Guest Editors' Preface — Foundations of constraint programming

Constraint programming is an alternative approach to computing in which the programming process is limited to a generation of requirements (constraints) and to solving of them by general and domain dependent methods. It has been successfully used in many areas of computer science including optimization, numerical computing, natural language processing, computer algebra and computer graphics, to mention a few.

This special issue of Fundamenta Informaticae is devoted to foundations of constraint programming. The papers here presented show that this style of programming can be seen as orthogonal to the usual programming paradigms: imperative, logic and functional ones.

To start with, the paper of Robert Kowalski, Francesca Toni and Gerhard Wetzel shows how such diverse developments as constraint logic programming, abductive logic programming and semantic query optimization can be realized within a single unified framework that extends logic programming. This calls for an extension of the usual resolution mechanism of logic programming by propagation and equality preserving transformations.

In turn, the paper of Claude Kirchner and Christophe Ringeissen shows how constraint programming can be realized using the term rewriting formalism augmented with strategies. This brings constraint programming closer to functional programming. In fact, the authors show how this approach to constraint programming can be realized in the functional programming language ELAN.

Then, Carlos Castro explains in his paper how this approach to constraint programming can be used to explain general techniques of constraint programming such as local consistency algorithms and search techniques. These ideas were realized in a system called COLETTE that was implemented in the ELAN language.

Next, the paper of Krzysztof Apt presents a related, but more abstract, proof theoretic framework to explain the basic principles of constraint programming. This leads to a view of constraint programming that can be realized independently of the assumed style of programming.

Finally, the paper of Jean-Hugues Réty deals with the problem of using constraints as a basis for distributed programming. The author shows how the framework of concurrent constraints can be extended with facilities that support distributed programming and illustrates the usefulness of this approach by a number of examples drawn from the literature on distributed programming.

We would like to take this opportunity and thank all the authors of the submitted papers for having agreed to contribute to this special issue and the referees for taking their time and providing helpful reviews of the papers.

Krzysztof R. Apt
CWI
Kruislaan 413
1098 SJ Amsterdam
The Netherlands

Claude Kirchner
LORIA - INRIA
615 rue du Jardin Botanique
BP 101
54602 Villers-lès-Nancy
France