# A tableux system for Deontic Interpreted Systems

Guido Governatori[1] Alessio Lomuscio[2] Marek J. Sergot[3]

[1] School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane, Australia. guido@itee.uq.edu.au

[2] Department of Computer Science, King's College London, London, UK. alessio@dcs.kcl.ac.uk

[3] Department of Computing, Imperial College, London, UK. mjs@doc.ic.ac.uk

**Abstract.** We develop a labelled tableaux system for the modal logic $KD45_n^{i-j}$ extended with epistemic notions. This logic characterises a particular type of interpreted systems used to represent and reason about states of correct and incorrect functioning behaviour of the agents in a system, and of the system as a whole. The resulting tableaux system provides a simple decision procedure for the logic. We discuss these issues and we illustrate them with the help of simple examples

## 1 Introduction

One of the main areas of interest in the use formal methods in Software Engineering involve the use of tools based on mathematical logic for the specification and verification of computing systems. This is true in general but it specially applies to the area of multi-agent systems. Here, multi-modal logics are normally used to *specify* the behaviour of a multi-agent systems. Several formalisms have been designed for this task, most importantly logics for knowledge [5], logics for Belief-Desires-Intentions [15], deontic logics [14], etc. The usual approach in this line of work is to suggest a logic, in terms of its syntax and axiomatisation, to show that it captures the intuitive properties of the concept under investigation, and to show metalogical properties of the logic system, such as its completeness and decidability.

This is adequate for the task of specifying distributed systems but methodologies need to be developed for *verifying* that a system complies with a given specification. Several methods are employed to perform this task, traditionally theorem provers, and more recently model checkers.

It has been argued elsewhere [13,12] that a formalism for specifying properties of a system could make use of a deontic component. That can be used for example to distinguish in a precise and unambiguous way among properties that *should* hold in a system, properties that *may* hold in a system, properties that simply hold in a system. While deontic concepts are useful on their own, especially when paired to temporal operators, they become even more of importance in multi-agent systems when paired with informational properties such as their knowledge, beliefs, intentions, etc. The formalism of *deontic interpreted systems* was designed to make a step in that direction.

In deontic interpreted systems a semantics based on interpreted systems [5] is given to interpret a multi-modal language consisting of a family of operators $\{O_i\}$, representing correct functioning behaviour of agent $i$, a family of operators $\{K_i\}$ representing

T.D. Gedeon and L.C.C. Fung (eds)
AI 2003: Advances in Artificial Intelligence.
LNCS 2903, pp. 339–350, 2003.

the knowledge of agent $i$, and a family of operators $\{\widehat{K}_i^j\}$ representing the knowledge agent $i$ has under the assumption of correctness of agent $j$. It was argued in [13,12] that this set of operators could be useful to represent a number of key interesting scenarios, including communication and security examples.

A complete axiomatisation for deontic interpreted systems limited to the fragment of $\{O_i, K_i\}$, was also shown. This comprises the logics $S5_n$ for the modalities $K_i$ for knowledge and the logic $KD45_n^{i-j}$ for the modalities $O_i$ for correct functioning behaviour. While a Hilbert style axiomatisation is a theoretically valuable result, proving properties of particular examples by means of this is notoriously awkward. Automated technologies, such as the ones based on theorem provers or model checkers are called for. In this paper we define and investigate a tableaux system for the full logic above.

The remaining of this paper is organised as follows. In Section 2 we define the language of the logic, the semantics, and present its axiomatisation. In Section 3 we define a tableaux system for it. In Section 4 we present an example, the bit transmission problem, and we prove properties about it by means of the tableaux strategy. In Section 5 we wrap up and point to further work.

## 2 Deontic Interpreted Systems

We present here the main definitions for the notation we are going to use in this paper, as from [5,13]. Due to space consideration we are forced to assume working knowledge with some of the technical machinery presented there.

*Interpreted Systems* Consider $n$ agents in a system and $n$ non-empty sets $L_1, \ldots, L_n$ of local states, one for every agent of the system, and a set of states for the environment $L_E$. Elements of $L_i$ will be denoted by $l_1, l_1', l_2, l_2', \ldots$. Elements of $L_E$ will be denoted by $l_E, l_E', \ldots$.

A *system of global states for n agents S* is a non-empty subset of a Cartesian product $L_1 \times \cdots \times L_n \times L_E$. When $g = (l_1, \ldots, l_n, l_E)$ is a global state of a system $S$, $l_i(g)$ denotes the local state of agent $i$ in global state $g$. $l_E(g)$ denotes the local state of the environment in global state $g$. An *interpreted* system of global states is a pair $IS = (S, h)$ where $S$ is a system of global states and $h : S \to 2^P$ is an interpretation function for a set of propositional variables $P$. Systems of global states can be used to interpret epistemic modalities $K_i$, one for each agent.

$$(IS, g) \models K_i \varphi \text{ iff } \forall g' : l_i(g) = l_i(g') \Rightarrow (IS, g') \models \varphi.$$

Alternatively one can consider generated models $(S, \sim_1, \ldots, \sim_n, h)$ of the standard form, where the equivalence relations $\sim_i$ are defined on equivalence of local states, and then interpret modalities in the standard modal tradition (e.g. [3,10]). The resulting logic for modalities $K_i$ is $S5_n$; this models agents with complete introspection capabilities and veridical knowledge.

*Deontic interpreted systems* The notion of interpreted systems can be extended to incorporate the idea of correct functioning behaviour of some or all of the components [13].

Given $n$ agents and $n+1$ non-empty sets $G_E, G_1, \ldots, G_n$, a *deontic system of global states* is any system of global states defined on $L_E \supseteq G_E, \ldots, L_n \supseteq G_n$. $G_E$ is called the *set of green states for the environment*, and for any agent $i$, $G_i$ is called *the set of green states for agent i*. The complement of $G_E$ with respect to $L_E$ (respectively $G_i$ with respect to $L_i$) is called the *set of red states for the environment (respectively for agent i)*.

The terms 'green' and 'red' are chosen as neutral terms, to avoid overloading them with unintended readings and connotations. The term 'green' can be read as 'legal', 'acceptable', 'desirable', 'correct', depending on the context of a given application.

Deontic systems of global states are used to interpret modalities such as the following

$$(IS,g) \models O_i\,\varphi \text{ iff } \forall g' : l_i(g') \in G_i \Rightarrow (IS,g') \models \varphi.$$

$O_i\,\varphi$ is used to represent that $\varphi$ holds in all (global) states in which agent $i$ is functioning correctly. Again, one can consider generated models $(S, \sim_1, \ldots, \sim_n, R_1^O, \ldots, R_n^O, h)$, where the equivalence relations are defined as above and the relations $R_i^O$ are defined by $g R_i^O g'$ if $l_i(g') \in G_i$, with a standard modal logic interpretation for the operators $O_i$.

Knowledge can be modelled on deontic interpreted systems in the same way as on interpreted systems, and one can study various combinations of the modalities such as $K_i O_j$, $O_j K_i$, and others. Another concept of particular interest is knowledge that an agent $i$ has *on the assumption that the system (the environment, agent j, group of agents X) is functioning correctly*. We employ the (doubly relativised) modal operator $\widehat{K}_i^j$ for this notion, interpreted as follows:

$$(IS,g) \models \widehat{K}_i^j\,\varphi \text{ iff } \forall g' : l_i(g) = l_i(g') \text{ and } l_j(g') \in G_j \Rightarrow (IS,g') \models \varphi.$$

*An axiomatisation of deontic interpreted systems* The multi-modal language defined by $O_i, K_i$ is axiomatised by the logics $S5_n$ union $KD45_n^{i-j}$ where there are defined as follows:

The component $S5_n$ is defined by the smallest normal multi-modal logic (i.e., closed under the necessitation rule for $K_i$) satisfying the axioms $T$, 4, and 5 for each modal operator $K_i$. Semantically $S5_n$ is determined by the class of Kripke frames $(W, \sim_1, \ldots, \sim_n)$ where each $\sim_i$ is an equivalence relation.

The component $KD45_n^{i-j}$ is defined by the smallest normal multi-modal logic (i.e., closed under the necessitation rule for $O_i$) satisfying the axioms $D$, 4, 5 and $\neg O_i \neg\varphi \rightarrow O_j \neg O_i \neg\varphi$. for each pair of modal operators $O_i$, $O_j$. Semantically $KD45_n^{i-j}$ is determined by the class of serial, transitive and *i-j* Euclidean Kripke frames $(W, R_1^O, \ldots, R_n^O)$ where a frame is *i-j* Euclidean iff for all $w', w'', w''' \in W$ an for all $i, j$ such that $1 \leq i, j \leq n$, we have that $wR_i^O w'$ and $wR_j^O w''$ implies $wR_i^O w''$.

For the operator $\widehat{K}_i^j$, determined semantically by $\sim_i \cap R^{O_j}$, we do not have a complete axiomatisation. In this paper we provide a sound and complete tableuax system for it.

## 3 Tableaux for Deontic Interpreted Systems

In [1,9,2] a tableau-like proof system, called KEM, has been presented, and it has been proven to be able to cope with a wide variety of logics accepting possible world semantics. KEM is based on D'Agostino and Mondadori's [4] classical proof 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 generates models and checks them using a label scheme for bookkeeping states in interpreted systems. In [7,8,9] it has been shown how this formalism can be extended to handle various systems of multi-modal logic with interaction axioms. The mechanism KEM uses in manipulating labels is close to the possible world semantic constructions. In the following section we show how to adapt it to deal with deontic interpreted systems.

*Label Formalism* KEM uses *Labelled Formulas* ($L$-formulas for short), where an $L$-formula is an expression of the form $A : t$, where $A$ is a wff of the logic, and $t$ is a label. In the case of deontic interpreted systems we have a type of labels corresponding to various modalities for each agent; the set of atomic labels for $i$ ($\Phi^i$) is defined as follows:

$$\Phi^i = \Phi^i_O \cup \Phi^i_K \cup \Phi$$

Each set of atomic labels for the modalities is partitioned into the (non-empty) sets of variables and constants.

$$\Phi^i_O = V^i_O \cup C^i_O; \qquad \Phi^i_K = V^i_K \cup C^i_K; \qquad \Phi = V \cup C \text{ for any } j$$

where $V^i_O = \{O^i_1, O^i_2, \dots\}$, $C^i_O = \{o^i_1, o^i_2, \dots\}$, $V^i_K = \{K^i_1, K^i_2, \dots\}$, $C^i_K = \{k^i_1, k^i_2, \dots\}$, $V = \{_1, _2, \dots\}$, and $C = \{_1, _2, \dots\}$. Finally we add a sets of auxiliary unindexed atomic labels $\Phi^A = V^A \cup C^A$ – here $V^A = \{W_1, W_2, \dots\}$ and $C^A = \{w_1, w_2, \dots\}$–, that will be used in unifications and proofs. With $\Phi_C$ and $\Phi_V$ we denote, respectively, the set of constants and the set of variables.

The set of labels $\Im$ is then defined inductively as follows: a label is either (i) an element of the set $\Phi_C$, or (ii) an element of the set $\Phi_V$, or (iii) a path term $(s', s)$ where (iiia) $s' \in \Phi_C \cup \Phi_V$ and (iiib) $s \in \Phi_C$ or $s = (t', t)$ where $(t', t)$ is a label. From now on we shall use $t, s, r, \dots$ to denote arbitrary labels.

As an intuitive explanation, we may think of a label $t \in \Phi_C$ as denoting a world (a *given* one), and a label $t \in \Phi_V$ as denoting a set of worlds (*any* world) in some Kripke model. A label $s = (t', t)$ may be viewed as representing a path from $t$ to a (set of) world(s) $t'$ accessible from $t$ (i.e., from the world(s) denoted by $t$).

For any label $t = (s', s)$ we shall call $s'$ the *head* of $t$, $s$ the *body* of $t$, and denote them by $h(t)$ and $b(t)$ respectively. Notice that these notions are recursive (they correspond to projection functions): if $b(t)$ denotes the body of $t$, then $b(b(t))$ will denote the body of $b(t)$, and so on. We call each of $b(t)$, $b(b(t))$, etc., a *segment* of $t$. The length of a label $t$, $\ell(t)$, is the number of world-symbols in it, i.e., $\ell(t) = n \Leftrightarrow t \in \Im_n$. $s^n(t)$ will denote the segment of $t$ of length $n$ and we shall use $h^n(t)$ as an abbreviation for $h(s^n(t))$. Notice that $h(t) = h^{\ell(t)}(t)$.

For any label $t, \ell(t) > n$, we define the *counter-segment-n* of $t$, as follows (for $0 < n < k < \ell(t)$):

$$c^n(t) = h(t) \times (\cdots \times (h^k(t) \times (\cdots \times (h^{n+1}(t), w_0))))$$

where $w_0$ is a dummy label, i.e., a label not appearing in $t$ (the context in which such a notion occurs will tell us what $w_0$ stands for). The counter-segment-$n$ defines what remains of a given label after having identified the segment of length $n$ with a 'dummy' label $w_0$. The appropriate dummy label will be specified in the applications where such a notion is used. However, it can be viewed also as an independent atomic label.

So far we have provided definitions about the structure of the labels without regard of the elements they are made of. The following definitions will be concerned with the type of world symbols occurring in a label.

Let $t$ be a label and $t'$ an atomic label, in what follows we shall use $(t';t)$ as a notation for the label $(t',t)$ if $t' \neq h(t)$, or for $t$ otherwise.

We say that a label $t$ is *i-preferred* iff $h(t) \in \Phi^i$, and a label $t$ is *i-pure* iff each segment of $t$ of length $n > 1$ is $i$-preferred, and we shall use $\Im^i$ to denote the set of $i$-pure labels. A label is *i-compatible* iff each segment of $t$ of length $n > 1$ is either $i$-preferred or -preferred (for any $j$). A label $t$ is *-ground* iff every label of type $\Phi$ is a constant.

*Label Unifications* In the course of proofs labels are manipulated in a way closely related to the semantic of the logics under analysis. Labels are compared and matched using a specialised logic dependent unification mechanism. The notion that two labels $t$ and $s$ unify means that the intersection of their denotations is not empty and that we can "move" to such a set of worlds, i.e., to the result of their unification.

According to the semantics each modality is evaluated using an appropriate binary relation on the model and the model results from the combination of the relations. Similarly we provide an unification for each modality, the unification characterising it in the KEM formalism, then we combine them into a single unification for the whole logic. Every unification is built from a basic unification defined in terms of a substitution $\rho : \Im_1 \mapsto \Im$ such that:

$$\rho : \mathbf{1}_{\Phi_C},\ V_O^i \mapsto \Phi_O^i \text{ for any } j,\ V_K^i \mapsto \Phi_K^i \cup C^A \text{ for any } j,\ V \mapsto \Phi,\ V^C \mapsto \Im.$$

The above substitution is appropriate to characterise the logic without interaction among the modal operators. To capture them we have to introduce two specialized substitutions based on it.

$$\begin{aligned} \rho^O &: \ \rho \cup V_O^i \mapsto \Phi_O^i \cup \Phi^{ji} \text{ for any } j \\ \rho^K &: \ \rho \cup V_K^i \mapsto \Phi_K^i \cup \Phi \cup C^A \text{ for any } j \end{aligned}$$

Accordingly we have that two atomic ("world") labels $t$ and $s$ $\sigma$-unify iff there is a substitution $\rho$ such that $\rho(t) = \rho(s)$, with the constraint that a label in $V$ cannot unify with another variable. We shall use $[s,t]\rho$ both to indicate that there is a substitution $\rho$ for $s$ and $t$, and the result of the substitution. The notion of $\sigma$-unification (or label unification) is extended to the case of composite labels (path labels) as follows:

$$[i,j]\sigma = k \text{ iff } \exists \rho : h(k) = \rho(h(i)) = \rho(h(j)) \text{ and } b(k) = [b(i), b(j)]\sigma.$$

Clearly $\sigma$ is symmetric, i.e., $[i,j]\sigma$ iff $[j,i]\sigma$. Moreover this definition offers a flexible and powerful mechanism: it allows for an independent computation of the elements of the result of the unification, and variables can be freely renamed without affecting the result of a unification. Notice that a label $W_i$ $\sigma$-unifies with every label. The intuition here is that $W_i$ denotes the set of world in a Kripke model.

We are now ready to introduce the unifications corresponding to the modal operators at hand. The first unification is that for $O_i$.

$$[s,t]\sigma^O = ([h(s),h(t)]\rho^O,[h^1(s),h^1(t)]\sigma) \text{ iff } \min\{\ell(s),\ell(t)\} \geq 2 \text{ and } s,t \text{ are -ground}$$

Here we notice that the main structure is the structure for a $KD45$ modal operator $(\min\{l(s),l(t)\} \geq 2)$ [8,1]. However here we have that $O_i$ is defined globally over the green states of an interpreted systems, so we can ignore the intermediate steps with the proviso that there are no variables of type . Intuitively we can think of a variable of type as the intersection of the worlds accessible from a given world using $R^O_j$ and $\sim_i$; but in general such intersection can be empty, hence the proviso about the -groundness of the labels; moreover the restriction to $\rho^O$ prevents unwanted unifications of labels in $V^j_O$ and in $V^i_K$. According to the above definition we have that the labels $t = (O^j_1,(K^m_1,w_1))$ and $s = (o^j_1,w_1)$ $\sigma^O$-unify. In the same way $t$ $\sigma^O$-unifies with $({}_1,(k^n_1,w_1))$, but not with $(o^j_1,({}_1,w_1))$.

The following is the unification for $K_i$

$$[s,t]\sigma^K = ([h(s),h(t)]\rho^K;[h^1(s),h^1(t)]\sigma) \text{ iff } s \text{ and } t \text{ are -ground, and -compatible}$$

This is the condition for a unification corresponding to an equivalence relation [1,9]. The important point here are that all the atomic symbols should be compatible. A label such as ${}_n$ denotes a world in the intersection of the world accessible from a given world by $R^O_j$ and $\sim_i$. But this means that it is also one of the world accessible from $\sim_i$.

Let us consider the label $t = (K^i_2,({}_1,(K^i_1,w_1)))$. The result of the $\sigma^K$-unification of $t$ and $w_1$ is $w_1$; similarly the unification of $t$ and $s = ({}_2,w_1)$ is $s$. Notice that the label $t$ does not $\sigma^K$-unify with $(O^j_1,w_1)$.

$$\begin{aligned} [s,t]\sigma^* &= ([h(s),h(t)]\sigma,[h^1(s),h^1(t)]\sigma) \\ &\quad \text{iff } \ s,t \text{ are -compatible, and either } h^2(s) \text{ or } h^2(t) \text{ is -restricted} \end{aligned}$$

This unification is mainly designed for labels of type , and it corresponds to the unification for a $K45$ modal operator [8,1]. A label $({}_1,w_1)$ is intended to denote the equivalence class of type  associated to $w_1$. Since $\widehat{K}^j_i$ is not serial the equivalence class associated to a given world may be empty. However if we have that one of the labels in position 2 is a constant of type , then we are guaranteed that the equivalence class is not empty, and we can use the unification conditions for equivalence classes. Accordingly the labels $({}_1,(K^i_1,w_1))$ and $({}_1,w_1)$ $\sigma^*$-unify, and so do $({}_1,({}_1,w_1))$ and $({}_2,({}_2,w_1))$.

The above three unifications cover occurrences of sequences of compatible labels (relations). However we have to cover occurrences of interleaved labels. To this end we are going to define a recursive unification combining the unifications for the various operators. For convenience we introduce a unification corresponding to their simple combination. Hence $[s,t]\sigma^{\mathrm{DIS}}$ iff either $[s,t]\sigma$ or $[s,t]\sigma^*$ or $[s,t]\sigma^O$ or $[s,t]\sigma^K$. At this point the (recursive) unification for the logic DIS is defined as follows.

$$[s,t]\sigma_{\mathrm{DIS}} = \begin{cases} [s,t]\sigma^{\mathrm{DIS}} \\ [c^n(s), c^m(t)]\sigma^{\mathrm{DIS}} \end{cases}$$

where $w_0 = [s^n(s), s^m(t)]\sigma_{\mathrm{DIS}}$.

As we have seen the labels $t = (K^i_2, ({}_1, (K^i_1, w_1)))$ and $s = (O^j_1, w_1)$ neither $\sigma^O$-unify, nor $\sigma^K$-unify. However $[t,s]\sigma_{\mathrm{DIS}} = ({}_1, w_1)$, and so the labels $\sigma_{\mathrm{DIS}}$-unify. We can decompose the unification as follows: $[c^3(t), c^2(s)]\sigma^K$, where $c^3(t) = (K^i_2, w_0)$, $c^2(s) = w_0$, and $w_0 = [s^3(t), s]\sigma_{\mathrm{DIS}}$. $s^3(t) = ({}_1, (K^i_1, w_1))$.

Let us consider the following set of labels $\{t = (O^j_1, w_1), s = (K^i_1, w_1), r = ({}_1, w_1)\}$. Intuitively $t, s, r$ denote, respectively, the set of worlds accessible from $w_1$ by the relation $R^O_j$, the set of worlds accessible from $w_1$ by the relation $\sim_i$, and a world in $R^O_j \cap \sim_i$. In general labels such as $t$ and $s$ should not unify. The intersection of their denotations may be empty, but in cases like the present one $h(s)$ and $h(t)$ can be mapped to a common label (i.e., ${}_1$) but with different substitution, and this is not permitted in $\sigma_{\mathrm{DIS}}$. So we have to introduce a label-unification that takes care of context in which labels occur.

Let $\mathscr{L}$ be a set of labels (i.e., the labels occurring in a KEM-proof). Then $[s,t]\sigma^{\mathscr{L}}_{\mathrm{DIS}}$ iff

1. $[s,t]\sigma_{\mathrm{DIS}}$ or
2. $\exists k \in \mathscr{L}$, $\exists n, m \in Nat$ such that
   - $[s^n(s), k]\sigma^{\mathscr{L}}_{\mathrm{DIS}} = [s^m(t), k]\sigma^{\mathscr{L}}_{\mathrm{DIS}}$ and
   - $[c^n(s), c^m(t)]\sigma^{\mathscr{L}}_{\mathrm{DIS}}$ where $w_0 = [s^n(s), k]\sigma^{\mathscr{L}}_{\mathrm{DIS}}$

It is easy to verify that that the labels $s$ and $t$ described in the previous paragraph now $\sigma^{\mathscr{L}}_{\mathrm{DIS}}$ unify in the presence of the label $r$.

*Inference Rules* For the presentation of the inference rules of KEM we shall assume familiarity with Smullyan-Fitting $\alpha$, $\beta$, $\nu$, $\pi$ unifying notation [6].

$$\frac{\alpha : t}{\begin{array}{c}\alpha_1 : t \\ \alpha_2 : t\end{array}} \quad \frac{A \wedge B : t}{\begin{array}{c}A : t \\ B : t\end{array}} \quad \frac{\neg(A \vee B) : t}{\begin{array}{c}\neg A : t \\ \neg B : t\end{array}} \quad \frac{\neg(A \rightarrow B) : t}{\begin{array}{c}A : t \\ \neg B : t\end{array}} \qquad (\alpha)$$

The $\alpha$-rules are just the familiar linear branch-expansion rules of the tableau method. For the $\beta$-rules (formulas behaving disjunctively) we exemplify only the rules for implication.

$$\frac{\begin{array}{c}\beta : t \\ \beta^c_i : s\end{array} (i = 1,2)}{\beta_{3-i} : [t,s]\sigma^{\mathscr{L}}_{\mathrm{DIS}}} \quad \frac{\begin{array}{c}A \rightarrow B : t \\ A : s\end{array}}{B : [t,s]\sigma^{\mathscr{L}}_{\mathrm{DIS}}} \quad \frac{\begin{array}{c}A \rightarrow B : t \\ \neg B : s\end{array}}{\neg A : [t,s]\sigma^{\mathscr{L}}_{\mathrm{DIS}}} \qquad (\beta)$$

The $\beta$-rules are nothing but natural inference patterns such as Modus Ponens, Modus Tollens and Disjunctive syllogism generalised to the modal case. In order to apply such rules it is required that the labels of the premises unify and the label of the conclusion is the result of their unification.

$$\frac{\nu : t}{\nu_0 : (X_n, t)} \quad \frac{O_i A : t}{A : (O^i_n, t)} \quad \frac{K_i A : t}{A : (K^i_n, t)} \quad \frac{\widehat{K}^j_i A : t}{A : ({}_n, t)} \qquad (\nu)$$

where $O^i_n$, $K^i_n$, and ${}_n$ are new labels.

$$\frac{\pi : t}{\pi_0 : (x_n, t)} \quad \frac{\neg O_i A : t}{A : (o^i_n, t)} \quad \frac{\neg K_i A : t}{A : (k^i_n, t)} \quad \frac{\neg \widehat{K}^j_i A : t}{A : ({}_n, t)} \qquad (\pi)$$

where $o^i_n$, $k^i_n$, and ${}_n$ are new labels. $\nu$- and $\pi$- rules allow us to expand labels according to the intended semantics, where, with "new" we mean that the label does not occur previously in the tree.

$$\frac{}{A : t \quad | \quad \neg A : t} \qquad (PB)$$

The "Principle of Bivalence" represents 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). PB is a zero-premise inference rule, so in its unrestricted version can be applied whenever we like. However, we impose a restriction on its application. Then PB can be only applied w.r.t. immediate sub-formulas of unanalysed $\beta$-formulas, that is $\beta$ formulas for which we have no immediate sub-formulas with the appropriate labels in the branch (tree).

$$\begin{array}{c} A : t \\ \dfrac{\neg A : s}{\times} \end{array} [\text{ if } [t, s]\sigma^{\mathscr{L}}_{\text{DIS}}] \qquad (PNC)$$

The rule PNC (*Principle of Non-Contradiction*) states that two labelled formulas are $\sigma_L$-complementary when the two formulas are complementary and their labels $\sigma_L$-unify.

**Theorem 1.** $\vdash_{\text{KEM}} A \iff IS \models A$

We sketch only the proof. The main idea is to define a Kripke model where the possible worlds are the labels ($\mathscr{L}$) occurring in a KEM-proof for $A$, where the accessibility relations are defined as follows: (i) $t \sim_i s$ iff $[(K^i_0, t), s]\sigma^{\mathscr{L}}_{\text{DIS}}$; (ii) $tR^O_i s$ iff $[(O^i_0, t), s]\sigma^{\mathscr{L}}_{\text{DIS}}$; and (iii) $(t, s) \in \sim_i \cap R^O_j$ iff $[({}_0, t), s]\sigma^{\mathscr{L}}_{\text{DIS}}$. For (i) and (ii) it is immediate to verify that the frame induced from the above construction is a frame for DIS. The result for (iii) depends on the definition of the substitution $\rho$ where labels of type $V^j_O$ and $V^i_K$ can be mapped to labels in $C$. Hence $t \sim_i ({}_n, t)$ and $tR^O_j({}_n, t)$.

*Proof search* Let $\Gamma = \{X_1, \ldots, X_m\}$ be a set of formulas. Then $\mathscr{T}$ is a KEM-*tree for $\Gamma$* if there exists a finite sequence $(\mathscr{T}_1, \mathscr{T}_2, \ldots, \mathscr{T}_n)$ such that (i) $\mathscr{T}_1$ is a 1-branch tree consisting of $\{X_1 : t_1, \ldots, X_m : t_m\}$; (ii) $\mathscr{T}_n = \mathscr{T}$, and (iii) for each $i < n$, $\mathscr{T}_{i+1}$ results from $\mathscr{T}_i$ by an application of a rule of KEM. A branch $\tau$ of a KEM-tree $\mathscr{T}$ of $L$-formulas is said to be $\sigma_{\text{DIS}}$-*closed* if it ends with an application of *PNC*, open otherwise. As usual with tableau methods, a set $\Gamma$ of formulas is checked for consistency by constructing a KEM-tree for $\Gamma$. 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. Moreover we say that a formula $A$ is a KEM-*consequence of a set of formulas* $\Gamma = \{X_1, \ldots, X_n\}$ ($\Gamma \vdash_{\text{KEM}} A$) if a KEM-tree for $\{X_1 : t, \ldots, X_n : t, \neg A : s\}$ is closed, where $s \in C^A$, and $t \in V^A$. The intuition behind this definition is that $A$ is a consequence of $\Gamma$ when we take $\Gamma$ as a set of global assumptions [6], i.e., true in every world in a Kripke model.

We now describe a systematic procedure for KEM. First we define the following notions.

Given a branch $\tau$ of a KEM-tree, we shall call an $L$-formula $X : t$ *$E$-analysed in $\tau$* if either (i) $X$ is of type $\alpha$ and both $\alpha_1 : t$ and $\alpha_2 : t$ occur in $\tau$; or (ii) $X$ is of type $\beta$ and one of the following conditions is satisfied: (a) if $\beta_1^C : s$ occurs in $\tau$ and $[t,s]\sigma_{\text{DIS}}^{\mathscr{L}}$, then also $\beta_2 : [t,s]\sigma_{\text{DIS}}^{\mathscr{L}}$ occurs in $\tau$, (b) if $\beta_2^C : s$ occurs in $\tau$ and $[t,s]\sigma_{\text{DIS}}^{\mathscr{L}}$, then also $\beta_1 : [t,s]\sigma_{\text{DIS}}^{\mathscr{L}}$ occurs in $\tau$; or (iii) $X$ is of type $\nu$ and $\nu_0 : (m,t)$ occurs in $\tau$ for some $m \in \Phi_V$, of the appropriate type, not previously occurring in $\tau$, or (iv) $X$ is of type $\pi$ and $\pi_0 : (m,t)$ occurs in $\tau$ for some $m \in \Phi_C$, of the appropriate type, not previously occurring in $\tau$.

A branch $\tau$ of a KEM-tree is *$E$-completed* if every $L$-formula in it is $E$-analysed and it contains no complementary formulas which are not $\sigma_{\text{DIS}}^{\mathscr{L}}$-complementary. We shall say a branch $\tau$ of a KEM-tree *completed* if it is $E$-completed and all the $L$-formulas of type $\beta$ in it either are analysed or cannot be analysed. We shall call a KEM-tree *completed* if every branch is completed.

The following procedure starts from the 1-branch, 1-node tree consisting of $\{X_1 : t, \ldots, X_m : s\}$ and applies the inference rules until the resulting KEM-tree is either closed or completed. At each stage of proof search (i) we choose an open non completed branch $\tau$. If $\tau$ is not $E$-completed, then (ii) we apply the 1-premise rules until $\tau$ becomes $E$-completed. If the resulting branch $\tau'$ is neither closed nor completed, then (iii) we apply the 2-premise rules until $\tau$ becomes $E$-completed. If the resulting branch $\tau'$ is neither closed nor completed, then (iv) we choose an *LS*-formula of type $\beta$ which is not yet analysed in the branch and apply *PB* so that the resulting *LS*-formulas are $\beta_1 : t'$ and $\beta_1^C : ti'$ (or, equivalently $\beta_2 : ti'$ and $\beta_2^C : t'$), where $t = t'$ if $t$ is restricted (and already occurring when $h(t) \in \Phi_C$), otherwise $t'$ is obtained from $t$ by instantiating $h(t)$ to a constant not occurring in $t$; (v) ("Modal *PB*") if the branch is not $E$-completed nor closed, because of complementary formulas which are not $\sigma_{\text{DIS}}^{\mathscr{L}}$-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-trees. 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 $\alpha$-, $\nu$- and $\pi$-rule, then the $\beta$-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 DIS) 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 have been given in [8].

## 4 The bit transmission problem

The bit-transmission problem [5] involves two agents, a *sender S*, and a *receiver R*, communicating over a faulty communication channel. The channel may drop messages but will not flip the value of a bit being sent. *S* wants to communicate some information—the value of a bit for the sake of the example—to $R$. We would like to design a protocol that accomplishes this objective while minimising the use of the communication channel.

One protocol for achieving this is as follows. $S$ immediately starts sending the bit to $R$, and continues to do so until it receives an acknowledgement from $R$. $R$ does nothing until it receives the bit; from then on it sends acknowledgements of receipt to $S$. $S$ stops sending the bit to $R$ when it receives an acknowledgement. Note that $R$ will continue sending acknowledgements even after $S$ has received its acknowledgement. Intuitively $S$ will know for sure that the bit has been received by $R$ when it gets an acknowledgement from $R$. $R$, on the other hand, will never be able to know whether its acknowledgement has been received since $S$ does not answer the acknowledgement.

We assume *fairness* ([5], p.164) for the communication channel: every message that is repeatedly sent in the run is eventually delivered.

What we would like to do is to check mechanically that the protocol above guarantees that when sender receives the acknowledgement it then knows (in the information-theoretic sense defined in Section 2) that the receiver knows the value of the bit. In order to do this, first we model the scenario in the interpreted systems paradigm.

An interesting scenario arises when we assume that the agents may not behave as they are supposed to. For example, the receiver may not send an acknowledgement message when it receives a bit ([12]). We deal with this case by considering a new protocol which extends the original one.

*Bit transmission problem — no violations* First of all we give an axiomatisation of the bit transmission problem (BTP). For a detailed discussion of the BTP in the framework of Deontic Interpreted Systems see [13,12].

- Sender
  (S1) $\mathbf{recack} \to K_S\,\mathbf{recack}$
  (S2) $(\mathbf{bit} = n) \to K_S\,(\mathbf{bit} = n)$, for $n = 1,2$
- Receiver
  (R1) $\mathbf{recbit} \wedge (\mathbf{bit} = n) \to K_R\,(\mathbf{bit} = n)$, for $n = 1,2$
- Communication
  (C1) $\mathbf{recack} \to \mathbf{recbit}$

We can derive the following key property

$$\mathbf{recack} \wedge (\mathbf{bit} = n) \to K_S K_R\,(\mathbf{bit} = n) \qquad \text{for } n = 1,2$$

So, if an acknowledgement is received by the sender $S$, then $S$ is sure that receiver $R$ knows the value of the bit: although the communication channel is potentially faulty, if messages do manage to travel back and forth between the sender and receiver the protocol is strong enough to eliminate any uncertainty in the communication. Let us examine the KEM-proof for this property

| | | |
|---|---|---|
| 1. | $\mathbf{recack} \rightarrow K_S\,\mathbf{recack} : W_1$ | |
| 2. | $(\mathbf{bit} = n) \rightarrow K_S\,(\mathbf{bit} = n) : W_1$ | |
| 3. | $\mathbf{recbit} \wedge (\mathbf{bit} = n) \rightarrow K_R\,(\mathbf{bit} = n) : W_1$ | |
| 4. | $\mathbf{recack} \rightarrow \mathbf{recbit} : W_1$ | |
| 5. | $\neg(\mathbf{recack} \wedge (\mathbf{bit} = n) \rightarrow K_S K_R\,(\mathbf{bit} = n)) : w_1$ | |
| 6. | $\mathbf{recack} : w_1$ | $5\alpha$ |
| 7. | $\mathbf{bit} = n : w_1$ | $5\alpha$ |
| 8. | $\neg K_S K_R\,(\mathbf{bit} = n)$ | $5\alpha$ |
| 9. | $\neg K_R\,(\mathbf{bit} = n) : (s_1, w_1)$ | $8\pi$ |
| 10. | $K_S\,\mathbf{recack} : w_1$ | $1,6\beta$ |
| 11. | $K_S\,(\mathbf{bit} = n) : w_1$ | $2,7\beta$ |
| 12. | $\mathbf{recack} : (S_1, w_1)$ | $10\nu$ |
| 13. | $\mathbf{bit} = n : (S_2, w_1)$ | $11\nu$ |
| 14. | $\neg(\mathbf{recbit} \wedge (\mathbf{bit} = n)) : (s_1, w_1)$ | $3,9\beta$ |
| 15. | $\neg\mathbf{recbit} : (s_1, w_1)$ | $13,14\beta$ |
| 16. | $\mathbf{recbit} : (S_1, w_1)$ | $4,12\beta$ |
| 17. | $\times$ | $15,16 PNC$ |

*Bit transmission problem — violation by the receiver* Now we admit the possibility that the receiver, in violation of the protocol, may send acknowledgements without having received the bit. In this version, the axiom $(C1)$ does not hold. It is replaced by

$$O_R(\,\mathbf{recack} \rightarrow \mathbf{recbit}\,) \qquad (C1*)$$

which represents what holds when $R$ is working correctly according to the protocol. All other parts of the formalisation are unchanged.

A particular form of knowledge still holds. Intuitively if $S$ makes the assumption of $R$'s correct functioning behaviour, then, upon receipt of an acknowledgement, it would make sense for $S$ to assume that $R$ does know the value of the bit. To model this intuition we use the operator $\widehat{K}_i^j$ "knowledge under the assumption of correct behaviour".

We now derive, given $(C1*)$ instead of $(C1)$

$$\mathbf{recack} \wedge (\mathbf{bit} = n) \rightarrow \widehat{K}_S^R K_R\,(\mathbf{bit} = n) \qquad \text{for } n = 1,2$$

We give a KEM-proof for it.

| | | |
|---|---|---|
| 1. | $\mathbf{recack} \rightarrow K_S\,\mathbf{recack} : W_1$ | |
| 2. | $(\mathbf{bit} = n) \rightarrow K_S\,(\mathbf{bit} = n) : W_1$ | |
| 3. | $\mathbf{recbit} \wedge (\mathbf{bit} = n) \rightarrow K_R\,(\mathbf{bit} = n) : W_1$ | |
| 4. | $\mathbf{recack} \rightarrow \mathbf{recbit} : W_1$ | |
| 5. | $\neg(\mathbf{recack} \wedge (\mathbf{bit} = n) \rightarrow \widehat{K}_S^R K_R\,(\mathbf{bit} = n)) : w_1$ | |
| 6. | $\mathbf{recack} : w_1$ | $5\alpha$ |
| 7. | $\mathbf{bit} = n : w_1$ | $5\alpha$ |
| 8. | $\neg \widehat{K}_S^R K_R\,(\mathbf{bit} = n)$ | $5\alpha$ |
| 9. | $\mathbf{recack} \rightarrow \mathbf{recbit} : (O_1^R, w_1)$ | $4\nu$ |
| 10. | $K_S\,\mathbf{recack} : w_1$ | $1,6\beta$ |
| 11. | $K_S\,(\mathbf{bit} = n) : w_1$ | $2,7\beta$ |
| 13. | $\neg K_R\,(\mathbf{bit} = n) : (sr_1, w_1)$ | $8\pi$ |
| 14. | $\neg(\mathbf{recbit} \wedge (\mathbf{bit} = n)) : (sr_1, w_1)$ | $3,13\beta$ |
| 15. | $\mathbf{recack} : (S_1, w_1)$ | $10\nu$ |
| 16. | $\mathbf{bit} = n : (S_2, w_1)$ | $11\nu$ |
| 17. | $\neg\mathbf{recbit} : (sr_1, w_1)$ | $14,16\beta$ |
| 18. | $\mathbf{recbit} : (sr_1, w_1)$ | $9,15\beta$ |
| 19. | $\times$ | $17,18 PNC$ |

The only step that deserves some attention is step 18. This step is the consequence of a $\beta$-rule on 9 and 15. The labels of the relevant formulas are $(O_1^R, w_1)$ and $(S_1, w_1)$. Normally such labels do not unify, and the $\beta$-rule would not be applicable. However, thanks to the presence of $(sr_1, w_1)$ in the tree, the labels of 9 and 15 do $\sigma_{\text{DIS}}^{\mathscr{L}}$-unify.

## 5 Conclusions

In this paper we presented a tableaux-based system for proving properties of a system whose properties can be expressed in an epistemic-deontic language. The tableaux system was used to prove properties about variations of the bit transmission problem, a widely explored protocol to reason about information exchange in communication protocols. The results obtained confirmed results obtained by model checking techniques presented elsewhere [11]. Further work involves an implementation of this method so that experimental results based on larger scenarios can be evaluated.

## References

1. A. Artosi, P. Benassi, G. Governatori, and A. Rotolo. Shakespearian modal logic: A labelled treatment of modal identity. In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyaschev, editors, *Advances in Modal Logic. Volume 1*, pages 1–21. CSLI Publications, Stanford, 1998.
2. A. Artosi, G. Governatori, and A. Rotolo. Labelled tableaux for non-monotonic reasoning: Cumulative consequence relations. *Journal of Logic and Computation*, 12(6):1027–1060, December 2002.
3. B. Chellas. *Modal Logic: An Introduction*. Cambridge University Press, Cambridge, 1980.
4. M. D'Agostino and M. Mondadori. The taming of the cut. *Journal of Logic and Computation*, 4:285–319, 1994.
5. R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge, 1995.
6. M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*. Reidel, Dordrecht, 1983.
7. D. M. Gabbay and G. Governatori. Fibred modal tableaux. In D. Basin, M. D'Agostino, D. Gabbay, S. Matthews, and L. Viganó, editors, *Labelled Deduction*, volume 17 of *Applied Logic Series*, pages 163–194. Kluwer, Dordrecht, 2000.
8. G. Governatori. Labelled tableaux for multi-modal logics. In P. Baumgartner, R. Hähnle, and J. Posegga, editors, *Theorem Proving with Analytic Tableaux and Related Methods*, volume 918 of *LNAI*, pages 79–94, Berlin, 1995. Springer-Verlag.
9. G. Governatori. *Un modello formale per il ragionamento giuridico*. PhD thesis, CIRFID, University of Bologna, Bologna, 1997.
10. G. E. Hughes and M. J. Cresswell. *A New Introduction to Modal Logic*. Routledge, New York, 1996.
11. A. Lomuscio, F. Raimondi, and M. Sergot. Towards model checking interpreted systems. In *Proceedings of Mochart — First International Workshop on Model Checking and Artificial Intelligence*, 2002.
12. A. Lomuscio and M. Sergot. Violation, error recovery, and enforcement in the bit transmission problem. In *Proceedings of DEON'02*, London, May 2002.
13. A. Lomuscio and M. Sergot. Deontic interpreted systems. *Studia Logica*, 75, 2003.
14. J.-J. C. Meyer and W. Hoek. *Epistemic Logic for AI and Computer Science*, volume 41 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1995.
15. A. S. Rao and M. P. Georgeff. Decision procedures for BDI logics. *Journal of Logic and Computation*, 8(3):293–343, June 1998.