# Chapter 5
# Software Tools for Automated Program Design, Synthesis, and Auto-Tuning


## ABSTRACT

*The authors consider the software tools based on algebra-algorithmic models and formal methods of constructing algorithms and programs. The algebra-algorithmic integrated toolkit for design and synthesis of programs IDS, the rewriting rules system TermWare, and the auto-tuning framework TuningGenie are presented. IDS uses algebraic specifications based on Glushkov's algebra of algorithms, which are represented in three forms: algebraic (regular scheme), natural linguistic, and graphical (flowgraphs). IDS is based on the method of dialogue design of syntactically correct algorithm schemes, which eliminates syntax errors during construction of algorithm specifications. To automate transformations of algorithms and programs being designed, the rewriting rules system TermWare is used. TuningGenie framework is applied to automate the adjustment of programs to a target computing environment.*


## INTRODUCTION

The algebra-algorithmic approach (Andon, Doroshenko, Tseytlin, & Yatsenko, 2007; Doroshenko, Tseytlin, Yatsenko, & Zachariya, 2006) is supported by a number of tools developed within the framework of Kyiv algebraic-cybernetic school. The first such tool was the system called MULTIPROCESSIST

DOI: 10.4018/978-1-5225-9384-3.ch005

(Yushchenko, Tseytlin, Hrytsay, & Terzyan, 1989), developed in the 1980s in the programming automation department of V. M. Glushkov Institute of Cybernetics. The system provided the multilevel design of algorithms and data structures represented in the form of schemes in Glushkov's system of algorithmic algebra (SAA) and synthesis of corresponding programs. The algebraic approach was also used at development of the system for transformation of algorithm schemes (Petrushenko, 1991). The method of controlling spaces based on algebraic approach was used in the parallel programming technology PARUS (Anisimov & Kulyabko, 1984).

In this chapter, the developed tools for programming automation are considered, which continue the aforementioned research, namely, the Integrated toolkit for Design and Synthesis of programs (IDS) (Andon et al., 2007; Doroshenko, Ivanenko, Ovdii, & Yatsenko, 2016; Doroshenko, Zhereb, & Yatsenko, 2013; Yatsenko, 2012), the term rewriting system TermWare (Doroshenko & Shevchenko, 2006; "TermWare", n.d.) (see also Chapter 3) and the auto-tuning framework TuningGenie (Ivanenko, Doroshenko, Zhereb, 2014).

The approach being considered is related to works on automated synthesis of program code from specifications (Flener, 2002; Gulwani, 2010). The important aspects of program synthesis include:

- format of inputs (specifications) — how specifications or other inputs for synthesis are provided;
- methods for supporting particular subject domains — how the proposed approach is specialized for a given domain, so that domain knowledge can be used to produce more efficient programs and/or to improve developer's productivity;
- techniques for implementing transformation from specifications to output program code.

These aspects roughly correspond to three dimensions of program synthesis discussed in (Gulwani, 2010):

- *user intent* describes specifications;
- *search space* restricts all possible programs to a manageable subset;
- *search technique* describes how transformations are implemented.

For input specification, a popular option is using domain-specific languages that allow capturing requirements of a subject domain. In (Bagheri & Sullivan,