K-CLUSTERS TABLEAUX
A Tool for Modal Logics and Inconsistent Belief Sets

Fabio Massacci

Dipartimento di Informatica e Sistemistica

Universitá di Roma "La Sapienza"

via Salaria 113, 00196 Roma

email: massacci@assi.dis.uniroma1.it

ABSTRACT

The goal of this paper is to present a tableau calculus for propositional modal logics that is effective for automated proof search, easy to use for proof presentation, and applicable to a wide range of modal logics for knowledge representation. The calculus is also enhanced to provide a form of contextual reasoning to deal with (classical) inconsistency.

The basic tableau calculus (*Single Step Tableaux*) combines the advantage of both sequent-like tableaux and prefixed tableaux and eliminates the need of complex closure operations for the accessibility relation, nondeterminism and cut rules. Thus analiticy is provided together with a strong completeness theorem without cut. Single Step rules are rather efficient (they satisfy suitable locality principles) and also strongly confluent. They can thus be parallelized and adapted to many search heuristics.

The system is then extended to cope with inconsistent belief sets (k-*Cluster Tableaux*), with a form of local reasoning. Yet it does not requires the introduction of ad hoc connectives, or multiple valued semantics. A simple and intuitive semantics based on Kripke models is presented. Thus, we combine the full power of a modal theorem prover with a system of context handling. It is weak enough to describe knowledge bases where two different consistent statements (modal formulae) may contradict each other (in classical logic), but strong enough to rule out the possibility of an inconsistent statement.

1 INTRODUCTION

The use of modal logic spreads across Artificial Intelligence and Computer Science, as a way of modelling knowledge and belief [15] or as a formalism to specify the behaviour of distributed systems [14]. Recently, monotonic modal logics have also been used as a basis for different nonmonotonic formalisms [18, 19, 23]. These applications of modal logics (monotonic or not) often require a notion of strong deducibility: given a set of modal formulae Γ, decide if a formula ϕ is a (non)monotonic consequence of Γ. Intuitively, this can be seen as the query "does the knowledge base Γ entail ϕ?".

Thus, devising decision procedures for strong deducibility becomes a compelling task in automated deduction. However, modal logics has faced a trade-off between systems which are easier to use and understand but computationally unfeasible (like sequent or Hilbert calculi [7, 8, 13, 15, 17, 28]) and systems computationally more effective which require to understand complex ad hoc unification procedures [24, 25, 26, 27, 30]. Labelled Deductive System of Gabbay [9] and prefixed tableaux of Fitting [7] require also the user to understand an algebra of labels. Thus, a major objective is the design of *effective and human oriented* decision procedure.

Sometimes, even an effective system for strong deducibility is not enough: in practice, a knowledge base will often be made by different (possibly large) chunks of epistemic data. Each chunk could (probably) be consistent but the whole knowledge base may well be inconsistent: consider a database built by putting together the opinions of different experts on various fields. However, human beings do not allow their reasoning ability on specific issues to be threatened by this kind of inconsistency. As Fagin & Halpern have noted in [6]:

> People don't focus on all issues simultaneously: Thus, when we say "a believes p", we more properly mean that in a certain frame of mind (when a is focussing on the issues that involve p), it is the case that a believes p. Even if a does perfect reasoning with respect to the limited number of issues on which he is focussing in any given frame of mind, he may not put his conclusion together. Indeed, although in each frame of mind agent a may be consistent, the different conclusions a draws in different frames of mind may be inconsistent. [...]

> Our key observation is that one reason that people hold inconsistent beliefs is that beliefs tend to come in non-interacting clusters. We can almost view an agent as a society of minds, each with is own set (or *cluster*) of beliefs which may contradict each other.

To cope with this new requirement, a new concept must be introduced: context (or cluster) reasoning. The relevance of contextual reasoning for AI has been discussed in [22], and the modelling of human reasoning as non-interacting clusters in [29]. Different solutions have been proposed in the literature: context systems can be seen as a tool to cope with lack of logical omniscience [6, 12] or even a way to get rid of modal logics [11].

Our goal is slightly different from those: we claim that some "real-world" features of contextual reasoning can still be achieved in the standard framework of modal logic and Kripke models. Thus, after devising a tableaux system for modal logics which is both effective and human oriented, we enhance it, and straightforwardly extend it to cope with inconsistent belief sets and to perform local (contextual) reasoning.

1.1 Modal Deduction

Many authors have tackled modal deduction from different perspectives and the research in the field may be split into two main threads: *direct* and *indirect* methods.

Direct methods deal with modal formulae and enhance the classical deduction mechanism with suitable rules to handle modal connectives. Besides Hilbert-style proof systems [15, 17], traditional systems are tableau and sequent calculi[7, 8, 13, 28]. They allow to present proofs in a simple way. The rules (axioms) they use (almost) remind the epistemic properties (omiscience, introspection etc.) a user would like to model. However, they have serious limitations when computational features are at stake. For instance, Hilbert systems are not analitic and thus cannot be thouroghly automated. Sequent and tableau calculi are (sometimes) analitic, yet they have other drawbacks: many formulae (possibly all) may have to be carried along in the deduction process and the absence of a Church-Rosser theorem (stating that rules may commute) hinders the independence of the deduction steps from the chosen search heuristics. Moreover, strong deducibility for symmetric and euclidean logics, which are now playing a major role in nonmonotonic reasoning [18, 19, 23], is still unsatisfactory. At this point, (sequent-like) tableaux or sequents treat these logics in an entirely non-deterministic way, since cut has been eliminated for K45 [13, 28] but only for weak deducibility (i.e. without premises). The only effective way seemed to be reverting from deduction to model building and explicitly constructing a (counter) model. However such a construction requires to calculate many times the euclidean, symmetric etc. closure of the accessibility relation [4, 15].

On the other side, indirect methods translate modal formulae into first order ones and then apply a (suitably modified) classical deduction technique. A general overview can be found in [27]. Some translations, like the relational translation, are definitely inefficient. Others, like the functional translation perform much better and can be easily applied either to matrix proof methods [30] or resolution [26, 27]. The key idea is to replace each modal propositional literal p by a monadic predicate $p(\sigma)$, such that σ (a suitable string of constants, functions and variables) "corresponds" to the possible world where the original p was supposed to hold. Therefore, $p(\sigma)$ and $\neg p(\sigma^*)$ can be resolved if, in the given modal logic, σ and σ^* can be unified. Unfortunately, each logic requires its own unification procedure. If we cannot build unification into the theorem prover, or we don't want to commit ourselves to a specific logic, a heavy equational theory is needed [27]. In this way the potential effectiveness of the procedure is limited. To overcome these difficulties, a mixed approach, which allows also a direct translation into PROLOG, has been proposed in [24, 25]. A common limitation of these proposals is that they seem to need a "user-friendly" interface for the presentation of proofs and proof search.

Prefixed tableaux [7] and Labelled Deductive Systems [9] stand in between, yet they are more oriented towards model checking. In these frameworks, formulae are labelled with a prefix, to "name" the possible world where each formula is supposed to hold. Proofs and proof search presentations are simple and elegant, since they closely reflect the Kripke semantics of possible worlds. Yet, they need an heavy equational (algebraic) theory to establish relations among prefixes (labels).

Our proof system — we called it *Single Step Tableaux* (SST) — also uses prefixes to keep track of the worlds where formulae hold. The key difference is that the underlying equational theory between prefixes is not used in the proof procedure, but only in the proof of correctness and completeness theorems (an idea also used for matrix proof methods, see [30]). Instead, we define rules to specify which formulae may be moved forward to the next close possible world, as sequent-like tableaux do, or backward, as they cannot do. The name "single step" has been used since the reduction of a formula, with a given prefix, makes only use of immediately accessibile prefixes, without caring of those accessible via the transitive, symmetric etc. closure.

The intuition is very close to "visa rules", proposed for first order modal logic in Labelled Deductive Systems [9]: each logic grants a different "visa" to a constant for moving from one world to another. Why can different logics not grant different visas to formulae?! Our SST can thus be viewed as a particular labelled deductive system. In this way, the final user must only specify what kind of knowledege (modal formulae) can be moved back and forth among possible worlds and does not have to know anything about transitivity, euclidity and so on. Understanding proofs becomes easier.

Moreover, going "step by step" does not reveal to be an obstacle: a "hidden" property of prefixed tableaux (a kind of proof theoretical counterpart of Generation Theorem by Segerberg [17]) is used to get completeness. Thus, the method gains a lot of efficiency, since only a small part of the worlds' graph must be taken into account for each reduction. Still, it is human oriented enough for doing proofs by hand.

The SST-calculus provides other advantages: it offers the same flexibility of Hilbert-style proof methods (to change logic we have to add or remove a rule), and may be used to obtain countermodels for non-valid formulae. Indeed, at the end of a failed proof, prefixes form a minimal spanning tree of the underlying frame. The transitive, reflexive etc. closure of the accessibility relation may be done afterwards, if needed. Moreover, being a tableau system, it can be used to establish either satisfiability or validity.

Other properties of our formal system are also of great interest from the point of view of automated deduction and practical implementation. SST rules satisfy the strong Church-Rosser property i.e. the calculus is strongly confluent. Hence, at any stage of the computation, the execution sequence of applicable rules may be changed in an arbitrary fashion. Therefore SST can be a good basis for a flexible, i.e. compatible with many search heuristic, and highly parallelizable implementation. In practice, any formula could be assigned to a different processor, provided processors exchange information to keep prefix naming consistent.

1.2 Clusters and Contextual Reasoning

A proposal to enhance standard Kripke models for modal logic with clusters has been made in [6]. The main idea was to introduce as many accessibility relations as the planned number of clusters. Thus, to conclude that $\mathbf{K}\phi$ is true we just need to find *some* accessibility relation such that ϕ holds in all possible worlds accessible *through that particular relation*. The main aim of [6] was different from ours: they were mainly interested in modelling agents with limited capabilities and lack of logical omniscience. Since local reasoning was introduced to get rid of the axiom of logical ominscience, sentences like $\mathbf{K}(\phi \rightarrow \psi) \wedge \mathbf{K}\phi \wedge \neg\mathbf{K}\psi$ became satifisable.

However, if we are intersted in local reasoning for its own sake, such system seems far too weak. In a certain context we may state that we believe ϕ (i.e. that $\mathbf{K}\phi$ is true), while, in a different context, we may well affirm that $\mathbf{K}\neg\phi$ is the case. However, nobody will ever trust someone uttering sentences like $\mathbf{K}\phi \wedge \mathbf{K}\neg\phi \wedge \neg\mathbf{K}\bot$, which is satisfiable in [6]. So, we would like the set $\{\mathbf{K}\phi, \neg\mathbf{K}\phi\}$ to be k-cluster satisfiable for a certain k, but not the sentence $\mathbf{K}\phi \wedge \neg\mathbf{K}\phi$: classically inconsistent set could be k-cluster satisfiable whereas classically inconsistent formulae should remain unsatisfiable for any k.

Therefore, our approch is largely different from those of paraconsistent logics [1] where tableau calculi which deal with inconsistent formulae are proposed. In that case inconsistency is resolved by using two different negations and the concept of safe knowledge [2, 3].

Our standpoint, w.r.t. the proof theory, is closer to [12], where clusters are also used to solve the problem of logical omniscience. In that framework k frame of minds are devised by using labelled formulae: $\langle \phi, i\rangle$ means that ϕ belongs to the i-th context. Classical sequent rules are only used for formulae belonging to the same context, whereas special bridge rules are used to move formulae from one context to another. Typically, bridge rules are used to model some form of metareasoning i.e. if $\langle \phi : i\rangle$ holds then $\langle \mathrm{Theor}('\phi') : i+1\rangle$ holds too [11, 12]. This use of contexts is extremely flexible, since different contexts can use formulae belonging to different languages or apply different rules. However, such a radical changes [11] or such a rich formalism [12] makes modal proofs less easy to read (a drawback in common with indirect methods). Their greater flexibility is thus necessary only if metaprogramming features are essential; something that we do not require.

The key point of our work is that we don't want to make radical changes either to the semantics (the usual Kripke model of possible worlds) or to the proof theory of SST. We claim that a great deal of context reasoning can be done in the traditional framework, by a suitable use of its hidden features.

To enhance the proof theory, we need to bring forward an intuition from Fitting [7, 8]: when performing modal deduction one has to cope with a set of global assumptions G, a set of local assumptions U and the formula ϕ, to be proved valid or satisfiable. Intuitively, formulae from G are supposed to hold on every possible world, whereas formulae from U just in the worlds where ϕ holds. Thus, when a traditional tableau (sequent-like, prefixed or SST) starts to check the satisfiability of ϕ, the formulae of U are all assigned to the initial world $\langle 1\rangle$. In this way, *it is implicitly assumed that all local assumptions belong to the same cluster*! This fact (used to prove the completeness of standard SST) is the key to the introduction of cluster reasoning: we need to give it up and to assign formulae from U to different initial worlds $\langle i\rangle$ with $i \leq k$.

From a semantical viewpoint, even less work is needed. After all, Kripke frames are graphs, and graphs have already something corresponding exactly to non interacting frames of mind: just *take a graph with k connected parts (or clusters)*. Once we have matched a cluster on a graph with the corresponding context of a tableau, we have made *k-Clusters Tableaux* (k-CT).

1.3 Plan of the Paper

In this paper we just sketch the main results with some important tips. For details and further readings we refer to [20, 21] for Single Step Tableaux and [20] for k-Cluster Tableaux. In Section 2 we give some preliminary notions about modal logics: Kripke semantics, validity and satisfiability, prefixed tableaux etc. Section 3 presents the rules of SST-calculus, and shows some examples, to give a flavour of the calculus. In the same section we present the major computational properties of SST viz. locality principle, rules commutativity and decidability. Clusters are finally introduced in Section 4, with an intuitive example. The revision of SST machinery, to perform the local reasoning we want, is done in the same section and the new properties of the system are discussed. Related works and conclusions are presented in Section 5.

2 PRELIMINARIES

2.1 Modal Logics

Familiarity with the usual definitions of modal logics is assumed (see [15, 17] for an introduction). For sake of simplicity, we will build a formula from propositional letters by the usual connectives $\vee, \neg, \mathbf{K}$, whereas the othes connectives may be seen as abbreviations. In [20, 21] a more general and compact notation due to Smullyan and Fitting [7, 8] is used. In the sequel ϕ, ψ will range over modal formulae.

The semantics of modal logic is the usual one, based on *frames* i.e. pairs $\langle W, \mathcal{R}\rangle$, where W is a non empty set (of possible worlds) and $\mathcal{R}$ is a relation over W. Different modal logics may be obtained by varying $\mathcal{R}$. For example, if $\mathcal{R}$ is reflexive and transitive we have the well known logic S4 of knowledge and positive introspection, whereas if $\mathcal{R}$ is transitive and euclidean we have the logic K45 of positive and negative introspection.

Once again, the purpose of this work is not to choose among these logics, but to develop a proof system for each alternative. Thus, to keep exposition as compact and parametric as possible, we refer to L-frames, L-models or L-tableaux etc. when referring to frames, models or tableaux etc. for a particular logic L, viz K, D, T, KB, K4, K5, K45, KDB, D4, KD5, KD45, B, S4, S5, OM, OB, OK4, OS4, OM$^+$, OB$^+$, OK4$^+$, OS4$^+$. A summary table of the corresponding frame properties can be found in [21]. More details can be found in [15, 17].

A L-model is thus a triple $\langle W, \mathcal{R}, \Vdash\rangle$ where $\langle W, \mathcal{R}\rangle$ is a L-frame and $\Vdash$ is a relation between possible worlds and modal formulae such that,

$$
\begin{array}{lll}
w \Vdash \top & \text{and} & w \nVdash \bot \\
w \Vdash \phi \vee \psi & \text{iff} & w \Vdash \phi \text{ or } w \Vdash \psi \\
w \Vdash \neg\phi & \text{iff} & w \nVdash \phi \\
w \Vdash \mathbf{K}\phi & \text{iff} & \forall w^* \in W : \\
 & & w\mathcal{R}w^* \Rightarrow w^* \Vdash \phi \\
w \Vdash \neg\mathbf{K}\phi & \text{iff} & \exists w^* \in W : \\
 & & w\mathcal{R}w^* \& w^* \Vdash \neg\phi
\end{array}
$$

Validity and satisfiability of a formula in a L-model are given in the usual way, using local and global assumptions (sets of modal formulae):

Definition 1 (Validity). A formula ϕ is L-valid with local assumptions U and global assumptions G, i.e. $G \models_{\mathrm{L}} U \Rightarrow \phi$, if and only if, in every L-model $\langle W, \mathcal{R}, \Vdash\rangle$, such that (i) for every $v \in W$ it is $v \Vdash G$, and (ii) for every $w \in W$ if $w \Vdash U$ then $w \Vdash \phi$.

For short we write $w \Vdash S$ instead of $\forall \psi \in S : w \Vdash \psi$.

2.2 Prefixed Tableaux

SST use *prefixed formulae* i.e. pairs $\langle \sigma : \phi\rangle$ where σ is a non empty sequence of integers called *prefix* and ϕ is a modal formula. Intuitively σ "names" the possible world where ϕ holds. In the sequel, σ, σn or $\langle n_1 n_2 \ldots n_p\rangle$, with $p \geq 1$, are prefixes.

The definitions of branch and tableau are equal (but reduction rules!) to those of prefixed tableau from Fitting [7]. The definition of closure and termination is equally simple: a branch is *closed* if there are contradictory prefixed formulae on that branch (i.e. there is a σ such that, for some ψ, both $\langle \sigma : \psi\rangle$ and $\langle \sigma : \neg\psi\rangle$ are present on the branch or either $\langle \sigma : \bot\rangle$ or $\langle \sigma : \neg\top\rangle$ is present); a tableau is closed if every branch is closed. *Termination* occurs when no operation is possible. A branch is *open* if it is terminated and not closed and a tableau is open if at least one branch is such.

Definition 2 (Proof). Given global assumptions G and local assumptions U, a *proof* in the logic L for the modal formula ϕ, i.e. $G \vdash_{\mathrm{L}} U \Rightarrow \phi$, is the tableau starting with $\langle 1 : \neg\phi\rangle$ and closed by using SST-rules for logic L with global assumptions G and local assumptions U.

Intuitively a tableau proof of ϕ is a failed attempt to prove that $\neg\phi$ is satisfiable. If the tableau closes, e.g. because $\langle \sigma : \bot\rangle$ is present on the branch, then assuming $\neg\phi$ leads to a contradiction in some possible world (the one "tagged" by σ) and thus ϕ must be valid. To check that ϕ is *satisfiable*, one needs to start with $\langle 1 : \phi\rangle$ and to end with an open tableau.

Remark. Sequent like tableaux may need many tableaux to prove a formula satisfiable, whereas here *exactly one* tableau is used for both satisfiability and validity.

$$\alpha : \frac{\sigma : \neg(\phi \vee \psi)}{\begin{array}{c}\sigma : \neg\phi \\ \sigma : \neg\psi\end{array}}$$

$$\beta : \frac{\sigma : \phi \vee \psi}{\sigma : \phi \quad | \quad \sigma : \psi}$$

$$\text{dneg} : \frac{\sigma : \neg\neg\phi}{\sigma : \phi}$$

Figure 1: Rules for Propositional Connectives

$$Loc : \frac{\vdots}{1 : \psi} \quad \text{if } \psi \in U$$

$$Glob : \frac{\vdots}{\sigma : \psi} \quad \text{if } \sigma \text{ is present on the branch and } \psi \in G$$

$$\pi : \frac{\sigma : \neg\mathbf{K}\phi}{\sigma n : \neg\phi} \quad \text{with } \sigma n \text{ unrestricted on the branch}$$

Figure 2: Modal-rules Common to All Logics

3 SINGLE STEP TABLEAUX

To deal with the usual propositional connectives, traditional rules are used, as shown in Fig. 1.

Figure 2 presents the modal rules which are common to all modal logics. These are also identical to those used in [7] for prefixed tableaux. These rules make possible to add local or global assumptions and to reduce $\neg\mathbf{K}\phi$ formulae. The latter is also known as π-rule, where π stands for possibility. In the case of the π-rule some more terminology is needed: a prefix is *unrestricted* on a branch ϑ, if it is not the initial part of any other prefix already present on the branch (intuitively σn must be a new prefix).

Finally, Fig. 3 (page 6) shows the rules which may be used to reduce $\mathbf{K}\phi$-formulae. Those are also called ν-formulae, where ν stands for necessity.

Remark. Each rule does not depend on a particular logic nor on some accessibility relation over prefixes: it just states which formulae (i.e. knowledge or belief) may be "passed" to some close neighbours.

In this way each rule has a simple epistemic interpretation. For instance 4 may be seen as the rule for "Belief-increasing"-logics i.e. such that beliefs are preserved in the passage to a possible world. On the contrary 4^R is restrictive and states that knowledge may only decrease, since we can take back our knowledge to the actual world. Therefore the interpretation of K45 as the logic in which all possible worlds have the same beliefs is natural. A final user does not have anymore to worry about transitivity, euclidity and so on.

By choosing a suitable collection of rules, we can get the desired logic, with the same flexibility as Hilbert-style axiomatizations. The correspondence between ν-rules and modal logics is shown in Table 1.

3.1 Examples

The example in Fig. 4, for the logic K4, should better clarify the key difference between our SST and prefixed tableaux (as devised in [7]).

In K4 prefixed tableaux á la Fitting, one has to prove that $\langle 1111\rangle$ is a prefix accessible from $\langle 1\rangle$ in the logic K4 and that $\langle 1111\rangle$ already occurs on the branch. To reduce the formula $\langle\sigma : \mathbf{K}\phi\rangle$, it is thus required to have access to all prefixes σ^* which satisfy a suitable relation $\sigma \rhd \sigma^*$, and which may be very different from the current prefix! Therefore, the efficiency of the system is severly limited. In K4-SST, $\langle 1\rangle$ only communicates with $\langle 11\rangle$, and the latter one only with $\langle 111\rangle$ etc. However, a Generation Lemma ensures us that, if a "far" prefix occurs on a branch, then all intermediate prefixes must occur too: if $\langle\sigma nmk\rangle$ occurs then $\langle\sigma nm\rangle$, $\langle\sigma n\rangle$ and $\langle\sigma\rangle$ also do. Thus, with the 4-rule we can reach the right prefix. Hence, an SST *can do step by step what a prefixed tableaux á la Fitting needs to do in a burst.* So, when a formula of the type $\langle\sigma : \pi\rangle$ is reduced, and a new prefix is introduced, the ν-formulae, which need to be reduced again, are only those with the same prefix σ. Far away formulae can be

prefixed K4
$1 : \mathbf{K}\phi$
$\Downarrow$
$1111 : \phi$ since $\langle 1\rangle \rhd_{\mathrm{K4}} \langle 1111\rangle$

K4-SST
$1 : \mathbf{K}\phi$
$11 : \mathbf{K}\phi$ $4 - \text{rule}$
$111 : \mathbf{K}\phi$ $4 - \text{rule}$
$1111 : \phi$ $K - \text{rule}$

Figure 4: Prefixed Tableau and SST for logic K4

$$K : \frac{\sigma : \mathbf{K}\phi}{\sigma n : \phi} \qquad D : \frac{\sigma : \mathbf{K}\phi}{\sigma : \neg\mathbf{K}\neg\phi} \qquad T : \frac{\sigma : \mathbf{K}\phi}{\sigma : \phi}$$

$$B : \frac{\sigma n : \mathbf{K}\phi}{\sigma : \phi} \qquad 4 : \frac{\sigma : \mathbf{K}\phi}{\sigma n : \mathbf{K}\phi} \qquad 4^R : \frac{\sigma n : \mathbf{K}\phi}{\sigma : \mathbf{K}\phi}$$

$$T^D : \frac{\sigma n : \mathbf{K}\phi}{\sigma n : \phi} \qquad B^D : \frac{\sigma nm : \mathbf{K}\phi}{\sigma n : \phi} \qquad 4^D : \frac{\sigma n : \mathbf{K}\phi}{\sigma nm : \mathbf{K}\phi}$$

Note. σ, σn and σmn must all be already present on the branch, i.e. some π-rules must have introduced them

Figure 3: SST-rules for ν Formulae

Logics	ν-SST-rules
K	K
KB	K + B
K4	K + 4
K5	K + 4^D + 4^R
K45	K + 4 + 4^R

Logics	ν-SST-rules
T	K + T
B	K + T + B
S4	T + 4
S5	T + 4 + 4^R
KD*	K*-rules + D

Logics	ν-SST-rules
OM	K + T^D
OB	K + T^D + B^D
OK4	K + 4^D
OS4	K + T^D + 4^D
O*+	O*-rules + D

Table 1: Modal Logics and SST-rules

left sleeping in secondary memory.

Since SST-rules only require access to immediate neighbours of the current prefix, they boost efficiency without affecting completeness.

To get a flavour of the calculus, a simple proof for the axiom of negative introspection in logic K5 is given in Fig. 5 (page-7). Negative information (i.e. $\neg\mathbf{K}\phi$) is used at steps (4) and (5), to create two new "possible worlds" $\langle 11 \rangle$ and $\langle 12 \rangle$. Positive knowledge is then used at step (7), since K5 allows the transmission of information back from a prefix (a possible world) to its parent. A K-step is finally performed at (8).

SST calculus reflects in the proof itself the almost[1] symmetric nature of positive (K4) and negative introspection (K5): in the former case knowledge may be inherited downward from a possible world to its offsprings, whereas in the latter it is transmitted from a world to its ancestors upward.

3.2 Completeness and Computational Features

A simple completeness proof for prefixed tableaux, yet restricted to non euclidean logics, has been given in [7]. The same ideas may be applied to SST:

1. apply a systematic procedure to the tableau;
2. if the algorithm terminates (no rule may be further applied) and the tableau does not close, then any open branch is saturated for the SST;
3. if the algorithm never terminates then, by König Lemma, there is an infinite branch that will never close, even "ad infinitum" [7]. This branch can be chosen as the saturated one;
4. prove that a branch saturated with the SST rules is also saturated with the prefixed rules of Fitting;
5. prove a strong model existence theorem, by building a model with the saturated branch as in [7]. Indeed the property proved at 4, with some additional lemmas, allow us to define a suitable relation among prefixes that is a "carbon copy" of the relation among possible worlds in a frame.

Then we can prove the following:

Theorem 1 (Strong Completeness). *If ϕ is* L*-valid with G as global assumptions and U as local assumptions, i.e. $G \models_{\mathrm{L}} U \Rightarrow \phi$, then ϕ has a* SST *proof for* L.

To prove the completeness theorem for SST, intermediate lemmas are needed and one (necessary for step 4 above) is particularly interesting:

[1] They are not exactly symmetric, as can be seen from Table 1, since rule-4 is sufficient for K4 logic whereas rule-4^R alone is not enough for K5.

(0)	$1 : \neg(\neg\mathbf{K}\phi \to \mathbf{K}\neg\mathbf{K}\phi)$	negated theorem
(1)	$1 : \neg(\neg\neg\mathbf{K}\phi \vee \mathbf{K}\neg\mathbf{K}\phi)$	from $\to$ to $\vee$
$(2a)$	$1 : \neg\neg\neg\mathbf{K}\phi$	$\alpha -$ rule at (1)
$(2b)$	$1 : \neg\mathbf{K}\neg\mathbf{K}\phi$	
(3)	$1 : \neg\mathbf{K}\phi$	dneg-rule at $(2a)$
(4)	$11 : \neg\phi$	$\pi -$ rule at (3) "create" a new world $\langle 11\rangle$
(5)	$12 : \neg\neg\mathbf{K}\phi$	$\pi -$ rule at $(2b)$ "create" a new world $\langle 12\rangle$
(6)	$12 : \mathbf{K}\phi$	dneg-rule at (5)
(7)	$1 : \mathbf{K}\phi$	Knowledge backward to $\langle 1\rangle$, $4^R -$ rule at (6)
(8)	$11 : \phi$	Knowledge forward with $K -$ rule at (7)
	Contradiction in $\langle 11\rangle$	tableau closes

Figure 5: Proof of negative introspection axiom $\neg\mathbf{K}\phi \to \mathbf{K}\neg\mathbf{K}\phi$ in K5

Lemma (Generated Submodel Property). *If ϑ is a branch of a tableau then it satisfies the generated submodel property i.e.*

1. *for any prefix $\langle n_0 n_1 n_2 \ldots n_k\rangle$ present in ϑ, all "ancestors" $\langle n_0 n_1 \ldots n_i\rangle$ must be in ϑ, with $i = 0, 1 \ldots k$*
2. *there is* exactly one *prefix $\langle i\rangle$ in ϑ such that, for every prefix $\langle n_0 n_1 \ldots n_i\rangle$ in ϑ, it is $n_0 \equiv i$.*

Remark. This lemma formally confirms our intuition: SST, prefixed tableaux or labelled deductive systems implicitly use one and only one cluster for their proofs. To overcome this limitation k-cluster tableaux are introduced in Section 4.

SST calculus satifies many other useful properties that make it particularly suited for automation. In the previous section, we have already noted that SST obeys the locality principle i.e. *only few data items at a time must be taken into account.* This is an extremely desiderable property from a computational point of view, since data can be left in secondary storage to be retrieved only when necessary.

Moreover, being a tableau system, we can design a core algorithm that that does not depend on the query type (i.e. satisfiability or validity). Indeed, the core algorithm could only receive global assumptions G and local assumptions U_{core}, and should only answer either OPEN or CLOSED. The task of providing the right U_{core} (from the original U and the query ϕ), and of interpreting the answer may be given to a separate front-end. This one has only to set $U_{\text{core}} = U \cup \{\phi\}$ for satifiability (and answers YES if the tableau is open) or $U_{\text{core}} = U \cup \{\neg\phi\}$ for validity (answers YES if the tableau is closed).

A key advantages is that it satisfies the Church Rosser property i.e. at any point of the computation the order of applicable rules may be commuted in arbitrary fashion. This gives us a system that is extremely flexible and adaptable to many different search heuristics.

Theorem 2 (Church/Rosser). *The* SST *calculus is strongly confluent up to isomorphic renaming of prefixes.*

The proof is a simple application of Newman's theorem for rewrite systems [16]. An important feature, shown by this proof, is that only ν-formulae must be reusable; the others can be deleted after reduction.

Decidability, important for a practical implementation, requires more work but can also be proved [7, 21].

Theorem 3 (Decidability). *If a tableau branch is infinite, it must contain a "growing chain" of prefixes and, after a finite number of steps, the chain becomes "periodical" i.e. formulae are just repeated with a longer prefix.*

Thus, even if the underlying countermodel is potentially infinite, an algorithm can check for this periodicity and terminate in any case. If we directly apply this lemma the number of steps necessary to realize a branch cannot be closed will be exponential in the size of $G \cup U \cup \{\phi\}$. Fortunately, smarter conditions have been devised in [20].

Remark. SST can be easily extended to cope with resource bounded reasoning. For instance, if an agent has only a limited power of positive introspection, the applicability of rule 4 can be restricted to prefixes of lenght shorter than the fixed bound of introspection.

4 K-CLUSTERS TABLEAUX

Before giving technical details, we want to give a intuitive flavour of the system with an example of local reasoning due to Halpern et al. Let's consider a politician dealing with the tradeoff between balanced budgets (bb) and increase in spending (is). S/he is globally aware that an increase in spending does not usually bring a balanced budget, i.e. $G = \{\mathbf{K}(is \to \neg bb)\}$. However, s/he may well believe, yet in different frames of minds, that it is necessary to have a balanced budget and that it is necessary to greatly increase spending, i.e. $U = \{\mathbf{K}bb, \mathbf{K}is\}$. We also assume that he (or she) is at least able to reject contradictory beliefs i.e. s/he reasons at least with modal logic KD.

If we try to solve this problem in SST (i.e. 1-cluster tableaux) it's easy to show that the tableau starting with $\langle 1 : \mathbf{K}bb\rangle$ and $\langle 1 : \mathbf{K}is\rangle$ closes and thus U is not satisfiable. This is not the case if we pass to 2-clusters tableaux: we can assign $\mathbf{K}bb$ to $\langle 1\rangle$ and $\mathbf{K}is$ to $\langle 2\rangle$. It's then an easy job to show that, using all available rules for KD (with the exceptions of **Loc**), the tableau below is open and thus U is 2-clusters satisfiable[2].

$$\begin{array}{lll} (1a) & 1 : \mathbf{K}bb & 1 - \mathrm{Loc} \\ (1b) & 2 : \mathbf{K}is & 2 - \mathrm{Loc} \end{array}$$

4.1 Rules and Definitions

To revise semantics, we need to formalize the intuition of a cluster as set of connected worlds in a Kripke frame.

Definition 3 (Cluster). Given a frame $\mathcal{F} = \langle W, \mathcal{R}\rangle$, let $\sim$ be the reflexive, symmetric and transitive closure of $\mathcal{R}$, a *cluster* is any of the set of possible worlds $\mathcal{C} \in \mathrm{Cluster}(\mathcal{F})$ where

$$\begin{array}{rcl} \mathrm{Cluster}(\mathcal{F}) & \doteq & \{[w] : w \in W\} \\ {[w]} & \doteq & \{\, v \in W \; : \; w \sim v \,\} \end{array}$$

Therefore a *k-clusters frame* is just a frame $\mathcal{F}$, where $|\mathrm{Cluster}(\mathcal{F})| \leq k$. Similarly, a L-model with k clusters will be a model based on a frame with k clusters. The definition of k-clusters satisfiability is straightforward:

Definition 4 (k-cluster satisfiability). Given the global assumptions G, the local assumptions U are k-cluster satisfiable for the logic L if and only if there is a k-cluster model $\langle W, \mathcal{R}, ||\!\!-\rangle$ for the logic L such that

- for every $w \in W$ it is $w||\!\!-G$ and
- there are h cluster representatives $[w_1], [w_2], [w_h] \in \mathrm{Cluster}(\mathcal{F})$ with $h \leq k$ such that
- for any $\psi \in U$ there is a w_i, with $i \in \{1 \ldots h\}$ so that $w_i||\!\!-\psi$.

[2] This also gives us the possiblity to conclude that it is unfair to claim politicians inconsistent: they are doing perfect mathematical reasoning with k-clusters tableaux.

This corresponds to the intuition that formulae from U may belong to different non-communicating frames of minds: U is split between (at most) k different possible worlds but each of them is completely unrelated to the others. So, any formula true in one of them cannot affect, via a suitable combination of modal connectives, formulae valid in the others. At the same time, we still assume that formulae from G must hold everywhere.

This property is easily reflected in the proof theory. Indeed, the only rule which need to be revised is the **Loc**-rule. Beforehand, some more terminology is needed: a *k-gambled assignment* is a mapping of formulae from U into k sets[3] $\Gamma_1, \ldots \Gamma_k$ such that $U = \bigcup_{i=1}^{n} \Gamma_i$. Intuitively, each Γ_i corresponds to the formulae that are supposed to hold in the i-th cluster.

Then SST rules can be revised to give k-CT rules:

1. devise a k-gambled assignment for U,
2. replace the Loc-rule with the following k rules, for $i \in \{1 \ldots k\}$

$$i - Loc : \quad \frac{\vdots}{i : \psi} \qquad \text{if} \quad \psi \in \Gamma_i$$

3. all other SST rules for the logic L remain untouched.

It is worth noticing that, when checking k-clusters satisfiability for a formula ϕ, we just have to find a suitable i such that the tableau starting with $\langle i : \phi\rangle$ is open. Hence, as for usual SST satisfiability, one adds ϕ to local options U and carries the proof as if he had a set U_{core}. Therefore, we may give the following

Definition 5 (k-CT Sat Proof). Given the global assumptions G, a satisfiability proof for the local assumptions U is the open tableau starting with a k-gambled assignments for U and terminated by using k-CT rules for the logic L.

It's immediate to see that SST (and prefixed tableaux á la Fitting) are just 1-CT. Hence our system subsumes the case of SST as a particular case in a more general framework.

However, in most practical cases, not all possible gambled assignments make sense. Consider again the incoherent knowledge base built from different chunks of (locally coherent) data: the only significant assignment is that one which brings together all formulae belonging to the same chunk. In practice, the assignment of formulae to different clusters will be done by a preprocessor on a semantical basis, and many assignement will be ruled out automatically.

k-CT allow us to go further and to devise a natural idea of *local validity*: once an assignment has been fixed,

[3] Some may be empty or not disjoint.

it makes sense to inquire if a formula is valid in a particular cluster. With k-CT this can be easily done: if we want to prove that ϕ is valid in the i-th cluster, we just need to start a tableau with $\langle i : \neg\phi \rangle$ and terminate with a closed tableau.

4.2 Completeness and Other Features

The correctness and completeness proof of standard SST carries over to k-CT with minor changes. Correctness requires to enhance the proof with a suitable mapping between the sets of local options $\Gamma_1, \ldots, \Gamma_k$ (in which U is split) and the clusters of a k-cluster Kripke model. To get completeness, we need to show that an open k-CT can be used to build a Kripke model (as the standard SST completeness proof does) and that the resulting model has indeed k clusters.

Theorem 4 (Soundness and Completeness). *Given the global assumptions G, the local assumptions U are k-clusters satisfiable if and only if there is a k-CT satisfiability proof for G and U.*

Remark. It's worth noticing the different use of "," w. r. t. classical logic. In classical logic, a set of facts (e.g. a knowledge base) is always considered as a coherent whole and therefore $\{\mathbf{K}\phi, \neg\mathbf{K}\phi\}$ is not distinguishable from $\{\mathbf{K}\phi \wedge \neg\mathbf{K}\phi\}$. With k-CT, a set is really a simple collection of facts, and not necessarily a global coherent whole. Therefore, the previous two sets can be recognized as different ones. In our formalism, passing from "," to "$\wedge$" means that we have definitely committed ourselves to consider two facts into the same context.

The introduction of local validity and local satisfiability, once the splitting of U into the various Γ_i is fixed, can be done in the same way i.e.

Theorem 5. *Given the global assumptions G and k sets of local option Γ_i, ϕ is* locally valid *in the i-th cluster iff the k-clusters tableau starting with $\langle i : \neg\phi \rangle$ closes.*

A further advantages of this approach is that all SST computational features (Church-Rosser, decidability, etc.) are also carried over to k-CT with little or no difficulties. The corresponding theorems can then be stated by simply replacing SST with k-CT.

As noticed in [11], the key difference between Multicontext Languages [11, 12] and Labelled Deductive Systems [9] or prefixed tableaux [7] is the different role of labels. In ML systems, indices encode the metalevel information about reasoning location (i.e. the cluster), whereas in LDS labels keep track of the metalvel information about the derivation process (e.g. the number of premises used to derive a formula or the current possible world). In k-CT, these metalevel information (knowledge location and stage of the derivation process) are merged into one. Thus k-CT combine two different approaches of automated deduction.

5 RELATED WORKS AND CONCLUSIONS

The traditional, Hilbert-style deduction system for modal logics can be found in [15, 17] or in [7], where it is compared with other approaches. Among direct approaches, the most common deduction calculi are tableaux and sequents [7, 8, 13, 28], whereas a resolution procedure can be found in [5]. The main drawback of these methods is their limited efficency, especially when strong deducibility is at stake. Indeed, cut has been eliminated for K45 [13, 28] but only for weak deducibility (i.e. without premises). A Church-Rosser property, which is useful for flexible or parallel implementations, is also missing.

Model checkers are the tableuax for S4, K45 and S5 proposed in [15] and the TABLEAUX system in [4]. The latter has been implemented and treats many logics.

Among the indirect methods, an effective approach is functional translation, which allows one to use FOL matrix proof methods [30] or resolution [26, 27]. Unfortunately, this approach requires a rather heavy equational theory if ad hoc unification procedure are not possible (or not wanted). The best solution seems to be a mixed functional/relational translation [24, 25], which allows also a PROLOG implementation. A translation method based on theory resolution and restricted quantification can be found in [10]. A common problem affects these approaches: they all seem to need a "user-friendly" interface for proofs and proof search presentation (see [30] for further references).

Prefixed tableaux [7] and labelled deductive systems [9] stand in between: they add labels to modal formulae and define rules that take into account both formulae and labels. These approaches are more natural, can be used for both satisfiability and validity, and allow easily to present proof and proof search. Moreover, proofs obtained with a functional translation can be easily translated into the corresponding prefixed tableau proof [30]. Yet they share a drawback with functional translations: an equational theory is needed to deal with labels.

To devise a calculus without the axiom of logical omniscence, clusters accessibility relations has been introduced in [6]. However, the proposed calculus is very weak since sentences like $\mathbf{K}\phi \wedge \mathbf{K}\neg\phi \wedge \neg\mathbf{K}\bot$ become satisfiable in KD logics too.

The same problem has been tackled in [12] in the framework of multicontext languages (ML). This formalism is more powerful than traditional ones, since it allows various form of metareasoning. Modal logic can also be simulated in it [11]. ML, as devised in [11, 12], also uses labels to distinguish among different clusters, yet labels bring metalevel information different from those used in [7, 9] or for SST in [20, 21]. The intuition behind labels handling in ML is closer to our k-CT. However, devising a modal proof in the proposed formalisms is not easy, and the added complexity can be useful only when metareasoning features are essential.

An approach largely different from ours is based on paraconsistent logics [1], where inconsistent formulae are admitted. A tableau calculus and its related semantics can be found in [2, 3].

We have proposed a new tableau calculus that brings together the advantages of sequent-like and prefixed tableaux and allows an analytic, deterministic, reasonably efficient, strongly confluent, uniform and "user friendly" approach to many normal propositional modal logics, including symmetric and euclidean ones: K,T,D, KB, K4, K5, K45, KDB, D4, KD5, KD45, KD45, T, B, S4, S5, OM, OB, OK4, OS4, OM^+, OB^+, $OK4^+$, $OS4^+$ and many others.

The basic calculus is then extended with k-Clusters Tableaux to cover the case of inconsistent belief sets. k-CT are weak enough to describe knowledge bases where two different consistent statements (modal formulae) may contradict each other (in classical logic), but strong enough to rule out the possibility of an inconsistent statement. Thus, our system combines the full power of a modal theorem prover with a system of context handling, by merging the metalevel information used in the labelling system from [7, 9, 21] with those from [11, 12]. Therefore, k-Cluster Tableaux builds a bridge between two different approaches towards automated deduction.

ACKNOWLEDGEMENTS

This work is part of my Tesi di Laurea carried under the supervision of prof. L. Carlucci Aiello. I would like to thank her and Fiora Pirri for reading drafts of this paper and for their encouragement and support.

References

[1] A. Arruda. A survey of paraconsistent logic. In A. Arruda, N. C. A. Da Costa, and R. Chuaqui, eds, *Mathemathical Logic in Latin America*. North-Holland, 1980.

[2] W. A. Carnielli, L. Fariñas del Cerro, and M. Lima-Marques. Contextual negations and reasoning with contradictions. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence IJCAI-91*.

[3] W. A. Carnielli and M. Lima-Marques. Reasoning under inconsistent knowledge. *J. of Applied Non-Classical Logics*, 1992.

[4] L. Catach. TABLEAUX, a general theorem prover for modal logics. *J. of Autom. Reasoning*, 7, 1991.

[5] P. Enjalbert and L. Fariñas del Cerro. Modal resolution in clausal form. *Theoretical Computer Science*, 65, 1989.

[6] R. Fagin and J. Y. Halpern. Belief, awareness, and limited reasoning. *Artificial Intelligence*, 34, 1988.

[7] M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*. Reidel, 1983.

[8] M. Fitting. First order modal tableaux. *J. of Automated Reasoning*, 4, 1991.

[9] D. M. Gabbay. *Labelled Deductive Systems*. Oxford Univ. Press, Oxford, 1993.

[10] I. Gent. Theory matrices (for modal logics) using alphabetical monotonicity. *Studia Logica*, 52, 1993.

[11] F. Giunchiglia, L. Serafini. Multilanguage hierarchical logics or: How can we do without modal logics. *Artificial Intelligence*, 65, 1994.

[12] F. Giunchiglia, L. Serafini, E. Giunchiglia, M. Frixione. Non-omniscient belief as context-based reasoning. In *Proc. of IJCAI-93*. Morgan Kaufmann.

[13] R. Goré. *Cut-free Sequent and Tableaux Systems for Propositional Normal Modal Logics*. PhD thesis, Computer Laboratory, Univ. of Cambridge, 1992.

[14] J. Y. Halpern and R. Fagin. Modelling knowledge and action in distributed systems. *Distributed computing*, 3(4), 1989.

[15] J. Y. Halpern and Y. Moses. A guide to the modal logic of knowledge and belief. In *Proc. of IJCAI-85*.

[16] G. Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *J. of the ACM*, 1980.

[17] G. E. Hughes and M. J. Cresswell. *A Companion to Modal Logic*. Methuen, London, 1984.

[18] K. Konolige. On relation between default and autoepistemic logic. *Artificial Intelligence*, 35(3), 1988.

[19] W. Marek, G. F. Schwarz, and M. Truszczynski. Modal nonmonotonic logics: ranges, characterization, computation. In *Proc. of the 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning KR-91*, San Matteo, CA.

[20] F. Massacci. *Ragionamento autoepistemico automatico*. Tesi di Laurea, Univ. di Roma "La Sapienza", 1993.

[21] F. Massacci. Strongly analytic tableaux for normal modal logics. In *Proc. of the 12th Int. Conf. on Automated Deduction CADE-94*. Springer Verlag.

[22] J. McCarthy. Notes on formalizing context. In *Proc. of IJCAI-93*.

[23] R. C. Moore. Possible world semantics for autoepistemic logic. In M. Ginsberg, ed, *Readings on Nonmonotonic Reasoning*. Morgan Kaufmann, 1990.

[24] A. Nonnengart. First-order modal logic theorem proving and standard PROLOG. Tech. Rep. MPI-I-92-228, Max-Plank-Inst. für Informatik, Saarbrüken (D) 1992.

[25] A. Nonnengart. First-order modal logic theorem proving and functional simulation. In *Proc. of IJCAI-93*.

[26] H. J. Ohlbach. A resolution calculus for modal logic. In *Proc. of CADE-88*, *LNCS* 310. Springer Verlag.

[27] H. J. Ohlbach. Translation methods for non-classical logics - an overview. *Bulletin of the Interest Group in Pure and Applied Logic* - *IGPL*, 1(1). Max-Plank-Institute/Imperial College, 1993.

[28] G. Schwarz. Gentzen style systems for K45 and KD45. In A. Meyer and M. Taitslin, eds, *Logic at Botik '89, Symposium on Logical Foundations of Computer Science*, *LNAI* 363. Springer Verlag.

[29] R. C. Stalnaker. *Inquiry*. MIT-press, 1984.

[30] L. A. Wallen. Matrix proof methods for modal logics. In *Proc. of IJCAI-87*.