A Labelled Tableau Calculus for Nonmonotonic (Cumulative) Consequence Relations[*]

Alberto Artosi[1] Guido Governatori[2] Antonino Rotolo[3]

[1] Department of Philosophy, University of Bologna,
via Zamboni 38, I-40126 Bologna, Italy. E-mail: artosi@cirfid.unibo.it
[2] School of Computing and Information Technology, Griffith University,
Nathan, QLD 4111, Australia. E-mail: guido@cit.gu.edu.au
[3] CIRSFID, University of Bologna,
Via Galliera 3, I-40121 Bologna, Italy. E-mail: 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 is based on the usual possible world semantics for conditional logic. 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 inference rules are provided by the propositional system KE^{+} —a tableau-like analytic proof system devised to be used both as a refutation and a direct method of proof— 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

Recently, a number of proposals have been put forward to find a unifying approach to a plethora of different nonmonotonic formalisms, and even to unify such seemingly distant areas as conditional logic, nonmonotonic inference, belief revision and update. We refer, in particular, to Shoham's [20] general semantic framework for nonmonotonic logics, Kraus, Lehman and Magidor's [16] approach to nonmonotonic consequence relations, and Katsuno and Satoh's [15] "unified" semantic view. All these approaches are based on a preference (ordering) semantics and exploit the strong semantic connections between nonmonotonic inference and conditional logic. In this paper we shall take a different view of what a suitable "unifying" framework looks like. In our view such a framework must pay greater attention to the computational aspects and to proof-theoretical formulations. This view finds strong justification both in the aim of comparing and combining different logics, such as the logic of nonmonotonic inference and conditional logic, and in the potential applications of nonmonotonic inference in the AI field. Accordingly, our purpose in this paper will be to provide a methodology for the proof theoretical treatment of nonmonotonic inference and conditional logic (henceforth CL). We shall outline the fundamentals of a tableau proof system construction

---

[*] Due to space limitations Theorems are provided without proofs. The full version of the paper is available at `http://www.cit.gu.edu.au/~guido/papers/tab2000.pdf`.

R. Dyckhoff (eds)
Automated Reasoning with Analytic Tableaux and Related Methods.
LNCS 1847, pp. 82–97, 2000.

aimed to compute nonmonotonic consequence relations in a (monotonic) CL whose "flat" (i.e., unnested) fragment is shown to correspond to Kraus, Lehmann and Magidor's [16] basic system C for cumulative relations. We shall give this construction a special presentation as an algorithmic proof system which uses a labelling discipline, in the wake of Gabbay's [11] *Labelled Deductive Systems* (LDS), to generate and check models. This closely reflects Gabbay's view of what a unifying framework for presenting and comparing logics comprises.

A detailed discussion of the merits of LDS as a unifying framework is beyond the scope of this paper. However, a key feature of LDS is worth mentioning. LDS 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. [12,3]). As is well-known, several attempts to establish close semantic connections between nonmonotonic consequence relations and (monotonic) modal and conditional logics, notably by Boutilier [4] and Katsuno and Satoh [15], rely on Kripke structures very close to Kraus, Lehmann and Magidor's "preferential" models. In particular, Boutilier [4] has shown on this basis that Kraus, Lehmann and Magidor's [16,18] stronger consequence relation systems $\mathbf{P}$ and $\mathbf{R}$ and Degrande's [7] logic N closely correspond to the flat parts of modal CLs definitionally equivalent to the standard modal systems $\mathbf{S4}$ and $\mathbf{S4.3}$. In LDS the usual modal semantics is incorporated in the syntactic label construction and only minor variations are needed to pass from a logic to another [1,3,12]. So, once an automated LDS for $\mathbf{S4}$ and $\mathbf{S4.3}$ is available, a wide range of logics admit computational treatment. However, if we wish automated LDS for CLs on their own, or we are interested in a less restricted fragment of the conditional language, only slight natural changes in the modal LDS are needed to yield the appropriate semantics: in the LDS to be presented in this paper only a simple indexing of labels with formulas.

The approach we propose in this paper can be motivated also from another perspective. Fariñas del Cerro *et al.* [9] have recently emphasized the need for a more computational treatment of nonmonotonic inference. The method they propose for this task consists of reducing computation to a validity test in a (monotonic) CL. This is viewed as a first step towards the "effective computation . . . of nonmonotonic inference relations via automated deduction method" in conditional logic. Unfortunately, CL is not particularly well suited for this task. Indeed, its inferential structure has not been sufficiently explored to provide reliable automated deduction methods for effectively computing the inferences sanctioned by nonmonotonic inference relations. We know only two attempts in this direction: Groeneboer and Delgrande's [13] and Lamarre's [17] 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 nonmonotonic consequence relations can be effectively computed by a countermodel validity test for the corresponding class of conditional formulas.

We shall proceed in the following way. First we briefly rehearse Kraus, Lehmann and Magidor's [16] sequent system $\mathbf{C}$ for cumulative relations. Then we introduce Lewis-type semantic structures akin to the kind of models used to characterize $\mathbf{C}$. Such structures will allow us to establish a correspondence between $\mathbf{C}$ and the flat fragment of a suitable extension $\mathbf{CU}$ of Chellas' [5] basic normal system $\mathbf{CK}$. 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 $\mathbf{CU}$. Then suitable tableau inference and label propagation rules are introduced which provide a sound and complete proof system for the flat fragment of $\mathbf{CU}$. These rules are implemented on a classical propositional system designed to be used both as a refutation and a direct method of proof. Finally, we provide some remarks on computational issues and related works.

2 Nonmonotonic Consequence Relations and Conditional Logic

The study of nonmonotonic consequence relations has been undertaken by Gabbay [10] who proposed three minimal conditions a (binary) consequence relation $\mid\sim$ on a language L should satisfy to represent a nonmonotonic logic. More recently, Kraus, Lehmann and Magidor [16] 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 $\mathbf{C}$ to define the (weakest) class of cumulative consequence relations, that closely corresponds to that satisfying Gabbay's minimal conditions (we assume that $\vdash$ and $\mid\sim$ are defined on the language L of classical propositional logic).

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

$$\frac{\vdash B \rightarrow C \quad A \mid\sim B}{A \mid\sim C}\text{Right Weakening} \qquad \frac{\vdash A \equiv B \quad A \mid\sim C}{B \mid\sim C}\text{Left Logical Equivalence}$$

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

Notice that the following

$$\frac{A \mid\sim B \quad A \mid\sim C}{A \mid\sim B \wedge C}\text{ And} \qquad \frac{A \mid\sim B \quad B \mid\sim A}{A \mid\sim C \iff B \mid\sim C}\text{ CSO}$$

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 $\mathbf{C}$.

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 [16] to characterize $\mathbf{C}$.

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 $M = \langle W, f, v\rangle$ 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. *v is a valuation assigning to each u in W and $A \in L_>$ an element from the set $\{T, F\}$.*

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 B-worlds, i.e., $\|B\| = \{w \in W : v(B, w) = T\}$. A formula A is valid ($\models_{SF}$) just when $M \models_u A$ for all worlds in all SF models.

Definition 2. *A* selection function cumulative model *(SFC) is an SF model $M = \langle W, f, v\rangle$ 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 $f(A \wedge B, u) \subseteq f(A, u)$ (Cut)*
4. *If $f(A, u) \subseteq \|B\|$, then $f(A, u) \subseteq f(A \wedge B, u)$ (Cautious Monotonicity).*

Notice that from 3 and 4 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 $\mathbf{CU}$— containing classical propositional logic and the following axioms

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

and closed under the usual inference rules $RCEA$ and RCK. Notice that ID, RT, CM, $RCEA$, and RCK correspond, respectively, to Reflexivity, Cut, Cautious Monotonicity, Left Logical Equivalence and Right Weakening. Of course, $\mathbf{CU}$ is nothing but Chellas' [5] conditional logic $\mathbf{CK} + ID$ augmented with RT and CM. A standard Henkin-style construction proves the completeness of $\mathbf{CU}$ with respect to the class of SFC models.

Theorem 1. *$\models_{SFC} A$ iff $\vdash_{\mathbf{CU}} A$.*

Whether $\mathbf{CU}$ is interesting in its own rights is an issue which need not detain us here. What matters is that we can establish a mapping between C and $\mathbf{CU}$ similar to the well-established correspondences between [16]'s stronger systems $\mathbf{P}$ and $\mathbf{R}$ of preferential and rational relations and the flat fragments of well-known conditional logics.

Let $\mid\!\sim_{\mathbf{S}}$ denote the consequence relation $\mathbf{S}$ and let $\mathbf{K}^-$ denote the conditional logic $\mathbf{K}$ restricted to the formulas of the form $A > B$ where $A, B \in L$. A consequence relation $\mid\!\sim$ is *defined by* an SF model M if the following condition is satisfied: $A \mid\!\sim B$ iff $M \models A > B$. A consequence relation system $\mathbf{S}$ is said to *correspond to* a conditional logic K if the following condition is satisfied: $A \mid\!\sim_{\mathbf{L}} B$ iff $\vdash_{\mathbf{K}^-} A > B$.

Theorem 2. *The consequence relation system* $\mathbf{C}$ *corresponds to the conditional logic* $\mathbf{CU}$.

The theorem follows from showing that the axioms and rules of $\mathbf{CU}$ are straightforward translations of the rules of $\mathbf{C}$ and that $A \mid\!\sim_{\mathbf{C}} B$ 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 follow immediately from Theorem 2 (see Corollaries 3.26, 3.27 and 3.28 of [16] for comparison).

Corollary 1. *Let* Γ *be a 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* $\mathbf{CU}$.
2. $A > B$ *is satisfied by all* SCF *models which satisfy* Γ'.

Corollary 2. *A 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 $\mathbf{C}$ 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

KEM [1] is an algorithmic modal proof system which, in the spirit of Gabbay's [11] LDS, brings semantics into proof theory using (syntactic) labels in order to simulate models in the proof language. In this section we show how it can be extended, with little modification, to handle $\mathbf{C}$. In what follows $\mathcal{L}$ 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

The format of the labels has been designed to cover general possible world semantics. 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.

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 3. $\Im = \bigcup_{1 \leq p} \Im_p$ *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. We define the length of a label i, $l(i)$, to be the number of atomic labels in i. From now on we shall use $i, j, k, \ldots$ to denote arbitrary labels. 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*. Let us stipulate that if $i \in \Phi_C \cup \Phi_V$ and $Y \in \mathcal{L}$ then $i^Y \in \Im_1$. We shall call a label i^Y a *formula-indexed label*, and Y the *label formula* of i. 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 structured label, we mean the label formula of the head of the label.

The notion of a formula-indexed label is meant to capture the intended semantics. An atomic label indexed with a formula Y (such as, e.g., w_1^A or W_1^A) is meant to represent either a Y-world or a set of Y-worlds (equivalently, any Y-world) in some SF model. A label of the form $({k'}^Y, k)$ is "structurally" designed to convey information about the worlds in it. For example, (W_1^A, w_1) can be viewed as representing (any world in) the set of those A-worlds that are minimal with respect to w_1 under some ordering relation $\prec$. The label (w_1^A, w_1) represents an A-world in such a set. In this perspective a *labelled signed formula* (*LS-formula*) [1] is a pair X, i where X is a signed formula (i.e., a formula of $\mathcal{L}$ prefixed with a "T" or "F") and i is a label. Intuitively, an LS-formula, e.g. $TC, (W_1^{A \vee B}, w_1)$, means that C is true at all the (minimal) $A \vee B$-worlds.

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 say that 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 $S \in \{T, F\}$, $A, Y \in \mathcal{L}$, X is a signed formula, and $i \in \Im$. That means that either SA is labelled with i, or 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 $B \wedge C$, since these are the formulas occurring in X, i^Y.

In what follows we assume familiarity with Smullyan [21] uniform notation for signed formulas.

3.2 Label Unifications

The key feature of KEM approach is that in the course of proof search labels are manipulated in a way closely related to the semantics of modal operators and "matched" using a specialized unification algorithm. That two labels i and k are unifiable means, intuitively, that the set of worlds they "denote" have a non-null intersection. In this section we define a special notion of unification for $\mathbf{C}$ ($\sigma_{\mathbf{C}}$-unification) which is meant to "simulate" the conditions on 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

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

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 definition of the unification for indexed labels is more complicated since we have to take into account 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., [6]). In other words, they avoid to substitute an antecedent by another antecedent by transitivity (via cut). The following definitions establish the basic (restricted) conditions for such a substitution. In particular, they say when two formulas are equivalent with respect to $\mid\!\sim$ ($\mid\!\sim$-equivalent).

Definition 4.

- *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* $\{A_1, \ldots, A_n\}$ c-fulfils A*, and* $\{A_{1_1}, \ldots, A_{1_m}\}, \ldots, \{A_{n_1}, \ldots, A_{n_m}\}$ c-fulfil *respectively* $A_1, \ldots, A_n$*, then* $\{A_{1_1}, \ldots, A_{1_m}, \ldots, A_{n_1}, \ldots, A_{n_m}\}$ 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.

Definition 5. *We say that A* forces *B, iff 1) either $A = B$ or A is of type α and $B = \alpha_i$; or 2) there exists a formula C such that A forces C and C forces B.*

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

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

Definition 6. A supports B *in a branch* $\mathcal{B}$ *iff*

1. $A \equiv B$*; or*
2. $\{B_1, \dots, B_n\}$ *c-fulfils* B*, and for each* k, $1 \leq k \leq n$, $B_k, (W^{A}_{i_k}, w_1) \in \mathcal{B}$*; or*
3. *there is a set of formulas* $\mathcal{A} = \{Z_1, \dots, Z_n\}$ *such that,* $1 \leq i \leq n$, $Z_i, (W^{A}_{i}, w_1) \in \mathcal{B}$, A *forces* Z_i*, and* $\mathcal{A}$ *c-fulfils* B*.*

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

Definition 7. $A \approx_{\mathcal{B}} B$ *iff*

1. A *and* B *support each other; or*
2. *if* $A \approx_{\mathcal{B}} C$ *and* $B \approx_{\mathcal{B}} C$*, then* $A \approx_{\mathcal{B}} B$

If $A \in \mathcal{B}$, with $A_{\approx_{\mathcal{B}}}$ we shall denote the set of formulas $\{B_1, \dots, B_n\}$ such that, $1 \leq i \leq n$, $B_i \in \mathcal{B}$ and $B_i \approx_{\mathcal{B}} A$. It is obvious that $A_{\approx_{\mathcal{B}}}$ is an equivalence class, thus we abuse the notation and we use $A_{\approx_{\mathcal{B}}}$ 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 6), 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 7, A and B are equivalent with respect to $\mid\!\sim$ in $\mathcal{B}$ if (a) the above sets are equal, or (b) such sets are equal to another set. This means that they prove each other.

At this point we are ready to introduce the notion of unification for indexed labels to be used in the calculus. Briefly, two labels unify wrt 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 8. *Let* i^X *and* j^Y *be two indexed labels, and let* $\mathcal{B}$ *be a set of* LS*-formulas. Then*

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

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

a) $Y \approx_{\mathcal{B}} X$*;*
b) $Y \equiv \top$ *and* $h(i) \in \Phi_V$*, or* $X \equiv \top$ *and* $h(j) \in \Phi_V$*;*
c) *i)* X *is of type* α, $Y \approx_{\mathcal{B}} \alpha_i$ *for* $i = 1, 2$*, and* $Z, (W^{Y_{\approx_{\mathcal{B}}}}, w_1) \in \mathcal{B}$ *such that* Z*'s c-fulfil* α_{3-i}*, or*
  ii) Y *is of type* α, $X \approx_{\mathcal{B}} \alpha_i$ *for* $i = 1, 2$*, and* $Z, (W^{X_{\approx_{\mathcal{B}}}}, w_1) \in \mathcal{B}$ *such that* Z*'s c-fulfil* α_{3-i}*.*

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" or the label is empty. Indeed $||\bot|| = \emptyset$, and, by reflexivity, for each $A \in L_{>}$ and $u \in W$, $f(A, u) \subseteq ||A||$, then $f(\bot, u) = \emptyset$.

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

According to b), $(W_2^{A\to A}, w_1)$ and (w_3^C, w_1) unify, as W_2 is a variable indexed with a tautology. In practice a unrestricted label indexed with a tautology unifies with any restricted label.

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-fulfillment. Such a notion is nothing else that the condition a set of formulas must satisfy to (propositionally) entail the formula which is "fulfilled". Notice that the notion of c-fulfillment 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: $\mathcal{A}_1 = TB, (W_2^A, w_1)$, $\mathcal{A}_2 = TD, (W_3^A, w_1)$. Here $(i,j)\sigma_{\mathbf{C}}^{\mathcal{B}}$, where $\mathcal{B}$ contains $\mathcal{A}_1$ and $\mathcal{A}_2$. 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 $\mathcal{B}$ 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 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 $(i,j)\sigma_{\mathbf{C}}^{\mathcal{B}}$ to denote both that i and j are $\sigma_{\mathbf{C}}^{\mathcal{B}}$-unifiable and the result of their $\sigma_{\mathbf{C}}^{\mathcal{B}}$-unification, and X^C to denote the conjugate of X (i.e., $X^C = FA$ (TA) if $X = TA$ (FA)).

$$\alpha\frac{\alpha : X, k^Y}{\alpha_i, k^Y}[i = 1, 2] \qquad \beta\frac{\begin{array}{c}\beta : X, k^Y\\ \beta^C_{3-i} : X', j^{Y'}\end{array}}{\beta_i, (k,j)\sigma_{\mathbf{C}}^{\mathcal{B}}}[i = 1, 2]$$

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.

$$T\mid\!\sim \frac{TA \mid\!\sim B : X, i^Y}{TB, (W_n^A, i^Y)}[W_n^A \text{ new}] \qquad F\mid\!\sim \frac{FA \mid\!\sim B : i^Y}{FB, (w_n^A, i^Y)}[w_n^A \text{ new}]$$

The rules $T\mid\!\sim$ and $F\mid\!\sim$ closely reflect the semantical evaluation rule 1. They are motivated by the general idea (see [5]) that $>$ can be regarded as a necessity operator on

A (i.e., $[A >]B$), from which 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); and whenever $A > B$ is false at u, there should be some A-world where B is false.

$$PB\,\frac{}{X, i \qquad X^C, i}[i \text{ unrestricted}] \qquad PNC\,\frac{\begin{array}{c}X : Y, i^{Y'}\\ X^C : Z, k^{Z'}\end{array}}{\times}[(i, k)\sigma_C^{\mathcal{B}}]$$

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 ("$\times$") to follow from two formulas which are contradictory "in the same world", represented by two $\sigma_{\mathbf{C}}^{\mathcal{B}}$-complementary LS-formulas, i.e., two LS-formulas $X, i^{Y'}$, $X^C, k^{Z'}$ whose labels are $\sigma_{\mathbf{C}}^{\mathcal{B}}$-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 it may occur a contradiction of the form $FA, (w_2^A, w_1)$, since this LS-formula states that there exists an A-world where A is false.

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 $\mid\!\sim$ 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**. Let us first define the following functions which map labels into elements of SF cumulative models.

Let g be a function from $\Im$ to $\wp(W)$ thus defined:

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

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

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

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

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

Lemma 2. *Let $\mathcal{B}$ be a set of LS-formulas and let i, j be labels in $\mathcal{B}$. If $m(SA, i)$, and $(i, j)\sigma^{\mathcal{B}}_{\mathbf{C}}$, then $m(SA, (i, j)\sigma^{\mathcal{B}}_{\mathbf{C}})$.*

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 the rule in a standard semantic setting, whence the following lemma.

Lemma 3. *If $\vdash_{KEM} A$, then $\models_{SFC} A$*

where, a If $\vdash_{\mathbf{CU}} A$ then $\vdash_{KEM} A$ s usual with tableau methods, $\vdash_{KEM} A$ means that the KEM-tree starting with FA, w_1 is closed.

Lemma 4. *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 $\mathbf{C}$ are derivable in KEM. D'Agostino and Mondadori [8] 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.

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

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

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

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

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

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

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

Finally we present side by side the KEM proofs of Cautious Monotonicity and Cut.

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

In both proofs the labels unify according to condition c) of Definition 8.

From Theorem 1, Lemmas 4 and 3 we obtain

Theorem 3. *$\vdash_{KEM} A$ iff $\models_{SFC} A$.*

and from Theorem 3 and Corollary 2

Corollary 3. *Let Γ be a set of conditional assertions. Γ cumulatively entails $A \mid\!\sim B$ iff $\vdash_{KEM} \bigwedge \Gamma \to (A \mid\!\sim B)$*

5 Proof Search with KE^+

The unification presented in section 3.2 compels us to check (label) formulas either for validity or for logical equivalence. 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, the main tree provides all the information we need so that we only have to make them 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 [8]'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 in 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 $\beta_i^C (i = 1, 2)$ 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 9. *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. us define

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 β_1^C is in the branch also β_2 is in the branch, or 2) if β_2^C is in the branch also β_1 is the branch. A β formula will be said *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.

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

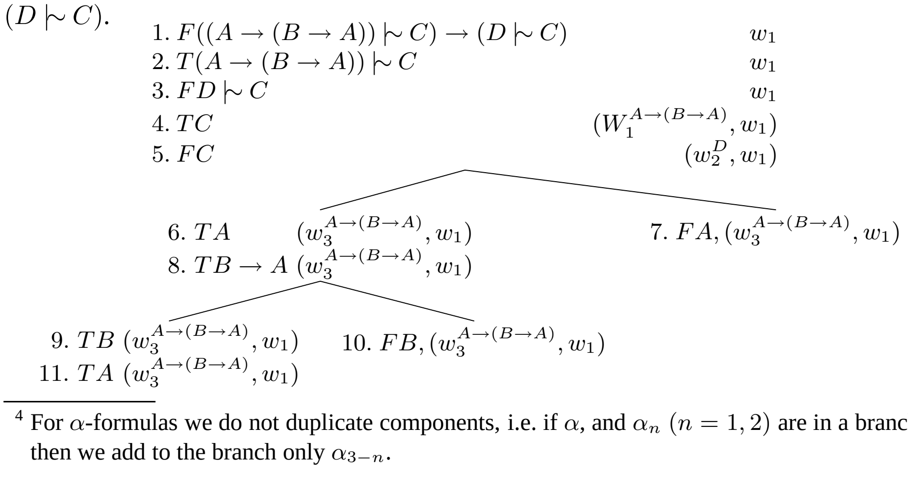
A branch obtained by applying PB to a β-formula with β_i^C as its root is a *β^C-branch*. Each branch generated by an application of PB to a formula occurring in a β^C-branch is a β^C-branch. A branch generated by an application of PB which is not a β^C-branch is a β-branch. A branch is a $\top$-branch 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 β^C-branch ϕ which is not yet completed and which is the β^C-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;[4] 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 β_1, β_1^C (or, equivalently β_2, β_2^C), and then it returns to step 1; otherwise it returns to step 1.

Theorem 4. *For a formula A, $A \equiv \top$ if either:*

1. *in one of the β^C-branches it generates there is an LS-formula which appears twice, and one occurrence depends on $\beta_i^C, i \in \{1, 2\}$, and the other depends on β, or*
2. *each β^C-branch is closed and the β-branches are $\top$-branches, or*
3. *each β^C-branch is a $\top$-branch.*

We provide an illustration of the above notion by proving $((A \to (B \to A)) \mid\!\sim C) \to (D \mid\!\sim C)$.



---

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

Here we have to see whether the labels in 4 and 5 unify. However the label formulas of W_1 is of type β and it is not yet analysed in the tree. So we apply PB. Notice that the left branch is a β^C-branch w.r.t. 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 label of 4 and 5 unify, thus closing the tree.

Definition 10. *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 $v(A) = v(\beta_n^C) \cup v(\beta_{3-n})$ or $v(A) = v(\beta_n)$, $n = 1, 2$. A set S of (atomic) formulas is said to* v-fulfils *a formula A iff $S = v(A)$.*

Corollary 4. *Two formulas A, B are equivalent iff each set of (atomic) formulas which v-fulfils A v-fulfils B.*

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

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

$$\begin{array}{ll}
5.\ TA\ (w_2^{A \to B}, w_1) & 6.\ FA\quad (w_2^{A \to B}, w_1) \\
7.\ TB\ (w_2^{A \to B}, w_1) &
\end{array}$$

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 $(w_2^{A \to B}, w_1)$ $\sigma_C^{\mathcal{B}}$-unify, thus closing the tree.

Remark 1. It is worth noting that Theorem 4 provides also 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 **C**, since the label formulas are always classical. The same holds for the equivalence test and Corollary 4.

6 Comparison with Other Works

Groeneboer and Delgrande [13] present a method for constructing Kripke models for CLs which generalizes Hughes and Cresswell's [14] classical method of semantic tableau diagrams for the modal logic $\mathbf{S4.3}$ to Delgrande's [7] conditional logic $\mathbf{N}$. This extension is made possible by the correspondence between $\mathbf{S4.3}$ and $\mathbf{N}$. However, as Boutilier [4] has shown, $\mathbf{N}$ fails to validate the rule of Cautious Monotonicity, and thus it lies outside the scope of Gabbay's [10] minimal conditions for nonmonotonic consequence relations. Lamarre [17] takes a more direct approach by relying on Lewis' [19] system of spheres models. However, his method does not cover $\mathbf{CU}$. Moreover, as proof systems for CL, the systems just mentioned can be said to suffer of all well-known computational drawbacks of the tableau method.

As far as the computational complexity of the method we have proposed is concerned, it lies in the same class as Lehmann's algorithm [18]. In fact it easy to prove that the complexity of σ is linear. However this is not the case with $\sigma^{\mathcal{B}}_{\mathbf{C}}$ in so far as it requires either tautology or equivalence or entailment tests on label formulas. Thus its absolute complexity is exponential. Nevertheless, when $\sigma^{\mathcal{B}}_{\mathbf{C}}$ is used in a KEM-proof its complexity weight does not cause any harm to the complexity of the proof since, as we have seen, the tests are performed in the proof itself. Therefore the complexity of $\sigma^{\mathcal{B}}_{\mathbf{C}}$ with respect to KEM-proofs turn out to be the same of σ. From this it follows that the complexity of the KEM-proofs for $\mathbf{C}$ is just the complexity of the propositional modulo (see [8] for a discussion). This is a well known result (see [18]), but we believe that the present approach offers some advantages over Lehmann's [18] algorithm in that it is deterministic and works for cumulative logics in general and not only for those cases where they coincide with the preferential ones.

Although their primary aim is not automated deduction, Crocco and Fariñas [6] present a sequent system for $\mathbf{CU}$ 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 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's as it allows 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, it do so without banishing the cut rule, thus avoiding the problems arising from defining connectives in the absence of such a rule.

Acknowledgments This work was partially supported by the Australia Research Council under Large Grant No. A49803544.

References

1. Alberto Artosi, Paola Benassi, Guido Governatori, Antonino Rotolo. Labelled Proofs for Quantified Modal Logic. In J.J. Alferes, L. M. Pereira, and E. Orlowska (eds.) *Logics in Artificial Intelligence*, 70–86, Springer-Verlag, Berlin, 1996.

2. Alberto Artosi and Guido Governatori. A Tableau Methodology for Deontic Conditional Logic. In *Deon'98. 4th International Workshop on Deontic Logic in Computer Science*, Bologna, January 8–10, 1998: 75–91. `http://arXiv.org/abs/cs.LO/0003050`.
3. Alberto Artosi, Guido Governatori, and Giovanni Sartor. Towards a Computational Treatment of Deontic Defeasibility. In M.A. Brown and J. Carmo (eds.), *Deontic Logic, Agency and Normative Systems*, Springer-Verlag, Berlin, 1996: 27–46.
4. Craig Boutilier. Conditional Logics of Normality: a Modal Approach. *Artificial Intelligence*, 68: 87–154, 1994.
5. Brian Chellas. Basic Conditional Logic. *Journal of Philosophical Logic*, 4: 133–153, 1975.
6. G. Crocco and L. Fariñas del Cerro. Structure, Consequence Relation and Logic. In D. Gabbay (ed.), *What is a Logical System*, Oxford UP, Oxford, 1994. 375–393, 1994.
7. James P. Delgrande. A First-Order Conditional Logic for Prototypical Properties. *Artificial Intelligence*, 33:105–139, 1987.
8. Marcello D'Agostino and Marco Mondadori. The Taming of the Cut. *Journal of Logic and Computation*, 4: 285–319, 1994.
9. L. Fariñas del Cerro, A. Herzig, and J. Lange. From Ordering-Based Nonmonotonic Reasoning to Conditional Logics. *Artificial Intelligence*, 66: 375–393, 1994.
10. Dov. M. Gabbay. Theoretical Foundations for Nonmonotonic Reasoning in Expert Systems. In K. R. Apt, ed., *Proc of the NATO Advanced Study Institute on Logics and Concurrent Systems*, 439–457, 1985, Berlin, Springer-Verlag, 1985.
11. Dov M. Gabbay. *Labelled Deductive Systems*. Oxford University Press, Oxford, 1996.
12. Dov M. Gabbay and Guido Governatori. Dealing with Label Dependent Deontic Modalities. In P. McNamara and H. Prakken (eds.), *Norms, Logics and Information Systems*, IOS Press, Amsterdam, 1998: 311–330.
13. Chris Groeneboer and James P. Delgrande. Tableau-Based Theorem Proving in Normal Conditional Logics. In *AAAI'88*, volume i, 171–176, 1988.
14. G.E. Hughes and M.J. Cresswell. *An Introduction to Modal Logic*. Methuen, London, 1968.
15. Hirofumi Katsuno and Ken Satoh. A Unified View of Consequence Relation, Belief Revision, and Conditional Logic. In G. Crocco, L. Fariñas del Cerro, A. Herzig (eds.) *Conditionals: From Philosophy to Computer Science*, Oxford University Press, 33–66, 1995.
16. Sarit Kraus, Daniel Lehmann, and Menachem Magidor. Nonmonotonic Reasoning, Preferential Models and Cumulative Logics. *Artificial Intelligence*, 44: 167–207, 1990.
17. Philippe Lamarre. A promenade from monotonicity to non-monotonocity following a theorem prover. In *Principles of Knowledge Representation and Reasoning (KR'92)*, 572–580, San Mateo (Ca), 1992, Morgan Kaufman Publishers.
18. Daniel Lehmann. What Does a Conditional Base Entail? In R. J. Brachman, H. J. Levesque, and R. Reiter (eds), *Proceedings of Knowledge Representation and Reasoning, (KR'89)*, Morgan Kaufmann Publishers, S. Mateo (Ca), 1989: 212–222.
19. David Lewis. *Counterfactuals*. Basil Blackwell, Oxford, 1986.
20. Yoav Shoham. A Semantical Approach to Nonmonotonic Logics. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence* Morgan Kaufmann Publ., Los Altos, CA, 1987: 388–392. Reprinted in M. L. Ginsberg (ed.), *Reading in Nonmonotonic Reasoning*, Morgan Kaufmann Publ., Los Altos, CA, 1987: 227–250.
21. Raymond M. Smullyan. *First-Order Logic*. Springer-Verlag, Berlin, 1968.