\widetilde{\varphi}) \text{ is left conjunctive}\}$$
$$\mathcal{T}_R := \{T^k(\widetilde{\varphi}) \mid \Gamma \vdash \Delta_k \in T(\widetilde{\varphi}) \text{ is right conjunctive}\}$$

We now extend the 'pseudo-derivation' shown above left, with a bottom-up application of $(\wedge)$ to obtain the 'pseudo-derivation' shown above right, where $\Gamma \vdash \Delta^l_{k+1}$ and $\Gamma \vdash \Delta^r_{k+1}$ are the initial elements of each thread $\mathcal{T}_L$ and $\mathcal{T}_R$, respectively. By successively repeating the above described process over $\mathcal{T}_L$ and $\mathcal{T}_R$, a proof in $\mathsf{NS}(\mathcal{V})$ will eventually be built above $\Gamma \vdash \Delta^l_{k+1}$ and $\Gamma \vdash \Delta^r_{k+1}$, giving a proof of $\Gamma \vdash \varphi$ in $\mathsf{NS}(\mathcal{V})$. □

Example 3. *To illustrate the procedure in Lem. 2 above, we provide an example of the proof construction process of $\emptyset \vdash \varphi$ with $\varphi := \Box_s(p \vee (\neg p \wedge \neg p)) \wedge \Diamond_s(q \vee \neg q)$. First, observe that φ has three proper colorings, given below:*

Algorithm 2: $\texttt{ProofSearch}_{\mathcal{V}}$ (Part II)

22 **if** *for some* $0 \leq i \leq n$, $(\Diamond_s \widetilde{\varphi})^\circ \in \Sigma_i$ *and* $s' \preceq^*_\Gamma s$, *and for some* $1 \leq j \leq n$, $s_j = s'$, *but* $\widetilde{\varphi} \notin \Sigma_j$ **then**
23   Let $\Phi' := \Gamma \vdash \Sigma_0, \ldots, (s_j)[\Sigma_j, \widetilde{\varphi}]_{\pi_j}, \ldots, (s_n)[\Sigma_n]_{\pi_n}$;
   // Add $\widetilde{\varphi}$ to the j^{th} nesting to obtain Φ'.
24   **return** $\texttt{Prove}_{\mathcal{V}}(\Phi')$;
25 **end**
26 **if** *for some* $0 \leq i \leq n$, $(\Diamond_* \widetilde{\varphi})^\circ \in \Sigma_i$, *but* $\widetilde{\varphi} \notin \Sigma_0$ **then**
27   Let $\Phi' := \Gamma \vdash \widetilde{\varphi}, \Sigma_0, (s_1)[\Sigma_1]_{\pi_1}, \ldots, (s_n)[\Sigma_n]_{\pi_n}$;
   // Add $\widetilde{\varphi}$ to the 0^{th} component to obtain Φ'.
28   **return** $\texttt{Prove}_{\mathcal{V}}(\Phi')$;
29 **end**
30 **if** *for some* $0 \leq i \leq n$, $(\Box_s \widetilde{\varphi})^\circ \in \Sigma_i$, *but for each* $1 \leq j \leq n$ *such that* $s_j = s$, $\widetilde{\varphi} \notin \Sigma_j$ **then**
31   Let $\Sigma' := \Gamma \vdash \Sigma_0, \ldots, (s_n)[\Sigma_n]_{\pi_n}, (s)[\widetilde{\varphi}]_{\pi_{n+1}}$;
   // Append $(s)[\widetilde{\varphi}]_{\pi_{n+1}}$ to obtain Φ' with π_{n+1} fresh.
32   **return** $\texttt{Prove}_{\mathcal{V}}(\Phi')$;
33 **end**
34 **if** *for some* $s \in \mathcal{S}$ *there does not exist a* $1 \leq j \leq n$ *such that* $s_j = s$ **then**
35   Let $\Phi' := \Gamma \vdash \Sigma_0, \ldots, (s_n)[\Sigma_n]_{\pi_n}, (s)[\emptyset]_{\pi_{n+1}}$;
   // Append $(s)[\emptyset]_{\pi_{n+1}}$ to obtain Φ' with π_{n+1} fresh.
36   **return** $\texttt{Prove}_{\mathcal{V}}(\Phi')$;
37 **end**

$$\widetilde{\varphi}^1 := ((\Box_s(p^\circ \vee (\neg p^\circ \wedge \neg p^\bullet)^\circ)^\circ)^\circ \wedge (\Diamond_s(q^\bullet \vee \neg q^\bullet)^\bullet)^\bullet)^\circ$$
$$\widetilde{\varphi}^2 := ((\Box_s(p^\circ \vee (\neg p^\bullet \wedge \neg p^\circ)^\circ)^\circ)^\circ \wedge (\Diamond_s(q^\bullet \vee \neg q^\bullet)^\bullet)^\bullet)^\circ$$
$$\widetilde{\varphi}^3 := ((\Box_s(p^\bullet \vee (\neg p^\bullet \wedge \neg p^\bullet)^\bullet)^\bullet)^\bullet \wedge (\Diamond_s(q^\circ \vee \neg q^\circ)^\circ)^\circ)^\circ$$

Each proper coloring $\widetilde{\varphi}^i$ with $i \in \{1, 2, 3\}$ gives rise to a corresponding thread $T(i)$ (shown at the top of Fig. 3) when $\texttt{ProofSearch}_{\mathcal{V}}(\emptyset \vdash \widetilde{\varphi}^i)$ is run. Note that in the figure we have omitted the active and inactive labels, as well as labels of the form π_i, from each thread $T(i)$ to improve readability, and each application of $(\wedge)$ is emphasized with a dashed inference line. By making use of the proof construction process described in Lem. 2, the three threads $T(1)$, $T(2)$, and $T(3)$ can be fused together to generate a proof of $\emptyset \vdash \varphi$ shown at the bottom of Fig. 3.

Theorem 3 (Correctness)**.** *Let $\Gamma \vdash \varphi$ be a sequent.*

1. *If $\texttt{ProofSearch}_{\mathcal{V}}(\Gamma \vdash \widetilde{\varphi}) = \texttt{True}$ for all proper colorings of φ, then a proof in $\mathsf{NS}(\mathcal{V})$ may be constructed witnessing that $\Gamma \vdash \varphi$ is valid.*
2. *If $\texttt{ProofSearch}_{\mathcal{V}}(\Gamma \vdash \widetilde{\varphi}) = \texttt{False}$ for some proper coloring of φ, then a counter-model may be constructed witnessing that $\Gamma \vdash \varphi$ is invalid.*

Proof. The first claim follows by Lem. 2 and the soundness of each nested calculus (see Thm. 1); therefore, we focus on

$$T(1) := \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\emptyset \vdash \varphi, \psi_0, (s)[\psi_1, p, \neg p, \psi_2]}\,(id)}{\emptyset \vdash \varphi, \psi_0, (s)[\psi_1, p, \neg p \wedge \neg p]}\,(\wedge)}{\emptyset \vdash \varphi, \psi_0, (s)[p \vee (\neg p \wedge \neg p)]}\,(\vee)}{\emptyset \vdash \varphi, \Box_s(p \vee (\neg p \wedge \neg p))}\,(\Box_s)}{\emptyset \vdash \varphi}\,(\wedge)$$

$$T(2) := \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\emptyset \vdash \varphi, \psi_0, (s)[\psi_1, p, \neg p, \psi_2]}\,(id)}{\emptyset \vdash \varphi, \psi_0, (s)[\psi_1, p, \neg p \wedge \neg p]}\,(\wedge)}{\emptyset \vdash \varphi, \psi_0, (s)[p \vee (\neg p \wedge \neg p)]}\,(\vee)}{\emptyset \vdash \varphi, \Box_s(p \vee (\neg p \wedge \neg p))}\,(\Box_s)}{\emptyset \vdash \varphi}\,(\wedge)$$

$$T(3) := \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\emptyset \vdash \varphi, \Diamond_s(q \vee \neg q), (s)[\psi_3, q, \neg q]}\,(id)}{\emptyset \vdash \varphi, \Diamond_s(q \vee \neg q), (s)[q \vee \neg q]}\,(\vee)}{\emptyset \vdash \varphi, \Diamond_s(q \vee \neg q), (s)[\emptyset]}\,(\Diamond_s)}{\emptyset \vdash \varphi, \Diamond_s(q \vee \neg q)}\,(n_s)}{\emptyset \vdash \varphi}\,(\wedge)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\emptyset \vdash \varphi, \psi_0, (s)[\psi_1, p, \neg p, \psi_2]}\,(id) \quad \dfrac{}{\emptyset \vdash \varphi, \psi_0, (s)[\psi_1, p, \neg p, \psi_2]}\,(id)}{\emptyset \vdash \varphi, \psi_0, (s)[\psi_1, p, \neg p \wedge \neg p]}\,(\wedge)}{\emptyset \vdash \varphi, \psi_0, (s)[p \vee (\neg p \wedge \neg p)]}\,(\vee)}{\emptyset \vdash \varphi, \Box_s(p \vee (\neg p \wedge \neg p))}\,(\Box_s) \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\emptyset \vdash \varphi, \Diamond_s(q \vee \neg q), (s)[\psi_3, q, \neg q]}\,(id)}{\emptyset \vdash \varphi, \Diamond_s(q \vee \neg q), (s)[q \vee \neg q]}\,(\vee)}{\emptyset \vdash \varphi, \Diamond_s(q \vee \neg q), (s)[\emptyset]}\,(\Diamond_s)}{\emptyset \vdash \varphi, \Diamond_s(q \vee \neg q)}\,(n_s)}{\emptyset \vdash \varphi}\,(\wedge)$$

Figure 3: An example of how the threads $T(1)$, $T(2)$, and $T(3)$ may be 'zipped' together to construct the derivation shown above bottom. Note that $\varphi := \Box_s(p \vee (\neg p \wedge \neg p)) \wedge \Diamond_s(q \vee \neg q)$, $\psi_0 := \Box_s(p \vee (\neg p \wedge \neg p))$, $\psi_1 := p \vee (\neg p \wedge \neg p)$, $\psi_2 := \neg p \wedge \neg p$, and $\psi_3 := q \vee \neg q$.

the second claim. Suppose that $\texttt{ProofSearch}_{\mathcal{V}}(\Gamma \vdash \widetilde{\varphi}) = \texttt{False}$ for some chosen proper coloring of φ. Then, $\texttt{ProofSearch}_{\mathcal{V}}$ generates a saturated nested sequent $\Gamma \vdash \Delta$ with Δ of the form $\Sigma_0, (s_1)[\Sigma_1]_{\pi_1}, \ldots, (s_n)[\Sigma_n]_{\pi_n}$. We will use $\Gamma \vdash \Delta$ to construct a counter-model for $\Gamma \vdash \varphi$, thus proving the second claim. Let us define $\mathcal{M} := \langle \Pi, \sigma, \delta \rangle$ as:

- $\Pi := \{\pi_0, \pi_1, \ldots, \pi_n\}$;
- $\sigma(*) := \Pi$ and for each $s \in \mathcal{S} \setminus \{*\}$, we define $\sigma(s)$:
  (i) $\pi_i \in \sigma(s)$ for each $i \in \{1, \ldots, n\}$ such that $s_i = s$, and
  (ii) $\pi_j \in \sigma(s)$ for each $\pi_j \in \sigma(s')$ such that $s' \preceq^*_\Gamma s$;
- $\delta(p) := \{\pi_i \mid p^\circ \notin \Sigma_i\}$.

We know that the recursive definition of σ will eventually terminate since $\mathcal{S}$ is finite.

Let us now prove that $\mathcal{M}$ is indeed a standpoint model; afterward, we will show that $\mathcal{M}, \pi_0 \not\models \iota(\Gamma \vdash \Delta)$. First, observe that the inclusion of π_0 in Π ensures that the domain of precisifications is non-empty. Furthermore, δ is a function from $\mathcal{P}$ to 2^Π, for each $s \in \mathcal{S}$, we have that $\sigma(s) \neq \emptyset$ by the n_s saturation condition, and $\sigma(*) = \Pi$ by definition. We now show that (a) for each $s' \preceq s \in \Gamma$ and $i \in \{0, \ldots, n\}$, $\mathcal{M}, \pi_i \models s' \preceq s$, and (b) for each $i \in \{0, \ldots, n\}$ and $\widetilde{\psi} \in \Sigma_i$, $\mathcal{M}, \pi_i \not\models \psi$, from which $\mathcal{M}, \pi_0 \not\models \iota(\Gamma \vdash \varphi)$ follows as $\widetilde{\varphi} \in \Sigma_0$.

(a) Let $s' \preceq s \in \Gamma$ and assume that $\pi \in \sigma(s')$. Then, by the second clause in the definition of $\sigma(s)$ it follows that $\pi \in \sigma(s)$, thus showing that $\mathcal{M}, \pi \models s \preceq s'$ for all $\pi \in \Pi$.

(b) By induction on the complexity of ψ.

Base case. First, let us suppose that ψ is a propositional atom p where $p^\circ \in \Sigma_i$. Then, by the definition of δ, we have that $\pi_i \notin \delta(p)$, i.e. $\mathcal{M}, \pi_i \not\models p$. Similarly, if ψ is a negated atom $\neg p$ where $\neg p^\circ \in \Sigma_i$, then by the id saturation condition, we know that $p^\circ \notin \Sigma_i$, implying that $\pi_i \in \delta(p)$ by the definition of δ, thus showing that $\mathcal{M}, \pi_i \not\models \neg p$.

Inductive step. We suppose that $\widetilde{\psi}^\circ \in \Sigma_i$ and we show that $\mathcal{M}, \pi_i \not\models \psi$ making a case distinction based on the main connective of ψ.

$\psi = \chi \vee \xi$: By the $\vee$ saturation condition, $\widetilde{\chi}^\circ, \widetilde{\xi}^\circ \in \Sigma_i$. By IH, $\mathcal{M}, \pi_i \not\models \chi$ and $\mathcal{M}, \pi_i \not\models \xi$, showing $\mathcal{M}, \pi_i \not\models \psi$.

$\psi = \chi \wedge \xi$: By the $\wedge$ saturation condition, $\widetilde{\chi}^\circ \in \Sigma_i$ or $\widetilde{\xi}^\circ \in \Sigma_i$. By IH, $\mathcal{M}, \pi_i \not\models \chi$ or $\mathcal{M}, \pi_i \not\models \xi$, so $\mathcal{M}, \pi_i \not\models \psi$.

$\psi = \Diamond_s \chi$ with $s \in \mathcal{S} \setminus \{*\}$: Assume that $\pi_j \in \sigma(s)$. By the definition of $\sigma(s)$, π_j was added via the first (i) or second (ii) condition. If (i), then there is a nesting $(s)\{\Sigma_j\}_{\pi_j}$ in Δ, and by the $\Diamond_s$ saturation condition, $\widetilde{\chi}^\circ \in \Sigma_j$. By IH then, $\mathcal{M}, \pi_j \not\models \chi$. If (ii), then there is a chain of statements $s_0 \preceq s_1, \ldots, s_n \preceq s \in \Gamma$ such that a nesting of the form $(s_0)\{\Sigma_j\}_{\pi_j}$ exists in Δ. Again, by the $\Diamond_s$ saturation condition, it follows that $\widetilde{\chi}^\circ \in \Sigma_j$, which implies that $\mathcal{M}, \pi_j \not\models \chi$ by IH. Hence, $\mathcal{M}, \pi_i \not\models \Diamond_s \chi$.

$\psi = \Diamond_* \chi$: Since $*$ is maximal relative to the $\preceq^*_\Gamma$ relation, i.e. $s \preceq^*_\Gamma *$ for all $s \in \mathcal{S}$, we have $\widetilde{\chi}^\circ \in \Sigma_j$ for each $j \in \{0, \ldots, n\}$ by the $\Diamond_*$ and $\Diamond_s$ saturation conditions, hence for all $\pi_j \in \Pi$, $\mathcal{M}, \pi_j \not\models \chi$, i.e. $\mathcal{M}, \pi_i \not\models \Diamond_* \chi$.

$\psi = \Box_s \chi$: By the $\Box_s$ saturation condition, we know that there exists a $j \in \{1, \ldots, n\}$ such that $s_j = s$ and $\widetilde{\chi}^\circ \in \Sigma_j$. By IH, $\mathcal{M}, \pi_j \not\models \chi$, thus proving the case. □

Example 4. *Let us provide an example of the counter-model construction procedure given in Thm. 3. We assume that the (invalid) sequent $s \preceq s' \vdash \varphi$ with $\varphi := \Box_{s'} p \vee \Diamond_s \neg p$ is input into $\texttt{Prove}_{\mathcal{V}}$. Since φ has one proper coloring (with all subformulae active), only the following single thread is generated, yielding the saturated sequent shown at the top of the proof below. We omit the active labels for readability.*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{s \preceq s' \vdash \varphi, \Box_{s'} p, \Diamond_s \neg p, (s')[p]_{\pi_1}, (s)[\neg p]_{\pi_2}, (*)[\emptyset]_{\pi_3}}{s \preceq s' \vdash \varphi, \Box_{s'} p, \Diamond_s \neg p, (s')[p]_{\pi_1}, (s)[\neg p]_{\pi_2}}\,(n_*)}{s \preceq s' \vdash \varphi, \Box_{s'} p, \Diamond_s \neg p, (s')[p]_{\pi_1}, (s)[\emptyset]_{\pi_2}}\,(\Diamond^1_s)}{s \preceq s' \vdash \varphi, \Box_{s'} p, \Diamond_s \neg p, (s')[p]_{\pi_1}}\,(n_s)}{s \preceq s' \vdash \varphi, \Box_{s'} p, \Diamond_s \neg p}\,(\Box_{s'})}{s \preceq s' \vdash \varphi}\,(\vee)$$

Then, we may extract the following (counter-)model $\mathcal{M} = \langle \Pi, \sigma, \delta \rangle$ from the top, saturated sequent in the proof above.

- $\Pi := \{\pi_0, \pi_1, \pi_2, \pi_3\}$;
- $\sigma(*) := \Pi$, $\sigma(s) := \{\pi_2\}$, *and* $\sigma(s') := \{\pi_1\}$;
- $\delta(p) := \{\pi_0, \pi_2, \pi_3\}$.

It is readily verifiable that $\mathcal{M}, \pi_0 \not\models \Box_{s'} p \vee \Diamond_s \neg p$.

We now show that $\texttt{ProofSearch}_{\mathcal{V}}$ (and hence $\texttt{Prove}_{\mathcal{V}}$) terminates after at most polynomially many rule applications in the *size* of the input sequent. For an input $\Phi := \Gamma \vdash \varphi$, its size is defined to be $|\Phi| := |\mathcal{S}| + |\varphi|$. That is, the size of Φ is the sum of the cardinality of the set $\mathcal{S}$ of standpoints and the size of φ. The size of a sequent incorporates a measure on the set $\mathcal{S}$ from the associated vocabulary $\mathcal{V}$ as opposed to a measure on the set Γ of sharpening statements because Γ only plays a role in bottom-up applications of $(\Diamond_s^1)$ and $(\Diamond_s^2)$, which are bounded in part by the cardinality of $\mathcal{S}$ and in part by the number of $\Box_s$ modalities occurring in φ, as explained in the proof of Thm. 4 below.

Theorem 4 (Termination). *Let $\Phi := \Gamma \vdash \varphi$ be a sequent. Then, the number of recursive calls in $\texttt{ProofSearch}_{\mathcal{V}}(\Gamma \vdash \widetilde{\varphi})$, and thus $\texttt{Prove}_{\mathcal{V}}(\Gamma \vdash \varphi)$, is bounded by a polynomial*

$$p(|\Phi|) = \mathcal{O}(|\Phi|^2).$$

Proof. Let $\Phi := \Gamma \vdash \varphi$ be a nested sequent, and $N_{\oplus}$ be the number of occurrences of the connectives $\{\vee, \wedge\} \cup \{\Diamond_s \mid s \in \mathcal{S}\}$ in φ. By the saturation conditions (Def. 8), we know that for each $s \in \mathcal{S}$, the $(\Box_s)$ rule will be applied bottom-up at most one time for each occurrence of $\Box_s$ in φ, which are bounded by $|\varphi|$. Also, (n_s) will be applied at most once for each $s \in \mathcal{S}$. Since only $(\Box_s)$ and (n_s) introduce nestings, the number of components (i.e. the nestings plus the root) throughout the course of proof-search is bounded by:

$$K := 1 + |\mathcal{S}| + |\varphi|$$

For each occurrence of $\vee$, $\wedge$, and $\Diamond_s$ in φ (with $s \in \mathcal{S}$), we know by the saturation conditions that $(\vee)$, $(\wedge)$, $(\Diamond_s^1)$, $(\Diamond_s^2)$, and $(\Diamond_*)$ can be applied a maximum number of K times during proof search. Then, since $N_{\vee} + N_{\wedge} + \sum_{s \in \mathcal{S}} N_{\Diamond_s} \leq |\varphi|$, the number of recursive calls (i.e. bottom-up applications of rules) during proof-search is bounded by $N := |\varphi| \cdot K$. Finally, $|\mathcal{S}|, |\varphi| \leq |\Phi|$ holds trivially, implying:

$$N \leq |\Phi| \cdot (1 + |\Phi| + |\Phi|)$$

Therefore, it follows that a polynomial $p(|\Phi|) = \mathcal{O}(|\Phi|^2)$ bounds the number of recursive calls of $\texttt{Prove}_{\mathcal{V}}(\Phi)$. $\Box$

Corollary 1. *Let $\mathcal{V}$ be a vocabulary. Then,*

1. *$\mathbb{S}(\mathcal{V})$ is decidable;*
2. *$\mathbb{S}(\mathcal{V})$ has the finite model property;*
3. *$\texttt{Prove}_{\mathcal{V}}$ is worst-case complexity-optimal, deciding the validity problem for $\mathbb{S}(\mathcal{V})$ in* CoNP*;*
4. *The validity problem for $\mathbb{S}(\mathcal{V})$ is* CoNP*-complete.*

Proof. Statements 1 and 2 follow from the fact that $\texttt{Prove}_{\mathcal{V}}$ is a correct (Thm. 3) and terminating (Thm. 4) decision procedure for $\mathbb{S}(\mathcal{V})$ that, in particular, returns a finite counter-model when the input is invalid.

To show statement 3, observe that $\texttt{Prove}_{\mathcal{V}}$ is a non-deterministic algorithm that takes a sequent $\Phi := \Gamma \vdash \varphi$ as input, guesses a proper coloring of φ, and constructs a thread. Each such thread is polynomial in the size of its input, since the number of rule applications (i.e. the length of the thread) is bounded by a polynomial $p(|\Phi|) = \mathcal{O}(|\Phi|^2)$, by Thm. 4. Moreover, since any sequent generated during proof-search can have at most $K \leq 1 + |\mathcal{S}| + |\varphi|$ many components (as stated in the proof of Thm. 4), each of which can only be inhabited by at most $|\text{sufo}(\varphi)| = |\varphi|$ many formulae, it follows that the size of each nested sequent in the thread is bounded by $\mathcal{O}(|\Phi|^2)$ since $|\mathcal{S}|, |\varphi| \leq |\Phi|$. Taking the functionality of $\texttt{Prove}_{\mathcal{V}}$ into account, one can see that if $\texttt{Prove}_{\mathcal{V}}(\Phi) = \texttt{False}$, then the corresponding thread is generated in polynomial time and its size is bounded above by a polynomial $q(|\Phi|) = \mathcal{O}(|\Phi|^4)$. Additionally, note that $\texttt{Prove}_{\mathcal{V}}$ is worst-case complexity-optimal as the validity problem for classical propositional logic is CoNP-complete, and can be solved by $\texttt{Prove}_{\mathcal{V}}$ as (id), $(\vee)$, and $(\wedge)$ form a sound and complete proof system for propositional logic (cf. (Lyon 2021a, App. B)). Last, statement 4 is an immediate consequence of statement 3. $\Box$

5 Conclusion and Future Work

In this paper, we introduced and employed nested sequent systems to automate reasoning with propositional standpoint logics. To obtain worst-case complexity-optimal proof-search, we presented a novel proof-search technique, referred to as *coloring*, whereby the subformulae of an input formula are non-deterministically colored with (in)active labels, yielding partial proofs (i.e. *threads*) of the input. By means of our technique, we designed a non-deterministic proof-search algorithm deciding the validity of standpoint implications in CoNP, showing how certain threads could be transformed into a counter-model for an invalid input, and how all threads could be transformed into a proof for a valid input. The attainment of these "certificates" from proof-search serve as explanations for the (in)validity of standpoint formulae, thus motivating our proof-theoretic approach.

For future work, we aim to extend our nested systems and proof-search algorithm to cover (i) first-order standpoint logics that (ii) incorporate complex standpoints, which have interesting applications in knowledge integration scenarios. Regarding point (i), placing standpoint logic on a first-order base increases the applicability of the framework along with its expressivity to better match that of contemporary knowledge representation languages. Our focus in this area is to provide results that can then be extrapolated to widely used decidable fragments of FOL. Regarding point (ii), we note that the set-theoretic interpretation of standpoints permits the definition of complex standpoints built atop atomic ones; e.g. union $s_1 \cup s_2$ (integrating knowledge from multiple perspectives), intersection $s_1 \cap s_2$ (expressing the knowledge jointly shared between multiple perspectives), and difference $s_1 \setminus s_2$ (yielding the sharpening of s_1 by ignoring all precisifications of s_2). Beyond providing nested systems for more expressive formulations of standpoint logic, we also aim to write and evaluate theorem provers based on our nested calculi.

Acknowledgments. Lucía Gómez Álvarez was supported by the Bundesministerium fur Bildung und Forschung (BMBF, Federal Ministry of Education and Research) in the Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI). Tim S. Lyon has received funding from the European Research Council (Grant Agreement no. 771779, DeciGUT).

References

Belnap, N. D. 1982. Display logic. *Journal of philosophical logic* 11(4):375–417.

Brünnler, K. 2009. Deep sequent systems for modal logic. *Arch. Math. Log.* 48(6):551–577.

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

Buss, S. R. 1998. An introduction to proof theory. *Handbook of proof theory* 137:1–78.

Dyckhoff, R. 1992. Contraction-free sequent calculi for intuitionistic logic. *The Journal of Symbolic Logic* 57(3):795–807.

Fine, K. 1975. Vagueness, truth and logic. *Synthese* 30(3-4):265–300.

Fitting, M., and Kuznets, R. 2015. Modal interpolation via nested sequents. *Annals of pure and applied logic* 166(3):274–305.

Gentzen, G. 1935a. Untersuchungen über das logische Schließen. i. *Mathematische Zeitschrift* 39(1):176–210.

Gentzen, G. 1935b. Untersuchungen über das logische Schließen. ii. *Mathematische Zeitschrift* 39(1):405–431.

Gómez Álvarez, L., and Bennett, B. 2018. Dealing with conceptual indeterminacy: A framework based on supervaluation semantics. In *Joint Proceedings of MedRACER and WOMoCoE*, volume 2237, 38–50. CEUR.

Gómez Álvarez, L., and Rudolph, S. 2021. Standpoint logic: Multi-perspective knowledge representation. In Neuhaus, F., and Brodaric, B., eds., *Proceedings of the 12th International Conference on Formal Ontology in Information Systems*, volume 344 of *FAIA*, 3–17. IOS Press.

Gómez Álvarez, L.; Bennett, B.; and Richard-Bollans, A. 2017. Talking about forests: An example of sharing information expressed with vague terms. In *Proceedings of the Joint Ontology Workshops 2017*, volume 2050. CEUR.

Gómez Álvarez, L. 2020. *Standpoint logic: a logic for handling semantic variability, with applications to forestry information*. Ph.D. Dissertation, School of Computing, The University of Leeds.

Kashima, R. 1994. Cut-free sequent calculi for some tense logics. *Studia Logica* 53(1):119–135.

Keefe, R., and Smith, P. 1997. *Theories of vagueness*. The MIT press.

Lyon, T., and van Berkel, K. 2019. Automating agential reasoning: Proof-calculi and syntactic decidability for STIT logics. In Baldoni, M.; Dastani, M.; Liao, B.; Sakurai, Y.; and Zalila Wenkstern, R., eds., *Proceedings of the 22nd International Conference on Principles and Practice of Multi-Agent Systems*, volume 11873, 202–218. Springer.

Lyon, T.; Tiu, A.; Goré, R.; and Clouston, R. 2020. Syntactic interpolation for tense logics and bi-intuitionistic logic via nested sequents. In Fernández, M., and Muscholl, A., eds., *Proceedings of the 28th Annual Conference on Computer Science Logic*, volume 152 of *LIPIcs*, 28:1–28:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Lyon, T. 2020. On the correspondence between nested calculi and semantic systems for intuitionistic logics. *Journal of Logic and Computation* 31(1):213–265.

Lyon, T. 2021a. *Refining Labelled Systems for Modal and Constructive Logics with Applications*. Ph.D. Dissertation, Technische Universität Wien.

Lyon, T. S. 2021b. Nested sequents for intuitionistic modal logics via structural refinement. In Das, A., and Negri, S., eds., *Automated Reasoning with Analytic Tableaux and Related Methods*, 409–427. Cham: Springer.

Pesquita, C.; Faria, D.; Santos, E.; and Couto, F. M. 2013. To repair or not to repair: Reconciling correctness and coherence in ontology reference alignments. In *Proceedings of the 8th International Conference on Ontology Matching*, volume 1111, 13–24. CEUR.

Poggiolesi, F. 2009. The method of tree-hypersequents for modal propositional logic. In Makinson, D.; Malinowski, J.; and Wansing, H., eds., *Towards Mathematical Philosophy*, volume 28 of *Trends in logic*. Springer. 31–51.

Simpson, A. K. 1994. *The proof theory and semantics of intuitionistic modal logic*. Ph.D. Dissertation, University of Edinburgh. College of Science and Engineering. School of Informatics.

Slaney, J. K. 1997. Minlog: A minimal logic theorem prover. In McCune, W., ed., *Proceedings of the 14th International Conference on Automated Deduction*, volume 1249 of *Lecture Notes in Computer Science*, 268–271. Springer.

Takeuti, G. 2013. *Proof theory*, volume 81. Courier Corporation.

Tiu, A.; Ianovski, E.; and Goré, R. 2012. Grammar logics in nested sequent calculus: Proof theory and decision procedures. In Bolander, T.; Braüner, T.; Ghilardi, S.; and Moss, L. S., eds., *Advances in Modal Logic 9, papers from the ninth conference on Advances in Modal Logic*, 516–537. College Publications.

Wansing, H. 2002. Sequent systems for modal logics. In Gabbay, D. M., and Guenthner, F., eds., *Handbook of Philosophical Logic: Volume 8*. Springer. 61–145.