Open access • Book Chapter • DOI:10.1007/978-1-4471-1488-8_2

Towards a computational treatment of deontic defeasibility — Source link

Alberto Artosi, Guido Governatori, Giovanni Sartor

Institutions: University of Bologna

Published on: 01 Jan 1996 - Deontic Logic in Computer Science

Topics: Normal modal logic, Deontic logic, Multimodal logic, Modal operator and Accessibility relation

Towards a Computational Treatment of Deontic Defeasibility

Alberto Artosi[1], Guido Governatori[2], and Giovanni Sartor[3]
[1] Dipartimento di Filosofia, Università di Bologna,
via Zamboni 38, 40126 Bologna, Italy

[2] CIRFID, Università di Bologna,
via Galliera 3, 40121 Bologna, Italy

[3] IDG
via Panciatichi, 56/16, 50127 Firenze, Italy
e-mail: governat,sartor@cirfid.unibo.it

Abstract

In this paper we describe an algorithmic framework for a multi-modal logic arising from the combination of the system of modal (epistemic) logic devised by Meyer and van der Hoek for dealing with nonmonotonic reasoning with a deontic logic of the Jones and Pörn-type. The idea behind this (somewhat eclectic) formal set-up is to have a modal framework expressive enough to model certain kinds of deontic defeasibility, in particular by taking into account preferences on norms. The appropriate inference mechanism is provided by a tableau-like modal theorem proving system which supports a proof method closely related to the semantics of modal operators. We argue that this system is particularly well-suited for mechanizing nonmonotonic forms of inference in a monotonic multi-modal setting.

1 Introduction

In the last few years application of defeasible (nonmonotonic) reasoning methods to normative rules has become a major topic in the study of norms and normative reasoning. Defeasible reasoning methods have been proven to be able to cope with a variety of important issues. A number of different approaches to formalizing defeasible deontic reasoning have been proposed and several frameworks have been exploited (see e.g. [1, 2, 3, 4, 5, 6, 7]). It turns out, however, that not much effort to account for deontic defeasibility in a computationally oriented manner has been made until now. This can partly be traced to the fact that usual nonmonotonic reasoning methods are not well-suited to computational treatment. On the other hand, it has been argued (see [6]) that computational approaches to deontic defeasibility, e.g. in a logic programming setting, do not fit in with existing inference techniques from the field of non-classical logic theorem proving. In this paper we shall present a computational framework for deontic defeasible reasoning based neither on logic programming nor on current nonmonotonic formalisms but on a system of modal (epistemic)

M. Brown and J. Carmo (eds)
Deontic Logic Agency and Normative Systems, Springer-Verlag., pp. 27–46, 1996.

logic devised by Meyer and van der Hoek [8] for dealing with nonmonotonic reasoning by monotonic means. The interesting point is that we extend Meyer and van der Hoek's system to a multi-modal system which subsumes a deontic logic of the Jones and Pörn-type [9, 10]. The idea behind this (somewhat eclectic) formal set-up is to have a modal framework expressive enough to model certain kinds of deontic defeasibility, in particular by taking into account preferences on norms, and that can be easily housed in the computational setting provided by the KEM theorem prover for (normal) modal logics presented in [11]. The paper is organized as follows. In the next two sections we provide a concise overview of Jones-Pörn's and Meyer-van der Hoek's logics. In section 4 we provide some hints about how represent defaults in Meyer van der Hoek's modal setting. In section 5 we shall outline the multi-modal system resulting from combining the above logics. In section 6 we shall provide a brief description of the computational framework KEM. In sections 7, 8 and 9 we shall show how all the above mentioned systems can be accommodated in KEM's computational setting. Finally in section 10 we shall discuss reasoning with preferences in such a setting.

2 Deontic Logic

The deontic basis of our system is provided by Jones and Pörn's [9, 10] deontic logic DL, an extension of standard deontic logic (SDL) which incorporates, besides the normal deontic operators O^i and P^i, the deontic operators O^s and P^s. $\mathsf{O}^i A$ and $\mathsf{P}^i A$ (at a world w) mean (as in SDL): A holds in all, respectively some, of w's deontically ideal versions. $\mathsf{O}^s A$ and $\mathsf{P}^s A$ (at a world w) mean: A holds in all, respectively some, of w's "sub-ideal versions" (intuitively, a sub-ideal version of w is a world where not everything that is ideal in w holds). DL allows us to define the following notions:

- $\mathsf{N}_D A =_{df} (\mathsf{O}^i A \wedge \mathsf{O}^s A)$ (*Deontic Necessity*)
- $\mathsf{O}_\mathsf{T} A =_{df} (\mathsf{O}^i A \wedge \mathsf{P}^s \neg A)$ (*Ought*).

Since DL is a straightforward extension of SDL both $\mathsf{O}^i, \mathsf{P}^i$ and $\mathsf{O}^s, \mathsf{P}^s$ behave as normal KD-modalities. A model for DL is thus a structure:

$$\langle W, R_i, R_s, v\rangle$$

where R_i, $R_s \subseteq W \times W$ are serial (non reflexive) relations on W (intuitive reading: $wR_iv = v$ is an ideal version of w, $wR_sv = v$ is a sub-ideal version of w), subject to the following conditions:

C1: $R_i \cap R_s = \emptyset$;

C2: $\{\langle w, w\rangle : w \in W\} \subseteq R_i \cup R_s$.

The meaning of **C1** and **C2** is that there cannot exist ideal worlds that are also sub-ideal, and every world is either ideal or sub-ideal relative to itself (notice that this amounts to introducing some form of reflexivity in the model). υ is as usual with the following clauses for O^i and O^s respectively:

$$\models_w \mathsf{O}^i A \Leftrightarrow \forall v \in W : wR_i v, \models_v A$$
$$\models_w \mathsf{O}^s A \Leftrightarrow \forall v \in W : wR_s v, \models_v A.$$

3 A Modal Logic for Defeasible Reasoning (MDL)

Nonmonotonic reasoning is concerned with reasoning about and with *defeasible* information. In [8] this is modelled by a system of modal logic incorporating, besides the standard modal operators $\Box$ and $\Diamond$, n operators $P_1 \ldots P_n$ which refer to "preferred" subsets of the set W of (epistemically) possible worlds. $P_i A$ $(1 \leq i \leq n)$ means: A holds in all the i-preferred worlds (i.e. all the worlds in the set Σ_i). Axiomatically, such a system is obtained by adding to the standard $S5$ system the following axioms:

1. $\Box P_i A \equiv P_i A$
2. $\neg P_i \bot \rightarrow (P_i P_j A \equiv P_j A)$
3. $\neg P_i \bot \rightarrow (P_i \Box A \equiv \Box A)$
4. $\Box A \rightarrow P_i A \, (1 \leq i \leq n)$.

It turns out that this system (Meyer and van der Hoek's $S5P_{(n)}$, henceforth referred to as MDL = *modal default logic*) is that of a multi-modal $K45/S5$ system with $\Box$ and $\Diamond$ behaving as $S5$ modalities and the P_i behaving as $K45$ modalities. A model for MDL is thus a structure

$$< W, \Sigma_1, \ldots, \Sigma_n, R, R_1, \ldots, R_n, \upsilon >$$

where $\Sigma_i \subseteq W$ are subsets (possibly empty) of preferred worlds; $R \subseteq W \times W$ is the standard $S5$ accessibility relation on W; $R_i = \Sigma \times \Sigma_i \subseteq R$ is a $K45$ (i.e. transitive and euclidean) accessibility relation on Σ_i; and υ is as usual with the following additional clause:

$$\models_w P_i A \Leftrightarrow \forall v \in W : wR_i v, \models_v A, (1 \leq i \leq n).$$

4 Representing defaults in MDL

According to [8] default reasoning is treated by translating the usual default rules in the MDL language as follows. Reiter's classical rule $\frac{A:B}{C}$ is translated into $A \wedge \Diamond B \rightarrow P_i C$. This is read as "if A is true and B is considered possible, then C is preferred (practically believed)". Similarly, normal defaults become $A \wedge \Diamond B \rightarrow P_i B$, and multiple defaults $A_1 \wedge \Diamond B_1 \rightarrow P_1 C_1$,

$A_2 \wedge \Diamond B_2 \rightarrow P_2 C_2 \ldots$ where P_1 and P_2 are preference operators which can be associated either with the same cluster or with distinct preferred sets. In [8] this representation is extended with a mechanism for belief revision to obtain defeasibility. We propose a different solution, where defeasibility is obtained by analyzing defaults and assigning indices to them, according to the following definition. Let the input of our system be a knowledge base consituted of two sets of formulas: a set F of facts, and a set D of normal defaults:

$$\frac{A : B}{B}.$$

Let each default $\frac{A:B}{B}$ in D be translated as

$$\Box(A \rightarrow B)$$

that we abridge as

$$A \Rightarrow B.$$

Let D' be the set of all such formulas and let $S_1, \ldots, S_n$ be all maximal subsets of D' which are consistent with the facts (i.e. $S_i \cup F$ is consistent and for no $S_j, S_i \subseteq S_j$). The key idea is that of introducing an operator P_i in the consequent of every formula in S_i, thus obtaining $A \Rightarrow P_i B$ for each formula $A \Rightarrow B$ in S_i. Let our translated knowledge base be $F \cup D_M$, where D_M denotes the set of modal defaults resulting from D' by assigning preferences.

As a result of this translation, we obtain that (a) the defaults included in each set S_i will be assigned the same preference index while defaults occurring in different sets are assigned different preference indices (obviously defaults occurring in every set will receive all indices); (b) every inconsistency is avoided, since inconsistent defaults are assigned to different preferred sets. This approach recalls the treatment of prioritized defaults of Brewka [12, 13]. Note that if $F \cup D'$ is consistent, then $D_M = D'$. Note also that in our translation we no longer need the justification part of the default, which was modalized in [8] as $\Diamond B$, since our mechanism for assigning preferences takes care of consistency checks.

The above translation enables us to perform a kind of skeptical default inference, as the following example shows.

Example 4.1

$$(\{p, q\}, \{\frac{p : r}{r}, \frac{r : w}{w}, \frac{q : \neg w}{\neg w}\}).$$

Given this knowledge base, our procedure gives rise to the following translation:

$$p, q, p \Rightarrow P_{\{1,2\}} r, r \Rightarrow P_1 w, q \Rightarrow P_2 \neg w.$$

Clearly, the above premises imply both $P_1 w$, and $P_2(\neg w)$. Defeasibility is obtained by revising indices after modifying the knowledge base. Let us assume that a new fact w is added to our knowledge base, so that we obtain the set

$$(\{p, q, w\}, \{\frac{p : r}{r}, \frac{r : w}{w}, \frac{q : \neg w}{\neg w}\}).$$

Clearly, we have now just one set of defaults whose conclusion is consistent with the facts, that is $\{\frac{p:r}{r}, \frac{r:w}{w}\}$. Therefore our translation simply gives:

$$p, q, w, p \Rightarrow P_1 r, r \Rightarrow P_1 w.$$

Note that the model just proposed can be extended with mechanisms for ordering single defaults, and/or for ordering sets of defaults, possibly on the basis of the ordering over single defaults (on the combination of consistency check and ordering, see [12, 13]. We will not consider such mechanisms, which have been much discussed in the literature, but if such methods were adopted it would be possible to restrict admitted set of defaults (and the corresponding preference operators) just to the best (maximal) sets of defaults.

Let us consider a further example to illustrate our method.

Example 4.2 *Acting in self-defence*

The following knowledge base contains two conflicting rules, one saying that committing a tort implies a responsibility, the and the other saying that acting in self-defence implies no responsibility.

1a. John committed a tort.

2a. John acted in self-defence.

3a. People commiting a tort is responsible.

4a. People acting in self-defence is not responsible.

This situation is translated into the following knowledge base

1b. t

2b. d

3b. $\frac{t : r}{r}$

4b. $\frac{d : \neg r}{\neg r}$.

The corresponding modal translation is

1c. t

2c. d

3c. $t \Rightarrow P_1 r$

4c. $d \Rightarrow P_2 \neg r$.

Clearly, this set safely implies the two consequences $P_1 r$ and $P_2 \neg r$.

5 A multi-modal MDL/DL system for defeasible normative reasoning

The idea of treating defeasible deontic reasoning by combining an existing deontic logic with an already existing nonmonotonic formalism has been suggested by [4, 5]. Following this suggestion, in this section we shall combine MDL with DL to obtain a formalism suited for dealing with defeasible rules in normative reasoning. It is immediate to see that the combination of MDL with DL results in a multi-modal $KD/K45/S5$ system having $\Box, \Diamond, P_1 \dots P_n$, $\mathsf{O}^i, \mathsf{P}^i, \mathsf{O}^s$ and P^s as its (independent but interacting) modalities. For ease of reference we shall call this system DDL (*defeasible deontic logic*). A models for DDL is a mixed MDL/DL structure

$$\langle W, \Sigma_1, \dots \Sigma_n, R, R_i, R_s, R_1, \cdots, R_n, v \rangle$$

where $\Sigma_1, \dots \Sigma_n, R, R_i, R_s$ and $R_j, (1 \leq j \leq n)$ are as before. This means, intuitively, that we have a semantic setting working with several kinds of worlds, i.e.

- possible worlds
- preferred worlds
- deontic worlds $\left\{ \begin{array}{l} \text{deontically ideal worlds} \\ \text{deontically sub-ideal worlds} \end{array} \right.$

for $\Box, P_j, \mathsf{N}_D, \mathsf{O}^i, \mathsf{O}^s$, respectively.

The resources available in DDL allow us to treat some cases involving defeasibility, and others appealing to the DL features. For example, the famous Chisholm paradox can be solved without using defeasibility, but by simply making use of DL.

Example 5.1 *Chisholm Paradox*

1. It ought to be that John does not impregnate Suzy Mae.
2. Not-impregnating Suzy Mae commits John to not-marrying her.
3. Impregnating Suzy Mae commits John to marrying her.
4. John impregnates Suzy Mae.

As it is well-known there have been proposed several formalizations of this celebrated paradox (see [14]). Let us take the following default-like version:

1a. $\mathsf{O}_\top \neg A$

2a. $\dfrac{\neg A : \mathsf{O}_\top \neg B}{\mathsf{O}_\top \neg B}$

3a. $\dfrac{A : \mathsf{O}_{\mathsf{T}} B}{\mathsf{O}_{\mathsf{T}} B}$

4a. A

Our $F \cup D'$ translation being consistent we simply obtain

1b. $\mathsf{O}_{\mathsf{T}} \neg A$

2b. $\neg A \Rightarrow P_1(\mathsf{O}_{\mathsf{T}} \neg B)$

3b. $A \Rightarrow P_1 \mathsf{O}_{\mathsf{T}} B$

4b. A

which corresponds substantially to the formalization proposed by Jones and Pörn [9]. In such a case we derive both $\mathsf{O}_{\mathsf{T}} B$ an $\mathsf{O}^i \mathsf{O}_{\mathsf{T}} \neg B$ (meaning that in all ideal situations there is the obligation $\mathsf{O}_{\mathsf{T}} \neg B$, although in the actual, sub-ideal situation, it holds $\mathsf{O}_{\mathsf{T}} B$).

Let us now consider the following example.

Example 5.2 *Acting in different normative systems*

1a. If Mustafa is muslim he can be poligamous.

2a. If Mustafa is Italian, he cannot be poligamous.

3a. Mustafa is both muslim and Italian.

This set has the following default–like representation:

1b. $\dfrac{m : \mathsf{P}^i p}{\mathsf{P}^i p}$

2b. $\dfrac{i : \neg \mathsf{P}^i p}{\neg \mathsf{P}^i p}$

3b. $m \wedge i$.

The corresponding modal translation is

1c. $m \Rightarrow P_1 \mathsf{P}^i p$

2c. $i \Rightarrow P_2 \neg \mathsf{P}^i p$

3c. $m \wedge i$.

From this we can derive that according to preference P_1 (which gives priority to religious prescriptions) Mustafa can be poligamous ($P_1 \mathsf{P}^i p$), and according to preference P_2 (which gives preference to Italian law) he cannot be poligamous ($P_2 \neg \mathsf{P}^i p$).

Let us finally consider an example which clearly involves not only a conflict, but also the need to choose between different preferences.

Example 5.3 *A question about table manners [2]*

1a. A person must use knife and fork when eating.

2a. A person must not use knife and fork when eating asparagus.

3a. A person is eating asparagus.

This set is translated into

1b. $e \Rightarrow P_1 \mathsf{O}_{\mathsf{T}} f$

2b. $e \wedge a \Rightarrow P_2 \mathsf{O}_{\mathsf{T}} \neg f$

3b. $e \wedge a$.

In such a case we need to add to our system a mechanism for comparing preferences. Here the preference P_2 should be chosen because it is more specific than P_1.

6 The computational framework KEM

In [11, 15] we presented a tableau-like proof system, called KEM which has been proved to be able to cope with a wide variety of (normal propositional) modal and multi-modal logics [16]. KEM is based on D'Agostino and Mondadori's [17] classical system KE, a combination of tableau and natural deduction inference rules which allows for a restricted ("analytic") use of the cut rule. The key feature of KEM, besides its being based neither on resolution nor on standard sequent/tableau inference techniques, is that it automatically generates models and checks them for putative contradictions using a label scheme to bookkeep "world" paths. In this section we shall present KEM in barest outline. We first recall some basic notions.

Our base language will be a modal propositional language L defined in the usual way. We shall use the letters $A, B, C, ...$ to denote arbitrary formulas of L. By a *signed formula* (*S-formula*) we shall mean an expression of the form SA where $S \in \{T, F\}$ and A is a formula of L (intended meaning: $TA = A$ holds, $FA = A$ does not hold at a given word in some Kripke model). We shall denote by X, Y, Z arbitrary signed formulas. As usual X^C will be used to denote the *conjugate* of a S-formula X, i.e. the result of changing S to its opposite (with the exception of the following S-formulas: $T\Box A, F\Diamond A, F\Box A$ and $F\Diamond A$ which also have $T\Diamond\neg A, F\Box\neg A, F\Diamond\neg A, T\Box\neg A$ respectively as their conjugates). Two S-formulas X, Z such that $Z = X^C$, will be called *complementary*.

As we have just said KEM approach wants we work with "world" labels. A "world" label is either a constant or a variable "world" symbol or a "structured" sequence of world-symbols we call a "world-path". Intuitively, constant and variable world-symbols stand for worlds and sets of worlds respectively, while a world-path conveys information about access between the worlds in it. We attach labels to S-formulas to yield *labelled signed formulas* (*LS-formulas*), i.e.

pairs of the form X, i where X is a S-formula and i is a label. A LS-formula SA, i means, intuitively, that A is true (false) at the (last) world (on the path represented by) i. 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, logic-dependent) unification algorithm. That two world-paths i and k are unifiable means, intuitively, that they virtually represent the same path, i.e. any world which you could get by the path i could be reached by the path k and vice versa. LS-formulas whose labels are unifiable turn out to be true (false) at the same world(s) relative to the accessibility relation that holds in the appropriate class of models. In particular two LS-formulas X, X^C whose labels are unifiable stand for formulas which are contradictory "in the same world". These ideas are formalized as follows.

6.1 Label formalism

Let L^n be a multi-modal language with n distinct operators. For the sake of the following discussion let us assume that the semantics of L^n is given by a structure $M = \langle W, \Sigma_1, \ldots, \Sigma_m, R_1, \ldots, R_n, \upsilon \rangle$ where W is a (non-empty) set of possible worlds, $\Sigma_i \subseteq W$ $(1 \leq i \leq m)$ and $R_i \subseteq W \times W$ $(1 \leq i \leq n)$. For $1 \leq i \leq n$ let $\Phi^i_C = \{w^i_1, w^i_2, \cdots\}$ and $\Phi^i_V = \{W^i_1, W^i_2, \cdots\}$ be (non empty) sets respectively of constant and variable world-symbols. Let us define

$$\Phi_C = \bigcup_{1 \leq i \leq n} \Phi^i_C \text{ and}$$

$$\Phi_V = \bigcup_{1 \leq i \leq n} \Phi^i_V.$$

The set $\Im$ of world-labels is now defined as

$$\Im = \bigcup_{1 \leq i} \Im_i \text{ where } \Im_i \text{ is :}$$

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

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

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

That is a world-label is either (i) an element of the set Φ_C, or (ii) an element of the set Φ_V, or (iii) a path term (k', k) where (iiia) $k' \in \Phi_C \cup \Phi_V$ and (iiib) $k \in \Phi_C$ or $k = (m', m)$ where (m', m) is a label. According to the above informal explanation, we may think of a label $i \in \Phi_C$ as denoting a (given) world, and a label $i \in \Phi_V$ as denoting a set or worlds (any world) in M. A label $i = (k', k)$ may be viewed as representing a path from k to a (set of) world(s) k' accessible from k. For instance, $(w^i_2, (W^i_1, w^i_1))$ represents a path which takes us to a world w^i_2 accessible via any world accessible from w^i_1 (i.e. accessible from the subpath (W^i_1, w^i_1)) according to R_i (notice that the labels are read from right to left). In what follows we shall use $w_j(W_j)$ to generically refer to a given (any) world quite apart from its i. To facilitate reference we shall sometimes use different names for different kinds of labels.

A bit of terminology. For any label $i = (k', k)$ we call k' the *head* of i, k the *body* of i, and denote them by $h(i)$ and $b(i)$ respectively. Notice that these notions are recursive: if $b(i)$ denotes the body of i, then $b(b(i))$ will denote the body of $b(i)$, $b(b(b(i)))$ will denote the body of $b(b(i))$; and so on. For example, if i is $(w_4, (W_3, (w_3, (W_2, w_1))))$, then $b(i) = (W_3, (w_3, (W_2, w_1)))$, $b(b(i)) = (w_3, (W_2, w_1))$, $b(b(b(i))) = (W_2, w_1)$, $b(b(b(b(i)))) = w_1$. We call each of $b(i)$,$b(b(i))$, etc., a *segment* of i. Let $s(i)$ denote any segment of i (obviously, by definition every segment $s(i)$ of a label i is a label); then $h(s(i))$ will denote the head of $s(i)$. For any label i, we define the length of i, $l(i)$, as the number of world-symbols in i, i.e. $l(i) = n \Leftrightarrow i \in \Im_n$. $s^n(i)$ will denote the segment of i of lenght n, i.e. $s^n(i) = s(i)$ such that $l(s(i)) = n$. For any label $i, l(i) > n$, we define the *countersegment-n* of i, $c^n(i)$ (i.e. what remains of i after deleting $s^n(i)$), as:

$$c^n(i) = h(i) \times (\cdots \times (h(s^k(i)) \times (\cdots \times (h(s^{n+1}, w_0) \cdots) \cdots)(n < k < l(i))$$

We shall call a label i *restricted* if $h(i) \in \Phi_C$, otherwise we call it *unrestricted*. We shall say that a label k is *i-preferred* iff $k \in \Im^i$ where

$$\Im^i = \{k \in \Im : h(k) \text{ is either } w^i_m \text{ or } W^i_m, 1 \leq i \leq n\},$$

and that a label k is *i-ground* $(1 \leq i \leq n)$ iff

1. $\forall s(k) : h(s(k)) \notin \Phi^i_V$, and
2. if $\exists s^m(k) : h(s^m(k)) \in \Phi^i_V$, then $\exists s^j(k), j < m : h(s^j(k)) \in \Phi^i_C$.

6.2 Unification scheme

KEM's label unification scheme involves two kinds of unifications, respectively "high" and "low" unifications. "High" unifications are meant to mirror specific accessibility constraints; they are used to build "low" unifications, which account for the full range of conditions governing the appropriate accessibility relation. We then begin by defining the basic notion of "high" unification. First we define a substitution in the usual way as a function

$$\begin{aligned} \sigma \quad &: \quad \Phi^0_V \longrightarrow \Im^- \\ &: \quad \Phi^i_V \longrightarrow \Im^i, (1 \leq i \leq n) \end{aligned}$$

where $\Im^- = \Im - \Phi_V$. For two labels i, k and a substitution σ, if σ is a unifier of i and k then we shall say that i and k are σ-unifiable. We shall (somewhat unconventionally) use $(i, k)\sigma$ to denote both that i and k are σ-unifiable and the result of their unification. On this basis we define several specialised, logic-dependent notions of both σ "high" (or σ^L) and σ "low" (or σ_L) unification (see [11]). For example, the notion of two labels i, k being σ^K-, σ^D-, and σ^{S5}-unifiable is defined in the following way:

$$
\begin{aligned}
(i,k)\sigma^K &= (i,k)\sigma \iff \\
&\quad \text{at least one of } i \text{ and } k \text{ is restricted, and} \\
&\quad \text{for every } s(i),\ s(k),\ l(s(i)) = l(s(k)),\ (s(i), s(k))\sigma^K \\
(i,k)\sigma^D &= (i,k)\sigma \\
(i,k)\sigma^{S5} &= (h(i), h(k))\sigma.
\end{aligned}
$$

For example, let $i = (w_3, (W_1, w_1))$ and $k = (W_3, (w_2, w_1))$ be two labels to be unified. They σ^K-unify to $(w_3, (w_2, w_1))$ according to the above definition of σ^K-unification. Notice that they are also σ^D-unifiable (this preserves the obvious relation between the K and D logics). Let us consider now two labels $i = (w_3, (W_1, w_1))$ and $k = (W_3, (W_2, w_1))$. They σ^D-unify to $(w_3, (w_2, w_1))$ according to the above definition of σ^D-unification. They are not, however, σ^K-unifiable (according to the second condition of the above definition) since the segments $(W_1, w_1))$, (W_2, w_1) are not σ^K-unifiable (by the first condition of the above definition). The reason is that in the "non idealisable" logic K the "denotations" of W_1 and W_2 may be empty (i.e. there can be no worlds accessible from w_1), which obviously makes their unification impossible, while in the "idealisable" logic D they are not empty, which makes them to be unifiable "on" any constant. As to the notion of σ^{S5}-unification, take $i = (w_3, (W_1, w_1))$ as before, and $k = (W_2, (W_1, (w_2, w_1)))$. These labels σ^{S5}-unify to w_3 according to the above definition of σ^{S5}-unification, which for a label of the form (m', m) amounts to deleting m from the path to a world m' (since, if access is "ubiquitous", then the "way to" m' is irrelevant).

For $L = K, D, S5$ the notion of two labels i, k being σ_L-unifiable is defined quite simply as:

$$
\begin{aligned}
(i,k)\sigma_K &= (i,k)\sigma^K \\
(i,k)\sigma_D &= (i,k)\sigma^D \\
(i,k)\sigma_{S5} &= (i,k)\sigma^{S5}.
\end{aligned}
$$

Notice that in the simple cases above $(i,k)\sigma_L = (i,k)\sigma^L$ (i.e. high and low unifications are alike). For a more complex case see section 9 below.

6.3 Inference rules

The following formulation uses a generalized "$\alpha, \beta, \nu_i, \pi_i$" form of Smullyan-Fitting's "α, β, ν, π" unifying notation.

$$
\frac{\alpha, i}{\alpha_1, i} \qquad\qquad \frac{\alpha, i}{\alpha_2, i} \qquad\qquad (\alpha)
$$

$$\frac{\begin{array}{c}\beta, i \\ \beta_1^C, k\end{array}}{\beta_2, (i,k)\sigma_L}[(i,k)\sigma_L] \qquad \frac{\begin{array}{c}\beta, i \\ \beta_1^C, k\end{array}}{\beta_2, (i,k)\sigma_L}[(i,k)\sigma_L] \tag{β}$$

$$\frac{\nu_i, i}{\nu_0, (m,i)}[m \in \Phi_V^i \text{ and new}] \tag{ν_i}$$

$$\frac{\pi_i, i}{\pi_0, (m,i)}[m \in \Phi_C^i \text{ and new}] \tag{π_i}$$

$$\frac{}{X, i \qquad X^C, i}[i \text{ restricted}] \tag{PB}$$

$$\frac{\begin{array}{c}X, i \\ X^C, k\end{array}}{\times(i,k)\sigma_L}[(i,k)\sigma_L] \tag{PNC}$$

Here the α-rules are just the familiar linear branch-expansion rules of the tableau method, while the β-rules correspond to such common natural inference patterns as *modus ponens*, *modus tollens*, etc. (i, k, m stand for arbitrary labels). The rules for the modal operators are as usual. "m new" in the proviso for the ν_i- and π_i-rule means: m must not have occurred in any label yet used. Notice that in all inferences via an α-rule the label of the premise carries over unchanged to the conclusion, and in all inferences via a β-rule the labels of the premises must be σ_L-unifiable, so that the conclusion inherits their unification. PB (the "Principle of Bivalence") represents the (LS-version of the) semantic counterpart of the cut rule of the sequent calculus (intuitive meaning: a formula A is either true or false in any *given* world, whence the requirement that j should be restricted). PNC (the "Principle of Non-Contradiction") corresponds to the familiar branch-closure rule of the tableau method, saying that from the occurrence of a pair of LS-formulas $X, i,\ X^C, k$ such that $(i,k)\sigma_L$ (let us call them *σ_L-complementary*) on a branch we may infer the closure ("$\times$") of the branch. The $(i,k)\sigma_L$ in the "conclusion" of PNC means that the contradiction holds "in the same world".

6.4 Proof search

Let $\Gamma = \{X_1, \ldots, X_m\}$ be a set of S-formulas. Then $\mathcal{T}$ is a *KEM-tree for* Γ if there exists a finite sequence $(\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n)$ such that (i) $\mathcal{T}_1$ is a 1-branch tree consisting of $\{X_1, i \ldots, X_m, i\}$, where i is an arbitrary constant label; (ii) $\mathcal{T}_n = \mathcal{T}$, and (iii) for each $i < n, \mathcal{T}_{i+1}$ results from $\mathcal{T}_i$ by an application of a rule of KEM. A branch τ of a KEM-tree $\mathcal{T}$ of LS-formulas is said to be *σ_L-closed* if it ends with an application of PNC, open otherwise. A KEM-tree $\mathcal{T}$ is said to be $\sigma_L - closed$ if all its branches are σ_L-closed. As usual with tableau methods, a set Γ of formulas is checked for consistency by constructing a KEM-tree for Γ. It is worth noting that each KEM-tree is a (class of) Hintikka's model(s) where the labels denote worlds (i.e. Hintikka's modal sets), and the unifications behave according to the conditions placed on the appropriate accessibility relations. By a $KEM-proof$ of a formula A we mean a σ_L-closed KEM-tree for FA, i. Moreover we say that a formula A is a *KEM-consequence of a set of formulas* Γ if A occurs in all the open branches of a KEM-tree for Γ. We now describe a sistematic procedure for KEM. First we define the following notions.

Given a branch τ of a KEM-tree, we shall call an LS-formula X, i *E-analysed in* τ if either (i) X is of type α and both α_1, i and α_2, i occur in τ; or (ii) X is of type β and one of the following conditions is satisfied: (a) if β_1^C, k occurs in τ and $(i,k)\sigma_L$, then also $\beta_2, (i,k)\sigma_L$ occurs in τ, (b) if β_2^C, k occurs in τ and $(i,k)\sigma_L$, then also $\beta_1, (i,k)\sigma_L$ occurs in τ; or (iii) X is of type ν_i and $\nu_0, (m,i)$ occurs in τ for some $m \in \Phi_V$ not previously occurring in τ, or (iv) X is of type π_i and $\pi_0, (m,i)$ occurs in τ for some $m \in \Phi_C$ not previously occurring in τ.

We shall call a branch τ of a KEM-tree *E-completed* if every LS-formula in it is E-analysed and it contains no complementary formulas which are not σ_L-complementary. We shall say a branch τ of a KEM-tree *completed* if it is E-completed and all the LS-formulas of type β in it either are analysed or cannot be analysed. We shall call a KEM-tree *completed* if every branch is completed.

At each stage of proof search (i) we choose an open non completed branch τ. If τ is not E-completed, then (ii) we apply the 1-premise rules until τ becomes E-completed. If the resulting branch τ' is neither closed nor completed, then (iii) we apply the 2-premise rules until τ becomes E-completed. If the resulting branch τ' is neither closed nor completed, then (iv) we choose an LS-formula of type β which is not yet analysed in the branch and apply PB so that the resulting LS-formulas are β_1, i' and β_1^C, i' (or, equivalently β_2, i' and β_2^C, i'), where $i = i'$ if i is restricted (and already occurring when $h(i) \in \Phi_C^h$), otherwise i' is obtained from i by instantiating $h(i)$ to a constant not occurring in i; (v) ("Modal PB") if the branch is not E-completed nor closed, because of complementary formulas which are not σ_L-complementary, then we have to see whether a restricted label unifying with both the labels of the complementary formulas occurs previously in the branch; if such a label exists, or can be built using already existing labels and the unification rules, then the branch is closed, (vi) we repeat the procedure in each branch generated by PB.

The above procedure is based on on a (deterministic) procedure working for *canonical* KEM-tree. A KEM-tree is said to be canonical if it is generated by applying the rules of KEM in the following fixed order: first the α-, ν_i- and π_i-rule, then the β-rule and PNC, and finally PB. Two interesting properties of canonical KEM-trees are (i) that a canonical KEM-tree always terminates, since for each formula there are a finite number of subformulas and the number of labels which can occur in the KEM-tree for a formula A (of L) is limited by the number of modal operators belonging to A, and (ii) that for each closed KEM-tree a closed canonical KEM-tree exists. Proofs of termination and completeness for canonical KEM-trees are given in [16].

7 KEM for MDL

To extend KEM to MDL we introduce n sets $\Phi^i_C\,(1 \leq i \leq n)$ of preferred constant world–symbols and as many sets Φ^i_V of preferred variable world–symbols. Moreover we define the following specialised notions of σ^L- and σ_L-unification for dealing with these sets.

$$\begin{array}{rcl} (i,k)\sigma^{MDL} & = & (h(i),h(k))\sigma \text{ iff} \\ & & i, \text{ or } k \text{ is } i\text{-ground}, 1 \leq i \leq n, \text{ or} \\ & & \exists s(i), s(k) : h(s(i)), h(s(k)) \in \Phi^i, \text{ and} \\ & & (h(s(i)), h(s(k))\sigma^{MDL}. \end{array}$$

On this basis we define the corresponding low unification as:

$$(i,k)\sigma_{MDL} = (i,k)\sigma^{MDL}.$$

Here we give an example proof of a characteristic axiom of MDL.

1. $F\neg P_i\bot \to (P_iP_jA \equiv P_jA)$	w_1
2. $T\neg P_i\bot$	w_1
3. $FP_iP_jA \equiv P_jA$	w_1
4. $FP_i\bot$	w_1
5. $F\bot$	(w^i_2, w_1)

6. TP_iP_jA	w_1	7. FP_iP_jA	w_1
8. FP_jA	w_1	13. TP_jA	w_1
9. TP_jA	(W^i_1, w_1)	14. FP_jA	(w^i_4, w_1)
10. TA	$(W^j_2, (W^i_1, w_1))$	15. FA	$(w^j_5, (w^i_4, w_1))$
11. FA	(w^j_3, w_1)	16. TA	(W^j_3, w_1)
12. $\times$	w^j_3	17. $\times$	w^j_5

The core of the proof is in the application of PB to (3) to obtain (6) and (7). Is should also be noted that the label in (10) is i-ground due to the label in (5), which allows us to close the branch ((10) and (11) are σ_{MDL}-complementary). In the right branch the label of (15) is obviously i-ground, which makes the branch to be closed ((15) and (16) are σ_{MDL}-complementary).

8 KEM for DL

In this section we show how to employ KEM to deal with MDL. We first introduce three kinds of world symbols respectively for:

1. Universal deontic worlds: $\Phi_N = \{N_1, N_2, \cdots\}$ and $\Phi_n = \{n_1, n_2, \cdots\}$
2. Ideal worlds: $\Phi_D = \{D_1, D_2, \cdots\}$ and $\Phi_d = \{d_1, d_2, \cdots\}$
3. Sub–ideal worlds: $\Phi_S = \{S_1, S_2, \cdots\}$ and $\Phi_s = \{s_1, s_2, \cdots\}$.

We define

$$\sigma : \Phi_N \rightarrow \Im^d \cup \Im^s$$

In order to get the appropriate unifications we need the following substitution:

$$\sigma^{\#}\Phi_W = \sigma\Phi_W.$$

$$\begin{aligned} \sigma^{\#} &: \Phi_N \rightarrow \Phi^d \cup \Phi^s \\ &: \Phi_S \rightarrow \Phi^s \\ &: \Phi_D \rightarrow \Phi^d \end{aligned}$$

where

$$\Phi^d = \{i^r : r = d\} \qquad \Phi^s = \{i^r : r = s\}$$

$$\begin{aligned} (i,k)\sigma^W &= (t \times (b(s(i)), b(k))\sigma^J \iff \\ & \quad l(k) > 1, \exists s(i) : \forall s'(i), l(s'(i)) > l(s(i)), \\ & \quad (h(s'(i)), h(k))\sigma^{\#} = (h(s(i)), h(k))\sigma = t \text{ and} \\ & \quad (b(s(i)), b(k))\sigma^J \text{ otherwise}, l(k) = 1 \\ (i,k)\sigma^W &= t \end{aligned}$$

or

$$\begin{aligned} (i,k)\sigma^W &= (t \times (b(i), b(s(k)))\sigma^J \iff \\ & \quad l(i) > 1, \exists s(k) : \forall s'(k), l(s'(k)) > l(s(k)), \\ & \quad (h(i), h(s'(k)))\sigma^{\#} = (h(i), h(s(k)))\sigma = t \text{ and} \\ & \quad (b(i), b(s(k)))\sigma^J \text{ otherwise}, l(i) = 1 \\ (i,k)\sigma^W &= t \end{aligned}$$

where

$$(i,k)\sigma^J = (i,k)\sigma^D \text{ or } (i,k)\sigma^W.$$

We are now able to characterize DL by the following notions of σ^{DL}-, σ_{DL}-unification:

$$(i,k)\sigma^{DL} = \begin{cases} (i,k)\sigma^D \\ (i,k)\sigma^W \end{cases}$$

and

$$(i,k)\sigma_{DL} = (i,k)\sigma^{DL}.$$

To complete the KEM characterization of DL we need the following additional inference rules which are meant to capture the "meaning" of N_D, **C1** and **C2** respectively:

$$\frac{F\mathsf{N}_D A, i}{F(\mathsf{O}^i A \wedge \mathsf{O}^s A), i} \qquad \frac{\begin{matrix} X, (S,i) \\ X, (D,k) \end{matrix}}{X, (W, (i,k)\sigma_{DL})}[(i,k)\sigma_{DL}] \qquad \frac{\begin{matrix} \nu_i, i \\ \nu_0^C, k \end{matrix}}{\begin{matrix} \nu_i, j^r \\ \nu_0^C, j^r \end{matrix}}[(i,k)\sigma_{DL} = j]$$

where

$$i^r = i^s \quad \text{if } \nu_i = T\mathsf{O}^i A\,(F\mathsf{P}^i A)$$
$$i^r = i^d \quad \text{if } \nu_i = T\mathsf{O}^s A\,(F\mathsf{P}^s A)$$

and

$$i^x = i : h(i) \in \Phi^x, (x \in \{d, s\}).$$

Obviously each $\Phi^r_X \subseteq \Phi_X$. Besides the usual closure rule we introduce the following

$$\frac{i \in \Phi^i, i \in \Phi^s}{\times}.$$

stating that no world can be at the same time an ideal and a sub-ideal version of itself.

What follows is an example proof of a characteristic theorem of DL.

1. $F(\mathsf{O}^i A \wedge \neg A) \rightarrow \mathsf{P}^s \neg A$	w_1
2. $T\mathsf{O}^i A \wedge \neg A$	w_1
3. $F\mathsf{P}^s \neg A$	w_1
4. $T\mathsf{O}^i A$	w_1
5. FA	w_1
6. $T\mathsf{O}^i A$	w_1^s
7. FA	w_1^s
8. TA	S_1, w_1^s
9. $\times$	

The steps leading to the nodes (1)–(5) are straightforward. The nodes (6)–(7) come from the application of the rule for **C2** since the world denoted by w_1 is a sub–ideal version of itself. Closure follows immediately from (7) and (8), which are σ_{DL}-complementary (their labels σ_{DL}-unify because of $(S_1, w_1^s)\sigma^{\#}$).

9 KEM for DDL

In order to extend KEM to deal with DDL we have only to define the following special unifications:

$$\begin{aligned}
(i,k)\sigma_{DDL} &= (i,k)\sigma^{S5} \iff \\
&\quad h(i) \text{ or } h(k) \in \Phi^0_V \\
&\quad i, k \text{ are } i\text{-ground, and} \\
&\quad (i,k)\sigma^{S5} \notin \Phi_d \cup \Phi_s; \text{ otherwise} \\
(i,k)\sigma_{DDL} &= (h(i), h(k))\sigma \times (s^1(i), s^1(k))\sigma
\end{aligned}$$

or

$$\begin{aligned}
(i,k)\sigma_{DDL} &= (c^n(i), c^m(k))\sigma^X \text{ where} \\
&\quad w_0 = (s^n(i), s^m(k))\sigma_{DDL}
\end{aligned}$$

where

$$(i,k)\sigma^X = \begin{cases} (i,k)\sigma^{MDL} \\ (i,k)\sigma^{DL} \end{cases}.$$

The rules of inference for DDL result now from the combination of the rules for MDL and DL.

10 Dealing with preferences

In this section we show how the computational framework developed in the preceeding sections can be used to solve contradictions in the way described in section 4. As it well known, tableau methods, and in particular KEM, can be used as model building systems [11]. In what follows we exploit the obvious fact that the set of formulas in each branch τ of a KEM-tree is consistent (i.e. it has a model) if τ is open, otherwise it is inconsistent. Since we check sets of premises for consistency, we first run KEM on the given set and, finally, we solve contradictions, on each closed branch, by assigning a different preference to the consequents of the translated defaults implying contradictions. To use practically this procedure we have to keep trace of dependencies according to the following definition.

Definition 10.1

- *Each formula depends on itself;*
- *a formula B depends on A either if it is obtained through an application of the α- ν_i- and π_i-rules or it is obtained through an application of KEM's rules on formulas depending on A;*
- *a formula C depends on A, B if it is obtained through an application of a β-rule where A, B are its premises;*
- *if C depends on A, B then C depends on A and C depends on B*

In the course of model construction we shall only keep trace of the dependencies from the premises, i.e. we avoid all the intermediate formulas. Obviously in order to solve pairs of complementary formulas and to assign preferences accordingly, we have to take into account only the premises which essentially imply one of them.

Let A and B be two complementary formulas in a closed branch τ; let $\mathcal{C}_A$ be the set of premises from which A depends; let $\mathcal{C}_B$ be the set of premises from which B depends; let $\mathcal{D}(\mathcal{C}_A \cup \mathcal{C}_B)$ be the defaults in $\mathcal{C}_A \cup \mathcal{C}_B$. Note that, since facts are assumed to be consistent, the set $\mathcal{D}(\mathcal{C}_A \cup \mathcal{C}_B)$, which we call a *culprit* set, is responsible for the inconsistency,

We say that a set $S \subseteq D$ is *conflict free* if it does not contain any culprit sets. We use KEM inference procedure to find all the $\mathcal{D}(\mathcal{C}_A \cup \mathcal{C}_B) \subseteq D$, and we use such information to build all maximal conflict free sets S of defaults. Finally we assign preferences in such a way that all formulas in each S have the same preference.

We give an example of how we establish consistent subsets and assign preferences.

Example 10.2

Let us consider the following knowledge base (F, D), where $F = \{p\}$ and $D = \{p \to q, p \to r, p \to s, p \to (\neg q \vee \neg r) \wedge \neg s\}$.

1. Tq	1
2. Tr	2
3. Ts	3
4. $T(\neg q \vee \neg r) \wedge \neg s$	4
5. $T\neg q \vee \neg r$	4
6. Fs	4
7. Fq	$2, 4$
8. Fr	$1, 4$

On the right column we wrote the dependencies (for the sake of economy we have deleted the irrelevant steps). We obtain the following inconsistent sets of defaults: $\{1, 2, 4\}$ and $\{3, 4\}$. Therefore the maximal consistent subsets and their associate preferences are:

$P_1 \rightsquigarrow \{1, 2, 3\}$

$P_2 \rightsquigarrow \{1, 4\}$

$P_3 \rightsquigarrow \{2, 4\}$.

It is worth noting that the ability of KEM to determine the inconsistent subsets of a contradictory one is due to its inference rules, in particular PB and the β-rules.

After assigning preferences, we can query the system whether a formula X is a consequence of our knowledge base simply by running a refutation KEM-tree for the set consisting of the premises and X^C.

11 Final Remarks

It was not the objective of this paper to develop a theory of defeasible deontic reasoning. Our motivation was rather practical. We sought for computationally tractable and easily implementable theorem proving techniques suitable for dealing with some pieces of normative reasoning involving defeasible rules. The preceding discussion was thus mainly aimed at showing the potential scope of application of the method. In effect, we believe that the method for solving contradictions by assigning preferences outlined in section 10 nicely exploits the computational and proof-theoretical advantages offerred by the modal theorem proving system KEM. As we have argued elsewere this system enjoys most of the features a suitable proof search system for modal (and in general non-classical) logics should have. In contrast with (both clausal and non-clausal) resolution methods, and in general "translation-based" methods [18, 19], it works for the full modal language (thus avoiding any preprocessing of the input formulas), and it is flexible enough to be extended to cover any setting having a Kripke-model based semantics (this is clearly shown by our treatment of Jones and Pörn logic DL where the rules specific for such a logic should take care not only of the propositional and modal part but also of the structure of the labels and the relationship between labels and formulas; for example we added another closure rule $\frac{i\in\Phi^i, i\in\Phi^s}{\times}$ which states that no world can be at the same time an ideal and a sub-ideal version of itself; this result is achieved by determining when a deontic word is ideally (sub-ideally) reflexive (i^r) by means of another peculiar inference rule). From this perspective our methods is similar to sequent or tableau proof methods ([20, 22]). Nevertheless, it has several advantages over most tableau/sequent based theorem proving methods: being based on D'Agostino an Mondadori's classical proof system KE, it eliminates the typical redundancy of the standard cut-free methods and, thanks to its label unification scheme, it offers a simple and efficient solution to the permutation problem which notoriously arises at the level of the usual tableau-sequent rules for the modal operators. However, unlike e.g. Wallen's [21] connection method, it uses a natural and easily implementable style of proof construction, and so it appears to provide an adequate basis for combining both efficiency and naturalness. (As to the implementation the reader is referred to [15] where a Prolog implementation is provided, and to [23] where some related issues are discussed).

References

[1] Asher N., Bonevac D. *Prima Facie* Obligation. In: Jones A.J.I., Sergot M. (eds.) Δ*eon '94*. Tano, 1993, pp 19–36.

[2] Horty J.F. Moral Dilemmas and Nonmonotonic Logic. *Journal of Philosophical Logic* 1994; 23: 35–65.

[3] Jones A.J.I. On the Logic of Deontic Conditionals. *Ratio Juris* 1991; 3: 355–66.

[4] Mc Carty L.T. Defeasible Deontic Reasoning. *Fundamenta Informaticae* 1994; 21: 125–148.

[5] Prakken H. Two Approaches to Defeasible Reasoning. In: Jones A.J.I., Sergot M. (eds.) Δ*eon '94*. Tano, 1993, pp 281–295.

[6] Ryu Y.U., Lee R.M. Defeasible Deontic Reasoning: A Logic Programming Model. In: *Proceedings of the First International Workshop on Deontic Logic in Computer Science*. Amsterdam, 1991, pp 347–363.

[7] Sartor G. The Structure of Norm Conditions and Nonmonotonic Reasoning in Law. In: *Proceedings of the Third International Conference on Artificial Intelligence and Law*. ACM Press, 1991, pp 155–164.

[8] Meyer J.-J.Ch., van der Hoek V. A Modal Logic for Nonmonotonic Reasoning. In: van der Hoek W., Meyer J.-J.Ch., Tan Y. H., Witteveen C.

(eds.) *Non-Monotonic Reasoning and Partial Semantics*. Ellis Horwood, N.Y., 1992, pp 37–77.

[9] Jones A.J.I., Pörn I. Ideality, Sub-ideality and Deontic Logic. *Synthese*, 1985; 65: 275–290.

[10] Jones A.J.I., Pörn I. “Ought” and “Must”. *Synthese*, 66, 1986: 89–93.

[11] Artosi A., Governatori G. Labelled Model Modal Logic. In *Proceedings of the CADE-12 Workshop on Automated Model Building*. Nancy, 1994, pp 11–17.

[12] Brewka G. Belief Revision in a Framework for Default Reasoning. In: Fuhrmann A., Morreau M. (eds.) *The Logic of Theory Change. Workshop.* Springer-Verlag, Berlin, 1989, pp 206–222.

[13] Brewka G. *Nonmonotonic Reasoning. Logical Foundations of Commonsense*. Cambridge University Press, Cambridge, 1991.

[14] Åqvist L.. *Introduction to Deontic Logic and the Theory of Normative Systems.* Bibliopolis, Napoli, 1987.

[15] Artosi A., Cattabriga P., Governatori G. An Automated Approach to Normative Reasoning. In: Breuker J. (ed.), *Proceeding of the Workshop ECAI Artificial Normative Reasoning*. Amsterdam, 1994, pp 132–145.

[16] Governatori G. Labelled Tableaux for Multi-Modal Logics. In: Baumgartner P., Hähnle R., Posegga J. (eds.) *Theorem Proving with Analytic Tableaux and Related Methods*. Springer-Verlag, Berlin, 1995, pp 79-95 (Lecture Notes in Artificial Intelligence no.918).

[17] D’Agostino M., Mondadori M. The Taming of the Cut. *Journal of Logic and Computation* 1994; 4: 285–319.

[18] Auffray Y., Enjalbert P. Modal Theorem Proving: An Equational Viewpoint. *Journal of Logic and Computation* 1992; 2: 247-259.

[19] Ohlbach H.J. Semantics-Based Translation Methods for Modal Logics. *Journal of Logic and Computation* 1991,1: 691–746.

[20] Fitting M. First Order Modal Tableaux. *Journal of Automated Reasoning* 1988; 4: 191–213.

[21] Wallen L.A. *Automated Deduction in Non-Classical Logics*. The MIT Press, Cambridge (Mass.), 1990.

[22] Catach L. Tableaux: A General Theorem Prover for Modal Logic. *Journal of Automated Reasoning* 1991; 7: 489–510.

[23] Artosi A., Cattabriga P., Governatori G. A Prolog Implementation of *KEM* In: Alpuente, Sessa M.I. (eds.) *Proceedings of GULP-PRODE‘95.* Universitá di Salerno, Salerno, 1995, pp 395–400.