Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Strongly Analytic Calculi for KLM Logics with SMT-Based Prover

Agata Ciabattoni, **Clemens Eisenhofer**, **Dmitry Rozplokhas**
TU Wien, Austria
{agata,dmitry}@logic.at, clemens.eisenhofer@tuwien.ac.at

Abstract

We introduce modular calculi for the logics for nonmonotonic reasoning defined by Kraus, Lehmann, and Magidor, featuring a strengthened form of analyticity. Our calculi are used to determine the computational complexity for the logics **C**, **CL**, **CM**, **P** (and **M**), and fragments thereof. The calculi are encoded into SMT solvers, yielding an efficient prover with countermodel generation capabilities. Our work encompasses known results and introduces new findings, including co-NP-completeness and a more effective semantics for **C**.

1 Introduction

Non-monotonic reasoning plays a pivotal role in artificial intelligence. (Kraus, Lehmann, and Magidor 1990) introduced an inferential framework for its formalization, which has since become foundational. The resulting logics, known as KLM logics, contain a conditional operator (expressing "... typically implies...") defined by inference rules, along with corresponding preference-model semantics. From the weakest to the strongest system, these are: the logic **C** of cumulative reasoning, loop-cumulative logic **CL**, and preferential logic **P**. For sake of completeness, the KLM logics also include the monotonic systems **CM** and **M**; **CM** later appeared in the context of normative reasoning as the logic out_3^+ of "reusable throughput" (Makinson and van der Torre 2001), while **M** collapses to classical logic.

Although the KLM logics originated as an inferential approach to non-monotonic reasoning, the development of proof calculi and provers for them has encountered significant challenges, leaving various problems unresolved.

In this paper we introduce uniform and modular sequent calculi for **C**, **CL**, **CM**, **P** (and **M**), leveraging them to provide *uniform* complexity results, and efficient reasoning tools. Our work encompasses known results and novel contributions, with particular emphasis on **C**. This base logic can be viewed as **P** without the rule (*Or*), expressing disjunctive reasoning. **P** was proposed as the central system in (Kraus, Lehmann, and Magidor 1990) and has served as the foundation for various systems of non-monotonic reasoning ever since. However, there are scenarios where the conditional operator is not intended to fulfill (*Or*). These include the epistemic interpretation in (Kraus, Lehmann, and Magidor 1990), in legal reasoning contexts, when formalizing a "count as" connection (Gelati et al. 2004; Governatori and Rotolo 2008), in the actual causality framework of (Bochman 2018), or when efficiency of fragments matter (see, e.g., Th 17 in the current paper).

The logic **C** of cumulative inference originates before KLM framework, with its axioms introduced by (Gabbay 1985) as "the minimal conditions a consequence relation should satisfy to represent a bona fide nonmonotonic logic" (Kraus, Lehmann, and Magidor 1990). Despite this, **C** remains the least understood KLM logic, with many open questions, including its exact complexity. One reason may be the challenges posed by the *smoothness condition*[1] in its semantic characterization. To tackle this, the original KLM paper proposed a modified semantics, and introduced **CL**, a strengthened logic incorporating an additional rule.

Analytic sequent (or tableau) calculi whose derivations contain only subformulas of the formulas to be proved are powerful tools to establish meta-logical results about the formalized logics (e.g. complexity), and a base for automated theorem provers. While the original KLM rules offer useful foundational principles for categorizing variations of conditional entailment, they cannot be directly employed for automated deduction or to prove complexity results, being a sort-of natural deduction systems. In particular, their derivations lack analyticity and involve additional formulas not in the original query, with no guidance for their construction. A variety of analytic calculi were introduced for the primary KLM logic **P** and its extensions. Some of these calculi were constructed directly based on the semantical characterization (e.g. (Giordano et al. 2009; Giordano et al. 2007; Britz and Varzinczak 2018), while others rely on embeddings of **P** into normal modal or conditional logics well-suited for analytic characterization (e.g. (Boutilier 1990; Britz, Heidema, and Labuschagne 2009)). Most of these approaches however are not extendable to weaker KLM logics. To the best of our knowledge the only analytic calculi for KLM logics weaker than **P** are the tableau calculi in (Artosi, Governatori, and Rotolo 2002) for **C** and **CL**, tableau calculi in (Giordano et al. 2009) for **C**, **CL** and **P**, and nested sequent calculi (Alenda, Olivetti, and Pozzato 2016) for some conditional logics including **C**. These calculi introduce syn-

[1]Related to the limit assumption in (Lewis 1973), smoothness ensures that each truth set in the preference model has a minimal state.

tactic rules for decomposing formulas that reflect the semantical definition of satisfaction in preference models. They inherit the drawback of the semantic characterizations, namely the difficulty to ensure smoothness (when it cannot be replaced by stronger frame properties, as for **P** or **CL** (Giordano et al. 2009)), and lead to relatively slow implementations as automated theorem provers (see the evaluation in Sec. 5.2).

In this paper we propose an alternative approach. We introduce modular sequent calculi for the logics **C**, **CL**, **CM**, **P** (and **M**) that enjoy a stronger form of analyticity. Their derivations indeed contain conditionals built from *formulas* already contained in the sequent to be proved (i.e., the entailment statement), without decomposing the formulas into subformulas. Viewing our derivations as directed graphs – whose nodes represent the formulas contained in the end sequent and edges the conditionals between them – the applications of rules add only edges (and not nodes) to the graphs.

A semantic counterpart of this stronger analyticity condition is also introduced; it turns out to be a weaker form of smoothness that only requires verification for the formulas present in the entailment (and not for all formulas in the language). This constitutes an alternative, and more manageable semantics for **C**, which demonstrates improved computational properties, e.g., polynomial-time model checking.

Our calculi are employed to uniformly establish co-NP-completeness for the entailment problem across all KLM logics. Previously, the exact complexity was only known for **P** (Lehmann and Magidor 1992) and **CL** (Giordano et al. 2009) (as well as **M**); prior upper bounds for **C** were NEXP (Giordano et al. 2009) and PSPACE (Alenda, Olivetti, and Pozzato 2016), and for out_3^+ (i.e. **CM**) it was P^{NP} (Sun and Robaldo 2017). We also show that for Horn and literal conditionals the entailment problem for **C**, **CL**, and **CM** is polynomial. Additionally, for Horn conditionals, we prove that the entailment in **M** and **CM** coincides, as does the entailment in **P** and **CL** (the latter result already shown via semantic methods in (Kraus, Lehmann, and Magidor 1990)).

Unlike analytic calculi that rely on formula decompositions, our calculi are based on saturation. Rule applications only expand sequents (or, when looking at the graph-perspective of derivations, edges are added) as much as possible. This enables a natural encoding into SMT solvers, resulting in an efficient theorem prover for the KLM logics.

2 Preliminaries on KLM logics

(Kraus, Lehmann, and Magidor 1990), henceforth referred to as "the KLM paper", introduced axiomatizations for the binary relation $\mid\!\sim$ (interpreted as "... typically implies ...") along with appropriate semantics, leading to the development of the KLM logics. Henceforth we will use the notation (A, B) (instead of $A \mid\!\sim B$), for the non-monotonic *conditionals*, where A (*condition*) and B (*consequence*) are propositional formulas of classical logic (called *components* of the conditional). We will denote classical entailment from a set of formulas with '$\Rightarrow$' and classical equivalence between two formulas with '$\Leftrightarrow$'. On the syntactic side, the logics of the KLM family are defined via the basic axioms in Figure 1. Different KLM logics are given by different subsets of these rules. The basic logic **C** of cumulative reasoning consists of the rules in Fig. 1(a). The extension of **C** with the rule (*Loop*) is the logic **CL**. The extension of **C** with the rule (*Or*) for reasoning by cases yields the logic **P** of preferential reasoning. Each logic **C**, **CL**, **P** captures different features of non-monotonic reasoning (none of them admits the monotonicity principle (*Mon*)). The KLM family also includes two monotonic logics: the logic **CM** of cumulative monotonic reasoning, which extends **C** with (*Mon*), and the logic of monotonic reasoning **M** which extends **CM** with (*Or*) (**M** collapses to classical logic). Notice that both **P** and **CM** are stronger than **CL**, as they derive the rule (*Loop*).

(a) Rules of basic logic **C**:	(*Id*):	$\vdash (A, A)$
	(*RWk*):	$(A, B) \vdash (A, B')$ when $B \Rightarrow B'$
	(*LEq*):	$(A, B) \vdash (A', B)$ when $A \Leftrightarrow A'$
	(*CMon*):	$(A, B), (A, C) \vdash (A \wedge B, C)$
	(*CCut*):	$(A, B), (A \wedge B, C) \vdash (A, C)$
(b) Additional KLM rules:	(*Loop*):	$(A_1, A_2), \ldots, (A_n, A_1) \vdash (A_1, A_n)$
	(*Or*):	$(A_1, B), (A_2, B) \vdash (A_1 \vee A_2, B)$
	(*Mon*):	$(A, B) \vdash (A', B)$ when $A' \Rightarrow A$

Figure 1: KLM rules

In each KLM logic the entailment of conditionals is defined by iterative applications of these rules (the notion of derivation is as usual, and always involve a finite number of conditionals). Semantically, KLM logics are characterized by a variant of *preference model* semantics (Shoham 1987).

Definition 1. *A* preference model *is a triple* $\langle \mathcal{S}, w, \prec \rangle$ *s.t. (i)* $\mathcal{S}$ *is a set of abstract states, (ii)* w *is a labeling function that maps each state in* $\mathcal{S}$ *to a (possibly empty) set of Boolean assignments of the propositional variables, (iii) the preference relation* $\prec$ *is an asymmetric binary relation on* $\mathcal{S}$.

A conditional (A, B) is read as "B is true in the minimal (most normal) state(s) satisfying A", see below. Note that in the most general case $\prec$ is not assumed to be transitive, so it is not necessarily a partial order. Nevertheless, minimality can be defined for arbitrary asymmetric relations, with the usual distinction between *minimal* and *least* elements.

Definition 2. *Given a set* X *equipped with an asymmetric binary relation* $\prec$*,* $m \in X$ *is a* minimal element of X *if there is no* $x \in X$ *s.t.* $x \prec m$*,* $l \in X$ *is the* least element of X *when* $l \prec x$ *for all* $x \in X$ *s.t.* $x \neq l$*. We denote the set of minimal elements and the (empty or one-element) set of least elements of* X *by* $min(X)$ *and* $least(X)$ *respectively.*

A formula A is satisfied in a state s of a preference model $M = \langle \mathcal{S}, w, \prec \rangle$, if it is satisfied by all Boolean assignments $w(s)$ (in symbols $M, s \models A$); we denote by $\|A\|^M$ the truth set of A, i.e., the subset of states in $\mathcal{S}$ that satisfy A. A conditional (A, B) is satisfied by the model M (in symbols $M \models (A, B)$) if $M, s \models B$ for all $s \in min(\|A\|^M)$.

A truth set of a formula may have multiple minimal states; an extra condition (*smoothness*) on preference models excludes the absence of minimal states for a truth set. Rooted in conditional logic tradition, this condition says that for any formula A and state $s \in \|A\|^M$, either $s \in min(\|A\|^M)$ or there is $s' \prec s$ with $s' \in min(\|A\|^M)$. Smoothness is the defining feature of the basic logic **C**. However, as already noted in (Kraus, Lehmann, and Magidor 1990), working with it can be challenging due to the difficulty of checking this condition for all truth sets.

Additional conditions on preference models characterize the other KLM logics, streamlining their semantics compared to **C**. In particular, **CL** and **P** assume the preference relation to be a (strict) partial order, and for the monotonic KLM logics **CM** and **M** the preference relation is empty. Additionally, for the logics **P** and **M**, the labeling has exactly one Boolean assignment for each state (we call this property *functionality* of a labeling function).

In the following, when discussing a KLM logic $\mathcal{L} \in \{\mathbf{C}, \mathbf{CL}, \mathbf{CM}, \mathbf{P}, \mathbf{M}\}$, we will refer to the models satisfying the respective properties as $\mathcal{L}$-models.

Theorem 1. *(Kraus, Lehmann, and Magidor 1990) A set Γ of conditionals entails a conditional $\mathcal{E}$ in a KLM logic $\mathcal{L}$ (in symbols, $\Gamma \vdash_{\mathcal{L}} \mathcal{E}$) iff every $\mathcal{L}$-model that satisfies all conditionals in Γ satisfies also $\mathcal{E}$.*

Remark 1. *In the KLM paper the authors notice that their canonical models for **C** and **CL** satisfy a stronger version of smoothness (which we refer to as* l-smoothness*) that requires each non-empty truth set to have the least element. Although the same cannot be observed for the other KLM logics, it is easy to see that entailment in **CM** and **M** can be characterized by models with exactly one state (and, hence, l-smooth): since the preference relation is empty, in a countermodel for entailment only the state falsifying the non-entailed conditional is needed. Moreover, (Lehmann and Magidor 1992, Lemma 8) shows that* finite *entailments in **P** can be characterized by finite linearly-ordered (and, hence, l-smooth) preference models. In sum, in the semantical characterizations of finite entailment, smoothness can be replaced by l-smoothness for all KLM logics. As we only deal with finite entailments, we will use both versions of smoothness.*

3 Strongly Analytic Sequent Calculi

We present strongly analytic calculi for the KLM logics. Our calculi manipulate sequents (denoted by $\Gamma \vdash \mathcal{E}$), where Γ is a finite set of conditionals, and $\mathcal{E}$ a conditional. Their rules enable to derive a sequent (*conclusion*) from a number of *premises*, each of which consists of either a sequent or a classical entailment $\Rightarrow$. The rules are presented in Fig. 2 and their intuition is explained in the respective sections dedicated to each logic. The superscript s on the rule's names serves to distinguish them from the corresponding rule (if any) in Fig. 1. All rules follow a specific structure: the sequent(s) in the premise(s) and in the conclusion differ only for the presence in the former of one conditional δ_j on the left hand side (LHS). The rules (Cn^s) and (CnM^s) additionally contain a classical entailment $\mathfrak{S}$ as premise.

Crucially, our rules in Fig. 2 apply only under two restrictions: (1) the conditional δ_j is not already on the LHS of the rule premise (*novelty*), and (2) the components in δ_j are already present among the components of the conditionals in the conclusion (*analyticity*)[2]. Note that our analyticity condition is, in a sense, stronger than the subformula property, which (only) requires that all formulas in the premises of a rule are subformulas of formulas in the conclusion. The graph view of derivations in our calculi, presented later, clarifies this matter.

Our calculi are constructed using a modular approach, akin to the calculi in the KLM paper. The calculus $\mathcal{SC}_{\mathbf{C}}$ for the basic logic **C** consists of the rules (Id^s), $(Equiv^s)$, and (Cn^s). The extension $\mathcal{SC}_{\mathbf{CL}}$ for **CL** includes also the rule $(Loop^s)$. The calculi $\mathcal{SC}_{\mathbf{CM}}$, $\mathcal{SC}_{\mathbf{P}}$ and $\mathcal{SC}_{\mathbf{M}}$ for **CM**, **P**, and **M** are achieved by adding the rules (Tr^s), (CnM^s), and both, respectively. Note that all rules, except (CnM^s), consist of one sequent-premise only.

Definition 3. *A* derivation *in $\mathcal{SC}_{\mathcal{L}}$ is a tree with internal nodes labeled as sequents, and s.t. the label of each node follows from the labels of its children using the calculus rules. A* proof *in $\mathcal{SC}_{\mathcal{L}}$ is a derivation whose leaves are either instances of a concluding axiom or valid classical entailments. A sequent is* derivable in *$\mathcal{SC}_{\mathcal{L}}$ if there is a proof in $\mathcal{SC}_{\mathcal{L}}$ with the root labeled by this sequent.*

In the bottom-up proof search for a sequent $\Gamma \vdash \mathcal{E}$, the set of conditionals on the LHS expands. A branch is closed by the concluding axiom when $\mathcal{E}$ appears on the LHS.

Graphs of conditional dependencies offer a convenient representation of $\mathcal{SC}_{\mathcal{L}}$ derivations and proofs. In this perspective we view the set of formulas from all components of conditionals in $\Gamma \cup \mathcal{E}$ (in symbols, $\mathcal{F}(\Gamma \cup \{\mathcal{E}\})$) as nodes in a graph, and conditionals on the LHS as directed edges between them. Each one-premise rule adds the edge δ_1 to the current graph, while (CnM^s) creates m copies of the graph with one additional edge $\{\delta_j\}_{j=1}^m$ each.

Definition 4 (Graph view)**.** *A derivation in $\mathcal{SC}_{\mathcal{L}}$ of a sequent $\Gamma \vdash \mathcal{E}$ is the process of saturating a graph G (graphs, in case of $\mathcal{SC}_{\mathbf{P}}$ and $\mathcal{SC}_{\mathbf{M}}$) with nodes $V = \mathcal{F}(\Gamma \cup \{\mathcal{E}\})$ and directed edges $E(G) \supseteq \Gamma$. Starting from the graph with Γ as set the of edges, each application of a rule in $\mathcal{SC}_{\mathcal{L}}$ to a graph with edges E replaces this graph with m graphs $\{(V, E \cup \{\delta_j\})\}_{j=1}^m$. A* proof *in $\mathcal{SC}_{\mathcal{L}}$ is a derivation in which an edge corresponding to $\mathcal{E}$ appears in each graph. If this condition remains unmet despite exhaustive rule applications, the derivation is said to be failed.*

The analyticity condition prevents the expansion[3] of nodes from $V = \mathcal{F}(\Gamma \cup \{\mathcal{E}\})$, and the novelty condition prohibits the addition of duplicate directed edges between the same pair of nodes. This results in the process reaching saturation – where no further edges can be added to the graphs by applying the calculus rules – within a polynomial[4] number of rule applications (for each copy of a graph).

Notice that *saturation* w.r.t. a single-premise rule (r) means that regardless of the choice of Σ among the edges of the graph, the edge δ_1 is already in the graph. Consequently, (r) cannot be applied due to the novelty restriction.

[2] $\mathcal{F}(\Gamma)$ denotes the set of all components of conditionals in Γ.

[3] In a graph perspective, rules enjoying the subformula property would create instead new nodes for the subformulas.

[4] At most $(2 \cdot |\Gamma| + 2)^2$ edges can be added.

Concluding axiom: $\Pi, \mathcal{E} \vdash \mathcal{E}$ **Rule form:** $\dfrac{\{\Pi, \Sigma, \delta_j \vdash \mathcal{E}\}_{j=1}^{m} \quad \mathfrak{S}}{\Pi, \Sigma \vdash \mathcal{E}}$

Rule	m	Σ	$\{\delta_j\}_{j=1}^m$	$\mathfrak{S}$
(Id^s)	$m=1$	$\emptyset$	$\{(A,A)\}$	$\top$ (no condition)
$(Equiv^s)$	$m=1$	$\{(A,A'),(A',A),(A,B)\}$	$\{(A',B)\}$	$\top$ (no condition)
$(Loop^s)$	$m=1$	$\{(A_i,A_{i+1})\}_{i=1}^{n-1} \cup \{(A_n,A_1)\}$	$\{(A_1,A_n)\}$	$\top$ (no condition)
(Tr^s)	$m=1$	$\{(A,B),(B,C)\}$	$\{(A,C)\}$	$\top$ (no condition)
(Cn^s)	$m=1$	$\{(A,B_i)\}_{i=1}^n$	$\{(A,C)\}$	$\bigwedge_{i=1}^n B_i \Rightarrow C$
(CnM^s)	$m \geq 1$	$\{(A,B_i)\}_{i=1}^n$	$\{(A,C_j)\}_{j=1}^m$	$\bigwedge_{i=1}^n B_i \Rightarrow \bigvee_{j=1}^m C_j$

Analyticity restriction: $\mathcal{F}(\delta_j) \subseteq \mathcal{F}(\Pi \cup \Sigma)$ **Novelty restriction:** $\delta_j \notin \Pi \cup \Sigma$

Figure 2: Strongly analytic calculi for KLM logics

For the rule (CnM^s) saturation is achieved if at least one of the edges $\{\delta_j\}_{j=1}^m$ is already present in the graph. A graph with nodes $\mathcal{F}(\Gamma \cup \{\mathcal{E}\})$ and starting edges Γ, reaching saturation w.r.t. all rules of $\mathcal{SC}_\mathcal{L}$ only, corresponds to either a proof of $\Gamma \vdash \mathcal{E}$ or a failed derivation. The latter provide a counterexample to the derivability of $\Gamma \vdash \mathcal{E}$.

However, to prove the completeness of our calculi and encode them into SMT solvers in Sec. 5.1, we will generalize the notion of counterexample to *separating graphs*. A separating graph for $\Gamma \vdash \mathcal{E}$ contains all conditionals from Γ as edges, but no edge $\mathcal{E}$, while still being saturated w.r.t. the rules of $\mathcal{SC}_\mathcal{L}$.

Definition 5. *$G = (V(G), E(G))$ is a* separating *graph for $\Gamma \vdash \mathcal{E}$ w.r.t. $\mathcal{SC}_\mathcal{L}$ if it satisfies the following four conditions: (1) $V(G) = \mathcal{F}(\Gamma \cup \{\mathcal{E}\})$, (2) $\Gamma \subseteq E(G)$, (3) $\mathcal{E} \notin E(G)$, and (4) for every rule (instance) of $\mathcal{SC}_\mathcal{L}$, if $\Sigma \subseteq E(G)$ then at least one of the edges $\{\delta_j\}_{j=1}^m$ belongs to $E(G)$.*

Although the edges in a separating graph for $\Gamma \vdash \mathcal{E}$ are not required to be derivable from Γ, all edges *not present* in the graph are *not* derivable from Γ due to saturation; hence a separating graph can indeed serve as a counterexample to derivability.

Lemma 2. *A separating graph for $\Gamma \vdash \mathcal{E}$ w.r.t. $\mathcal{SC}_\mathcal{L}$ exists iff $\Gamma \vdash \mathcal{E}$ is not derivable in $\mathcal{SC}_\mathcal{L}$.*

Proof. ($\Leftarrow$) If $\Gamma \vdash \mathcal{E}$ is not derivable in $\mathcal{SC}_\mathcal{L}$ then there is a failed derivation that ends up in a graph that satisfies the conditions of Def. 5.

($\Rightarrow$) We cannot simultaneously have a separating graph G and a proof in $\mathcal{SC}_\mathcal{L}$ for $\Gamma \vdash \mathcal{E}$. Suppose to have both, then there exists a proof (a proof branch, in case of **P** and **M**) whose conditionals on the LHS of every sequent belong to $E(G)$: for the root this is true by condition (2) of Def. 5 and for every rule application there is a premise for which this is true by condition (4). Then this also holds for the sequent-leaf of this branch, which has $\mathcal{E}$ on the LHS, contradicting condition (3) of Def. 5. ☐

Example 1. *Fig. 3 (left) depicts the sequent proof of $(A,B),(A,C) \vdash (A \wedge B, C)$ in the calculus $\mathcal{SC}_\mathbf{C}$; Γ_i denotes all the conditionals on the LHS apart from the one added at the previous (i-th) step and from those used at the next ((i+1)-th) step. Fig. 3 (right) depicts its graph view. The solid arrows represent the initial edges of the graph. The dashed arrows are added by applying the rules, and are labeled with the rule's name. Those corresponding to rule applications in the proof (left) are also labeled with the step number. $(A \wedge B, C)$ is derived at step 5. Not needed to reach a proof, the edges labeled with $*$ are present in the minimal saturated graph.*

Consider now the sequent $(A,B),(A,C) \vdash (C, A \wedge B)$, not provable in $\mathcal{SC}_\mathbf{C}$. The same graph in Fig. 3 (right) corresponds to the failed derivation and it is separating. We can get other separating graphs by introducing extra edges (for example, the edge (B,C) can be added without violating saturation w.r.t. the rules of $\mathcal{SC}_\mathbf{C}$).

Henceforth we use both sequent and graph perspectives of our calculi and derivations, depending on which is more convenient at the moment. In the the upcoming soundness and completeness proofs, we will mainly employ the sequent view for the former and the graph view for the latter. Specific logic subsections will provide relevant details.

Roadmap of Soundness: For **C**, **CL** and **CM** the proofs proceed by emulating the calculus rules using the rules in the KLM paper for the corresponding logic. The rules of $\mathcal{SC}_\mathbf{C}$, $\mathcal{SC}_\mathbf{CL}$, and $\mathcal{SC}_\mathbf{CM}$ are indeed "sequent-ized" versions of (or derivable from) the original KLM rules (see Fig. 1). The same does not hold for (CnM^s), present in **P** and **M**, for which soundness is instead proved semantically.

Roadmap of Completeness: For each KLM logic $\mathcal{L}$, we show that there is no proof of $\Gamma \vdash \mathcal{E}$ in $\mathcal{SC}_\mathcal{L}$ only when the given entailment does not hold. As failed derivations end up in a separating graph G (Lem. 2), we use G to define an $\boxed{\mathcal{L}\text{-countermodel } \mathbf{M}_\mathcal{L}(G) \text{ for } \Gamma \vdash \mathcal{E}.}$ $\mathbf{M}_\mathcal{L}(G)$ is constructed to satisfy exactly the conditionals between formulas in $V(G)$ that appear as edges in G.

Definition 6. *A preference model M is* aligned *with a graph G if for all $A, B \in V(G)$, $M \models (A,B)$ iff $(A,B) \in E(G)$.*

Lemma 3. *If M is aligned with a separating graph G for $\Gamma \vdash \mathcal{E}$ then M is a countermodel for $\Gamma \vdash \mathcal{E}$.*

Proof. Being G a separating graph $\Gamma \subseteq E(G)$ and $\mathcal{E} \notin E(G)$, so, due to alignment, M satisfies all conditionals in Γ, but not $\mathcal{E}$. ☐

Our countermodel construction is inspired by the canonical model construction in the KLM paper, albeit with a distinct "analytic" approach. We indeed focus exclusively on conditionals between formulas from $V(G)$, rather than all

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{\Gamma_5, (A\wedge B, C) \vdash (A \wedge B, C)}}{\Gamma_4, (A, C), (A, A\wedge B), (A \wedge B, A) \vdash (A\wedge B, C)}\, 5, (Equiv^s) \quad A\wedge B \Rightarrow A}{\Gamma_3, (A\wedge B, A\wedge B) \vdash (A\wedge B, C)}\, 4, (Cn^s)}{\Gamma_2, (A, A\wedge B) \vdash (A\wedge B, C)}\, 3, (Id^s) \quad A, B \Rightarrow A\wedge B}{\Gamma_1, (A, B), (A, A) \vdash (A\wedge B, C)}\, 2, (Cn^s)}{(A, C), (A, B) \vdash (A\wedge B, C)}\, 1, (Id^s)$$

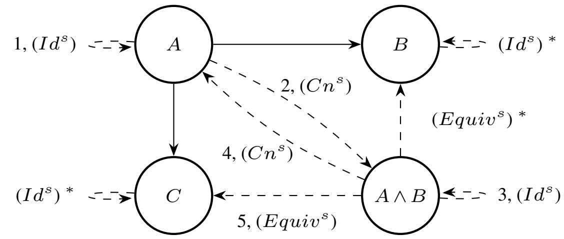

Figure 3: Derivation of $(A, B), (A, C) \vdash (A \wedge B, C)$ in $\mathcal{SC}_\mathbf{C}$. Left: Sequent view. Right: Graph view.

possible conditionals. In particular, the states of our countermodels will be equivalence classes on $V(G)$ (vs on all formulas as in the KLM paper) w.r.t. *conditional equivalence in* G defined as: $A \leftrightarrow B$ when $\{(A, B), (B, A)\} \subseteq E(G)$ (vs $A \sim B$ when $A \mathrel{|\!\sim} B$ and $B \mathrel{|\!\sim} A$). Notice that $\leftrightarrow$ is an equivalence relation for a graph G saturated w.r.t. the rules (Id^s) (for reflexivity) and $(Equiv^s)$ (for transitivity), present in all our calculi.

Henceforth, $[A]_\leftrightarrow$ will denote the equivalence class of the formula A and ${}^{V(G)}/_\leftrightarrow$ will denote the set of all equivalence classes.

The labeling function and the preference relation for countermodels will be tailored to each logic $\mathcal{L}$, following a methodology similar to the canonical model constructions in the KLM paper. This will ensure compliance with the conditions imposed on $\mathcal{L}$-models, relying on saturation w.r.t. the rules of the calculus $\mathcal{SC}_\mathcal{L}$ in each case.

The labeling of a state will be constructed to satisfy exactly the consequences of all its formulas in G, as stated in the requirement below. $N_G(A)$ will denote $\{B \mid (A, B) \in E(G)\}$, i.e. the set of consequences (or neighbors) in G for all $(A, B) \in E(G)$. We will extend this notation to sets of formulas: $N_G(\alpha) = \bigcup_{A\in\alpha} N_G(A)$, and its complement $\overline{N}_G(\alpha) = V(G) \setminus N_G(\alpha)$.

Requirement 1 (Labeling requirement)**.** *We seek for a labeling function* w *on* ${}^{V(G)}/_\leftrightarrow$ *such that for any* $\alpha \in {}^{V(G)}/_\leftrightarrow$ *and* $B \in V(G)$, $w(\alpha) \models B$ *iff* $B \in N_G(\alpha)$.

For the monotonic logics **CM** and **M**, the fulfilment of Req. 1 will be enough to prove alignment with G for the degenerate case of an empty preference relation (as required in these logics). For the non-monotonic logics **C**, **CL**, and **P** we chose a preference relation $\prec$ in such a way that for every formula A state $[A]_\leftrightarrow$ is the least state (and therefore, due to the asymmetry of $\prec$, the only minimal state) satisfying A.

Requirement 2 (Preference requirement)**.** *We seek for a model* $M = \langle {}^{V(G)}/_\leftrightarrow, w, \prec\rangle$ *such that* $[A]_\leftrightarrow \in$ *least*$(\|A\|^M)$ *for any* $A \in V(G)$.

Satisfying Req. 1 *and* 2 implies alignment with G.

By Lem. 3, alignment of $\mathbf{M}_\mathcal{L}(G)$ suffices for it to serve as a countermodel for $\Gamma \vdash \mathcal{E}$. To achieve completeness, it remains to verify that $\mathbf{M}_\mathcal{L}(G)$ is an $\mathcal{L}$-model. Ensuring smoothness for our countermodels requires special attention. For the KLM logics that mandate the preference relation to be a strict partial order (i.e., all but **C**), smoothness naturally follows from the finiteness of the constructed model (in a finite partially ordered set, every decreasing chain has the least element). The situation for **C** is more intricate. Req. 2 guarantees the existence of the least element in $\|A\|^M$ for all $A \in V(G)$. This condition is a weaker version of l-smoothness (Remark 1) confined to formulas from $\mathcal{F}(\Gamma \cup \{\mathcal{E}\})$. We refer to this new condition as *analytic l-smoothness*. As we will see, any analytically l-smooth countermodel, which we will term as *analytic countermodel*, can be transformed into an l-smooth countermodel by adding new states. Thus, analytic l-smoothness can be used as an alternative semantic characterization of (finite) entailment in **C**.

Remark 2. *Analytic l-smoothness can be verified in polynomial time, and our construction* $\mathbf{M}_\mathbf{C}(G)$ *shows the existence of polynomial-size analytic countermodels. In contrast, it is unclear whether smoothness (or l-smoothness) from the KLM paper always permits a polynomial-size countermodel, and whether these properties (which apply to all formulas in the language) can be verified in polynomial-time.*

3.1 The base logic C

The calculus $\mathcal{SC}_\mathbf{C}$ for **C** consists of the rules (Id^s), $(Equiv^s)$, and (Cn^s) from Fig. 2. (Id^s) is a "sequent-ized" version of one of the original rules for **C**, (Cn^s) allows the rule (*RWk*) to be applied to multiple conditionals having the same conditions, while $(Equiv^s)$ mirrors the (*Equivalence*) rule in the KLM paper and allows swapping conditionally equivalent conditions. Their soundness follows from the lemma below, stating that for each rule Σ derives the generated conditional δ_1 via the original KLM rules for **C**.

Lemma 4 (Emulation)**.** *For every rule of* $\mathcal{SC}_\mathbf{C}$*, (the classical validity of the rule proviso* $\mathfrak{S}$ *implies)* $\Sigma \vdash_\mathbf{C} \delta_1$.

Proof. For (Id^s), $\delta_1 = (A, A)$ is derived by (*Id*). The claim for (*Equivalence*) is in the KLM paper. For (Cn^s) we use the (*And*) rule $(A, B_1), (A, B_1) \vdash (A, B_1 \wedge B_2)$, shown to be derivable in **C** in the KLM paper. The conditional $\delta_1 = (A, C)$ generated by (Cn^s) arises by n applications of (*And*) followed by (*RWk*), as $\delta = (B_1 \wedge \cdots \wedge B_n \Rightarrow C)$. ☐

Theorem 5 (Soundness of $\mathcal{SC}_\mathbf{C}$)**.** *If* $\Gamma \vdash \mathcal{E}$ *is derivable in* $\mathcal{SC}_\mathbf{C}$ *then* $\Gamma \vdash_\mathbf{C} \mathcal{E}$.

For completeness, as described in the roadmap, we build a countermodel $\mathbf{M}_\mathbf{C}(G)$ satisfying Req. 1 and Req. 2 for a separating graph G for $\Gamma \vdash \mathcal{E}$ w.r.t. $\mathcal{SC}_\mathbf{C}$.

The definition of the labeling function will rely on the saturation of G w.r.t. (Cn^s). Hence $N_G(A)$ contains all the formulas from $V(G)$ that are classically entailed by $N_G(A)$. So, if $B \in \overline{N}_G(A)$ then $N_G(A) \not\Rightarrow B$. We can lift it to the level of equivalence classes, since $N_G([A]_\leftrightarrow) = N_G(A)$ due to the saturation w.r.t. $(Equiv^s)$: $N_G(\alpha) \not\Rightarrow B$ for any $\alpha \in {}^{V(G)}/_\leftrightarrow$ and $B \in \overline{N}_G(\alpha)$. Hence there exist Boolean assignments satisfying all formulas in $N_G(\alpha)$ and falsifying B. We denote by $\rho(\alpha, B)$ one (arbitrarily chosen) such assignment. We can then define a labeling function $W_G(\alpha) = \{\rho(\alpha, B) \mid B \in \overline{N}_G(\alpha)\}$, satisfying Req. 1 and mapping each state into linearly many Boolean assignments.

We also reuse the definition of preference relation from the canonical model for **C** in the KLM paper, adapting it to our analytic setting.

Definition 7. *For $\alpha, \beta \in {}^{V(G)}/_\leftrightarrow$, $\beta \prec_G \alpha$ when $\alpha \neq \beta$ and there exists $B \in \beta$ such that $B \in N_G(\alpha)$.*

Notice that $\prec_G$ is asymmetric for G saturated w.r.t. $(Equiv^s)$. Indeed $\alpha \prec_G \beta$ and $\beta \prec_G \alpha$ would imply $A \leftrightarrow B$ for some $A \in \alpha$ and $B \in \beta$, contradicting $\alpha \neq \beta$.

Countermodel definition: $\mathbf{M_C}(G) = \langle {}^{V(G)}/_\leftrightarrow, W_G, \prec_G \rangle$

Lemma 6. *If G is a separating graph for $\Gamma \vdash \mathcal{E}$ w.r.t. $\mathcal{SC}_\mathbf{C}$ then $\mathbf{M_C}(G)$ is a countermodel for $\Gamma \vdash \mathcal{E}$.*

Proof. W_G satisfies Req. 1 by definition. $\mathbf{M_C}(G)$ satisfies Req. 2 since for every $B \in V(G)$ and for every $\alpha \in {}^{V(G)}/_\leftrightarrow$ such that $\alpha \neq [B]_\leftrightarrow$ if $\alpha \in \|B\|^{\mathbf{M_C}(G)}$ then $B \in N_G(\alpha)$ due to Req. 1, so $[B]_\leftrightarrow \prec_G \alpha$ by definition, so $[B]_\leftrightarrow \in least(\|B\|^{\mathbf{M_C}(G)})$. Req. 1 and Req. 2 together imply alignment with G, which proves the claim by Lem. 3. ☐

$\mathbf{M_C}(G)$ satisfies analytic l-smoothness, because of Req. 2 but may not qualify as a **C**-model, as it is not necessarily smooth. Indeed we cannot say anything about smoothness on the formulas outside $V(G)$. We show that analytic l-smoothness can serve as an alternative characterization for **C**, by transforming any analytic countermodel into an l-smooth countermodel.

Lemma 7. *For a finite Γ, $\Gamma \vdash_\mathbf{C} \mathcal{E}$ iff there is no analytic countermodels for $\Gamma \vdash \mathcal{E}$.*

Proof. Since l-smoothness implies analytic l-smoothness, it suffices to show that any analytic countermodel can be turned into an l-smooth countermodel. Without loss of generality, we restrict Var to the variables occurring in $\Gamma \vdash \mathcal{E}$ (any l-smooth countermodel can be extended to any larger set of variables by assigning the additional variables to false in all labelings).

Consider the following model transformation $M \mapsto M^*$ "repairing" l-smoothness for one formula X. If M does not satisfy l-smoothness for X (i.e. $\|X\|^M \neq \emptyset$ and $least(\|X\|^M) = \emptyset$), we add to M one state s^* with labeling $w(s^*) = \{\rho \mid \rho \models X\}$ and make $s^* \prec s$ for all $s \in \|X\|^M$ and $s \prec s^*$ for all $s \in \mathcal{S} \setminus \|X\|^M$, without changing the labeling or the preference relation for other states. We then have $s^* \in least(\|X\|^{M^*})$ by definition. The asymmetry of $\prec$ is preserved. We show that the transformation also preserves l-smoothness for any $A \neq X$. If $\|A\|^M = \emptyset$ then $X \not\Rightarrow A$ (since $\|X\|^M \neq \emptyset$), so $w(s^*) \not\models A$, so $\|A\|^{M^*} = \emptyset$. If $\|A\|^M \neq \emptyset$ and $s_A \in least(\|A\|^M)$, s_A remains the least after the model transformation: it can only be violated if both (1) $w(s^*) \models A$ and (2) $s^* \prec s_A$ in M^*, but (1) implies $X \Rightarrow A$ and (2) implies $s_A \in \|X\|^M$, which would imply $s_A \in least(\|X\|^M)$ contradicting the absence of l-smoothness on X in M.

Hence any analytic countermodel M for $\Gamma \vdash \mathcal{E}$ can be converted into an l-smooth countermodel in a finite number of steps: select a maximal set of pairwise classically non-equivalent formulas (finite, given that so is Var) and "repair" l-smoothness for each one using the transformation above. The resulting model achieves l-smoothness for all formulas while remaining a countermodel for $\Gamma \vdash \mathcal{E}$ (since the transformation preserves the least states). ☐

Theorem 8 (Completeness of $\mathcal{SC}_\mathbf{C}$)**.** *If $\Gamma \vdash_\mathbf{C} \mathcal{E}$ then $\Gamma \vdash \mathcal{E}$ is derivable in $\mathcal{SC}_\mathbf{C}$.*

Proof. By contraposition: if $\Gamma \vdash \mathcal{E}$ is not derivable in $\mathcal{SC}_\mathbf{C}$, by Lem. 2 there is a separating graph G for it. Hence we can construct a countermodel $\mathbf{M_C}(G)$ for $\Gamma \vdash \mathcal{E}$ that is analytic. By Lem. 7 follows that $\Gamma \not\vdash_\mathbf{C} \mathcal{E}$. ☐

3.2 Logics CL and CM

The calculi $\mathcal{SC}_\mathbf{CL}$ and $\mathcal{SC}_\mathbf{CM}$ for the **CL** and **CM** extend $\mathcal{SC}_\mathbf{C}$ with the rule $(Loop^s)$ and the transitivity rule (Tr^s), respectively. Note that $(Loop^s)$ is derivable from (Tr^s).

Theorem 9 (Soundness of $\mathcal{SC}_\mathbf{CL}$ and $\mathcal{SC}_\mathbf{CM}$)**.** *If $\Gamma \vdash \mathcal{E}$ is derivable in $\mathcal{SC}_\mathbf{CL}$ (resp. $\mathcal{SC}_\mathbf{CM}$) then $\Gamma \vdash_\mathbf{CL} \mathcal{E}$ ($\Gamma \vdash_\mathbf{CM} \mathcal{E}$).*

Proof. Similar to the proof of Th. 5. Observe that Lemma 4 holds also for **CL** and **CM**, since the additional rules $(Loop^s)$ and (Tr^s) are sequent-ized versions of the original KLM rule (*Loop*) (of **CL**), and of (*Transitivity*), shown to be derivable in **CM** in the KLM paper, respectively. ☐

For the countermodels $\mathbf{M_{CL}}(G)$ and $\mathbf{M_{CM}}(G)$ we redefine the preference relation in the countermodel $\mathbf{M_C}(G)$ to fulfill the semantic requirements for these logics – transitivity for **CL** and the empty relation for **CM** – using the saturation w.r.t. the additional rules $(Loop^s)$ and (Tr^s).

Following the KLM paper, transitivity for **CL** is ensured by taking the (positive) transitive closure of $\prec_G$ (denoted as $\prec_G^+$) as the preference relation.

Countermodel definition: $\mathbf{M_{CL}}(G) = \langle {}^{V(G)}/_\leftrightarrow, W_G, \prec_G^+ \rangle$

Note that saturation w.r.t. $(Loop^s)$ is needed to ensure that the transitive closure is asymmetric (and, therefore, it can be taken as a preference relation).

Lemma 10. *If G is saturated w.r.t. $(Loop^s)$ and $(Equiv^s)$, $\prec_G^+$ is asymmetric.*

Proof. Suppose $\alpha \prec_G^+ \beta$ and $\beta \prec_G^+ \alpha$. By the definition of $\prec_G$ and saturation w.r.t. $(Equiv^s)$ there exists a directed cycle in G going through some $A \in \alpha$ and some $B \in \beta$. Due to the saturation w.r.t. $(Loop^s)$, for each edge of the cycle its inverse edge also belongs to $E(G)$, then due to saturation w.r.t. $(Equiv^s)$ we have $A \leftrightarrow B$, so $\alpha = \beta$, which contradicts the initial assumption. ☐

Since $\mathbf{M_{CL}}(G)$ has the same labeling as $\mathbf{M_C}(G)$ and Req. 2 is preserved by taking a positive transitive closure (it preserves the least elements) we have completeness.

Theorem 11 (Completeness of $\mathcal{SC}_{\mathbf{CL}}$)**.** *If* $\Gamma \vdash_{\mathbf{CL}} \mathcal{E}$ *then* $\Gamma \vdash \mathcal{E}$ *is derivable in* $\mathcal{SC}_{\mathbf{CL}}$*.*

Proof. If $\Gamma \vdash \mathcal{E}$ is not derivable, a separating graph G exists by Lem. 2. $\mathbf{M_{CL}}(G)$ satisfies Req. 1 and 2, so it is aligned with G and hence is a countermodel for $\Gamma \vdash_{\mathbf{CM}} \mathcal{E}$ by Lem. 3. $\prec_G^+$, being asymmetric and transitive (a strict partial order), ensures smoothness due to the finiteness of $\mathbf{M_{CL}}(G)$, so $\mathbf{M_{CL}}(G)$ is a **CL**-countermodel for $\Gamma \vdash \mathcal{E}$. □

For **CM**, we take an empty preference relation.

Countermodel definition: $\mathbf{M_{CM}}(G) = \langle {}^{V(G)}/{\leftrightarrow}, W_G, \varnothing \rangle$

Req. 1, satisfied by W_G, and saturation w.r.t. (Tr^s) are enough to prove alignment with G (and, therefore, completeness) directly.

Theorem 12 (Completeness of $\mathcal{SC}_{\mathbf{CM}}$)**.** *If* $\Gamma \vdash_{\mathbf{CM}} \mathcal{E}$ *then* $\Gamma \vdash \mathcal{E}$ *is derivable in* $\mathcal{SC}_{\mathbf{CM}}$*.*

Proof. If $\Gamma \vdash \mathcal{E}$ is not derivable, by Lem. 2 there is a separating graph G. Since $\mathbf{M_{CM}}(G)$ has an empty preference relation, it is a **CM**-model and all states within each truth set are minimal. By Lem. 3 it suffices to show that $\mathbf{M_{CM}}(G)$ is aligned with G. If $(A', B') \notin E(G)$ then $\mathbf{M_{CM}}(G), [A']_{\leftrightarrow} \models\!\equiv A'$ and $\mathbf{M_{CM}}(G), [A']_{\leftrightarrow} \not\models\!\equiv B'$ (both due to Req. 1 satisfied by W_G), so $\mathbf{M_{CM}}(G) \not\models (A', B')$. Now let us show that $(A, B) \in E(G)$ implies $\mathbf{M_{CM}}(G) \models (A, B)$. For any $\alpha \in min(\|A\|^{\mathbf{M_{CM}}(G)})$ we have $\alpha \in N_G(A)$ due to Req. 1 satisfied by W_G, i.e. there is $C \in \alpha$ such that $(C, A) \in E(G)$. Due to saturation of G w.r.t. (Tr^s) the edges of G are transitive, so $(C, B) \in E(G)$, which implies $\mathbf{M_{CM}}(G), \alpha \models\!\equiv B$ again by Req. 1. □

3.3 Logics P and M

The calculi $\mathcal{SC}_{\mathbf{CL}}$ and $\mathcal{SC}_{\mathbf{CM}}$ for **P** and **M**[5] are obtained by extending $\mathcal{SC}_{\mathbf{CL}}$ and $\mathcal{SC}_{\mathbf{CM}}$, with the (CnM^s) rule

$$\frac{\{\Pi, (A, B_1), \dots (A, B_n), (A, C_j) \vdash \mathcal{E}\}^m \quad \bigwedge^n B_i \Rightarrow \bigvee^m C_j}{\Pi, (A, B_1), \dots (A, B_n) \vdash \mathcal{E}}$$

(Note that (Cn^s) is an instance of (CnM^s) for $m = 1$). The motivation for (CnM^s) stems from a semantic perspective. Indeed **P** and **M** differ from the cumulative logics **CL** and **CM** by the restriction that each state is labeled with exactly one Boolean assignment. Hence, to extend our approach to **P** and **M** we need to replace the set $W_G(\alpha)$ of assignments falsifying each formula from $\overline{N}_G(\alpha)$ individually with a single assignment falsifying all of them simultaneously. This is achieved by transforming the rule (Cn^s) to a multi-premise version (CnM^s). Saturation w.r.t. this rule ensures the existence of one assignment satisfying Req. 1: take the instance of (CnM^s) with $\Sigma = N_G(\alpha)$ and $\{\delta_i\}_{i=1}^m = \overline{N}_G(\alpha)$, then the side condition $\mathfrak{S}$ is $N_G(\alpha) \Rightarrow \bigvee_{C \in \overline{N}_G(\alpha)} C$ and it should be false due to saturation w.r.t. (CnM^s). Therefore there exists a Boolean assignment satisfying all formulas in $N_G(\alpha)$ and falsifying all formulas in $\overline{N}_G(\alpha)$ (thus satisfying Req. 1). We will denote such assignment $W_G^1(\alpha)$.

We now replace W_G with W_G^1 in the definition of countermodel for **CL** (resp. **CM**) and get a countermodel in which each state is labeled by exactly one Boolean assignment.

Countermodel definitions:

- For **P**: $\mathbf{M_P}(G) = \langle {}^{V(G)}/{\leftrightarrow}, W_G^1, \prec_G^+ \rangle$.
- For **M**: $\mathbf{M_M}(G) = \langle {}^{V(G)}/{\leftrightarrow}, W_G^1, \varnothing \rangle$.

Since we changed only labeling, by preserving Req. 1 and simultaneously ensuring functionality, the completeness proofs in Th. 11 and Th. 12 easily extend to **P** and **M**.

Theorem 13 (Completeness)**.** *If* $\Gamma \vdash_{\boldsymbol{P}} \mathcal{E}$ *(resp.* $\Gamma \vdash_{\boldsymbol{M}} \mathcal{E}$*) then* $\Gamma \vdash \mathcal{E}$ *is derivable in* $\mathcal{SC}_{\boldsymbol{P}}$ *(resp.* $\mathcal{SC}_{\boldsymbol{M}}$*).*

The soundness of the rule (CnM^s) for **P** and **M** is proved semantically using the characterization of finite entailments in terms of l-smoothness (see Remark 1).

Theorem 14 (Soundness of $\mathcal{SC}_{\mathbf{P}}$ and $\mathcal{SC}_{\mathbf{M}}$)**.** *If* $\Gamma \vdash \mathcal{E}$ *is derivable in* $\mathcal{SC}_{\mathcal{L}}$ *then* $\Gamma \vdash_{\mathcal{L}} \mathcal{E}$*, for* $\mathcal{L} \in \{\boldsymbol{P}, \boldsymbol{M}\}$*.*

Proof. By induction on the length of the proof. If the last applied rule is a concluding axiom, the claim is trivial; for the rules other than (CnM^s) it follows from the demonstrated soundness w.r.t. the weaker KLM logics. The proof for (CnM^s) is by contradiction. If $\Gamma \not\vdash_{\mathcal{L}} \mathcal{E}$ then there is a $\mathcal{L}$-countermodel M for $\Gamma \vdash \mathcal{E}$ satisfying l-smoothness. For the rule (CnM^s), we assume $\{(A, B_i)\}_{i=1}^n \subseteq \Gamma$, and $\bigwedge_{i=1}^n B_i \Rightarrow \bigvee_{j=1}^m C_j$. In M there is exactly one state in $least(\|A\|^M)$ labeled with one world, which satisfies B_i for every i, so it should satisfy also $\bigvee_{j=1}^m C_j$, and therefore C_k for some k. Then M satisfies (A, C_k) and thus is a $\mathcal{L}$-countermodel for a premise $\Gamma, (A, C_k) \vdash \mathcal{E}$, which contradicts (one of) the inductive hypotheses. □

4 Applications: Complexity Results

We utilize the introduced calculi to establish uniform complexity results for the entailment problem in KLM logics, as well as for two useful restrictions on the form of conditionals: *Horn* and *literal* conditionals. In Horn (resp. literal) conditional (A, B), A is a conjunction of propositional variables (resp. literals, i.e. possibly negated variables) and B is a single propositional variable (resp. literal). These restrictions play a significant role in domains like logic programming (Dantsin et al. 2001) and causality (Bochman 2021).

As seen before, entailment checking in a KLM logic $\mathcal{L}$ can be reduced to the search for a separating graph G w.r.t. $\mathcal{SC}_{\mathcal{L}}$ (Lem. 2). Note that the saturation of G w.r.t. every rule can be checked in polynomial time *modulo checks of classical entailment conditions* $\mathfrak{S}$.

Lemma 15. *Checking the saturation of graph w.r.t.* $\mathcal{SC}_{\mathcal{L}}$ *can be reduced in polynomial time to a polynomial number of checks of classical non-entailment.*

Proof. Saturation w.r.t. (Id^s), $(Equiv^s)$ and (Tr^s) can be checked directly by examining all (triplets of) nodes in $V(G)$. Saturation w.r.t. $(Loop^s)$ means that there is no cycle

[5] We include **M** (= classical logic) in our investigation for sake of completeness, and to highlight the uniformity of our approach.

in G going through several equivalence classes of $^{V(G)}/_{\leftrightarrow}$, which can be easily checked in polynomial time. For (Cn^s) and (CnM^s) we check violations of the side condition $\mathfrak{S}$ for subset-maximal choices of Σ and $\{\delta_j\}_{j=1}^m$, since removing conditionals from Σ or $\{\delta_j\}_{j=1}^m$ cannot make $\mathfrak{S}$ valid. Hence for (Cn^s) we check $N_G(A) \not\Rightarrow B$ for all $A \in V(G)$ and $B \in \overline{N}_G(A)$, and for (CnM^s) we check $N_G(A) \not\Rightarrow \bigvee_{B\in\overline{N}_G(A)} B$ for all $A \in V(G)$. □

Thus, we get co-NP-completeness with a separating graph playing the role of a witness for non-entailment.

Theorem 16. *The entailment problem in each $\mathcal{L} \in \{\boldsymbol{C}, \boldsymbol{CL}, \boldsymbol{CM}, \boldsymbol{P}, \boldsymbol{M}\}$ is co-NP-complete.*

Proof. $\Gamma \not\vdash_{\mathcal{L}} \mathcal{E}$ can be checked in non-deterministic polynomial time by guessing a separating graph G for $\Gamma \vdash \mathcal{E}$ and checking its saturation w.r.t. $\mathcal{SC}_{\mathcal{L}}$ as in Lem. 15 and guessing falsifying assignments for the resulting classical entailments. Co-NP-hardness is given by the following reduction: A is classically valid iff $\vdash_{\mathcal{L}} (\top, A)$. □

If we constrain the shape of conditionals s.t. classical entailments in $\mathfrak{S}$ can be verified in polynomial time, we have:

Theorem 17. *The entailment problem for Horn and literal conditionals in $\mathcal{L} \in \{\boldsymbol{C}, \boldsymbol{CL}, \boldsymbol{CM}\}$ is in PTIME.*

Proof. Entailment between conjunctions of literals is verifiable in polynomial time. Hence so is the check of possible rule applications using Lem. 15. For $\mathcal{L} \in \{\mathbf{C}, \mathbf{CL}, \mathbf{CM}\}$, $\mathcal{SC}_{\mathcal{L}}$ derivations consist of a single branch, so either a proof or a separating graph is found in polynomial time. □

As shown in Th. 19, this result extends to **P** (and **M**) for Horn conditionals, although it does not apply to literal conditionals. Indeed

Theorem 18. *The entailment problem in $\mathcal{L} \in \{\boldsymbol{P}, \boldsymbol{M}\}$ for literal conditionals is co-NP-hard.*

Proof. By a reduction of CNF-SAT to non-entailment in $\mathcal{L}$: A CNF-formula $F = \bigwedge_{i=1}^m \bigvee_{j=1}^{n_i} l_{ij}$ is classically satisfiable iff $\Gamma(F) \not\vdash_{\mathcal{L}} (\top, \bot)$ where $\Gamma(F) = \{(\bigwedge_{j=1}^{n_i} \neg l_{ij}, \bot)\}_{i=1}^m$ (with double negations simplified). The satisfying assignment for F corresponds to the labeling of $\top$-minimal state in the countermodel for KLM-entailment, which has to satisfy at least one l_{ij} for every i. □

Hence for KLM logics, the inclusion of (*Or*) results in increased complexity for entailment of literal conditionals.

The KLM paper semantically shows that **P** and **CL** derive identical conditionals under Horn restrictions. This and the analogous result for **M** and **CM**, is proved using our calculi.

Theorem 19. *If $\Gamma \cup \{\mathcal{E}\}$ is finite and contains only Horn conditionals, then $\Gamma \vdash_{\boldsymbol{CL}} \mathcal{E}$ (resp. $\Gamma \vdash_{\boldsymbol{CM}} \mathcal{E}$) iff $\Gamma \vdash_{\boldsymbol{P}} \mathcal{E}$ (resp. $\Gamma \vdash_{\boldsymbol{M}} \mathcal{E}$).*

Proof. If $\{B_i\}_{i=1}^n$, and $\{C_j\}_{j=1}^m$ are conjunctions of variables, $\bigwedge_{i=1}^n B_i \Rightarrow \bigvee_{j=1}^m C_j$ is true iff some C_k contains only variables occurring on the antecedent, which implies $\bigwedge_{i=1}^n B_i \Rightarrow C_k$. So for literal conditionals we can replace each application of (CnM^s) with an application of (Cn^s) (removing all sequent premises but one). □

5 SMT-based solver

SMT (Satisfiability Modulo Theory) solvers like Z3 (de Moura and Bjørner 2008) or cvc5 (Barbosa et al. 2022) are important extensions of SAT solvers which have built-in support for certain theories, for example integers or sequences. While their primary domain lies within classical logic and first-order theories, they have recently found application in automated deduction for logics different from classical logic, e.g. (Areces, Fontaine, and Merz 2015; Eisenhofer et al. 2023; Fiorentini, Goré, and Graham-Lengrand 2019). An SMT solver is usually implemented as a propositional logic (SAT) solvers with the ability to incorporate additional Boolean constraints on demand via some variant of the DPLL($\mathcal{T}$) algorithm (Biere et al. 2021) by so-called "theory propagations". We use $J_1, \ldots J_m \Vdash F$ to denote that the formula $\bigwedge_{1\le i\le m} J_i \to F$ is added to the SAT solver whenever all literals $J_1, \ldots, J_m$ are assigned true. As propagations add constraints on-demand, we do not need to know which constraints will be relevant for the solver. They are included only when required during reasoning. This can improve memory usage and reasoning speed, which is why we opted for an SMT encoding.

5.1 Implementing $\mathcal{SC}_{\mathcal{L}}$: KLMPropagator

The rules of our calculi lend themselves to a natural encoding into SMT solvers, resulting in the KLMPropagator prover. We address the entailment problem in a KLM logic $\mathcal{L}$ by encoding the existence of a separating graph G for $\Gamma \vdash \mathcal{E}$. By Lem. 2, such a graph exists iff the sequent $\Gamma \vdash \mathcal{E}$ is not derivable in $\mathcal{SC}_{\mathcal{L}}$. Thus, if the solver fails to construct a separating graph G, we conclude that the sequent is provable. Otherwise, we construct the countermodel $\mathrm{M}_{\mathcal{L}}(G)$.

For KLMPropagator, the problem boils down to disproving classical entailment problems and choosing the edges actually present in a separating graph G. Classical reasoning on the formulas labeling the nodes will occur solely in the form of disproving entailments and in order to deal with (Cn^s) and (CnM^s).

To encode a separating graph G (with nodes $V(G) = \mathcal{F}(\Gamma \cup \{\mathcal{E}\})$), we represent its edges using propositional atoms $\{e_{A,B} \mid A, B \in V(G)\}$ (referred to as edge atoms). Assigning true to $e_{A,B}$ in the encoding indicates the existence of the edge between nodes A and B in the graph. Let $\Gamma = \{(A_1, B_1), \ldots, (A_{n-1}, B_{n-1})\}$ and $\mathcal{E} = (A_n, B_n)$. We check if $\bigwedge_{1\le i<n} e_{A_i,B_i} \wedge \neg e_{A_n,B_n}$ is inconsistent. This encodes the second and third condition of Def. 5 (the first condition is implicitly encoded within the edge atoms $e_{A,B}$).

$Inst_{A,B}(C)$ denotes a copy of the propositional formula C, where each atom in C is replaced by a fresh copy. For instance, $Inst_{A,C}(A \wedge B) = A_{A,C} \wedge B_{A,C}$.

Based on our calculi, the solver monitors the Boolean assignments made to the edge atoms $e_{A,B}$ and implements propagations based on the rules in Fig. 2. These enforce the last condition (saturation) for a graph to be separating. The encoding of the rules for **C** as propagations is as follows (for all nodes $A, B, C, A_1, \ldots, A_n \in V(G)$):

(Id^s): $\Vdash e_{A,A}$ ($Equiv^s$): $e_{A,A'}, e_{A',A}, e_{A,B} \Vdash e_{A',B}$
(Cn^s): $\neg e_{A,C} \Vdash \neg Inst_{A,C}(C)$ and
$\neg e_{A,C}, e_{A,B} \Vdash Inst_{A,C}(B)$

For the other KLM logics, we additionally have:

($Loop^s$): $e_{A_1,A_2}, e_{A_2,A_3}, \ldots, e_{A_n,A_1} \Vdash e_{A_1,A_n}$
(Tr^s): $e_{A,B}, e_{B,C} \Vdash e_{A,C}$
(CnM^s): $\neg e_{A,C}, \neg e_{A,B} \Vdash \neg Inst_{A,C}(B)$

These rules, in combination with the initial constraint, express the existence of a separating graph.

In practice, the identities (Id^s) can be eagerly asserted by unit-clauses to the solver and ($Equiv^s$), ($Loop^s$), and (Tr^s) can be easily added on-demand by keeping track of the existing (transitive) children of each node. These four rules mirror exactly the ones in Fig. 2. For example, (Id^s) forces the graph to be reflexive and (Tr^s) to be transitive.

(Cn^s) requires that in any separating graph G where $N_G(A) \Rightarrow C$ also $(A, C) \in E(G)$. Equivalently, $(A, C) \notin E(G)$ requires $N_G(A) \not\Rightarrow C$, which can be justified by a propositional model for $M_{A,C} := Inst_{A,C}(B_1) \wedge \ldots \wedge Inst_{A,C}(B_n) \wedge \neg Inst_{A,C}(C)$. We use copies ($Inst$) of nodes, as each justification for the non-existence of an edge between A and C is independent of the others and thus requires independent atoms. In particular, we can add $(\neg e_{A,C} \wedge e_{A,B_1} \wedge \ldots \wedge e_{A,B_n}) \rightarrow M_{A,C}$ for each potential combination of edges to enforce the solver to satisfy (Cn^s).

Example 2 (Ex. 1 in SMT)**.** *Consider step (2) in the proof of Fig. 3. Assume the solver has assigned $e_{A,A}, e_{A,B}$, and $e_{A,C}$ true and decides to assign $e_{A,A\wedge B}$ false. It runs into a conflict with $(\neg e_{A,A\wedge B} \wedge e_{A,A} \wedge e_{A,B} \wedge e_{A,C}) \rightarrow (A_{A,A\wedge B} \wedge B_{A,A\wedge B} \wedge C_{A,A\wedge B} \wedge \neg(A_{A,A\wedge B} \wedge B_{A,A\wedge B}))$, an instance of ($Cn^s$). This way, the solver is prevented from violating (Cn^s) by either setting $e_{A,A}, e_{A,B}$, or $e_{A,C}$ to false or by assigning $e_{A,A\wedge B}$ true.*

Optimized Propagation: *For the encoding of (Cn^s) we can incrementally add the needed parts. Assume that the solver has set $e_{A,A}$ true due to (Id^s) and then assigns $e_{A,A\wedge B}$ false. It can immediately propagate $\neg e_{A,A\wedge B} \Vdash \neg(A_{A,A\wedge B} \wedge B_{A,A\wedge B})$ and $\neg e_{A,A\wedge B}, e_{A,A} \Vdash A_{A,A\wedge B}$. If the prover later sets $e_{A,B}$ and we thus propagate $\neg e_{A,A\wedge B}, e_{A,B} \Vdash B_{A,A\wedge B}$, it detects the conflict. Being $e_{A,A}$ and $e_{A,B}$ true, the prover concludes that $e_{A,A\wedge B}$ has to be set to true as well. In contrast to the lengthy clause above, the solver is aware of which edge set to true is responsible for which part of the set of formulas that led to the conflict.*

The propagation of the rule (CnM^s) requires $N_G(A) \Rightarrow \overline{N}_G(A)$. By contraposition, this means that there is a propositional model that satisfies all $N_G(A)$ but none of $\overline{N}_G(A)$. In terms of our propagation rule for (CnM^s), this means that we additionally require that all non-neighbors are falsified.

Remark 3 (Countermodel reconstruction)**.** *The encoding of (Cn^s) and (CnM^s) ensures that the classical model over atoms in $Inst_{A,B}$ found by* KLMPROPAGATOR *give us a labeling on $^{V(G)}/_{\leftrightarrow}$ satisfying Req. 1. Specifically, by (the encoding of) (Cn^s) each $[A]_{\leftrightarrow} \in {}^{V(G)}/_{\leftrightarrow}$ can be labeled by a set consisting of a single model for each formula in $\{\neg Inst_{A,B}(B) \wedge \bigwedge_{F|e_{A,F}=\top} Inst_{A,B}(F) \mid B \in V(G), e_{A,B} = \bot\}$, while for ($CnM^s$)* a single *model for $\bigwedge_{F|e_{A,F}=\bot} \neg Inst_{A,B}(F) \wedge \bigwedge_{F|e_{A,F}=\top} Inst_{A,B}(F)$ for an arbitrary B, is enough to satisfy Req. 1 (providing a functional labeling). Together with a preference relation, which can be reconstructed from a separating graph as shown in the countermodel constructions in Sec. 3, this enables to turn the model found by our prover into a preference countermodel for the given entailment.*

5 conditional premises				
Logic	*KLMPropagator*	KLMlean 2.0	NESCOND	IOCondProver
C	100 (82 + 18)	3 (0 + 3)	22 (7 + 15)	-
CL	100 (82 + 18)	23 (5 + 18)	-	-
P	100 (75 + 25)	41 (16 + 25)	-	-
CM	100 (76 + 24)	-	-	98 (74 + 24)
20 conditional premises				
Logic	*KLMPropagator*	KLMlean 2.0	NESCOND	IOCondProver
C	100 (47 + 53)	4 (0 + 4)	18 (0 + 18)	-
CL	98 (45 + 53)	28 (0 + 28)	-	-
P	100 (13 + 87)	50 (0 + 50)	-	-
CM	100 (23 + 77)	-	-	33 (10 + 23)

Figure 4: Experimental results: number of instances (out of 100) solved within 10s (SAT + UNSAT)

KLMPROPAGATOR[6] implements the described approach in the Z3 SMT solver using the user-propagator framework from (Bjørner, Eisenhofer, and Kovács 2023) in C++. The current implementation works as described before and can issue countermodels in the form of separation graphs, in case the given input is not provable. Each non-existing edge $e_{A,B}$ is labeled by a respective propositional witness that could be used to construct a labeling as described in Remark 3. With no constraints on the nodes' formulas, our solver seamlessly supports quantifiers and theory atoms.

5.2 Experimental results

We compared KLMPROPAGATOR against existing tools for verifying entailment of conditionals in KLM logics, namely KLMLEAN 2.0 (Giordano, Gliozzi, and Pozzato 2007) (for **C**, **CL** and **P**), NESCOND (Olivetti and Pozzato 2014) (for **C**), and IOCONDPROVER (Lellmann 2021) (for **CM**). In contrast with our solver, that employs SAT-based reasoning, they implement (sequent or tableau) calculi and use Prolog to graze the search space for potential proofs/countermodels.

Following methodologies used in the evaluation of these existing tools, we relied on randomly generated queries (due to the absence of recognized benchmarks for the KLM logics). We conducted two batches of experiments depicted in Fig. 4, each consisting of 100 queries, using 5 and 20 conditional premises, respectively, constructed using 5 different propositional variables. This approach minimizes any advantage gained from relying on an efficient SAT solver for classical entailment checks. The experiments revealed a substantial performance gap across all logics. With the exception of a few difficult instances, our tool successfully solved all problems within 10 seconds, often much faster. In contrast, KLMLEAN 2.0 and NESCOND encountered

[6] https://github.com/CEisenhofer/KLMPropagator

significant challenges, even with small, satisfiable instances. Larger instances rapidly became infeasible for all solvers except ours, regardless of satisfiability, unless they turned out to be easy special cases.

6 Conclusions and Future work

We have introduced modular sequent calculi for the KLM logics **C**, **CL**, **CM**, **P** (and **M**). The calculi have been employed to obtain uniform complexity results for the logics and fragments thereof, a more effective semantics for **C** – featuring polynomial-time model checking – and an efficient SMT-based prover with countermodel generation capabilities. Our approach is modular and does not rely on well-behaved semantics, so it can be potentially modified both for some well-known extension of the logic **P** (such as rational entailment (Lehmann and Magidor 1992) or disjunctive rational entailment (Booth and Varzinczak 2021)) and for weaker logics rejecting some of KLM postulates. Furthermore, our approach can provide a solid foundation for the proof-theoretic study of practical closure operations based on KLM logics, such as rational closure (Lehmann and Magidor 1992), lexicographical closure (Lehmann 1999), multipreference closure (Giordano and Gliozzi 2021) and their many modifications used in description logic (Casini and Straccia 2010; Giordano et al. 2018; Bonatti 2019; Britz et al. 2021).

The KLM logics correspond to the flat fragments (i.e., without nested conditionals) of known conditional logics; e.g., **P** is related to Preferencial Conditional Logic (Burgess 1981), **C** to some extension of the basic normal conditional logic CK (Alenda, Olivetti, and Pozzato 2016), and **CM** to Lewis' counterfactual logic $\mathbb{V}$ (Lewis 1973). We plan to explore the extensions of our calculi that deal with nested edges as handled by KLMPropagator. The resulting calculi are expected to correspond to certain conditional logics, and our goal is to determine which ones.

From an automated deduction perspective, our prover built upon SMT solvers, can readily incorporate reasoning modulo theories, and deal with complex queries including Boolean combinations of conditionals or nested conditionals. However, it currently lacks logic-specific optimizations, such as preemptively checking and asserting simple entailments between nodes or deriving additional lemmas from axioms to expedite searches. Furthermore, it does not yet generate KLM proofs for derivable sequents, which poses a challenge in itself. We intend to address these issues in future iterations.

Acknowledgements

We thank the anonymous reviewers for their detailed and constructive feedback. The work is partially supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 101034440, the WWTF project ICT22-023, the FWF project I 6372-N (Grant-DOI 10.55776/I6372), and the MOSAIC project financed by the European Union's Marie Skłodowska-Curie grant No. 101007627.

References

Alenda, R.; Olivetti, N.; and Pozzato, G. L. 2016. Nested sequent calculi for normal conditional logics. *J. Log. Comput.* 26(1):7–50.

Areces, C.; Fontaine, P.; and Merz, S. 2015. Modal satisfiability via SMT solving. In Nicola, R. D., and Hennicker, R., eds., *Software, Services, and Systems - Essays Dedicated to Martin Wirsing on the Occasion of His Retirement from the Chair of Programming and Software Engineering*, volume 8950 of *LNCS*, 30–45. Springer.

Artosi, A.; Governatori, G.; and Rotolo, A. 2002. Labelled tableaux for nonmonotonic reasoning: Cumulative consequence relations. *J. Log. Comput.* 12(6):1027–1060.

Barbosa, H.; Barrett, C. W.; Brain, M.; Kremer, G.; Lachnitt, H.; Mann, M.; Mohamed, A.; Mohamed, M.; Niemetz, A.; Nötzli, A.; Ozdemir, A.; Preiner, M.; Reynolds, A.; Sheng, Y.; Tinelli, C.; and Zohar, Y. 2022. cvc5: A versatile and industrial-strength SMT solver. In Fisman, D., and Rosu, G., eds., *TACAS*, volume 13243 of *LNCS*, 415–442. Springer.

Biere, A.; Heule, M.; van Maaren, H.; and Walsh, T., eds. 2021. *Handbook of Satisfiability - Second Edition*, volume 336 of *Frontiers in Artificial Intelligence and Applications*. IOS Press.

Bjørner, N. S.; Eisenhofer, C.; and Kovács, L. 2023. Satisfiability modulo custom theories in Z3. In Dragoi, C.; Emmi, M.; and Wang, J., eds., *VMCAI*, volume 13881 of *LNCS*, 91–105. Springer.

Bochman, A. 2018. Actual causality in a logical setting. In Lang, J., ed., *IJCAI*, 1730–1736. ijcai.org.

Bochman, A. 2021. *A Logical Theory of Causality*. MIT Press.

Bonatti, P. 2019. Rational closure for all description logics. *Artificial Intelligence* 274:197–223.

Booth, R., and Varzinczak, I. 2021. Conditional inference under disjunctive rationality. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 6227–6234. AAAI Press.

Boutilier, C. 1990. Conditional logics of normality as modal systems. In Shrobe, H. E.; Dietterich, T. G.; and Swartout, W. R., eds., *Proceedings of the 8th National Conference on Artificial Intelligence. Boston, Massachusetts, USA, July 29 - August 3, 1990, 2 Volumes*, 594–599. AAAI Press / The MIT Press.

Britz, K., and Varzinczak, I. J. 2018. From klm-style conditionals to defeasible modalities, and back. *J. Appl. Non Class. Logics* 28(1):92–121.

Britz, K.; Casini, G.; Meyer, T.; Moodley, K.; Sattler, U.; and Varzinczak, I. 2021. Principles of KLM-style defeasible description logics. *ACM Trans. Comput. Log.* 22(1):1:1–1:46.

Britz, K.; Heidema, J.; and Labuschagne, W. A. 2009. Semantics for dual preferential entailment. *J. Philos. Log.* 38(4):433–446.

Burgess, J. 1981. Quick completeness proofs for some logics of conditionals. *Notre Dame Journal of Formal Logic* 22(3):76–84.

Casini, G., and Straccia, U. 2010. Rational closure for defeasible description logics. In Janhunen, T., and Niemelä, I., eds., *Logics in Artificial Intelligence*, 77–90. Berlin, Heidelberg: Springer Berlin Heidelberg.

Dantsin, E.; Eiter, T.; Gottlob, G.; and Voronkov, A. 2001. Complexity and expressive power of logic programming. *ACM Comput. Surv.* 33(3):374–425.

de Moura, L. M., and Bjørner, N. S. 2008. Z3: an efficient SMT solver. In Ramakrishnan, C. R., and Rehof, J., eds., *TACAS*, volume 4963 of *LNCS*, 337–340. Springer.

Eisenhofer, C.; Alassaf, R.; Rawson, M.; and Kovács, L. 2023. Non-classical logics in satisfiability modulo theories. In Ramanayake, R., and Urban, J., eds., *TABLEAUX*, volume 14278 of *LNCS*, 24–36. Springer.

Fiorentini, C.; Goré, R.; and Graham-Lengrand, S. 2019. A proof-theoretic perspective on smt-solving for intuitionistic propositional logic. In Cerrito, S., and Popescu, A., eds., *TABLEAUX*, volume 11714 of *LNCS*, 111–129. Springer.

Gabbay, D. M. 1985. Theoretical foundations for non-monotonic reasoning in expert systems. In Apt, K. R., ed., *Logics and Models of Concurrent Systems*, 439–457. Berlin, Heidelberg: Springer Berlin Heidelberg.

Gelati, J.; Rotolo, A.; Sartor, G.; and Governatori, G. 2004. Normative autonomy and normative co-ordination: Declarative power, representation, and mandate. *Artif. Intell. Law* 12(1-2):53–81.

Giordano, L., and Gliozzi, V. 2021. A reconstruction of multipreference closure. *Artificial Intelligence* 290:103398.

Giordano, L.; Gliozzi, V.; Olivetti, N.; and Pozzato, G. L. 2007. An implementation of a free-variable tableaux for KLM preferential logic P of nonmonotonic reasoning: The theorem prover FreeP 1.0. In Basili, R., and Pazienza, M. T., eds., *AI*IA*, volume 4733 of *LNCS*, 84–96. Springer.

Giordano, L.; Gliozzi, V.; Olivetti, N.; and Pozzato, G. L. 2009. Analytic tableaux calculi for KLM logics of nonmonotonic reasoning. *ACM Trans. Comput. Log.* 10(3):18:1–18:47.

Giordano, L.; Dupré, D. T.; Fiorentini, C.; Momigliano, A.; and Pettorossi, A. 2018. Defeasible reasoning in SROEL: from rational entailment to rational closure. *Fundamenta Informaticae* 161(1–2):135–161.

Giordano, L.; Gliozzi, V.; and Pozzato, G. L. 2007. KLMLean 2.0: A theorem prover for KLM logics of nonmonotonic reasoning. In Olivetti, N., ed., *TABLEAUX*, 238–244. Berlin, Heidelberg: Springer Berlin Heidelberg.

Governatori, G., and Rotolo, A. 2008. A computational framework for institutional agency. *Artif. Intell. Law* 16(1):25–52.

Kraus, S.; Lehmann, D.; and Magidor, M. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artif. Intell.* 44(1):167–207.

Lehmann, D., and Magidor, M. 1992. What does a conditional knowledge base entail? *Artif. Intell.* 55(1):1–60.

Lehmann, D. 1999. Another perspective on default reasoning. *Annals of Mathematics and Artificial Intelligence* 15.

Lellmann, B. 2021. From input/output logics to conditional logics via sequents - with provers. In Das, A., and Negri, S., eds., *TABLEAUX*, volume 12842 of *LNCS*, 147–164. Springer.

Lewis, D. K. 1973. *Counterfactuals*. Malden, Mass.: Blackwell.

Makinson, D., and van der Torre, L. W. N. 2001. Constraints for input/output logics. *J. Philos. Log.* 30(2):155–185.

Olivetti, N., and Pozzato, G. L. 2014. NESCOND: an implementation of nested sequent calculi for conditional logics. In Demri, S.; Kapur, D.; and Weidenbach, C., eds., *IJCAR*, volume 8562 of *LNCS*, 511–518. Springer.

Shoham, Y. 1987. A semantical approach to non-monotonic logics. In *LICS*, 275–279. IEEE Computer Society Press.

Sun, X., and Robaldo, L. 2017. On the complexity of input/output logic. *J. Appl. Log.* 25:69–88.