A new deduction system for deciding validity in modal logic K *

Golińska-Pilarek, J.[1], Muñoz-Velasco, E.[2] and Mora, A.[2]
[1] Institute of Philosophy, Warsaw University.
National Institute of Telecommunications, Poland.
j.golinska-pilarek@itl.waw.pl
[2] Dept. Matemática Aplicada, Universidad de Málaga, Spain.
{emilio, amora}@ctima.uma.es

June 30, 2010

Abstract

A new deduction system for deciding validity for the minimal decidable normal modal logic K is presented in this paper. Modal logics could be very helpful in modeling dynamic and reactive systems such as bio-inspired systems and process algebras. In fact, recently it has been presented the *Connectionist Modal Logics* which combines the strengths of modal logics and neural networks. Thus, modal logic K is the basis for these approaches. Soundness, completeness, and the fact that the system itself is a decision procedure are proved in this paper. The main advantages of this approach are: first, the system is deterministic, that is, it generates one proof tree for a given formula; second, the system is a validity-checker, hence it generates a proof of a formula (if such exists); third, the language of deduction and the language of a logic coincide. Some of these advantages are compared to other classical approaches.

Keywords: Relational Logic, Modal Logic, Dual Tableau Methods, Decision Procedures, Theorem Proving.

1 Introduction

Modal logics are widely applicable methods of reasoning for many areas of computer science. These areas include artificial intelligence, database theory, distributed systems, program verification, cryptography theory. Artificial neural networks exhibit many properties of intelligent systems, like being

*This work is partially supported by the Spanish research projects TIN2009-14562-C05-01 and P09-FQM-5233

massively parallel, context-sensitive, adaptable, and robust; while logic systems were designed to cope with structured objects and structure-sensitive processes and, consequently, it seems to be very interesting to combine the advantages of connectionist and logic systems in a single one [2, 18].

Neural-symbolic integration covers different types of reasoning: neural representations exist for modeling propositional logic, whole classes of many-valued logics, modal logic, temporal logic, and epistemic logic. Modal logics could be very helpful in modeling dynamic and reactive systems such as bio-inspired systems and process algebras. In fact, recently it has been presented a logic called *Connectionist Modal Logics*, which combines the strengths of modal logics and neural networks [1]. We think this approach could be useful for knowledge management, multi-agent systems, and case-based reasoning [7, 15, 16, 23].

On the other hand, the synergy of logical techniques and biological approaches can help to explain certain biological process. In [3], the logic used to reason in presence of incomplete specifications in biological problems, includes modal operators. As an example of the importance of modal logics in these ambiences, we emphasize, the Brane Logic [19], a modal logic for expressing formally properties about systems in Brane Calculus.

Modal logic K is the basis for the previous approaches. Since K is decidable [4], many efforts have been taken in order to develop effective and simple-to-use decision procedures for it [8, 20]. One of the most popular deduction systems for K is a tableau system. Generally, there are two kinds of tableau systems for modal logics: sequent-like and labeled ones. Both kinds of tableaux have been constructed for K. Tableau is a refutation system determined by the rules and axioms. In order to prove validity of a formula φ, one must built a closed proof tree for $\neg\varphi$. As far as we know, all tableaux for K include some non-deterministic rules. Thus, in general, the system may generate many trees for a valid formula and some of them may be non-closed, i.e., the system may generate trees which are not proofs of a valid formula. In order to prove that a formula is not valid, one must check if all possible trees are not closed. However, the system itself does not provide any method of finding all the trees, this is shifted to the strategy of a proof searching, which turns out to be an extension of the original tableau by means of special techniques, e.g., backtracking. In this sense, tableau systems for K are not decision procedures; they only provide a method of deciding whether a modal formula is valid or not. From a methodological point of view, this feature of tableaux for modal logics is of big importance, which is manifested through the implementation of the system in order to get a decision procedure.

We present a new deduction system for the logic K by proving its soundness and completeness and we show that the system itself is a decision procedure. The system is deterministic, that is, it generates only one proof tree for a given formula and does not need any *external* technique, such as backtracking, in order to get the desired proof tree. As far as we know, it is the first theorem prover for modal logic with these properties, which make it very interesting for computational purposes. More details about this, from the implementation point of view can be seen in [11]. It is also a validity checker, thus it provides a method of constructing a proof of a valid formula φ. Moreover, the language of deduction of the system in question and the language of the logic for which the system is constructed coincide, which is a further advantage of our approach.

The system presented is based on relational dual tableaux which are validity checkers [13, 14, 21]. They are extensions of Rasiowa-Sikorski diagrams for first-order logic [22]. The common language of all relational dual tableaux is the logic of relations. Usually, it is a classical logic of binary relations, which was introduced as a logical counterpart to the class RRA of representable relation algebras given by Tarski [24]. It has been studied systematically in the last decades. A relational logic enables us to represent within a uniform formalism the three basic components of any logical system: syntax, semantics, and deduction. Relational formalism provides a means of reasoning in a great variety of very different theories, e.g., modal, temporal, spatial, information, program, as well as intuitionistic, and many-valued logics. Therefore, the relational logic can be regarded as the generic framework supporting representations of many non-classical logics [21]. Relational dual tableaux are powerful tools for performing the four major reasoning tasks: verification of validity, verification of entailment, model checking, and verification of satisfaction. Moreover, it has an almost automatic way of transforming a complete dual tableau proof tree into a complete Gentzen sequent calculus proof tree [17]. In addition, this method can be used for non-classical extensions of the classical logic as well as for logics with non-classical semantics of classical connectives (e.g., intuitionistic logics).

The implementation of this prover has been presented in [11]. An implementation of the proof system for the relational logic is available in [6] while in [9] a translation procedure from non-classical logics to relational logic is presented. Moreover, in [5,12] there are implementations of relational logics for order of magnitude quatitative reasoning.

In Section 2, we present the modal logic K together with the relational logic appropriate for expressing formulas of K. Decision procedure for K, its soundness, completeness, and termination are given in Section 3. Conclusions and prospects of future work are presented in Section 4.

2 Relational formalization of logic K

We introduce the relational formalization of modal logic K. First, the modal logic is shown, second its relational translation and the equivalence of validity between a modal formula and its relational translation.

2.1 Modal logic K

We present the modal logic K, its syntax, its semantics, and a function which links the two. The language of logic K consists of the symbols from the following pairwise disjoint sets $\mathbb{V}$ - a countable infinite set of propositional variables; $\{\neg, \vee, \wedge\}$ - the set of the classical propositional operations of negation ($\neg$), disjunction ($\vee$), and conjunction ($\wedge$); $\{\langle R\rangle\}$ - the set consisting of modal propositional operation called the *possibility* operation. As usual, the modal operation $[R]$ of *necessity* is defined by $[R] \stackrel{\text{def}}{=} \neg\langle R\rangle\neg$. The set of K-formulas is the smallest set including the set of propositional variables and closed with respect to all the propositional operations.

A K-*model* is a structure $\mathcal{M} = (U, R, m)$, where U is a non-empty set (of states), R is a binary relation on U, and m is a meaning function such that $m(p) \subseteq U$, for every propositional variable $p \in \mathbb{V}$. The relation R is referred to as the *accessibility relation.* The *satisfaction relation* is defined inductively for propositional and boolean connectives as usual, and in addition, for every K-formula φ, $\mathcal{M}, s \models \langle R\rangle\varphi$ iff there exists $s' \in U$ such that $(s, s') \in R$ and $\mathcal{M}, s' \models \varphi$. A K-formula φ is said to be *true* in a K-model $\mathcal{M} = (U, R, m)$, $\mathcal{M} \models \varphi$, whenever for every $s \in U$, $\mathcal{M}, s \models \varphi$, and it is K-*valid* whenever it is true in all K-models.

2.2 Relational logic $\mathsf{RL_K}$

We define the relational logic, $\mathsf{RL_K}$, appropriate for expressing formulas of the modal logic K. The vocabulary of the language of the relational logic $\mathsf{RL_K}$ consists of the symbols from the following pairwise disjoint sets: $\mathbb{OV} = \{z_0, z_1, \ldots\}$ - a countable infinite set of object variables; $\mathbb{RV} = \{S_1, S_2, \ldots\}$ - a countable infinite set of relational variables; - the set consisting of the relational constant R representing the accessibility relation from K-models and $\{-, \cup, \cap, ;\}$ - the set of relational operations.

The set of relational terms, $\mathbb{RT}$, is the smallest set which includes $\mathbb{RV}$ and satisfies: If $P, Q \in \mathbb{RT}$, then $-P, P \cup Q, P \cap Q, (R; P) \in \mathbb{RT}$. $\mathsf{RL_K}$-formulas are of the form $z_i T z_j$, where z_i, z_j are object variables and T is any relational term.

An $\mathsf{RL_K}$-*model* is a structure $\mathcal{M} = (U, R, m)$, where U is a non-empty set, R is a binary relation on U, and m is a meaning function satisfying: $m(S) = X \times U$, where $X \subseteq U$, for every relational variable S; $m(R) = R$, i.e., R is the interpretation of the relational constant R; m extends to all the compound relational terms as follows: $m(-P) = (U \times U) - m(P)$; $m(P \cup Q) = m(P) \cup m(Q)$; $m(P \cap Q) = m(P) \cap m(Q)$; $m(R; P) = \{(x, y) \in U \times U : \exists z \in U\, ((x, z) \in R \wedge (z, y) \in m(P))\}$.
Let $\mathcal{M} = (U, R, m)$ be an $\mathsf{RL_K}$-model. A *valuation* in $\mathcal{M}$ is any function $v\colon \mathbb{OV} \to U$.

An $\mathsf{RL_K}$-formula $z_i T z_j$ is *satisfied* in an $\mathsf{RL_K}$-model $\mathcal{M}$ by a valuation v, $\mathcal{M}, v \models z_i T z_j$ whenever $(v(z_i), v(z_j)) \in m(T)$. A formula is *true* in $\mathcal{M}$ whenever it is satisfied by all the valuations in $\mathcal{M}$, and it is $\mathsf{RL_K}$-*valid* whenever it is true in all $\mathsf{RL_K}$-models.

The translation of K-formulas into relational terms starts with a one-to-one assignment of relational variables to the propositional variables. Let τ' be such an assignment. Then the translation τ of formulas is defined inductively as follows: $\tau(p) = \tau'(p)$, for any propositional variable $p \in \mathbb{V}$; $\tau(\neg\varphi) = -\tau(\varphi)$; $\tau(\varphi \vee \psi) = \tau(\varphi) \cup \tau(\psi)$; $\tau(\varphi \wedge \psi) = \tau(\varphi) \cap \tau(\psi)$; $\tau(\langle R \rangle \varphi) = (R; \tau(\varphi))$. The following theorem shows the desired result of preservation of validity via translation of formulas of modal logic into relational terms.

Theorem 1 *For every* K*-formula* φ*,* φ *is* K*-valid iff* $z_1 \tau(\varphi) z_0$ *is* $\mathsf{RL_K}$*-valid.*

Due to lack of space, the proofs of all the results presented in this paper can be seen in [10].

3 Decision procedure

Relational dual tableaux are founded on the Rasiowa-Sikorski system for the first order logic. They are powerful tools for performing the four major reasoning tasks: verification of validity, verification of entailment, model checking, and verification of satisfaction.

To begin with, we define an *order* of relational terms and relational formulas which will be useful in the rest of the paper. The *length* of a relational term T, $l(T)$ is defined as follows: $l(S) = 0$, for every relational variable S; $l(-T) = l(T) + 1$; $l(T \# T') = l(T) + l(T') + 1$, for $\# \in \{\cup, \cap\}$ and $l(R; T) = l(T) + 1$.

For $i = 0, 1$ and a relational term T, we define $-^i T = T$, if $i = 0$ and $-^i T = -T$, if $i = 1$.

The *type*, $t(T)$, of a relational term T is defined as follows. If a relational term is of the form $-^i S_j$, for $i = 0, 1$ and for some relational variable S_j, then it is said of type $(-^i S)$. If a relational term is of the form $--T$, for some relational term T, then it is said to be of type $(-)$. A relational term is said to be of type $(-^i \#)$, $\# \in \{\cup, \cap\}$, (resp. $(-^i ;)$) whenever it is of the form $-^i(P \# Q)$ (resp. $-^i(R; P)$), for some relational terms P and Q.

We define an ordering $<$ on the set of types as follows:

$$(S) < (-S) < (-) < (\cup) < (-\cap) < (\cap) < (-\cup) < (;) < (-;).$$

Now, the ordering $<$ of relational terms is defined inductively. We say that $T < T'$ if and only if either of the following possibilities holds:

(1) $t(T) < t(T')$, or

(2) T and T' are of the same type and $l(T) < l(T')$, or

(3) T and T' are of the same type and of the same length and satisfy either of the following: $j < k$, if $T = -^i S_j$ and $T' -^i S_k$, for some relational variables S_j, S_k and $i = 0, 1$; or $P < P'$, if $T = --P$ and $T' = --P'$ or $T = -^i(R; P)$ and $T' = -^i(R; P')$, for some relational terms P and P' and $i = 0, 1$; or $P < P'$ or both $P = P'$ and $Q < Q'$, if $T = -^i(P\#Q)$ and $T' = -^i(P'\#Q')$, for some relational terms P, P', Q, Q', $i = 0, 1$, and $\# \in \{\cup, \cap\}$.

We extend the ordering $<$ to all $\mathsf{RL_K}$-formulas as follows: $z_{k_1} T z_{k_2} < z_{l_1} T' z_{l_2}$ whenever either of the following conditions is satisfied: $k_1 < l_1$; or $k_1 = l_1$ and $T < T'$; or $k_1 = l_1$ and $T \leq T'$ and $k_2 < l_2$. Notice that the set of all $\mathsf{RL_K}$-formulas is well ordered by $<$.

Let X be a finite set of $\mathsf{RL_K}$-formulas, let $\# \in \{\cup, \cap\}$, and let $i = 0, 1$. A formula φ of type $(-^i\#)$ (resp. $(-^i;)$) is said to be *minimal with respect to X and* $(-^i\#)$ (resp. $(-^i;)$) whenever for every $\psi \in X$ of the same type as φ, $\varphi < \psi$. Similarly, a formula φ of type $(-)$ is said to be *minimal with respect to X and* $(-)$ whenever for every formula $\psi \in X$ of type $(-)$, $\varphi < \psi$.

3.1 The rules

Relational proof systems are determined by axiomatic sets of formulas and rules which apply to finite sets of relational formulas. The axiomatic sets take the place of axioms. The rules are intended to reflect properties of relational operations and constants. There are two groups of rules: decomposition rules and specific rules. Given a formula, the decomposition rules of the system enable us to transform it into simpler formulas, and the specific rules enable us to replace a formula by some other formulas. The rules have the following general form: $(*)\ \dfrac{X \cup \Psi}{X \cup \Phi}$ or $(**)\ \dfrac{X \cup \Psi}{X \cup \Phi_1 \,|\, X \cup \Phi_2}$, where $X, \Psi, \Phi, \Phi_1, \Phi_2$ are finite non-empty sets of formulas such that $X \cap \Psi = \emptyset$. A rule of the form (**) is a branching rule. In a rule, the set above the line is referred to as its *premise* and the set(s) below the line is (are) its *conclusion(s)*. A rule of the form $(*)$ (resp. $(**)$) is *applicable* to a finite set Y if and only if $Y = X \cup \Psi$ and $\Phi \not\subseteq X$ (resp. $\Phi_1 \not\subseteq X$ or $\Phi_2 \not\subseteq X$), that is an application of a rule must introduce a new formula.

Decomposition rules of $\mathsf{RL_K}$-dual tableau are $(\cup)$, $(\cap)$, $(-\cup)$, $(-\cap)$, $(-)$, (K_1), and (K_2) of the following forms. For every $k \geq 1$ and for all relational terms P and Q,

$$(\cup)\quad \frac{X \cup \{z_k(P \cup Q)z_0\}}{X \cup \{z_k P z_0, z_k Q z_0\}} \qquad\qquad (\cap)\quad \frac{X \cup \{z_k(P \cap Q)z_0\}}{X \cup \{z_k P z_0\} \,|\, X \cup \{z_k Q z_0\}}$$

$$(-\cup)\quad \frac{X \cup \{z_k -(P \cup Q)z_0\}}{X \cup \{z_k - P z_0\} \,|\, X \cup \{z_k - Q z_0\}} \qquad\qquad (-\cap)\quad \frac{X \cup \{z_k -(P \cap Q)z_0\}}{X \cup \{z_k - P z_0, z_k - Q z_0\}}$$

where $z_k -^i(P\#Q)z_0$ is minimal with respect to X and $(-^i\#)$, for $\# \in \{\cup, \cap\}$ and $i = 0, 1$.

$$(-) \quad \frac{X \cup \{z_k--Pz_0\}}{X \cup \{z_k P z_0\}} \quad \text{where } z_k--Pz_0 \text{ is minimal with respect to } X \text{ and } (-).$$

For all $k, l, m \geq 1$ and for all relational terms P_i, Q_j, $1 \leq i \leq j$, $1 \leq j \leq m$,

$$(\mathsf{K}_1) \quad \frac{X \cup \{z_k-(R;Q_1)z_0, \ldots, z_k-(R;Q_m)z_0\}}{X \cup \{z_{k_1}-Q_1 z_0, \ldots, z_{k_1+(m-1)}-Q_m z_0\}}$$

$$(\mathsf{K}_2) \quad \frac{X \cup \{z_k(R;P_i)z_0\}_{i \in I_l} \cup \{z_k-(R;Q_1)z_0, \ldots, z_k-(R;Q_m)z_0\}}{X \cup \{z_{k_1}P_i z_0, \ldots, z_{k_1+(m-1)}P_i z_0\}_{i \in I_l} \cup \{z_{k_1}-Q_1 z_0, \ldots, z_{k_1+(m-1)}-Q_m z_0\}}$$

being $I_l = \{1, \ldots, l\}$ and provided that: $-(R;Q_i) < -(R;Q_j)$, for all $1 \leq i < j \leq m$ and $z_k-(R;Q_1)z_0$ is minimal with respect to X and $(-;)$; $z_k(R;T)z_0 \notin X$ and $z_k-(R;T')z_0 \notin X$ for all terms T and T'; $k < k_1$ and k_1 is the minimal number such that z_{k_1} does not occur in X.

The specific rule of $\mathsf{RL_K}$-dual tableau is as follows. For all $k \geq 1$, $j \geq 0$ and for every relational variable S:

$$(\text{right}) \quad \frac{X \cup \{z_k S z_j\}}{X \cup \{z_k S z_l, z_k S z_j\}}$$

provided that k and l are minimal satisfying: $z_k S z_j < z_{k'} S' z_{j'}$, for every relational variable S' and for all object variables $z_{k'}$ and $z_{j'}$ such that $z_{k'} S' z_{j'} \in X$; z_l occurs $X \cup \{z_k S z_j\}$, $l \neq j$ and $z_k S z_l \notin X$.

3.2 Soundness and Completeness

A finite set of $\mathsf{RL_K}$-formulas is said to be an $\mathsf{RL_K}$-*axiomatic set* whenever it is a superset of $\{z_k P z_j, z_k-P z_j\}$, for some object variables z_k, z_j and for some relational term P. A finite set of $\mathsf{RL_K}$-formulas $\{\varphi_1, \varphi_2, \ldots, \varphi_n\}$, $n \geq 1$, is $\mathsf{RL_K}$-*valid* whenever for every $\mathsf{RL_K}$-model $\mathcal{M} = (U, R, m)$ and for every valuation v in $\mathcal{M}$ there exists $i \in \{1, \ldots, n\}$ such that $\mathcal{M}, v \models \varphi_i$. A rule of the form $(*)$ (resp. $(**)$) is $\mathsf{RL_K}$-*correct* whenever $\mathsf{RL_K}$-validity of $X \cup \Phi$ (resp. $\mathsf{RL_K}$-validity of $X \cup \Phi_1$ and $X \cup \Phi_2$) implies $\mathsf{RL_K}$-validity of $X \cup \Psi$.

Proposition 2 *The* $\mathsf{RL_K}$*-rules are* $\mathsf{RL_K}$*-correct and the* $\mathsf{RL_K}$*-axiomatic sets are* $\mathsf{RL_K}$*-valid.*

Given a formula, successive applications of the rules result in a tree whose nodes consist of finite sets of formulas. Each node includes all the formulas of its predecessor node, possibly except for those which have been transformed according to a rule. A node of the tree does not have successors whenever its set of formulas includes an axiomatic subset.

Let $z_1 T z_0$ be an $\mathsf{RL_K}$-formula, an $\mathsf{RL_K}$-*proof tree of* $z_1 T z_0$ is a tree with the following properties: $z_1 T z_0$ is at the root of this tree; each node except the root is obtained by an application of a rule to its predecessor node; the rules are applied with the following ordering: $(-)$, $(\cup)$, $(-\cap)$, $(\cap)$, $(-\cup)$, (right), (K_1), and (K_2) and a node does not have successors whenever its set of formulas is an $\mathsf{RL_K}$-axiomatic set or none of the rules is applicable to its set of formulas.

A branch of an $\mathsf{RL_K}$-proof tree is *closed* whenever it contains a node with an $\mathsf{RL_K}$-axiomatic set of formulas. An $\mathsf{RL_K}$-proof tree is closed if and only if all of its branches are closed. An $\mathsf{RL_K}$-formula z_1Tz_0 is $\mathsf{RL_K}$*-provable* whenever there is a closed $\mathsf{RL_K}$-proof tree of it, which is then referred to as its $\mathsf{RL_K}$*-proof.*

The previous Proposition ensures the desired transference of validity from the bottom to the top of a proof tree. As a consequence of this, we have:

Proposition 3 (Soundness) *If an* $\mathsf{RL_K}$*-formula* z_1Tz_0 *is* $\mathsf{RL_K}$*-provable, then it is* $\mathsf{RL_K}$*-valid.*

The rules of $\mathsf{RL_K}$-dual tableau guarantee that:

Proposition 4 *Every* $\mathsf{RL_K}$*-formula of the form* z_1Tz_0 *has exactly one finite* $\mathsf{RL_K}$*-proof tree.*

In order to obtain completeness, as usual in proof theory, a concept of a complete non-closed proof tree is needed. Intuitively, completeness of a non-closed tree means that all the rules that can be applied have been applied. By abuse of notation, for any branch b and any formula φ, we write $\varphi \in b$, if φ belongs to a set of formulas of a node of branch b. By $\mathbb{OV}_b$ we mean the set of all object variables that occur in formulas belonging to the branch b. A branch b of $\mathsf{RL_K}$-proof tree of an $\mathsf{RL_K}$-formula z_1Tz_0 is said to be *complete* whenever either it is closed or it satisfies the following completion conditions, for every $k \geq 1$ and for all relational terms P and Q:

Cpl($\cup$) (resp. Cpl($-\cap$)) If $z_k(P \cup Q)z_0 \in b$ (resp. $z_k-(P \cap Q)z_0 \in b$), then both $z_kPz_0 \in b$ (resp. $z_k-Pz_0 \in b$) and $z_kQz_0 \in b$ (resp. $z_k-Qz_0 \in b$);

Cpl($\cap$) (resp. Cpl($-\cup$)) If $z_k(P \cap Q)z_0 \in b$ (resp. $z_k-(P \cup Q)z_0 \in b$), then either $z_kPz_0 \in b$ (resp. $z_k-Pz_0 \in b$) or $z_kQz_0 \in b$ (resp. $z_k-Qz_0 \in b$);

Cpl($-$) If $z_k(--P)z_0 \in b$, then $z_kPz_0 \in b$;

For every $k \geq 1$ and for all relational terms P and Q,

Cpl(K_1) If $z_k-(R;Q)z_0 \in b$ and for every relational T, $z_k(R;T)z_0 \notin b$, then for some $z \in \mathbb{OV}_b$, $z-Qz_0 \in b$, obtained by an application of the rule (K_1);

Cpl(K_2) If $z_k(R;P)z_0 \in b$ and $z_k-(R;Q)z_0 \in b$, then for some $z \in \mathbb{OV}_b$, both $zPz_0 \in b$ and $z-Qz_0 \in b$, obtained by an application of the rule (K_2);

For every $k \geq 1$, for every $j \geq 0$, and for every relational variable S,
Cpl(right) If $z_kSz_j \in b$, then for every object variable $z \in \mathbb{OV}_b$, $z_kSz \in b$.

An $\mathsf{RL_K}$-proof tree is said to be *complete* whenever all of its branches are complete. It is easy to prove that the $\mathsf{RL_K}$-proof tree of an $\mathsf{RL_K}$-formula of the form z_1Tz_0 is complete.

The $\mathsf{RL_K}$-rules guarantee that whenever a branch contains formulas xSy and $x{-}Sy$, for some relational variable S and object variables x and y, then these formulas belong to the same node of the branch, hence branch is closed. Let b be a non-closed branch of $\mathsf{RL_K}$-proof tree of an $\mathsf{RL_K}$-formula z_1Tz_0, we define a branch structure $\mathcal{M}^b = (U^b, R^b, m^b)$ as follows: $U^b = \mathbb{OV}_b$; R^b is defined by $(z_k, z_j) \in R^b$ iff there exists a term Q such that $z_k{-}(R;Q)z_0 \in b$, $z_j{-}Qz_0 \in b$ and $z_j{-}Qz_0$ is obtained by an application either of rule (K_1) or (K_2); $m^b(S) = \{(z_k, z_j) \in U^b \times U^b : z_kSz_j \notin b\}$, for every relational variable S; m^b extends to all the compound relational terms as in $\mathsf{RL_K}$-models.

Proposition 5 (Branch Model Property) *Let b be a non-closed branch of $\mathsf{RL_K}$-proof tree of a formula z_1Tz_0. Then $\mathcal{M}^b$ is an $\mathsf{RL_K}$-model.*

A branch structure is referred to as the *branch model*. Let $v^b\colon \mathbb{OV} \to U^b$ be a valuation in $\mathcal{M}^b$ such that $v^b(x) = x$, for every $x \in \mathbb{OV}_b$. The following result can be proved by induction on the complexity of relational terms.

Proposition 6 (Satisfaction in Branch Model Property) *Let b be a non-closed branch of $\mathsf{RL_K}$-proof tree of a formula z_1Tz_0. Then for every $\mathsf{RL_K}$-formula φ, $\varphi \in b$ implies $\mathcal{M}^b, v^b \not\models \varphi$.*

From the previous results we obtain the following two main theorems.

Theorem 7 (Relational Soundness and Completeness of K) *For every K-formula φ, φ is K-valid iff $z_1\tau(\varphi)z_0$ is $\mathsf{RL_K}$-provable.*

Theorem 8 (Decision Procedure for K) *$\mathsf{RL_K}$-dual tableau is a decision procedure for the logic K.*

Example: Consider the K-formula $\varphi = \neg([R]p \wedge [R]q) \vee [R](p \wedge q)$. Let $\tau'(p) = S_1$ and let $\tau'(q) = S_2$. Figure 1 presents an $\mathsf{RL_K}$-proof of $z_1\tau(\varphi)z_0$. In order to prove that φ is K-valid, we need to show that $z_1\tau(\varphi)z_0$ is $\mathsf{RL_K}$-provable. In the second step, we use our ordering of types of relational terms. As a consequence, we apply rule $(-\cap)$ to formula $z_1{-}({-}(R;-S_1) \cap -(R;-S_2))z_0$ because it is of type $(-\cap)$, while $z_1{-}(R;-(S_1 \cap S_2))z_0$ is of type $(-;)$. Similarly in the next step, we use the fact that $(-) < (-;)$. Then we apply the only possible rules (K_2), $(-)$, and $(\cap)$. Finally, we close all the branches of the tree. This means that φ is valid.

4 Conclusions and Future Work

We have presented a new deterministic proof system for modal logic K. Its main difference with other approaches is that it is deterministic, that is, it generates only one tree for a given formula and does not need any *external* technique, such as backtracking in order to get the desired proof tree.

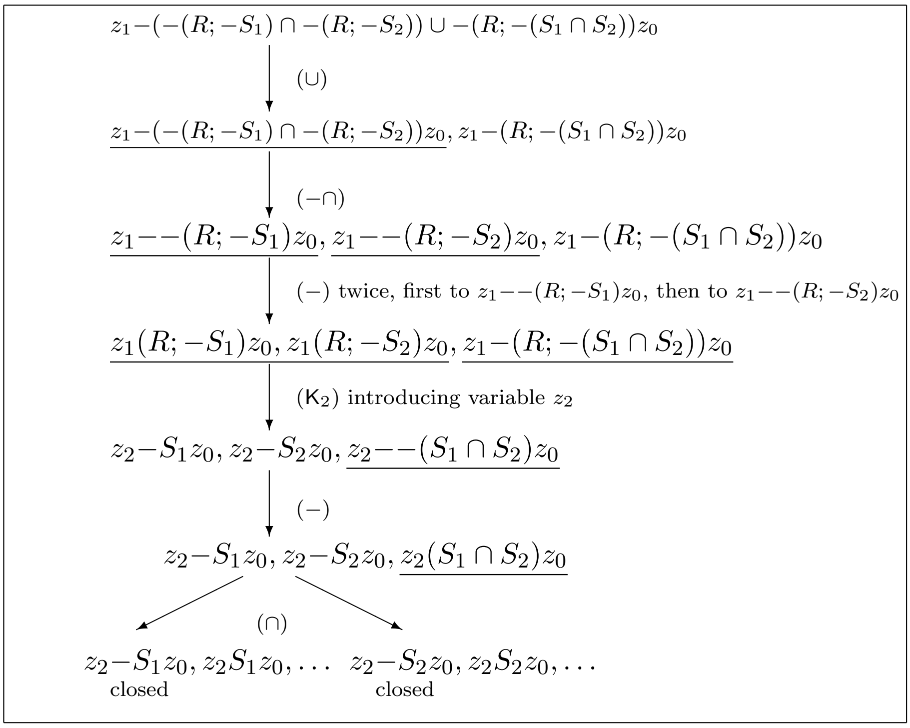

Figure 1: $\mathsf{RL_K}$-proof of K-formula $\neg([R]p\wedge[R]q)\vee[R](p\wedge q)$.

We proved its soundness and completeness and we showed that the system itself is a decision procedure for modal logic K. With the results presented in the paper many questions and problems arise. The most important concerns complexity. It seems that the decision procedure presented is highly complex, but it is very probable that its complexity could be improved if it would be applied to some normal forms of $\mathsf{RL_K}$-formulas. Thus, the future work could be oriented towards normal forms (clause forms) of $\mathsf{RL_K}$-formulas, in order to get a good bound for complexity. Moreover, we are working in the extension of our decision procedure to cover reflexive, symmetric, and transitive modalities while retaining termination. The construction of the rules for semantic constraints remains as an open problem.

References

[1] A. d'Avila Garcez, L.C. Lamb and D.M. Gabbay. *Connectionist modal logic: Representing modalities in neuronal networks.* Theor. Comp. Sci. 371, pp. 34-53, 2007.

[2] S. Bader and P. Hitzler. *Dimensions of Neural-symbolic Integration - A Structured Survey.* Vol. 1, pp. 167–194. King's College Publications, 2005.

[3] P. Baldan, A. Bracciali, L. Brodo, and R. Bruni. *Deducing Interactions in Partially Unspecified Biological Systems*, AB 2007, 262–276.

[4] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Camb. Univ. Press, 2002.

[5] A. Burrieza, A. Mora, M. Ojeda-Aciego, and E Orlowska. *An implementation of a dual tableaux system for order-of-magnitude qualitative reasoning*. Int. J. Comp. Math., 86 (10–11):1852-1866, 2009.

[6] Dallien, J. and MacCaull, W., *RelDT: A relational dual tableaux automated theorem prover*, `http://www.logic.stfx.ca/reldt/`

[7] J. F. De Paz; S. Rodríguez; J. Bajo; J. M. Corchado. *Mathematical model for dynamic case-based planning.* Int. J. Comp. Math., 86 (10–11): 1719 –1730, 2009.

[8] del Cerro, L. F and Gasquet, O. *A General Framework for Pattern-Driven Modal Tableaux*. Logic Journal IGPL 2002 10(1):51-83.

[9] A. Formisano, E. Omodeo, E. Orłowska, *A PROLOG tool for relational translation of modal logics: A front-end for relational proof systems*, TABLEAUX 2005, 1–10.

[10] Golińska-Pilarek, J.; Muñoz-Velasco, E. , Mora, A: *A New Decision Procedure for Modal Logic* K. Techical Report, 2009. `http://www.matap.uma.es/~emilio/TechRepDPK09`.

[11] Golińska-Pilarek, J. Mora, A; Muñoz-Velasco, E.: *$RePML_K$: A Relational Prover for Modal Logic* K. Proceedings CMMSE 2009 Vol. III, pp. 743–764.

[12] Golińska-Pilarek, J. Mora, A; Muñoz-Velasco, E.: *An ATP of a Relational Proof System for Order of Magnitude Reasoning with Negligibility, Non-Closeness and Distance.* Lecture Notes in Artificial Intelligence 5351, pp. 129-139, 2008.

[13] Golińska-Pilarek, J.; Muñoz-Velasco, E.: *Relational approach for a logic for order of magnitude qualitative reasoning with negligibility, non-closeness and distance.* Logic Journal IGPL 17(4):375-394, 2009.

[14] Golińska-Pilarek, J. and Orłowska, E.: *Tableaux and dual Tableaux: Transformation of proofs*, Studia Logica 85 (2007), 291–310.

[15] C. Gros, G. Kaczor. *Semantic learning in autonomously active recurrent neural networks.* Logic Journal IGPL, Adv. Acc. publ. Sept 14, 2009. doi:10.1093/jigpal/jzp045.

[16] A. Herrero, E. Corchado, L. Saiz, and A. Abraham. *DIPKIP: A Connectionist Knowledge Management System to Identify Knowledge Deficits in Practical Cases.* Computational Intelligence, Vol. 26 (1): 26-56, 2010.

[17] Konikowska, B., *Rasiowa-Sikorski deduction systems in computer science applications.* Theor. Comp. Sci. 286, 323–366, 2002.

[18] J. M. Mesa; C. Menendez; F. A. Ortega; P. J. Garcia. *A smart modelling for the casting temperature prediction in an electric arc furnace.* Int. J. Comp. Math., 86 (7): 1182 – 1193, 2009.

[19] M. Miculan and G. Bacci. *Modal Logics for Brane Calculus*, LNCS 4210, 2006, 1– 16.

[20] Sara Negri. Proof analysis in modal logic. *J. of Phil. Logic*, 34(5,6): 507 –544, 2005.

[21] Orłowska, E. and Golińska-Pilarek, J.: *Dual Tableaux: Foundation, Methodology, Case Studies*, A draft of the book, 2010.

[22] H. Rasiowa and R. Sikorski, *The Mathematics of Metamathematics*, Polish Scientific Publishers, Warsaw, 1963.

[23] Sindhu Joseph, Carles Sierra, Marco Schorlemmer, and Pilar Dellunde: *Deductive coherence and norm adoption.* Logic J. IGPL. 2010; 18:118-156.

[24] A. Tarski, *On the calculus of relations*, The JSL Journal 6 (1941), 73–89.