Automating Agential Reasoning: Proof-Calculi and Syntactic Decidability for STIT Logics*

Tim Lyon(✉) and Kees van Berkel

Institut für Logic and Computation, Technische Universität Wien, 1040 Wien, Austria
{lyon,kees}@logic.at

Abstract. This work provides proof-search algorithms and automated counter-model extraction for a class of STIT logics. With this, we answer an open problem concerning syntactic decision procedures and cut-free calculi for STIT logics. A new class of cut-free complete labelled sequent calculi $\mathsf{G3Ldm}^m_n$, for multi-agent STIT with at most n-many choices, is introduced. We refine the calculi $\mathsf{G3Ldm}^m_n$ through the use of propagation rules and demonstrate the admissibility of their structural rules, resulting in the auxiliary calculi $\mathsf{Ldm}^m_n\mathsf{L}$. In the single-agent case, we show that the refined calculi $\mathsf{Ldm}^m_n\mathsf{L}$ derive theorems within a restricted class of (forestlike) sequents, allowing us to provide proof-search algorithms that decide single-agent STIT logics. We prove that the proof-search algorithms are correct and terminate.

Keywords: Decidability · Labelled calculus · Logics of agency · Proof search · Proof theory · Propagation rules · Sequent · STIT logic

1 Introduction

Modal logics of STIT, an acronym for 'seeing to it that', have a long tradition in the formal investigation of agency, starting with a series of papers by Belnap and Perloff in the 1980s and culminating in [3]. For the past decades, STIT logic has continued to receive considerable attention, proving itself invaluable in a multitude of fields concerned with formal reasoning about agentive choice making. For example, the framework has been applied to epistemic logic [5], deontic logic [11,13], and the formal analysis of legal reasoning [5,12]. Surprisingly, investigations of the mathematical properties of STIT logics are limited [2,15] and its proof-theory has only been addressed recently [4,19]. What is more, despite AI-oriented STIT papers motivating the need of tools for automated reasoning about agentive choice-making [1,2,4], the envisaged automation results are still lacking. The present work will be the first to provide terminating, automated proof-search for a class of STIT logics, including counter-model extraction directly based on failed proof-search.

* This is a pre-print of an article published in Lecture Notes in Artificial Intelligence. The final authenticated version is available online at: https://doi.org/10.1007/978-3-030-33792-6_13. Work funded by the projects WWTF MA16-028, FWF I2982 and FWF W1255-N23.

The *sequent calculus* [7] is an effective framework for proof-search, suitable for automated deduction procedures. Given the metalogical property of *analyticity*, a sequent calculus allows for the construction of proofs by merely decomposing the formula in question. In the present work, we employ the *labelled* sequent calculus—a useful formalism for a large class of modal logics [14,18]—and introduce labelled sequent calculi $\mathsf{G3Ldm}_n^m$ (with $n, m \in \mathbb{N}$) for multi-agent STIT logics containing limited choice axioms, discussed in [20].

In order to appropriate the calculi $\mathsf{G3Ldm}_n^m$ for automated proof-search, we take up a *refinement* method presented in [17]—developed for the more restricted setting of display logic—and adapt it to the more general setting of labelled calculi. In the refinement process the *external* character of labelled systems—namely, the explicit presence of the semantic structure—is made *internal* through the use of alternative, yet equivalent, *propagation rules* [17]. The tailored propagation rules restrict and simplify the sequential structures needed in derivations, producing, for example, shorter proofs. Moreover, one can show that through the use of propagation rules, the structural rules, capturing the behavior of the logic's modal operators, are admissible. In our case, the resulting refined calculi $\mathsf{Ldm}_n^m\mathsf{L}$ derive theorems using only forestlike sequents, allowing us to adapt methods from [17] and provide correct and terminating proof-search algorithms for this class of STIT logics.

In short, the contribution of this paper will be threefold: First, we provide new sound and cut-free complete labelled sequent calculi $\mathsf{G3Ldm}_n^m$ for all multi-agent STIT logics Ldm_n^m (with $n, m \in \mathbb{N}$) discussed in [20]—thus extending the class of logics addressed in [4]. Second, we show how to refine these calculi to obtain new calculi $\mathsf{Ldm}_n^m\mathsf{L}$, which are suitable for proof-search. Last, for each $n \in \mathbb{N}$, we provide a terminating proof-search algorithm deciding the single-agent STIT logic Ldm_n^1. Although [9] provides a polynomial reduction of Ldm_n^m into the modal logic $\mathsf{S5}$ (providing decidability via $\mathsf{S5}$-SAT), the present work has the advantage that it offers a syntactic decision procedure within the unreduced Ldm_n^m language and is modular, that is, it will allow us to extend our work to a variety of STIT logics. We conclude by discussing the prospects of generalizing the latter results to the multi-agent setting.

The paper is structured as follows: We start by introducing the class of logics Ldm_n^m in Sec. 2. In Sec. 3, corresponding labelled calculi $\mathsf{G3Ldm}_n^m$ are provided, which will subsequently be refined, resulting in the calculi $\mathsf{Ldm}_n^m\mathsf{L}$. We devote Sec. 4 to proof-search algorithms and counter-model extraction.

2 Logical Preliminaries

STIT logic refers to a group of modal logics using operators that capture agential choice-making. The STIT logics Ldm_n^m, which will be considered throughout this paper, employ two types of modal operators: First, they contain a *settledness* operator $\Box$ expressing which formulae are 'settled true' at a current moment. Second, they contain, for each agent i in the language, an atemporal—i.e., instantaneous—*choice* operator $[i]$ expressing that 'agent i sees to it that'. This basic choice operator is referred to as the *Chellas* STIT [3]. Using both operators, one can define the more refined notion of *deliberative* STIT: i.e., $[i]_d\phi$ iff $[i]\phi \wedge \neg\Box\phi$. Intuitively, $[i]_d\phi$ holds when 'agent i sees to it that ϕ and it is possible for ϕ not to hold'. The multi-agent language for Ldm_n^m is defined accordingly:

Definition 1 (The Language $\mathcal{L}^m$ [10]). *Let $Ag = \{1, 2, ..., m\}$ be a finite set of agent labels and let $Var = \{p, q, r...\}$ be a countable set of propositional variables. $\mathcal{L}^m$ is defined via the following BNF grammar:*

$$\phi ::= p \mid \overline{p} \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\Box\phi) \mid (\Diamond\phi) \mid ([i]\phi) \mid (\langle i\rangle\phi)$$

where $i \in Ag$ and $p \in Var$.

Notice, the language $\mathcal{L}^m$ consists of formulae in negation normal form. This notation allows us to reduce the number of rules in our calculi, enhancing the readability and simplicity of our proof theory. The negation of ϕ, written as $\overline{\phi}$, is obtained by replacing each operator with its dual, each positive atom p with its negation $\overline{p}$, and each $\overline{p}$ with its positive variant p [4]. Consequently, we obtain the following abbreviations: $\phi \rightarrow \psi$ *iff* $\overline{\phi} \vee \psi$, $\phi \leftrightarrow \psi$ *iff* $(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$, $\top$ *iff* $p \vee \overline{p}$, and $\bot$ *iff* $p \wedge \overline{p}$. We will freely use these abbreviations throughout this paper. Since we are working in negation normal form, diamond-modalities are introduced as separate primitive operators. We take $\langle i\rangle$ and $\Diamond$ as the duals of $[i]$ and $\Box$, respectively.

Following [10], since we work with instantaneous, atemporal STIT it suffices to regard only single choice-moments in our relational frames. This means that we can forgo the traditional branching time structures of basic, atemporal STIT logic [3]. In what follows, we define Ldm_n^m frames as those STIT frames in which $n > 0$ limits the amount of choices available to each agent to at most n-many choices (imposing no limitation when $n = 0$).[1]

Definition 2 (Relational Ldm_n^m Frames and Models). *Let $|Ag| = m$ and let $\mathcal{R}_i(w) := \{v \in W \mid (w, v) \in \mathcal{R}_i\}$ for $i \in Ag$. An Ldm_n^m-*frame *is defined as a tuple $F = (W, \{\mathcal{R}_i \mid i \in Ag\})$ where $W \neq \emptyset$ is a set of worlds $w, v, u...$ and:*

(C1) *For each $i \in Ag$, $\mathcal{R}_i \subseteq W \times W$ is an equivalence relation;*
(C2) *For all $u_1, ..., u_m \in W$, $\bigcap_i \mathcal{R}_i(u_i) \neq \emptyset$;*
(C3) *Let $n > 0$ and $i \in Ag$, then*

$$\text{For all } w_0, w_1, \cdots, w_n \in W, \bigvee_{0 \leq k \leq n-1,\ k+1 \leq j \leq n} \mathcal{R}_i w_k w_j$$

*An Ldm_n^m-model is a tuple $M = (F, V)$ where F is an Ldm_n^m-frame and V is a valuation assigning propositional variables to subsets of W, i.e. $V \colon Var \mapsto \mathcal{P}(W)$. Additionally, we stipulate that condition **(C3)** is omitted when $n = 0$.*

As in [10], the set of worlds W is taken to represent a single moment in which agents from Ag are making their decision. Following **(C1)**, for every agent i, the

[1] For a discussion of the philosophical utility of reasoning with limited choice see [20].

relation $\mathcal{R}_i$ is an equivalence relation; that is, $\mathcal{R}_i$ functions as a partitioning of W into what will be called *choice-cells* for agent i. Each choice-cell represents a set of possible worlds that may be realized by a choice of the agent. The condition **(C2)** expresses the STIT principle *independence of agents*, ensuring that any combination of choices, available to different agents, is consistent. The last condition **(C3)**, represents the STIT principle which limits the amount of choices available to an agent to a maximum of n. For a philosophical discussion of these principles we refer to [3, Ch. 7C].

Definition 3 (Semantic Clauses for $\mathcal{L}^m$ [4,10]). *Let M be an Ldm_n^m-model $(W, \{\mathcal{R}_i \mid i \in Ag\}, V)$ and let w be a world in its domain W. The* satisfaction *of a formula $\phi \in \mathcal{L}^m$ on M at w is inductively defined as follows:*

1. $M, w \Vdash p$ *iff* $w \in V(p)$
2. $M, w \Vdash \overline{p}$ *iff* $w \notin V(p)$
3. $M, w \Vdash \phi \wedge \psi$ *iff* $M, w \Vdash \phi$ *and* $M, w \Vdash \psi$
4. $M, w \Vdash \phi \vee \psi$ *iff* $M, w \Vdash \phi$ *or* $M, w \Vdash \psi$
5. $M, w \Vdash \Box\phi$ *iff* $\forall u \in W$, $M, u \Vdash \phi$
6. $M, w \Vdash \Diamond\phi$ *iff* $\exists u \in W$, $M, u \Vdash \phi$
7. $M, w \Vdash [i]\phi$ *iff* $\forall u \in \mathcal{R}_i(w)$, $M, u \Vdash \phi$
8. $M, w \Vdash \langle i \rangle\phi$ *iff* $\exists u \in \mathcal{R}_i(w)$, $M, u \Vdash \phi$

A formula ϕ is globally true *on M (i.e. $M \Vdash \phi$) iff it is satisfied at every world w in the domain W of M. A formula ϕ is* valid *(i.e. $\Vdash \phi$) iff it is globally true on every Ldm_n^m-model. Last, Γ semantically implies ϕ, written $\Gamma \Vdash \phi$, iff for all models M and worlds w of W in M, if $M, w \Vdash \psi$ for all $\psi \in \Gamma$, then $M, w \Vdash \phi$.*

It is worth emphasizing that the semantic interpretation of $\Box$ refers to the domain of the model in its entirety; i.e., ϕ is settled true iff ϕ is globally true. This is an immediate consequence of considering instantaneous STIT in a single-moment setting (cf. semantics where a relation $\mathcal{R}_\Box$ is introduced for $\Box$, e.g., [4]).

The Hilbert calculus for Ldm_n^m in Fig. 1 is taken from [20]. Apart from the propositional axioms, it consists of S5 axiomatizations for $\Box$ and $[i]$, for each $i \in Ag$. It contains the standard bridge axiom (Bridge), linking $[i]$ to $\Box$. Furthermore, it contains an independence of agents axiom (IOA), as well as an n-choice axiom (APC_n^i) for each $i \in Ag$. The rules are modus ponens and $\Box$-necessitation.

Theorem 1 (Soundness and Completeness [10,20]). *For any formula $\phi \in \mathcal{L}^m$, $\Gamma \vdash_{\mathsf{Ldm}_n^m} \phi$ if and only if $\Gamma \Vdash \phi$.*

3 Refinement of the Calculi $\mathsf{G3Ldm}_n^m$

In this section, we introduce the labelled calculi $\mathsf{G3Ldm}_n^m$ for multi-agent STIT logics (with limited choice). Our calculi are modified, extended versions of the labelled calculi for the logics Ldm_0^m (with $m \in \mathbb{N}$) proposed in [4] and cover a larger class of logics. The calculi $\mathsf{G3Ldm}_n^m$ possess fundamental proof-theoretic properties such as contraction- and cut-admissibility which follow from the general results on labelled calculi established in [14]. The main goal of this section is to refine the $\mathsf{G3Ldm}_n^m$ calculi through the elimination of structural rules, resulting in new calculi $\mathsf{Ldm}_n^m\mathsf{L}$ that derive theorems within a restricted class of

$$\phi \to (\psi \to \phi) \quad (\overline{\psi} \to \overline{\phi}) \to (\phi \to \psi) \quad (\phi \to (\psi \to \chi)) \to ((\phi \to \psi) \to (\phi \to \chi))$$

$$(\mathsf{S5}\Box)\ \Box(\phi \to \psi) \to (\Box\phi \to \Box\psi) \quad \Box\phi \to \phi \quad \Diamond\phi \to \Box\Diamond\phi \quad \Box\phi \lor \Diamond\overline{\phi}$$

$$(\mathsf{S5}[i])\ [i](\phi \to \psi) \to ([i]\phi \to [i]\psi) \quad [i]\phi \to \phi \quad \langle i\rangle\phi \to [i]\langle i\rangle\phi \quad [i]\phi \lor \langle i\rangle\overline{\phi}$$

$$(\mathsf{IOA})\ \bigwedge_{i \in Ag} \Diamond[i]\phi_i \to \Diamond(\bigwedge_{i \in Ag}[i]\phi_i) \quad (\mathsf{Bridge})\ \Box\phi \to [i]\phi \quad \frac{\phi}{\Box\phi} \quad \frac{\phi \qquad \phi \to \psi}{\psi}$$

$$(\mathsf{APC}^i_n)\ \Diamond[i]\phi_1 \land \Diamond(\overline{\phi}_1 \land [i]\phi_2) \land \cdots \land \Diamond(\overline{\phi}_1 \land \cdots \land \overline{\phi}_{n-1} \land [i]\phi_n) \to \phi_1 \lor \cdots \lor \phi_n$$

Fig. 1: The Hilbert calculus for Ldm^m_n [3,20]. A *derivation of* ϕ *in* Ldm^m_n from a set of premises Γ, is written as $\Gamma \vdash_{\mathsf{Ldm}^m_n} \phi$, and is defined inductively in the usual way. When Γ is the empty set, we refer to ϕ as a *theorem* and write $\vdash_{\mathsf{Ldm}^m_n} \phi$.

sequents. As a result of adopting the approach in [10], the omission of the relational structure corresponding to the $\Box$ modality offers a simpler approach to proving the admissibility of structural rules in the the presence of propagation rules (Sec. 3.2). Let us start by introducing the class of $\mathsf{G3Ldm}^m_n$ calculi.

3.1 The $\mathsf{G3Ldm}^m_n$ Calculi

We define labelled sequents Λ via the following BNF grammar:

$$\Lambda ::= x : \phi \mid \Lambda, \Lambda \mid \mathcal{R}_i xy, \Lambda$$

where $i \in Ag$, $\phi \in \mathcal{L}^m$ and x, y are from a denumerable set of labels $Lab = \{x, y, z, ...\}$. Labelled sequents consist exclusively of labelled formulae of the form $x : \phi$ and relational atoms of the form $\mathcal{R}_i xy$. For this reason, sequents can be partitioned into two parts: we sometimes use the notation $\mathcal{R}, \Gamma$ to denote labelled sequents, where $\mathcal{R}$ is the part consisting of relational atoms and Γ is the part consisting of labelled formulae. Last, we interpret the commas between relational atoms in $\mathcal{R}$ conjunctively, the comma between $\mathcal{R}$ and Γ in $\mathcal{R}, \Gamma$ implicationally, and the commas between labelled formulae in Γ disjunctively (cf. Def. 7).

The labelled STIT calculi $\mathsf{G3Ldm}^m_n$ (where $n, m \in \mathbb{N}$) are shown in Fig. 2. Note that for each agent $i \in Ag$, we obtain a copy for each of the rules $(\langle i\rangle)$, $([i])$, (refl_i), (eucl_i), and (APC^i_n). We refer to (refl_i), (eucl_i), (IOA), and (APC^i_n) as the *structural rules* of $\mathsf{G3Ldm}^m_n$. The rule (IOA) captures the *independence of agents* principle. Furthermore, the rule schema (APC^i_n), limiting the amount of choices available to agent i, provides different rules depending on the value of n in $\mathsf{G3Ldm}^m_n$ (we reserve $n = 0$ to assert that the rule does not appear). When $n > 0$, the (APC^i_n) rule contains $n(n+1)/2$ premises, where each sequent $\mathcal{R}, \mathcal{R}_i x_k x_j, \Gamma$ (for $0 \leq k \leq n-1$ and $k+1 \leq j \leq n$) represents a different premise of the rule. As an example, for $n = 1$ and $n = 2$ the rules for agent i are:

$$\frac{}{\mathcal{R}, w : p, w : \overline{p}, \Gamma}\ (\mathsf{id}) \quad \frac{\mathcal{R}, w : \phi \land \psi, w : \phi, \Gamma \qquad \mathcal{R}, w : \phi \land \psi, w : \psi, \Gamma}{\mathcal{R}, w : \phi \land \psi, \Gamma}\ (\land)$$

$$\frac{\mathcal{R}, w : \phi \lor \psi, w : \phi, w : \psi, \Gamma}{\mathcal{R}, w : \phi \lor \psi, \Gamma}\ (\lor) \quad \frac{\mathcal{R}, \mathcal{R}_i wv, v : \phi, \Gamma}{\mathcal{R}, w : [i]\phi, \Gamma}\ ([i])^\dagger$$

$$\frac{\mathcal{R}, w : \Box\phi, v : \phi, \Gamma}{\mathcal{R}, w : \Box\phi, \Gamma}\ (\Box)^\dagger \quad \frac{\mathcal{R}, w : \Diamond\phi, u : \phi, \Gamma}{\mathcal{R}, w : \Diamond\phi, \Gamma}\ (\Diamond) \quad \frac{\mathcal{R}, \mathcal{R}_1 u_1 v, \ldots, \mathcal{R}_m u_m v, \Gamma}{\mathcal{R}, \Gamma}\ (\mathsf{IOA})^\dagger$$

$$\frac{\mathcal{R}, \mathcal{R}_i wu, w : \langle i\rangle\phi, u : \phi, \Gamma}{\mathcal{R}, \mathcal{R}_i wu, w : \langle i\rangle\phi, \Gamma}\ (\langle i\rangle) \quad \frac{\mathcal{R}, \mathcal{R}_i ww, \Gamma}{\mathcal{R}, \Gamma}\ (\mathsf{refl}_i) \quad \frac{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, \mathcal{R}_i uv, \Gamma}{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, \Gamma}\ (\mathsf{eucl}_i)$$

$$\frac{\Big\{\mathcal{R}, \mathcal{R}_i w_k w_j, \Gamma \;\Big|\; 0 \leq k \leq n-1,\ k+1 \leq j \leq n\Big\}}{\mathcal{R}, \Gamma}\ (\mathsf{APC}^i_n)$$

Fig. 2: The $\mathsf{G3Ldm}^m_n$ labelled calculi. The superscript † on the $(\Box)$, $([i])$, and (IOA) rule names indicates an eigenvariable condition: the variable v occurring in the premise of the rule cannot occur in the context of the premise (or, equivalently, in the conclusion).

$$\frac{\mathcal{R}, \mathcal{R}_i w_0 w_1, \Gamma}{\mathcal{R}, \Gamma}\ (\mathsf{APC}^i_1) \qquad \frac{\mathcal{R}, \mathcal{R}_i w_0 w_1, \Gamma \qquad \mathcal{R}, \mathcal{R}_i w_0 w_2, \Gamma \qquad \mathcal{R}, \mathcal{R}_i w_1 w_2, \Gamma}{\mathcal{R}, \Gamma}\ (\mathsf{APC}^i_2)$$

Theorem 2. *The $\mathsf{G3Ldm}^m_n$ calculi have the following properties:*

1. *All sequents of the form $\mathcal{R}, w : \phi, w : \overline{\phi}, \Gamma$ are derivable;*
2. *Variable-substitution is height-preserving admissible;*
3. *All inference rules are height-preserving invertible;*
4. *Weakening and contractions are height-preserving admissible:*

$$\frac{\mathcal{R}, \Gamma}{\mathcal{R}, \mathcal{R}', \Gamma', \Gamma}\ (\mathsf{wk}) \quad \frac{\mathcal{R}, \mathcal{R}', \mathcal{R}', \Gamma}{\mathcal{R}, \mathcal{R}', \Gamma}\ (\mathsf{ctr})_\mathsf{R} \quad \frac{\mathcal{R}, \Gamma', \Gamma', \Gamma}{\mathcal{R}, \Gamma', \Gamma}\ (\mathsf{ctr})_\mathsf{F}$$

5. *The cut rule is admissible:*

$$\frac{\mathcal{R}, x : \phi, \Gamma \qquad \mathcal{R}, x : \overline{\phi}, \Gamma}{\mathcal{R}, \Gamma}\ (\mathsf{cut})$$

6. *For every formula $\phi \in \mathcal{L}^m$, $w : \phi$ is derivable in $\mathsf{G3Ldm}^m_n$ if and only if $\vdash_{\mathsf{Ldm}^m_n} \phi$, i.e., $\mathsf{G3Ldm}^m_n$ is sound and complete relative to Ldm^m_n.*

Proof. The proof is a basic adaption of [14] and can be found in App. A. □

Proof-theoretic properties like those expressed in (4) and (5) of Thm. 2 are essential when designing decidability procedures via proof-search. In constructing a proof of a sequent, proof-search algorithms proceed by applying inference rules of a calculus bottom-up. A bottom-up application of the (cut) rule in a proof-search procedure, however, requires one to guess the *cut formula* ϕ, and thus risks non-termination in the algorithm. (One can think of similar arguments why $(\mathsf{ctr})_{\mathsf{R}}$ and $(\mathsf{ctr})_{\mathsf{F}}$ risk non-termination.) It is thus crucial that such rules are *admissible*; *i.e.* everything derivable with these rules, is derivable without them.

Remark 1. To obtain contraction admissibility (Thm. 2-(4)) labelled calculi must satisfy the *closure condition* [14]: if a substitution of variables in a structural rule brings about a duplication of relational atoms in the conclusion, then the calculus must contain another instance of the rule with this duplication contracted.

We observe that if we substitute the variable u for v in the structural rule (eucl_i) (below left), we obtain the rule $(\mathsf{eucl}_i)^*$ (below right), when the atom $\mathcal{R}_i wu$ is contracted:

$$\frac{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wu, \mathcal{R}_i uu, \Gamma}{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wu, \Gamma}\ (\mathsf{eucl}_i) \qquad \frac{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i uu, \Gamma}{\mathcal{R}, \mathcal{R}_i wu, \Gamma}\ (\mathsf{eucl}_i)^*$$

Thus, following the closure condition, we must also add $(\mathsf{eucl}_i)^*$ to our calculus. However, $(\mathsf{eucl}_i)^*$ is a special instance of the (refl_i) rule, and hence it is admissible; therefore, we can omit its inclusion in our calculi. None of the other structural rules possess duplicate relational atoms in their conclusions under a substitution of variables, and so, each $\mathsf{G3Ldm}_n^m$ calculus satisfies the closure condition.

3.2 Extracting the $\mathsf{Ldm}_n^m\mathsf{L}$ Calculi

We now refine the $\mathsf{G3Ldm}_n^m$ calculi, extracting new $\mathsf{Ldm}_n^m\mathsf{L}$ calculi to which proof-search techniques from [17] may be adapted. In short, we introduce new rules to our calculi, called *propagation rules*, which are well-suited for proof-search and imply the admissibility of the less suitable structural rules (refl_i) and (eucl_i).

Propagation rules are special sequent rules that possess a nonstandard side condition, consisting of two components. For the first component (1), we transform the sequent occurring in the premise of the rule into an *automaton*. The labels appearing in the sequent determine the states of the automaton, whereas the relational atoms of the sequent determine the transitions between these states. The following definition, based on [17, Def. 4.1], makes this notion precise:

Definition 4 (Propagation Automaton). *Let Λ be a labelled sequent, $Lab(\Lambda)$ be the set of labels occurring in Λ, and $w, u \in Lab(\Lambda)$. We define a* propagation automaton *$\mathcal{P}_\Lambda(w,u)$ to be the tuple $(\Sigma, S, I, F, \delta)$ s.t. (i) $\Sigma := \{\langle i \rangle \mid i \in Ag\}$ is the automaton's* alphabet*, (ii) $S := Lab(\Lambda)$ is the* set of states*, (iii) $I := \{w\}$ is the* initial state*, (iv) $F := \{u\}$ is the* accepting state*, and (v) $\delta : S \times \Sigma \to S$ is the* transition function *where $\delta(v, \langle i \rangle) = v'$ and $\delta(v', \langle i \rangle) = v$ iff $\mathcal{R}_i vv' \in \Lambda$.*

We will often write $v \xrightarrow{\langle i \rangle} v'$ instead of $\delta(v, \langle i \rangle) = v'$ to denote a transition between states. A string *is a, possibly empty, concatenation of symbols from Σ*

(where ε indicates the empty string). We say that an automaton accepts *a string* $\omega = \langle i_1 \rangle \langle i_2 \rangle \cdots \langle i_k \rangle$ *iff there exists a transition sequence* $w \xrightarrow{\langle i_1 \rangle} v \xrightarrow{\langle i_2 \rangle} \cdots \xrightarrow{\langle i_k \rangle} u$ *from the initial state* w *to the accepting state* u*. Last, we will abuse notation and use* $\mathcal{P}_\Lambda(w, u)$ *equivocally to represent both the automaton and the set of strings* ω *accepted by the automaton, i.e.* $\{\omega \mid \mathcal{P}_\Lambda(w, u)$ *accepts string* $\omega\}$*. The use of notation can be determined from the context.*

The second component (2) of the rule's side condition restricts the application of the rule to a particular language that specifies and determines which types of strings occurring in the automaton allow for a correct application of the propagation rule. We define this language accordingly:

Definition 5 (Agent i Application Language). *For each* $i \in Ag$*, we define the* application language L_i *to be the language generated from the regular expression* $\langle i \rangle^*$*, that is,* $L_i = \{\varepsilon, \langle i \rangle, \langle i \rangle\langle i \rangle, \langle i \rangle\langle i \rangle\langle i \rangle, \cdots\}$ *with* ε *the empty string.*[2]

Bringing components (1) and (2) together, a propagation rule is applicable only if the associated propagation automaton accepts a certain string—corresponding to a path of relational atoms in the premise of the rule—and the string is in the application language.

Definition 6 (Propagation Rule). *Let* $i \in Ag$*,* $\Lambda_1 = \mathcal{R}, w : \langle i \rangle \phi, u : \phi, \Gamma$*, and* $\Lambda_2 = \mathcal{R}, w : \langle i \rangle \phi, \Gamma$*. The* propagation rule (Pr_i) *is defined as follows:*

$$\frac{\mathcal{R}, w : \langle i \rangle \phi, u : \phi, \Gamma}{\mathcal{R}, w : \langle i \rangle \phi, \Gamma} \ (\mathsf{Pr}_i)^{\dagger\dagger}$$

The superscript $\dagger\dagger$ *indicates that* $\mathcal{P}_{\Lambda_k}(w, u) \cap L_i \neq \emptyset$ *for* $k \in \{1, 2\}$*.*[3]

We use $\mathsf{PR} := \{(\mathsf{Pr}_i) \mid i \in Ag\}$ *to represent the set of all propagation rules.*

The underlying intuition of the rule (applied bottom-up) is that, given some labelled sequent Λ, a formula ϕ is propagated from $w : \langle i \rangle \phi$ to another label u, if w and u are connected by a sequence of $\mathcal{R}_i$ relational atoms in Λ (with i fixed). In the corresponding propagation automaton $\mathcal{P}_\Lambda(w, u)$, this amounts to the existence of a string $\omega \in \mathcal{P}_\Lambda(w, u) \cap L_i$ which represents a sequence of transitions from w to u, such that all transitions are solely labelled with $\langle i \rangle$. To see how the language L_i secures the soundness of the rule, we refer to Thm. 4. For an introduction to propagation rules and propagation automata, see [17].

Let us make the introduced notions more concrete by providing an example:

Example 1. Let $\Lambda = \mathcal{R}_1 wu, \mathcal{R}_2 uv, \mathcal{R}_1 vz, w : \langle 1 \rangle \phi$. The propagation automaton $\mathcal{P}_\Lambda(w, z)$ is depicted graphically as (where the single-boxed node w designates the initial state and a double-boxed node z represents the accepting state):

[2] For further information on regular languages and expressions, consult [16].

[3] Observe that $\mathcal{P}_{\Lambda_1}(w, u) = \mathcal{P}_{\Lambda_2}(w, u)$. Hence, deciding which automaton to employ in determining the side condition is inconsequential: when applying the rule top-down we may consult Λ_1, whereas during bottom-up proof-search we may regard Λ_2.

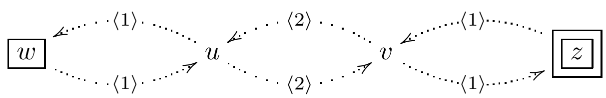
$$\frac{}{\mathcal{R}, w : p, w : \overline{p}, \Gamma}\ (\mathsf{id}) \qquad \frac{\mathcal{R}, w : \phi \wedge \psi, w : \phi, \Gamma \qquad \mathcal{R}, w : \phi \wedge \psi, w : \psi, \Gamma}{\mathcal{R}, w : \phi \wedge \psi, \Gamma}\ (\wedge)$$

$$\frac{\mathcal{R}, w : \phi \vee \psi, w : \phi, w : \psi, \Gamma}{\mathcal{R}, w : \phi \vee \psi, \Gamma}\ (\vee) \quad \frac{\mathcal{R}, w : \Box\phi, v : \phi, \Gamma}{\mathcal{R}, w : \Box\phi, \Gamma}\ (\Box)^{\dagger} \quad \frac{\mathcal{R}, w : \Diamond\phi, u : \phi, \Gamma}{\mathcal{R}, w : \Diamond\phi, \Gamma}\ (\Diamond)$$

$$\frac{\mathcal{R}, \mathcal{R}_1 u_1 v, ..., \mathcal{R}_m u_m v, \Gamma}{\mathcal{R}, \Gamma}\ (\mathsf{IOA})^{\dagger} \quad \frac{\mathcal{R}, \mathcal{R}_i wv, w : [i]\phi, v : \phi, \Gamma}{\mathcal{R}, w : [i]\phi, \Gamma}\ ([i])^{\dagger}$$

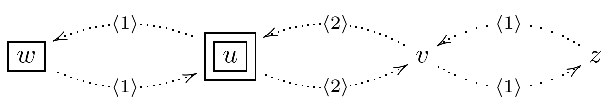
$$\frac{\mathcal{R}, w : \langle i \rangle\phi, u : \phi, \Gamma}{\mathcal{R}, w : \langle i \rangle\phi, \Gamma}\ (\mathsf{Pr}_i)^{\dagger\dagger} \quad \frac{\Big\{\mathcal{R}, \mathcal{R}_i w_k w_j, \Gamma \;\Big|\; 0 \leq k \leq n-1,\ k+1 \leq j \leq n\Big\}}{\mathcal{R}, \Gamma}\ (\mathsf{APC}^i_n)$$

Fig. 3: The labelled calculus $\mathsf{Ldm}^m_n\mathsf{L}$. The superscript † on the $(\Box)$, $([i])$, and (IOA) rules indicate that v is an eigenvariable. The †† side condition is the same as in Def. 6. Last, we have $([i])$, (Pr_i), and (APC^i_n) rules for each $i \in Ag$.



Observe that every string the automaton accepts must contain at least one $\langle 2 \rangle$ symbol. Since no string of this form exists in L_1, it is not valid to propagate the formula ϕ to z. That is, the sequent $\mathcal{R}_1 wu, \mathcal{R}_2 uv, \mathcal{R}_1 vz, w : \langle 1 \rangle\phi, z : \phi$ does not follow from applying the propagation rule (Pr_1) (bottom-up) to Λ.

On the other hand, consider the propagation automaton $\mathcal{P}_\Lambda(w, u)$:



The automaton accepts the simple string $\langle 1 \rangle$, which is included in the language L_1. Therefore, it is permissible to apply the propagation rule (Pr_1) (bottom-up) and derive $\mathcal{R}_1 wu, \mathcal{R}_2 uv, \mathcal{R}_1 vz, w : \langle 1 \rangle\phi, u : \phi$ from Λ.

Remark 2. We observe that both of the languages $\mathcal{P}_\Lambda(w, u)$ and L_i are regular, and thus, the problem of determining whether $\mathcal{P}_\Lambda(w, u) \cap L_i \neq \emptyset$, is decidable [17]. Consequently, the propagation rules in PR may be integrated into our proof-search algorithm without risking non-termination.

The proof theoretic properties of $\mathsf{G3Ldm}^m_n$ are preserved when extended with the set of propagation rules PR (Lem. 1). Moreover, the nature of our propagation rules allows us to prove the admissibility of the structural rules (refl_i) and (eucl_i), for each $i \in Ag$ (resp. Lem. 2 and 3), which results in the refined calculi $\mathsf{Ldm}^m_n\mathsf{L}$ (shown in Fig. 3). The proofs of Lem. 1 and 2 are App. A (the latter is similar to the proof of Lem. 3 presented here).

Lemma 1. *The* $\mathsf{G3Ldm}_n^m$+PR *calculi have the following properties: (i) all sequents* Λ *of the form* $\Lambda = \mathcal{R}, w\colon \phi, w\colon \overline{\phi}, \Gamma$ *are derivable; (ii) variable-substitution is height-preserving admissible; (iii) all inference rules are height-preserving invertible; (iv) the* (wk)*,* $(\mathsf{ctr})_{\mathsf{R}}$ *and* $(\mathsf{ctr})_{\mathsf{F}}$ *rules are height-preserving admissible.*

Lemma 2 **(**(refl_i)**-Elimination).** *Every sequent* Λ *derivable in* $\mathsf{G3Ldm}_n^m + \mathsf{PR}$ *is derivable without the use of* (refl_i)*.*

Lemma 3 **(**(eucl_i)**-Elimination).** *Every sequent* Λ *derivable in* $\mathsf{G3Ldm}_n^m + \mathsf{PR}$ *is derivable without the use of* (eucl_i)*.*

Proof. The result is proven by induction on the height of the given derivation. We show that the topmost instance of a (eucl_i) rule can be permuted upward in a derivation until it is eliminated entirely; by successively eliminating each (eucl_i) inference from the derivation, we obtain a derivation free of such inferences. Also, we evoke Lem. 2 and assume that all instances of (refl_i) have been eliminated from the given derivation.

Base Case. An application of (eucl_i) on an initial sequent (below left) can be re-written as an instance of the (id) rule (below right).

$$\dfrac{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, \mathcal{R}_i uv, z : p, z : \overline{p}, \Gamma}{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, z : p, z : \overline{p}, \Gamma}\ (\mathsf{eucl}_i) \qquad \dfrac{}{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, z : p, z : \overline{p}, \Gamma}\ (\mathsf{id})$$

Inductive step. We show the inductive step for the non-trivial cases: $(\langle i \rangle)$ and (Pr_i) (case (i) and (ii), respectively). All other cases are resolved by applying IH to the premise followed by an application of the corresponding rule.

(i). Let $\mathcal{R}_i uv$ be active in the $(\langle i \rangle)$ inference of the initial derivation (below (1)). Observe that when we apply the (eucl_i) rule first (below (2)), the atom $\mathcal{R}_i uv$ is no longer present in $\Lambda = \mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, u : \langle i \rangle \phi, v : \phi, \Gamma$, and so, the $(\langle i \rangle)$ rule is not necessarily applicable. Nevertheless, we may apply the (Pr_i) rule to derive the desired conclusion since $\langle i \rangle \langle i \rangle \in \mathcal{P}_\Lambda(u, v) \cap L_i$. Namely, the fact that $\langle i \rangle \langle i \rangle \in \mathcal{P}_\Lambda(u, v)$ only relies on the presence of $\mathcal{R}_i wu, \mathcal{R}_i wv$ in Λ.

$$\dfrac{\dfrac{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, \mathcal{R}_i uv, u : \langle i \rangle \phi, v : \phi, \Gamma}{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, \mathcal{R}_i uv, u : \langle i \rangle \phi, \Gamma}\ (\langle i \rangle)}{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, u : \langle i \rangle \phi, \Gamma}\ (\mathsf{eucl}_i) \tag{1}$$

$$\dfrac{\dfrac{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, \mathcal{R}_i uv, u : \langle i \rangle \phi, v : \phi, \Gamma}{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, u : \langle i \rangle \phi, v : \phi, \Gamma}\ (\mathsf{eucl}_i)}{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, u : \langle i \rangle \phi, \Gamma}\ (\mathsf{Pr}_i) \tag{2}$$

(ii). Let Λ_1 be the first premise $\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, \mathcal{R}_i uv, x : \langle i \rangle \phi, y : \phi, \Gamma$ of the initial derivation (below (3)). In the (Pr_i) inference of the top derivation, we assume that $\mathcal{R}_i uv$ is active, that is, the side condition of (Pr_i) is satisfied because some string $\langle i \rangle^n \in \mathcal{P}_{\Lambda_1}(x, y) \cap L_i$ with $n \in \mathbb{N}$. (NB. For the non-trivial case, we assume that $\langle i \rangle^n \in \mathcal{P}_{\Lambda_1}(x, y)$ relies on the presence of $\mathcal{R}_i uv \in \Lambda_1$, that is, the

automaton $\mathcal{P}_{\Lambda_1}(x, y)$ makes use of transitions $u \xrightarrow{\langle i \rangle} v$ or $v \xrightarrow{\langle i \rangle} u$ defined relative to $\mathcal{R}_i uv$.) When we apply the (eucl_i) rule first in our derivation (below (4)), we can no longer rely on the relational atom $\mathcal{R}_i uv$ to apply the (Pr_i) rule. However, due to the presence of $\mathcal{R}_i wu, \mathcal{R}_i wv$ in $\Lambda_2 = \mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, x : \langle i \rangle \phi, y : \phi, \Gamma$ we may still apply the (Pr_i) rule. Namely, since $\langle i \rangle^n \in \mathcal{P}_{\Lambda_1}(x, y)$, we know there is a sequence of n transitions $x \xrightarrow{\langle i \rangle} z_1 \xrightarrow{\langle i \rangle} \cdots z_{n-1} \xrightarrow{\langle i \rangle} y$ from x to y. We replace each occurrence of $u \xrightarrow{\langle i \rangle} v$ with $u \xrightarrow{\langle i \rangle} w \xrightarrow{\langle i \rangle} v$ and each occurrence of $v \xrightarrow{\langle i \rangle} u$ with $v \xrightarrow{\langle i \rangle} w \xrightarrow{\langle i \rangle} u$. There will thus be a string in $\mathcal{P}_{\Lambda_2}(x, y) \cap L_i$, and so, the (Pr_i) rule may be applied.

$$\dfrac{\dfrac{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, \mathcal{R}_i uv, x : \langle i \rangle \phi, y : \phi, \Gamma}{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, \mathcal{R}_i uv, x : \langle i \rangle \phi, \Gamma}(\mathsf{Pr}_i)}{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, x : \langle i \rangle \phi, \Gamma}(\mathsf{eucl}_i) \tag{3}$$

$$\dfrac{\dfrac{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, \mathcal{R}_i uv, x : \langle i \rangle \phi, y : \phi, \Gamma}{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, x : \langle i \rangle \phi, y : \phi, \Gamma}(\mathsf{eucl}_i)}{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i wv, x : \langle i \rangle \phi, \Gamma}(\mathsf{Pr}_i) \tag{4}$$

□

Theorem 3 (Cut-free Completeness of $\mathsf{Ldm}_n^m\mathsf{L}$). *For any formula $\phi \in \mathcal{L}^m$, if $\Vdash \phi$, then $x : \phi$ is cut-free derivable in $\mathsf{Ldm}_n^m\mathsf{L}$.*

Proof. Follows from Thm. 2, Lem.'s 1–3, and the fact that, for each $i \in Ag$, the $(\langle i \rangle)$ rule is admissible, that is, the $(\langle i \rangle)$ rule is an instance of the rule (Pr_i). □

Last, we must ensure that $\mathsf{Ldm}_n^m\mathsf{L}$ is sound. To prove this, we need to stipulate how to interpret sequents on Ldm_n^m-models. Our definition is based on [4]:

Definition 7 (Interpretation, Satisfaction, Validity). *Let M be an Ldm_n^m-model with domain W, $\Lambda = \mathcal{R}, \Gamma$ a labelled sequent, and Lab the set of labels. Let I be an* interpretation function *mapping labels to worlds: i.e. I: $Lab \mapsto W$.*

Λ is satisfied *in M with I iff for all relational atoms $\mathcal{R}_i xy \in \mathcal{R}$, if $\mathcal{R}_i x^I y^I$ holds in M, then there exists some $z : \phi \in \Gamma$ such that $M, z^I \Vdash \phi$.*

Λ is valid *iff it is satisfiable in every M with any interpretation function I.*

Theorem 4 ($\mathsf{Ldm}_n^m\mathsf{L}$ Soundness). *Every sequent derivable in $\mathsf{Ldm}_n^m\mathsf{L}$ is valid.*

Proof. We know by Thm. 2 that all rules of $\mathsf{Ldm}_n^m\mathsf{L}$, with the exception of (Pr_i), preserve validity. Details of the (Pr_i) rule are given in App. A. □

4 Proof-Search and Decidability

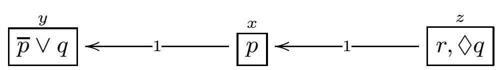
In this section, we provide a class of proof-search algorithms, each deciding a logic Ldm_n^1 (with $n \in \mathbb{N}$). (We use 1 to denote the agent in the single-agent setting.) In the single-agent case, the independence of agents condition is trivially satisfied, meaning we can omit the (IOA) rule from each calculus and from consideration during proof-search. We end the section by commenting on the more complicated multi-agent setting.

In what follows, we prove that derivations in $\mathsf{Ldm}^1_n\mathsf{L}$ need only use *forestlike sequents*. The forestlike structure of a sequent Λ refers to a graph corresponding to the sequent. This control in sequential structure is what allows us to adapt methods from [17] to $\mathsf{Ldm}^1_n\mathsf{L}$, and produce a proof-search algorithm that decides Ldm^1_n, for each $n \in \mathbb{N}$. Let us start by making the aforementioned notions precise.

Definition 8 (Sequent Graph). *We define a* graph G *to be a tuple* (V, E, L), *where* V *is the non-empty* set of vertices, *the* set of edges $E \subseteq V \times V$, *and* L *is the* labelling function *that maps edges from* E *into some non-empty set* S *and vertices from* V *into some non-empty set* S'.

Let $\Lambda = \mathcal{R}, \Gamma$ *be a labelled sequent and let* $Lab(\Lambda)$ *be the set of labels in* Λ. *The* graph of Λ, *denoted* $G(\Lambda)$, *is the tuple* (V, E, L), *where (i)* $V = Lab(\Lambda)$, *(ii)* $(w, u) \in E$ *and* $L(w, u) = i$ *iff* $\mathcal{R}_i wu \in \mathcal{R}$, *and (iii)* $L(w) = \phi$ *iff* $w : \phi \in \Gamma$.

Example 2. The sequent *graph* $G(\Lambda)$ corresponding to the labelled sequent $\Lambda = \mathcal{R}_1 xy, \mathcal{R}_1 zx, x : p, y : \overline{p} \vee q, z : r, z : \Diamond q$ is shown below:



Definition 9 (Tree, Forest, Forestlike Sequent, Choice-tree). *We say that a graph* $G = (V, E, L)$ *is a* tree *iff there exists a node* w, *called the* root, *such that there is exactly one directed path from* w *to any other node* u *in the graph. We say that a graph is a* forest *iff it consists of a disjoint union of trees.*

A sequent Λ *is* forestlike *iff its graph* $G(\Lambda)$ *is a forest. We refer to each disjoint tree in the graph of a forestlike sequent as a* choice-tree *and for any label* w *in* Λ, *we let* $CT(w)$ *represent the choice-tree that* w *belongs to.*

The above notions will be significant for our proof-search algorithms, for example:

Remark 3. When interpreting a sequent, each choice-tree that occurs in the graph of the sequent is a syntactic representation of an equivalence class of $\mathcal{R}_1$ (i.e., a choice-cell for agent 1). Using this insight, we know that if agent 1 is restricted to n-many choices, then if there are $m > n$ choice-trees in the sequent, at least two choice-trees must correspond to the same equivalence class in $\mathcal{R}_1$. We use this observation to specify how APC^1_n is applied in the algorithm.

The following definitions introduce the necessary tools for the algorithms:

Definition 10 (Saturation, $\Box-$, $[1]$-realization, $\Diamond-$, $\langle 1 \rangle$-propagated). *Let* Λ *be a forestlike sequent and let* w *be a label in* Λ.

The label w *is* saturated *iff the following hold: (i) If* $w : \phi \in \Lambda$, *then* $w : \overline{\phi} \notin \Lambda$, *(ii) if* $w : \phi \vee \psi \in \Lambda$, *then* $w : \phi \in \Lambda$ *and* $w : \psi \in \Lambda$, *(iii) if* $w : \phi \wedge \psi \in \Lambda$, *then* $w : \phi \in \Lambda$ *or* $w : \psi \in \Lambda$.

A label w in Λ is $\Box$-realized *iff for every $w : \Box\phi \in \Lambda$, there exists a label u such that $u : \phi \in \Lambda$. A label w in Λ is* $[1]$-realized *iff for every $w : [1]\phi \in \Lambda$, there exists a label u in $CT(w)$ such that $u : \phi \in \Lambda$.*

A label w in Λ is $\Diamond$-propagated *iff for every $w : \Diamond\phi \in \Lambda$, we have $u : \phi \in \Lambda$ for all labels u in Λ. A label w in Λ is* $\langle 1 \rangle$-propagated *iff for every $w : \langle 1 \rangle\phi \in \Lambda$, we have $u : \phi \in \Lambda$ for all labels u in $CT(w)$.*

Definition 11 (n-choice Consistency). *Let Λ be a forestlike sequent and let our logic be Ldm_n^1 with $n > 0$. We say that Λ is* n-choice consistent *iff $G(\Lambda)$ contains at most n-many choice-trees.*

Definition 12 (Stability). *A forestlike labelled sequent Λ is* stable *iff (i) all labels w in Λ are saturated, (ii) all labels are $\Box$- and $[1]$-realized, (iii) all labels are $\Diamond$- and $\langle 1 \rangle$-propagated, and (iv) Λ is n-choice consistent.*

We are now able to define our proof-search algorithms for the logics Ldm_n^1. The algorithms are provided in Fig. 4 and are inspired by [17]. We emphasize that the execution of instruction 4 in Fig. 4 corresponds to an instance of the (Pr_1) rule. The algorithms are correct (Thm. 5) and terminate (Thm. 6). Last, Lem. 4 ensures that the concepts of realization, propagation, n-choice consistency, and stability are defined at each stage of the computation (Def. 10 - 12). The proofs of Lem. 4 and Thm. 6 can be found in App. A.

Lemma 4. *Every labelled sequent generated throughout the course of computing $\mathtt{Prove_n}(w : \phi)$ is forestlike.*

Theorem 5 (Correctness). (i) *If $\mathtt{Prove_n}(\mathtt{w} : \phi)$ returns* `true`*, then $w : \phi$ is $\mathsf{Ldm}_n^1\mathsf{L}$-provable.* (ii) *If $\mathtt{Prove_n}(\mathtt{w} : \phi)$ returns* `false`*, then $w : \phi$ is not $\mathsf{Ldm}_n^1\mathsf{L}$-provable.*

Proof. (i) It suffices to observe that each step of $\mathtt{Prove_n}(\cdot)$ is a backwards application of a rule in $\mathsf{Ldm}_n^1\mathsf{L}$, and so, if the proof-search algorithm returns `true`, the formula $w : \phi$ is derivable in $\mathsf{Ldm}_n^1\mathsf{L}$ with arbitrary label w.

(ii) To prove this statement, we assume that $\mathtt{Prove_n}(\mathtt{w} : \phi)$ returned `false` and show that we can construct a counter-model for ϕ. By assumption, we know that a stable sequent Λ was generated with $w : \phi \in \Lambda$. We define our counter-model $M = (W, \mathcal{R}_1, V)$ as follows: $W = Lab(\Lambda)$; $\mathcal{R}_1 uv$ *iff* $\mathcal{P}_\Lambda(u, v) \cap L_1 \neq \emptyset$; and $w \in V(p)$ *iff* $w : \overline{p} \in \Lambda$.

We argue that $F = (W, \mathcal{R}_1)$ is an Ldm_n^m-frame. It is easy to see that $W \neq \emptyset$ (at the very least, the label w must occur in Λ). Moreover, condition (C2) is trivially satisfied in the single-agent setting. We prove (C1) and (C3):

(C1) We need to prove that $\mathcal{R}_1$ is (i) reflexive and (ii) Euclidean. To prove (i), it suffices to show that for each $u \in Lab(\Lambda)$ there exists a string ω in both $\mathcal{P}_\Lambda(u, u)$ and L_1. By Def. 4, we know that $\varepsilon \in \mathcal{P}_\Lambda(u, u)$ since u is both the initial and accepting state. Also, by Def. 5 we know that $\varepsilon \in L_1$. To prove (ii), we assume that $\mathcal{R}_1 wu$ and $\mathcal{R}_1 wv$ hold, and show that $\mathcal{R}_1 uv$ holds as well. By our assumption, there exist strings $\langle 1 \rangle^k \in \mathcal{P}_\Lambda(w, u) \cap L_1$ and $\langle 1 \rangle^m \in \mathcal{P}_\Lambda(w, v) \cap L_1$ (with $k, m \in \mathbb{N}$). It is not difficult to prove that if $\langle 1 \rangle^k \in \mathcal{P}_\Lambda(w, u)$, then

Function $\texttt{Prove}_{\texttt{n}}$(Sequent $\mathcal{R}, \Gamma$) : Boolean

1. If $\mathcal{R}, \Gamma = \mathcal{R}, w : p, w : \overline{p}, \Gamma'$, return true.
2. If $\mathcal{R}, \Gamma$ is stable, return false.
3. If some label w in $\mathcal{R}, \Gamma$ is not saturated, then:
   (i) If $w : \phi \vee \psi \in \mathcal{R}, \Gamma$, but either $w : \phi \notin \mathcal{R}, \Gamma$ or $w : \psi \notin \mathcal{R}, \Gamma$, then let $\mathcal{R}, \Gamma' = \mathcal{R}, w : \phi, w : \psi, \Gamma$ and return $\texttt{Prove}_{\texttt{n}}(\mathcal{R}, \Gamma')$.
   (ii) If $w : \phi \wedge \psi \in \mathcal{R}, \Gamma$, but neither $w : \phi \notin \mathcal{R}, \Gamma$ nor $w : \psi \notin \mathcal{R}, \Gamma$, then let $\mathcal{R}, \Gamma_1 = \mathcal{R}, w : \phi, \Gamma$, let $\mathcal{R}, \Gamma_2 = \mathcal{R}, w : \psi, \Gamma$, and return false if $\texttt{Prove}_{\texttt{n}}(\mathcal{R}, \Gamma_i)$ = false for some $i \in \{1, 2\}$, and return true otherwise.
4. If some label w in $\mathcal{R}, \Gamma$ is not $\langle 1 \rangle$-propagated, then there is a label u in $CT(w)$ such that $u : \phi \notin \Gamma$. Let $\mathcal{R}, \Gamma' = \mathcal{R}, u : \phi, \Gamma$ and return $\texttt{Prove}_{\texttt{n}}(\mathcal{R}, \Gamma')$.
5. If some label w in $\mathcal{R}, \Gamma$ is not $\Diamond$-propagated, then there is a label u such that $u : \phi \notin \Gamma$. Let $\mathcal{R}, \Gamma' = \mathcal{R}, u : \phi, \Gamma$ and return $\texttt{Prove}_{\texttt{n}}(\mathcal{R}, \Gamma')$.
6. If there is a label w that is not $[1]$-realized, then there is a $w : [1]\phi \in \Gamma$ such that $u : \phi \notin \Gamma$ for every label $u \in CT(w)$. Let $\mathcal{R}', \Gamma' = \mathcal{R}, \mathcal{R}_1 wv, v : \phi, \Gamma$ with v fresh and return $\texttt{Prove}_{\texttt{n}}(\mathcal{R}', \Gamma')$.
7. If there is a label w that is not $\Box$-realized, then there is a $w : \Box\phi \in \Gamma$ such that $u : \phi \notin \Gamma$ for every label u in $\mathcal{R}, \Gamma$. Let $\mathcal{R}, \Gamma' = \mathcal{R}, v : \phi, \Gamma$ with v fresh and return $\texttt{Prove}_{\texttt{n}}(\mathcal{R}, \Gamma')$.
8. If $\mathcal{R}, \Gamma$ is not n-choice consistent, then let $\mathcal{R}_{k,j}, \Gamma = \mathcal{R}, \mathcal{R}_1 w_k w_j, \Gamma$ (with $0 \leq k \leq n-1$ and $k+1 \leq j \leq n$) and where each w_k and w_j are distinct roots of choice-trees in $\mathcal{R}, \Gamma$. Return false if $\texttt{Prove}_{\texttt{n}}(\mathcal{R}_{k,j}, \Gamma)$ = false for some k and j, and return true otherwise.

Fig. 4: The proof-search algorithms for Ldm^1_n with $n > 0$. The algorithm for Ldm^1_0 is obtained by deleting line 8.

$\langle 1 \rangle^k \in \mathcal{P}_\Lambda(u, w)$, and also that if $\langle 1 \rangle^k \in \mathcal{P}_\Lambda(u, w)$ and $\langle 1 \rangle^m \in \mathcal{P}_\Lambda(w, v)$, then $\langle 1 \rangle^{k+m} \in \mathcal{P}_\Lambda(u, v)$. Hence, we know $\langle 1 \rangle^{k+m} \in \mathcal{P}_\Lambda(u, v)$, which, together with $\langle 1 \rangle^{k+m} \in L_1$ (Def. 5), gives us the desired $\mathcal{R}_1 uv$.

(C3) By assumption we know Λ is stable. Consequently, when $n > 0$ for $\mathsf{Ldm}^1_n\mathsf{L}$, the sequent Λ must be n-choice consistent. Hence, the graph of Λ must contain $k \leq n$ choice-trees. Condition (C3) follows straightforwardly.

Since F is an Ldm^m_n-frame, M is an Ldm^m_n-model. We show by induction on the complexity of ψ that for any $u : \psi \in \Lambda$, $M, u \not\Vdash \psi$. Consequently, M is a counter-model for ϕ, and so, by Thm. 4, we know $w : \phi$ is not provable in $\mathsf{Ldm}^1_n\mathsf{L}$.

Base Case. Assume $u : p \in \Lambda$. Since Λ is stable, we know that $u : \overline{p} \notin \Lambda$. Hence, by the definition of V, we know that $u \notin V(p)$, implying that $M, u \not\Vdash p$.

Inductive Step. We consider each connective in turn. (i) Assume that $u : \theta \vee \chi \in \Lambda$. Since Λ is stable, it is saturated, meaning that $u : \theta, u : \chi \in \Lambda$. Hence, by IH, $M, u \not\Vdash \theta$ and $M, u \not\Vdash \chi$, which implies that $M, u \not\Vdash \theta \vee \chi$. (ii) The case $u : \theta \wedge \chi \in \Lambda$ is similar to the previous case. (iii) Assume $u : \langle 1 \rangle\theta \in \Lambda$. Since Λ is stable, we know that every label is $\langle 1 \rangle$-propagated. Therefore, for all labels $v \in CT(u)$ we have $v : \theta \in \Lambda$. By IH, $M, v \not\Vdash \theta$ for all $v \in CT(u)$. In general, the definition of $\mathcal{R}_1$ implies that $\mathcal{R}_1 xy$ iff $y \in CT(x)$. The former two statements imply that $M, v \not\Vdash \theta$ for all v such that $\mathcal{R}_1 uv$, and so, $M, u \not\Vdash \langle 1 \rangle \theta$. (iv) Assume that $u : \Diamond\theta \in \Lambda$. Since Λ is stable, every label is $\Diamond$-propagated, which implies that for all labels v in Λ, $v : \theta \in \Lambda$. By IH, this implies that for all $v \in W$, $M, v \not\Vdash \theta$. Thus, $M, u \not\Vdash \Diamond\theta$. (v) Assume $u : [1]\theta \in \Lambda$. Since Λ is stable, we know every label in Λ is [1]-realized. Therefore, there exists a label v in $CT(u)$ such that $v : \theta \in \Lambda$. By IH, we conclude that $M, v \not\Vdash \theta$. Moreover, since $\mathcal{R}_1 xy$ iff $y \in CT(x)$, we also know that $\mathcal{R}_1 uv$, which implies $M, u \not\Vdash [1]\psi$. (vi) Assume $u : \Box\theta \in \Lambda$. Since Λ is stable, we know that every label is $\Box$-realized. Consequently, there exists a label v such that $v : \theta \in \Lambda$. By IH, we conclude $M, v \not\Vdash \theta$; hence, $M, u \not\Vdash \Box\theta$. □

Theorem 6 (Termination). *For each formula $w : \phi$,* $\texttt{Prove}_{\texttt{n}}(\texttt{w}\colon \phi)$ *terminates.*

Corollary 1 (Decidability and FMP). *For each $n \in \mathbb{N}$, the logic Ldm^1_n is decidable and has the finite model property.*

Proof. Follows from Thm. 5 and 6 above. The finite model property follows from the fact that the counter-models constructed in Thm. 5 are all finite. □

Additionally, from a computational viewpoint, it is interesting to know if completeness is preserved under a restricted class of sequents (cf. [6]). Indeed, Lem. 4, Thm. 5 and Thm. 6, imply that completeness is preserved when we restrict $\mathsf{Ldm}^1_n\mathsf{L}$ derivations to forestlike sequents; that is, when inputting a formula into our algorithms, the sequent produced at each step of the computation will be forestlike. Interestingly, this result was obtained via our proof-search algorithms.

Corollary 2 (Forestlike Derivations). *For each $n \in \mathbb{N}$, if a labelled formula $w : \phi$ is derivable in $\mathsf{Ldm}^1_n\mathsf{L}$, then it is derivable using only forestlike sequents.*

A Note on the Multi-Agent Setting of $\mathsf{Ldm}^m_n\mathsf{L}$. As a concluding remark, we briefly touch upon extending the current results to the multi-agent calculi $\mathsf{Ldm}^m_n\mathsf{L}$. In the multi-agent setting (when $n = 0$), our sequents have the structure of *directed acyclic graphs* (i.e., directed graphs free of cycles), due to the independence of agents rule (IOA). In such graphs, one can easily recognize loop-nodes—i.e., a path from an ancestor node to the alleged loop-node such that both nodes are labelled with the same multiset of formulae—and use this information to bound the depth of the sequent during proof-search (cf. [17]).

The main challenge concerns the (IOA) rule, which when applied bottom-up during proof-search, introduces a fresh label v to the sequent. As a consequence, one must ensure that if proof-search terminates in a counter-model construction, this label v satisfies the independence of agents condition in that model. At first glance, one might conjecture that for every application of the (IOA) rule an additional application of the rule is needed to saturate the independence of agents condition. Of course, in such a case the algorithm will not terminate with a sequent that is readily convertible to a counter-model. Fortunately, it turns out that only finitely many applications of (IOA) are needed to construct a counter-model satisfying independence of agents. The authors have planned to devote their future work to answer this open problem for the multi-agent setting.

5 Conclusion

This paper introduced the first cut-free complete calculi for the class of multi-agent Ldm_n^m logics, introduced in [20]. We adapted propagation rules, discussed in [17], in order to refine the multi-agent $\mathsf{G3Ldm}_n^m$ labelled calculi and generate the proof-search friendly $\mathsf{Ldm}_n^m\mathsf{L}$ calculi. For the single agent case, we provided a class of terminating proof-search algorithms, each deciding a logic Ldm_n^1 (with $n \in \mathbb{N}$), including counter-model extraction from failed proof-search.

As discussed in Sec. 4, we plan to devote future research to leveraging the current results for the multi-agent setting and to provide terminating proof-search procedures for the entire Ldm_n^m class. As a natural extension, we aim to implement the proof-search algorithms from Sec. 4 in Prolog (e.g., as in [8]). Additionally, we plan to expand the current framework to include deontic STIT operators (e.g., from [11,13]) with the goal of automating normative, agent-based reasoning. Last, it is shown in [2] that Ldm_0^1 has an NP-complete satisfiability problem and each logic $\mathsf{Ldm}_0^\mathsf{m}$, with $m > 0$, is NEXPTIME-complete. Along with expanding our proof-search algorithms to the class of all Ldm_n^m logics, we aim to investigate the complexity and optimality of our associated algorithms.

References

1. Arkoudas, K., Bringsjord S., Bello, P.: Toward ethical robots via mechanized deontic logic. In: AAAI Fall Symposium on Machine Ethics, pp.17–23 (2005)
2. Balbiani, P., Herzig, A., Troquard, N.: Alternative axiomatics and complexity of deliberative STIT theories. Journal of Philosophical Logic, 37(4), pp.387–406. Springer (2008)
3. Belnap, N, Perloff, M., Xu, M.: Facing the future: Agents and choices in our indeterminist world. Oxford University Press on Demand, Oxford (2001)
4. Berkel, K. van, Lyon, T.: Cut-free Calculi and Relational Semantics for Temporal STIT Logics. In: Joint European Conference on Logics in Artificial Intelligence (JELIA), pp.403–419, Springer International Publishing, Cham (2018)
5. Broersen, J.: Deontic epistemic stit logic distinguishing modes of mens rea. Journal of Applied Logic 9(2), pp.137–152. Elsevier (2011)
6. Ciabattoni, A., Lyon, T., Ramanayake, R.: From display to labelled proofs for tense logics. In: Editor, Artemov, S., Nerode, A. (eds.) Logical Foundations of Computer Science 2018, LNCS, vol. 10703, pp. 120–139. Springer, Cham (2018)
7. Gentzen, G.: Untersuchungen über das logische Schließen. Mathematische Zeitschrift 39 (3), pp.405–431. Springer-Verlag (1935)
8. Girlando M., Lellmann B., Olivetti N., Pozzato G.L., Vitalis Q.: VINTE: An Implementation of Internal Calculi for Lewis Logics of Counterfactual Reasoning. In: Schmidt, R., Nalon, C. (eds.) TABLEAUX 2017. LNCS, vol.10501, pp.149–159. Springer, Cham (2017)
9. Grossi, D., Lorini, E., Schwarzentruber, F.: The ceteris paribus structure of logics of game forms. Journal of Artificial Intelligence Research (53), pp.91–126 (2015)
10. Herzig, A., Schwarzentruber, F.: Properties of logics of individual and group agency. In: Advances in Modal Logic (7), pp. 133–149. College Publications (2008)
11. Horty, J.: Agency and Deontic Logic. Oxford University Press (2001)

12. Lorini, E., Sartor, G.: Influence and responsibility: A logical analysis. In: Legal Knowledge and Information Systems, pp. 51–60. IOS Press (2015)
13. Murakami, Y.: Utilitarian deontic logic. In: Advances in Modal Logic (5), pp. 211–230. King's College Publications (2005)
14. Negri, S.: Proof analysis in modal logic. Journal of Philosophical Logic 34(5-6), pp. 507–544. Kluwer Academic Publishers (2005)
15. Schwarzentruber, F.: Complexity results of stit fragments. Studia logica, Vol. 100.5, pp.1001–1045 (2012)
16. Sipser, M.: Introduction to the Theory of Computation. Course Technology (2006)
17. Tiu, A., Ianovski, E., Goré, R.: Grammar Logics in Nested Sequent Calculus: Proof Theory and Decision Procedures. CoRR (2012)
18. Viganò, L.: Labelled non-classical logics. Kluwer Academic Publishers (2000)
19. Wansing, H.: Tableaux for multi-agent deliberative-stit logic. In: Advances in Modal Logic (6), pp.503–520. College Publications (2006)
20. Xu, M.: Decidability of deliberative stit theories with multiple agents. In: Gabbay, D. and Ohlbach, H. (eds.) ICTL 1994, vol. 827, pp. 332–348. Springer (1994)

Appendix

A Proofs

Theorem 2 *The* $\mathsf{G3Ldm}_n^m$ *calculi have the following properties:*

1. *All sequents of the form* $\mathcal{R}, w : \phi, w : \overline{\phi}, \Gamma$ *are derivable;*
2. *Variable-substitution is height-preserving admissible;*
3. *All inference rules are height-preserving invertible;*
4. *The weakening rule* (wk) *and two contraction rules* (ctr) *(below) are height-preserving admissible;*

$$\frac{\mathcal{R}, \Gamma}{\mathcal{R}, \mathcal{R}', \Gamma', \Gamma}\ (\mathsf{wk}) \quad \frac{\mathcal{R}, \mathcal{R}', \mathcal{R}', \Gamma}{\mathcal{R}, \mathcal{R}', \Gamma}\ (\mathsf{ctr})_{\mathsf{R}} \quad \frac{\mathcal{R}, \Gamma', \Gamma', \Gamma}{\mathcal{R}, \Gamma', \Gamma}\ (\mathsf{ctr})_{\mathsf{F}}$$

5. *The cut rule* (cut) *(below) is admissible;*

$$\frac{\mathcal{R}, x : \phi, \Gamma \qquad \mathcal{R}, x : \overline{\phi}, \Gamma}{\mathcal{R}, \Gamma}\ (\mathsf{cut})$$

6. *For every formula* $\phi \in \mathcal{L}$, $\vdash w : \phi$ *is derivable in* $\mathsf{G3Ldm}_n^m$ *if and only if* $\vdash_{\mathsf{Ldm}_n^m} \phi$, *i.e.* $\mathsf{G3Ldm}_n^m$ *is sound and complete relative to* Ldm_n^m.

Proof. Clause (1) is proven by induction on the complexity of the formula ϕ, whereas clauses (2)-(4) are proven by induction on the height of the given derivation. Moreover, clause (5) is proven by induction on the complexity of the cut formula ϕ with a subinduction on the sum of the heights of the premises of the (cut) rule. All proofs are similar to proofs of the same properties for modal calculi presented in [14].

Concerning clause (5), the proof of soundness is argued along the same lines as in [**?**, Sec. 5.1] and uses Def. 7. Completeness is proven by showing that all axioms of Ldm_n^m can be derived and that if the premise of an inference rule is derivable, then so is the conclusion. All proofs are fairly simple, with the exception of the derivation of the (IOA) and (APC_n^i) axioms, which we provide below.

(IOA):

$$\begin{array}{c}
\Pi_1 \qquad \cdots \qquad \Pi_m \\ \hline
\mathcal{R}_1 y_1 v, \ldots, \mathcal{R}_m y_m v, x : \Box\langle 1\rangle\overline{\phi}_1, \ldots, x : \Box\langle m\rangle\overline{\phi}_m, y_1 : \langle 1\rangle\overline{\phi}_1, \ldots, y_m : \langle m\rangle\overline{\phi}_m, x : \Diamond([1]\phi_1 \wedge \cdots \wedge [m]\phi_m), v : [1]\phi_1 \wedge \cdots \wedge [m]\phi_m \\ \hline
\mathcal{R}_1 y_1 v, \ldots, \mathcal{R}_m y_m v, x : \Box\langle 1\rangle\overline{\phi}_1, \ldots, x : \Box\langle m\rangle\overline{\phi}_m, y_1 : \langle 1\rangle\overline{\phi}_1, \ldots, y_m : \langle m\rangle\overline{\phi}_m, x : \Diamond([1]\phi_1 \wedge \cdots \wedge [m]\phi_m) \\ \hline
x : \Box\langle 1\rangle\overline{\phi}_1, \ldots, x : \Box\langle m\rangle\overline{\phi}_m, y_1 : \langle 1\rangle\overline{\phi}_1, \ldots, y_m : \langle m\rangle\overline{\phi}_m, x : \Diamond([1]\phi_1 \wedge \cdots \wedge [m]\phi_m) \\ \hline
x : \Box\langle 1\rangle\overline{\phi}_1, \ldots, x : \Box\langle m\rangle\overline{\phi}_m, x : \Diamond([1]\phi_1 \wedge \cdots \wedge [m]\phi_m) \\ \hline
x : \Box\langle 1\rangle\overline{\phi}_1 \vee \cdots \vee \Box\langle m\rangle\overline{\phi}_m \vee \Diamond([1]\phi_1 \wedge \cdots \wedge [m]\phi_m)
\end{array}$$

with Π_i (for $1 \leq i \leq m$) given by:

$$\begin{array}{c}
\mathcal{R}_i vu, \mathcal{R}_i y_i v, \mathcal{R}_i y_i u, ..., y_i : \langle i\rangle\overline{\phi}_i, u : \overline{\phi}_i, u : \phi_i \\ \hline
\mathcal{R}_i vu, \mathcal{R}_i y_i v, \mathcal{R}_i y_i y_i, \mathcal{R}_i v y_i, \mathcal{R}_i y_i u, ..., y_i : \langle i\rangle\overline{\phi}_i, u : \phi_i \\ \hline
\mathcal{R}_i vu, \mathcal{R}_i y_i v, \mathcal{R}_i y_i y_i, \mathcal{R}_i v y_i, ..., y_i : \langle i\rangle\overline{\phi}_i, u : \phi_i \\ \hline
\mathcal{R}_i vu, \mathcal{R}_i y_i v, \mathcal{R}_i y_i y_i, ..., y_i : \langle i\rangle\overline{\phi}_i, u : \phi_i \\ \hline
\mathcal{R}_i vu, \mathcal{R}_i y_i v, ..., y_i : \langle i\rangle\overline{\phi}_i, u : \phi_i \\ \hline
\mathcal{R}_i y_i v, ..., y_i : \langle i\rangle\overline{\phi}_i, v : [i]\phi_i
\end{array}$$

(APC_n^i):

$$\begin{array}{c}
\left\{ \Pi_{k,j} \;\middle|\; 0 \leq k \leq n-1, \, k+1 \leq j \leq n \right\} \\ \hline
w_0 : \Box\langle i\rangle\overline{\phi_1}, w_1 : \overline{\phi_1}, w_0 : \Box(\phi_1 \vee \langle i\rangle\overline{\phi_2}), w_2 : \phi_1, w_2 : \langle i\rangle\overline{\phi_2}, \ldots, w_0 : \Box(\phi_1 \vee \cdots \phi_{n-1} \vee \langle i\rangle\overline{\phi_n}), w_n : \phi_1, \ldots, w_n : \phi_{n-1}, w_n : \langle i\rangle\overline{\phi_n}, w_0 : \phi_1, \ldots, w_0 : \phi_n \\ \hline
w_0 : \Box\langle i\rangle\overline{\phi_1}, w_0 : \Box(\phi_1 \vee \langle i\rangle\overline{\phi_2}), \ldots, w_0 : \Box(\phi_1 \vee \cdots \phi_{n-1} \vee \langle i\rangle\overline{\phi_n}), w_0 : \phi_1, \ldots, w_0 : \phi_n \\ \hline
w_0 : \Box\langle i\rangle\overline{\phi_1} \vee \Box(\phi_1 \vee \langle i\rangle\overline{\phi_2}) \vee \cdots \vee \Box(\phi_1 \vee \cdots \phi_{n-1} \vee \langle i\rangle\overline{\phi_n}) \vee \phi_1 \vee \cdots \vee \phi_n
\end{array}$$

The $\Pi_{0,j}$ (for $1 \leq j \leq n$) derivations are shown below left, and the $\Pi_{k,j}$ (for $0 < k \leq n-1$ and $k+1 \leq j \leq n$) derivations are shown below right.

$$\begin{array}{c}
\mathcal{R}_i w_0 w_j, \mathcal{R}_i w_j w_0, , \ldots, w_0 : \phi_j, w_j : \langle i\rangle\overline{\phi_j}, w_0 : \overline{\phi_j} \\ \hline
\mathcal{R}_i w_0 w_j, \mathcal{R}_i w_j w_0, \ldots, w_0 : \phi_j, w_j : \langle i\rangle\overline{\phi_j} \\ \hdashline
\mathcal{R}_i w_0 w_j, \ldots, w_0 : \phi_j, w_j : \langle i\rangle\overline{\phi_j}
\end{array}
\qquad
\begin{array}{c}
\mathcal{R}_i w_k w_j, \ldots, w_k : \langle i\rangle\overline{\phi_k}, w_j : \overline{\phi_k}, w_j : \phi_k \\ \hline
\mathcal{R}_i w_k w_j, \ldots, w_k : \langle i\rangle\overline{\phi_k}, w_j : \phi_k
\end{array}$$

Note that the dashed lines in the above derivations represent applications of the (sym_i) rule (below left); however, we may apply this rule since it is admissible in $\mathsf{G3Ldm}_n^m$ (as shown below right).

$$\dfrac{\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i uw, \Gamma}{\mathcal{R}, \mathcal{R}_i wu, \Gamma} \, (\mathsf{sym}_i)
\qquad
\begin{array}{c}
\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i uw, \Gamma \\ \hline
\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i uu, \mathcal{R}_i uw, \Gamma \\ \hline
\mathcal{R}, \mathcal{R}_i wu, \mathcal{R}_i ww, \Gamma \\ \hline
\mathcal{R}, \mathcal{R}_i wu, \Gamma
\end{array}
\begin{array}{l}
\text{Lem. 2, Clause (4)} \\
(\mathsf{eucl}_i) \\
(\mathsf{refl}_i)
\end{array}$$

Lemma 1 *The calculus* $\mathsf{G3Ldm}_n^m + \mathsf{PR}$ *has the following properties: (i) All sequents of the form* $\mathcal{R}, w : \phi, w : \overline{\phi}, \Gamma$ *are derivable, (ii) Variable-substitution is height-preserving admissible, (iii) All inference rules are height-preserving invertible, (iv) The* (wk), $(\mathsf{ctr})_\mathsf{R}$ *and* $(\mathsf{ctr})_\mathsf{F}$ *rules are height-preserving admissible.*

Proof. (i) is proved by induction on the complexity of ϕ and (ii)-(iv) are shown by induction on the height of the given derivation. All proofs are routine, so we only prove the most significant result: height-preserving admissibility of contraction.

We proceed by induction on the height of the given derivation. With the exception of (Pr_i), the proof is exactly the same as for Thm. 2 clause 4 (see [14,**?**] for details). Hence, we only prove the (Pr_i) case in the inductive step.

First, we show that $(\mathsf{ctr})_\mathsf{F}$ can be permuted with (Pr_i). There are two cases: either the derivation performs a formula contraction solely in the formula-context Γ (below left), or the derivation performs a contraction with the auxiliary formula $w : \langle i \rangle \phi$ (below right) (potentially performing a contraction in Γ):

$$\dfrac{\dfrac{\mathcal{R}, w : \langle i \rangle \phi, u : \phi, \Gamma}{\mathcal{R}, w : \langle i \rangle \phi, \Gamma}\,(\mathsf{Pr}_i)}{\mathcal{R}, w : \langle i \rangle \phi, \Gamma'}\,(\mathsf{ctr})_\mathsf{F} \qquad \dfrac{\dfrac{\mathcal{R}, w : \langle i \rangle \phi, w : \langle i \rangle \phi, u : \phi, \Gamma}{\mathcal{R}, w : \langle i \rangle \phi, w : \langle i \rangle \phi, \Gamma}\,(\mathsf{Pr}_i)}{\mathcal{R}, w : \langle i \rangle \phi, \Gamma'}\,(\mathsf{ctr})_\mathsf{F}$$

In both cases, any sequence of transitions between w and u will be preserved, meaning that the $(\mathsf{ctr})_\mathsf{F}$ rule may be freely permuted with the (Pr_i) rule without affecting the side condition of the propagation rule.

Secondly, we consider the $(\mathsf{ctr})_\mathsf{R}$ case, which has the following form:

$$\dfrac{\dfrac{\mathcal{R}, w : \langle i \rangle \phi, u : \phi, \Gamma}{\mathcal{R}, w : \langle i \rangle \phi, \Gamma}\,(\mathsf{Pr}_i)}{\mathcal{R}', w : \langle i \rangle \phi, \Gamma}\,(\mathsf{ctr})_\mathsf{R}$$

Since the (Pr_i) rule is applied first, we know there exists a sequence of transitions $w \xrightarrow{\langle i \rangle} v_1 \xrightarrow{\langle i \rangle} \cdots v_n \xrightarrow{\langle i \rangle} u$ from w to u. Notice that $(\mathsf{ctr})_\mathsf{R}$ contracts identical relational atoms in $\mathcal{R}$, resulting in a single copy still present in $\mathcal{R}'$. Hence, if we apply $(\mathsf{ctr})_\mathsf{R}$ first, we may still apply (Pr_i) afterwards, since the same sequence of transitions from w to u remains present in R'. See [**?**, Lem. 6.12] for further details on the preservation of transitions under contractions.

Lemma 2 ((refl_i)-Elimination) *Every sequent Λ derivable in $\mathsf{G3Ldm}_n^m + \mathsf{PR}$ is derivable without the use of (refl_i).*

Proof. The result is proven by induction on the height of the given derivation. We assume that (refl) is used once as the last inference in the given derivation. The general result follows by successively eliminating topmost occurrences of (refl) rule instances.

Base Case. An application of (refl_i) on an initial sequent (below left), where possibly $w = u$, can be re-written as an instance of the (id) rule (below right):

$$\dfrac{\mathcal{R}, \mathcal{R}_i ww, u : p, u : \overline{p}, \Gamma}{\mathcal{R}, u : p, u : \overline{p}, \Gamma}\,(\mathsf{refl}_i) \qquad \dfrac{}{\mathcal{R}, u : p, u : \overline{p}, \Gamma}\,(\mathsf{id})$$

Inductive step. We show the inductive step for the (IOA) and (APC_n^i) rules, as well as for the non-trivial $(\langle i \rangle)$, (eucl_i), and (Pr_i) cases. All other cases are resolved by applying IH to the premise followed by an application of the corresponding rule.

(i) The (refl_i) rule may be freely permuted with the (IOA) rule:

$$\dfrac{\dfrac{\mathcal{R}, \mathcal{R}_i ww, \mathcal{R}_1 u_1 v, ..., \mathcal{R}_n u_n v, \Gamma}{\mathcal{R}, \mathcal{R}_i ww, \Gamma}\ (\mathsf{IOA})}{\mathcal{R}, \Gamma}\ (\mathsf{refl}_i) \qquad \dfrac{\dfrac{\mathcal{R}, \mathcal{R}_i ww, \mathcal{R}_1 u_1 v, ..., \mathcal{R}_n u_n v, \Gamma}{\mathcal{R}, \mathcal{R}_1 u_1 v, ..., \mathcal{R}_n u_n v, \Gamma}\ (\mathsf{refl}_i)}{\mathcal{R}, \Gamma}\ (\mathsf{IOA})$$

(ii) We may easily permute the (refl_i) rule with the (APC^i_n) rule:

$$\dfrac{\dfrac{\Big\{\mathcal{R}, \mathcal{R}_i uu, \mathcal{R}_i w_k w_j, \Gamma \;\Big|\; 0 \leq k \leq n-1,\, k+1 \leq j \leq n\Big\}}{\mathcal{R}, \mathcal{R}_i uu, \Gamma}\ (\mathsf{APC}^i_n)}{\mathcal{R}, \Gamma}\ (\mathsf{refl}_i)$$

$$\dfrac{\dfrac{\Big\{\mathcal{R}, \mathcal{R}_i uu, \mathcal{R}_i w_k w_j, \Gamma \;\Big|\; 0 \leq k \leq n-1,\, k+1 \leq j \leq n\Big\}}{\Big\{\mathcal{R}, \mathcal{R}_i w_k w_j, \Gamma \;\Big|\; 0 \leq k \leq n-1,\, k+1 \leq j \leq n\Big\}}\ (\mathsf{refl}_i) \times \tfrac{n(n+1)}{2}}{\mathcal{R}, \Gamma}\ (\mathsf{APC}^i_n)$$

(iii) In the case of $(\langle i \rangle)$, when applying the (refl_i) rule to the first premise of the left derivation instead, the propagation rule (Pr_i) may be applied to the resulting sequent $\Lambda = \mathcal{R}, w : \langle i \rangle \phi, w : \phi, \Gamma$ since the empty string $\varepsilon \in \mathcal{P}_\Lambda(w, w) \cap L_i$:

$$\dfrac{\dfrac{\mathcal{R}, \mathcal{R}_i ww, w : \langle i \rangle \phi, w : \phi, \Gamma}{\mathcal{R}, \mathcal{R}_i ww, w : \langle i \rangle \phi, \Gamma}\ (\langle i \rangle)}{\mathcal{R}, w : \langle i \rangle \phi, \Gamma}\ (\mathsf{refl}_i) \qquad \dfrac{\dfrac{\mathcal{R}, \mathcal{R}_i ww, w : \langle i \rangle \phi, w : \phi, \Gamma}{\mathcal{R}, w : \langle i \rangle \phi, w : \phi, \Gamma}\ (\mathsf{refl}_i)}{\mathcal{R}, w : \langle i \rangle \phi, \Gamma}\ (\mathsf{Pr}_i)$$

(iv) The non-trivial case of permuting the (refl_i) rule with the (eucl_i) rule is shown below. The case may be resolved by leveraging admissibility of contraction. Note that dashed lines (below right) have been used to represent an application of height-preserving admissibility of contraction (Lem. 1), the use of which decreases the height of the derivation by 1. As a consequence, the height of the (refl_i) rule application also decreases by 1. The two cases are accordingly:

$$\dfrac{\dfrac{\mathcal{R}, \mathcal{R}_i ww, \mathcal{R}_i ww, \mathcal{R}_i ww, \Gamma}{\mathcal{R}, \mathcal{R}_i ww, \mathcal{R}_i ww, \Gamma}\ (\mathsf{eucl}_i)}{\mathcal{R}, \mathcal{R}_i ww, \Gamma}\ (\mathsf{refl}_i) \qquad \dfrac{\mathcal{R}, \mathcal{R}_i ww, \mathcal{R}_i ww, \mathcal{R}_i ww, \Gamma}{\mathcal{R}, \mathcal{R}_i ww, \Gamma}\ \text{Lem. 1}$$

$$\dfrac{\dfrac{\mathcal{R}, \mathcal{R}_i ww, \mathcal{R}_i wu, \mathcal{R}_i wu, \Gamma}{\mathcal{R}, \mathcal{R}_i ww, \mathcal{R}_i wu, \Gamma}\ (\mathsf{eucl}_i)}{\mathcal{R}, \mathcal{R}_i wu, \Gamma}\ (\mathsf{refl}_i) \qquad \dfrac{\dfrac{\mathcal{R}, \mathcal{R}_i ww, \mathcal{R}_i wu, \mathcal{R}_i wu, \Gamma}{\mathcal{R}, \mathcal{R}_i ww, \mathcal{R}_i wu, \Gamma}\ \text{Lem. 1}}{\mathcal{R}, \mathcal{R}_i wu, \Gamma}\ (\mathsf{refl}_i)$$

(v) Last, the non-trivial case of permuting (refl_i) over (Pr_i) occurs when a relational atom of the form $\mathcal{R}_i ww$ is auxiliary in (Pr_i). We therefore assume that in the first premise $\Lambda_1 = \mathcal{R}, \mathcal{R}_i ww, x : \langle i \rangle \phi, y : \phi, \Gamma$ of the initial derivation (below left), the propagation rule is applied because there exists some string $\langle i\rangle^n \in \mathcal{P}_{\Lambda_1}(x,y) \cap L_i$ with $n \neq 0 \in \mathbb{N}$ and there exists a sequence of transitions from x to y containing transitions of the form $w \xrightarrow{\langle i\rangle} w$ (defined relative to $\mathcal{R}_i ww$). If instead we apply the (refl_i) rule first (below right), then we obtain the sequent $\Lambda_2 = \mathcal{R}, x : \langle i\rangle\phi, y : \phi, \Gamma$ which no longer contains the relational atom $\mathcal{R}_i ww$ that was used to apply the propagation rule in the initial derivation (left). Nevertheless, since $\langle i\rangle^n \in \mathcal{P}_{\Lambda_1}(x,y)$, there exists a sequence of transitions $x \xrightarrow{\langle i\rangle} z_1 \xrightarrow{\langle i\rangle} \cdots z_{n-1} \xrightarrow{\langle i\rangle} y$ containing transitions of the form $w \xrightarrow{\langle i\rangle} w$. If we delete all transitions of the form $w \xrightarrow{\langle i\rangle} w$ from the sequence of transitions, we will still have a valid sequence of transitions between x and y. That is, if $\cdots z_i \xrightarrow{\langle i\rangle} w \xrightarrow{\langle i\rangle} w \xrightarrow{\langle i\rangle} z_j \cdots$ occurs in our sequence, then clearly $\cdots z_i \xrightarrow{\langle i\rangle} w \xrightarrow{\langle i\rangle} z_j \cdots$ is a valid sequence.[4] Moreover, the resulting string $\langle i\rangle^k$ with $k < n$ will be in L_i. Therefore, since there exists some string $\langle i\rangle^k \in \mathcal{P}_{\Lambda_2}(x,y) \cap L_i$, the propagation rule may still be applied to Λ_2.

$$\dfrac{\dfrac{\mathcal{R}, \mathcal{R}_i ww, x : \langle i\rangle\phi, y : \phi, \Gamma}{\mathcal{R}, \mathcal{R}_i ww, x : \langle i\rangle\phi, \Gamma}\,(\mathsf{Pr}_i)}{\mathcal{R}, x : \langle i\rangle\phi, \Gamma}\,(\mathsf{refl}_i) \qquad \dfrac{\dfrac{\mathcal{R}, \mathcal{R}_i ww, x : \langle i\rangle\phi, y : \phi, \Gamma}{\mathcal{R}, x : \langle i\rangle\phi, y : \phi, \Gamma}\,(\mathsf{refl}_i)}{\mathcal{R}, x : \langle i\rangle\phi, \Gamma}\,(\mathsf{Pr}_i)$$

Theorem 4 (Soundness of $\mathsf{Ldm}_n^m\mathsf{L}$) *Every sequent derivable in $\mathsf{Ldm}_n^m\mathsf{L}$ is valid.*

Proof. We know by Thm. 2 that all rules of $\mathsf{Ldm}_n^m\mathsf{L}$, with the exception of (Pr_i), preserve validity. We therefore only consider the (Pr_i) case and argue by contraposition that if the conclusion of the rule is invalid, then so is the premise. Consider the following inference:

$$\dfrac{\mathcal{R}, w : \langle i\rangle\phi, u : \phi, \Gamma}{\mathcal{R}, w : \langle i\rangle\phi, \Gamma}\,(\mathsf{Pr}_i)$$

Let $\Lambda_1 = \mathcal{R}, w : \langle i\rangle\phi, u : \phi, \Gamma$ and $\Lambda_2 = \mathcal{R}, w : \langle i\rangle\phi, \Gamma$. Assume that there exists a model M with interpretation I such that $\mathcal{R}, w : \langle i\rangle\phi, \Gamma$ is not satisfied in M with I. In other words, $\mathcal{R}$ holds in M with I, but for all labelled formulae $v : \psi$ in $w : \langle i\rangle\phi, \Gamma$ the following holds: $M, v^I \not\Vdash \psi$.

We additionally assume that the side condition holds; that is, there exists some string $\omega \in \mathcal{P}_{\Lambda_1}(w,u) \cap L_i$. Since $\omega \in L_i$, we know that ω is of the form $\langle i\rangle^n$ with $n \in \mathbb{N}$. Additionally, since $\omega = \langle i\rangle^n \in \mathcal{P}_{\Lambda_1}(w,u)$, this means that there exists a sequence of relational atoms of the form $\mathcal{R}_i wz_1, \cdots, \mathcal{R}_i z_{n-1}u$ or $\mathcal{R}_i uz_1, \cdots, \mathcal{R}_i z_{n-1}w$ connecting w and u. Therefore, we have that $(w^I, z_1^I), \cdots, (z_{n-1}^I, u^I) \in \mathcal{R}_i$ or $(u^I, z_1^I), \cdots, (z_{n-1}^I, w^I) \in \mathcal{R}_i$ in M under I, which implies that w and u are mapped to worlds in the same equivalence class, *i.e.*, $\mathcal{R}_i w^I u^I$ holds without loss of generality. By assumption, $M, w^I \not\Vdash \langle i\rangle\phi$ holds, meaning

[4] Note that in the marginal case where our sequence of transitions is of the form $w \xrightarrow{\langle i\rangle} w \xrightarrow{\langle i\rangle} \cdots w \xrightarrow{\langle i\rangle} w$, with all states (including x and y) equal to w, the corresponding propagation automaton is $\mathcal{P}_{\Lambda_2}(w,w)$, which accepts the empty string $\varepsilon \in L_i$, thus allowing for the desired application of (Pr_i).

that $M, w^I \Vdash [i]\overline{\phi}$ holds. Since both $R_i w^I u^I$ and $M, w^I \Vdash [i]\overline{\phi}$ hold, we know that $M, u^I \Vdash \overline{\phi}$, implying that $M, u^I \not\Vdash \phi$. Thus, the premise is falsified by M under I.

Lemma 4 *Every labelled sequent generated throughout the course of computing* $\mathtt{Prove_n}(w : \phi)$ *is forestlike.*

Proof. We prove the result by induction on the number of instructions executed. Note that the input sequent $w : \phi$ is trivially forestlike.

Base Case. We assume that one of the instructions (2 - 7) has been executed in the algorithm (initially, instructions 1 and 8 cannot be executed): (2) If $w : \phi$ is stable, then no sequent is generated. (3) If instruction 3 is executed, then in case (i) $w : \phi = w : \psi \vee \chi$, so the generated sequent is $w : \phi, w : \psi, w : \chi$, which is forestlike; in case (ii) $w : \phi = w : \psi \wedge \chi$, so the sequents $w : \phi, w : \psi$ and $w : \phi, w : \chi$ are generated, which are both forestlike. (4) If instruction 4 is executed, then $w : \phi = w : \langle 1 \rangle \psi$, so the sequent generated is of the form $w : \phi, w : \psi$, which is forestlike. (5) If instruction 5 is executed, then $w : \phi = w : \Diamond \psi$, so the sequent generated is of the form $w : \phi, w : \psi$, which is forestlike. (6) If instruction 6 is executed, then $w : \phi = w : [1]\psi$, and the sequent $\mathcal{R}_1 wv, w_\phi, v : \psi$ is generated, which is forestlike. (7) If instruction 7 is executed, then $w : \phi = w : \Box \psi$, so the sequent $w : \phi, v : \psi$ is generated, which is forestlike.

Inductive Step. We assume that our input sequent is forestlike, and argue that the generated sequent is forestlike: (1) If instruction 1 is executed, then no sequent is generated. (2) If instruction 2 is executed, then no sequent is generated. (3) Each of the operations in instruction 3 preserves the set of relational atoms $\mathcal{R}$ as well as the set of labels in the sequent; hence, the sequent generated after an execution of instruction 3 will be forestlike. (4) Instruction 4 labels nodes in the graph of the input sequent with additional formulae, but does not change the structure of the graph; therefore, the generated (output) sequent will be forestlike. (5) Similar to case 4. (6) Instruction 6 adds one additional $\mathcal{R}_1$ edge to a fresh labelled node v in the graph of the input sequent, which adds additional branching in the graph of the generated sequent, and thus preserves the forestlike structure of the sequent. (7) Instruction 7 adds a new labelled formula to the sequent, which is akin to adding a new, disjoint labelled node to the graph of the input sequent; this preserves the forestlike structure of the sequent. (8) Instruction 8 connects the root of one choice-tree to the root of another choice-tree in the graph of the input sequent; the result is another tree, and thus, the generated sequent will be forestlike.

Theorem 6 (Termination) *For every labelled formula* $w : \phi$*,* $\mathtt{Prove_n}(\mathtt{w} : \phi)$ *terminates.*

Proof. Let *sufo*(ϕ) be the multiset of subformulae of ϕ defined in the usual way. Observe that new labels are only created through instructions 6 and 7 of the algorithm, and each time instruction 6 or 7 executes, the formula $w : [1]\psi$ or $w : \Box \psi$ (resp.) responsible for the instruction's execution, no longer influences the non-[1]-realization or non-$\Box$-realization of w. Therefore, the number of labels in any sequent generated by the algorithm is bounded by the number of [1]- and $\Box$-formulae contained in $sufo(\phi)$ plus 1 (which is the label of the input formula). Moreover, the number of $\mathcal{R}_1$ relational atoms is bounded by the number of [1]-formulae.

Instructions 3–5 add strict subformulae of formulae occurring in $sufo(\phi)$ and do not create new labels or relational atoms. Due to the blocking conditions in instructions 3–5, any formula that is added with a label will only be added once. Therefore, the number of executions of instructions 3–5 is bounded by $|sufo(\phi)|$ multiplied by 1 plus the number of [1]- and $\Box$-formulae occurring in $sufo(\phi)$.

Last, observe that instruction 6 increases the breadth or height of a choice-tree in the input sequent, whereas instruction 7 adds a new label which acts as the root of a new choice-tree. This implies that the number of choice-trees is bounded by the number of $\Box$-formulae occurring in $sufo(\phi)$. Each time instruction 8 executes the number of choice-trees in a resulting sequent decreases by 1. Since the number of choice-trees is bounded by the number of $\Box$-subformulae, eventually the number of choice-trees will decrease to $k \leq n$, at which point, the generated sequent will be n-choice consistent, and instruction 8 will no longer be applicable.

Therefore, the algorithm will terminate.