
Labelled modal tableaux

GUIDO GOVERNATORI

ABSTRACT. Labelled tableaux are extensions of semantic tableaux with annotations (labels, indices) whose main function is to enrich the modal object language with semantic elements. This paper consists of three parts. In the first part we consider some options for labels: simple constant labels vs labels with free variables, logic depended inference rules vs labels manipulation based on a label algebra. In the second and third part we concentrate on a particular labelled tableaux system called KEM using free variable and a specialised label algebra. Specifically in the second part we show how labelled tableaux (KEM) can account for different types of logics (e.g., non-normal modal logics and conditional logics). In the third and final part we investigate the relative complexity of labelled tableaux systems and we show that the uses of KEM's label algebra can lead to speed up on proofs.

Keywords: labelled tableaux, non-normal modal logic, conditional logic, relative complexity

1 Introduction

Since the seminal work by Fitch [12] labels have been widely used in modal logic to simulate possible world semantics in the proof theory to improve, simplify and speed up proofs. Usually the main function of labels is to “import” or simulate semantic structures in the object language. Accordingly, in semantic based proof methods (cf., among others, [32]), labels represent possible worlds and accessibility relations (using sequences of atomic labels) in Kripke models.

Semantic tableaux (cf., [35]) is one of the most common form of semantic based proof procedures and, we believe, it offers the best format for the use of labels. The basic idea is to supplement the object language with a label language and a label algebra. The basic entities of labelled deductions are labelled formulas, i.e., expressions of the form $A : x$, where x is a label drawn from the label language, and A (the declarative unit) is a well-formed formula of the logic at hand (cf., [14]). Intuitively the meaning of a labelled formula such as $A : x$ is that the declarative unit (A) is true at the world(s) denoted by the label x .

The structure of the paper is as follows: in Section 2 we introduce the formalism and we discuss some options to combine labels and tableaux. In Section 3 we examine how to extend labelled tableaux to cover other logics having possible world semantics, namely non-normal modal logic (modal

logic where necessitation does not hold) and conditional logics. Finally in Section 4 we investigate the relative complexity of two labelled tableaux systems presented in Section 2 and we discuss the general issue for the methodology to compare this kind of proof systems.

2 Labelled modal tableaux

As is well known semantic tableaux calculus is a refutation proof method¹. Therefore a proof of A is a failed attempt to provide a model for $\neg A$. A tableaux for a formula A is a (binary) tree whose root is $A : i_0$ where i_0 is the initial label, and the nodes are derived from previous nodes according to the inference rules of the system. A branch is *closed* iff it contains a pair of complementary formulas (the notion of complementary formulas may vary from system to system), otherwise it is *open*; a tree is *closed* iff every branch in it is closed. Finally a proof of A is a closed tree with root $\neg A : i_0$. A tree is *complete* iff every rule that can be applied has been applied.

A labelled modal tableaux systems is defined by the structure of labels and the inference rules for analysing the formulas. In most systems (new) labelled formulas are generated from previous formulas using inference rules that closely resemble the semantic evaluation of the premises. Given the semantic conditions, it is indeed possible that the conclusion of a premise holds in a set of possible worlds instead of a single worlds; for example, just consider the semantic clause for $\Box A$ which requires A to be true in all worlds accessible from the world where $\Box A$ is currently evaluated. We have two alternative ways for representing such conclusions using labels:

1. we can use ground labels and generate all possible/relevant instances of such worlds;
2. we can use a label with a free-variable, where the variable is intended to range over such worlds.

The second issue we have to tackle is how to represent the structure of the model. Different modal logics determine different structures on Kripke frames (or better, possible world frames in the general case). Again we have multiple options.

1. We can define logic dependant inference rules assigning formulas to existing labels or generating new labels and formulas (see Section 2.2 for examples).
2. We can use a single logic neutral inference rule for a modal operator, make use of an explicit representation of the relevant semantics structures (e.g., the accessibility relation) and, then use an external mechanism to resolve and compute the semantic structure and propagate the formulas to the labels accordingly. For example for a transitive

¹For a comprehensive account of tableaux for modal logic we refer the reader to [17].

logic one can have the following rules (see, among others [5, 16])

$$\frac{\frac{\diamond A : w}{A : w'} w' \text{ new on the branch}}{wRw'}}{\frac{wRw', w'Rw''}{wRw''}}$$

3. For the last alternative, the option we are going to investigate in the rest of the paper, we follow the previous case in so far as each modal operator has a single inference rule common to all logics and the various logics are differentiated by logic specific operations that manipulate the labels. In other words every logic has its own label ‘algebra’.

In the rest of the section we are going to introduce a two label tableaux systems, both using free variables, the first SST adopts the first strategy, and the second KEM adopts the third strategy.

2.1 Label formalism

In this section we present the KEM label formalism. The formalism will also be used for SST. In fact most of the differences between the formalisms of the two systems are just notational ones, and the differences that are not notational are not relevant for the present investigation.

KEM has two basic kinds of atomic labels: variables and constants. Formally, let $\Phi_C = \{w_1, w_2, \dots\}$ and $\Phi_V = \{W_1, W_2, \dots\}$ be two arbitrary sets of *atomic labels*: the set of *constant world-symbols* (or simply *constants*) and the set of *variable world-symbols* (or simply *variables*). A *label* is then an element of the set of labels \mathfrak{S} defined as follows:

DEFINITION 1. $\mathfrak{S} = \bigcup_{1 \leq p} \mathfrak{S}_p$ where \mathfrak{S}_p is:

$$\begin{aligned} \mathfrak{S}_1 &= \Phi_C \cup \Phi_V \\ \mathfrak{S}_2 &= \mathfrak{S}_1 \times \Phi_C \\ \mathfrak{S}_{n+1} &= \mathfrak{S}_1 \times \mathfrak{S}_n, \quad n > 1. \end{aligned}$$

Thus, a label i is either a variable or a constant or a “structured” sequence of atomic labels. For a structured label $i = (k', k)$ we have the following cases: (i) k' is an atomic world-symbol and (ii) $k \in \Phi_C$ or $k = (m', m)$ where (m', m) is a label. As we have alluded to in the previous section, we may think of constant and variable world-symbols as denoting respectively worlds and sets of worlds in a standard Kripke setting. A label of the form (k', k) is called a “world-path”. For instance, the label (W_1, w_1) represents a path from w_1 to the set W_1 of worlds accessible from w_1 ; $(w_2, (W_1, w_1))$ represents a path which takes us to a world w_2 accessible by any world accessible from w_1 (i.e., accessible by the sub-path (W_1, w_1)) according to the appropriate accessibility relation. Thus a label of the form (k', k) is “structurally” designed to record information about the accessibility relation when we move from a label (a world or a set of worlds) to another

label. We define the length of a label i , $\ell(i)$, as the number of atomic labels in i . From now on we shall use i, j, k, \dots to denote arbitrary labels.

DEFINITION 2. 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.

The notions of body and head are obviously recursive (they can be defined as projection functions), and allow us to identify any sub-label of a given label; thus, 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. 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); $h(s(i))$ will denote the head of $s(i)$. With $s^n(i)$ we will denote the segment of i of length n , i.e., $s^n(i) = s(i)$ such that $\ell(s(i)) = n$. We shall use $h^n(i)$ as an abbreviation for $h(s^n(i))$. A label is *restricted* if its head is a constant, and *unrestricted* otherwise.

DEFINITION 3. For any label i , $\ell(i) \geq n$, we define the *countersegment- n* of i , as follows:

$$c^n(i) = h(i) \times (\dots \times (h^k(i) \times (\dots \times (h^{n+1}(i), w_0)))) \text{ for } n < k < \ell(i)$$

where w_0 is a dummy label, i.e., a label not appearing in i (the context in which such a notion is applied will tell us what w_0 stands for).

If $n = \ell(i)$ we have that $c^n(i) = w_0$, and $s^n(i) = i$.

EXAMPLE 4. If $i = (w_4, (W_3, (w_3, (W_2, w_1))))$, then $\ell(i) = 5$, $h^3(i) = w_3$, $s^3(i) = (w_3, (W_2, w_1))$, and its countersegment-3 is $c^3(i) = (w_4, (W_3, w_0))$; intuitively $c^n(i)$, is what remains of i after deleting $s^n(i)$.

To clarify the notion of countersegment, which will be used frequently in this work, we present, in the following table the list of the segments of i in the left-hand column and the relative countersegments in the right-hand column.

$s^1(i) = w_1$	$c^1(i) = (w_4, (W_3, (w_3, (W_2, w_0))))$
$s^2(i) = (W_2, w_1)$	$c^2(i) = (w_4, (W_3, (w_3, w_0)))$
$s^3(i) = (w_3, (W_2, w_1))$	$c^3(i) = (w_4, (W_3, w_0))$
$s^4(i) = (W_3, (w_3, (W_2, w_1)))$	$c^4(i) = (w_4, w_0)$
$s^5(i) = i$	$c^5(i) = w_0$

The dummy label w_0 is considered as an atomic label, and it is used to encapsulated complex labels into an atomic one.

We are now ready to introduce the two labelled tableaux systems.

2.2 Single step tableaux (SST)

Single Step Tableaux [27] originate from and add modularity to Fitting's prefix tableaux [13]. The free-variable version we shall focus on here has been proposed by Beckert and Goré [7].

The basic idea of SST is that (modal) formulas are used to move the evaluation point to the “neighbourhood” of the labels they are associated with, that is, each time we are allowed to move only one step apart. In other words the information that can be extracted from a formula is propagated only to the labels that the current label extends immediately or are an immediate extension of the current label.

SST has the following inference rules. For the presentation of the inference rules of SST, and subsequently of KEM we shall assume familiarity with Smullyan-Fitting α , β , ν , π unifying notation [13]. For the propositional part we give only the rules for \wedge .

$$(\alpha\text{-rules}) \quad \frac{A \wedge B : i}{A : i} \\ B : i$$

$$(\beta\text{-rules}) \quad \frac{\neg(A \wedge B) : i}{\neg A : i \quad | \quad \neg B : i}$$

$$(\pi\text{-rules}) \quad \frac{\diamond A : i}{A : (w_{[\pi]}, i)} \quad \frac{\neg \square A : i}{\neg A : (w_{[\pi]}, i)}$$

where $[\cdot]$ is an arbitrary but fixed bijection from the set of formulas to N

$$(\nu_D\text{-rules}) \quad \frac{\square A : i}{A : (W_n, i)} \quad \frac{\neg \diamond A : i}{\neg A : (W_n, i)}$$

$$(\nu_T\text{-rules}) \quad \frac{\square A : i}{A : i} \quad \frac{\neg \diamond A : i}{A : i}$$

$$(\nu_4\text{-rules}) \quad \frac{\square A : i}{\square A : (W_n, i)} \quad \frac{\neg \diamond A : i}{\neg \diamond A : (W_n, i)}$$

$$(\nu_{4^*}\text{-rules}) \quad \frac{\square A : i}{\square A : b(i)} \quad \frac{\neg \diamond A : i}{\neg \diamond A : b(i)}$$

$$(\nu_B\text{-rules}) \quad \frac{\square A : i}{A : b(i)} \quad \frac{\neg \diamond A : i}{\neg A : b(i)}$$

$$(\nu_5\text{-rules}) \quad \frac{\square A : i}{\square A : (W_n, h^1(i))} \quad \frac{\neg \diamond A : i}{\neg \diamond A : (W_n, h^1(i))}$$

In the above rules W_n must be a new label, i.e., a label that does not previously occur in the tableaux. The ν_D -rules are also known as the ν_K -rules. SST has an additional mechanism to keep track of which labels are

denoting, and the mechanism essentially differentiates between serial and non-serial logics (see [7] for the details).

The α -, π -, and ν_D -rules are common to KEM and SST and the β -rules are the usual branching rules of tableau methods. The ν_T -rules are the rules specific to reflexive logics; the ν_4 -rules for the transitive logics; the ν_{4^*} -rules and ν_5 -rules for Euclidean logics; and finally the ν_B -rules are the specific rules for symmetric logics. The intuition behind these logic is that we ‘move’ a formula to a labels that is one single step (using the accessibility relation) from the labels the formula in the antecedent is associated to. Thus for the ν_B -rules the idea is that symmetry allows us to travel backward in the accessibility relation. For the full list of characterisation of the fifteen basic modal logics see [7].

The tableaux system for a logic is given by a combination of the above rules. For example SST for S4 has the following (modal) rules: π , ν_D , ν_T and ν_4 ; and the symmetric and serial logic DB is characterised by the rules π , ν_D and ν_B . The main consequence of n sets of ν -rule is that every time we have a formula of type ν we have to introduce n new labelled formulas.

We say that two labelled formulas $A : i$ and $B : j$ are complementary in SST when $B = \neg A$ and there exists a substitution ρ which is a unifier of i and j .

Let L be one of the fifteen basic modal logics. With $\vdash_{\text{SST}(L)} A$ we mean that there is a close SST-tree for $\neg A : w_1$; or, in other words, that SST proves that A is a theorem of L .

THEOREM 5. $\vdash_{\text{SST}(L)} A \text{ iff } \models_L A$.

For the proof and for detailed accounts of SST see [7, 29].

2.3 KEM

KEM (see [18, 1, 20]) is a labelled analytic proof system based on a combination of tableau and natural deduction inference rules which allows for a suitably restricted (“analytic”) application of the cut rule and a specialised, yet modular, unification mechanism for the labels.

Unifications

In the course of proofs labels are manipulated in a way closely related to the semantics of the logic under analysis. Labels are compared and matched using a specialised logic dependent unification mechanism. The notion of two labels i and j being unifiable means that the intersection of their denotations is not empty and that we can “move” to such a set of worlds through the path corresponding to the result of the unification of the two labels.

The definition of the unification appropriate for the various logics (or logic unification) is carried out in several steps with the help of several auxiliary notions of unification.

First we have to provide the foundation of our unification (σ -unification). The basic unification is defined, as usual, in terms of a substitution, then we use the basic unification to define the unifications corresponding to the various modal axioms (axiom unifications); in the same way a modal logic

is obtained by combining several axioms we combine the axiom unifications in combined unification. Finally we apply, in a recursive way, the combined unification to define the unification for the logic (logic unification).

Before presenting the formal machinery for the various unifications we have to give the notation used for them. Let L be a modal logic, and A_1, \dots, A_n be the axioms of L . With σ^{A_i} we denote the unification for the axiom A_i ; with $\sigma^{A_1 \dots A_n}$ the unification obtained from the combination of the σ^{A_i} -unifications; and with σ_L the unification for the logic L . Given two labels i and j and a unification σ^* we shall use $[i, j]\sigma^*$ to denote both the result of the σ^* -unification of i and j , and the fact that i and j σ^* -unify.

DEFINITION 6. A substitution is a mapping $\rho : \mathfrak{S}_1 \rightarrow \mathfrak{S}_1$ such that

$$\rho(i) = \begin{cases} i & i \in \Phi_C \\ j & \text{otherwise} \end{cases}$$

Accordingly we have that two atomic (“world”) labels i and j σ -unify iff there is a substitution ρ such that $\rho(i) = \rho(j)$. The notion of σ -unification (or label unification) is extended to the case of composite labels (path labels) as follows:

DEFINITION 7. Let $i, j \in \mathfrak{S}$

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

Clearly σ is symmetric, i.e., $[i, j]\sigma$ iff $[j, i]\sigma$. Moreover this definition offers a flexible and powerful mechanism: in Section 3.1 we show that different classes of modal logics (in particular classes of non-normal modal logics such as regular and monotonic modal logics) are determined by conditions on the underlying substitution but the axiom unifications can be left unchanged. At the same time 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.

Let A be a modal axiom. In general the “axiom” unification can be described as follows:

$$[i, j]\sigma^A \iff [f_A(i), g_A(j)]\sigma \text{ and } C^A$$

where f_A and g_A are given logic-dependent functions from labels to labels and C^A is a set of constraints.

We now give the axiom unifications for the axioms for the fifteen basic normal modal logics².

DEFINITION 8. Let $i, j \in \mathfrak{S}$

$$[i, j]\sigma^K = [i, j]\sigma \text{ if at least one of } i \text{ and } j \text{ is restricted, and} \\ \forall n \leq \ell(i), [s^n(i), s^n(j)]\sigma^K$$

²For the full details and explanations of the unifications, see [15, 20].

DEFINITION 9. Let $i, j \in \mathfrak{S}$. $[i, j]\sigma^D = [i, j]\sigma$

EXAMPLE 10. For the difference between σ^K and σ^D , let us consider first the labels

$$i = (w_3, (W_1, w_1)) \quad j = (W_2, (w_2, w_1))$$

Obviously i and j σ^K - and σ^D -unify on $(w_3, (w_2, w_1))$ with the substitution

$$\rho : W_1 \mapsto w_2, \quad W_2 \mapsto w_3$$

On the other hand the labels

$$i = (w_2, (W_1, w_1)) \quad j = (W_2, (W_1, w_1))$$

σ^D - but not σ^K -unify. This is due to the fact that both $s^2(t)$ and $s^2(s)$ are variables, while in the definition of σ^K it is required that at least one of them is a constant. The reason for this condition on σ^K is that the interpretation of W_1 is the set of worlds accessible from w_1 , but such a set may be empty so the denotation of W_1 would be empty; this is not the case with σ^D since the corresponding accessibility relation is serial, so W_1 cannot be empty.

To simplify the remaining definition of axiom unifications we introduce the following notation: given two labels i and j (of different length), we use \bar{i} to denote the longest of the two and \bar{j} for the shortest of the two.

DEFINITION 11. Let $i, j \in \mathfrak{S}$

$$[i, j]\sigma^T = [s^{\ell(\bar{j})}(i), j]\sigma \text{ if } \forall n \geq \ell(\bar{j}), [h^n(\bar{i}), h(\bar{j})]\sigma = [h(\bar{i}), h(\bar{j})]\sigma$$

DEFINITION 12. Let $i, j \in \mathfrak{S}$

$$[i, j]\sigma^4 = c^{\ell(\bar{j})}(\bar{i}) \text{ if } h(\bar{j}) \in \Phi_V \text{ and } w_0 = [\bar{j}, s^{\ell(\bar{j})}(\bar{i})]\sigma$$

DEFINITION 13. Let $i, j \in \mathfrak{S}$

$$[i, j]\sigma^5 = \begin{cases} ([h(t), h(s)]\sigma; c^1(s^2(t))) & \text{if } \ell(t) > 2, \ell(s) > 1, h(t) \in \Phi_V, \text{ or} \\ & h(t) = h(s) \in \Phi_C \\ [t, s]\sigma & \text{if } \ell(t) = \ell(s) = 2 \\ ([t, h(s)]\sigma; c^1(s^2(s))) & \text{if } \ell(s) > 2, \ell(t) > 1, h(s) \in \Phi_V, \text{ or} \\ & h(t) = h(s) \in \Phi_C \end{cases}$$

where $w_0 = [s^1(t), s^1(s)]\sigma$.

For the unification for axiom B or σ^B -unification we first have to introduce some auxiliary definitions and constructions. For a label i and an integer n we define the sets of restricted and unrestricted segments longer than n .

$$\Phi_C^{i,n} = \{s^m(i) : m > n \text{ and } h^m(i) \in \Phi_C\}$$

$$\Phi_V^{i,n} = \{s^m(i) : m > n \text{ and } h^m(i) \in \Phi_V\}$$

We can give now the key notion to be used in the definition of the σ^B -unification.

DEFINITION 14. Given a label i and an integer n , i has the *bmorphism property for n* iff there is a morphism $\eta : \Phi_C^{i,n} \mapsto \Phi_V^{i,n}$ such that

1. η is injective, and
2. if $\eta(s^k(i)) = s^l(i)$, then $k < l$.

We are now ready to give the definition of σ^B -unification.

DEFINITION 15. Let $i, j \in \mathfrak{S}$

$$[i, j]\sigma^B = [s^{\ell(\bar{j})}(\bar{i}), \bar{j}]\sigma \text{ iff } \begin{array}{l} (1) \ell(\bar{i}) - \ell(\bar{j}) = 2n, (n > 0), \text{ and} \\ (2) \bar{i} \text{ has the bmorphism property for } \ell(\bar{j}). \end{array}$$

According to the above definition labels like

$$(W_1, (w_2, w_1)) \qquad w_1$$

provide a simple instance of this unification. Intuitively W_1 denotes the set of worlds accessible from w_2 , but, since w_2 is accessible from w_1 , so, by symmetry, w_1 is one of the world accessible from w_2 .

The key idea of σ^B -unification is to match world symbols laying an even number of steps apart, where the number of steps is determined by the sequences of variable and constants. In the above example the head of the first label is a variable we can go back by two steps. In general every constant must be compensated for by a variable following it.

EXAMPLE 16. Let us consider the labels

$$(1) \quad i = (W_3, (W_2, (w_3, (W_1, (w_2, w_1)))))) \qquad j = (W_4, w_1)$$

The labels i and j σ^B -unify since the difference of the lengths of the two labels is even ($\ell(i) = 6$ and $\ell(j) = 2$); moreover $s^2(i) = (w_2, w_1)$ and j σ -unify, and the restricted segment $s^4(i)$ can be mapped to the unrestricted segment $s^5(i)$.

In similar way the labels

$$(2) \quad i = (W_3, (w_3, (W_2, (w_2, w_1)))) \qquad j = w_1$$

σ^B -unify, with the injective morphism η that maps $s^2(i)$ to $s^3(i)$ and $s^4(i)$ to $s^5(i)$. On the other hand the labels

$$(3) \quad i = (w_3, (W_1, (w_2, w_1))) \qquad j = (W_2, w_1)$$

do not σ^B -unify since there is no (injective) morphism that satisfies condition (2) of Definition 14.

Before introducing the main unification, the unification for the various logics at hand we introduce the combined unification (or $\sigma^{A_1 \dots A_n}$ -unification), where $A_1 \dots A_n$ is the list of axioms defining a logic L .

DEFINITION 17. Let $i, j \in \mathfrak{S}$

$$[i, j]\sigma^{A_1 \dots A_n} = \begin{cases} [i, j]\sigma^{A_1} & \text{or} \\ \vdots \\ [i, j]\sigma^{A_n} \end{cases}$$

Finally we are ready to give the main unification for a logic L (or σ_L), where the logic is defined by axioms $A_1 \dots A_n$. This unification which will be used within the inference rules.

DEFINITION 18. Let $i, j \in \mathfrak{S}$

$$[i, j]\sigma_L = \begin{cases} [i, j]\sigma^{A_1 \dots A_n} & \text{or} \\ [c^n(i), c^m(j)]\sigma^{A_1 \dots A_n} & \exists n, m : 1 \leq n \leq \ell(i), 1 \leq m \leq \ell(j) \end{cases}$$

where $w_0 = [s^n(i), s^m(j)]\sigma_L$.

Notice that σ_L has a recursive definition.

EXAMPLE 19. Let us consider the labels

$$(4) \quad i = (W_2, (w_2, w_1)) \quad j = (W_3, (w_3, w_1))$$

The two labels do not σ^B -unify, they have the same length and do not σ^D -unify: the segments of length 2 are restricted and they have different heads thus there is no substitution ρ such that $\rho(w_2) = \rho(w_3)$. However, the two labels σ_{DB} -unify. We use the following decompositions

$$\begin{aligned} c^1(i) &= (W_2, (w_2, w_0)) & s^1(i) &= w_1 \\ c^3(j) &= w_0 & s^3(j) &= (W_3, (w_2, w_1)). \end{aligned}$$

It is easy to see that $[c^1(i), c^3(j)]\sigma^B = w_0$ and $w_0 = [s^1(i), s^3(j)]\sigma^B = w_1$.

When we consider the interpretation of the labels in (4) we have that i , intuitively, denotes the set of worlds accessible from w_2 which is accessible from w_1 and, similarly, the interpretation of j is the set of worlds accessible from w_3 which, in turn, is accessible from w_1 . Since the accessibility relation is symmetric, w_1 belongs to both interpretations; thus the denotations of i and j have a non empty intersection and thus they labels hold unify. The σ_{DB} -unification takes care of cases like this.

Inference rules

For the propositional part of KEM we exemplify only the rules for conjunction.

$$(\alpha\text{-rules}) \quad \frac{A \wedge B : i}{\begin{array}{l} A : i \\ B : i \end{array}}$$

The α -rules are just the familiar linear branch-expansion rules of the tableau method.

$$(\beta\text{-rules}) \quad \frac{\neg(A \wedge B) : i}{A : j} \quad \frac{\neg(A \wedge B) : i}{B : j}$$

$$\frac{\quad}{\neg B : [i, j]\sigma_L} \quad \frac{\quad}{\neg A : [i, j]\sigma_L}$$

The β -rules are nothing but natural inference patterns such as Modus Ponens, Modus Tollens and Disjunctive syllogism generalised to the modal case. 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.

$$(\pi\text{-rules}) \quad \frac{\diamond A : i}{A : (w_n, i)} \quad \frac{\neg \Box A : i}{\neg A : (w_n, i)}$$

where w_n is new, that is, it does not occur in the tree.

$$(\nu\text{-rules}) \quad \frac{\Box A : i}{A : (W_n, i)} \quad \frac{\neg \diamond A : i}{\neg A : (W_n, i)}$$

where W_n is new.

ν - and π -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.

$$(\text{PB}) \quad \frac{}{A : i \quad | \quad \neg A : i}$$

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 it 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 β -formulas, that is β formulas for which we have no immediate sub-formulas with the appropriate labels in the branch (tree).

$$(\text{PNC}) \quad \frac{A : i \quad \neg A : j}{\times} [i, j]\sigma_L$$

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

Let L be one of the fifteen basic modal logics. With $\vdash_{\text{KEM}(L)} A$ we mean that there is a close KEM-tree for $\neg A : w_1$; or, in other words, that SST proves that A is a theorem of L .

THEOREM 20. [15, 20] $\vdash_{\text{KEM}(L)} A$ iff $\models_L A$.

3 Beyond basic modal logics

3.1 Non-normal modal logics

Normal modal logics are extensions of classical propositional logic with axiom K (i.e., $\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$) and the necessitation rule (i.e., $\vdash A / \vdash \Box A$). However, weaker extensions are possible, when we consider the following rules to extend classical propositional logic:

$$(RE) \quad \frac{\vdash A \leftrightarrow B}{\vdash \Box A \leftrightarrow \Box B}$$

$$(RK) \quad \frac{\vdash (A_1 \wedge \dots \wedge A_n) \rightarrow A}{\vdash (\Box A_1 \wedge \dots \wedge \Box A_n) \rightarrow \Box A} \quad n \geq 0$$

and, in particular, we shall consider

$$(Nec) \quad \frac{\vdash A}{\vdash \Box A} \quad (RK, n = 0)$$

$$(RM) \quad \frac{\vdash A \rightarrow B}{\vdash \Box A \rightarrow \Box B} \quad (RK, n = 1)$$

$$(RR) \quad \frac{\vdash (A \wedge B) \rightarrow C}{\vdash (\Box A \wedge \Box B) \rightarrow \Box C} \quad (RK, n = 2)$$

We can now classify modal logics according to their deductive power.

DEFINITION 21. A modal logic L is *classical* iff it is closed under RE; *monotonic* iff it is closed under RM; *emphregular* iff it is closed under RR; *normal* iff it is closed under RK.

According to [9] the smallest classical logic is called E , the smallest regular logic R , the smallest monotonic logic M , and the smallest normal logic K .

The semantics of non-normal modal logic is given in terms of neighbourhood semantics. A model is a structure

$$\mathcal{M} = \langle W, N, v \rangle$$

where W is a set of possible worlds, N is a function from W to 2^{2^W} and v is an evaluation function: $v : WFF \times W \mapsto \{T, F\}$, where WFF is the set of well-formed formulas.

Before providing the evaluation clauses for the formulas we need to define the notion of truth set.

DEFINITION 22. Let \mathcal{M} be a model and A be a formula. The truth set of A wrt to \mathcal{M} , $\|A\|^{\mathcal{M}}$ is thus defined: $\|A\|^{\mathcal{M}} = \{w \in W : v(A, w) = T\}$.

The evaluation clauses for atomic and boolean formulas are as usual while those for modal operators are given below.

DEFINITION 23. Let w be a world in $\mathcal{M} = \langle W, N, v \rangle$:

1. $w \models \Box A \iff \|A\|^{\mathcal{M}} \in N_w$;
2. $w \models \Diamond A \iff W - \|A\|^{\mathcal{M}} \notin N_w$.

It is natural to add some conditions on the function N in neighbourhood models. The conditions relevant for the present work are given in the following definition.

DEFINITION 24. Let \mathcal{M} be a model. For every world $w \in W$ and every proposition A , and B .

- (m) If $\|A\| \cap \|B\| \subseteq N_w$, then $\|A\| \in N_w$ and $\|B\| \in N_w$;
- (c) If $\|A\| \in N_w$ and $\|B\| \in N_w$, then $\|A\| \cap \|B\| \in N_w$;
- (n) $W \in N_w$.

According as the function N in a neighbourhood model satisfies condition (m), (c), or (n), we shall say that the model is *supplemented*, is *closed under intersections*, or *contains the unit*. When a model is both supplemented and closed under intersections then we shall call it a *quasi-filter*; when a quasi-filter contains the unit it is a *filter*.

The conditions determining the minimal non-normal modal logics are as follows:

1. E is characterised by the class of neighbourhood models;
2. M is characterised by the class of supplemented models;
3. R is characterised by the class of quasi-filters;
4. K is characterised by the class of filters.

From now on we shall use $\models_L A$ to denote that A is valid in the class of model characterising L.

KEM for non-normal modal logic

Here we illustrate how to modify KEM to capture monotonic and regular modal logics. The required modifications involve the definition of substitution, Definition 6.

We shall denote the constants occurring in labels obtained as the result of an unification with $*$, and we shall denote the set of such constants by Φ_C^* .

It is worth noting that the variables can be mapped on more than a label in the course of a proof; imposing restrictions on the number of labels a variable can be mapped to in the course of a proof makes us able to characterise the classes of modal logics at hand. More precisely the world substitutions for the classes of logics under analysis are:

Monotonic logics

$$\rho^M : \Phi_V \mapsto \mathfrak{S}_{\text{branch}} \quad \text{injective}$$

$$\mathbf{1}_{\Phi_C^*}$$

The condition for monotonic logics states that a variable can be mapped to a unique label in a branch of a KEM-proof, while constants are mapped to themselves only if they are the result of a unification. It is worth noting that it is possible to map a variable to different labels if they occur in distinct branches.

Regular logics

$$\rho^R : \Phi_V \mapsto \mathfrak{S}$$

$$\mathbf{1}_{\Phi_C^*}$$

For regular logics the restriction on variables is released, while that on constants still obtains.

THEOREM 25. [21] *Let L be either M or R. $\vdash_{\text{KEM}(L)} A$ iff $\models_L A$.*

Unfortunately at the moment it is not known whether it is possible to capture classical modal logics by imposing similar restrictions to the substitution functions.

More on non-normal modal logics

Jones and Pörn [24, 25] defined a non-normal multi-modal deontic logic where, semantically, the set of worlds accessible from a given world w is partitioned into ideal and sub-ideal worlds: an accessible world is ideal if all obligations in w are respected and sub-ideal if some of the obligations in w are violated. In addition each world is either an ideal or a sub-ideal version of itself.

First of all atomic labels are indexed with either d , s or nothing. Thus, for example, (w_2^d, w_1) means that w_2 is an ideal version of w_1 ; (W_1^s, w_1) denotes the set of all subideal versions of w_1 , and (w_2, w_1) says that we do not know if w_2 is an ideal or subideal version of w_1 .

To accommodate the above conditions it is possible to define new inference rules operating on labels instead of the declarative units of labelled formulas.

$$\text{Exec} \frac{A : (W^s, i) \quad A : (W^d, j)}{A : (W, [i, j]_{\sigma_{JP}})} \quad \text{LPNC} \frac{A : i^s \quad B : i^d}{\times} \quad \text{LPB} \frac{}{A : i^s \mid B : i^d} i \text{ restricted}$$

The above three rules give us that the set of worlds accessible from a given world is a partition. Exec tells us that if A holds in all ideal versions of a world as well as in all subideal versions then it holds in all accessible worlds. LPNC says that it is not possible to have a world that is at the same time an ideal and a subideal version of another world. Finally, the meaning of LPB

is that the classes in a partition are mutually exclusive and so we can create two mutually exclusive branches for our tableaux tree.

$$\begin{array}{c} \Box A : i \\ \text{Ref} \frac{\neg A : j}{\Box A : k^s} [i, j] \sigma_{\text{JP}} = k \\ \neg A : k^s \end{array}$$

Finally the Ref rules (reflexivity rules) allows us on detection of a violation (i.e, we both have an obligation and the negation of the content of the obligations with two labels that unify) to determine when a world is a subideal version of itself. For a full account, we refer to [19, 4].

Labelled tableaux systems using the explicit representation of the accessibility relation have been proposed for non-normal modal logic by [16].

3.2 Conditional logics

Conditional logics are extensions of classical logic with a binary intensional operators, $>$, meant to represent hypothetical, conditional or counterfactual reasoning [26, 31]. Different possible world semantics have been put forward for conditional logics: the system of sphere semantics [26] and the selection function semantics [31]. If one wants to use labels to mimic the semantics of conditional logics, one has to chose the most appropriate semantics. Most of the (labelled) tableaux systems for conditional logics assume the selection function semantics, where a model \mathcal{M} is a structure $\langle W, f, v \rangle$, where W is a set of possible worlds, f is a selection function which picks for every formula A a subset $f(A, w)$ of W for each world $w \in W$, and v is a valuation function assigning to every formula A and $w \in W$ a boolean value. We refer to $f(A, w)$ as the set of A -worlds relative to w . The valuation condition for a conditional formula $A > B$ is as follows:

$$w \models A > B \text{ iff } f(A, w) \subseteq \|B\|$$

Chellas [8] proposed the reading of a conditional operators as a parametrised modal operator, that is $A > B$ can be understood as $[A]B$. Based on this reading it is possible to consider conditional logics as a type of multi-modal logic and to use the idea of having different types of labels for each modal operator. Accordingly, to cope with conditional logics, we extend the label formalism, and atomic labels can be indexed by formulas. Hence we can have labels as (W_1^A, w_1) , intuitively denoting the set of A -worlds relative to the world denoted by w_1 and (w_2^B, w_1) for a possible word in $f(B, w_1)$.

In general the unification for a conditional logic has the following structure:

$$[i^Y, j^Z] \sigma_{>} = [i, j] \sigma$$

and for each ‘turning point’ $(i^{Y'}, j^{Z'})$ one of the following conditions (for normal conditional logics) (i) $Y' \equiv Z'$ or (ii) $Y' \equiv \top$ and $h(i') \in \Phi_V$ or (iii)

For the inference rules we have to consider that now formulas can occur both in the declarative part of a labelled formula but also as index of a

label. Thus the notation $A@X : i^Y$ means that X is either Y or C . Based on the intuition described so far the inference rules for $>$ are as follows:

$$\frac{A > B@X : i^Y}{B : (W_n^A, i^Y)} \quad \frac{A > B@X : i^Y \quad A@X' : j^A}{A : (c^{\ell(j)-1})^A} \quad \frac{\neg(A > B)X@i^Y}{\neg B : (w_n^A, i^Y)}$$

The presence of two rules for the case of a positive conditional is due to the fact that positive conditional behaves both as α and β formulas (according to Smullyan classification). Notice that the β -version can be problematic in some conditional logic.

In [2] we discuss the issues of the design of a labelled tableaux for conditional logic (some of them are related to the fact that one has to begin a new tableaux to check the equivalence of two formulas when computing a unification), and in [3] we provided a sound and complete system for the flat fragment of some particular logics. Pozzato and co-workers [33] used and extended the ideas of [2, 3] to provide sequents and tableaux systems for a larger class of conditional or conditional like logics. Priest [34], on the other hand presented tableaux for conditional logics using the propagation of formulas based on the representation of the semantic structure in first-order logic.

4 Relative complexity: the beauty of symmetry

In the last few years several comparisons (competitions) of theorem provers for modal logic have been held (cf. [6, 28, 30]) and experimental research has been carried out (cf. [23, 22]). Despite the potential interest for eventual applications, we believe that this kind of research provided little or no insight on better theoretical architectures for modal theorem provers. Very often the overall performance is heavily influenced by external factors, for example, language specific optimisations of the implementation.

In this section we compare SST and KEM from a theoretical perspective. This means that we do not consider implementation issues, but only logical ones; moreover we are not interested in the propositional features and in the interaction of modal operators and propositional connectives, but only in the modal characteristics.

To prove that a proof system \mathcal{A} is essentially better than a proof system \mathcal{B} we have to exhibit at least one formula (or a class of formulas) for which \mathcal{A} is better than \mathcal{B} , and for all formulas \mathcal{A} is not essentially worse than \mathcal{B} .³ There are many distinct modal logics, and it is possible that results in a logic do not apply to a different logic. Moreover \mathcal{A} may cover some modal logics which are not covered by \mathcal{B} and the other way around. However, any general purpose modal theorem prover should cover the basic fifteen normal modal logics. Among them, some offer too simple modal structures while other lend themselves to specialised optimisation procedures (in particular the logics

³We shall give a precise definition of what “better” and “worse” mean in this context in Section 4.2.

with a finite number of distinct modalities). In both cases, these logics do not provide the best scenario to really test the theoretical architecture behind a modal theorem prover. Therefore we have to identify a modal logic with the following properties:

1. it is one of the basic fifteen normal modal logics;
2. the proof procedures are modular for both systems, that is, they are the combination of the proof procedures of the single components of the logic; and
3. there are no specialised proof procedures.

The basic normal modal logic DB satisfies the criteria listed above to be a representative candidate to test the capability of a theorem prover for modal logic. Moreover, due to some well-known difficulties [13], symmetric logics lie outside most of the current modal theorem prover methods, though they play an important role in different applications areas.

4.1 The complexity of KEM unifications for DB

To provide a comparison of the two methods at hand first we have to study the complexity of the KEM unification procedure. We start by showing that the unification of two world symbols can be computed in constant time.

LEMMA 26. *The σ -unification of two world symbols w and w' can be computed in constant time.*

Proof. It is immediate to see that the unification of two world symbols requires at most three steps, and thus it has constant complexity. ■

As we have seen the unification of two world symbols is just the first basic step of the unification. The next step is the σ -unification of two labels; in this case, we can prove that its complexity is linear in the length of the two labels.

LEMMA 27. *The σ -unification of two labels i and j can be computed in linear time.*

Proof. All we have to do is to see whether the word symbols in the two labels stepwise unify.

Thus at the end we have to verify n unifications of world symbols, but from Lemma 26, we know that the unification of world symbols can be computed in constant time. Therefore the σ -unification of two labels can be computed in linear time. ■

The next unification we have to examine is the unification for the axiom B.

LEMMA 28. *The σ^B -unification of two labels i and j can be computed in linear time.*

Proof. The computation of the σ^B -unification of two labels i and j can be reduced to three sub-problems. (1) To compute the lengths of the two labels and to determine whether the difference is even. This obviously can be computed in linear time: all we have to do is to scan sequentially the two labels. (2) To determine whether an injective morphism from $\Phi_C^{\bar{i}, \ell(\bar{j})}$ to $\Phi_V^{\bar{i}, \ell(\bar{j})}$ exists. It is to implement a linear time algorithm that scans the labels and increments or decrements a counter to verify that there is such an injective morphism. (3) To compute $[s^{\ell(\bar{j})}(\bar{i}), \bar{j}]\sigma$: by Lemma 27 the σ -unification of two labels has linear complexity. Therefore the complexity of σ^B is linear. ■

Unfortunately we cannot prove such good complexity results for σ_{DB} ; however, for special labels we can prove the following result.

LEMMA 29. *The σ_{DB} -unification of two labels i and j such that $\ell(i) = 1$ can be computed in quadratic time.*

Proof. For a label j of length n there are n distinct segments and n distinct countersegments, namely

$$\begin{array}{ll} c^n(j) = w_0 & s^n(j) = j \\ c^{n-1}(j) = (h^n(j), w_0) & s^{n-1}(j) = b(j) \\ c^{n-2}(j) = (h^n(j), (h^{n-1}(j), w_0)) & s^{n-2}(j) = b(b(j)) \\ \vdots & \vdots \end{array}$$

Now we have to see whether i either σ^B - or σ -unifies with the countersegments and whether i σ_{DB} -unifies with the segments. Thus we have to compute $2n$ linear unifications and n σ_{DB} -unifications. Let us examine the first of these, i.e., $[s^{n-1}(j), i]\sigma_{DB}$. This time the length of $s^{n-1}(j)$ is $n - 1$, and thus we have $n - 1$ ways to split it in segments and countersegments. That is:⁴

$$\begin{array}{ll} c^{n-1}(c^{n-1}(j)) = w_0 & s^{n-1}(j) = b(j) \\ c^{n-2}(c^{n-1}(j)) = (h^{n-1}(j), w_0) & s^{n-2}(j) = b(b(j)) \\ \vdots & \vdots \end{array}$$

A close inspection shows that only the countersegments are different from the previous step. Therefore we can repeat this process for all the segments of j , and each time we can replace the σ_{DB} unification for the appropriate segment of length m , with $2m$ linear unifications. Hence, at the end, the number of linear unifications we have to compute is

$$2 \sum_{n=1}^{n=\ell(j)} n = O(n^2)$$

which shows that the σ_{DB} -unification for the case at hand is quadratic. ■

⁴Notice that for $m \leq n$ $s^m(s^n(i)) = s^m(i)$.

4.2 KEM vs. SST

So far the standard way to compare the relative complexity of two proof systems was given by the notion of p-simulation.

DEFINITION 30. A proof system \mathcal{A} p-simulates a proof system \mathcal{B} iff there is a function g , computable in polynomial time, which maps derivations in \mathcal{B} for any given formula ϕ , to derivations in \mathcal{A} for ϕ (cf. [10]).

The main problem with p-simulation is that it considers only proofs, i.e., closed trees in tableaux terminology, and it says nothing about open trees. While this notion is fully appropriate for semi-decidable logics and non deterministic proof systems, it does not offer a good measure to compare tableaux-like proof-systems for decidable modal propositional logics. The main point is that this notion does not contemplate proof-procedures. Modal tableaux proof-procedures, in effect, are systematic searches for models that make the initial formula true with respect to the initial world. In this perspective modal tableaux can show that a formula is not a theorem by showing that the negation of the formula is satisfiable. However, to show that a formula is satisfiable we have to complete its tree. In general, to complete a tree we have to explore the whole search space generated by the formula.

Therefore, to obviate the above problem, we propose a stepwise simulation. Here the main idea is that a proof system \mathcal{A} stepwise simulates a proof system \mathcal{B} iff \mathcal{A} does not perform any inference steps for which no corresponding inference steps exist in \mathcal{B} .

DEFINITION 31. A proof system \mathcal{A} p-search-simulates a proof system \mathcal{B} iff there is a polynomial function g such that for any formula ϕ , g maps derivations (trees) from ϕ in \mathcal{A} to derivations (trees) from ϕ in \mathcal{B} (cf. [11]).

Note that a stepwise simulation is independent of whether the considered derivations (trees) are proofs or not.

We are now ready to present the main result of the paper. To prove it we have to identify a formula (or a class of formulas) whose complete KEM-tree is polynomial in the size of the formula while the complete SST-tree is exponential in the size of the formula. Surprisingly the formula is extremely simple, namely:

$$(5) \quad p \rightarrow (\Box\Diamond)^n p$$

As we shall see (5) involves only one propositional linear step and there are no interaction between propositional connectives and modal operators. Therefore the discriminant is only the way the two proof systems deal with modal operators.

THEOREM 32. *The length of the complete proof of $p \rightarrow (\Box\Diamond)^n p$ in KEM is $O(n^2)$.*

Proof.

1. $\neg(p \rightarrow (\Box\Diamond)^n p) : w_1$
2. $p : w_1$
3. $\neg(\Box\Diamond)^n p : w_1$
4. $\neg\Diamond(\Box\Diamond)^{n-1} p : (w_2, w_1)$
5. $\neg(\Box\Diamond)^{n-1} p : (W_1, (w_2, w_1))$
- \vdots
- $2n + 3.$ $\neg p : (W_n, (w_{n+1}, (\dots, (W_1, (w_2, w_1)) \dots)))$

The initial formula, i.e., $\neg(p \rightarrow (\Box\Diamond)^n p) : w_1$, is of type α , then we expand the tree with two nodes both labelled with w_1 : the first of such nodes (2) consists of p which is atomic and does not need further investigations; the second node (3) contains a formula of type π labelled with w_1 . From (3) we obtain (4), which is of type ν . Applying the ν -rule on it, we get (5). We repeat the above steps $n - 1$ times, for a total of $2n + 3$ steps (nodes).

At this point we have two complementary formulas, the formulas in (2) and $(2n + 3)$. We have to verify whether the two labels σ_{DB} -unify.

From Lemma 29 we know that the complexity of the instance of σ_{DB} -unification at hand is quadratic in the length of the labels involved, which in turn, is linear in the size of the formula. Therefore the complexity of the complete KEM-proof of $p \rightarrow (\Box\Diamond)^n p$ is $2n + 3 + O(n^2) = O(n^2)$. ■

THEOREM 33. *The length of the complete proof of $p \rightarrow (\Box\Diamond)^n p$ in SST is $O(2^{n+1})$.*

Proof.

1. $\neg(p \rightarrow (\Box\Diamond)^n p) : w_1$
2. $p : w_1$
3. $\neg(\Box\Diamond)^n p : w_1$
4. $\neg\Diamond(\Box\Diamond)^{n-1} p : (w_2, w_1)$
5. $\neg(\Box\Diamond)^{n-1} p : (W_1, (w_2, w_1))$
6. $\neg(\Box\Diamond)^{n-1} p : w_1$
7. $\neg\Diamond(\Box\Diamond)^{n-2} p : (w_3, (W_1, (w_2, w_1)))$
8. $\neg\Diamond(\Box\Diamond)^{n-2} p : (w_3, w_1)$
9. $\neg(\Box\Diamond)^{n-2} p : (W_2, (w_3, (W_1, (w_2, w_1))))$
10. $\neg(\Box\Diamond)^{n-2} p : (w_2, w_1)$
11. $\neg(\Box\Diamond)^{n-2} p : (W_2, (w_3, w_1))$
12. $\neg(\Box\Diamond)^{n-2} p : w_1$
- \vdots

The formula we start with $(\neg(p \rightarrow (\Box\Diamond)^n p) : w_1)$ is of type α , and then we obtain two formulas $p : w_1$ and $\neg(\Box\Diamond)^n p : w_1$. At this point we have an atomic formula and a formula of type π . We apply the π -rule on it deriving $\neg\Diamond(\Box\Diamond)^{n-1} p : (w_2, w_1)$. Now we have a formula of type ν , and we have to apply both the ν -rule for **D** and **B**, thus we have to produce the formulas $\neg(\Box\Diamond)^{n-1} p : w_1$ and $\neg(\Box\Diamond)^{n-1} p : (W_1, (w_2, w_1))$. These last two

We apply PB with respect to ν_0 , and with label $b(i)$; in the right branch we apply the ν rule and we obtain $\nu_0 : (w_n, i)$, but $[b(i), (W_n, i)]\sigma_{DB}$, and thus the branch is closed. In particular it is possible to show that the labels involved σ^B -unify, and we have seen (Lemma 28) that the σ^B -unification can be computed in linear time. Therefore the derivation of ν_B has linear complexity. ■

Lemma 35 allows us to define a proof-search in KEM where we use both the new derived ν -rule and the original ν -rules of KEM, and the unification is restricted to σ . It is immediate to see that this proof procedure corresponds to SST, and the components involved have linear complexity, we have thus proved the following theorem.

THEOREM 36. *KEM p -search-simulates SST.*

Notice that the results above extends immediately to the ground version of SST [27, 29] as well as to Fitting's prefix tableaux [13].

5 Conclusions

Labels can be a very powerful tool for the design of (semantic based) deductive systems. In this paper we have seen how labels can be used to create tableaux system for a variety of logics amenable of possible world semantics. In addition we have shown that the use of free-variable labels with particular logic dependant label algebra can speed up the complexity of modal tableaux.

Acknowledgements. This work was supported by Australia Research Council under Discovery Project No. DP0452628 on "Combining modal logic for dynamic and multi-agents systems".

National ICT Australia is funded by the Australian Government's Department of Communications, Information Technology and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Centre of Excellence program.

BIBLIOGRAPHY

- [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. Zakharyashev, editors, *Advances in Modal Logic. Volume 1*, pages 1–21. CSLI Publications, 1998.
- [2] A. Artosi and G. Governatori. A tableaux methodology for deontic conditional logics. In *$\Delta EON'98$* , pages 65–81, Bologna, 1998.
- [3] 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, 2002.
- [4] A. Artosi, G. Governatori, and G. Sartor. Towards a computational treatment of deontic defeasibility. In M. Brown and J. Carmo, editors, *Deontic Logic Agency and Normative Systems*, pages 27–46. Springer-Verlag, 1996.
- [5] M. Baldoni, L. Giordano, and A. Martelli. A tableau for multimodal logics and some (un)decidability results. In Harrie C. M. de Swart, editor, *TABLEAUX'98*, LNCS 1397, pages 44–59. Springer, 1998.

- [6] P. Balsiger and A. Heuerding, *Comparison of theorem provers for modal logics – introduction and summary*. In H. C. M. de Swart, editor, *TABLEAUX '98*, LNCS 1397, pages 25–26, Springer-Verlag, 1998.
- [7] B. Beckert and R. Goré. Free variable tableaux for propositional modal logics. *Studia Logica*, 69(1):59–96, 2001.
- [8] B.F. Chellas. Basic conditional logic. *Journal of Philosophical Logic*, 4:133–153, 1975.
- [9] B.F. Chellas. *Modal Logic, An Introduction*. Cambridge University Press, 1980.
- [10] S.A. Cook and R.A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44:36–50, 1979.
- [11] H. de Nivelle, R. Schmidt, and U. Hustadt. Resolution-based methods for modal logics. *Logic Journal of IGPL*, 8:265–292, 2000.
- [12] F.B. Fitch. Tree proofs in modal logic. *Journal of Symbolic Logic*, 31:152, 1966.
- [13] M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*. Reidel, 1983.
- [14] D.M. Gabbay. *Labelled Deductive System*. Oxford University Press, 1996.
- [15] D.M. Gabbay and G. Governatori. Fibred modal tableaux. In D. Basin, M. D’Agostino, D.M. Gabbay, S. Matthews, and L. Viganó, editors, *Labelled Deduction*, pages 163–194. Kluwer, 2000.
- [16] R. Girle. *Modal Logic and Philosophy*. Acumen, 2000.
- [17] R. Goré. Tableau methods for modal and temporal logics. In M. D’Agostino, D.M. Gabbay, R. Heinle, and J. Posegga, editors, *Handbook of Tableau Methods*, pages 297–396. Kluwer, 1999.
- [18] G. Governatori. Labelled tableaux for multi-modal logics. In P. Baumgartner, R. Hähnle, and J. Posegga, editors, *TABLEAUX'95*, LNAI 918, pages 79–94, Springer-Verlag, 1995.
- [19] G. Governatori. Labelling ideality and subideality. In D.M. Gabbay and H.J. Ohlbach, editors, *Practical Reasoning*, LNAI 1085, pages 291–304, Springer-Verlag, 1996.
- [20] G. Governatori. *Un modello formale per il ragionamento giuridico*. PhD thesis, CIRFID, University of Bologna, Bologna, 1997.
- [21] G. Governatori and A. Luppi. Labelled tableaux for non-normal modal logics. In E. Lamma and P. Mello, editors, *AI*IA 99*, LNAI 1792, pages 119–130, Springer-Verlag, 2000.
- [22] I. Horrocks, P.F. Patel-Schneider, and R. Sebastiani. An analysis of empirical testing for modal decision procedures. *Logic Journal of IGPL*, 8:293–323, 2000.
- [23] U. Hustadt and R. Schmidt. On evaluating decision procedures for modal logic. In *IJCAI'97*, pages 202–207, 1997.
- [24] A.J.I. Jones and I. Pörn. Ideality, sub-ideality and deontic logic. *Synthese*, 65:275–290, 1985.
- [25] A.J.I. Jones and I. Pörn. “Ought” and “Must”. *Synthese*, 66:89–93, 1986.
- [26] D. Lewis. *Counterfactuals*. Basil Blackwell, 1986.
- [27] F. Massacci. Strongly analytic tableaux for normal modal logic. In A. Bundy, editor, *CADE-12*, LNAI 814, pages 723–737, Springer-Verlag, 1994.
- [28] F. Massacci. Design and results of the Tableaux-99 non-classical (modal) systems comparison. In N.V. Murray, editor, *TABLEAUX '99*, LNCS 1617, pages 14–18, Springer-Verlag, 1999.
- [29] F. Massacci. Single step tableaux for modal logic. *Journal of Automated Reasoning*, 24(3):319–364, 2000.
- [30] F. Massacci and F.M. Donini. Design and results of tancs-2000 non-classical (modal) systems comparison. In R. Dyckhoff, editor, *TABLEAUX 2000*, LNCS 1847 pages 52–56, Springer-Verlag, 2000.
- [31] Donald Nute. *Topics in Conditional Logic*. Reidel, Dordrecht, 1980.
- [32] H.J. Ohlbach. Semantic based translation methods for modal logics. *Journal of Logic and Computation*, 1:691–746, 1991.
- [33] G.L. Pozzato. *Proof Methods for Conditional and Preferential Logics*. PhD thesis, Università degli Studi di Torino, 2007.
- [34] G. Priest. *An Introduction to Non-Classical Logic*. Cambridge University Press, 2001.

- [35] R.M. Smullyan. *First-Order Logic*. Springer-Verlag, Berlin, 1968.

Guido Governatori
Queensland Research Laboratory,
National ICT Australia,
Brisbane, Queensland, Australia
guido.governatori@nicta.com.au