Chapter 3
Term Rewriting-Based Programming

ABSTRACT

One of the directions of algebraic programming is the use of rewriting rules technique. This direction formalizes the transformational aspects of programming, which allows us to describe transformations of some formal objects and research properties of such transformations. The rewriting rules technique is both a powerful formal tool for transformation of formal systems and a practical tool for programming that allows implementing transformations of complex objects. In this chapter, the main definitions associated with algebraic programming based on rewriting rules are given, term rewriting systems are overviewed, and applications of these systems for processing and transformation of programs are considered.

INTRODUCTION

At present, metaprogramming problems (i.e. the development of programs, the input and/or output data of which are also programs) become increasingly relevant. Some metaprogramming tools, such as compilers and programming language editors, are necessary for programming arbitrary tasks, as without them the development becomes too difficult or impossible. The other tools are not so irreplaceable, but still form an important part of modern software lifecycle. Static analysis allows improving the quality of source code, revealing potential errors or programming style violations. Automatic transformations

DOI: 10.4018/978-1-5225-9384-3.ch003

of programs contribute to increasing their readability, performance, security, reduce development time and allow avoiding mechanical mistakes. Specialized languages reduce program size, allowing to concisely and clearly express subject domain concepts. Code generation provides the opportunity to use a higher level of abstraction and automatically obtain relevant executable code. Thus, metaprogramming plays a crucial role in development of modern software.

For solving metaprogramming problems, general-purpose programming languages such as C/C++, Java, C# can be used. However, these problems have some features which are not taken into account in general-purpose languages. Metaprograms do not require means for defining complex data structures: their working object is source code in some standard structured representation, most often it is a parse tree. Most metaprogramming problems can be presented as transformations of this tree. Traditional imperative languages contain general-purpose control structures (branching, loops) that form the basis of any imperative program. However, in metaprograms there is no need for complex control: neither in data access area (usually the whole tree is traversed in some fixed order), nor at applying transformations (which are applied while conditions of their application are present). On the other hand, metaprogramming requires operations that are not so easily implementable in imperative languages. For example, searching for a part of a tree corresponding to a given template or application of a transformation to all parts where it is applicable. Another significant requirement is a possibility to easily add or modify transformations; in traditional languages, it will require at least recompilation and maybe modification of other transformations for their correct cooperation.

Thus, application of general-purpose languages for solving the problems of processing and transforming programs is not justified. However, there are specialized programming tools that are well suited exactly for such problems. These are rewriting rules systems working with terms (Baader & Nipkow, 1999; Dershowitz & Plaisted, 2001). The term is a natural representation of a tree and rewriting rules include a search for a template and replacement of a subterm, which allows modeling any transformations. Rewriting rules system controls the process of transformations application (i.e. rewriting), so the developer has only to specify necessary transformations. The addition of new transformations consists in implementation of new rules which automatically correctly cooperate with already existing ones. Therefore, rewriting rules