Declarative Diagnosis of Wrong Answers in Constraint Functional-Logic Programming

Rafael Caballero, Mario Rodríguez Artalejo, and Rafael del Vado Vírseda[⋆]

Dep. Sistemas Informáticos y Programación, Univ. Complutense de Madrid
{rafa, mario, rdelvado}@sip.ucm.es

Debugging tools are a practical need for diagnosing the causes of erroneous computations. Declarative programming paradigms involving complex operational details, such as constraint solving and lazy evaluation, do not fit well to traditional debugging techniques relying on the inspection of low-level computation traces. As a solution to this problem, *declarative diagnosis* uses *Computation Trees* (shortly, *CTs*) in place of traces. *CTs* are built *a posteriori* to represent the structure of a computation whose top level outcome is regarded as an *error symptom* by the user. Each node in a *CT* represents the computation of some observable result, depending on the results of its children nodes. Declarative diagnosis explores a *CT* looking for a so-called *buggy node* which computes an incorrect result from children whose results are correct; such a node must point to an incorrect program fragment. The search for a buggy node can be implemented with the help of an external *oracle* (usually the user with some semi-automatic support) who has a reliable declarative knowledge of the expected program semantics, the so-called *intended interpretation*.

The generic description of declarative diagnosis in the previous paragraph follows [8]. Declarative diagnosis was first proposed in the field of logic programming [10], and it has been successfully extended to other declarative programming paradigms, including lazy functional programming [9], constraint logic programming [11,4] and functional logic programming [2,3]. In contrast to recent approaches to error diagnosis using *abstract interpretation* [5], declarative diagnosis often involves complex queries to the user. This problem has been tackled by means of various techniques, such as user-given partial specifications of the program's semantics [3], safe inference of information from answers previously given by the user [2], or *CTs* tailored to the needs of a particular debugging problem over a particular computation domain [4]. Current research in declarative diagnosis has still to face many challenges regarding both the foundations and the development of practical tools.

The aim of this work is to present a declarative method for diagnosing wrong computed answers in $CFLP(\mathcal{D})$, a newly proposed generic programming scheme which can be instantiated by any constraint domain $\mathcal{D}$ given as parameter, and supports a powerful combination of functional and constraint logic programming over $\mathcal{D}$ [6]. Borrowing ideas from $CFLP(\mathcal{D})$ declarative semantics we obtain a

[⋆] The authors have been partially supported by the Spanish National Projects MELODIAS (TIC2002-01167), MERIT-FORMS (TIN2005-09207-C03-03) and PROMESAS-CAM (S-0505/TIC/0407).

suitable notion of intended interpretation, as well as a convenient definition of proof tree with a sound logical meaning to play the role of CTs. Our aim is to achieve a natural combination of previous approaches that were separately developed for the $CLP(\mathcal{D})$ scheme [11] and for lazy functional logic languages [2]. We have proved theoretical results showing that the proposed debugging method is logically correct for any sound $CFLP(\mathcal{D})$-system [12] whose computed answers are logical consequences of the program in the sense of $CFLP(\mathcal{D})$ semantics. We have implemented a debugging tool called $\mathcal{DDT}$, developed as an extension of previously existing but less powerful tools [1,3] and available at `http://toy.sourceforge.net`. $\mathcal{DDT}$ implements the proposed diagnosis method for $CFLP(\mathcal{R})$-programming in the $\mathcal{TOY}$ system [7] using the domain $\mathcal{R}$ of arithmetic constraints over the real numbers. Moreover, $\mathcal{DDT}$ provides some facilities for navigating proof trees and avoiding redundant queries to the user. As future work, we plan to develop a formal framework for the declarative diagnosis of *missing answers* in $CFLP(\mathcal{D})$ and we plan several improvements of $\mathcal{DDT}$, such as enabling the diagnosis of missing answers, supporting finite domain constraints, and providing new facilities for simplifying the presentation of queries to the user.

References

1. R. Caballero. *A Declarative Debugger of Incorrect Answers for Constraint Functional-Logic Programs.* Proc. WCFLP'05, ACM SIGPLAN, pp. 8–13, 2005.
2. R. Caballero and M. Rodríguez-Artalejo. *A Declarative Debugging System for Lazy Functional Logic Programs.* ENTCS 64, 63 pages, 2002.
3. R. Caballero and M. Rodríguez-Artalejo. *$\mathcal{DDT}$: A Declarative Debugging Tool for Functional Logic Languages.* Proc. FLOPS'04, Springer LNCS 2998, pp. 70–84, 2004.
4. G. Ferrand, W. Lesaint and A. Tessier. *Towards declarative diagnosis of constraint programs over finite domains.* ArXiv Computer Science e-prints, 2003.
5. M. Hermenegildo, G. Puebla, F. Bueno and P. López-García. *Abstract Verification and Debugging of Constraint Logic Programs.* Proc. CSCLP'02, pp. 1–14, 2002.
6. F.J. López-Fraguas, M. Rodríguez-Artalejo and R. del Vado-Vírseda. *A New Generic Scheme for Functional Logic Programming with Constraints.* To appear in the Journal Higher-Order and Symbolic Computation, 2006. (Extended version of *Constraint Functional Logic Programming Revisited*, WRLA'04, ENTCS 117, pp. 5–50, 2005.)
7. F.J. López-Fraguas, J. Sánchez-Hernández. *$\mathcal{TOY}$: A Multiparadigm Declarative System.* In Proc. RTA'99, Springer LNCS 1631, pp 244–247, 1999. System and documentation available at `http://toy.sourceforge.net`.
8. L. Naish. *A Declarative Debugging Scheme.* Journal of Functional and Logic Programming, 1997-3.
9. B. Pope and L. Naish. *Practical aspects of declarative debugging in Haskell 98.* Proc. PPDP'03, ACM Press, pp. 230–240, 2003.
10. E.Y. Shapiro. *Algorithmic Program Debugging.* The MIT Press, Cambridge, 1982.
11. A. Tessier and G. Ferrand. *Declarative Diagnosis in the CLP Scheme.* Springer LNCS 1870, Chapter 5, pp. 151–174, 2000.
12. R. del Vado-Vírseda. *Declarative Constraint Programming with Definitional Trees.* In Proc. FroCoS'05, Springer LNAI 3717 pp. 184–199, 2005.