Labelled Tableaux for Nonmonotonic Reasoning: Cumulative Consequence Relations

Alberto Artosi
Department of Philosophy,
University of Bologna,
via Zamboni 38, I-40126 Bologna, Italy
email: artosi@cirfid.unibo.it

Guido Governatori
School of Information Technology and Electrical Engineering
The University of Queensland
Brisbane, QLD 4072, Australia
email: guido@itee.uq.edu.au

Antonino Rotolo
CIRSFID, University of Bologna,
Via Galliera 3, I-40121 Bologna, Italy
email: rotolo@cirfid.unibo.it

Abstract

In this paper we present a labelled proof method for computing nonmonotonic consequence relations in a conditional logic setting. The method exploits the strong connection between these deductive relations and conditional logics, and it is based on the usual possible world semantics devised for the latter. The label formalism *KEM*, introduced to account for the semantics of normal modal logics, is easily adapted to the semantics of conditional logic by simply indexing labels with formulas. The basic inference rules are provided by the propositional system KE^{+} —a tableau-like analytic proof system devised to be used both as a refutation method and a direct method of proof— that is the classical core of *KEM* which is thus enlarged with suitable elimination rules for the conditional connective. The resulting algorithmic framework is able to compute cumulative consequence relations in so far as they can be expressed as conditional implications.

1 Introduction

In this paper we present a labelled proof method suitable for computing cumulative nonmonotonic consequence relations in a conditional logic setting. Generally speaking, our aim is to define a framework that, paying great attention to proof-theoretical formulations, will turn out to be a fruitful step towards further computational developments in treating wide

Journal of Logic and Computation, 12 (6): 1027-1060.

The original publication is available at doi: 10.1093/logcom/12.6.1027.

classes of nonmonotonic inferences.

It is commonly acknowledged that the idea to study nonmonotonic logics in terms of their consequence relations can be traced back to Gabbay [18]. Since nonmonotonic reasoning was brought up in the computer science field in the seventies, a great number of formalisms have been developed. In the light of this plethora of different systems, the merit of Gabbay was to focus on the minimal theoretical properties which should characterize all systems that exhibit nonmonotonic behavior. Following this idea, some proposals have been put forward to find a unifying approach to nonmonotonic reasoning. We refer, in particular, to Shoham's [37] general semantic framework for nonmonotonic logics, and Kraus, Lehman and Magidor's [31] and [29] approach to nonmonotonic consequence relations (see [28] for a later "unified" semantic account of nonmonotonic consequence relations and related areas such as, e.g., belief revision and conditional logic). Shoham first proposed a preference (ordering) semantics with partially ordered worlds. In this perspective, a proposition B is a nonmonotonic (preferential) consequence of a proposition A if both A and B are satisfied in the most normal worlds selected by the preference relation. Later on, Lehman and his colleagues extended Shoham'work by using too preferential models or similar machineries to characterize some particular nonmonotonic inference. In general, they studied five families of consequence relations and, for all of them, they provided both a semantic characterization and a proof-theoretical definition by using Gentzen-style inference rules. The two views were therefore integrated by means of "representation theorems" which associate semantics to nonmonotonic deductive relations. Among such families it is worth mentioning the cumulative, preferential and rational consequence relations. While preferential and rational systems have been widely investigated since they embody some general properties enjoyed by well-known nonmonotonic logics (see, in particular [31]), the study of cumulative relations seems to deserve more enquiry. Cumulative reasoning, in fact, is not only closely related to some interesting systems such as defeasible logics (see, [7]), but, as the weakest system, it encodes just the basic conditions identified by Gabbay.

In particular, a computational treatment of this family of consequence relations is still missing. The only attempt in this direction is Lehmann's algorithm [31] which, unfortunately, works only for preferential reasoning. A different and more general route has been shown by Fariñas del Cerro, Herzig and Lange [16]. They have pointed out that a computational account of nonmonotonic inference can be provided by simply reducing computation to a validity test in a (monotonic) conditional logic (henceforth CL). In fact, the idea is to exploit the strong connections between nonmonotonic consequence relations and (monotonic) modal and conditional logics. As is well-known, such a connection has been emphasized by several scholars, such as Boutilier [8] and Katsuno and Satoh [28], whose analysis rely on Kripke structures very close to Kraus, Lehmann and Magidor's "preferential" models. In particular, Boutilier [8] has shown on this basis that Kraus, Lehmann and Magidor's [29, 31] preferential and rational consequence relation systems and Degrande's [15] logic **N** closely correspond to the flat parts of modal CLs definitionally equivalent to the standard modal systems **S4** and **S4.3**.

In this paper we follow Fariñas del Cerro's advice in order to develop a general and "effective computation ... of nonmonotonic inference relations via automated deduction method" [16, p. 387] in CL. We know only two previous attempts in this direction: Groeneboer and Delgrande's [26] and Lamarre's [30] tableau-based theorem provers for some normal CLs. In both approaches a conditional formula is checked for validity by attempting to construct a model for its negation. What we undertake in this paper can be viewed as a further step in the same direction, as in our approach cumulative nonmonotonic consequence relations can be effectively computed by a counter-model validity test for the corresponding class of conditional formulas.

Until now, the inferential structure of CLs has not been sufficiently explored to provide reliable automated deduction methods for effectively computing the inferences sanctioned by cumulative reasoning. In fact, in contrast with the striking development of *CL*'s semantic setting, its inferential structure has remained largely unexplored (with the notable exceptions of [39, 14, 23, 35]). To accomplish the above goal we shall proceed by first looking for a suitable CL that can be used as an appropriate counterpart of the class of cumulative consequence relations. Such a logic, called **CU**, is a simple extension of Chellas' [10] basic normal system **CK**. Actually, our tableau proof system construction is just aimed to compute nonmonotonic consequence relations in this (monotonic) CL whose "flat" (i.e., unnested) fragment is shown to correspond to Kraus, Lehmann and Magidor's basic system **C** for cumulative relations. Let us point out that **CU** is in itself an additional outcome of this paper, which seems to offer potentially a flexible proof method even for other normal conditional logics.

As regards the technical tools and the formalism adopted in the paper, our construction is based on an algorithmic proof system which uses a labelling discipline, in the wake of Gabbay's [20] *Labelled Deductive Systems* (LDSs), to generate and check models. A detailed discussion of the merits of LDSs as a unifying framework is beyond the scope of this paper. However, a key feature of LDSs is worth mentioning. LDSs are in general very sensitive to the various features of different logics so that differently motivated and formulated logics can very often be combined in a simple and natural way provided we have a suitable LDS formulation for them (see, e.g., [21, 22, 4]). In LDSs the usual modal semantics is incorporated in the syntactic label construction and only minor variations are needed to pass from one logic to another [1, 4, 21, 22, 25, 5, 36, 40]. Thus, once an automated LDS is available for some appropriate modal systems, only slight natural changes in the modal LDS are needed to yield the appropriate semantics for CLs and nonmonotonic consequence relations.

More precisely, we use a labelled tableau system, called *KEM*, suitable for a variety of intensional (modal) logics that can be characterized in terms of possible world semantics (see [1, 22]). The general idea behind *KEM* is to represent modalities, and more generally intensional operators, as labels which are nothing but structured sequences of world symbols. Due to the basic format of such labels the semantics developed by Shoham [37] and adapted by Kraus, Lehmann and Magidor [29] to nonmonotonic reasoning can hardly be encoded in the *KEM* label formalism. To cover nonmonotonic consequence relations we thus need to treat this kind of reasoning in a possible world setting. This requirement is far from being unnatural because of the mentioned strong correspondence between nonmonotonic consequence relations and CLs, which enjoy a modal-like semantics. In this sense, our system can fruitfully exploit the intuition of Fariñas del Cerro and his colleagues about the methods for computing nonmonotonic inferences. In particular, we refer to Chellas's [10] approach according to which a conditional can be conceived of as a modality parameterized by its antecedent. To reflect this idea we propose to enrich *KEM*'s label formalism by attaching formulas to world symbols. Accordingly, that B is a nonmonotonic consequence of a proposition A is intuitively represented by an expression like $TB : i^A$, that means that B is true in the worlds, denoted by the label i, where A is true.

A crucial feature of our system is that it works both on the declarative and on the labelled part of such expressions. However, much of the job is done thanks to an appropriate algebra and to specific algorithmic procedures for manipulating labels. In the literature, it is usual to distinguish two ways to deal with labels: 1) labels are propagated using logic-dependent inference rules (see, for example, Fitting's prefixed tableaux [17] and single-step tableaux [33, 6]); 2) labels are matched according to logic-depended label unifications. Even if our system employs the second strategy, it is worth noting that, in both cases, it is necessary to take care of the formulas that parameterize labels when inference rules or unifications are defined, and the conditions on such formulas should mimic the semantics of the nonmonotonic system under analysis. Very often these requirements have a semantic nature, so that a simple inspection of the formulas involved is not enough. Intuitively, suppose we need to see whether the labels i^X and j^Y denote the same set of worlds. This is quite usual in labelled tableaux, at least when the closure of a proof tree is obtained by checking for a contradiction in the same worlds. To do so, for example, it could be necessary to check that the formulas X and Y attached to the labels are equivalent, since a given formula A is true at the first label and false at the second. If so, i^X and j^Y correspond to the same worlds and it is possible to make the contradiction explicit.

What can be done to verify these conditions is to devise auxiliary proof methods whenever they are needed. However, this solution is far from being optimal; indeed, it could be computationally expensive because, in some cases, we are forced to check several times some conditions, leading thus to redundancies and useless pieces of information. Fortunately, this is not necessary for some logics, and, in such cases, a more elegant solution is to provide syntactic criteria under which the conditions are met, thus using the information present in the main proofs. Accordingly, we define a specific proof search method which enables us to perform the check without generating redundant auxiliary trees. More precisely, this is done by using a suitable modification of the classical proof system *KE* proposed by D'Agostino and Mondadori [13]. Actually, the *KEM* system generalizes *KE* by using a labelled language since its propositional core employs a similar mixture of tableau, natural deduction and structural rules. However, such a new version of *KE*, called KE^+, makes the system completely analytical. Thus, it is possible to keep track of strict dependencies between the different subformulas generated by other formulas, so that it is easy to exploit this information to check if such formulas are, for example, tautologies or classically equivalent. Unless we want to accept a redundant system, this procedure is highly desirable in our system because of the structure of the adopted label formalism. However, this is a result to be emphasized besides the specific requirements of our proof system, since it is based on an intuition that can be applied to other labelled systems which employ a similar package of inference rules.

To sum up, we shall proceed in the following way. First we briefly review Kraus, Lehmann and Magidor's [29] sequent system **C** for cumulative relations. Then we introduce Lewis-type semantic structures akin to the kind of models used to characterize **C**. Such structures will allow us to establish a correspondence between **C** and the flat fragment of the above mentioned conditional logic **CU**. At this point, we shall be able to show how cumulative relations can be effectively computed by an LDS provided by a tableau-like proof system together with a label formalism adequate to represent the intended semantics. The system is presented in two steps. First, the labelling (formalism + label unification) scheme introduced in [1] to account for the semantics of normal modal logics is adapted to represent Lewis-type semantic structures for **CU**. Then a suitable tableau inference and label propagation rules are introduced which provide a sound and complete proof system for the flat fragment of **CU**. The system is then improved by developing a proof search method based on KE^+. Finally, we provide some remarks on further extensions and related works.

2 Nonmonotonic Consequence Relations and Conditional Logic

The study of nonmonotonic consequence relations has been undertaken by Gabbay [18] who proposed three minimal conditions a (binary) consequence relation $\mid\!\sim$ on a language L should satisfy to represent a nonmonotonic logic, i.e.,

$$\Delta, A \mid\!\sim A \qquad \text{(Reflexivity)}$$

$$\frac{\Delta, A \mid\!\sim B; \Delta \mid\!\sim A}{\Delta \mid\!\sim B} \qquad \text{(Cut)}$$

$$\frac{\Delta \mid\!\sim A; \Delta \mid\!\sim B}{\Delta, A \mid\!\sim B} \qquad \text{(Cautious Monotonicity)}$$

More recently, Kraus, Lehmann and Magidor [29] have investigated the proof-theoretic and semantic properties of a number of increasingly stronger families of nonmonotonic consequence relations. In particular, they have provided the following sequent system **C** to define the (weakest) class of cumulative consequence relations, that closely corresponds to that satisfying Gabbay's minimal conditions (we assume that both the usual monotonic consequence relation $\vdash$ and its nonmonotonic counterpart $\mid\!\sim$ are defined on the language L of classical propositional logic).

$$A \mid\!\sim A \qquad \text{(Reflexivity)}$$

$$\frac{\vdash B \rightarrow C \quad A \mid\!\sim B}{A \mid\!\sim C} \qquad \text{(Right Weakening)}$$

$$\frac{\vdash A \equiv B \quad A \mid\!\sim C}{B \mid\!\sim C} \qquad \text{(Left Logical Equivalence)}$$

$$\frac{A \mid\!\sim B \quad A \mid\!\sim C}{A \wedge B \mid\!\sim C} \qquad \text{(Cautious Monotonicity)}$$

$$\frac{A \wedge B \mid\!\sim C \quad A \mid\!\sim B}{A \mid\!\sim C} \qquad \text{(Cut)}$$

Notice that the following

$$\frac{A \mid\!\sim B \quad A \mid\!\sim C}{A \mid\!\sim B \wedge C} \qquad \text{(And)}$$

$$\frac{A \mid\!\sim B \quad B \mid\!\sim A \quad A \mid\!\sim C}{B \mid\!\sim C} \qquad \text{(CSO)}$$

are derived rules of **C**. A sequent $A \mid\!\sim B$, $A, B \in L$ (intended reading: B is a plausible consequence of A), is called a *conditional assertion*. The (proof-theoretic) notion of cumulative entailment is defined for such assertions. Let Γ be a set of conditional assertions. A conditional assertion $A \mid\!\sim B$ is said to be *cumulatively entailed* by Γ iff $A \mid\!\sim B$ is derived from Γ using the rules of **C**.

Let $L_>$ be the language obtained by adding the conditional connective $>$ to L. The set of (well-formed) formulas of $L_>$ is defined in the usual way. Formulas of $L_>$ are interpreted in terms of Lewis-type semantic structures akin to the kind of models used by Kraus, Lehmann and Magidor [29] to characterize **C**.

More precisely, it is enough to introduce some constraints (see definition 2) on the basic selection function model presented in definition 1.

Definition 1 A *selection function* (*SF*) *model* is a triple $M = \langle W, f, \upsilon \rangle$ where

1. W is a nonempty set (of possible worlds);
2. f is a selection function which picks out a subset $f(A,u)$ of W for each u in W and $A \in L_>$;
3. υ is a valuation assigning to each u in W and $A \in L_>$ an element from the set $\{T,F\}$.

We refer to the set of worlds $f(A,u)$ as the set of A-worlds with respect to u.

Truth of a formula A at a world u in a model M, $M \models_u A$, is defined as usual with the conditional case given by

$$M \models_u A > B \text{ iff } f(A,u) \subseteq \|B\| \tag{1}$$

where $\|B\|$ denotes the set of worlds where B is true, i.e., $\|B\| = \{w \in W : \upsilon(B,w) = T\}$. A formula A is valid ($\models_{SF}$) just when $M \models_u A$ for all worlds u in all *SF* models.

Definition 2 A *selection function cumulative model* (*SFC*) is an *SF* model $M = \langle W, f, \upsilon \rangle$ satisfying the following conditions:

1. $f(A,u) \subseteq \|A\|$ (Reflexivity)
2. If $\|A\| = \|B\|$, then $f(A,u) = f(B,u)$ (Left Logical Equivalence)
3. If $f(A,u) \subseteq \|B\|$, then $f(A \wedge B,u) \subseteq f(A,u)$ (Cautious Monotonicity)
4. If $f(A,u) \subseteq \|B\|$, then $f(A,u) \subseteq f(A \wedge B,u)$ (Cut)

Notice that from 3 and 4 above we obtain

$$f(A,u) \subseteq \|B\| \Rightarrow f(A \wedge B,u) = f(A,u) \tag{2}$$

It is not hard to see that the class of *SFC* models fits exactly the conditional logic —call it **CU**— containing classical propositional logic and the following axioms

1. $A > A$ (*ID*)
2. $(A > B) \wedge (A \wedge B > C) \rightarrow (A > C)$ (*RT*)
3. $(A > B) \wedge (A > C) \rightarrow (A \wedge B > C)$ (*CM*)

and closed under the usual inference rules

$$\frac{A \equiv B}{(A > C) \rightarrow (B > C)} \quad (RCEA)$$

and

$$\frac{(A_1 \wedge \cdots \wedge A_n) \rightarrow B}{(C > A_1 \wedge \cdots \wedge C > A_n) \rightarrow C > B} \qquad (RCK)$$

Notice that *ID*, *RT*, *CM*, *RCEA*, and *RCK* correspond, respectively, to Reflexivity, Cut, Cautious Monotonicity, Left Logical Equivalence and Right Weakening. Of course, **CU** is nothing but Chellas' [10] conditional logic $\mathbf{CK} + ID$ augmented with *RT* and *CM* (Burgess' [9] axiom $A3$). A standard Henkin-style construction proves the completeness of **CU** with respect to the class of *SFC* models.

Lemma 3 Let u be a set of formulas closed under **CU**. Let $v = \{B : A > B \in u\}$. If u, v are consistent and $\neg(A > C) \in u$ then $\{\neg C\} \cup v$ is consistent.

Proof Suppose $\{\neg C\} \cup v$ is not consistent. Then $\{B_1, \ldots, B_n\}$ is a finite subset of v such that $\bigwedge_{i=1}^{n} B_i \wedge \neg C \vdash \bot$ which implies $\bigwedge_{i=1}^{n} B_i \vdash C$. In u we have $A > B_i$, therefore by *RCK* we can conclude $A > C \in u$ contradicting the consistency of u.

Theorem 4 $\models_{SFC} A$ iff $\vdash_{\mathbf{CU}} A$.

Proof One half of the theorem is easy to prove, for it is a tedious but entirely routine exercise to check that each axiom is valid in all the *SFC* models. For the other half let us consider the canonical *SFC* model

$$\langle W, f, v \rangle$$

where

- W is the set of **CU**-maximal consistent sets (a usual Henkin construction shows that W is not empty);
- $f(A, u) = \{w : \{B : A > B \in u\} \subseteq w\}$;
- for any propositional letter p, $v(p, u) = T$ iff $p \in u$.

We first show that the last clause can be extended to generic formulas of L, i.e., $v(A, u) = T$ iff $A \in u$.

We use induction on the complexity of a formula A. If A is an atomic proposition then the property holds trivially.

Let us assume that $v(A > B, u) = T$. According to (1), $f(A, u) \subseteq ||B||$; moreover, by maximality, either $A > B$ or $\neg(A > B)$ is in u. If the latter is the case, by Lemma 3, for any $w \in f(A, u)$, $w \cup \{\neg B\}$ is consistent; but $B \in w$, thus $w \cup \{\neg B\}$ is not consistent, and so we have a contradiction; hence $A > B \in u$.

If $A > B \in u$ then, by construction, $f(A, u) \subseteq ||B||$, which implies $v(A > B) = T$.

We have now to prove that the model is cumulative. We only show the case for Cautious Monotonicity and *CM*. The other cases can be found in [34]. We assume the antecedent of clause 2 of Definition 2, that is $f(A, u) \subseteq ||B||$, and we have to show that $f(A \wedge B, u) \subseteq f(A, u)$.

By maximality either $A > B \in u$ or $\neg(A > B) \in u$. If the latter is the case then, by Lemma 3, $\forall w \in f(A, u)$, $w \cup \{\neg B\}$ is consistent. However, according to our assumption, each w contains B, so $w \cup \{\neg B\}$ is inconsistent, therefore $\neg(A > B)$ cannot be in u, hence $A > B \in u$. Each instance of axiom *CM* is in u, and, by maximality, u is closed under classical propositional logic; thus, for any formula $A > C$, if $A > C \in u$, then $A \wedge B > C \in u$; this implies that for every formula A

$$\{C : A > C \in u\} \subseteq \{D : A \wedge B > D \in u\}.$$

Finally, by the definiton of the selection function f for the canonical SFC model

$$f(A \wedge B, u) \subseteq f(A, u).$$

Whether **CU** is interesting in its own right is an issue which need not detain us here. What matters is that we can establish a mapping between **C** and **CU** similar to the well-established correspondences between [29]'s stronger systems **P** and **R** of preferential and rational relations and the flat fragments of well-known conditional logics.

In order to prove the equivalence between a nonmonotonic consequence relation and a conditional logic, we have to define the kind of rules describing a nonmonotonic consequence relation. First of all we notice that a property of a nonmonotonic consequence relation has the form of a rule

$$\frac{A_1, \ldots, A_n}{B} \tag{3}$$

where each A_i and B are either of the form $C \mid\!\sim D$ or $\vdash E$, where C, D, and E are classical formulas.

At this point, we need some machinery for translating any structure of a nonmonotonic consequence relation system into the language $L_>$. Such a machinery has already been provided by Crocco and Lamarre [12]. According to their approach, it is enough to classify properties of a nonmonotonic consequence relation as follows:

Definition 5 (cf. Definition 2.4 in [12]) A rule is:

- of type 1 if and only if the symbol of monotonic deduction ($\vdash$) does not appear in it;
- of type 2 if and only if the symbol of monotonic deduction appears only in its premises.

The above classification gives us a base for introducing a transformation "$*$" replacing $\mid\!\sim$ with the conditional connective $>$.

Definition 6 (cf. Definition 2.6 in [12]) Let F be an expression of L; F^* is the expression of $L_>$ corresponding to F, such that:

- for every classical expression F, $F^* = F$;
- for every expression $F = A \mid\!\sim B$, $F^* = A > B$;
- for every expression $F = A \not\mid\!\sim B$, $F^* = \neg(A > B)$;
- for every rule $F = \dfrac{A_1, \ldots, A_n}{C}$ of type 1, $F^* = (A_1^* \wedge \cdots \wedge A_n^*) \rightarrow C^*$;
- for every rule $F = \dfrac{\vdash B_1, \ldots, \vdash B_n, A_1, \ldots, A_m}{C}$ of type 2, $F^* = \dfrac{B_1, \ldots, B_2}{(A_1^* \wedge \cdots \wedge A_m^*) \rightarrow C^*}$

According to the above transformation, rules of type 1 correspond to axioms of CLs, whereas rules of type 2 correspond to inference rules of CLs.

We can now state formally the relationship between a nonmonotonic consequence relation and a conditional logic. Let **S** be a nonmonotonic consequence relation, and let **K** be the conditional logic whose axioms and inference rules are the translation of the properties characterizing **S**.

Theorem 7 [12] Let M be a sound and complete semantics for **K**. Then F is a general property of **S** iff F^* is valid in M.

Proof For the proof see [12].

Let $\mid\sim_{\mathbf{S}}$ denote the consequence relation **S** and let $\mathbf{K}^-$ denote the conditional logic **K** restricted to the formulas of the form $A > B$ where $A, B \in L$.

Definition 8 A consequence relation $\mid\sim$ is *defined by* an *SF* model M if the following condition is satisfied: $A \mid\sim B$ iff $M \models A > B$.

Definition 9 A consequence relation system **S** is said to *correspond to* a conditional logic **K** if the following condition is satisfied: $A \mid\sim_{\mathbf{S}} B$ iff $\vdash_{\mathbf{K}^-} A > B$.

Theorem 10 The consequence relation system **C** corresponds to the conditional logic **CU**.

From Theorem 4 we know that **CU** is characterized by *SFC*-models. The theorem follows from showing that the axioms and rules of **CU** are the translations, according to Definition 6, of the rules of **C** and thus $A \mid\sim_{\mathbf{C}} B$ is the consequence relation defined by an *SFC* model.

From this it follows as a corollary that a consequence relation $\mid\sim$ is cumulative iff it is defined by some *SFC* model. The same holds for the notion of cumulative entailment. For a set Γ of conditional assertions let us denote by Γ' the set containing the $\mathbf{CU}^-$ translations of the conditional assertions in Γ (i.e., $A > B \in \Gamma'$ for each $A \mid\sim B \in \Gamma$). The following corollaries are derived immediately from Theorem 10 (see Corollaries 3.26, 3.27 and 3.28 of [29] for comparison).

Corollary 11 Let Γ be a finite set of conditional assertions and $A \mid\sim B$ a conditional assertion. The following conditions are equivalent. In case they hold we shall say that Γ cumulatively entails $A \mid\sim B$.

1. $A > B$ is derived from Γ' using the axioms and the rules of **CU**.

2. $A > B$ is satisfied by all *SCF* models which satisfy Γ'.

Corollary 12 A finite set of conditional assertions Γ cumulatively entails $A \mid\sim B$ iff $\vdash_{\mathbf{CU}}$ $\bigwedge \Gamma' \rightarrow (A > B)$.

We conclude that the system **C** may be viewed itself as a restricted CL of the standard (normal) type provided the relation symbol $\mid\sim$ is interpreted as a $>$-type conditional connective. With this background we shall be able, in the upcoming sections, to provide an algorithmic framework for computing cumulative consequence relations in so far as they can be expressed as conditional implications.

3 *KEM* for Nonmonotonic Consequence Relations

In [1] we presented a proof system for normal modal logics, called *KEM*, which seems to enjoy most of the features a suitable proof search system for modal (and in general non-classical) logics should have. *KEM* is an algorithmic modal proof system which, in the spirit of Gabbay's [20] LDS, brings semantics into proof theory using (syntactic) labels in order to simulate models in the proof language. Very briefly, it is based on a combination of tableau and natural deduction inference rules which allows for a suitably restricted ("analytic") application of the cut rule; the label scheme arises from an alphabet of constant and variable "world" symbols.[1] A "world" label is either a world-symbol or a "structured" sequence of world-symbols we call a "world-path". Constant and variable world-symbols denote worlds and sets of worlds respectively (in a Kripke model), while a world-path conveys information about access between the worlds in it. As we have argued elsewhere, this proof system appears to be flexible enough to be extended to cover the full range of non-classical logics which are extensions of (or logically similar to) modal logic — indeed flexible enough to be adapted to any setting having a Kripke-model based semantics (see, e.g., [21, 22]). This is largely due to the particular label formalism it uses to generate and check models.

In this section we show how it can be extended, with little modification, to handle **C**. In what follows $\mathscr{L}$ will denote the sublanguage of $L_>$ including L and all the boolean combinations of formulas of the form $A > B$ where $A, B \in L$.

3.1 Label Formalism

As we have already alluded to, *KEM* has two basic kinds of atomic labels: variables and constants. Formally, let $\Phi_C = \{w_1, w_2, \dots\}$ and $\Phi_V = \{W_1, W_2, \dots\}$ be two arbitrary sets of *atomic labels*, respectively constants and variables. A *label* in the sense of [1] is an element of the set of labels $\Im$ defined as follows:

Definition 13 $\Im = \bigcup_{1 \leq p} \Im_p$ where $\Im_p$ is:

$$\Im_1 = \Phi_C \cup \Phi_V$$

$$\Im_2 = \Im_1 \times \Phi_C$$

$$\Im_{n+1} = \Im_1 \times \Im_n,\ n > 1\ .$$

Thus, a label is any $i \in \Im$ such that either i is an atomic label or $i = (k', k)$ where (i) k' is atomic and (ii) $k \in \Phi_C$ or $k = (m', m)$ where (m', m) is a label, i.e., i is generated as a "structured" sequence of atomic labels. As we said, in the standard Kripke setting we may think of constant and variable world-symbols as denoting respectively worlds and sets of worlds. A label of the form (k', k) is nothing else than a "world-path". For instance, the label (W_1, w_1) represents a path from w_1 to the set W_1 of worlds accessible from w_1; $(w_2, (W_1, w_1))$ represents a path which takes us to a world w_2 accessible by any world accessible from w_1 (i.e., accessible by the sub-path (W_1, w_1)) according to the appropriate accessibility relation. Thus a label of the form (k', k) is "structurally" designed to store information when we move from a world (or a set of worlds) to another. We define the length of a label i, $\ell(i)$, to be the number of atomic labels in i. From now on we shall use $i, j, k, \dots$ to denote arbitrary labels.

[1] See section 3.3, definition 23 for a formal definition of a *KEM* proof.

For a label $i = (j,k)$, we shall call j the *head* and k the *body* of i, and denote them by $h(i)$ and $b(i)$ respectively; $h^n(i)$ will denote the head of the sub-label of i whose length is n. We shall call a label i *restricted* if its head is a (possibly indexed) constant, otherwise we shall call it *unrestricted*.

In passing from Kripke models for modal logics to *SF* models the format of the labels is left unchanged. The only modification is that atomic labels are now indexed by formulas. Accordingly, let us stipulate that if $i \in \Phi_C \cup \Phi_V$ and $Y \in \mathscr{L}$ then $i^Y \in \Im_1$. We shall call a label i^Y a *formula-indexed label*, and Y the *label formula* of i.[2] The notion of a formula-indexed label is then meant to capture the intended semantics. For example, (W_1^A, w_1) can be viewed as representing (any world in) the set of the A-worlds with respect to w_1 under some selection function f. The label (w_1^A, w_1) represents an A-world in such a set. The interpretation of labels of the form (k', k) varies accordingly. Thus, the label $(w_2^A, (W_1^{A\vee C}, w_1))$ represents an A-world with respect to any $A \vee C$-world (with respect to w_1). Such a formalism is motivated by the general idea (see [10]) that $>$ can be regarded as a necessity operator on the antecedent of the conditional (i.e., $A > B$ is read as $[A >]B$). Thus, it follows that whenever $A > B$ is true at a world u, B should be true at all the worlds in $f(A,u)$ (A-worlds with respect to u); and whenever $A > B$ is false at u, there should be some A-world where B is false.

Definition 14 A *labelled signed formula* (*LS-formula*) [1] is a pair $X : i$ where X is a signed formula (i.e., a formula of $\mathscr{L}$ prefixed with a "T" or "F") and i is a label.

In the original *KEM* approach we attached labels to *signed* formulas (i.e., formulas of the modal language prefixed with a "T" or "F") to yield *labelled signed formulas* (*LS-formulas*), that is, pairs of the form $X : i$, where X is a signed formula and i is a label. Intuitively, an *LS*-formula, $TA : i$ is intended to mean: A is true at the world(s) denoted by the label i; for instance, $TA \rightarrow B : (W_1, w_1)$ means that $A \rightarrow B$ is true at all the worlds (any world) accessible from w_1. Similarly $FA \rightarrow B : (W_1, w_1)$ means that $A \rightarrow B$ is false at the worlds denoted by (W_1, w_1)

According to definition 14, this can be extended immediately to *SF* semantics. For instance an *LS*-formula, $TC : (W_1^{A\vee B}, w_1)$, means that C is true at all the $A \vee B$-worlds with respect to w_1.

As we have seen, formulas can occur in *LS*-formulas either as the declarative part or as label formulas; moreover formulas in both parts can and must be used to draw inferences. To deal with this fact we define when *SA occurs* in $X : i^Y$ ($SA@X : i^Y$). More precisely:

$$SA@X : i^Y \iff \begin{cases} X = SA \text{ or} \\ Y = A \text{ and } S = T \end{cases}$$

where $S \in \{T, F\}$, $A, Y \in \mathscr{L}$, X is a signed formula, and $i \in \Im$. That means that either SA is labelled with i^Y, or i ($h(i)$) is indexed with A. For example, in the expression $SA@X : i^Y$, where $X = FB \rightarrow C$ and $i^Y = (W_1^{B\wedge C}, w_1)$, SA stands both for $FB \rightarrow C$, and $TB \wedge C$, since these are the formulas occurring in $X : i^Y$.

In what follows we assume familiarity with Smullyan's [38] uniform notation for signed formulas and the usual conversions between formulas and signed formulas.

[2] Notice that, for a label i, we shall use i^Y to indicate that the label formula of $h(i)$ is Y. In general, when we speak of the label formula of a structured label, we mean the label formula of the head of the label.

3.2 Label Unifications

The key feature of the *KEM* approach is that in the course of a proof search labels are manipulated in a way closely related to the semantics of modal operators and "matched" using a specialized (logic-dependent) unification algorithm. That two labels i and k are unifiable means intuitively that any world which one could get to by the path i could be reached by the path k and vice versa (equivalently, that the sets of worlds they "denote" have a non-null intersection). For example, $(w_3,(W_1,w_1))$ and $(W_3,(w_2,w_1))$ are unifiable (by simultaneously linking W_3 to w_3 and W_1 to w_2); thus they virtually represent the same path (since w_3 is a world in W_3 and w_2 is a world in W_1). *LS*-formulas whose labels are unifiable turn out to hold at the same world(s) relative to the accessibility relation characterizing the appropriate class of models. In particular two complementary *LS*-formulas such as $TA : i$ and $FA : k$ whose labels are unifiable stand for formulas which are contradictory "in the same world".

In this section we define a special notion of unification for **C** ($\sigma_{\mathbf{C}}$-unification) which is meant to "simulate" the conditions on the selection function f in *SFC*-models. We shall proceed by first defining the unification for unindexed labels, and then by extending it to formula-indexed labels.

First of all we introduce a label substitution $\rho : \Im \mapsto \Im$ thus defined:

$$\rho(i) = \begin{cases} i & i \in \Phi_C \\ j \in \Im & i \in \Phi_V \\ (\rho(h(i)),\rho(b(i))) & i \in \Im_n, n > 1 \end{cases}$$

For two labels i and j, and a substitution ρ, if ρ is a unifier of i and j then we shall say that i, j are *σ-unifiable*. We shall use $(i, j)\sigma$ to denote both that i and j are σ-unifiable and the result of their unification. In particular

Definition 15 For all $i, j, k \in \Im$

$$(i,j)\sigma = k \text{ iff } \exists\rho : \rho(i) = \rho(j) \text{ and } \rho(i) = k, \text{ and}$$
$$\text{for each } n \text{ at least one of } h^n(i) \text{ or } h^n(j) \text{ is in } \Phi_C.$$

According to the above condition, the labels $(w_3,(W_1,w_1))$ and $(W_2,(w_2,w_1))$ σ-unify on $(w_3,(w_2,w_1))$. On the other hand the labels $(w_2,(W_1,w_1))$ and $(W_2,(W_1,w_1))$ do not σ-unify because both h^2s are not in Φ_C.

The same holds for *SFC* models. For example, that two labels, e.g., $(W_1^{A\vee B},w_1)$ and $(w_3^{A\vee B},w_1)$, are unifiable will mean that $w_3^{A\vee B}$ is an $A \vee B$-world in the set of $A \vee B$-worlds denoted by $(W_1^{A\vee B},w_1)$. Accordingly, the pair of *LS*-formulas $TC : (W_1^{A\vee B},w_1)$ and $FC : (w_3^{A\vee B},w_1)$ expresses a contradiction in the same world. However, this is just a trivial case because of the identity of the label formulas. Generally speaking, the definition of the unification for indexed labels is more complicated since we have to take into account any label formulas.

As said before, the conditions on label formulas should mimic the semantics of *SFC*-models, but we have to devise them in a syntactic way. In particular, to check that two sets of worlds denoted by different indexed labels overlap, we have to determine a specific mechanism for comparing distinct label formulas. From a proof-theoretical point of view, such a comparison is concerned with the definition of a criterion for composing different structures of formulas. However, it is well-known that cumulative logics do not allow unrestricted compositions of proofs (see, e.g., [11]). In other words, they avoid substituting an antecedent for another antecedent by transitivity (via cut).

The aim of the following definitions is to establish the basic (restricted) conditions for the substitution of two formulas equivalent w.r.t. $\mid\!\sim$ ($\mid\!\sim$*-equivalent*) and a given set of formulas. The conditions for such an equivalence are given in Definition 20. In general equivalence is a bidirectional relation; thus, to accomplish this goal we have to provide the conditions for the two directions (Definition 19). The basic relations are given directly by the formulas in the just mentioned set and are basically detected (in particular, via Right Weakening) as entailment relations between the subformulas or the conjunctions of the relevant literals of the formulas in such a set (Definitions 18 and 16).

Definition 16

- For every formula A, $\{A\}$ *c-fulfils* A.
- If $\{A_1,\dots,A_n\}$ contains a pair of complementary literals then, for every formula A, $\{A_1,\dots,A_n\}$ *c-fulfils* A.
- If A is of type α, then $\{\alpha_1,\alpha_2\}$ *c-fulfils* A.
- If A is of type β, then $\{\beta_1\}$ *c-fulfils* A, and $\{\beta_2\}$ *c-fulfils* A.
- If $\{A_1,\dots,A_n\}$ *c-fulfils* A, and $\{A_{1_1},\dots,A_{1_m}\},\dots,\{A_{n_1},\dots,A_{n_m}\}$ *c-fulfil* respectively $A_1,\dots,A_n$, then $\{A_{1_1},\dots,A_{1_m},\dots,A_{n_1},\dots,A_{n_m}\}$ *c-fulfils* A.

It is easy to see that whenever a set of formulas c-fulfils a formula A the conjunction of the formulas in the set propositionally entails A.

Proposition 17 Let $\mathscr{A}$ be a set of classical formulas. If $\mathscr{A}$ c-fulfils A, then $\left\|\bigwedge_{A_i\in\mathscr{A}} A_i\right\| \subseteq \|A\|$.

Proof We prove the property by induction on the number n of boolean operators occurring in a formula A.

Case $n=1$: If A is of type α, then the only set that c-fulfils it is $\{\alpha_1,\alpha_2\}$; hence

$$\left\|\bigwedge_{A_i\in\mathscr{A}} A_i\right\| = \|\alpha_1\wedge\alpha_2\| = \|A\|$$

If A is of type β, then both $\{\beta_1\}$ and $\{\beta_2\}$ c-fulfil A. Moreover,

$$\|A\| = \|\beta_1\vee\beta_2\| = \|\beta_1\|\cup\|\beta_2\|$$

hence $\left\|\bigwedge_{A_i\in\mathscr{A}} A_i\right\| \subseteq \|A\|$. We have thus proved the inductive base.

It is worth noting that the formulas occurring in $\mathscr{A}$ have less operators than A.

Case $n>1$: Let $\mathscr{B}=\{A_1,\dots,A_n\}$ be a set of formulas that c-fulfils A, and assume that the property holds for it. Thus

$$\left\|\bigwedge_{1\le i\le n} A_i\right\| \subseteq \|A\|$$

Let $\mathscr{C}_i = \{B_{i_1}, \ldots, B_{i_m}\}$, $1 \leq i \leq n$ be a set of formulas such that $\mathscr{C}_i$ c-fulfils A_i. Then, according to Definition 16, $\bigcup_{1 \leq i \leq n} \mathscr{C}_i$ c-fulfils A. By the inductive hypothesis, the proposition holds for each pair $\mathscr{C}_i$ and A_i, since each A_i has less operators than A. Hence

$$\left\| \bigwedge_{1 \leq j \leq m} B_{i_j} \right\| \subseteq \|A_i\|$$

However,

$$\bigcap_{1 \leq i \leq n} \left\| \bigwedge_{1 \leq j \leq m} B_{i_j} \right\| \subseteq \bigcap_{1 \leq i \leq n} \|A_i\|$$

But

$$\bigcap_{1 \leq i \leq n} \left\| \bigwedge_{1 \leq j \leq m} B_{i_j} \right\| = \left\| \bigwedge_{\substack{1 \leq j \leq m \\ 1 \leq i \leq n}} B_{i_j} \right\| \qquad\qquad \bigcap_{1 \leq i \leq n} \|A_i\| = \left\| \bigwedge_{1 \leq i \leq n} A_i \right\|$$

therefore

$$\left\| \bigwedge_{\substack{1 \leq j \leq m \\ 1 \leq i \leq n}} B_{i_j} \right\| \subseteq \|A\|$$

Let $\mathscr{B}$ be any set of *LS*-formulas. (In the course of a proof search, $\mathscr{B}$ will be the set of *LS*-formulas occurring in a branch of a proof tree.)

Definition 18 We say that A *forces* B in (a branch) $\mathscr{B}$, iff A and B are in $\mathscr{B}$ and either 1) $A = B$ or A is of type α and $B = \alpha_n$, $n \in \{1,2\}$; or 2) there exists a formula C in $\mathscr{B}$ such that A forces C in $\mathscr{B}$ and C forces B in $\mathscr{B}$.

Given a formula A, the notion of "forcing" is meant to determine the subformulas of A that are propositionally entailed by A itself.

Definition 19 A *supports* B in (a branch) $\mathscr{B}$ iff

1. $\{B_1, \ldots, B_n\}$ c-fulfils B, and $B_k : (W^A_{i_k}, w_1) \in \mathscr{B}$ for each k, $1 \leq k \leq n$; or

2. there is a set of formulas $\mathscr{A} = \{Z_1, \ldots, Z_n\}$ such that, for all i, $1 \leq i \leq n$, $Z_i : (W^A_i, w_1) \in \mathscr{B}$, A forces Z_i in $\mathscr{B}$, and $\mathscr{A}$ c-fulfils B.

We are now ready to say when two formulas, A and B, are $\mid\!\sim$-equivalent in $\mathscr{B}$ ($A \approx_{\mathscr{B}} B$).

Definition 20 $A \approx_{\mathscr{B}} B$ iff A and B are in $\mathscr{B}$ and either

1. $A \equiv B$; or

2. A and B support each other; or

3. there is a formula C in $\mathscr{B}$ such that $A \approx_{\mathscr{B}} C$ and $B \approx_{\mathscr{B}} C$.

If $A \in \mathscr{B}$, with $A_{\approx_{\mathscr{B}}}$ we shall denote the set of formulas $\{B_1, \dots, B_n\}$ such that, for all i, $1 \leq i \leq n$, $B_i \in \mathscr{B}$ and $B_i \approx_{\mathscr{B}} A$. It is obvious that $A_{\approx_{\mathscr{B}}}$ is an equivalence class, thus we abuse the notation and we use $A_{\approx_{\mathscr{B}}}$ to denote a formula in such a class.

Two formulas A and B are obviously equivalent with respect to $\mid\!\sim$ if they are classically equivalent. Otherwise, through the notion of *support* (see definition 19), we have basically the following cases: (i) the set of truth-value assignments which correspond to the consequences of A satisfies B; (ii) the set of consequence relations of A propositionally entails B. So, according to definition 20, A and B are equivalent with respect to $\mid\!\sim$ in $\mathscr{B}$ if (a) the above sets are equal, or (b) such sets are equal to another set. This means that they prove each other. To further clarify this notion let us examine the following example.

Example 21 Let $\mathscr{B}$ be the following set of *LS*-formulas:

$$\{TA : (W_1^A, w_1),\ TB : (W_1^A, w_1),\ TC : (W_1^A, w_1),\ TA : (W_2^C, w_1),\ TA : (W_3^{A \wedge B}, w_1)\}$$

We want to show that $A \wedge B \approx_{\mathscr{B}} C$. First of all both $A \wedge B$ and C occur in $\mathscr{B}$. It is immediate to see that the two formulas do not support each other, therefore, if they are not equivalent, and we assume that this is not the case, then we have to find a formula in $\mathscr{B}$ which is $\mid\!\sim$-equivalent to both. We have two candidates: A and B. Clearly B does not support any formula in $\mathscr{B}$, thus we consider A. It is obvious that $A \approx_{\mathscr{B}} C$: they support each other in $\mathscr{B}$ given the *LS*-formulas $TA : (W_2^C, w_1)$ and $TC : (W_1^A, w_1)$ and condition 2 of Definition 19. At this point we have to see whether $A \approx_{\mathscr{B}} A \wedge B$. That $A \wedge B$ supports A can be verified from the *LS*-formula $TA : (W_3^{A \wedge B}, w_1)$; on the other hand $\mathscr{B}$ contains the formulas $TA : (W_1^A, w_1)$, and $TB : (W_1^A, w_1)$, where $\{A, B\}$ c-fulfils $A \wedge B$, therefore $A \approx_{\mathscr{B}} A \wedge B$. Hence, by condition 2 of Definition 20 $C \approx_{\mathscr{B}} A \wedge B$.

At we are now ready to introduce the notion of unification for indexed labels to be used in the calculus. Briefly, two labels unify with respect to a set of *LS*-formulas if the unindexed labels unify and the label formulas satisfy conditions corresponding to clauses 1–4 of the semantic evaluation. In the next definition we provide such conditions.

Definition 22 Let i^X and j^Y be two indexed labels, and let $\mathscr{B}$ be a set of *LS*-formulas. Then

$$(i^X, j^Y)\sigma_{\mathbf{C}}^{\mathscr{B}} = (i, j)\sigma$$

where 1) $X \not\equiv \bot$ if $h(i) \in \Phi_V$; 2) $Y \not\equiv \bot$ if $h(j) \in \Phi_V$, and one of the following conditions is satisfied

a) $X \approx_{\mathscr{B}} Y$;

b) i) $X \equiv \top$ and there is a set of formulas $\mathscr{A} = \{Z_1, \dots, Z_n\}$ such that for all k, $1 \leq k \leq n$, $Z_k : (W_{m_k}^{\top}, w) \in \mathscr{B}$ and $\mathscr{A}$ c-fulfils Y; or

ii) $Y \equiv \top$ and there is a set of formulas $\mathscr{A} = \{Z_1, \dots, Z_n\}$ such that for all k, $1 \leq k \leq n$, $Z_k : (W_{m_k}^{\top}, w) \in \mathscr{B}$ and $\mathscr{A}$ c-fulfils X.

c) i) X is of type α, $Y \approx_{\mathscr{B}} \alpha_n$ for $n \in \{1, 2\}$, and there is a set of formulas $\mathscr{A} = \{Z_1, \dots, Z_n\}$ such that for all k, $1 \leq k \leq n$, $Z_k : (W_{m_k}^{Y_{\approx_{\mathscr{B}}}}, w) \in \mathscr{B}$ and $\mathscr{A}$ c-fulfils α_{3-n}; or

ii) Y is of type α, $X \approx_{\mathscr{B}} \alpha_n$ for $n \in \{1, 2\}$, and there is a set of formulas $\mathscr{A} = \{Z_1, \dots, Z_n\}$ such that for all k, $1 \leq k \leq n$, $Z_k : (W_{m_k}^{X_{\approx_{\mathscr{B}}}}, w) \in \mathscr{B}$ and $\mathscr{A}$ c-fulfils α_{3-n}.

According to 1) and 2) no label unifies with an unrestricted label whose label formula is unsatisfiable. Intuitively, this excludes that two propositionally indexed sets of worlds have a null intersection, which would be possible with an unrestricted label indexed with a contradiction: since $f(Y,u) = \emptyset$ if $Y \equiv \bot$, so the "denotation" of the label is empty. Indeed $||\bot|| = \emptyset$, and, by reflexivity, for each $A \in L_{>}$ and $u \in W$, $f(A,u) \subseteq ||A||$, hence $f(\bot,u) = \emptyset$.

Clause a) corresponds to Left Logical Equivalence and CSO: both establish when two formulas are equivalent with respect to $\mid\!\sim$; but logically and nonmonotonically equivalent formulas have the same selection function sets.

According to b), given a set of *LS*-formulas $\mathscr{B}$ containing $A : (W_1^{C\to C}, w_1)$, the labels $(W_2^{A\to A}, w_1)$ and $(w_2^{A\vee B}, w_1)$ $\sigma_{\mathbf{C}}^{\mathscr{B}}$-unify. The intuition behind this unification is the following: $C \to C$ and $A \to A$ are two tautologies built from different propositional letters; however, the reading of $A : (W_1^{C\to C}, w_1)$ is "A is true in every $\top$-world with respect to w_1"; this means $f(\top, w_1) \subseteq \|A\|$. On the other hand $\{A\}$ c-fulfils $A \vee B$, thus, by Proposition 17 $\|A\| \subseteq \|A \vee B\|$, hence $f(\top, u) \subseteq \|A \vee B\|$. Therefore, by cumulativity (2), $f(\top, w_1) = f(A \vee B, w_1)$.

Clause c) is meant to characterize cumulativity. Cumulativity is a restricted version of Left Weakening. Accordingly, we have to see whether a conjunction is a weakening of one conjunct and the other conjunct is derivable in each minimal world with respect to the former component. This is achieved thanks to the notion of c-fulfilment. Such a notion is nothing else than the condition a set of formulas must satisfy to (propositionally) entail the formula which is "fulfilled". Notice that the notion of c-fulfilment is also strictly related to Right Weakening. As an example, consider the following labels: $i = (w_2^{A\wedge(C\to(B\wedge D))}, w_1)$, $j = (W_1^A, w_1)$, and the following *LS*-formulas: $\mathscr{A}_1 = TB : (W_2^A, w_1)$, $\mathscr{A}_2 = TD : (W_3^A, w_1)$. Here $(i,j)\sigma_{\mathbf{C}}^{\mathscr{B}}$, where $\mathscr{B}$ contains $\mathscr{A}_1$ and $\mathscr{A}_2$. Notice that $A \wedge (C \to (B \wedge D))$ is of type α, and A is α_1. Moreover $\{B, D\}$ c-fulfils $B \wedge D$ which, in turn, c-fulfils $C \to (B \wedge D)$, i.e., α_2. Thus $\mathscr{B}$ contains a set of *LS*-formulas whose labels are appropriate, and whose declarative units c-fulfil α_2.

Although the above conditions seem to be very cumbersome, as we shall see in section 5, they can be easily detected by the *LS*-formulas occurring in a proof tree, and closely correspond to the semantic conditions of *SFC*-models.[3]

3.3 Inference Rules

The heart of the proof system for **C** is constituted by the following rules which are designed to work both as inference rules (to make deductions from both the declarative and the labelled part of *LS*-formulas), and as ways of manipulating labels during proofs. In what follows we write $(i,j)\sigma_{\mathbf{C}}^{\mathscr{B}}$ to denote both that i and j are $\sigma_{\mathbf{C}}^{\mathscr{B}}$-unifiable and the result of their $\sigma_{\mathbf{C}}^{\mathscr{B}}$-unification, and $\bar{X}$ to denote the conjugate of X (i.e., $\bar{X} = FA$ (or TA) if $X = TA$ (or FA)).

$$\frac{\alpha @ X, k^Y}{\alpha_n : k^Y}[n \in \{1,2\}] \tag{α}$$

$$\frac{\begin{array}{c}\beta @ X, k^Y\\ \bar{\beta}_{3-n} @ X', j^{Y'}\end{array}}{\beta_n : (k,j)\sigma_{\mathbf{C}}^{\mathscr{B}}}[n \in \{1,2\}] \tag{β}$$

[3] In particular, see Lemma 24 in Section 4. This lemma shows that if two labels i and j unify and they have a non-null intersection, then the result of their unification corresponds to an element of the appropriate model. In other words in Lemma 24 we shall prove that, provided that i and j unify, there exists a corresponding world in the model.

These are exactly the α and β rules of the original *KEM* method [1] in a slightly modified version: the formulas the rule is applied to are either the declarative parts or the label formulas. The α rules are just the usual linear branch-expansion rules of the tableau methods, whereas the β rules correspond to such common natural inference patterns as modus ponens, modus tollens, disjunctive syllogism, etc.

$$\frac{TA \mathrel{|\!\!\sim} B@X, i^{Y}}{TB : (W_n^A, i^Y)}[W_n^A \text{ new}] \tag{T$\mathrel{|\!\!\sim}$}$$

$$\frac{FA \mathrel{|\!\!\sim} B@X, i^{Y}}{FB : (w_n^A, i^Y)}[w_n^A \text{ new}] \tag{F$\mathrel{|\!\!\sim}$}$$

The rules $T\mathrel{|\!\!\sim}$ and $F\mathrel{|\!\!\sim}$ closely reflect the semantical evaluation clause 1 for $>$ (see section 2 above). In other words, whenever $A > B$ is true at a world u, B should be true at all the worlds in $f(A,u)$ (A-worlds); and whenever $A > B$ is false at u, there should be some A-world where B is false. Thus, such rules correspond to the elimination rules for $>$ and their structure derives from *KEM* modal rules ν and π (the elimination rules for the standard modal operators in a labelled context, see [1]).

$$\frac{}{X : i \qquad \bar{X} : i}[i \text{ unrestricted}] \tag{PB}$$

$$\frac{\begin{array}{c} X@Y, i^{Y'} \\ \bar{X}@Z, k^{Z'} \end{array}}{\times}[(i,k)\sigma_C^{\mathscr{B}}] \tag{PNC}$$

PB (the "Principle of Bivalence") is exactly the "cut" rule of the original method (it can be thought of as the semantic counterpart of the cut rule of the sequent calculus). *PNC* (the "Principle of Non-Contradiction") is the modified version of the familiar branch-closure rule of the tableau method. As it stands, it allows closure ("×") to follow from two formulas which are contradictory "in the same world", represented by two $\sigma_{\mathbf{C}}^{\mathscr{B}}$-complementary *LS*-formulas, i.e., two *LS*-formulas $X : i^{Y'}$ and $\bar{X} : k^{Z'}$ whose labels are $\sigma_{\mathbf{C}}^{\mathscr{B}}$-unifiable (such as, e.g, $TC : (W_1^{A\vee B}, w_1)$ and $FC : (w_3^{A\vee B}, w_1)$). Notice that, in contrast with the usual normal modal setting, in the present setting a contradiction of the form $FA : (w_2^A, w_1)$ may occur, since this *LS*-formula states that there exists an A-world where A is false.

Now we are ready to introduce the formal definitions of a *KEM* tableau and a *KEM* proof for a given formula.

Definition 23 A *KEM-tableau* (or simply tableau) for an *LS*-formula $X : i$ is a tree, whose root is $X : i$ and nodes are *LS*-formulas obtained from previous nodes using the inference rules of *KEM*. A branch is *closed* when it is possible to apply PNC; a tree is *closed* when all its branches are closed. A *KEM-proof* (or simply proof) of A is a closed *KEM*-tableau for $FA : i$, where i is in Φ_C. Finally $\vdash_{KEM} A$ means that there is a *KEM*-proof for A.

In the following section the above set of rules will be proved to be sound and complete for **C**. Notice that the format of the rules prevents labels from having a length greater than two. This is obviously due to the fact that **C** corresponds to $\mathbf{CU}^-$ (in the context of **C** the nesting of $\mathrel{|\!\!\sim}$ is meaningless).

4 Soundness and Completeness

In this section we prove soundness and completeness theorems for *KEM*. We shall proceed as usual by first proving that the rules for **C** are derivable in *KEM*, and then that the rules of *KEM* are sound with respect to the semantics for **C**.

In the course of *KEM*-proofs labels are generated according to the structure of the formulas involved, and, at the same time, they also generate (counter)-models. The labels are intended to denote possible worlds and relations among them. The idea is that all the relevant information is recorded in the labels. So, to extract such information, we have to map labelled signed formulas to elements of *SFC* models. This is achieved with the help of three functions, namely g, r, and m. The function g will map labels to sets of possible worlds: a singleton for constants, a set of worlds (possibly empty) for variables. Moreover this set should satisfy some constraints. The selection function f is assumed to be closed under the conditions specifying cumulativity, but, we want to reconstruct it, through r, from the labels: path labels are intended to represent not only worlds, but also record the selection function. Finally, m, given an *LS*-formula, returns the evaluation of the formula with respect to the world(s) denoted by its label. Let us now define these three functions which map labels into elements of *SFC* models.

Let g be a function from $\Im$ to 2^W defined thus:

$$g(i^X) = \begin{cases} \{w_i\} \subseteq f(X, g(h(i))) & \text{if } h(i^X) \in \Phi_C \\ \{w_i \in W : w_i \in f(X, g(h(i)))\} & \text{if } h(i^X) \in \Phi_V \\ \{w_i\} & \text{if } i \in \Phi_C \\ W & \text{if } i^X \in \Phi_V \end{cases}$$

Let r be a function from $\Im$ to f defined thus:

$$r(i^X) = \begin{cases} \emptyset & \text{if } \ell(i) = 1 \\ f(X, g(i^X)) & \text{if } \ell(i) > 1 \end{cases}$$

Let m be a function from *LS*-formulas to v thus defined:

$$m(SA@i^X) =_{def} v(A, w_j) = S$$

for all $w_j \in g(i^X)$.

Lemma 24 Let $\mathscr{B}$ be a set of *LS*-formulas and let i, j be labels in $\mathscr{B}$. If $(i^X, j^Y)\sigma_{\mathbf{C}}^{\mathscr{B}}$, then $g(i^X) \cap g(j^Y) \neq \emptyset$.

This lemma, proved by induction on the length of labels, states that whenever two labels unify, their denotations have a non-null intersection (the result of their unification).

Proof We confine ourselves to the case where i, j are both of length 2 and $b(i) = b(j) = w_1$. By definition, the following cases are present: 1) $h(i^X), h(j^Y)$ are two constants; 2) $h(i^X)$ is a constant and $h(j^Y)$ is a variable (the opposite case is analogous).

Case 1) Two constants unify iff they are the same constant. But then, according to the rules of *KEM*, they have the same label formula X, and thus $g(i^X) = g(j^Y)$, from which it follows $g(i^X) \cap g(i^Y) \neq \emptyset$.

Case 2) We first note that $g(i^X) = f(X,u)$ and $g(j^Y) = f(Y,u)$, where $\{u\} = g(b(i)) = g(b(j))$ since we have assumed, granted unification, that $b(i) = b(j)$. If $h(i) \in \Phi_C$ and $h(j) \in \Phi_V$ then $g(i^X)$ is a world in the set of X-worlds with respect to u, while $g(j^Y)$ is the set of Y-worlds with respect to u. Then we have to see the relations between the set of X-worlds and the set of Y-worlds. The two labels $\sigma_{\mathrm{C}}^{\mathscr{B}}$-unify: this means that at least one among conditions a), b) and c) of Definition 22 is satisfied.

Case a) We are now going to prove that if $X \approx_{\mathscr{B}} Y$, then $f(X,u) = f(Y,u)$. It is immediate to see that $\approx_{\mathscr{B}}$ is an equivalence relation in $\mathscr{B}$, so we prove only the cases where $A \equiv B$ (that is the classical equivalence relation), and when X and Y support each other. If $X \equiv Y$, according to condition 2 of Definition 2, $f(X,u) = f(Y,u)$; since the labels unify $Y \not\equiv \bot$, and $g(i^X) \neq \emptyset$, so $g(i^X) \cap g(j^Y) \neq \emptyset$.

Otherwise if X supports Y, then either

i) there is a set of formulas $\{Z_1,\dots,Z_n\}$ that c-fulfils Y and for each formula Z_k in such a set $Z_k : (W_k^X, w_1)$ occurs in $\mathscr{B}$; or

ii) there is a set of formulas $\mathscr{A} = \{Z_1,\dots,Z_n\}$ such that for each formula Z_k in the set, X forces Z_k in $\mathscr{B}$, and $\mathscr{A}$ c-fulfils Y.

For i) it is immediate to see that if $\{Z_1,\dots,Z_n\}$ c-fulfils Y, then $||\bigwedge_{k=1}^n Z_n|| \subseteq ||Y||$. So,

$$f(X,u) \subseteq ||Z_k||,\ 1 \leq k \leq n,$$

which implies

$$f(X,u) \subseteq ||Y||.$$

For ii) As $\mathscr{A}$ forces each Z_k in $\mathscr{B}$, it is immediate to see that $||X|| \subseteq ||Z_k||$, so $||X|| \subseteq ||\bigwedge_{k=1}^n Z_k||$. From the fact that $\{Z_1,\dots,Z_n\}$ c-fulfils Y, we obtain $||\bigwedge_{k=1}^n Z_n|| \subseteq ||Y||$. By reflexivity $f(X,u) \subseteq ||X||$, therefore $f(X,u) \subseteq ||Y||$. Thus we have the same result in both cases. We can repeat the same argument when Y supports X, obtaining $f(Y,u) \subseteq ||X||$. If X supports Y, then, by (2) $f(X \wedge Y,u) = f(Y,u)$. If Y supports X, then, again by (2), $f(X \wedge Y,u) = f(X,u)$; therefore $f(X,u) = f(Y,u)$.

Case b) If $X \equiv \top$, then there is a set of formulas $\mathscr{A} = \{Z_1,\dots,Z_n\}$ such that i) $\mathscr{A}$ c-fulfils X and ii) for all Z_k in $\mathscr{A}$, we have $Z_k : (W_{k_m}^{\top}, w_1) \in \mathscr{B}$. From i) we obtain that $\|\bigwedge_{k=1}^n Z_k\| \subseteq \|Y\|$; moreover, we know that $g(i^X) = f(\top,u)$, and ii) implies that $f(\top,u) \subseteq \|Z_k\|$ for each Z_k in $\mathscr{A}$, from which we have

$$f(\top,u) \subseteq \bigcap_{1 \leq k \leq n} \|Z_k\| = \left\| \bigwedge_{k=1}^n Z_k \right\| \subseteq \|Y\| \ ;$$

from this and the property (2) we conclude $f(\top,u) = f(Y,u)$. Therefore also in this case $g(i^X) \cap g(j^Y) \neq \emptyset$; the case $Y \equiv \top$ is analogous.

Case c) For the remaining condition of Definition 22 let us suppose that X is of type α, Y is α_1, Z is α_2, $\{Z_1,\dots,Z_n\}$ c-fulfils Z, and $Z_1,\dots,Z_n$ are in the branch with the right label, namely j^Y (the other case, namely Y is of type α, is similar). As $\{Z_1,\dots,Z_n\}$ c-fulfils Z, $||\bigwedge_{k=1}^n Z_n|| \subseteq ||Z||$. So,

$$f(Y,u) \subseteq ||Z_k||,\ \text{for all } k, 1 \leq k \leq n,$$

which implies

$$f(Y,u) \subseteq ||Z||;$$

by the semantic conditions we have

$$f(Y \wedge Z,u) = f(Y,u)$$

but $f(Y \wedge Z,u) = f(X,u)$. From the last condition of Definition 2, and the fact that $g(i^X) \neq \emptyset$ we can also conclude in this case that $g(i^X) \cap g(j^Y) \neq \emptyset$.

Lemma 25 Let $\mathscr{B}$ be a set of *LS*-formulas and let i,j be labels in $\mathscr{B}$. If $m(SA : i)$, and $(i,j)\sigma_{\mathbf{C}}^{\mathscr{B}}$, then $m(SA : (i,j)\sigma_{\mathbf{C}}^{\mathscr{B}})$.

Proof Let us suppose that the lemma does not hold. Thus the proof trivially follows from Lemma 24 and the definition of m.

According to the previous lemma if a formula has a given evaluation in a world denoted by a label, and this label unifies with another label, then the value of the formula remains unchanged in the worlds corresponding to the unification of the labels. This fact allows us to verify the correctness of any rule in a standard semantic setting, whence the following lemma.

Lemma 26 If $\vdash_{KEM} A$, then $\models_{SFC} A$.

Proof The α-rules and *PB* are obviously sound rules in **CU**, insofar as they are local rules, they do not involve unifications (see also [13] for a proof of the soundness of such rules in a classical propositional setting). For the β-rules and *PNC*: by the hypothesis $(i,j)\sigma_C^{\mathscr{B}}$, then, by Lemma 25, the formulas involved have the same value in $g(i)$, $g(j)$ and $g((i,j)\sigma_C^{\mathscr{B}})$; after that these rules become rules of *KE* (classical propositional rules), and thus they are sound rules in **CU**. For $T{\mid\!\sim}$: let us suppose that it does not hold, then $m(TA \mid\!\sim B : w_1)$ and not $m(B : (W_1^A, w_1))$. The former implies $v(A \mid\!\sim B, g(w_1)) = T$ which is equivalent to $f(A, g(w_1)) \subseteq ||B||$. On the other hand we have that for some $w \in f(A, g(w_1))$ $v(B,w) = F$, i.e., $w \in ||\neg B||$, but this implies that $f(A, g(w_1)) \not\subseteq ||B||$, so we have a contradiction. The proof for $F{\mid\!\sim}$ is similar.

Lemma 27 Let Γ be a set of conditional assertions, and A be a conditional assertion. If Γ cumulatively entails A, then $\vdash_{KEM} \bigwedge\Gamma \to A$.

Proof We show that the inference rules and the axioms of **C** are derivable in *KEM*. D'Agostino and Mondadori [13] have shown that *KE* is sound and complete for classical propositional logic and enjoys the property of transitivity of deductions. We provide *KEM*-proofs for Reflexivity, Left Logical Equivalence, Right Weakening, Cautious Monotonicity and Cut.

Reflexivity

$$\begin{array}{ll}
1. FA \mid\!\sim A & w_1 \\
2. FA & (w_2^A, w_1) \\
3. \times & (w_2^A, w_1)
\end{array}$$

Notice that closure follows from having two complementary formulas FA and A both labelled with (w_2^A, w_1).

Left Logical Equivalence

$$\begin{array}{ll}
1.\,TA \mid\!\sim C & w_1 \\
2.\,FB \mid\!\sim C & w_1 \\
3.\,TC & (W_1^A, w_1) \\
4.\,FC & (w_2^B, w_1) \\
5.\,\times & (w_2^B, w_1)
\end{array}$$

Here closure is obtained from $TC : (W_1^A, w_1)$ and $FC : (w_2^B, w_1)$. The labels $\sigma_C^{\mathscr{B}}$-unify due to the equivalence of the label formulas: by hypothesis A and B are equivalent.

Right Weakening

$$\begin{array}{lll}
1. & TA \mid\!\sim B & w_1 \\
2. & FA \mid\!\sim C & w_1 \\
3. & TB & (W_1^A, w_1) \\
4. & FC & (w_2^A, w_1)
\end{array}$$

$$\begin{array}{llll|llll}
5. & TB \rightarrow C & (w_2^A, w_1) & & 6. & FB \rightarrow C & (w_2^A, w_1) \\
7. & TC & (w_2^A, w_1) & & & \times & \\
8. & \times & (w_2^A, w_1) & & & &
\end{array}$$

Notice that we have applied *PB* to $B \rightarrow C$ with respect to (w_2^A, w_1). The right branch is closed since, by hypothesis, we have already a *KEM* proof for $B \rightarrow C$.

Cautious Monotonicity

$$\begin{array}{ll}
1.\,TA \mid\!\sim B & w_1 \\
2.\,TA \mid\!\sim C & w_1 \\
3.\,FA \wedge B \mid\!\sim C & w_1 \\
4.\,TB & (W_1^A, w_1) \\
5.\,TC & (W_2^A, w_1) \\
6.\,FC & (w_2^{A \wedge B}, w_1) \\
7.\,\times & (w_2^{A \wedge B}, w_1)
\end{array}$$

In this proof and in the next we can close the trees because of the condition c) of Definition 22.

The label formula of FC is of type α $(A \wedge B)$, and the label formula of TC is α_1 (A). Moreover the branch contains a formula that c-fulfils α_2 with the appropriate label ($TB : (W_1^A, w_1)$), thus the formulas in 5 and 6 are $\sigma_C^{\mathscr{B}}$-complementary since their labels $\sigma_C^{\mathscr{B}}$-unify.

Cut

$$\begin{array}{ll}
1.\,TA \wedge B \mid\!\sim C & w_1 \\
2.\,TA \mid\!\sim B & w_1 \\
3.\,FA \mid\!\sim C & w_1 \\
4.\,TC & (W_1^{A \wedge B}, w_1) \\
5.\,TB & (W_2^A, w_1) \\
6.\,FC & (w_2^A, w_1) \\
7.\,\times & (w_2^A, w_1)
\end{array}$$

The label formula of TC is of type α, and the label formula of FC is α_1. Moreover the branch contains a formula that c-fulfils α_2 with the appropriate label. Hence we can again apply conditin c) of Definition 22; thus the formulas in 4 and 6 are $\sigma_C^{\mathscr{B}}$-complementary since their labels $\sigma_C^{\mathscr{B}}$-unify.

From Theorem 4, Lemmas 27 and 26 we obtain

Theorem 28 $\vdash_{KEM} A$ iff $\models_{SFC} A$.

and from Theorem 28 and Corollary 12

Corollary 29 Let Γ be a set of conditional assertions. Γ cumulatively entails $A \mathrel{|\!\sim} B$ iff $\vdash_{KEM} \bigwedge\Gamma \rightarrow (A \mathrel{|\!\sim} B)$.

5 Proof Search with KE^{+}

5.1 Introduction

It is easy to see, from the above definition of unification and the form of the inference rules, what problems arise for a tableau system which computes nonmonotonic consequence relations in a *CL* setting.[4] Each time we have to unify two labels we have to verify that a complex relation between two formulas A and B holds. Essentially this relation amounts to the following three cases: 1) $A \approx_{\mathscr{B}} B$, 2) $A \equiv B$, 3) either A or B is a classical tautology.

We shall examine these cases with the help of examples.

Let us first consider the set of conditionals $\Gamma = \{A \mathrel{|\!\sim} B, A \mathrel{|\!\sim} C, A \wedge B \mathrel{|\!\sim} D, C \mathrel{|\!\sim} A \wedge B\}$. It is easy to prove that Γ cumulatively entails $C \mathrel{|\!\sim} D$. In fact from $A \mathrel{|\!\sim} B$ and $A \mathrel{|\!\sim} C$ we can derive, by cumulativity, $A \wedge B \mathrel{|\!\sim} C$, which, together with $A \wedge B \mathrel{|\!\sim} D$, implies $C \mathrel{|\!\sim} D$ by CSO.

We give now the *KEM* proof of this entailment:

1. $TA \mathrel{|\!\sim} B$	w_1
2. $TA \mathrel{|\!\sim} C$	w_1
3. $TA \wedge B \mathrel{|\!\sim} D$	w_1
4. $TC \mathrel{|\!\sim} A \wedge B$	w_1
5. $FC \mathrel{|\!\sim} D$	w_1
6. TB	(W_1^A, w_1)
7. TA	(W_1^A, w_1)
8. TC	(W_2^A, w_1)
9. TA	(W_2^A, w_1)
10. TD	$(W_3^{A\wedge B}, w_1)$
11. $TA \wedge B$	$(W_3^{A\wedge B}, w_1)$
12. TA	$(W_3^{A\wedge B}, w_1)$
13. TB	$(W_3^{A\wedge B}, w_1)$
14. $TA \wedge B$	(W_4^C, w_1)
15. TA	(W_4^C, w_1)
16. TB	(W_4^C, w_1)
17. TC	(W_4^C, w_1)
18. FD	(w_2^C, w_1)
19. TC	(w_2^C, w_1)
20. $\times$	

To close the tree we have to verify that the formulas in steps 10 and 18 are $\sigma_{\mathbf{C}}^{\mathscr{B}}$-complementary; this means that their labels $\sigma_{\mathbf{C}^{\mathscr{B}}}$-unify, that is their label formulas meet the conditions stated

[4] Accordingly, similar remarks hold for a tableau system for conditional logics.

in Definition 22. In particular we have that $A \wedge B \approx_{\mathscr{B}} D$. To check this, it is enough to notice that the branch at hand is a superset of the set of *LS*-formulas considered in Example 21. As we can see from this example the notion of two formulas being $\mathrel{\mid\!\sim}$-equivalent ($\approx_{\mathscr{B}}$) can be mainly detected in *KEM* proofs by inspection of the formulas occurring in the tree.

Two formulas can be $\mathrel{\mid\!\sim}$-equivalent just because they are classically equivalent. Therefore the next question concerns the treatment of equivalences. For example, it is quite obvious that the assertion $(A \rightarrow B) \mathrel{\mid\!\sim} C$ is cumulatively entailed by a set of assertions which includes $(\neg A \vee B) \mathrel{\mid\!\sim} C$, because of the equivalence between the antecedents. How to check it in our proof system? In general, the easiest solution might seem to open an auxiliary tree each time we have to check whether two labels unify, which means that the label formulas turn out to be equivalent[5]. Indeed, this solution leads to a remarkable increase in the complexity of proof as the following tree shows.

$$\begin{array}{lr}
1. T(\neg A \vee B) \mathrel{\mid\!\sim} C & w_1 \\
2. F(A \rightarrow B) \mathrel{\mid\!\sim} C & w_1 \\
3. TC & (W_1^{\neg A \vee B}, w_1) \\
4. FC & (w_2^{A \rightarrow B}, w_1)
\end{array}$$

Notice that the tree closes if the labels of the complementary formulas TC and FC $\sigma^{\mathscr{B}}_{\mathrm{C}}$-unify, which happens iff $\neg A \vee B$ is equivalent to $A \rightarrow B$. Accordingly, we have to open a new tree for proving their equivalence:

$$F(\neg A \vee B) \equiv (A \rightarrow B)$$

$$\begin{array}{ll}
T\neg A \vee B & F\neg A \vee B \\
FA \rightarrow B & TA \rightarrow B \\
TA & TA \\
FB & FB \\
TB & TB
\end{array}$$

Let us now consider the set of conditional assertions Γ such that $\{(\top \mathrel{\mid\!\sim} B), (B \mathrel{\mid\!\sim} \top), (A \vee \neg A) \mathrel{\mid\!\sim} C)\} \subseteq \Gamma$. It is not hard to see that Γ cumulatively entails the conditional assertion $B \mathrel{\mid\!\sim} C$. However, let us suppose that it is not the case. Since $A \vee \neg A \mathrel{\mid\!\sim} C$ is equivalent to $\top \mathrel{\mid\!\sim} C$ this means that C holds in all the $\top$-worlds. Furthermore, $f(\top, u) = f(B, u)$. On the other hand, according to the conclusion there is a B-world where C is false. Thus, since a B-world is also a $\top$-world, there is a $\top$-world where C is at the same time true and false. Let us see the *KEM*-tree for this case.

$$\begin{array}{lr}
1. T(A \vee \neg A) \mathrel{\mid\!\sim} C & w_1 \\
2. T\top \mathrel{\mid\!\sim} B & w_1 \\
3. TB \mathrel{\mid\!\sim} \top & w_1 \\
4. TB & (W_1^{\top}, w_1) \\
5. T\top & (W_2^{B}, w_1) \\
6. TC & (W_3^{A \vee \neg A}, w_1) \\
7. FC & (w_2^{B}, w_1)
\end{array}$$

[5] See Section 3.2.

The steps from 1 to 7 are straightforward. At this point we have two complementary formulas (6 and 7), so we have to check whether they are also $\sigma_{\mathbf{C}}^{\mathscr{B}}$-complementary (i.e, complementary under the $\sigma_{\mathbf{C}}^{\mathscr{B}}$-unification of their labels). In this case we have to go on with the proof, because the label formula $A \vee \neg A$ has not yet been analysed. Let $\mathscr{D}$ stand for the first 7 steps of the proof. Thus we have:

$$\begin{array}{c}
\mathscr{D} \\
\begin{array}{lll|lll}
8. & TA & (W_1^{A\vee\neg A}, w_1) & 9. & FA & (W_1^{A\vee\neg A}, w_1) \\
10. & TA & (W_1^{A\vee\neg A}, w_1) & 11. & FA & (W_1^{A\vee\neg A}, w_1)
\end{array}
\end{array}$$

It is immediate to see that both branches are open, but a closer inspection of the tree reveals that it satisfies the condition of Theorem 36 below, thus $A \vee \neg A \equiv \top$, from which it follows that $(W_1^{A\vee\neg A}, w_1)$ is equivalent to $(W_1^{\top}, w_1)$. Therefore the label $(W_1^{A\vee\neg A}, w_1)$ $\sigma_{\mathbf{C}}^{\mathscr{B}}$-unifies with (w_2^{B}, w_1); thus the tree is closed. Alternatively, we could open a tree for the negation of all non-atomic label formulas. The above example makes an essential use of $B \mid\!\sim \top$, but such a formula expresses a general property of **C**, thus is present, implicitly, in every nonmonotonic theory.

Let us now consider the conditional assertion

$$(A \wedge \neg A) \mid\!\sim B \tag{4}$$

which trivially holds, with the corresponding tree

$$\begin{array}{ll}
1. F(A \wedge \neg A) \mid\!\sim B & w_1 \\
2. FB & (w_2^{A\wedge\neg A}, w_1)
\end{array}$$

What tree have we now to develop? That for $TA \wedge \neg A$ or that for $FA \wedge \neg A$? According to the above remarks, we would have to develop the tree for $FA \wedge \neg A$, but in the present case this will not lead us to the desired result, since all we learn from an open tree is that the negation of the formula is satisfiable. On the other hand, going on with the application of the inference rules generates the following tree:

$$\begin{array}{ll}
1. F(A \wedge \neg A) \mid\!\sim B & w_1 \\
2. FB & (w_2^{A\wedge\neg A}, w_1) \\
3. TA & (w_2^{A\wedge\neg A}, w_1) \\
4. FA & (w_2^{A\wedge\neg A}, w_1) \\
5. \times &
\end{array}$$

Do the above remarks mean that we are doomed to have a proof system which is computationally unsatisfactory? Of course, from a proof-theoretical point of view it is not incorrect to resort to auxiliary trees each time we have to compare label formulas. The point is that such a strategy can be viewed as redundant because it uses sub-proofs to verify properties that theoretically can be proved by referring only to the main proof-tree. Thus, in the next section we shall develop a particular proof search procedure which allows us not to open auxiliary tableaux or external oracles.

5.2 The Proof Search

The unification presented in section 3.2 compels us to check (label) formulas either for validity or for logical equivalence. As discussed in the previous section, this can be a very expensive task whose accomplishment may require us to open an auxiliary proof tree whenever we have to check either condition (see [2] for details). Fortunately, as we said above the main tree provides all the information we need so that we only have to make it available by appealing to a suitable proof method. One such method is provided by the classical system KE^+. KE^+ is based on D'Agostino and Mondadori [13]'s *KE*, a tableau-like proof system which employs a mixture of tableau, natural deduction and structural rules (in practice, the α-, β-, *PB* and *PNC* rules of section 3.3 restricted to the propositional part). KE^+ uses the same rules but adopts a different proof search procedure which makes it completely analytical and able to detect whether 1) a formula is either a tautology, or a contradiction, or only a satisfiable one; and 2) a sub-formula of the formula to be proved is a tautology, and to use this fact in the proof of the initial formula. The KE^+ based method consists of verifying whether the truth of the conjugate of an immediate sub-formula of a β-formula implies the truth of the other immediate sub-formula. If it is so, then we have enough information to conclude that the formula is provable. This result follows from the fact that the branch beginning with $\bar{\beta}_n$ $(n \in \{1,2\})$ contains no pair of complementary formulas leading to closure. This in turn is proved by seeing whether a formula occurs twice in a branch, and that those occurrences "depend on" appropriate formulas. This last notion is captured by the following definition.

Definition 30 Each formula *depends on* itself. A formula *B depends on A* either if it is obtained by an application of the α-rule or it is obtained by an application of the *KE* rules on formulas depending on A. A formula C *depends on* A,B if it is obtained by an application of a β-rule with A,B as its premises. The formulas obtained by an application of *PB depend on* the formula *PB* is applied to. If C *depends on* A,B then C *depends on* A and C *depends on* B.

We go now to the proof search, but first we need some terminology and definitions.

Definition 31 An α-formula is *analysed* in a branch when both α_1 and α_2 are in the branch. A β-formula is *analysed* in a branch when either 1) if $\bar{\beta}_1$ is in the branch also β_2 is in the branch, or 2) if $\bar{\beta}_2$ is in the branch also β_1 is the branch.

A β formula is said to be *fulfilled* in a branch if: 1) either β_1 or β_2 occurs in the branch provided they depend on β, or 2) either β_1 or β_2 is obtained from applying *PB* on β.

Definition 32 A branch is *E-completed* if all the formulas occurring in it are analysed. A branch is *completed* if it is E-completed and all the β-formulas occurring in it are fulfilled.

Definition 33 A branch is *closed* if it ends with an application of *PNC*. A tree is *closed* if all its branches are closed.

Definition 34 A branch obtained by applying *PB* to a β-formula with $\bar{\beta}_i$ as its root is a $\bar{\beta}$*-branch*. Each branch generated by an application of *PB* to a formula occurring in a $\bar{\beta}$-branch is a $\bar{\beta}$-branch. A *semi* $\bar{\beta}$-branch is a branch obtaind from a $\bar{\beta}$-branch by removing the formulas depending only on the root $\bar{\beta}_i$. A branch generated by an application of *PB* which is not a $\bar{\beta}$-branch is a β-branch.

Definition 35 A branch is a $\top$-branch if it contains only formulas which are equivalent to $\top$ and the formulas depending on them.

The proof search procedure starts with the formula to be proved; then

1. it selects a $\bar{\beta}$-branch ϕ which is not yet completed and which is the $\bar{\beta}$-branch with respect to the greatest number of formulas;
2. if ϕ is not E-completed, it expands ϕ by means of the α- and β-rules until it becomes E-completed;[6]
3. if the resulting branch is neither completed nor closed, then it selects a β-formula which is not yet fulfilled in the branch — if possible a β-formula resulting from step 2 — then it applies PB with $\beta_1, \bar{\beta}_1$ (or, equivalently $\beta_2, \bar{\beta}_2$), and then it returns to step 1; otherwise it returns to step 1.

Theorem 36 [24] For a formula A, $A \equiv \top$ if either:

1. in one of the $\bar{\beta}$-branches it generates there is an LS-formula which appears twice, and one occurrence depends on $\bar{\beta}_n, n \in \{1,2\}$, and the other depends on β, or
2. each $\bar{\beta}$-branch is closed and the β-branches are $\top$-branches, or
3. each semi $\bar{\beta}$-branch is a $\top$-branch.

Proof A proof for a formula A is a closed tree for $\bar{A}$. In other words every branch in a tree for $\bar{A}$ contains a pair of complementary formulas B and $\bar{B}$. Let us now recall some relationships among complementary formulas:

$$\alpha = \bar{\beta} \qquad \alpha_1 = \bar{\beta}_1 \qquad \alpha_2 = \bar{\beta}_2 \tag{5}$$

$$\beta = \bar{\alpha} \qquad \beta_1 = \bar{\alpha}_1 \qquad \beta_2 = \bar{\alpha}_2 \tag{6}$$

It is clear that the complement of a formula of type α is a formula of type β and the other way around. Moreover this relationship is also true for their components.

At this point we examine the structure of the trees for formulas of type α and type β.

$$\begin{array}{c} \text{tree for } \alpha \\ \alpha \\ \alpha_1 \\ \alpha_2 \end{array} \qquad\qquad \begin{array}{cc} \multicolumn{2}{c}{\text{tree for } \beta,\ n \in \{1,2\}} \\ \multicolumn{2}{c}{\beta} \\ \bar{\beta}_n & \beta_n \\ \beta_{3-n} & \end{array}$$

Let us consider two complementary formulas α and β. Given the relationships in (5) and (6) we have

$$\begin{array}{c} \text{tree for } \alpha \\ \alpha \\ \alpha_1 \\ \alpha_2 \end{array} \qquad \begin{array}{cc} \multicolumn{2}{c}{\text{tree for } \beta = \bar{\alpha}} \\ \multicolumn{2}{c}{\bar{\alpha}} \\ \alpha_1 & \bar{\alpha}_1 \\ \bar{\alpha}_2 & \end{array} \qquad \begin{array}{cc} \multicolumn{2}{c}{\text{tree for } \beta = \bar{\alpha}} \\ \multicolumn{2}{c}{\bar{\alpha}} \\ \alpha_2 & \bar{\alpha}_2 \\ \bar{\alpha}_1 & \end{array} \tag{7}$$

and

$$\begin{array}{cc} \multicolumn{2}{c}{\text{tree for } \beta} \\ \multicolumn{2}{c}{\beta} \\ \bar{\beta}_1 & \beta_1 \\ \beta_2 & \end{array} \qquad \begin{array}{cc} \multicolumn{2}{c}{\text{tree for } \beta} \\ \multicolumn{2}{c}{\beta} \\ \bar{\beta}_2 & \beta_2 \\ \beta_1 & \end{array} \qquad \begin{array}{c} \text{tree for } \alpha = \bar{\beta} \\ \bar{\beta} \\ \bar{\beta}_1 \\ \bar{\beta}_2 \end{array} \tag{8}$$

[6] For α-formulas we do not duplicate components, i.e., if α, and α_n (for $n \in \{1,2\}$) are in a branch, then we add to the branch only α_{3-n}.

To prove the theorem we have to show that each time a tree for A meets the conditions of the theorem then the tree for $\bar{A}$ is closed and the other way around, that is if a tree for A is closed then there is a tree for $\bar{A}$ satisfying the conditions of the theorem.

We prove the theorem by induction on the number n of binary connectives occurring in a formula.

Inductive base, $n = 1$. Let us consider the trees in (7). Let us suppose that the tree for α is closed, i.e., $\alpha \equiv \bot$ and, consequently, $\bar{\alpha} \equiv \top$. Then either (i) $\alpha_1 = \bot$ or (ii) $\alpha_2 = \bot$ or (iii) $\alpha_1 = \bar{\alpha}_2$.

In the first case the left branch ($\bar{\beta}$-branch) of the first tree for β is closed: it contains α_1, which is $\bot$. On the other hand in the left branch we have $\bar{\alpha}_1$ which is $\top$, and thus the branch is a $\top$-branch. If we consider the second tree then it is immediate to see that the semi $\bar{\beta}$-branch is a $\top$-branch. We can repeat the same argument for (ii).

In the last case, (iii), we consider again one of the $\bar{\beta}$-branches where we have $\alpha_n = \bar{\alpha}_{3-n}$, and in both cases there are two occurrences of the same formula with the appropriate dependencies.

We have now to analyse the case where the tree for α satisfies the conditions of the theorem. In this case we have only one β-branch, then it should be a $\top$-branch. This is possible only when both α_1 and α_2 are $\top$. Hence $\bar{\alpha}_1$ and $\bar{\alpha}_2$ are $\bot$; therefore all the branches of the tree for $\bar{\alpha}$ are closed.

It is now time to examine the trees in (8). We assume that the trees for β are closed. The only case when these trees are closed is when both β_1 and β_2 are $\bot$. This means that $\bar{\beta}_1 = \top$ and $\bar{\beta}_2 = \top$; hence the branch in the tree for $\bar{\beta}$ is a $\top$-branch and a β-branch.

If the trees for β satisfy the conditions of the theorem, then we have to explore three cases; we consider only the first tree for β, the argument for the other tree is similar: (i) $\bar{\beta}_1 = \beta_2$, (ii) the $\bar{\beta}$-branch is closed and the β-branch is a $\top$-branch, and (iii) the semi $\bar{\beta}$-branch is a $\top$-branch.

(i) If $\bar{\beta}_1 = \beta_2$, then $\bar{\beta}_1$ and $\bar{\beta}_2$ are the complement of each other, thus the tree for $\bar{\beta}$ is closed.

(ii) In this case we have that $\beta_1 = \top$, therefore $\bar{\beta}_1 = \bot$. Hence the tree for $\bar{\beta}$ is closed.

(iii) If the semi $\bar{\beta}$-branch is a $\top$-branch, then $\beta_2 = \top$ and, consequently, $\bar{\beta}_2 = \bot$; thus the tree for β is closed.

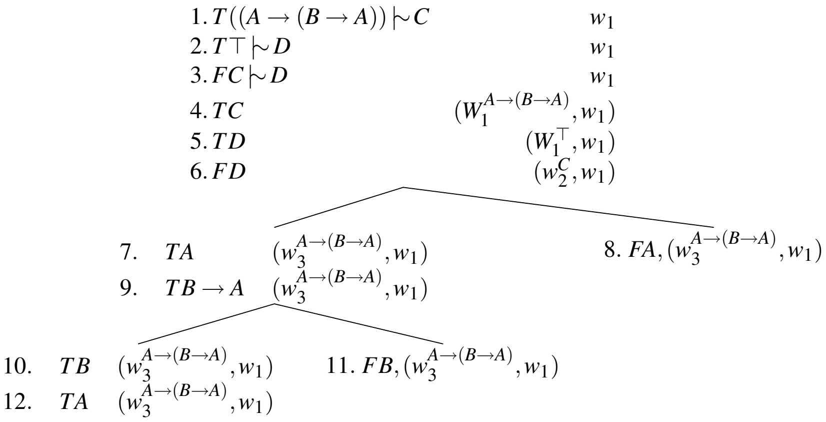
Inductive Step, $n \geq 1$. We assume that the theorem holds for formulas with up to n binary connectives.

Let us consider again the trees in (7). First of all we notice that the formulas α_1, α_2, and $\bar{\alpha}_2$ have less that n binary connectives, and thus we can use the theorem and the trees they generate to determine whether they are equivalent to $\top$ or $\bot$. Moreover α_1 is common to the tree for α and the first tree for $\bar{\alpha}$, thus the sub-tree generated from it is common to the two trees (the same is true when we consider α_2 and the second tree for $\bar{\alpha}$).

An analysis of the trees involved shows that part of the trees is common to the tree for A and the tree for $\bar{A}$, while the remaining parts are the dual of each other; i.e., if B occurs in the non-common part of A, then $\bar{B}$ occurs in the non-common part of $\bar{A}$ and vice versa.

At this point, without any loss of generality, we can assume that the conditions of the theorem can be seen from the immediate expansion of the dual parts. Then a detailed comparison of the resulting trees, using the same arguments as the inductive base, will show that whenever a tree for A meets the conditions of the theorem a corresponding tree for $\bar{A}$ is closed and the other way around.

We provide an illustration of the above notion by proving


1. $T((A \to (B \to A)) \mid\!\sim C$ w_1
2. $T\top \mid\!\sim D$ w_1
3. $FC \mid\!\sim D$ w_1
4. TC $(W_1^{A\to(B\to A)}, w_1)$
5. TD $(W_1^{\top}, w_1)$
6. FD (w_2^{C}, w_1)
7. TA $(w_3^{A\to(B\to A)}, w_1)$
8. $FA, (w_3^{A\to(B\to A)}, w_1)$
9. $TB \to A$ $(w_3^{A\to(B\to A)}, w_1)$
10. TB $(w_3^{A\to(B\to A)}, w_1)$
11. $FB, (w_3^{A\to(B\to A)}, w_1)$
12. TA $(w_3^{A\to(B\to A)}, w_1)$

Here we have to see whether the labels in 5 and 6 unify. According to definition 22 this holds if the label formula in 4 is a tautology. Notice that the label formula of W_1 is of type β and it is not yet analysed in the tree. Thus we apply *PB*. Furthermore, the left branch is a $\bar{\beta}$-branch with respect to the label formula. We then apply a β rule, and we obtain another β formula. According to the proof search we have to apply again *PB* and then we have another application of a β rule. At this point we have two occurrences of *TA* with the right dependencies. So the label formula $A \to (B \to A)$ is $\top$, and the labels in 5 and 6 unify, thus closing the tree.

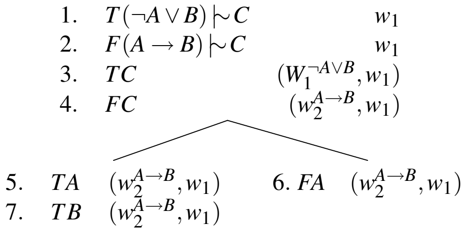
Definition 37 Let v be a function which maps each formula A into a set of (atomic) formulas in such a way that 1) if A is atomic, then $v(A) = \{A\}$; 2) if A is of type α, then $v(A) = v(\alpha_1) \cup v(\alpha_2)$; 3) if A is of type β, then $v(A) = v(\bar{\beta}_n) \cup v(\beta_{3-n})$ or $v(A) = v(\beta_n)$, $n \in \{1,2\}$. A set S of (atomic) formulas *v-fulfils* a formula A iff $S = v(A)$.

Corollary 38 Two formulas A, B are equivalent iff

- both A and B are $\top$; or
- both A and B are $\bot$; or
- each set of (atomic) formulas which v-fulfils A v-fulfils B and vice versa.

Proof The first two cases are obvious, then it suffices to note that each set S that v-fulfils a formula A corresponds to a truth-value assignment for A, and that two formulas are equivalent if they are satisfied by the same assignments.

The following proof is provided as an illustration of the use of the above notions.


1. $T(\neg A \vee B) \mid\!\sim C$ w_1
2. $F(A \to B) \mid\!\sim C$ w_1
3. TC $(W_1^{\neg A \vee B}, w_1)$
4. FC $(w_2^{A\to B}, w_1)$
5. TA $(w_2^{A\to B}, w_1)$
6. FA $(w_2^{A\to B}, w_1)$
7. TB $(w_2^{A\to B}, w_1)$

Obviously $\{TA, TB\}$ and $\{FA\}$ v-fulfil both $\neg A \vee B$ and $A \to B$. Accordingly, $(W_1^{\neg A \vee B}, w_1)$ and $(w_2^{A \to B}, w_1)$ $\sigma_C^{\mathscr{B}}$-unify, thus closing the tree.

Remark 39 It is worth noting that Theorem 36 also shows the completeness of KE^+ for classical propositional logic. This is enough for the tautology test required by Definition 2. It is not necessary to extend it to the whole of **C**, since the label formulas are always classical. The same holds for the equivalence test and Corollary 38.

6 Discussion and Further Extensions

In this paper we have provided a unification scheme for computing the consequence relation of **C**. Since **C** corresponds to the flat fragment of the conditional logic **CU** we have confined ourselves to the case of the unification with labels of length 2. However, it is clear that such a scheme can be extended to **CU** with nested conditionals. Generally speaking, this accords with what is presented by Gabbay in some recent works (see, e.g., [19]). In particular, he has shown that the recursive self-fibring of a nonmonotonic consequence relation corresponds to a particular CL. As a second step, Gabbay and Governatori [22] have shown how to adapt the *KEM* label formalism in order to deal with combined (fibred) modal logics. Moreover they have proved that if *KEM* is sound and complete for each component then *KEM* is sound and complete for the logic resulting from the fibred combination of the components. Thus, by the same techniques we can provide sound and complete labelled tableaux calculi for CLs corresponding to (fibred) nonmonotonic consequence relations.

Let us show very briefly how to do it. First of all, for any label $i, \ell(i) > n$, we call each $b(i), b(b(i)), \ldots$ a *segment* of i and denote it by $s(i)$. Since the length of a label i is the number of the world symbols it is made of, we use $s^n(i)$ to denote the segment of i whose length is n. Secondly, we have to define the *countersegment-n* of i, as follows:

$$c^n(i) = h(i)^X \times (\cdots \times (h^k(i)^Y \times (\cdots \times (h^{n+1}(i)^Z, w_0)))) \quad (n < k < \ell(i))$$

where w_0 is an auxiliary label.[7] In other words the countersegment-n of a label i is the label obtained from i by replacing $s^n(i)$ with an auxiliary world symbol. At this point, we are able to define the unification for **CU** as follows:

Definition 40 (Cf. [21, 22]) For all $i^Y, j^X \in \Im$,

$$(i^Y, j^X)\sigma_{\mathbf{CU}}^{\mathscr{B}} = \begin{cases} (i^Y, j^X)\sigma_{\mathbf{C}}^{\mathscr{B}} \\ (c^n(i^Y), c^n(j^X))\sigma_{\mathbf{C}}^{\mathscr{B}} \end{cases}$$

where $w_0 = (s^n(i^Y), s^n(j^X))\sigma_{\mathbf{CU}}^{\mathscr{B}}$.

Informally, definition 40 can be explained as follows. In general, each unification scheme has several turning points, i.e., pairs of segments that should be matched (unified). In order to obtain $\sigma_{\mathbf{CU}}^{\mathscr{B}}$ we impose the same constraints for **C** on the formulas indexing such turning points. In other words, it is enough to apply recursively the unification for **C** throughout the paths represented by the labels. Notice that the process is reduced to a "step-by-step"

[7] An auxiliary label is nothing else than a dummy label, i.e., a label not appearing in i. The context in which such a notion is applied will tell us what it stands for. For further details, see [1, 21, 22].

matching of atomic labels since the "underlying" modality of **CU** corresponds to the system **K**.

Unfortunately, this is not enough to define a suitable proof system for CLs. It is quite clear that the problems examined in section 5.1 arise again for label formulas with nested conditionals. However, KE^+ obviously does not cover this case. Let us consider the following proof:

$$\begin{array}{lr}
1. F(A>A)>C \rightarrow (B\rightarrow B)>C & w_1 \\
2. T(A>A)>C & w_1 \\
3. F(B\rightarrow B)>C & w_1 \\
4. TC & (W_1^{A>A}, w_1) \\
5. FC & (w_2^{B\rightarrow B}, w_1) \\
6. \times & (w_2^{A>A}, w_1)
\end{array}$$

Obviously, we know that the proof tree should close since the label formulas $A>A$ and $B\rightarrow B$ correspond to $\top$. The question is: how to check this fact without refuting them? For example, by applying $T\mid\!\sim$ to the step 4 we obtain:

$$TA : (W_2^A, (W_1^{A>A}, w_1))$$

We can note that the structure of the label closely mimics the structure of the tree generated by the corresponding label formula: by refuting $A>A$, the proof tree closes because we have that A is false in a A-world, thus obtaining a contradiction. However, this is only a trivial case. In general we have to deal with formulas with different structures.

To sum up, what we have to do is to devise a tool similar to KE^+ for the language with nested conditionals.

A second point to be considered as a matter of future work is the extension of the system to other relevant notions of nonmonotonic consequence relation. We briefly suggest how to treat some of them. In particular, let us recall at least three families of nonmonotonic consequence relations which can be potentially covered by our proof system (see [29, 28]).

The first family is obtained by adding to **C** the following rule:

$$\frac{A_1 \mid\!\sim A_2, \ldots, A_{k-1} \mid\!\sim A_k, A_k \mid\!\sim A_1}{A_1 \mid\!\sim A_k} \qquad \text{(Loop)}$$

It is easy to see that Loop is characterized in an *SFC* model by imposing this supplementary condition:

> For all $1 \leq i, j \leq k$, if $f(A_i, u) \subseteq \|A_{i+1}\|$ and $f(A_k, u) \subseteq \|A_1\|$, then $f(A_i, u) = f(A_j, u)$.

Accordingly, we should be able to establish whether a sequence of labels in the branch of a proof tree denotes a set of equivalent worlds. This can be achieved by the following definition.

Definition 41 A label formula B is said to be *A-supported* if 1) $B, (W^A, w_1)$ is in the branch, or 2) there is a formula C such that

(i) B is C-supported; and

(ii) C is A-supported.

Then two labels i^Y and j^X unify if Y is X-supported and X is Y-supported.

As it is well-known, the system **P**, corresponding to the family of preferential relations, consists of all the rules of **C** and the following:

$$\frac{A \mid\!\sim C \quad B \mid\!\sim C}{A \vee B \mid\!\sim C} \tag{Or}$$

The semantic condition for Or is the following:

$$f(A \vee B, u) \subseteq f(A, u) \cup f(B, u).$$

The case for **P** is slightly more complicated, due to the interplay between **C** and the semantic condition for Or. Basically, we have to define new rules for manipulating connectives in the label formulas. This can be achieved by introducing new clauses which allow the composition of different label formulas via Or and capture the formal properties of the relation $\leq$ [29].

The third family of consequence relations to be considered corresponds to the system **R** and consists of **P** and the following rule:

$$\frac{A \mid\!\sim B \quad A \not\mid\!\sim \neg C}{A \wedge C \mid\!\sim B} \tag{Rational Monotonicity}$$

The semantic condition for *Rational Monotonicity* can be formulated as follows:

If $f(A, u) \cap \|B\| \neq \emptyset$ then $f(A \wedge B, u) \subseteq f(A, u)$

Rational Monotonicity says that the consequence relation contains $A \wedge C \mid\!\sim B$ whenever it contains $A \mid\!\sim B$ and does not contain $A \mid\!\sim \neg C$. Since KE^+ allows us to verify when $\neg C$ is v-fulfilled, the general conditions for Rational Monotonicity should require that the branch of a proof tree contains a set of formulas Z such that Z v-fulfils $\neg C$ and $Z : (w_n^A, w_1)$.

As we said before, the unification σ_C is defined from the unification for the modal logic **K**. If we replace it with the unification for the modal logic **D** (namely we release the constraint that two variables cannot unify) we characterize the property of consistency: if $A \not\equiv \bot$ and $A \mid\!\sim B$, then $A \not\mid\!\sim \neg B$, whose semantic condition is: if $A \not\equiv \bot$ then for all u, $f(A, u) \neq \emptyset$.

In conclusion, the system we have presented in this paper offers (at least potentially) a uniform proof-theoretical method for treating a wide range of nonmonotonic consequence relations as well as CLs. This can be seen as an advantage over the theorem proving methods for CLs to be discussed in the following section, as it is not immediate how they could be extended to logics other than those they have been devised for.

7 Comparison with Other Works

Groeneboer and Delgrande [26] present a method for constructing Kripke models for CLs which generalizes Hughes and Cresswell's [27] classical method of semantic tableau diagrams for the modal logic **S4.3** to Delgrande's [15] conditional logic **N**. This extension is made possible by the correspondence between **S4.3** and **N**. However, as Boutilier [8] has shown, **N** fails to validate the rule of Cautious Monotonicity, and thus it lies outside the scope of Gabbay's [18] minimal conditions for nonmonotonic consequence relations. Lamarre [30] takes a more direct approach by relying on Lewis' [32] system of spheres models. However, his method does not cover **CU**. Moreover, as proof systems for CL, the systems just mentioned suffer all the well-known computational drawbacks of the tableau method.

Although their primary aim is not automated deduction, Crocco and Fariñas del Cerro [11] present a sequent system for **CU** which turns out to be very similar to ours. In their approach the cut rule is replaced by more restricted rules for identifying formulas in deduction. Deductive contexts and restrictions on the transitivity of the deduction relation are represented at the level of auxiliary sequents, i.e., sequents involving a non-transitive deduction relation. Accordingly, structural and logical operations are performed both on this level and on the level of the principal (transitive) relation. The deductive context is fixed by a prefixing rule in the antecedents of auxiliary sequents. Augmentation and reduction rules in such antecedents allow us to identify those deductive contexts which are identical or compatible with other contexts, thus providing criteria for substituting conditional antecedents by conditional antecedents. In the present approach conditional antecedents are fixed by the inference rules at the "auxiliary" level of label formulas, whereas the notion of compatible contexts—or of criteria for antecedent identification—is captured by the label unification rule. Structural and logical operations are performed both at the "principal" level of labelled formulas and at the "auxiliary" level of label formulas, the only deduction relation involved being the transitive one. Thus our approach can be said to perform what Crocco and Fariñas del Cerro call an "extra-logical" control on the composition of proofs in the sense that the restrictions on the transitivity of the deduction relation are represented at the "auxiliary" level of our labelling scheme. This can be seen as an advantage of our method over Crocco and Fariñas del Cerro's as it allows us to treat a wide range of CLs by providing different constraints, closely related to the appropriate semantic conditions, on the respective unifications (see [2]). Moreover, this is achieved without banishing the cut rule, thus avoiding the problems arising from defining connectives in the absence of such a rule.

Acknowledgments

This paper is an extended and revised version of [3] presented at TABLEAUX 2000, International Workshop on Analytic Tableaux and Related Methods, St. Andrews, July 3–7, 2000.

We would like to thank all the anonymous referees for their valuable criticisms, Marcello D'Agostino for the fruitful discussions, and David Billington for his accurate proof-reading of the final draft of the paper.

References

[1] Alberto Artosi, Paola Benassi, Guido Governatori, and Antonino Rotolo. Shakespearian modal logic: A Labelled Treatment of Modal Identity. In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyaschev, editors, *Advances in Modal Logic. Volume 1*, pages 1–20. CSLI Publications, Stanford, 1998.

[2] Alberto Artosi and Guido Governatori. A tableaux methodology for deontic conditional logics. In *Deon'98. 4^{th} International Workshop on Deontic Logic in Computer Science*, pages 75–91, Bologna, 1998. CIRFID. `http://arXiv.org/abs/cs.LO/0003050`.

[3] Alberto Artosi, Guido Governatori, and Antonino Rotolo. A labelled tableau calculus for nonmonotonic (cumulative) consequence relations. In R. Dyckhoff, editor, *Automated Reasoning with Analytic Tableaux and Related Methods*, pages 82–97, Berlin, 2000. Springer-Verlag.

[4] Alberto Artosi, Guido Governatori, and Giovanni Sartor. Towards a Computational Treatment of Deontic Defeasibility. In M.A. Brown and J. Carmo, editors, *Deontic Logic, Agency and Normative Systems*, pages 27–46. Springer-Verlag, Berlin, 1996.

[5] Matteo Baldoni. Normal multimodal logic with interaction axioms. In D. Basin, M. D'Agostino, D. Gabbay, S. Matthews, and L. Viganò, editors, *Labelled Deduction*, pages 33–58. Kluwer, Dordrecht, 2000.

[6] Bernhard Beckert and Rajeev Goré. Free-variable tableaux for propositional modal logic. *Studia Logica*, forthcoming.

[7] David Billington. Defeasible logic is stable. *Journal of Logic and Computation*, 3:370–400, 1993.

[8] Craig Boutilier. Conditional logics of normality: a modal approach. *Artificial Intelligence*, 68:87–154, 1994.

[9] John P. Burgess. Quick completeness proofs for some logic of conditionals. *Notre Dame Journal of Formal Logic*, 22:76–84, 1981.

[10] Brian Chellas. Basic conditional logic. *Journal of Philosophical Logic*, 4:133–153, 1975.

[11] Gabriella Crocco and Luis Fariñas del Cerro. Structure, consequence relation and logic. In D. Gabbay, editor, *What is a Logical System*, pages 375–393. Oxford University Press, Oxford, 1994.

[12] Gabriella Crocco and Philippe Lamarre. On the connections between non-monotonic inference systems and conditional logics. In B. Nebel, C. Rich, and W. Swartout, editors, *Principles of Knowledge Representation and Reasoning (KR'92)*, pages 565–571, San Mateo (Ca), 1992. Morgan Kaufman Publishers.

[13] Marcello D'Agostino and Marco Mondadori. The taming of the cut. *Journal of Logic and Computation*, 4:285–319, 1994.

[14] Herrie de Swart. A Gentzen- or Beth-type system, a practical decision procedure and a constructive completeness proof for the counterfactuals logics *vc* and *vcs*. *Journal of Symbolic Logic*, 48:1–20, 1983.

[15] James P. Delgrande. A first-order conditional logic for prototypical properties. *Artificial Intelligence*, 33:105–139, 1987.

[16] Luis Fariñas del Cerro, Andreas Herzig, and Jérôme Lange. From ordering-based nonmonotonic reasoning to conditional logics. *Artificial Intelligence*, 66:375–393, 1994.

[17] Melvin Fitting. *Proof Methods for Modal and Intuitionistic Logic*. Reidel, Dordrecht, 1983.

[18] Dov M. Gabbay. Theoretical foundations for nonmonotonic reasoning in expert systems. In K.R. Apt, editor, *Proc of the NATO Advanced Study Institute on Logics and Concurrent Systems*, pages 439–457, Berlin, 1985. Springer-Verlag.

[19] Dov M. Gabbay. Conditional implications and non-monotonic consequence. In G. Crocco, L. Fariñas del Cerro, and A. Herzig, editors, *Conditionals: from philosophy to computer science*, pages 337–360. Oxford University Press, Oxford, 1995.

[20] Dov M. Gabbay. *Labelled Deductive Systems*. Oxford University Press, Oxford, 1996.

[21] Dov M. Gabbay and Guido Governatori. Dealing with label dependent deontic modalities. In P. McNamara and H. Prakken, editors, *Norms, Logics and Information Systems*, pages 311–330. IOS Press, Amsterdam, 1998.

[22] Dov M. Gabbay and Guido Governatori. Fibred modal tableaux. In D. Basin, M. D'Agostino, D. Gabbay, S. Matthews, and L. Viganò, editors, *Labelled Deduction*, pages 163–194. Kluwer, Dordrecht, 2000.

[23] Ian Gent. A tableau- or sequent-style system for Lewis's counterfactual logic VC. *Notre Dame Journal of Formal Logic*, 33, 1992.

[24] Guido Governatori. KE$^+$: Beyond Refutation In L. Dreschler-Fischer and S. Pribbenow, editors, *KI-95 Activities: Workshops, Posters, Demos*, pages 75–76, Bonn, 1995. Gesellschaft für Informatik.

[25] Guido Governatori. *Un modello formale per il ragionamento giuridico*. PhD thesis, CIRFID, University of Bologna, Bologna, 1997.

[26] Chris Groeneboer and James P. Delgrande. Tableau-based theorem proving in normal conditional logics. In *AAAI'88*, volume i, pages 171–176, 1988.

[27] George E. Hughes and Max J. Cresswell. *An Introduction to Modal Logic*. Methuen, London, 1968.

[28] Hirofumi Katsuno and Ken Satoh. A unified view of consequence relation, belief revision, and conditional logic. In G. Crocco, L. Fariñas del Cerro, and A. Herzig, editors, *Conditionals: From Philosophy to Computer Science*, pages 33–66. Oxford University Press, Oxford, 1995.

[29] Sarit Kraus, Daniel Lehmann, and Menachem Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44:167–207, 1990.

[30] Philippe Lamarre. A promenade from monotonicity to nonmonotonocity following a theorem prover. In B. Nebel, C. Rich, and W. Swartout, editors, *Principles of Knowledge Representation and Reasoning (KR'92)*, pages 572–580, San Mateo (Ca), 1992. Morgan Kaufman Publishers.

[31] Daniel Lehmann. What does a conditional base entail? In R.J. Brachman, H.J. Levesque, and R.Reiter, editors, *Proceedings of Knowledge Representation and Reasoning, (KR'89)*, pages 212–222, San Mateo (Ca), 1989. Morgan Kaufman Publishers.

[32] David Lewis. *Counterfactuals*. Basil Blackwell, Oxford, 1986.

[33] Fabio Massacci. Strongly analytic tableaux for normal modal logics. In A. Bundy, editor, *Proceedings, 12th International Conference on Automated Deduction (CADE)*, volume 814 of *LNCS*, pages 723–737, Berlin, 1994. Springer-Verlag.

[34] Donald Nute. *Topics in Conditional Logic*. Reidel, Dordrecht, 1980.

[35] Claudio Pizzi. Decision procedures for logics of consequential implication. *Notre Dame Journal of Formal Logic*, 32:618–636, 1993. And 1993 correction (same journal).

[36] Alessandra Russo. *Modal Logic as Labelled Deductive Systems*. PhD thesis, Department of Computing, Imperial College, London, 1996.

[37] Yoav Shoham. A semantical approach to nonmonotonic logics. In J. McDermott, editor, *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 388–392, Los Altos (Ca), 1987. Morgan Kaufman Publishers.

[38] Raymond M. Smullyan. *First-Order Logic*. Springer-Verlag, Berlin, 1968.

[39] Richmond H. Thomason. A Fitch-style formulation of conditional logic. *Logique et Analyse*, 52:397–412, 1970.

[40] Luca Viganò. *Labelled Non-Classical Logics*. Kluwer, Boston, 2000.