An ATP of a Relational Proof System for Order of Magnitude Reasoning with Negligibility, Non-Closeness and Distance*

Joanna Golińska-Pilarek[1]**, Angel Mora[2], and Emilio Muñoz-Velasco[2]

[1] Institute of Philosophy, Warsaw University
National Institute of Telecommunications, Poland
`j.golinska@uw.edu.pl`
[2] Dept. Matemática Aplicada. Universidad de Málaga. Spain
`{amora,emilio}@ctima.uma.es`

Abstract. We introduce an Automatic Theorem Prover (ATP) of a dual tableau system for a relational logic for order of magnitude qualitative reasoning, which allows us to deal with relations such as negligibility, non-closeness and distance. Dual tableau systems are validity checkers that can serve as a tool for verification of a variety of tasks in order of magnitude reasoning, such as the use of qualitative sum of some classes of numbers. In the design of our ATP, we have introduced some heuristics, such as the so called *phantom variables*, which improve the efficiency of the selection of variables used un the proof.

1 Introduction

Qualitative reasoning (QR) is the area of AI which provides an intermediate level between discrete and continuous models in order to develop representations for continuous aspects of the world, such as space, time, and quantity, without the kind of precise quantitative information needed by conventional analysis techniques [20].

A form of QR is to manage numerical data in terms of orders of magnitude, that is, to stratify values according to some notion of scale [7, 14, 16, 19]. Two approaches to order of magnitude reasoning have been identified in [20]: absolute order of magnitude, which is represented by a partition of the real line $\mathbb{R}$ where each element of $\mathbb{R}$ belongs to a qualitative class and relative order of magnitude, introducing a family of binary order of magnitude relations which establish different comparison relations in $\mathbb{R}$ (e.g., *comparability*, *negligibility*, and *closeness*). In general, both models need to be combined to capture the relevant information.

The introduction of the logic formalism in QR tries to solve the problem about the soundness of the reasoning supported by the formalism and to give some

* Partially supported by Spanish projects TIN2006-15455-C03-01 and P6-FQM-02049.
** The author is a recipient of the 2007 and 2008 Grant for Young Scientists of the Foundation for Polish Science and is partially supported by the Polish Ministry of Science and Higher Education grant N N206 399134.

answers about the efficiency of using that. Several logics have been developed in different contexts, e.g., spatial and temporal reasoning [1,17,21]. In particular, logics dealing with order of magnitude reasoning have been developed in [3–5] by combining the absolute and relative approaches, that is, by defining different qualitative relations using the intervals provided by a specific absolute order of magnitude model.

In this paper, we focus our attention on the multimodal propositional logic $\mathcal{L}(OM)^{\mathrm{NCD}}$ (from now on, OM for short) presented in [3], which introduces a sound and complete axiom system to deal with relations such as negligibility, non-closeness and distance.

We introduce an ATP for a relational proof system in the style of dual tableaux for the relational logic associated with OM, given in [10]. This system can be used as a tool for verification of a variety of tasks in order of magnitude reasoning, such as the use of qualitative sum of some classes of numbers. We emphasize, that the interaction between the theoretical study and the implementation of the ATP has contributed in a new style of proving the formulas by using deduction natural.

Our relational system, is based on the Rasiowa-Sikorski system for the first-order logic [18] extended to the classical relational logic originated in [15], following the ideas presented in [11]. Another approach to relational logics for order of magnitude reasoning has been given in [6].

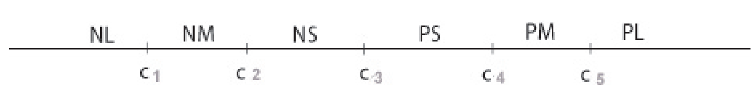
An implementation of the proof system for the classical relational logic is described in [8]. In [9] an implementation of translation procedures from non-classical logics to relational logic is presented. Moreover, in [2], there is an implementation of the system presented in [6].

The paper is organized as follows: In Section 2, we give a short presentation of the syntax, semantics, and the axiomatization of the logic OM, for more details see [3]. In Section 3, we give a survey of the relational logic appropriate for OM and its dual tableau system, presented in [10]. In Section 4, we show the details about the ATP with some examples and, finally, in Section 5, some conclusions and prospects future work are commented.

2 The multimodal logic OM

In this section, we summarize the logic OM introduced in [3]. We consider a strict linearly ordered set $(\mathbb{S}, <)$ [3] divided into seven equivalence classes using five landmarks chosen depending on the context [20]. The cases with a different number of classes could be treated similarly.

The system corresponds to the following schematic representation, where $c_i \in \mathbb{R}$, being $i \in \{1, 2, 3, 4, 5\}$ such that $c_j < c_{j+1}$, for all $j \in \{1, 2, 3, 4\}$:



[3] For practical purposes, this set could be the real line.

The labels correspond, respectively, to the qualitative classes "negative large", "negative medium", "negative small", "zero", "positive small", "positive medium", and "positive large".

The concepts of order of magnitude, non-closeness, distance and negligibility we consider in this paper introduce the 'relative part' of the approach, which builds directly on the 'absolute part' just presented.

First of all, we define the relation $\overrightarrow{d_\alpha}$ to give the intuitive meaning of a *constant distance*, called α. Let $(\mathbb{S}, <)$ be a strict linearly ordered set which contains the constants c_i, for $i \in \{1, 2, 3, 4, 5\}$ as defined above. Given $n \in \mathbb{N}$, we define $\overrightarrow{d_\alpha}$ as a relation on $\mathbb{S}$ such that, for every $x, y, z, x', y' \in \mathbb{S}$ the following hold:

(i) If $x\,\overrightarrow{d_\alpha}\,y$, then $x < y$ (ii) $c_j\,\overrightarrow{d_\alpha}\,c_{j+1}$, for $j \in \{1, 2, 3, 4\}$
(iii) If $x\,\overrightarrow{d_\alpha}\,y$ and $x\,\overrightarrow{d_\alpha}\,z$, then $y = z$ (iv) If $x\,\overrightarrow{d_\alpha}\,y$, $x'\,\overrightarrow{d_\alpha}\,y'$ and $x < x'$ then $y < y'$

In the definition above, we assume for simplicity that every two consecutive constants are at the same distance, called α. This choice arises from the idea of taking α as the basic pattern for measuring. It could be easily generalized by assuming that the distance between two consecutive constants should be a multiple of α.

Now we define the remaining relations on $\mathbb{S}$. For every $x, y \in \mathbb{S}$ we define: $x\,\mathrm{OM}\,y$ if and only if $x, y \in \mathrm{EQ}$, where EQ denotes a qualitative class, that is, an element in the set $\{\mathrm{NL}, \mathrm{NM}, \mathrm{NS}, \mathrm{C}_0, \mathrm{PS}, \mathrm{PM}, \mathrm{PL}\}$. Analogously, we define: $x\,\overline{\mathrm{OM}}\,y$ whenever x, y do not belong to the same class. The relations of *non-closeness* $\overrightarrow{\mathrm{NC}}$ and *distance* $\overrightarrow{\mathrm{D}}$, are defined as follows:

$$
\begin{array}{lll}
x\overrightarrow{\mathrm{NC}}\,y & \text{if and only if} & \text{either } x\overline{\mathrm{OM}}y \text{ and } x < y \\
 & & \text{or there exists } z \in \mathbb{S} \text{ such that } z < y \text{ and } x\overrightarrow{d_\alpha}z \\
x\,\overrightarrow{\mathrm{D}}\,y & \text{if and only if} & \text{there exist } z, z' \in \mathbb{S} \text{ such that } z < z' < y \text{ and } x\overrightarrow{d_\alpha}^{\,2}z.
\end{array}
$$

Notice that $\overrightarrow{d_\alpha}^{\,2} = \overrightarrow{d_\alpha} \circ \overrightarrow{d_\alpha}$, being $\circ$ the usual composition of relations.

If we assume that $\mathbb{S}$ is a set of real numbers, the intuitive interpretation of non-closeness relation is that x is non-close to y if, and only if, either x and y have not the same order of magnitude, or y is obtained from x by adding a medium or large number. On the other hand, x is distant from y if and only if y is obtained from x by adding large number.

In order to define the negligibility relation, note that it seems to be reasonable that if $x \neq c_3$ is *neglibible* with respect to y, then x is distant to y.

Now, we can give the following definition for all $x, y \in \mathbb{S}$: x is *negligible* with respect to y (denoted by $x\overrightarrow{N}y$) if and only if either of the following holds:

$$(i) \quad x = c_3 \qquad (ii) \quad x \in \mathrm{NS} \cup \mathrm{PS} \text{ and, either } y\,\overrightarrow{\mathrm{D}}\,c_2 \text{ or } c_4\,\overrightarrow{\mathrm{D}}\,y.$$

Note that item (i) above corresponds to the intuitive idea that zero is negligible with respect to any real number and item (ii) corresponds to the intuitive idea that a *sufficiently small* number is negligible with respect to any *sufficiently large* number, independently of the sign of these numbers. This definition ensures that if $x \neq c_3$ and $x\overrightarrow{N}y$, then either $y\overrightarrow{\mathrm{D}}x$ or $x\overrightarrow{\mathrm{D}}y$.

The relations of *non-closeness*, *distance* and *negligibility* can defined in terms of $<, \overrightarrow{d_\alpha}$, their inverses, and the constants c_i, for $i \in \{1,2,3,4,5\}$, for this reason, we only consider in our logic, connectives associated to these relations.

The syntax and semantics of OM are defined as usual in modal logics. We consider modal connectives $\overrightarrow{\Box}, \Box_{\overrightarrow{d_\alpha}}$ and $\overleftarrow{\Box}, \Box_{\overleftarrow{d_\alpha}}$ associated to the accessibility relations $<, \overrightarrow{d_\alpha}$ and their inverses, respectively. The intuitive meaning of the constants c_i is that c_i is true only in the constant c_i. The sound and complete axiom system of OM consists of all tautologies of classical propositional logic together with the following axiom schemata, being $i \in \{1, \dots, 5\}$ and $j \in \{1, \dots, 4\}$:

K1 $\overrightarrow{\Box}(A \to B) \to (\overrightarrow{\Box}A \to \overrightarrow{\Box}B)$ **K2** $A \to \overrightarrow{\Box}\overleftarrow{\Diamond}A$ **K3** $\overrightarrow{\Box}A \to \overrightarrow{\Box}\overrightarrow{\Box}A$
K4 $(\overrightarrow{\Box}(A \vee B) \wedge \overrightarrow{\Box}(\overrightarrow{\Box}A \vee B) \wedge \overrightarrow{\Box}(A \vee \overrightarrow{\Box}B)) \to (\overrightarrow{\Box}A \vee \overrightarrow{\Box}B)$
C1 $\overleftarrow{\Diamond}c_i \vee c_i \vee \overrightarrow{\Diamond}c_i$ **C2** $c_i \to (\overleftarrow{\Box}\neg c_i \wedge \overrightarrow{\Box}\neg c_i)$ **d1** $\overrightarrow{\Box}A \to \Box_{\overrightarrow{d_\alpha}}A$ **d2** $\Diamond_{\overrightarrow{d_\alpha}}A \to \Box_{\overrightarrow{d_\alpha}}A$
d3 $(\Diamond_{\overrightarrow{d_\alpha}}A \wedge \overrightarrow{\Diamond}\Diamond_{\overrightarrow{d_\alpha}}B) \to \overrightarrow{\Diamond}(A \wedge \overrightarrow{\Diamond}B)$ **d4** $c_j \to \Diamond_{\overrightarrow{d_\alpha}}c_{j+1}$
d5 $\Box_{\overrightarrow{d_\alpha}}(A \to B) \to (\Box_{\overrightarrow{d_\alpha}}A \to \Box_{\overrightarrow{d_\alpha}}B)$ **d6** $A \to \Box_{\overrightarrow{d_\alpha}}\Diamond_{\overleftarrow{d_\alpha}}A$

The corresponding mirror images of **K1**–**K4** and **d1**–**d6** are also considered as axioms. We also consider the rules of inference as usual in modal logic.

3 Relational formalization of OM

This section summarizes the more important concepts about relational logics needed to obtain the relational formalization of our logic, for more details, see [10, 11, 15].
The language of the logic $\mathsf{RL}_{\mathsf{OM}}$ appropriate for expressing OM-formulas consists of the following pairwise disjoint sets of symbols:

$\mathbb{OV} = \{x, y, z, \dots\}$ - a countably infinite set of object variables;
$\mathbb{OC} = \{c_i : i \in \{1, \dots, 5\}\}$ - the set of object constants;
$\mathbb{RV} = \{P, Q, \dots\}$ - a countably infinite set of binary relational variables;
$\mathbb{RC} = \{1, 1', <, d_\alpha\} \cup \{\Psi_i : i \in \{1, \dots, 5\}\}$ - the set of relational constants [4];
$\mathbb{OP} = \{-, \cup, \cap, ;, ^{-1}\}$ - the set of relational operation symbols.

The set of *relational terms* $\mathbb{RT}$ is the smallest set of expressions including the set $\mathbb{RV} \cup \mathbb{RC}$ of *atomic* terms and closed with respect to the operation symbols from $\mathbb{OP}$. The set of $\mathsf{RL}_{\mathsf{OM}}$-*formulas* (or, simply formulas if it is clear from the context), consists of expressions of the form xPy where $x, y \in \mathbb{OS} = \mathbb{OV} \cup \mathbb{OC}$ and $P \in \mathbb{RT}$.
The semantics of $\mathsf{RL}_{\mathsf{OM}}$ can be given as usual in relational logic, by using the previous definitions of our accessibility relations and constants. The respective semantics of OM and $\mathsf{RL}_{\mathsf{OM}}$ give us the concepts of OM-validity and $\mathsf{RL}_{\mathsf{OM}}$-validity. Now, we define a translation function in order to have a relationship

[4] 1 and $1'$ represent, respectively, the universal and equality relations.

between these concepts. The translation of OM-formulas into relational terms starts with a one-to-one assignment of relational variables to the propositional variables, called τ'. Then the translation τ of OM-formulas is defined inductively as follows, being ; and $-$ the composition and opposite of relations, respectively:

$\tau(p) = \tau'(p); 1$, for every propositional variable p.

$\tau(c_i) = \Psi_i; 1$, for every $i \in \{1, \ldots, 5\}$

τ extends to all compound OM-formulas as follows [5]:

$$\tau(\neg\varphi) = -\tau(\varphi) \qquad \tau(\varphi \vee \psi) = \tau(\varphi) \cup \tau(\psi) \quad \tau(\varphi \wedge \psi) = \tau(\varphi) \cap \tau(\psi)$$
$$\tau(\varphi \to \psi) = -\tau(\varphi) \cup t(\psi) \quad \tau(\overrightarrow{\Box}\varphi) = -(<; -\tau(\varphi)) \quad \tau(\Box_{\overrightarrow{d_\alpha}}\varphi) = -(d_\alpha; -\tau(\varphi))$$

The following theorem shows the semantical relationship between OM and $\mathsf{RL_{OM}}$:

Theorem 1. *For every* OM*-formula* φ *and for all object variables* x *and* y*,* φ *is* OM*-valid iff* $x\tau(\varphi)y$ *is* $\mathsf{RL_{OM}}$*-valid.*

Dual tableau systems are determined by axiomatic sets of formulas and rules which apply to finite sets of formulas. The axiomatic sets take the place of axioms. There are two groups of rules: the *decomposition rules* which reflect definitions of the standard relational operations and the *specific rules* which reflect the properties of the specific relations assumed in $\mathsf{RL_{OM}}$-models. The rules are of the form $\frac{\Phi}{\Phi_1 \mid \ldots \mid \Phi_n}$, where $\Phi_1, \ldots, \Phi_n$ are finite non-empty sets of formulas, $n \geq 1$, and Φ is a finite (possibly empty) set of formulas. Φ is called the *premise* of the rule, and $\Phi_1, \ldots, \Phi_n$ are called its *conclusions*. A rule is said to be *applicable* to a set X of formulas whenever $\Phi \subseteq X$. As a result of an application of a rule to a set X, we obtain the sets $(X \setminus \Phi) \cup \Phi_i$, $i = 1, \ldots, n$.

We say that an object variable in a rule is *new* whenever it appears in a conclusion of the rule and does not appear in its premise.

Decomposition rules of $\mathsf{RL_{OM}}$-dual tableau have the following forms, for all object symbols $x, y \in \mathbb{OS}$ and for all relational terms $P, Q \in \mathbb{RT}$, where z is any object symbol and w is a new object variable:

$$(\cup)\ \frac{x(P \cup Q)y}{xPy, xQy} \quad (-\cup)\ \frac{x-(P \cup Q)y}{x-Py \mid x-Qy} \quad (\cap)\ \frac{x(P \cap Q)y}{xPy \mid xQy} \quad (-\cap)\ \frac{x-(P \cap Q)y}{x-Py, x-Qy}$$

$$(-)\ \frac{x--Py}{xPy} \quad ({}^{-1})\ \frac{xP^{-1}y}{yPx} \quad (-{}^{-1})\ \frac{x-P^{-1}y}{y-Px}$$

$$(;)\ \frac{x(P;Q)y}{xPz, x(P;Q)y \mid zQy, x(P;Q)y} \quad z\ (-;)\ \frac{x-(P;Q)y}{x-Pw, w-Qy}$$

Specific rules of $\mathsf{RL_{OM}}$-dual tableau have the following forms, for all object symbols $x, y \in \mathbb{OS}$, for every atomic relational term R, and for every $i \in \{1, \ldots, 5\}$, where z, v are any object symbols:

[5] The translation of the inverse formulas is trivial.

$$(1'1)\ \frac{xRy}{xPz, xPy \mid y1'z, xPy} \qquad (1'2)\ \frac{xRy}{x1'z, xPy \mid zPy, xPy}$$

$$(\mathrm{Irref}<)\ \frac{}{x<x} \qquad (\mathrm{Tran}<)\ \frac{x<y}{x<y, x<z | x<y, z<y}$$

$$(C_i1)\ \frac{}{x\Psi_i y \mid x{-}\Psi_i y} \qquad (C_i2)\ \frac{x\Psi_i y}{x\Psi_i y, x1'c_i} \qquad (C_i3)\ \frac{x{-}\Psi_i y}{x{-}\Psi_i y, x{-}1'c_i}$$

$$(\mathrm{D1})\ \frac{x<y}{xd_\alpha y, x<y} \qquad (\mathrm{D2})\ \frac{x1'y}{zd_\alpha x, x1'y | zd_\alpha y, x1'y}$$

$$(\mathrm{D3})\ \frac{x<y}{zd_\alpha x, x<y | vd_\alpha y, x<y | z<v, x<y}$$

A finite set of $\mathsf{RL_{OM}}$-formulas is said to be an $\mathsf{RL_{OM}}$-*axiomatic set* whenever it includes either of the following subsets, for any $x, y \in \mathbb{OS}, R \in \mathbb{RT}, i \in \{1, \dots, 4\}$:
(Ax1) $\{x1'x\}$, (Ax2) $\{x1y\}$, (Ax3) $\{xRy, x{-}Ry\}$, (Ax4) $\{c_i d_\alpha c_{i+1}\}$, (Ax5) $\{x<y, y<x, x1'y\}$.

An $\mathsf{RL_{OM}}$-*proof tree* for a formula xPy is a tree with the following properties:

- xPy is at the root of this tree;
- each node except the root is obtained by an application of an $\mathsf{RL_{OM}}$-rule to its predecessor node;
- a node does not have successors whenever it is an $\mathsf{RL_{OM}}$-axiomatic set.

Due to the forms of the rules for atomic formulas, if a node of an $\mathsf{RL_{OM}}$-proof tree contains an $\mathsf{RL_{OM}}$-formula xPy or $x{-}Py$, for some atomic P, then all of its successors contain this formula as well.
A branch of an $\mathsf{RL_{OM}}$-proof tree is said to be *closed* whenever it contains a node with an $\mathsf{RL_{OM}}$-axiomatic set of formulas. A *closed tree* is an $\mathsf{RL_{OM}}$-proof tree such that all of its branches are closed. A formula xPy is $\mathsf{RL_{OM}}$-*provable* whenever there is a closed proof tree for xPy, which is then referred to as an $\mathsf{RL_{OM}}$-*proof of* xPy.

The following main result ensures the correspondence between OM-validity and $\mathsf{RL_{OM}}$-provability.

Theorem 2 (Soundness and Completeness).
Let φ be an OM-formula. Then for all object variables x and y, φ is OM-valid iff $x\tau(\varphi)y$ is $\mathsf{RL_{OM}}$-provable.

4 The ATP

We show in broad strokes the implementation realized in Prolog [6] of an ATP for obtained an automatic Rasiowa-Sikorski proof system associated to the relational translation $\mathsf{RL_{OM}}$ of the multimodal logic of qualitative order of magnitude reasoning OM.
We have represented the formula $x_m R_i y_n$ as the Prolog fact: $rel([1], R_i, x_m, y_n)$. Node [1] denotes the root of the proof tree that the Prolog tool develops when it applies the rules of the $\mathsf{RL_{OM}}$.

[6] See http://www.matap.uma.es/˜ emilio/omr.zip, for a revision of the ATP.

Example 1. The union of expressions $xRy \cup x{-}(\overrightarrow{d_\alpha};-(a;1))y$ is translated to the following facts in Prolog:
`rel([1],r,x,y).`
`rel([1],opposite(comp(dalpha,opposite(comp(a,universal)))),x,y).`

Prolog knows the leaf in which it must apply any rule, because the Prolog predicate `leaves`([[1, ..., 1], ..., [1, ..., k]]) stores the leaves that the tool must close. Prolog will try to satisfy the relations in the leaf nodes. If the tool can close all the leaves in the tree, then *formula* is true.

The rules in $\mathsf{RL_{OM}}$ have the following general form: $\dfrac{\Phi}{\Phi_1|\ldots|\Phi_n}$ where $\Phi_1, \ldots, \Phi_n$ are non-empty set of formula and Φ is a finite (possibly finite) set of formula. Let X a set of formulas, and if $\Phi \subseteq X$ then, as said before, the system transform Φ in $X \setminus \Phi \cup \Phi_i, i = 1\ldots, n$. That's to say , if X is represented in the leaf $[i_1, i_2, \ldots, i_k]$, the system divides the the leaf in n new leaves, labeled as $[i_1, i_2, \ldots, i_k, i_{k+1}], \ldots [i_1, i_2, \ldots, i_k, i_{k+n}]$ and copies $(X \setminus \Phi) \cup \Phi_1$ to the node $[i_1, i_2, \ldots, i_k, i_{k+1}]$, and copies $X \setminus \Phi \cup \Phi_2$ to the node $[i_1, i_2, \ldots, i_k, i_{k+2}]$ (see Figure 1).

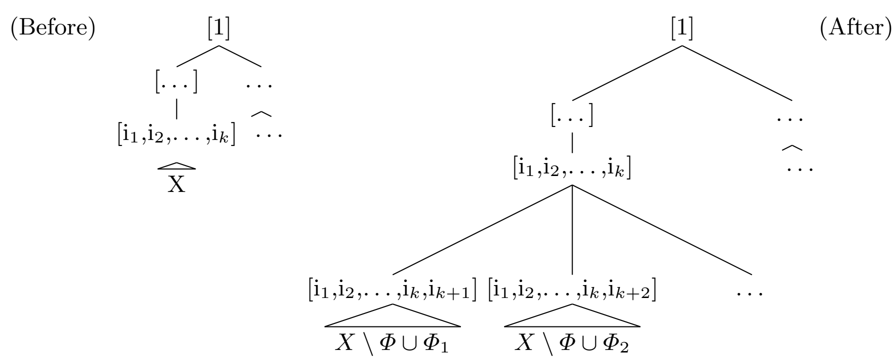


Fig. 1. Division of a leaf of the tree.

We have translated the rules for $\mathsf{RL_{OM}}$ to clauses in Prolog. For example, for the union rule $(\cup)$:

```
uni(Leaf):-
          rel(Leaf,uni(R,S),X,Y),
          new_rule_deduced([rel(Leaf,R,X,Y),rel(Leaf,S,X,Y)]),
          \+rule_used(Leaf,uni,[rel(uni(R,S),X,Y)]),
          write_rule('Union', [rel(Leaf,uni(R,S),X,Y)],
                              [rel(Leaf,R,X,Y), rel(Leaf,S,X,Y)]),
          add_list_of_relations([rel(Leaf,R,X,Y),rel(Leaf,S,X,Y)]).
```

Any rule of $\mathsf{RL_{OM}}$ in Prolog checks the preconditions $x(R \cup S)y$ in which the rule is applicable. If the rule fulfils these conditions, we control if the relations

deduced by the rule are new (`new_rule_deduced predicate`) and the rule has not previously applied (`rule_used predicate`), then we write the rule in the display and store the rule applied. Finally, we apply the rule, that normally adds some facts to the adequate leaf.
The (D2) rule divides the node labeled *Leaf* in two new leaves and copy all formulas of *Leaf* to the two new ones, by using the predicate `divideInLeaves`. The predicate `copyToLeaves` adds $zd_\alpha x$ to the first leaf and adds $zd_\alpha y$ to the second leaf.

```
d2(Leaf):-
          rel(Leaf,equal,X,Y),
         \+rule_used(Leaf,d2,[rel(equal,X,Y)]),
          any_variable('d2  (equal) ',Leaf,[rel(Leaf,equal,X,Y)],Z),!,
          divideInLeaves(Leaf,2),
          copyToLeaves(Leaf,1,[[rel(Leaf,dalpha,Z,X)]
                                ,[rel(Leaf,dalpha,Z,Y)]],[],ListNewLeaves),
          remove_leaf_after_divide(Leaf),
          write_and_rule('d2  (equal) ', [ rel(equal,X,Z)],
                             [[ rel(Leaf,dalpha,Z,X)],[rel(Leaf,dalpha,Z,Y)]]
                             ,ListNewLeaves),!.
```

Now, we show the engine of the ATP. The main predicate in the inference engine is `run_engine` that examine the first leaf of the tree that the proof system needs to check and tries to apply the rules to the relations that contains this leaf. The engine tries first to apply the rules that no divide the leaves and then the rules that divide the leaves.

```
new_run_engine:-
     leaves([FirstLeaf|Leaves]),
    new_apply_rules_in_leaf(FirstLeaf),
    new_run_engine,!.
new_run_engine:-
    write(' OK. There are no Leaves in the proof tree. '),
    write(' VALID. '),!.
new_apply_rules_in_leaf(FirstLeaf):-
     new_one_rule_no_divide(FirstLeaf),!.
new_apply_rules_in_leaf(FirstLeaf):-
     new_one_rule_divide(FirstLeaf),!.
new_one_rule_no_divide(FirstLeaf):-
  uni(FirstLeaf)->             axiomatic_set;
  notinter(FirstLeaf)->        axiomatic_set;
  ...
new_one_rule_divide(FirstLeaf):-
  notuni(FirstLeaf)->          axiomatic_set;
  d2(FirstLeaf)->              axiomatic_set;
  ...
```

While the tree has opened leaves, `new_run_engine` is recursively called. If all leaves are closed in the proof system, then system informs to the user that the proof is finished and it is possible to trace (`used_rules predicate`) what rules have been used in the proof process. The engine of the ATP use the mechanism of pattern machine of Prolog to detect if exists, in a leaf of the tree, an axiomatic set, then deletes the corresponding leaf and informs to the user.

```
axiomatic_set:-
          rel(NumLeaf,equal,X,X),
          nl,
          remove_leaf(NumLeaf,[rel(NumLeaf,equal,X,X)]),!.
.....
```

In this point, we introduce an important idea which improves the efficiency of our system. Some rules of the logic need to introduce *any object symbol*, that is, either a constant or any of the previously used variables. The ATP delays the substitution by any of the possible object symbols and introduces a *phantom variable*. The system replaces the phantom variable by any of the possible objects, only when obtains an axiomatic set with this substitution and then it closes the tree.

We emphasize that this mechanism avoids the process of selecting a possible variable and checking the validity of the formula in this leaf with this variable. In that case, it would be necessary to expand the leaf in a enormous sub-tree and, if the formula could not be proved, to return to the previous leaf by selecting another variable. The process would be repeated for all possible variables.

The *phantom variables* prune the search tree in a efficient way. The instantiation of the phantom variable is delayed until the ATP is able to obtain an axiomatic set. In this moment, the unification of the correct variable is done in the tree and some sub-trees are closed.
In the following example, we outline how the ATP works and emphasize the use of the phantom variables for detecting axiomatic sets in an automatic way.

Example 2. In this example, we execute the ATP to prove the axiom **d2** of the system OM. We represent it as follows:

```
rel([1],opp(comp(dalpha, comp(p, universal))),x,y).
rel([1],opp(comp(dalpha, opp(comp(p, universal)))), x, y).
```

This example is satisfied by the ATP with the Prolog predicate:

$$?tad('axioms \setminus axiomd2.pl','logaxiomd2.txt').$$

The following report in logaxiomd2.txt file is returned:

```
------>Input file: axioms\axiomd2.pl
leaves([[1]]).
--->Opposite composition Rule
[rel([1],opp(comp(dalpha,comp(p,universal))),x,y)]
-----------------------------------------------------------------------
[rel([1],opp(dalpha),x,z),rel([1],opp(comp(p,universal)),z,y)]
...
```

```
[rel([1],comp(p,universal),t,y)]
-------------------------------------------------------------------------
rel([1,1],p,t,t1) | rel([1,2],universal,t1,y)

  Found axiomatic set. Leaf: [1,2]
    - Axiomatic set: [rel([1,2],universal,t1,y)]
    - Deleted relations in Leaf: [1,2]
...
[rel([1,1,1,1,1,1,2,2],opp(p),z,u)]
-------------------------------------------------------------------------
rel([1,1,1,1,1,1,2,2,1],equal,z,t8) | rel([1,1,1,1,1,1,2,2,2],opp(p),t8,u)

Substitute in all relations variable phantom:t8 by t
Substitute in all relations variable phantom:t1 by u
  Found axiomatic set. Leaf: [1,1,1,1,1,1,2,2,2]
    - Axiomatic set: [rel([1,1,1,1,1,1,2,2,2],opp(p),t8,u),
                      rel([1,1,1,1,1,1,2,2,2],p,t,t1)]
    - Deleted relations in Leaf: [1,1,1,1,1,1,2,2,2]
....
[rel([1,1,1,1,1,1,2,2,1],equal,z,t)]
-------------------------------------------------------------------------
rel([1,1,1,1,1,1,2,2,1,1],dalpha,t9,z)|rel([1,1,1,1,1,1,2,2,1,2],dalpha,t9,t)

Substitute in all relations variable phantom:t9 by x
  Found axiomatic set. Leaf: [1,1,1,1,1,1,2,2,1,2]
    - Axiomatic set: [rel([1,1,1,1,1,1,2,2,1,2],opp(dalpha),x,t),
                      rel([1,1,1,1,1,1,2,2,1,2],dalpha,t9,t)]
    - Deleted relations in Leaf: [1,1,1,1,1,1,2,2,1,2]
  Found axiomatic set. Leaf: [1,1,1,1,1,1,2,2,1,1]
    - Axiomatic set: [rel([1,1,1,1,1,1,2,2,1,1],opp(dalpha),x,z),
                      rel([1,1,1,1,1,1,2,2,1,1],dalpha,x,z)]
    - Deleted relations in Leaf: [1,1,1,1,1,1,2,2,1,1]
 OK. There are no Leaves in the proof tree. VALID.
```

Notice that the substitution of the phantom variable `t1` has been delayed until the appearance of variable `t8`, because in this moment, the leaf can be closed by replacing `t1` and `t8` by `u` and `t`, respectively. In this case, the last two leaves of the tree are closed with this unification process.

Finally, we remark that the system has some abduction mechanism. It is capable to give explanations about what rules have been used to prove a set of relations. We have the predicate `used_rules` that store knowledge about the reasoning process of the inference engine. We give below a trace in inverse order of the proof process:

```
used_rules([1,1,1,1,1,1,2,2,1],d2,[rel(equal,z,t)]).
used_rules([1,1,1,1,1,1,2,2],equality2,[rel(opp(p),z,u)]).
...
used_rules([1],notcomp,[rel(opp(comp(dalpha,comp(p,universal))),x,y)]).
```

5 Conclusions and future work

In this paper, we have implemented an ATP for the relational proof system in the style of dual tableaux for the relational logic associated with the multimodal propositional logic for order of magnitude qualitative reasoning OM. This system can be used as a tool for verification of a variety of tasks in order of magnitude reasoning, such as the use of qualitative sum of some classes of numbers.

Nowadays, we are working in a more intelligent engine for the ATP and implementing a mechanism that selects what is the better rule by analyzing the relations and the variables in the tree. Also, we are improving the use of phantom variables to obtain an ATP more efficient.

The ATP works with depth-first search, at the moment. We are going to programme a more intelligent engine for the ATP, that combines the depth-first search with breadth-first search, depending on the analysis of the knowledge obtained from the formulas.

The goal for the future is to generalize this implementation for different logics (not only for order of magnitude reasoning). The idea is to develop an ATP more general that receives as input the description of the logic: constants, rules, constraints, etc. and renders the translation in a relational system. Moreover, the ATP will be allowed to prove the validity of any set of formulas of this logic.

Other future works are related to the study of decidability of this logic and, in the case of positive answer, to obtain decision procedures, by using some of the ideas presented in this paper. Last, but not least, it is planned to extend our ATP, in order to be used for model checking and verification of entailment.

References

1. Bennett, B., Cohn, A.G., Wolter, F. and Zakharyaschev, M., *Multi-Dimensional Modal Logic as a Framework for Spatio-Temporal Reasoning.*, Applied Intelligence 17(3), 239–251, 2002
2. Burrieza, A., Mora, A., Ojeda-Aciego, M., and Orłowska, E., *Implementing a relational system for order-of-magnitude reasoning.* Technical Report, 2008.
3. Burrieza, A., Muñoz-Velasco, E., and Ojeda-Aciego, M., *A Logic for Order of Magnitude Reasoning with Negligibility, Non-closeness and Distance.* , Lecture Notes in Artifical Intelligence 4788: 210-219, 2007.
4. Burrieza, A., Muñoz-Velasco, E., and Ojeda-Aciego, M., *Order of magnitude reasoning with bidirectional negligibility.* Lecture Notes in Artifical Intelligence Vol. 4177: 370-378, 2006.
5. Burrieza, A., and Ojeda-Aciego, M., *A multimodal logic approach to order of magnitude qualitative reasoning with comparability and negligibility relations.*, Fundamenta Informaticae 68: 21–46, 2005.
6. Burrieza, A., Ojeda-Aciego, M., and Orłowska, E., *Relational approach to order-of-magnitude reasoning.* Lecture Notes in Computer Science, 4342: 105-124, 2006.
7. Dague, P., *Symbolic reasoning with relative orders of magnitude.*, in: Proc. 13th Intl. Joint Conference on Artificial Intelligence, Morgan Kaufmann, 1509–1515, 1993

8. Dallien, J. and MacCaull, W., *RelDT: A relational dual tableaux automated theorem prover*, http://www.logic.stfx.ca/reldt/
9. Formisano, A. and Orłowska, E. and Omodeo, E., *A PROLOG tool for relational translation of modal logics: A front-end for relational proof systems.*, TABLEAUX 2005 Position Papers and Tutorial Descriptions, B. Beckert (ed.), Fachberichte Informatik No 12, Universitaet Koblenz-Landau, 1–10, 2005. http://www.di.univaq.it/TARSKI/transIt/
10. Golińska-Pilarek, J., and Muñoz-Velasco, E., *Relational approach for a logic for order of magnitude qualitative reasoning with negligibility, non-closeness and distance*. Technical Report, 2008.
11. Golińska-Pilarek, J., and Orłowska, E., *Relational logics and their applications.*, in: H. de Swart, E. Orłowska, M. Roubens, and G. Schmidt, *Theory and Applications of Relational Structures as Knowledge Instruments II*, Lecture Notes in Artificial Intelligence (LNAI), Nr. 4342, Springer, Heidelberg, 125–161, 2006.
12. MacCaull W., and Orłowska, E., *Correspondence results for relational proof systems with application to the Lambek calculus.*, Studia Logica 71, 279–304, 2002.
13. Mavrovouniotis, M.L., and Stephanopoulos, G., *Reasoning with orders of magnitude and approximate relations.*, Proc. 6th National Conference on Artificial Intelligence, The AAAI Press/The MIT Press, 1987.
14. Mavrovouniotis, M.L. *A belief framework for order-of-magnitude reasoning and other qualitative relations.*, Artificial Intelligence in Engineering, Volume 11, Issue 2, Pages 121-134, 1997.
15. Orłowska, E., *Relational interpretation of modal logics.*, in: H. Andréka, D. Monk and I. Nemeti (eds.), Algebraic Logic, Col. Math. Soc. J. Bolyai 54, North Holland, Amsterdam, 443–471, 1988.
16. Raiman, O., *Order of magnitude reasoning.*, Artificial Intelligence 51 (1991), 11–38.
17. Randell, D., Cui, Z. and Cohn, A., *A spatial logic based on regions and connections.*, Proc. of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92), 165–176, 1992.
18. Rasiowa, H., and Sikorski, R., *The Mathematics of Metamathematics.*, Polish Scientific Publishers, Warsaw 1963.
19. Sánchez, M., Prats, F., and Piera, N., *Una formalizacin de relaciones de comparabilidad en modelos cualitativos.*, Boletín de la AEPIA (Bulletin of the Spanish Association for AI) 6, 15–22, 1996.
20. Travé-Massuyès, L., Ironi, L. and Dague P., *Mathematical Foundations of Qualitative Reasoning.*, AI Magazine, American Asociation for Artificial Intelligence, 91–106, 2003.
21. Wolter, F. and Zakharyaschev, M., *Qualitative spatio-temporal representation and reasoning: a computational perspective.*, in G. Lakemeyer and B. Nebel (eds.), Exploring Artificial Intelligence in the New Millenium, Morgan Kaufmann, 2002.