(online) = ISSN 2285 – 3642
ISSN-L = 2285 – 3642
Journal of Economic Development, Environment and People
Volume 7, Issue 1, 2018

URL: http://jedep.spiruharet.ro
e-mail: office_jedep@spiruharet.ro

Deductive systems for BigData integration

Radu BUCEA-MANEA-ȚONIȘ[1]

[1] Hyperion University, Titu Maiorescu University

Abstract. *The globalization is associated with an increased data to be processed from E-commerce transactions. The specialists are looking for different solutions, such as BigData, Hadoop, Datawarehoues, but it seems that the future is the predicative logic implemented through deductive database technology. It has to be done the swift from imperative languages, to not declaratively languages used for the application development. The deductive databases are very useful in the student teaching programs, too. Thus, the article makes a consistent literature review in the field and shows practical examples of using predicative logic in deductive systems, in order to integrate different kind of data types.*

Keywords: deductive systems, predicative logic, Datalog, parse tree, grammars

JEL Codes: *M15*

1. Introduction

The volume of data collected and the type of data that has to be integrated and analyzed is in constant increasing trend. The knowledge is presented in lots of formats: relational datasets, XML databases, rule bases, ontologies, and this make the classical approach not reliable. In order to find a common denominator, the predicative logic seems to be a solution for correct and appropriate data interpretation in modern companies… From different version of databases, the predicative logic can extract common features or differences, based on predicate dependency and rule predicate graphs, for example. The ontologies, frequently used in intelligent systems, used in semantic web application help the examination of the “use of abstractions of rule bases by predicate dependency and rule predicate graphs” [Seipel, 2017].

The article shows an appropriate method to extract information from different types of data, using predicative logic through deductive databases.

2. Literature review

Deductive systems were originally designed for educational purposes, helping students to translate the relational data structure model in a deductive data model, where “atomic data are arranged in predicates which can be understood as relations, i.e., relational tables” [Sáenz-Pérez, 2018]. Students are thought to integrate different kind of data type in a predicative logic. In this article the author presents interesting examples for students, helping them to understand the syntax.

It seems that modern e-learning platforms, such as Moodle, Blackboard, Academika, provides the information in a fixed-sequence, without customizing it in accordance with students' differences in background knowledge. The Zone of Proximal Development (ZPD) theory states that the frustration, boredom and incomprehensiveness of the information can be prevented. In this regard was developed a new framework that facilitate the users online distance interaction. "The adaptive e-learning systems include real-time dynamic adaptation and context modelling in addition to the learner model, the domain model and the pedagogical strategy" [Maravanyika, 2017].

New technologies (IoT), such as Virtual Reality, can offer a complete experience in the e-learning system, e.g. a better (3D) view that supports demonstration and promotion, the development of creativity, increased understanding, less study time and fun during learning [Manea, 2018].

Deductive systems are used in business field, too. Facebook is a good example of application that integrates different types of data. Facebook developed a new query protocol that provides a unified interface between the client and the server for data fetching and manipulation application layer, in order to work with different types of data handling of various sources and to combine, e.g., relational with NoSQL databases. This protocol can be used as interface for deductive databases, too [Nogatz, 2017].

Another modern solution to manipulate different format of date is Open Rule Bench. Is dedicated to rule engines, including deductive databases. An example of translation of Datalog to C++ based on a method that "pushes" derived tuples immediately to places where they are used was provided by [Brass,2017].

Other studies demonstrate that the semantics of deductive databases can be implemented in the spiking neural P systems model, allowing the integration of symbolic reasoning systems based on logic and connectionist systems based on the functioning of living neurons [Diaz-Pernil, 2018].

Further bellow we demonstrate how natural language can be translated in first order logic statements and stored into a knowledge database.

3. Formal systems

In the essay on Universal Language, GW Leibnitz proposes replacing words with numbers so that the language form corresponds to the logical sense. For example, if the "animal" word is associated with figure 2, and the "rational" word is associated with figure 3, it results that man will be the product of 2 and 3, so 6. When constructing sentences, the author proposes as a grammatical rule the exact division of the subject number to the predicate number in in the case of true affirmations. Basically, the new rational language will make "any reasoning a kind of arithmetic calculation" so that the correspondence that exists between things and ideas is taken into account. The author's approach follows the emphasis on the relation between words and concepts, because we can have a simple, finite alphabet that would render the multitude of infinite concepts [Leibnitz, 2015].

A formal system is (D, R) where ***d*** is a set of data structures, and ***R*** is a set of rules that determines what transitions between objects in ***D*** are allowed. The characteristics of a formal system are:

- Completeness: (D, R) does everything that the set of rules requires.
- Correctness: (D, R) does nothing to ban the set of rules.
- Simplicity: the formal system must meet its set of rules or come closer to meeting the set of rules with minimal complexity.
- Naturalness: the formal system should be in line with the intuition.

Formal systems based on text mimics how natural language is generated and semantically regulated by grammar.

3.1. General formal systems based on text

General formal systems based on text are formal systems whose data structures are strings/symbols, and whose transformations are rewriting the rules.

Let Σ be a finite symbol alphabet. Let Σ* a set with all strings of length finite and which can be formed using symbols from Σ. The elements Σ* will be called words on the Σ set.

A rewrite rule S is an expression of α ⇝ β, where α and β are in Σ*. Note that S* contains the word Null. A rule α ⇝ β means that it is allowed to replace α with β in any context.

Exemple:

Suppose we have the alphabet Σ = {p, g,-}, where xp-gx- ↔ x={-} , and the following rule P, after [Hofstadter, 2015]:

```
xpygz ⇝ xpy-gz-
```

Prove that the W be the word --p-g--- ⊂ Σ*

To apply the set of rules P to the word W, we replace each part of W with the corresponding P rule. We keep -- as --, replace - with --, and --- with ---- . Therefore we have the property W ⇒ --p--g---- and the W is a theorem because evaluates TRUE the axiom xp-gx-.

Artificial languages are generated using formal grammars.

Grammars

A grammar is a rewriting system P along with an initial word I. Such grammar (P, I) generates language.

Given the following grammar, let’s find a parse tree for the string 1 + 2 * 3[Nelson, 2017]:

```
<E> --> <D>
<E> --> <F>
<E> --> <G>
<E> --> <H>
<E> --> <I>
<E> --> ( <E> )
<F> --> <E> + <E>
<H> --> <E> - <E>
<G> --> <E> * <E>
<I> --> <E> / <E>
<D> --> 0 | 1 | 2 | ... 9
```

The parse tree is:

```
        E
       / \
      E + E
     / \   \
    E * E   N
   /     \   \
  N       N   1
 /         \
3           2
```

Fig. 1: The parse tree for the string 1 + 2 * 3

```
E(F(E(N 1),G(E(N 2),E(N 3)))--> E(E(N 1),E(E(N 2),E(N 3)))
```

A more intuitive way of expressing more complex statements like those found in natural languages is predicative logic.

Predicative logic

A predicate is a function whose code is the truth values {T, F}.

For example, the uncle (X, Y), who asserts that X is uncle for Y, defines a predicate. For any value pair X and Y, this statement will be either TRUE or FALSE.

The number of variables that appear in a predicate and that can be instantiated in this way is called the predicate arithmetic.

For example, the red predicate (X) means that X is red and has the arithmetic 1, and the predicate between (X, Y, Z) means that X is between Y and Z, having the arithmetic 3.

Monadic predicates, such as hair (X), are called properties.

The **Logical operators** used in the predicative logic are presented below:

1. ***and***, symbolically written as ∧
2. ***or***, symbolically written as ∨
3. ***non***, symbolically written as ¬
4. ***implies***, symbolically written as →
5. ***if and only if***, symbolically written as ↔
6. ***for all***, symbolically written as ∀
7. ***exists***, symbolically written as ∃

Below we present an example from [Bird & all, 2009], explaining how the sentence “Everybody admires someone” is transformed in predicative logic

It seems there are two predicates involved here admire (x, y) that says ***x*** admires ***y***, and person (x) that says x is a person. We mean that if x is a person, then there must be at least one-person ***y*** that ***x*** will admire.

There are (at least) two ways of expressing this in first-order logic:

```
        a. all x.(person(x) -> exists y.(person(y) & admire(x,y)))
        b. exists y.(person(y) & all x.(person(x) -> admire(x,y)))
```

The next step is to demonstrate how the predicative logic helps querying databases using Datalog language.

3.2. Deductive databases

Deductive databases are a set of basic relationships containing explicit tuples [Date, 2005].

A query is -> the evaluation of a Boolean expression over explicit relations and tuples or the demonstration as the specified formula represents the logical consequence of the basic axioms, so it is a theorem.

A database is form of basic axioms set and deductive axioms. There are two types of databases:

- extensive database which is a set of basic axioms,
- intensive database which is a form of deductive axioms and integrity restrictions.

The characteristics of deductive databases are listed below:

- Uniformity of representation
- Operational uniformity
- Semantic modeling
- Extended application

The article has the aim to demonstrate the benefits of using deductive databases. These benefits are listed below:

- Representation of disjointed information
- Reflection of negative information
- Performing recursive queries

Datalog language

Further bellow there's an example of how Datalog is used to query a knowledge database:

```
Fz_concitadini(fx,fy)<=F(fx,nfx,sfx,of)    AND    F(fy,nfy,sfy,of)    AND
NOT(fx=fy)

Fz_bun(f,sf,of) <= F(f,nf,sf,of) AND sf>50
```

4. Conclusions

Deductive systems prove to be the right solution for many issues in modern society, from teaching students natural language processing, to business fields, such as Facebook that integrated different types of data based on predicative logic, through a friendly interface of expert systems, artificial intelligence applications and E-Commerce applications.

In the near future, arithmetical operations by using aggregate functions, useful in On-Line Analytical Pro-cessing applications involving data mining and data warehouses we'll be done exclusively by evaluating first order logic expressions and lambda calculus.

5. References

[1] C. Nelson Randal (2017) University of Rochester, NY 14627-0226, Online course available at: https://www.cs.rochester.edu/~nelson/courses/csc_173/grammars/parsetrees.html

[2] C. Sung-Pil, M. Sung-Hyon, Terminological paraphrase extraction from scientific literature based on predicate argument tuples, *Journal of Information Science*, **38**(6), 2012, http://journals.sagepub.com/doi/pdf/10.1177/0165551512459920

[3] C.J. Date, *Baze de date*, Plus, 2005.

[4] D, Diaz-Pernil, MA, Gutierrez-Naranjo, Semantics of deductive databases with spiking neural P systems, *Neurocomputing*, 2018, **272**: 365-373, DOI: 10.1016/j.neucom.2017.07.007.

[5] D. Richardson, (2006) *Formal systems, logic and semantics*, On line course of Department of Computer Science, University of Bath, http://www.cs.bath.ac.uk/pb/EMCL/DS/DS-Ref-2011/c19.pdf

[6] D. Seipel, Knowledge Engineering for Hybrid Deductive Databases, *Electronic Proceedings in Theoretical Computer Science*, 2017, **234**: 1-12, DOI: 10.4204/EPTCS.234.1.

[7] F. Nogatz, D. Seipel, Implementing GraphQL as a Query Language for Deductive Databases in SWI-Prolog Using DCGs, Quasi Quotations, and Dicts, *Electronic Proceedings in Theoretical Computer Science*, 2017, **234**: 42-56, DOI: 10.4204/EPTCS.234.4.

[8] F. Sáenz-Pérez, Relational calculi in a deductive system, *Expert Systems With Applications,* 2018, **97**: 106–116, DOI: https://doi.org/10.1016/j.eswa.2017.12.007.

[9] G. W. Leibnitz, Elementele caracteristicii universale, 1679, Limba universală, caracteristică universală, calcul logic, *Univers Enciclopedic*, 2015.

[10] M. Maravanyika, N. Dlodlo, N. Jere, An Adaptive Recommender-System Based Framework for Personalised Teaching and Learning on E-Learning Platforms, Book Group Author(s):*IEEE,* 2017 IST-AFRICA WEEK CONFERENCE (IST-AFRICA)

[11] Rocsana Bucea-Manea-Ţoniş, M. Andronie, M. Iatagan, E-LEARNING IN THE ERA OF VIRTUAL REALITY, The 14th International Scientific Conference eLearning and Software for Education, Bucharest, April 19-20, 2018,10.12753/2066-026X-18-000

[12] S. Bird, E.Klein, E. Loper, E. *Natural Language Processing with Python*. O'Reilly, 2009

[13] S. Brass, H. Stephan, Experiences with Some Benchmarks for Deductive Databases and Implementations of Bottom-Up Evaluation*, Electronic Proceedings in Theoretical Computer Science*, 2017, **234**: 57-72, DOI: 10.4204/EPTCS.234.5.