

Chapter LXXXIV

On the Implementation of a Logic Language for NP Search and Optimization Problems

Sergio Greco

University of Calabria, Italy

Cristian Molinaro

University of Calabria, Italy

Irina Trubitsyna

University of Calabria, Italy

Ester Zumpano

University of Calabria, Italy

INTRODUCTION

It is well known that NP search and optimization problems can be formulated as DATALOG⁻ (datalog with unstratified negation; Abiteboul, Hull, & Vianu, 1994) queries under nondeterministic stable-model semantics so that each stable model corresponds to a possible solution (Gelfond & Lifschitz, 1988; Greco & Sacca, 1997; Kolaitis & Thakur, 1994). Although the use of (declarative) logic languages facilitates the process of writing complex applications, the use of unstratified negation allows programs to be written that in some cases are neither intuitive

nor efficiently valuable. This article presents the logic language NP Datalog, a restricted version of DATALOG⁻ that admits only controlled forms of negation, such as stratified negation, exclusive disjunction, and constraints. NP Datalog has the same expressive power as DATALOG⁻, enables a simpler and intuitive formulation for search and optimization problems, and can be easily translated into other formalisms. The example below shows how the vertex cover problem can be expressed in NP Datalog.

Example 1. Given an undirected graph $G = \langle N, E \rangle$, a subset of the vertexes $V \subseteq N$ is a vertex

cover of G if every edge of G has at least one end in V . The problem can be formulated in terms of the NP Datalog query $\langle P_1, v(X) \rangle$ with P_1 defined as follows:

$$\begin{aligned} v(X) &\subseteq \text{node}(X) \\ \text{edge}(X, Y) &\Rightarrow v(X) \vee v(Y), \end{aligned}$$

where the output relation v gives a vertex cover, $\subseteq$ denotes the subset relation, and $\Rightarrow$ is used to define constraint. The first rule guesses a subset v of the relation *node* whereas the latter constraint verifies the vertex cover condition: It states that if X and Y are two connected nodes, then X or Y or both must be in the cover. The optimization problem computing a cover with minimum cardinality can be simply expressed by means of the query $\langle P_1, \min|v(X)| \rangle$.

The evaluation of logic programs with stable-model semantics can be carried out by means of current ASP (answer set programming) systems that compute the semantics of DATALOG⁻ programs. An alternative solution consists of reducing problems expressed by means of logic formalisms (usually extensions of datalog) into equivalent SAT problems and evaluating the target problems by means of SAT solvers. In this article, a different solution, based on the rewriting of logic programs into constraint programming, is proposed. More specifically, the implementation of NP Datalog consists of translating NP Datalog queries into OPL (optimization programming language; Van Hentenryck, 1999; Van Hentenryck, Michel, Peron, & Regin, 1999), a constraint language well suited for solving both search and optimization problems, and then executing the target OPL code by means of the ILOG OPL Studio platform.

Example 2. The optimization query of Example 1 is translated into the following OPL code.

```
var boolean v[node];
minimize sum (x in node) v[x]
subject to {
    forall (<x,y> in edge)
        1 => (v[x] + v[y] > 0);
}
```

where the second statement expresses the optimization condition (minimizes the cardinality of v), while the last one corresponds to the vertex cover condition.

NP DATALOG

Familiarity is assumed with disjunctive logic programs, disjunctive deductive databases, (disjunctive) stable-model semantics, and computational complexity (Abiteboul et al., 1994; Gelfond & Lifschitz, 1991; Johnson, 1990). In this section, the language NP Datalog is presented. This language restricts the use of unstratified negation of DATALOG⁻ without loss of its expressive power and can be executed more efficiently or easily translated into other formalisms. NP Datalog extends DATALOG^{-s} (datalog with stratified negation) with two simple forms of unstratified negation embedded into built-in constructs: exclusive disjunction, used in partition rules, and constraint rules.

An NP Datalog partition rule is a disjunctive rule of the form

$$p_1(\mathbf{X}) \oplus \dots \oplus p_k(\mathbf{X}) \leftarrow \text{Body}(\mathbf{X}, \mathbf{Y}) \quad (1)$$

or of the form

$$p_0(\mathbf{X}, c_1) \oplus \dots \oplus p_0(\mathbf{X}, c_k) \leftarrow \text{Body}(\mathbf{X}, \mathbf{Y}), \quad (2)$$

where (a) $p_0, p_1, \dots, p_k$ are distinct IDB predicates not defined elsewhere in the program, (b) $c_1, \dots, c_k$ are distinct constants, (c) $\text{Body}(\mathbf{X}, \mathbf{Y})$ is a conjunction of literals not depending on predicates defined by disjunctive rules, and (d) $\mathbf{X}$ and $\mathbf{Y}$ are vectors of variables. The intuitive meaning of these rules is that if the body is true, then exactly one head atom must be true too, hence the projection of the body relation on $\mathbf{X}$ is partitioned nondeterministically into k relations or k distinct sets of the same relation. Clearly, every rule of the form of Equation 2 can be rewritten into a rule of the form of Equation 1 and vice versa.

5 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/implementation-logic-language-search-optimization/20765

Related Content

Improving Business Intelligence Traceability and Accountability: An Integrated Framework of BI Product and Metacontent Map

Chin-Hoong Chee, William Yeoh, Shijia Gao and Gregory Richards (2014). *Journal of Database Management* (pp. 28-47).

www.irma-international.org/article/improving-business-intelligence-traceability-and-accountability/118087

Databases Modeling of Engineering Information

Z.M. Ma (2009). *Selected Readings on Database Technologies and Applications* (pp. 65-84).

www.irma-international.org/chapter/databases-modeling-engineering-information/28546

Method Chunks to Federate Development Processes

Isabelle Mirbel (2007). *Research Issues in Systems Analysis and Design, Databases and Software Development* (pp. 146-184).

www.irma-international.org/chapter/method-chunks-federate-development-processes/28436

On the Efficiency of Querying and Storing RDF Documents

Maria-Esther Vidal, Amadís Martínez, Edna Ruckhaus, Tomas Lampo and Javier Sierra (2012). *Graph Data Management: Techniques and Applications* (pp. 354-385).

www.irma-international.org/chapter/efficiency-querying-storing-rdf-documents/58619

A Scalable Middleware for Web Databases

Athman Bouguettaya, Zaki Malik, Abdelmounaam Rezgui and Lori Korff (2006). *Journal of Database Management* (pp. 20-46).

www.irma-international.org/article/scalable-middleware-web-databases/3361