Integrating Incomplete Information into the Relational Data Model

Jorge Ribeiro, José Machado, António Abelha, Manuel Fernandéz-Delgado and José Neves

***Abstract*— Knowledge and belief are generally incomplete, contradictory, or even error sensitive, being desirable to use formal tools to deal with the problems that arise from the use of partial, contradictory, ambiguous, imperfect, nebulous, or missing information. Historically, uncertain reasoning has been associated with probability theory. However, qualitative models and qualitative reasoning have been around in database theory and Artificial Intelligence research for some time, in particular due to the growing need to offer user support in decision making processes. In this paper, and under the umbrella of the Multi-valued Extended Logic Programming formalism to knowledge representation and reasoning we present an evaluative perspective of such an approach, in order to select the best theories (or logic programs) that model the universe of discourse to solve a problem, in terms of a process of quantification of the quality-of-information that stems out from those theories. Additionally, we present a novel approach to integrate incomplete information into the relational data model, making possible the use of the relational algebra operators and the potential inherent to the Structured Query Languages to present solutions to a particular problem and to measure their degree of self-reliance.**

***Index Terms*— Incomplete Information; Quality-of-Information; Decision Support Systems; Relational Data Model; Extended Logic Programming.**

I. Introduction

In almost all decisions that one may take, the information is not always exact, but indeed imperfect, in the sense that we handle estimated values, probabilistic measures, or degrees of uncertainty [1, 2]. Logic and logic programs have emerged as an attractive knowledge representation and reasoning formalism, i.e. as an efficient mechanism to solve search problems. In the past few decades, many non-classical techniques for modelling the universe of discourse and reasoning procedures of intelligent systems have been proposed [3, 4, 5]. Although there exists the ought to treat the problem of uncertain information, one is faced with a second must, related to the problem of handling incomplete information. In this paper we use the Extended Logic Programming formalism [6] to knowledge representation and reasoning, presenting an evaluative perspective of such approach in order to select the best theories (or logic programs) to solve a problem. We use the quantification of the quality-of-information [5, 18, 19] that stems out from a logic program to select those theories. Additionally, it is presented a novel approach to integrate incomplete information into the relational data model, making possible the use of relational algebra operations and the potential inherent to the Structured Query Languages to answer possible queries on demand from the user.

J. Ribeiro is Assistant Lecturer at School of Technology and Management, Viana do Castelo Polytechnic Institute, Portugal (e-mail: jribeiro@estg.ipvc.pt).
J. Machado is Professor at the Informatics Department of the University of Minho, Portugal (e-mail: jmac@di.uminho.pt).
A. Abelha is Professor at the Informatics Department of the University of Minho, Portugal (e-mail: abelha@di.uminho.pt).
M. Fernandéz-Delgado is Professor at Department of Electronic and Computation, University of Santiago de Compostela, Spain, (e-mail: manuel.fernandez.delgado@usc.es).
J. Neves is Cathedratic Professor at the Informatics Department of the University of Minho, Portugal (e-mail: jneves@di.uminho.pt).

A. Related Work

Historically, uncertain reasoning has been associated with Probability Theory [8] but promising research have been done using other formalisms linking logic with probability theory. These formalisms include the theory of fuzzy sets [9], multi-valued logics [10], the Dempster-Shafer theory of evidence [11], hybrid (i.e. numerical and non-numerical) formalisms, and non standard logics. The Abductive Logic Programming (ALP) [17, 12, 4] is a promising computational paradigm and has been recognized as a way to solve some limitations of logic programming with respect to higher level knowledge representation and reasoning tasks. Abduction is a way of reasoning on incomplete or uncertain knowledge, in the form of hypothetical reasoning, more appropriate to model generation and satisfiability checking. Pereira et al. [4] that study the relation between abduction, Well-Founded Semantics and Stable Models, focus their recent research on the problem of the agent's state when confronted with a possible course of evolution, giving special attention to possible levels of commitments and preferences in order to evaluate achievable goals.

However, qualitative models and qualitative reasoning have been around in Artificial Intelligence research for some time [13, 14], in particular due the growing need to offer support in decision-making processes. The evaluation of knowledge that stems out from logic programs becomes a point of research. In this sense, the evaluation of knowledge that stems out from logic programs becomes a point of research. Lucas [15] and Hommersom [16] work is a good example of quality evaluation using logic. They used abduction [17] and temporal logic for quality-checking of medical guidelines, proposing a method to diagnose potential problems in a guideline, regarding the fulfillment of general medical quality criteria at a meta-level characterization. They explored an approach which uses a relational translation to map the temporal logic formulas to first-order logic and a resolution-based theorem prover [16]. In another research

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

line, the Quality-of-Information concept (*QoI*) [5] demonstrated their applicability in many dynamic environments and for decision making purposes. The objective is to built a quantification process of the quality-of-information that stems from a logic program or theory during the evolutive process of searching solutions in order to resolve a problem in environments with incomplete information. Based their work in the mechanical proving theorem [6] and in the *QoI* concept [5], Neves et. al. [18, 5, 7] focused in represent knowledge (e.g. universe of the discourse of an agent' knowledge) and to create mechanisms to infer knowledge (mechanical theorem proving) in non-monotic reasoning with incomplete information. Following these concepts good results were achieved for different purposes, namely in, medical [19], Law [20], Multi-agent Systems [21], Virtual Entities [7], ambient assisted living [22] and Decision Making Environments [23].

II. KNOWLEDGE REPRESENTATION AND QUALITY-OF-INFORMATION

A. Scenario

A data base management system [24] is a powerful tool for creating and managing large amounts of data efficiently and allowing it to persist over long periods of time. Intelligent systems require the ability to reason with incomplete information, by the fact that in the real world complete information is hard to obtain, even in the most controlled situation. We take most of our decisions, if not all, based on incomplete, not precise and even uncertain information. Also, a major factor in the flexibility of human reasoning about complex systems comes from the natural ability to use partial information and to combine it according to its availability. On the other hand most of the information systems just ignore this characteristic of the information about the real world and build upon models where some idealization expunges the inherent uncertainty. The result is a system that never provides the expected answers, due to its inability to model the world. Instead, one should deal with the uncertainty in the model itself. Indeed, to implement useful information systems, namely knowledge based ones, it is necessary to represent and reason with defective information.

To exemplify the applicability of our work we based on the logic databases, prolog and relational data model concepts. According to Kowalski [1] a logic database comprises a collection of Horn clauses and Prolog was chosen as representative of a logic programming language. The relational data model [24] was chosen as the working data base model due to its affinity to the subset of predicate logic known as the Horn clausal form of predicate logic, of which the programming language Prolog is a qualified realization. Consider the scenario where a relational database model is composed by four relations as presented in the figure 1. The objective is to represent a simple model to store and manage information about suppliers and companies that provide products for costumers.

suppliers

#	*idS*	*nameS*	*ratingS(%)*
1.	1	supplier1	{20,40}
2.	2	supplier2	60
3.	3	supplier3	25

p_suppliers

#	*idS*	*idP*	*plafond($)*
1.	1	50	{200,400}
2.	2	51	300
3.	3	52	500

companies

#	*idC*	*idS*	*idP*	*nameC*	*ratingC(%)*
1.	1	3	52	company1	35
2.	2	1	50	company2	{40,50,60}
3.	3	2	51	company3	45

products

#	*idP*	*nameP*	*color*	*price($)*
1.	50	product1	black	200
2.	51	product2	white	150
3.	52	product3	blue	300

Fig.1. A simple Relational Database model

This relational model gives a single way to represent the data as a two-dimensional table called relation. Consider the schema (name of a relation and the set of attributes) of the relational data model referred above as:

1. *suppliers(#idS, names,ratingS)*
2. *p_suppliers(#idS,#idP, plafond)*
3. *companies(#idC,#idS,#idP,nameC,ratingC)*
4. *products(#idP,nameP,Color,Price)*

Table 1. Relations of a Relational data model.

where in the first line the relation suppliers has three attributes, the identification of the supplier (*idS*), the name of the supplier (*nameS*) and the rating market of the supplier (*ratingS*). The second line presents the plafond of the supplier with the attributes, identification of the supplier (*idS*), identification of the product (idP) and the plafond provided by the company. In the third line it is presented the companies relation with the attributes: identification of the company (*idC*), identification of the supplier (*idS*), the identification of the product (*idP*), the name of the company and the rating of the company in the market (*ratingC*). In the last line it is presented the relation of the products with the attributes: identification of the product (*idP*), the name of the product, the color and the price of the product. In this relational model exists some relationships to guarantee the normalization and the integrity relations between relations [24], as presented in the figure 1.

However, some incomplete information is presented in the data:

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

(a) For the supplier with the identification '1' the rating is unknown but can be selected only by one value of the set '20 or '40'.

(b) The plafond (ex in dollars) of the supplier with the identification '*1*' is unknown but can be selected only by one value of the set '200' or '400'.

(c) The rating of the company with the identification '2', for the supplier '1' and product '50' is unknown but can be selected from the set of values 40, 50 or 60.

A. Knowledge Representation

A logic program is a finite set of clauses in the form:

$$A \leftarrow L_1 \wedge ... \wedge L_m \wedge not \ L_{m+1} \wedge ... \wedge not \ L_n \quad (1)$$

such as $\forall_i N_0$, A is a domain atom and the terms L_i and $not\, L_i$ are domain literals. Weak negation - operator *not* in conventional Logic Programming (LP) - is the ***negation-by-failure***: ***not A*** is true if it is not possible to prove ***A***, and ***not A*** is false when is possible to prove ***A***. This kind of reasoning would be enough in a Closed World Assumption system [3, 5], but is insufficient when there is incomplete information. A suitable logic is needed, one that permits the representation of incomplete, inconsistent and default information and to support non-monotonic reasoning [3]. In this sense an extension of the Logic Program is presented as follows:

Definition 1 – Extended Logic Program

An Extended Logic Program (ELP for short), on the other hand, is a finite collection of rules of the form:

$$q \leftarrow p_1 \wedge ... \wedge p_m \wedge not \ p_{m+1} \wedge ... \wedge not \ p_{m+n} \quad (2)$$

$$?\, p_1 \wedge ... \wedge p_m \wedge not \ p_{m+1} \wedge ... \wedge not \ p_{m+n} \quad (3)$$

*where **?** is a domain atom denoting falsity, and q and every p_i are literals, i.e. formulas* like a *or* $\neg a$ *, being* a *an atom, for* $m, n \in N_0$.

ELP introduces another kind of negation: strong negation, represented by the classical negation sign ¬. In most situations, it is useful to represent ¬*A* as a literal, if it is possible to prove ¬*A*. In EPL, the expressions *A* and *not A*, being *A* a literal, are extended literals, while *A* or ¬*A* are simple literals. Intuitively, *not p* is true whenever there is no reason to believe *p*, whereas ¬*p* requires a proof of the negated literal. Three types of answers to a given question are then possible: true, false and unknown. The representation of null values will be scoped by the ELP. We consider two types of null values: the first will allow for the representation of unknown values, not necessarily from a given set of values, and the second will represent unknown values from a given set of possible values. Many examples of this type of representation could be found in [5, 7, 18, 19, 21, 22, 23]. To reason about the body of knowledge presented in a particular knowledge, set on the base of the formalism referred to above, let us consider a procedure given in terms of the extension of a predicate called *demo*, using ELP as the logic programming language.

Definition 2 - Meta Theorem Problem Solver for an Universe of Discourse with Incomplete Information

A meta theorem problem solver in this context is given by the signature `demo:T,V →{true,false,unknown}`, infers the valuation `V` of a theorem `T` in terms of the truth value `false` (or `0`), truth value `true` (or `1`) and `unknown` (with truth values in the interval `]0,1[)`, according to the following set of productions:

```
demo(T, true) ← T.
demo(T, false) ← ¬T.
demo(T, unknown) ← ¬ not T, not¬T.
```

In the definition 2, the first clause establish that is a question recurring to a knowledge base of positive information; the second clause determines that the questions reveals false recurring to the negative information and the knowledge represented in the level; and the third clause is based on the concept of unknown/incomplete information is connected to that of null values. These elements are atoms that represent abstract concepts with no particular definition, i.e. elements which have a well-defined (or even non-defined) range of values have valid options. Indeed, in the search for an answer, it is postulated that each solution to the problem is to be given in terms of a logic theory, built upon the extensions and the abduciles of the predicates that make their realm, i.e. for all problem solutions in memory and for each property inherited by them, their relevance to the answer to be evaluated will be given in terms of a measure of the quality of the information that a program carries along the time.

Based on the knowledge representation mentioned above the following programs will be drawn:

Program 1 - Knowledge representation in terms of the extension of predicate *suppliers*.

```
1.  ¬suppliers(X,Y,Z)←
              not suppliers (X,Y,Z),
              not abducible_suppliers(X,Y,Z).
2.  abducible_suppliers(1, supplier1, 20).
3.  abducible_suppliers(1, supplier1, 40).
4.  ?((abducible_suppliers(X1,Y1,Z1) ∨
       abducible_suppliers(X2,Y2,Z2)) ∧
      ¬ abducible_suppliers(X1,Y1,Z1) ∧
       abducible_suppliers(X2,Y2,Z2)))
5.  suppliers(2,supplier2,60).
6.  suppliers(3,supplier3,25).
```

In Program 1, the symbol ¬ represents the strong negation, denoting what should be interpreted as false, and the term *not* designates negation-by-failure. The first clause represents the closure of the predicate *suppliers*. The second and third clauses represent the fact that the value of the rating for the predicate *suppliers* is unknown but one knows that it is specifically '*20* or '40'. The fourth clause presents the invariant that implements the *XOR* operator, i.e. it states that the predicate *suppliers* is either *X* or *Y*, but not an amalgam of both.

Program 2 – Knowledge representation in terms of the extension of predicate *p_suppliers*.

```
1.  ¬p_suppliers(X,Y,Z)←
              not p_suppliers (X,Y,Z),
```

```
                        not abducible_p_suppliers(X,Y,Z).
2. abducible_p_suppliers(1,50,200).
3. abducible_p_suppliers(1,50,400).
4. ? ((abducible_p_supplier(X1,Y1,Z1) ∨
         abducible_p_suppliers(X2,Y2,Z2)) ∧
   ¬ (abducible_p_suppliers(X1,Y1,Z1) ∧
        abducible_p_suppliers(X2,Y2,Z2)))
5. p_suppliers(2,51,300).
6. P_suppliers(3,52,500).
```

Program 3 – Knowledge representation in terms of the extension of predicate *companies*.

```
1. ¬companies(X,Y,Z,W,H) ←
                not companies(X,Y,Z,W,H),
                not abducible_companies(X,Y,Z,W,H).
2. abducible_companies(2,1,50,company2,40).
3. abducible_companies(2,1,50,company2,50).
4. abducible_companies(2,1,50,company2,60).
5. companies(1,3,50,company1,35).
6. companies(3,2,51,company3,45).
```

In program 3 the second, third and fourth clauses present the case where the value of the attribute rating of the predicate *companies* is unknown but can be obtained from a set of values 40, 50 or 60.

Program 4 – Knowledge representation in terms of the extension of predicate *products*.

```
1. ¬products(X,Y,Z,W) ←
                   not products(X,Y,Z,W),
                   not abducible_products(X,Y,Z,W).
2. products(50,product1,black,200).
3. products(51,product2,white,150).
4. products(52,product3,blue,300).
```

A. Quality-of-Information

The **Quality-of-Information** (*QoI*) with respect to a generic predicate *P* can be analyzed in four situations and can be measure from the interval [0-1], when the information is positive and negative, when the information is unknown, when the information is unknown but can be selected from one or more values, and when the information is unknown but can be derived from a set of values, but only one can be selected. If the information is know (positive) or false (negative) the quality of the information for the predicate is "1" (4) corresponding to the max value from the known knowledge. For situations where the value is unknown the formula of the quality of information is given by:

$$QoI_P = \lim_{N\to\infty} \frac{1}{N} = 0(N >> 0) \quad (5)$$

For situations when the information is unknown but can be derived from a set of values the *QoI* is therefore given by $QoI_P = 1/Card$ (6), where *Card* denotes the cardinality of the abducibles set for *p*, if the abducibles set is disjoint. If the abducibles set is not disjoint, the quality-of-information is given by:

$$QoI_P = \frac{1}{C_1^{Card} + \cdots + C_{Card}^{Card}} \quad (7)$$

where C_{Card}^{Card} is a card-combination subset, with *Card* elements. The next element of the model to be considered is the ***relative importance*** that a predicate assigns to each of its attributes under observation, i.e. w_{ij} stands for the relevance of attribute *j* for predicate *i*. It is also assumed that the weights of all predicates are normalized, that is:

$$\forall i \sum_{j=1}^{n} w_{ij} = 1 \quad (8)$$

It is now possible to define a predicate's ***scoring function***, i.e., for a value $x = (x_1, ..., n)$ in the multi dimensional space defined by the attributes domains, which is given in the form:

$$V_i(x) = \sum_{j=1}^{n} w_{ij} * V_{ij}(x_j) \quad (9)$$

it is viable to measure the *QoI* that occurs as a result of invoking a logic program to prove a theorem (e.g. Theorem), by posting the $V_i(x)$ values into a multi-dimensional space and projecting it onto a two dimensional one (figure 2, 3, 4).

III. INTEGRATING INCOMPLETE INFORMATION INTO THE RELATIONAL DATA MODEL

The first aim of this work is to present a computational model under the Extended Logic Programming paradigm [6] (definition 1) to knowledge representation and reasoning in environments with incomplete information. The objective is to discover which theories (or logical programs) are able to solve a problem and with the set of those theories, which one's is the best to solve a specific problem. In our approach, to evaluate the theories we use the quantification of the quality-of-information [5, 18] that stems from those theories. The selection of the best theory will be based on the relation order of its *QoI* value. In practical terms, in the end of the creation of the model we will achieve a value (and a theory) that corresponds to the best quantification of the universe of discourse. Knowing this optimal value, we will get the best logical mathematical theory (represented as logic programs), and consequently the best modulation of the system for the problem to solve. In our approach, we will not get a solution to a particular problem, but rather a logic representation (or program) of the universe of discourse. The second contribution of our approach is to present a new representation of the incomplete information, materialized by the theories referred above. This representation follows the semantic of the relational data model [24] and permitting to explore the advantages and potentialities of the relational algebra operations.

Following the problem to submit to the inference engine the question (10): "Which suppliers are able to provide products with black color", we will obtain the possible solutions and its confidence degree is given in terms of theories (or logic programs) to solve the problem:

The extended logic program or Theory 1
{

```
¬suppliers(X,Y,Z)←
        not suppliers (X,Y,Z),
        not abducible_suppliers(X,Y,Z),
  abducible_suppliers(1, supplier1, 20),
  abducible_suppliers(1, supplier1, 40),
  ?((abducible_suppliers(X1,Y1,Z1) ∨
     abducible_suppliers(X2,Y2,Z2)) ∧
     ¬ abducible_suppliers(X1,Y1,Z1) ∧
```



```
    abducible_suppliers(X2,Y2,Z2))),
   ¬p_suppliers(X,Y,Z)←
        not p_suppliers (X,Y,Z),
        not abducible_p_suppliers(X,Y,Z),
   abducible_p_suppliers(1,50,200),
   abducible_p_suppliers(1,50,400),
   ?((abducible_p_supplier(X1,Y1,Z1) ∨
     abducible_p_suppliers(X2,Y2,Z2)) ∧
     ¬ (abducible_p_suppliers(X1,Y1,Z1) ∧
     abducible_p_suppliers(X2,Y2,Z2)))
   ¬products(X,Y,Z,W) ←
          not products(X,Y,Z,W),
          not abducible_products(X,Y,Z,W),
     products(50,product1,black,200)
}
```

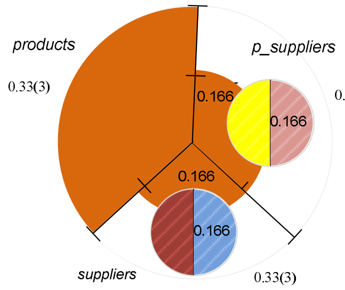

Fig. 2. A measure of the quality-of-information for the logic program or theory 1.

The extended logic program or Theory 7

```
{
 ¬suppliers(X,Y,Z)←
       not suppliers (X,Y,Z),
       not abducible_suppliers(X,Y,Z),
 abducible_suppliers(1, supplier1, 20),
 abducible_suppliers(1, supplier1, 40),
 ¬p_suppliers(X,Y,Z)←
        not p_suppliers (X,Y,Z),
        not abducible_p_suppliers(X,Y,Z),
 abducible_p_suppliers(1,50,200),
 abducible_p_suppliers(1,50,400),
 ? ((abducible_p_supplier(X1,Y1,Z1) ∨
    abducible_p_suppliers(X2,Y2,Z2)) ∧
   ¬ (abducible_p_suppliers(X1,Y1,Z1) ∧
     abducible_p_suppliers(X2,Y2,Z2))),
 ¬companies(X,Y,Z,W,H) ←
          not companies(X,Y,Z,W,H),
          not abducible_companies(X,Y,Z,W,H),
  abducible_companies(2,1,50,company2,40),
  abducible_companies(2,1,50,company2,50),
  abducible_companies(2,1,50,company2,60),
 ¬products(X,Y,Z,W) ←
            not products(X,Y,Z,W),
            not abducible_products(X,Y,Z,W),
   products(50,product1,black,200).
}
```

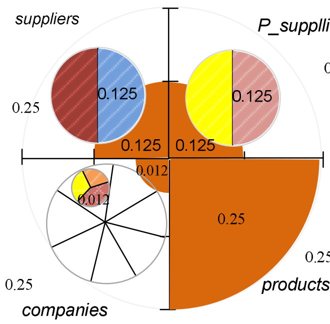

Fig. 3. A measure of the quality-of-information for the logic program or theory 7.

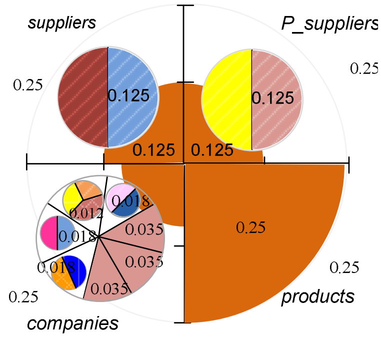

Fig. 4. A measure of the quality-of-information for the global logic program or theory.

A. Adding Incomplete Information to the Relational Data Model

Any computer system stores and processes information. Database [24] and knowledge based systems deal with pieces of the real world and may be assessed in terms of the way they handle the information available. For instance, the relational data model presents an approach to represent data as two-dimensional tables, named *relations*. In this sense, it is possible to represent the extensions of the predicates that make the theories referred to above, as terms in the form:

$$R(a_1,\dots,a_n,\ relevance,\ Truth\ Value) \qquad (10)$$

where R denotes the name of the predicate (or relation), $a_1,..,a_n$ indicate the predicate attributes, and *relevance* and *Truth Value* (or *QoI* for short) stand for themselves. It is now possible to represent incomplete information in terms of the relational data model, and use all the potential of the relational algebra operators. As an example, let us consider the Theory 1 referred to above, which is made in terms of the extensions of the predicates:

```
suppliers(#idS, nameS,RatingS,
             Relevance,TruthValue)
p_suppliers(#idS,#idP,plafond,Relevance,
            TruthValue)
companies(#idC,#idS,#idP,nameC,Rating,
            Relevance,TruthValue)
products(#idP,nameP,Color,Price,Relevance,
          TruthValue)
```

suppliers	IdS	nameS	ratingS	Rel	TV
	1	supplier1	20	0.25	0.166
	2	Supplier2	60	0.3	0.166

p_suppliers	IdS	idP	plafond	Rel	TV
	1	50	200	0.15	0.166
	2	51	200	0.2	0.166
	1	50	400	0.25	0.33

companies	IdC	idS	idP	nameC	Rat	Rel	TV
	2	1	50	company1	200	0.7	0.66

products	idP	nameP	Color	Price	Rel	TV
	50	product1	black	200	0.5	0.33
	51	product2	white	250	0.6	0.33

Now, let us suppose that we intend to list the suppliers that at present are able to supply products of black color. In order to fulfill this goal, we consider the operator Ψ, defined as follows: Being *A* and *B* two database relations, having *A* the

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

attributes $(K, X_1..X_n, Rel, TV)$, such that $n \geq 0$ and B the attributes $(K, Y_1..Y_m, Rel, TV)$, such that $m \geq 0$, and

$$A\ \Psi_{(K)}\ B = \Pi_{K,X1..Xn,\ Y1..Ym,((A.Rel+B.Rel)/2,(B.TV+B.TV))}\ (A \infty_K B)$$

where Π and ∞ denote, respectively, the relational algebra operations of projection and joining. It is now possible to get an answer to the query referred to above, as it is depicted below:

answer=(suppliers $\Psi_{(\#idS)}$ (($\sigma_{plafond>0}$(p_suppliers)) $\Psi_{(\#idP)}$ ($\sigma_{Color=black}$(products)))

where σ stands for the selection operator in relational algebra, and answer is given in terms of the attributes:

answer(*#idS,#idP,name,plafond,nameP,Color,Price,newRelevance,newTruthValue*)

being the extension of relation answer ordered according a degree of confidence (i.e. in terms of the *newTruthValue*) given below:

answer(1,50,supplier1,200,product1,black, 200,0.3,0.66)
answer(2,50,supplier2,400,product1,black, 200,0.35,0.826)

IV. Conclusion and Future Directions

In recent years, formalisms have been proposed to handle the problem of uncertainty, incompleteness in logic programs and databases, in order to deal with uncertain information. However, qualitative models and qualitative reasoning have been around in Artificial Intelligence research for some time, in particular due the growing need to offer support in decision-making processes. Our approach to the evaluation of the quality of knowledge that stems out from logic programs may become a point of departure. In this paper, under the Extended Logic Programming paradigm to knowledge representation and reasoning, we present an evaluative perspective of such an approach. In our work we use the quantification of the quality-of-information that stems out from a logic program to select the best theories or logic programs involved. Additionally, we present a new way to represent incomplete information using the relational data model. It is therefore possible to use the potentialities of the relational algebra, or the potential of the Structured Query Languages to make inferences.

References

[1] R. Kowalski, “The logical way to be artificially intelligent”, in: Toni, F., Torroni, O. (eds), Proceedings of the CLIMA VI.LNCS (LNAI), pp.1-22, Springer, Heidelberg, 2006.

[2] F. Sheridan, “A Survey of techniques for inference under uncertainty”, Artificial Intelligent Review 5(1):89, 1991.

[3] M. L. Ginsberg, “Readings in Nonmonotonic Reasoning”, Los Altos, Califórnia, EUA, Morgan Kauffman Publishers, Inc, 1991.

[4] L. M. Pereira,H. Anh, “Evolution Prospection”, in: K. Nakamatsu (ed.), Procs. First KES Intl. Symposium on Intelligent Decision Technologies (KES-IDT'09), Springer Verlag book in Engineering Series, Himeji, Japan, 2009.

[5] C. Analide, P. Novais, J. Machado, J. Neves, “Quality of Knowledge in Virtual Entities”, in: Encyclopedia of Communities of Practice in Information and Knowledge Management, pages: 436-442. Idea Group Inc, USA, 2006.

[6] J. Neves, “A Logic Interpreter to Handle Time and Negation in Logic Data Bases”, in Proceedings of ACM’84, The Fifth Generation Challenge, pp. 50-54, 1984.

[7] J. Machado, F. Andrade, J. Neves, P. Novais, C. Analide, “Formal Models in Web Based Contracting”, in the 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, (WI-IAT 2006 Workshop Proceedings), Hong Kong, ISBN 0-7695-2749-3, pp 548-551, 2006.

[8] Subrahmanian, V.: Probabilistic databases and logic programming, In Proc. of the 17th International Conference of Logic Programming, 2001.

[9] L. A. Zadeh, “Fuzzy Logic”, in the MIT Encyclopedia of the Cognitive Sciences. R. A. Wilson and F. C. Keil, MIT Press, 2001.

[10] Y. Loyer, N. Spyratos, D. Stamate, “Parametrised Semantics for Logic Programs - a unifying framework”, Theoretical Computer Science, 308 (1-2), 2003.

[11] G. Shafer, “The Dempster-Shafer theory”, Encyclopedia of Artificial Intelligence, Second Edition. S. C. Shapiro, Wiley, 1992.

[12] A. Kakas, R. Kowalski, F. Toni, “The role of abduction in logic programming”, in Gabbay, D., Hogger, C., and Robinson, I., editors, Handbook of Logic in Artificial Intelligence and Logic Programming, volume 5, pages 235-324, Oxford U.P., 1998.

[13] K. D. Forbus, ”Qualitative reasoning”, CRC Handbook of Computer Science and Engineering: 715-733, 1996.

[14] B. Kuipers, ”Qualitative reasoning: modeling and simulation with incomplete knowledge”, MIT press, 1994.

[15] P. Lucas, “Quality checking of medical guidelines through logical abduction”, Proc. of AI-2003 20: 309-321, 2003.

[16] A. Hommersom, P. J. F. Lucas, P. van Bommel, “Checking the quality of clinical guidelines using automated reasoning tools”, Theory and Practice of Logic Programming 8(5-6): 611-641, 2008.

[17] D. Poole, “A methodology for using a default and abductive reasoning system”, International Journal of Intelligent Systems **5**(5): 521-548, 1990.

[18] J. Neves, J. Machado, C. Analide, A. Abelha, L. Brito, “The Halt Condition in Genetic Programming. . In: J. M. Neves, M. F. Santos, J. M. Machado (eds.), Progress in Artificial Intelligence, Procs. 13th Portuguese Intl.Conf. on Artificial Intelligence (EPIA'07), pp. 73-86, Springer LNAI 4784, Guimarães, Portugal, 2007.

[19] J. Machado, A. Abelha, P. Novais, J. Neves, J. Neves, “Quality of service in healthcare units”, Int. J. Computer Aided Engineering and Technology, Vol. 2, No. 4, pp. 436-449, 2010.

[20] P. Novais, F. Andrade, J. Machado, J. Neves, “Agents, Trust and Contracts, in Information Communication Technology Law, Protection and Access Rights: Global Approaches and Issues, Portela I., Cunha M. (Eds), IGI Global, 2010.

[21] J. Machado, M. Miranda, G. Pontes, D. Santos, M. F. Santos, A. Abelha, J. Neves, “ Intelligent Agents and Medical Moral Dilemmas”, in Proceedings of the 8th WSEAS International Conference on Applied Computer & Applied Computational Science (ACACOS '09), Hangzhou, China, 2009.

[22] R. Costa, P. Novais, G. Marreiros, C. Ramos, J. Neves: VirtualECare: Group Decision Supported by Idea Generation and Argumentation. The 9th IFIP Working Conference on Virtual Enterprises (PRO-VE 2008). Poznan, Poland, 2008.

[23] L. Lima, P. Novais, J. Bulas Cruz, “A Process Model For Group Decision Making With Quality Evaluation”, in Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing and Ambient Assisted Living, Omatiu S., et al. (Eds.) LNCS 5518, Springer-Verlag, ISBN 978-3-642-02480-1, pp. 566-573, 2009

[24] H. Molina-Garcia, J. Ullman, J. Widom, “Database systems the complete book”, Prentice hall eds, 2002.

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)