DLV: An Advanced System for Knowledge Representation and Reasoning

Nicola Leone[1], Wolfgang Faber[1], Gerald Pfeifer[2],
Annamaria Bria[1], Francesco Calimeri[1], Gelsomina Catalano[1], Susanna Cozza[1],
Tina Dell'Armi[1], Thomas Eiter[2], Georg Gottlob[3], Gianluigi Greco[1],
Giovambattista Ianni[1], Giuseppe Ielpa[1], Marco Maratea[1], Claudio Panetta[1],
Simona Perri[1], Francesco Ricca[1], Francesco Scarcello[4], and Giorgio Terracina[1]

[1] Department of Mathematics, University of Calabria, 87030 Rende (CS), Italy
{leone,faber,a.bria,calimeri,catalano,cozza,dellarmi,ggreco,ianni,ielpa,maratea,
panetta,perri,ricca,terracina}@mat.unical.it
[2] Institut für Informationssysteme, TU Wien, A-1040 Wien, Austria
pfeifer@dbai.tuwien.ac.at, eiter@kr.tuwien.ac.at
[3] Oxford University, Oxford, United Kingdom
Georg.Gottlob@comlab.ox.ac.uk
[4] DEIS, University of Calabria, 87030 Rende (CS), Italy
scarcello@deis.unical.it

DLV [5] is a KRR system which is based on Disjunctive Logic Programming (DLP) [6] under the stable model semantics (also called Answer Set Programming) [4]. Roughly, a DLP program is a set of disjunctive rules, i.e., clauses of the form

$$a_1 \texttt{v} \cdots \texttt{v} a_n \texttt{:-}\ b_1, \cdots, b_k, \texttt{not}\ b_{k+1}, \cdots, \texttt{not}\ b_m$$

where atoms $a_1, \ldots, a_n, b_1, \ldots, b_m$ may contain variables. The intuitive reading of such a rule is "If all $b_1, \ldots, b_k$ are true and none of $b_{k+1}, \ldots, b_m$ is true, then at least one atom in $a_1, \ldots, a_n$ must be true." DLP has a very high expressive power – it allows to express all problems in the complexity class Σ_2^P (i.e., NP^{NP}) [3]. Thus, under usual complexity conjuctures, DLP is strictly more expressive than both SAT and CSP, the power of which is "limited" to NP, and it can naturally represent a large class of relevant problems ranging from artificial intelligence to advanced database applications.

DLV is generally considered the state-of-the-art implementation of disjunctive logic programming. Its efficiency has been confirmed by the results of First Answer Set Programming System Competition (http://asparagus.cs.uni-potsdam.de/contest/), where DLV won the DLP category. Moreover, DLV turned out to be very efficient also on non-disjunctive logic programs, as it finished first also in the general category MGS (Modeling, Grounding, Solving – also called *royal* competition, open to all ASP systems).

The implementation of the DLV system is based on very solid theoretical foundations, and exploits major results that have been achieved in the area of logic programming and nonmonotonic reasoning in the last 15 years, ranging from database optimization techniques, to heuristics and new AI methods. The system has been recently engineered for industrial exploitation, and is successfully employed in many challenging real-world applications, for instance in the area of Knowledge Management, and advanced Information Integration.

Among the many features of the system, it is worth remarking the following:

Advanced knowledge modeling capabilities. DLV provides support for declarative problem solving in several respects:
 - High expressiveness in a formally precise sense (Σ_2^P), so any such problem can be uniformly solved by a fixed program over varying input.
 - Rich language for knowledge modeling, extending DLP with weak constraints (for preferences handling) [1], powerful aggregate functions [2], and other useful KR constructs.
 - Full declarativeness: ordering of rules and subgoal is immaterial, the computation is sound and complete, and its termination is always guaranteed.
 - Declarative problem solving following a “Guess&Check” paradigm [5] where a solution to a problem is guessed by one part of a program and then verified through another part of the program.
 - A number of front-ends for dealing with specific AI applications.

Solid Implementation. Much effort has been spent on sophisticated algorithms and techniques for improving the performance, including
 - database optimization techniques (join ordering methods, indexing, etc.), and
 - artificial intelligence computation techniques (heuristics, backjumping, etc.).

DLV is able to solve complex problems and can efficiently deal also with large input data.

Database Interfaces. The DLV system provides a general ODBC interface to relational database management systems.

For up-to-date information on the system and a full manual we refer to `http://www.dlvsystem.com`, where also download binaries of the current release and various examples are available.

References

1. Francesco Buccafurri, Nicola Leone, and Pasquale Rullo. Enhancing Disjunctive Datalog by Constraints. *IEEE Transactions on Knowledge and Data Engineering*, 12(5), September/October 2000.
2. Tina Dell'Armi, Wolfgang Faber, Giuseppe Ielpa, Nicola Leone, and Gerald Pfeifer. Aggregate Functions in Disjunctive Logic Programming: Semantics, Complexity, and Implementation in DLV. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI) 2003*, pages 847–852, Acapulco, Mexico, August 2003. Morgan Kaufmann Publishers.
3. Thomas Eiter, Georg Gottlob, and Heikki Mannila. Disjunctive Datalog. *ACM Transactions on Database Systems*, 22(3):364–418, September 1997.

4. Michael Gelfond and Vladimir Lifschitz. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing*, 9:365–385, 1991.
5. Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Thomas Eiter, Georg Gottlob, Simona Perri, and Francesco Scarcello. The DLV System for Knowledge Representation and Reasoning. *ACM Transactions on Computational Logic*, 7(3):499–562, July 2006.
6. Jack Minker. On Indefinite Data Bases and the Closed World Assumption. In D.W. Loveland, editor, *Proceedings* 6^{th} *Conference on Automated Deduction (CADE '82)*, volume 138 of *Lecture Notes in Computer Science*, pages 292–308, New York, 1982. Springer.