Deduction in a GEFRED database using Datalog

Ignacio J. Blanco*
Dept. Languages and Computer Science
University of Almería
Carretera de Sacramento S/N
La Cañada de San Urbano, 04120, Almería, SPAIN
iblanco@ual.es

Olga Pons, Jose M. Serrano, and M. Amparo Vila
Dept. Computer Science and Artificial Intelligence
University of Granada
Periodista Daniel Saucedo Aranda S/N
18071, Granada, SPAIN
{opc,jmserrano,vila}@decsai.ugr.es

Abstract

In this paper, some considerations on extension of the Datalog language are proposed in order to apply this language on a GEFRED fuzzy relational database. For this integration, we use some extended comparators from the GEFRED model and an existing generalization of the concept of rule to work with flexible values.

Keywords: Datalog, Fuzzy Relational Database, GEFRED Model, Logical Database.

1 Introduction

The GEFRED Model was introduced by Medina et al. [9] to integrate several previous proposals which extend the Relational Model with capabilities of handling and storing imprecise and incomplete information by the extension of the concepts of domain and relation so imprecise and incomplete information can be represented. Because of the difficulty when doing a low-level handling on these extended relations and domains, Galindo et al. [7] and Blanco et al. [4] developed an extended SQL language.

At the same time, Pons et al. introduced FREDDI[8, 10], an architecture to represent logical information (rules in clausal form) into the Relational Model.

Using these extensions as an starting point, Pons et al. [8, 10] and Blanco et al. [3, 2, 5] considered that merging them can give the capability of making classical deductions on flexible data in the Relational Model.

This fusion process requires the extension of the concept of rule used by the FREDDI architecture. These generalization of rule was introduced by Blanco et al. [5]. This extension does not focus on the domains but on predicates and relations among them. The FREDDI Architecture has to be modified so that new types of conditions, added by the GEFRED Model, can be represented in order to soften the relations among predicates in the rule. An approach to the modification can be found in Blanco[1].

As a result of this fusion, Blanco et al. [5] have proposed a concrete algorithm (intuitively based on the Prolog mechanism) to deduce new data from imprecise or incomplete information stored in a GEFRED database.

This approach is based on the application of all predicates of a rule to an empty tuple. When all predicates are applied, the calculated set of values (the string of instantiation) is a fact of the calculated predicate. Nevertheless, this approach focus on data and not on the structure of rules when the algorithm explores the expansion tree.

An alternative point of view is Datalog[6]. This approach is based on the translation of rules into relational languages in order to apply them on a relational database. In this paper, we try to extend this translation so it can be used on a GEFRED database.

In section 2 we resume briefly the Datalog Model, the generalized concept of rule to be translated and the concepts of the GEFRED Model that determine this translation. In section 3, we present our proposal for the extension of Datalog. Finally, in section 4, we explain the conclusions and the future work to develop a complete extension for Datalog.

2 Previous models and concepts

In this section, we resume some important concepts to extend the Datalog language.

2.1 GEFRED concepts

The GEFRED Model extends the concept of domain so imprecise or incomplete information can be represented. In the same way, the concept of comparison operator has to be modified in order to work with this kind of domains.

An **extended fuzzy comparator** θ is a fuzzy relation defined on the domain of discourse U as follows:

$$\begin{array}{c}\theta : U \times U \rightarrow [0,1] \\ \theta(u_i, u_j) \mapsto a\end{array}$$

Let be D a fuzzy domain on U and a function Θ^θ defined as follows:

$$\begin{array}{c}\Theta^\theta : D \times D \rightarrow [0,1] \\ \Theta^\theta(\tilde{d_1}, \tilde{d_2}) \mapsto [0,1]\end{array}$$

Then Θ^θ is named a **generalized fuzzy comparator** on D and generated by the extended comparator θ if it verifies:

$$\Theta^\theta(\tilde{d_1}, \tilde{d_2}) = \theta(d_1, d_2), \forall d_1, d_2 \in U$$

where $\tilde{d_1}$ and $\tilde{d_2}$ are the possibility distributions $\{1/d_1\}$ and $\{1/d_2\}$ defined on the values d_1 and d_2, respectively.

2.2 Generalized rule for deduction with flexible data

Let be:

- P a n-ary predicate and $Q_i, i \in \{1, \dots, m\}$ n_i-ary predicates.
- $X_i, i \in \{1, \dots, n\}$ and $Y_{j,k}, j \in \{1, \dots, m\}, k \in \{1, \dots, n_j\}$ variables having their values in GEFRED domains $D_{X_i}, i \in \{1, \dots, n\}$ and $Y_{j,k}, j \in \{1, \dots, m\}, k \in \{1, \dots, n_j\}$.

We define a **generalized rule for deduction with fuzzy data**, or simply a **generalized rule**, a rule with the following structure:

$$\begin{array}{l}P(X_1, X_2, \dots, X_n) : - \\ \quad Q_1(Y_{1,1}, Y_{1,2}, \dots, Y_{1,n_1}) \wedge \dots \wedge \\ \quad Q_m(Y_{m,1}, Y_{m,2}, \dots, Y_{m,n_m}) \wedge \Psi\end{array} \tag{1}$$

where Ψ is a formula:

$$\begin{array}{l}\Psi \equiv \bigwedge (X_i =_{\alpha_{i,j,k}} Y_{j,k}) \wedge \\ \quad \bigwedge (Y_{j,k} =_{\beta_{j,k,l,p}} Y_{l,p}) \wedge \\ \quad \bigwedge \phi_{j,k,l,p}(\Theta^\theta_{j,k,l,p}(Y_{j,k}, Y_{l,p}), \gamma_{j,k,l,p})\end{array} \tag{2}$$

and where:

- $\bigwedge A_i$ represents $A_1 \wedge A_2 \wedge \dots \wedge A_n$ summarized,
- conditions $(X_i =_{\alpha_{i,j,k}} Y_{j,k})$ refer to variables shared between a predicate in the body of the rule and the predicate in the head,
- conditions $(Y_{j,k} =_{\beta_{j,k,l,p}} Y_{l,p})$ refer to variables shared between two predicates in the body of the rule,
- expressions $\phi_{j,k,l,p}(\Theta^\theta_{j,k,l,p}(Y_{j,k}, Y_{l,p}), \gamma_{j,k,l,p})$ refer to comparisons explicitly introduced by the user to constraint the rule meaning. In those ones, $\phi_{j,k,l,p}$ is the threshold function for the comparison degree, $\Theta^\theta_{j,k,l,p}$ is a generalized fuzzy comparator defined on the classical comparator θ, and $\gamma_{j,k,l,p}$ is the minimum degree for the comparison to be evaluated to *true*.
- every variable in the head of the rule is linked to one or more variables in the body of the rule, and
- every variable in the body of the rule is linked to a variable in the head or in the body of the rule.

2.3 The Datalog Language

Datalog is a logic programming language that was designed to be used as a language for querying databases[6]. It is a non procedural, set-oriented and non order-sensitive (with respect to the predicates in a rule) language with no special predicates nor function symbols. Although it is syntactically equal to Prolog, the evaluation of the pair of rules shown in expression (3) does not lead to an endless loop.

$$\begin{array}{l}ancestor(X,Y) \leftarrow ancestor(Z,Y) \wedge parent(X,Z) \\ ancestor(X,Y) \leftarrow parent(X,Y)\end{array} \tag{3}$$

A **base conjunction** is a list of predicates of the database and arithmetic predicates, as an instance, the one shown in expression (4).

$$\begin{array}{l}bc_1 : db_1(X_1, a, X_3), db_2(X_3, X_4, X_5), (X_4 = b), \\ \quad db_3(X_5, X_6, X_7), (X_6 = c)\end{array} \tag{4}$$

A base conjunction can be a part of a Prolog rule or a set of Prolog rules, as seen in the translation of expression (4) into the expression (5).

$$\begin{aligned} &go_1(X_1,X_4) :- db_1(X_1,a,X_3), p_1(X_3,X_4).\\ &p_1(Y_1,Y_4) :- db_2(Y_1,Y_2,Y_3), (Y_2 = b), p_2(Y_3,Y_4).\\ &p_2(Z_1,Z_3) :- db_3(Z_1,Z_2,Z_3), (Z_2 = c). \end{aligned} \tag{5}$$

A **connected conjunction** is a base conjunction that can not be divided in two parts with no common variables. A **string** is a connected conjunction only if there exists an internal sequence of predicates $db_1, \ldots, db_n$ where the predicate $db_i, \forall i \in \{2, \ldots, n-1\}$ only has shared variables with predicates db_{i-1} and db_{i+1}. A **cyclic conjunction** is a string where the predicate db_n has shared variables with db_{n-1} and db_1.

A **shared variable** in a base conjunction is a variable that is shared with other predicates that are in the rule and not in the base conjunction.

Finally, a base conjunction is equivalent to a *join* $\bowtie$ operation in the database, that represented in the Relational Algebra is an expression including the following elements:

- selection on relations corresponding to predicates in the database when there are variables which value has been fixed with another variable or variables appearing in an arithmetic predicate:

$$R(X,a) \equiv \sigma_{Y=a}(R)$$

- join of relations corresponding to predicates in the database having shared variables:

$$R(X,Y), S(Y,Z) \equiv R \bowtie S$$

- a projection applied on the attributes corresponding to the shared variables of the base conjunction.

For example, let us suppose a rules go_1 having a base conjunction bc_1 as shown in expression (6).

$$\begin{aligned} &go_1(X_1,X_7) :- bc_1(X_1,X_7).\\ &bc_1(X_1,X_7) :- db_1(X_1,a,X_3), db_2(X_3,b,X_5),\\ &\quad db_3(X_5,c,X_7). \end{aligned} \tag{6}$$

If the translation of base conjunctions is applied to these rules we obtain:

$$jc_1 : \Pi_{X_1,X_7}((\sigma_{X_2=a}(R_1) \bowtie \sigma_{X_4=b}(R_2)) \bowtie \sigma_{X_6=c}(R_3))$$

where schemes of relations R_1, R_2 and R_3 are $\{X_1,X_2,X_3\}$, $\{X_3,X_4,X_5\}$ and $\{X_5, X_6,X_7\}$, respectively.

3 The Datalog Extension

The aim of this work is to extend the Datalog language in order to be applied on a fuzzy relational database based on the GEFRED Model.

As seen in section 2.3, a rule can be translated into a join sentence, based on the selection and the projection operations. The first problem when trying to apply this translation to generalized rules shown in section 2.2 is the type of conditions that can be represented in the selection. The classical comparison operators are not suitable when working with imprecise values. We propose to extend the set of possible comparators with the ones introduced in GEFRED and incorporated to generalization of the concept of rule.

A generalized rule as seen in expression (1) is translated into the query shown in expression (7). The formula Ψ, shown in expression (2), can be seen as $\Psi = \Psi_X \wedge \Psi_Y \wedge \Psi_C$, where Ψ_X is the sub-formula containing all comparison predicates that relate variables in the head and the body of the rule, Ψ_Y is the sub-formula containing those ones that relate variables in the body of the rule, and Ψ_C is the sub-formula containing all comparison predicates that establish constraints on the resulting set of facts.

$$\Pi_{(X_1,X_2,\ldots,X_n)}(\sigma_{\Psi_X \wedge \Psi_C}(R_x)) \tag{7}$$

The relation R_x is a relation obtained by the query:

$$R_x = \begin{cases} R_y \bowtie R_z \\ \Pi_{(Y_{i,1},Y_{i,2},\ldots,Y_{i,n_i})}(Q_i) \end{cases} \tag{8}$$

that is, the relation R_x can be:

- a relation $\Pi_{(Y_{1,1},Y_{1,2},\ldots,Y_{1,n_1})}(Q_1)$ corresponding to the predicate in a rule with an only predicate,
- a relation corresponding to a joint operation $\Pi_{(Y_{i,1},Y_{i,2},\ldots,Y_{i,n_i})}(Q_i) \bowtie \Pi_{(Y_{j,1},Y_{j,2},\ldots,Y_{j,n_j})}(Q_j)$ where $\exists k \in \{1,\ldots,n_i\}, \exists l \in \{1,\ldots,n_j\}, (Y_{i,k} =_{\beta_{i,k,j,l}} Y_{j,l})$ is a sub-formula within Ψ_Y, or
- a relation corresponding to a joint operation $R_y \bowtie R_z$ where schemes of relations R_y and R_z have

shared variables. Two variables $Y_{i,k}$ and $Y_{j,l}$ in the scheme of relations R_y and R_z, respectively, are shared only if $(Y_{i,k} =_{\beta_{i,k,j,l}} Y_{j,l})$ is in Ψ_Y.

When calculating facts satisfying a rule, Datalog translates every rule in a joint operation so the result if this translation is an only joint operation containing others. But when solving this query, deeper predicates are calculated earlier. With this consideration, conditions in sub-formula Ψ_X could not be applied until values of variables X_i are known, so translation would be re-written as shown in expression (9).

$$\sigma_{\Psi_X}(\Pi_{(X_1,X_2,\ldots,X_n)}(\sigma_{\Psi_C}(R_x))) \quad (9)$$

4 Conclusion and future work

In the present work, we have shown a first approach to an extension of Datalog so it can be applied on a GEFRED database. This extension have been built on the basis of the GEFRED Model, an extension of logical rules for a relational deductive database and the Datalog language.

As seen in section 3, the evaluation of some conditions have to be delayed so it would be interesting to study the order when applying all conditions in the translated query.

Furthermore, when working on imprecise values, values are not completely equal, so the equality degree has to be represented and operated in order to give an imprecision degree with the resulting fact.

References

[1] I. Blanco. *Deducción en Bases de Datos Relationales Difusas*. PhD thesis, E. T. S. I. Informática, Universidad de Granada, Spain, 2001.

[2] I. Blanco, J. C. Cubero, O. Pons, and M. A. Vila. An implementation for fuzzy relational databases. In G. Bordogna and G. Passi, editors, *Recent Research Issues on the Management of Fuzziness in Databases*, Studies in Fuzziness and Soft Computing, pages 183–207. Physica-Verlag, 2000.

[3] I. Blanco, F. Cuenca, O. Pons, and M. A. Vila. Implementation of an inference engine for fuzzy databases. In *Proceedings of the EUSFLAT/ESTYLF 99 Joint Conference*, pages 493–496. European Society for Fuzzy Logic and Technology, 1999.

[4] I. Blanco, N. Marín, O. Pons, and M. A. Vila. An extension of data description language (ddl) for fuzzy data handling. In Larsen, Kacprzyk, Zadrozny, Andreasen, and Christiansen, editors, *Flexible Query Answering Systems, Recent Advances*, Advances in Soft Computing, pages 54–64. Physica-Verlag, 2000.

[5] I. Blanco, M. J. Martín-Bautista, O. Pons, and M. A. Vila. A mechanism for deduction in a fuzzy relational database. In *Proceedings of the 9th Information Processing and Management of Uncertainty in Knowledge-based Systems IPMU 2002 Conference*, pages 291–298, Annecy, France, July 2002.

[6] S. Ceri, G. Gottlob, and L. Tanca, editors. *Logic Programming and Databases*. Surveys in Computer Science. Springer-Verlag, 1990.

[7] J. Galindo. *Tratamiento de la Imprecisión en Bases de Datos Relacionales: Extensión del modelo y adaptación de los SGBD actuales*. PhD thesis, Department of Computer Science and Artificial Intelligence, University of Granada, España, 1999.

[8] J. M. Medina, O. Pons, J. C. Cubero, and M. A. Vila. Freddi: A fuzzy relational deductive database interface. *International Journal of Intelligent Systems*, 12:597–613, 1997.

[9] J. M. Medina, O. Pons, and M. A. Vila. Gefred. a generalized model of fuzzy relational databases. *Information Sciences*, 1994.

[10] O. Pons, J. M. Medina, J. C. Cubero, and A. Vila. An architecture for a deductive fuzzy relational database. In Z.W. Ras and M. Michaliewicz, editors, *Foundations of Intelligent Systems*, volume 1079 of *Lectures Notes in Artificial Intelligence*. Springer, 1996.