Optimal Tableaux for Conditional Logics with Cautious Monotonicity[1]

Lutz Schröder [2] and **Dirk Pattinson** [3] and **Daniel Hausmann**[4]

Abstract. Conditional logics capture default entailment in a modal framework in which non-monotonic implication is a first-class citizen, and in particular can be negated and nested. There is a wide range of axiomatizations of conditionals in the literature, from weak systems such as the basic conditional logic CK, which allows only for equivalent exchange of conditional antecedents, to strong systems such as Burgess' system $\mathcal{S}$, which imposes the full Kraus-Lehmann-Magidor properties of preferential logic. While tableaux systems implementing the actual complexity of the logic at hand have recently been developed for several weak systems, strong systems including in particular disjunction elimination or cautious monotonicity have so far eluded such efforts; previous results for strong systems are limited to semantics-based decision procedures and completeness proofs for Hilbert-style axiomatizations. Here, we present tableaux systems of optimal complexity PSPACE for several strong axiom systems in conditional logic, including system $\mathcal{S}$; the arising decision procedure for system $\mathcal{S}$ is implemented in the generic reasoning tool CoLoSS.

1 Introduction

A recurring theme in the formal representation of common-sense reasoning is non-monotonic entailment, where one expects those conclusions to hold that are not invalidated by the premise. To avoid the penguins vs. birds cliché, a typical example is the assertion that John *normally* goes to work on Mondays, which can be formalized as a non-monotonic conditional

$$\mathsf{Monday} \Rightarrow \mathsf{work}$$

where we use $\Rightarrow$ to denote a defeasible implication 'if – then normally'. The non-monotonicity manifests itself in the fact that the conclusion is not maintained if more information becomes available, such as that on some given Monday, John is sick. That is, our judgement that John normally goes to work on Mondays should be consistent with

$$\neg(\mathsf{Monday} \wedge \mathsf{sick} \Rightarrow \mathsf{work}).$$

Situations such as the above are commonplace, and non-monotonic reasoning has maintained a core position in artificial intelligence ever since the first non-monotonic systems were proposed in the late 70s and early 80s [15, 19]. Besides the philosophical interest in studying the formal basis of common-sense reasoning, there is a growing interest in implementations of non-monotonic systems for dealing with real-life knowledge bases and semantic web applications. (See, e.g., the special session on NMR Systems and Applications in [8].)

While there is a substantial body of work concerning automated reasoning in *flat* non-monotonic logics (without nesting of non-monotonic implications) there are surprisingly few results that deal with the general case (where non-monotonic implications may be nested), in particular if strong reasoning principles such as cautious monotony are adopted. If we categorize non-monotonic logics according to

- whether non-monotonic implications may be nested, and
- which set of reasoning principles is supported,

the present work fills a gap in the literature on *conditional logics* — which allow for nesting and Boolean combination of conditionals — supporting strong reasoning principles, for which we provide tight complexity bounds and tableau algorithms.

It is well-known that weaker approaches that treat non-monotonicity as an external entailment relation can be encoded in the more expressive framework of conditional logic (see, e.g., [7, 2]). We shall refer to non-monotonic entailment systems as *flat* non-monotonic logics. Nested conditionals play a prominent role, e.g., in belief revision [3] to model update of conditional knowledge bases by new conditional beliefs. To give a second example, defeasible implications can be used to model combinations of generic and enterprise-specific rules: if your company has specific rules saying you should not discuss your salary with others, you should, as a general rule, be discrete about your payslip and not, e.g., leave it lying around on your desk (although this does not, strictly speaking, logically follow from the injunction on *discussing* your salary with your colleagues). Clearly, both these rules only apply in default situations as we need to divulge our salary if we ask for a pay rise, and (maybe unfortunately) need to disclose our payslip to the taxman. We can use the nested conditional

$$(\mathsf{employed} \Rightarrow \neg\mathsf{discuss\_salary}) \Rightarrow (\mathsf{discrete\_about\_payslip})$$

to specify such situations. A similar example ('documents whose disclosure would normally harm the interests of the company are normally confidential') is discussed in [21].

Throughout, we discuss extensions of the minimal conditional logic $\mathbf{CK}$ [6]. The only reasoning principles supported by $\mathbf{CK}$ are normality in the right hand argument of the conditional (the *consequent*), and replacement of equivalents on the left-hand side (the *antecedent*). Widely accepted additional reasoning principles have emerged in the so-called KLM axioms [13] that stipulate the identity axiom $A \Rightarrow A$, *disjunction elimination*, which may be expressed in conditional logic as

$$(A \Rightarrow C) \wedge (B \Rightarrow C) \to (A \vee B) \Rightarrow C,$$

[1] Work by the first author supported by BMBF grant FormalSafe (FKZ 01IW07002); work by the second author supported by EPSRC grant EP/F031173/1.

[2] SSCS, DFKI Bremen and Dept. of Comput. Sci., Univ. of Bremen
[3] Department of Computing, Imperial College London
[4] SSCS, DFKI Bremen

and *cautious monotonicity*

$$(A \Rightarrow B) \wedge (A \Rightarrow C) \to (A \wedge B) \Rightarrow C,$$

where $\to$ denotes material implication. We refer to systems that support one of the latter two axioms, which allow for manipulation of conditional antecedents, as *strong* conditional logics, and to systems that support only manipulation of conditional consequents (as discussed, e.g., in [17]) as *weak* conditional logics. Cautious monotony is a particularly compelling principle. Continuing our logical discussion of John's attendance record at work, cautious monotonicity would allow us to conclude

$$\mathsf{Monday} \wedge \mathsf{sick} \Rightarrow \mathsf{work}$$

after all if we were given the additional information that John is normally sick on Mondays ($\mathsf{Monday} \Rightarrow \mathsf{sick}$), say because he tends to be hung over after the weekend – given that we cannot very well assume that John normally skips work on Mondays, this is indeed quite realistic. The conditional logic implementing the KLM axioms is known as *system* $\mathcal{S}$ [4].

Contributions of the present work. We develop *unlabelled* tableau systems (which in general are structurally simpler and easier to implement than labelled systems) for strong systems of conditional logic, specifically for system $\mathcal{S}$ and weaker systems obtained by dropping disjunction elimination. All these systems admit proof search in PSPACE, thereby for the first time matching the known complexity bound [9] within a tableau system in the case of system $\mathcal{S}$, and establishing new bounds in the other cases. We obtain these systems as duals of cut-free sequent systems which arise by application of generic modal principles [18] to suitable sets of modal rules for the nesting-free fragment. Clearly, the crucial step here is to find these rules, and indeed their description turns out to be non-trivial. The complexity bounds follow by application of a generic criterion first published in [16], which we develop further in the present work by providing a purely syntactic version of it. We have implemented the tableau procedure for system $\mathcal{S}$ within the framework of the generic modal reasoner CoLoSS [5]; to our knowledge, this constitutes the first implemented reasoner for a strong conditional logic.

Related work. Despite the widespread acceptance of the KLM axioms as the basic set of non-monotonic reasoning principles, there is to date only little work on tableau calculi for logics containing these axioms, which would be seen as a prerequisite for efficient automated reasoning. Specifically, optimal tableau systems have been developed in [10] for the flat non-monotonic logics considered in [13]; however, all of these systems need to employ extensions of the language by an additional operator. Earlier, a labelled tableau calculus of unclear complexity for one of these logics has been given [1]. At the level of full conditional logics, optimal tableau systems for several weak systems have been presented in [17, 18]. The only tableau system for a strong system of conditional logic that we are aware of is a *labelled* tableau system for the conditional logic CE [11] (corresponding in the terminology of [9] to conditional logic with the uniformity property); it is presently unclear whether this system matches known complexity bounds.

2 Preliminaries on Conditional Logics

Conditional logic is based on a single binary modal operator $\Rightarrow$, the *conditional* operator. Formulas of conditional logic are given by the grammar

$$\mathcal{F} \ni A, B ::= \bot \mid a \mid \neg A \mid A \wedge B \mid A \Rightarrow B$$

where a ranges over a set P of *propositional atoms*. The reading of $A \Rightarrow B$ is 'if A, then normally B'. This grammar allows for Boolean combination and nesting of conditionals, the hallmark of conditional logic. We refer to the left-hand argument A of a conditional $A \Rightarrow B$ as its *antecedent*, and to the right-hand argument B as its *consequent*.

The semantics of conditional logics can be given in terms of either *selection functions* [6] or local *preference orderings* and related structures [4, 9], depending on the postulated reasoning principles. We do not elaborate details here, but shall return to preference semantics when we discuss the tableau system for system $\mathcal{S}$. We consider various axiomatizations for conditional logics, which we formulate in single-sided sequent style. Thus, a *sequent* Γ is a multiset of formulas, to be read disjunctively. As usual, we let a formula denote also its singleton sequent and denote multi-set union by comma-separated juxtaposition; combining these notations, Γ, B denotes the multiset union of Γ and $\{B\}$. Moreover, we regard sets as multisets where all elements have multiplicity 1. All our systems extend the basic rules of $\mathbf{CK}$, namely replacement of equivalents,

$$\frac{A_1 = A_2 \quad B_1 = B_2}{\neg(A_1 \Rightarrow B_1), A_2 \Rightarrow B_2},$$

where $A_1 = A_2$ abbreviates the sequents $\neg A_1, A_2$ and $A_1, \neg A_2$ throughout, and normality of consequents,

$$\frac{A_0 = A_1 = A_2 \quad \neg B_1, \neg B_2, B_0}{\neg(A_1 \Rightarrow B_1), \neg(A_2 \Rightarrow B_2), A_0 \Rightarrow B_0} \qquad \frac{B}{A \Rightarrow B}$$

Further systems will be defined by the addition of more sequent rules of this type, explicitly:

Definition 1. A *literal* over a set L is an element of $L \cup \neg L$, where $\neg L = \{\neg a \mid a \in L\}$. A *sequent over* L is a multiset over $L \cup \neg L$. We write $\Rightarrow\!(L)$ for the set of formal terms $\{a \Rightarrow b \mid a, b \in L\}$. A *one-step rule* $\Gamma_1 \dots \Gamma_n / \Gamma_0$ consists of sequents $\Gamma_1, \dots, \Gamma_n$ over the set P of propositional atoms (abused as variables) and a sequent Γ_0 over $\Rightarrow\!(P)$ such that every propositional atom appears at most once in Γ_0 (so that Γ_0 is actually a set). An *L-instance* of $\Gamma_1 \dots \Gamma_n / \Gamma_0$ is a rule $\Gamma_1\sigma \dots \Gamma_n\sigma / \Gamma_0\sigma, \Delta$ obtained by applying a substitution $\sigma : P \to L$ and a *weakening context* Δ.

In particular, the above rules for $\mathbf{CK}$ are one-step rules. Most conditional logics in the literature are *rank-1*, i.e. can be defined by a system of one-step rules (one exception being conditional modus ponens $(A \Rightarrow B) \to A \to B$, which however is obviously incompatible with a view of the conditional as an internalized default implication). Below, when we refer to a *(conditional) rule system* $\mathcal{R}$, we mean a set of one-step rules that extends the one-step rules of $\mathbf{CK}$.

All systems moreover contain a fixed propositional part consisting of the standard rules

$$\frac{}{\Gamma, a, \neg a} \quad \frac{\Gamma, \neg A, \neg B}{\Gamma, \neg(A \wedge B)} \quad \frac{\Gamma, A \quad \Gamma, B}{\Gamma, A \wedge B} \quad \frac{\Gamma, A}{\Gamma, \neg\neg A}$$

where the leftmost rule is called the *axiom rule* (a generalized version of which with formulas A in place of atoms a is admissible). The set of sequents *derivable* in a rule system from a set of assumptions is the smallest set containing the assumptions and closed under application of substitution instances of the rules; here, the given one-step rules are applied as $\mathcal{F}$-instances $\Gamma_1\sigma \dots \Gamma_n\sigma / \Gamma_0\sigma, \Delta$ according to Definition 1, i.e. with weakening built in. More generously, a sequent is *cut-derivable* if it can be derived using additionally the *cut rule*

$$\frac{\Gamma, A \quad \Gamma, \neg A}{\Gamma}$$

— a rule that is evidently undesirable from the perspective of proof search, as its backwards application requires the invention of A. The main point of the analysis below is to give criteria for the cut rule to be *admissible* under the remaining rules, so that proof search algorithms can proceed without having to try the cut rule.

A sequent system immediately gives rise to a dual *tableau system* which is obtained by dualizing all rules (and, conventionally, turning them upside down). E.g., one has the tableau rule

$$\frac{\Gamma, \neg(A \wedge B)}{\Gamma, \neg A \quad \Gamma, \neg B}$$

where multisets of formulas are now called *labels* and read conjunctively, and a satisfiability proof of the premise $\Gamma, \neg(A \wedge B)$ demands a satisfiability proof for either one of the two conclusions $\Gamma, \neg A$ or $\Gamma, \neg B$. *We shall present all rule systems below as sequent systems, on the understanding that tableau systems are presented dually.*

We inherit the driving principle of our analysis of conditional systems from *coalgebraic modal logic* [18]: one can reduce properties of a full modal logic with nested modalities to its *one-step logic*, characterized syntactically by excluding nested modalities as well as lonely propositional atoms. Formally, formulas A, B of the one-step fragment of conditional logic are defined by the grammar

$$A, B ::= \bot \mid \neg A \mid A \wedge B \mid \alpha \Rightarrow \beta$$

where α and β range over Boolean combinations of propositional atoms. As shown in [20], such formulas may equivalently be presented as one-step rules. E.g., monotonicity of $\Rightarrow$ in the second argument may be expressed either by the one-step formula $(A \Rightarrow (B \wedge C)) \rightarrow (A \Rightarrow B)$ or equivalently by the one-step rule

$$\frac{A_1 = A_2 \qquad B \rightarrow C}{(A_1 \Rightarrow B) \rightarrow (A_2 \Rightarrow C)}.$$

Definition 2. We say that a one-step rule $\Gamma_1 \dots \Gamma_n / \Gamma_0$ is *cut-derivable* in $\mathcal{R}$ if Γ_0 is cut-derivable from $\Gamma_1 \dots \Gamma_n$ in $\mathcal{R}$ using $\mathsf{Prop}(P)$-instances of the one-step rules, and *derivable* if only cuts between purely propositional formulas are needed in the derivation (which then necessarily uses only a P-instance of a single one-step rule). We say that $\mathcal{R}$ *admits one-step cut-elimination* if every cut-derivable one-step rule is derivable.

Example 3. [18] It is easily checked that the rules for **CK** as presented above do *not* admit one-step cut elimination. The following alternative rule does admit one-step cut elimination (see Example 6):

$$(CK) \quad \frac{A_0 = \dots = A_n \quad \neg B_1, \dots, \neg B_n, B_0}{\neg(A_1 \Rightarrow B_1), \dots, \neg(A_n \Rightarrow B_n), (A_0 \Rightarrow B_0)} \; n \geq 0.$$

Note that the rule scheme (CK), which is subsumed by all rule schemes to be introduced below, subsumes replacement of equivalents in both arguments.

Definition 4. [18] We say that $\mathcal{R}$ *absorbs contraction* if for every rule $\Gamma_1 \dots \Gamma_n / \Gamma_0 \in \mathcal{R}$ and every renaming $\sigma : P \rightarrow P$, there exists a rule $\Delta_1 \dots \Delta_m / \Delta_0 \in \mathcal{R}$ and a renaming $\rho : P \rightarrow P$ such that $\Delta_0 \rho \subseteq \Gamma_0 \sigma$, ρ does not identify any literals in Δ_0, and for $j = 1, \dots, m$, Δ_j is derivable from $\{\Gamma_1, \dots, \Gamma_n\}$ using the propositional rules and cut.

Moreover, $\mathcal{R}$ additionally *absorbs cut* if for all rules $\Gamma_1 \dots \Gamma_n / \Gamma_0$ and $\Delta_1 \dots \Delta_m / \Delta_0$ in $\mathcal{R}$ and all injective renamings $\sigma, \rho : P \rightarrow P$ such that $\Gamma_0 \sigma = \Gamma, A$ and $\Delta_0 \rho = \Delta, \neg A$ where Γ and Δ do not share any propositional atoms, there exists a rule $\Sigma_1 \dots \Sigma_l / \Sigma_0$ and an injective renaming $\kappa : P \rightarrow P$ such that $\Sigma_0 \kappa \subseteq \Gamma, \Delta$ and for all $j = 1, \dots, l$, $\Sigma_j \kappa$ can be derived from $\{\Gamma_1 \sigma, \dots, \Gamma_n \sigma, \Delta_1 \rho, \dots, \Delta_m \rho\}$ using the propositional rules and cut.

Proposition 5. *[18] A rule system admits one-step cut elimination if it absorbs cut and contraction.*

Example 6. It is easy to see that the rule set (CK) of Example 3 absorbs contraction. To see that it absorbs cut, we have to consider two instances ('left' and 'right') of the rule, where we prime n and all propositional atoms in the right hand instance but assume that $A_0 \Rightarrow B_0$ equals one of the negative literals on the right-hand side, w.l.o.g. $A'_1 \Rightarrow B'_1$. We then need a rule that proves $\neg(A_1 \Rightarrow B_1), \dots, \neg(A_n \Rightarrow B_n), \neg(A'_2 \Rightarrow B'_2), \dots, \neg(A'_{n'} \Rightarrow B'_{n'}), A'_0 \Rightarrow B'_0$. This has the right form for a conclusion of an instance of (CK), so that we are left with proving the corresponding premises from the ones of the original rule instances (by propositional reasoning with cut). This is straightforward; e.g. we prove the premise $\neg B_1, \dots, \neg B_n, \neg B'_2, \dots, \neg B'_{n'}, B'_0$ by cutting the corresponding premises of the original two rule instances.

The full logic inherits cut elimination from the one-step logic:

Theorem 7. *[18] If a rule system admits one-step cut elimination, then it* admits cut elimination*, i.e. whenever a sequent is cut-derivable from a set of assumptions, then it is derivable.*

Similarly, we may reduce the standard complexity of the full logic to the complexity of the one-step logic, employing an approach where we equip decision procedures for the one-step logic with discounted space for input in the same way as standardly used for logspace Turing machines. The criterion presented below is a novel, purely syntactic variant of a semantics-based criterion from [16].

Definition 8. The *strict one-step derivability problem* of $\mathcal{R}$ is to decide whether a one-step rule $\Gamma_1 \dots \Gamma_n / \Gamma_0$ is cut-derivable in $\mathcal{R}$, where the input size of the problem is defined to be the size of Γ_0 alone (note that the combined premises may be of exponential size) and the premises $\Gamma_1 \dots \Gamma_n$ are assumed to be stored on an input tape that does not count towards overall space consumption.

Theorem 9. *If the strict one-step derivability problem is in* PSPACE*, then cut-derivability of sequents in the full logic from the empty set of assumptions is in* PSPACE.

Proof sketch. By a non-deterministic polynomial-space algorithm (exploiting NPSPACE =PSPACE) where a sequent is first analysed propositionally until one arrives at a sequent over $\Rightarrow(\mathcal{F})$, and then decomposed into a topmost layer of conditionals and the formulas below these conditionals. One then peels off one layer of conditionals by calling a PSPACE solver for the strict one-step derivability problem. Here, the input tape of the one-step derivability checker is emulated by means of recursive calls back to the global derivability algorithm; thus, one keeps space consumption polynomial, in analogy to the composition of logspace functions. □

Having recalled the necessary ingredients of the generic theory, we now proceed to apply them to the conditional logics of interest.

3 A Cut-Free Sequent System for CK+CM

The first system we consider is the extension of minimal conditional logic **CK** with cautious monotonicity alone. As a one-step rule, cautious monotonicity becomes

$$\frac{A_0 = A_2 \wedge B_1 \quad A_1 = A_2 \quad B_0 = B_2}{\neg(A_1 \Rightarrow B_1), \neg(A_2 \Rightarrow B_2), (A_0 \Rightarrow B_0)}. \tag{1}$$

The resulting system $\mathbf{CK} + \mathbf{CM}$ is weaker than the conditional logic corresponding to the cumulative logic of [13]; we study it here both due to the evident semantics-independent interest in the cautious monotonicity axiom as such and with a view to demonstrating a case that can be handled purely syntactically – in fact, the only known semantics for $\mathbf{CK} + \mathbf{CM}$ is selection function semantics, which is not terribly helpful in detecting good rule sets. In the present case, one finds a rule set that absorbs cut by repeated application of cut to the original rules. The result is the following.

A *compatibility tree* over a non-empty finite index set I with $0 \notin I$ is a finite tree T with nodes v labelled by non-empty sets $l(v) \subseteq I$ such that the root is labelled by I and the label of any node is the union of the labels of its children; i.e. T represents a hierarchical decomposition of I into unions of subsets. For each such T, we have a rule $(comp_T)$:

$$\frac{\begin{array}{ll} A_i = A_j & \text{for } i,j \in l(v), v \text{ leaf} \\ \bigcup_{i\in l(v)}\{\neg B_i, \neg A_i\}, A_j & \text{for } v \text{ child of } w \text{ and } j \in l(w) \\ \{\neg B_i \mid i \in I\}, B_0 & \\ \bigcup_{i\in I}\{\neg B_i, \neg A_i\}, A_0 & \\ \neg A_0, A_i & \text{for } i \in I \end{array}}{\{\neg(A_i \Rightarrow B_i) \mid i \in I\}, (A_0 \Rightarrow B_0)}$$

The core idea of this rule is that under its premises, one can derive

$$\{\neg(A_i \Rightarrow B_i) \mid i \in I\}, (\textstyle\bigwedge_{j\in l(v)} A_j \Rightarrow \bigwedge_{j\in l(v)} B_j) \tag{2}$$

for every node v of T, as shown by well-founded induction on v.

We note that we can clearly restrict the second premise to cases where $j \notin l(v)$, as instances for $j \in l(v)$ are discharged by the axiom rule. Moreover, we assume w.l.o.g. that all unions of labels represented by T are irredundant (i.e. no set can be omitted from the union without decreasing its resulting set).

Both the rule (CK) of Example 3 and the one-step rule (1) for cautious monotonicity are derivable from rules $(comp_T)$ for trivial compatibility trees that consist only of the root. Conversely, we have to prove that every rule $(comp_T)$ is actually derivable:

Lemma 10. *Every rule $(comp_T)$ induced by a compatibility tree T is derivable in $\mathbf{CK} + \mathbf{CM}$.*

Proof sketch. By (2), we have in particular that $\Gamma, (\bigwedge_{i\in I} A_i \Rightarrow \bigwedge_{i\in I} B_i)$ is derivable. Then we obtain $\Gamma, (\bigwedge_{i\in I} A_i \Rightarrow \bigwedge_{i\in I} A_i \rightarrow A_0)$ by monotonicity on the right and the fourth premise, thus $\Gamma, (A_0 \Rightarrow \bigwedge_{i\in I} B_i)$ by CM and the fifth premise, and finally $\Gamma, (A_0 \Rightarrow B_0)$ by monotonicity and the third premise. (The mentioned induction for (2) is similar.) ☐

Next, we apply the generic techniques discussed in Section 2 to prove that the rule set engenders a sequent system that admits cut.

Lemma 11. *The set of rules $(comp_T)$ induced by compatibility trees T absorbs cut.*

Proof sketch. Assume we have a left-hand instance of the rule as above, and a right-hand instance with all entities primed, where $A_0 \equiv A'_{i_0}$ and $B_0 \equiv B'_{i_0}$ for some $i_0 \in I'$. We have to find a rule instance proving

$$\{\neg(A_i \Rightarrow B_i) \mid i \in I\}, \{\neg(A'_j \Rightarrow B'_j) \mid i_0 \neq j \in I'\}, (A'_0 \Rightarrow B'_0).$$

To this end, we construct a compatibility tree U over J with labelling l_U, where $J = I \cup I' - \{i_0\}$, by modifying T' as follows, mimicking substitution of I for $\{i_0\}$ in the corresponding union term:

- Replace i_0 by the elements of I in every label of T'.
- Whenever v is a leaf of T' with $l'(v) = \{i_0\}$, then attach T as a subtree at v, with the root of T replacing v.
- Whenever v is a leaf of T' with $\{i_0\} \subsetneq l'(v)$, then attach T as a subtree, with the root of T becoming a child of v, and create a second child of v labelled $l'(v) - \{i_0\}$.

It is then straightforward to derive the premises of the rule instance for U from the ones for T and T'. ☐

Lemma 12. *The set of rules $(comp_T)$ induced by compatibility trees T absorbs contraction.*

Proof sketch. Assume $1, 2 \in I$ and identify A_1 and B_1 with A_2 and B_2, respectively, in a generic instance of rule $(comp_T)$ (it is clear that considering the case of a single identified pair is sufficient). We obtain a new rule conclusion

$$\{\neg(A_i \Rightarrow B_i) \mid i \in I - \{1\}\}, (A_0 \Rightarrow B_0).$$

A compatibility tree U over the new index set $I - \{1\}$ is constructed by replacing 1 with 2 in all labels $l(v)$ of T. It is easy to check that this is indeed a compatibility tree, and that the premise of the rule instance for U follows from the one for T. ☐

Summing up, *the set of rules $(comp_T)$ admits one-step cut elimination*, and hence

Corollary 13. *The sequent system for $\mathbf{CK} + \mathbf{CM}$ induced by the rules $(comp_T)$ admits cut elimination.*

3.1 Proof Search in PSPACE

By Theorem 9, it suffices to prove that the strict one-step derivability problem of $\mathbf{CK} + \mathbf{CM}$ is in PSPACE to establish that proof search can be performed in PSPACE. Since our rules for $\mathbf{CK} + \mathbf{CM}$ admit one-step cut elimination, we only have to check for applicability of a single rule $(comp_T)$ to the conclusion of a given one-step rule $\Gamma_1 \dots \Gamma_n / \Gamma_0$ to decide its derivability (with or without cut). Assuming that the premises are stored on an input tape, we can do this using polynomial space in Γ_0 by traversing a compatibility tree. Explicitly, the decision procedure for derivability of $\Gamma_1 \dots \Gamma_n / \Gamma_0$ is as follows.

Algorithm 14. 1. Guess a subsequent Γ'_0 of Γ_0 of the form $\Gamma'_0 = \{\neg(A_i \Rightarrow B_i) \mid i \in I\}, (A_0 \Rightarrow B_0)$.

2. Check that $\{\neg B_i \mid i \in I\}, B_0$; $\bigcup_{i\in I}\{\neg B_i, \neg A_i\}, A_0$; and $\neg A_0, A_i$ (for all $i \in I$) are propositionally entailed by $\{\Gamma_1, \dots, \Gamma_n\}$. (This can be done without reading all premises from the input tape at once; instead, one goes through all valuations κ of the propositional atoms occurring in the rule, checks — in logarithmic space by [14] — whether κ validates the Γ_i for $i \geq 1$, and in that case checks that κ validates also the respective target formula.)

3. Check by means of Algorithm 15 that $\Gamma_1 \dots \Gamma_n / \{\neg(A_i \Rightarrow B_i) \mid i \in I\}, (\bigwedge_{i\in I} A_i \Rightarrow \bigwedge_{i\in I} B_i)$ is one-step derivable.

The last step uses the following recursive procedure.

Algorithm 15. (Check that $\Gamma_1 \dots \Gamma_n / \{\neg(A_i \Rightarrow B_i) \mid i \in I\}, (\bigwedge_{j\in J} A_j \Rightarrow \bigwedge_{j\in J} B_j)$ is one-step derivable for $J \subseteq I$.)

1. Check whether $A_j = A_k$ is propositionally entailed by $\{\Gamma_1, \dots, \Gamma_n\}$ for all $j, k \in J$. If yes, answer 'yes', otherwise:

2. Guess an irredundant decomposition $J = \bigcup_{k \in K} J_k$.

3. Check that for each $k \in K$ and each $j \in J$, $\bigcup_{i \in J_k} \{\neg B_i, \neg A_i\}, A_j$ is propositionally entailed by $\{\Gamma_1, \dots, \Gamma_n\}$.

4. Check recursively that $\Gamma_1 \dots \Gamma_n / \{\neg(A_i \Rightarrow B_i) \mid i \in I\}, (\bigwedge_{j \in J_k} A_j \Rightarrow \bigwedge_{j \in J_k} B_j)$ is one-step cut-free derivable for each $k \in K$.

It is clear that both algorithms use polynomial space in Γ_0, noting in the case of Algorithm 15 that the depth of the recursion is at most $|I|$. The algorithms can also be phrased in terms of a one-step proof calculus as follows: the initial steps in Algorithm 14 correspond to a single application of the rule

$$\frac{\Gamma, (A \Rightarrow B) \quad \neg B, \neg A, C \quad \neg C, A \quad \neg B, D}{\Gamma, (C \Rightarrow D)}$$

to $A = \bigwedge_{i \in I} A_i$, $B = \bigwedge_{i \in I} B_i$, and $\Gamma = \{\neg(A_i \Rightarrow B_i) \mid i \in I\}$, while the recursive procedure in Algorithm 15 corresponds to iterated application of the rule

$$\frac{\begin{array}{c}\Gamma, (\bigwedge_{j \in J_k} A_j \Rightarrow \bigwedge_{j \in J_k} B_j) \text{ for all } k \in K \\ \bigcup_{j \in J_k} \{\neg B_j, \neg A_j\}, A_i \text{ for all } k \in K, i \in J\end{array}}{\Gamma, (\bigwedge_{j \in J} A_j \Rightarrow \bigwedge_{j \in J} B_j)} \quad (J = \bigcup J_k)$$

finished at the base of the recursion by application of the rule

$$\frac{A_i = A_j \text{ for } i, j \in I}{\{\neg(A_i \Rightarrow B_i) \mid i \in I\}, (\bigwedge_{i \in I} A_i \Rightarrow \bigwedge_{i \in I} B_i)}.$$

4 Cautious Monotonicity and the Identity Axiom

Our next conditional logic, $\mathbf{CK} + \mathbf{CM} + \mathbf{ID}$, is obtained from $\mathbf{CK} + \mathbf{CM}$ by adding the identity axiom $A \Rightarrow A$. The absorbing rule system consists of the expected modification of the rule for $\mathbf{CK} + \mathbf{CM}$,

$$\frac{\begin{array}{ll} A_i = A_j & \text{for } i, j \in l(v), v \in T \text{ leaf} \\ \bigcup_{i \in l(v)} \{\neg B_i, \neg A_i\}, A_j & \text{for } v \in T \text{ and } j \in l(w) \text{ for} \\ & \text{some successor } w \text{ of } v \\ \{\neg B_i \mid i \in I\}, \neg A_0, B_0 & \\ \bigcup_{i \in I} \{\neg B_i, \neg A_i\}, A_0 & \\ \neg A_0, A_i & \text{for } i \in I \end{array}}{\{\neg(A_i \Rightarrow B_i) \mid i \in I\}, (A_0 \Rightarrow B_0)},$$

referred to as $(comp_T^{ID})$ and again indexed over compatibility trees T over I, where we continue to insist that I and all labels of T be non-empty, and additionally the rule

$$(ID) \quad \frac{\neg A, B}{A \Rightarrow B}$$

(obtained by cutting identity $A = B / A \Rightarrow B$ with monotony on the right). In the rule $(comp_T^{ID})$, we may again assume $j \notin l(v)$ in the second premise.

Lemma 16. *The above rule set absorbs cut and contraction.*

Thus, the results already obtained for $\mathbf{CK} + \mathbf{CM}$ extend to $\mathbf{CK} + \mathbf{CM} + \mathbf{ID}$:

Corollary 17. *The sequent system for* $\mathbf{CK} + \mathbf{CM} + \mathbf{ID}$ *given by the rules* $(comp_T^{ID})$ *and* (ID) *admits cut elimination and proof search in* PSPACE.

5 A Cut-Free Sequent Calculus for System $\mathcal{S}$

We now turn to the design of a cut-free sequent calculus with PSPACE proof search for full system $\mathcal{S}$ [4], the conditional logic counterpart of the preferential logic of Kraus, Lehmann, and Magidor [13]. That is, system $\mathcal{S}$ extends $\mathbf{CK}$ by cautious monotony, the identity axiom, and disjunction elimination, which as a one-step rule becomes

$$\frac{\neg A_0, A_1, A_2 \quad \neg A_1, A_0 \quad \neg A_2, A_0 \quad B_0 = B_1 = B_2}{\neg(A_1 \Rightarrow B_1), \neg(A_2 \Rightarrow B_2), (A_0 \Rightarrow B_0)}.$$

We change tack in comparison to the previous sections in that now, we take the lead to an absorbing rule set from the semantics, specifically from the small model property for one-step formulas proved in [9]. More specifically, we use the fact (proved in [9]) that universal validity of a sequent $\Gamma_0 = \neg(A_1 \Rightarrow B_1), \dots, \neg(A_n \Rightarrow B_n), (A_0 \Rightarrow B_0)$ is equivalent to validity in all *finite linear preferential models*. To formalise this as a sequent rule, we define an *S-structure for* Γ_0 to be a pair $M = (S, \preceq)$ where $S \subseteq \hat{I} = \{0, 1, \dots, n\}$, $0 \in S$, and $\preceq$ is a linear pre-order on S (i.e. for $i, k \in S$, $i \preceq k$ or $k \preceq i$) with greatest element 0. The state $i \in S$ is intended as a smallest A_i-state under $\preceq$, and $i \notin S$ signifies the non-existence of an A_i-state. We write $k \prec i$ for $k \preceq i \wedge i \not\preceq k$, $k \simeq i$ for $k \preceq i \wedge i \preceq k$, and in writing $k \preceq i$ etc. understand that $i, k \in S$. For $[i] \in S/\simeq$, we define a sequent $\Delta_M[i]$ by

$$\Delta_M[i] \equiv \bigcup_{k \simeq i} \{\neg A_k, C_k\}, \{A_j \mid i \prec j \vee j \notin S\}$$

where $C_0 \equiv B_0$, and $C_k = \neg B_k$ for $k \neq 0$. Intuitively speaking, every such $\Delta_M[i]$ expresses that M does not expand to a model violating Γ_0 in the intended fashion, by stating that if the state $[i]$ behaves according to its role in a putative model of $\neg \bigvee \Gamma_0$ (i.e. satisfies $A_k \wedge C_k$ for all $k \simeq i$), then either one of the states $[j]$ above it is not a minimal A_j-state, or $[i]$ is an A_j-state for some $j \notin S$. If we write $\mathfrak{S}_{\Gamma_0}$ for the set of S-structures for Γ_0, we arrive at the following rule:

$$(\mathcal{S}) \quad \frac{\Delta_M(\nu(M)) \text{ for each } M \in \mathfrak{S}_{\Gamma_0}}{\underbrace{\{\neg(A_i \Rightarrow B_i) \mid i \in I\}, (A_0 \Rightarrow B_0)}_{\equiv:\Gamma_0}},$$

indexed over a map ν such that for each $M = (S, \preceq) \in \mathfrak{S}_{\Gamma_0}$, $\nu(M) \in S/\simeq$. With this notation, we can state:

Theorem 18. *The system consisting of all instances of rule* $(\mathcal{S})$ *is sound, complete and admits one-step cut elimination.*

Proof. We briefly recall the preference semantics of the one-step logic of system $\mathcal{S}$ (which is essentially the original semantics of Kraus, Lehmann, and Magidor, except that we admit Boolean combinations of conditionals in the syntax). We emphasize it suffices to concentrate on the one-step logic, as the conditional version of the Kraus-Lehmann-Magidor axioms is already known to be complete for full system $\mathcal{S}$ [4, 9]. This is irrespective of the fact that the preference semantics of system $\mathcal{S}$ differs from the preferential semantics of [13] as recalled below in that it conflates worlds and states.

A *preferential model* $(S, l, \preceq)$ over a set X of worlds consists of a set S of states, equipped with a partial ordering $\preceq$ 'less remote than' and an assignment l of a world $l(s) \in X$ to each state $s \in S$. By the results of [9] we may assume that S is finite. Given a valuation τ of the propositional atoms as subsets of X, we have an evident definition of satisfaction of $A \in \mathsf{Prop}(P)$ in worlds $x \in X$, which we can pull back to S along l; we call a state an A-state if it satisfies A.

Then, $(S, l, \preceq)$ satisfies $A \Rightarrow B$ for $A, B \in \mathsf{Prop}(P)$ iff all minimal A-states in S satisfy B. It follows from the completeness result of [9] that a one-step formula is consistent in system $\mathcal{S}$ iff it is satisfiable in a preferential model. Dually, a one-step rule $\Gamma_1 \dots \Gamma_n / \Gamma_0$ is derivable iff it is valid in preferential models, where the latter means that whenever all worlds $x \in X$ satisfy the premises $\Gamma_1 \dots \Gamma_n$, then the preferential model satisfies the conclusion Γ_0.

To continue, we note that in the search for a rule set that absorbs cut, we can limit ourselves to rules whose conclusion $\Gamma_0 \equiv \neg(A_1 \Rightarrow B_1), \dots, \neg(A_n \Rightarrow B_n), (A_0 \Rightarrow B_0)$ is a *Horn sequent*, i.e. has exactly one positive literal, as all the axioms of system $\mathcal{S}$ are of this form and the set of Horn sequents is closed under cut. (The same is also easily seen semantically by taking disjoint unions of preference orderings.) Lemma 0.1 of [9] (which has a simple proof using selection of minimal worlds and totalization of partial orders) implies that such a rule is derivable iff it is valid in all finite *linear* preferential models, i.e. those where the states are linearly ordered. Evidently, the only relevant states in such a model w.r.t. to satisfaction of the rule conclusion Γ_0 are those worlds which are minimal A_i-worlds for some $i \in \hat{I} = \{0, \dots, n\}$. Moreover, it is easy to see that satisfaction of Γ_0 is unaffected by the removal of states properly above ('more remote than') 0. By another selection step, we can thus restrict attention to linear preferential models $M = (S, l, \preceq)$ where S is a subset of $\hat{I}$ with greatest element $0 \in S$, which is precisely what the rule stipulates. □

Note that although we arrived at the rule coming from the semantics, its eventual character is syntactic enough, its premises being essentially indexed by linear orderings of the literals of the conclusion. Since S-structures are of polynomial size, the complexity of strict one-step derivability can be kept at bay.

Lemma 19. *Strict one-step derivability under the rules $(\mathcal{S})$ is in* PSPACE.

Corollary 20. *The rules $(\mathcal{S})$ induce a sequent system for system $\mathcal{S}$ which admits cut elimination and proof search in* PSPACE.

Implementation issues As indicated in the introduction, we have implemented proof search in system $\mathcal{S}$ as an extension of the generic public domain reasoner CoLoSS [5, 12]. Our present implementation uses only very elementary optimizations, but is nevertheless able to prove (or disprove) all consequences and non-consequences of the axiom system discussed in [13]. Several optimizations to the implementation of rule $(\mathcal{S})$ are currently under investigation. In particular, note that we may abstract the premises $\Delta_M[i]$ to $\Delta(R,T) \equiv \bigcup_{k \in R}\{\neg A_k, C_k\}, \{A_j \mid i \in T\}$, and then need only find the minimal choices of R, T such that $\Delta_{R,T}$ is derivable; coverage of all S-structures then translates into a moderate-sized SAT-instance. The number of possible R, T to consider is related to the size of antichains in a partial order on pairs of sets; as such, it is still theoretically exponential but may be hoped to be feasible in practice.

6 Conclusion

We have established cut-free sequent calculi for several strong variants of conditional logic, including the standard system $\mathcal{S}$ that internalizes Kraus-Lehmann-Magidor preferential entailment. Dually, this amounts to the presentation of unlabelled tableau systems for these logics. We have shown that in each case, proof search can be performed in PSPACE. In particular, we have presented the first tableau system that matches the known complexity of system $\mathcal{S}$, which was previously established using a small model property [9]. Technically, our results come about by non-trivial application of generic methods for cut elimination [18] and complexity analysis [16] requiring sets of rules that absorb cut and contraction; we have demonstrated both syntactic and semantic approaches to finding such sets. Our implementation of the tableau algorithm for system S constitutes, to our knowledge, the first implemented reasoner for this important logic.

References

[1] A. Artosi, G. Governatori, and A. Rotolo. Labelled tableaux for non-monotonic reasoning: Cumulative consequence relations. *J. Log. Comput*, 12:1027–1060, 2002.

[2] C. Boutilier. Conditional logics of normality: A modal approach. *Artif. Intell*, 68:87–154, 1994.

[3] C. Boutilier and M. Goldszmidt. Revision by conditional beliefs. In *National Conference on Artificial Intelligence, AAAI 93*, pp. 649–654. AAAI Press/MIT Press, 1993.

[4] J. Burgess. Quick completeness proofs for some logics of conditionals. *Notre Dame J. Formal Logic*, 22:76–84, 1981.

[5] G. Calin, R. Myers, D. Pattinson, and L. Schröder. CoLoSS: The coalgebraic logic satisfiability solver (system description). In *Methods for Modalities, M4M-5*, vol. 231 of *ENTCS*, pp. 41–54. Elsevier, 2009.

[6] B. Chellas. *Modal Logic*. Cambridge University Press, 1980.

[7] G. Crocco and P. Lamarre. On the connection between non-monotonic inference systems and conditional logics. In *Principles of Knowledge Representation and Reasoning, KR 92*, pp. 565–571. Morgan Kaufmann, 1992.

[8] J. Dix and A. Hunter, eds. *Nonmonotonic Reasoning, NMR 06*, vol. IfI-06-04 of *IfI Technical Report Series*. TU Clausthal, 2006.

[9] N. Friedman and J. Y. Halpern. On the complexity of conditional logics. In *Principles of Knowledge Representation and Reasoning, KR 94*, pp. 202–213. Morgan Kaufmann, 1994.

[10] L. Giordano, V. Gliozzi, N. Olivetti, and G. L. Pozzato. Analytic tableaux calculi for KLM logics of nonmonotonic reasoning. *ACM Trans. Comput. Log*, 10(3), 2009.

[11] L. Giordano, V. Gliozzi, N. Olivetti, and C. Schwind. Tableau calculi for preference-based conditional logics. In *Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX 03*, vol. 2796 of *LNCS*, pp. 81–101. Springer, 2003.

[12] D. Hausmann and L. Schröder. Optimizing conditional logic reasoning within CoLoSS. In *Methods for Modalities, M4M-6*, ENTCS. Elsevier, 2009. To appear.

[13] S. Kraus, D. J. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artif. Intell.*, 44:167–207, 1990.

[14] N. A. Lynch. Log space recognition and translation of parenthesis languages. *J. ACM*, 24:583–590, 1977.

[15] D. McDermott and J. Doyle. Non-monotonic logic I. *Artif. Intell*, 13:41–72, 1980.

[16] R. Myers, D. Pattinson, and L. Schröder. Coalgebraic hybrid logic. In *Foundations of Software Science and Computation Structures, FOSSACS 2009*, vol. 5504 of *LNCS*, pp. 137–151. Springer, 2009.

[17] N. Olivetti, G. L. Pozzato, and C. Schwind. A sequent calculus and a theorem prover for standard conditional logics. *ACM Trans. Comput. Logic*, 8(4:22):1–51, 2007.

[18] D. Pattinson and L. Schröder. Cut elimination in coalgebraic logics. *Inform. Comput.* To appear.

[19] R. Reiter. A logic for default reasoning. *Artif. Intell*, 13:81–132, 1980.

[20] L. Schröder. A finite model construction for coalgebraic modal logic. *J. Log. Algebr. Prog.*, 73:97–110, 2007.

[21] I. Song and G. Governatori. Nested rules in defeasible logic. In *Rules and Rule Markup Languages for the Semantic Web, RuleML 2005*, vol. 3791 of *LNCS*, pp. 204–208. Springer, 2005.