Joint NCC & IIS Bull., Comp. Science, 12 (1999), 24–28

Expert system for simulation of program optimization

M.A. Knyazeva, O.A. Kupnevich

This paper presents a brief description of an expert system I_MESOP, a tool for investigation of already known systems of optimizing transformations and for development of the new ones.

1. Introduction

Research in program optimization is an important and promising direction of computer science. Though the memory size and processor's speed of computers are constantly growing, the increasing complexity of computer-solved problems makes automated construction of high-quality software still necessary.

Program optimization is widely used in translation from high-level programming languages. To optimize a program is to apply some special transformations to the program in the source, or intermediate, or target programming language in order to reduce time and/or memory needed to execute it.

By now, some significant results have been achieved in the theory and practice of program optimization, such as the framework of program schemes which provides its theoretical background, a large set of optimizing transformations and sufficient conditions of their applicability, etc [1, 2]. Almost every compiler provides now a large set of optimizing facilities.

Obviously, program optimization is important for both the theory and practice of computer science. However, implementation of even the simplest optimizing transformations requires a lot of efforts from the students, so only the theory of program optimization is tought.

There are no necessary tools for analysis and design of new algorithms of program transformations, which complicates the process of teaching.

As a rule, students can not obtain practical skills in this problem. So, post-graduates need a lot of practice to become experts in program optimization.

One way of changing this situation is the development of tools for program optimization which allow a student to define a set of optimizing transformations and explore it.

The purpose of this paper is to describe I_MESOP system intended to formalize information about the systems of optimizing transformations and to conduct experiments with them. The project is based on both theoretical results in the field of program optimization and the experience gained from using other systems working with program semantics [2, 3, 4, 5].

2. The systems of optimizing transformations

The practice of optimizing compilation has proved that a program in a high-level programming language can be translated into a high-quality output program by applying special transformations named *optimizing transformations* [1].

Each optimizing transformation (OT) is focused on enhancing certain characteristics (quality) of a program while preserving its semantics. The application of a combination of several OTs, though does not necessarily result in the best program equivalent to the initial one, but allows us to get rid of the so-called basic inefficiencies inherent to programs directly compiled from high-level languages.

For each transformation included into a compiler, there should be stated conditions of its applicability (its *context conditions*) and a relation which links the source and the transformed program.

A transformation and its context conditions concern some parts of the program, the so-called *regions of economy*. If the region of economy is the entire program, then we speak of global optimization.

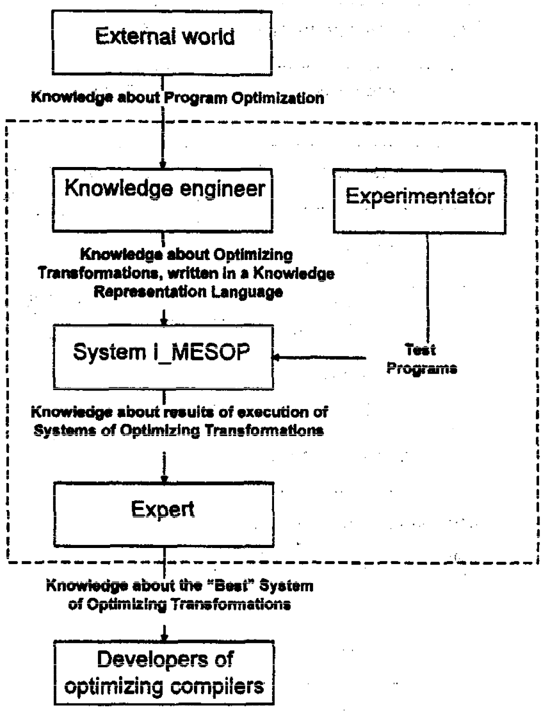

Figure 1. Architecture-context diagram of I_MESOP system

A local optimization is such an optimization whose region of economy is limited to a linear band [1].

Another important and difficult problem is to choose a strategy of applying the transformations, i.e., a sequence of their application.

A set of transformations and a strategy together are called a *system of optimizing transformations*. A "good" system of optimizing transformations should be correct, aimed, powerful and simple [1, 12].

As a rule, the systems of optimizing transformations are constructed of quite elementary transformations called "base" transformations.

Both the theory and practice of optimizing compilation [1] show that powerful "base" transformations do not necessarily constitute a "good" system of OTs.

3. System I_MESOP

At present we develop a system I_MESOP which is focused on the following:

1. Research in efficiency of various systems of transformations (including its dependence on the sets of transformations, the choice of context conditions and the strategy of application),
2. A tool support for training students in the methods of program optimization.

In what follows, by an optimizing transformation we mean a transformation of a program in a high-level programming language.

A scheme in Fig. 1 illustrates the structure of the system.

Being guided by the received knowledge about OTs and the strategy of their application, this system should be able to output a text in Russian which explains in detail the process and the results of optimization of a test program. Since this task is related to sophisticated knowledge processing and generation, this system is designed as an expert system which simulates the process of program optimization [5, 6, 7, 8].

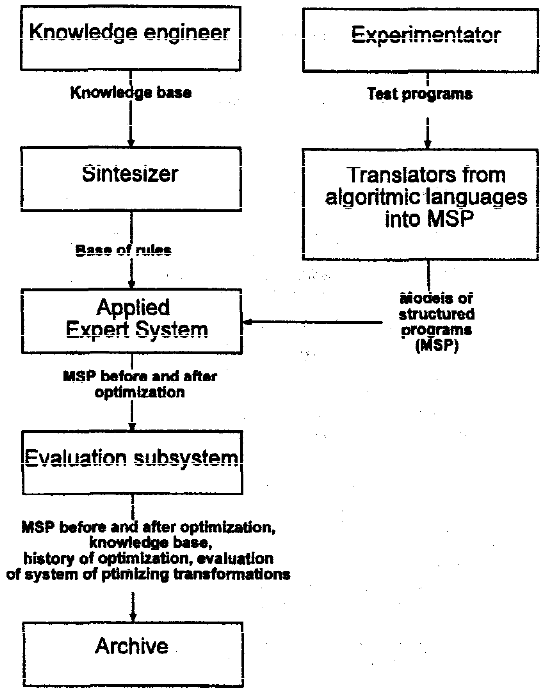

Figure 2. The architecture of I_MESOP

The expert system I_MESOP includes a knowledge base about program optimization, a translator from the Knowledge Representation Language (synthesizer) and a set of compilers from High-Level Programming Languages (HLPL) (such as C, PASCAL) to a universal intermediate representation of programs — a Model of Structured Program (MSP) [8].

Figure 2 shows the architecture of the system I_MESOP.

Knowledge about OTs and a strategy of their application is written in a special logical Knowledge Representation Language (KRL). On the basis of this data, a special subsystem "Synthesizer" generates an applied expert system.

This expert system works under the control of an expert system generator "REPRO" [11]. At the first step, this applied expert system calls one of the translators from HLPL to transform the source program into the MSP. For this MSP, the applied expert system simulates the process of program optimization and constructs the history of optimization. During this process, the applied expert system generates explanations in a natural language to describe in detail the whole process of optimization.

Later, the "Evaluation Subsystem" measures various metrics of the source program and its resulting MSP, and then evaluates the quality of the system of optimizing transformations [9]. After that, an expert can use this knowledge to give some recommendations to compiler designers.

The expert system I_MESOP is developed within the framework of a problem-oriented knowledge representation. This approach requires an expert system to consist of (1) a knowledge base scheme, (2) a scheme of the expert system database, and (3) rules of conformity between the bases of rules and knowledge (defaults) [10, 11].

The knowledge base is a semantic network which represents relations between terms employed in the program optimization.

The database contains the history of application of optimizing transformations to the source MSP. The database scheme describes the structure of an entry in the history of optimization [10, 11].

The rules of conformity between the knowledge base and the database are explicitly stated by relations defined in the theory of program optimization [10, 11].

The previous implementation of I_MESOP deals with a number of such well-known OTs as moving statements out of recursive functions, elimination of common subexpressions, moving statements out of loops. However, the possibilities of I_MESOP were restricted by expressive capabilities of MSP, which could not adequately represent powerful constructions of modern programming languages. In particular, all variables in MSP were of the same type (integer), which does not allow for optimization of indices.

By now we have developed a new version of MSP which supports all scalar data types and arrays of arbitrary dimensions. In addition, it includes pointers and address arithmetics. This allows us to study a considerably larger class of OTs and makes context conditions more adequate to real applications in translators [8].

Now the Synthesizer for the new MSP is being developed and MSP itself is being extended to deal with parallel programs.

The system described above is employed in the lectures on knowledge bases and expert systems and the methods of translation and program optimization.

4. Related works

There is a large number of systems which can be classified as systems for program manipulation: packages of tools for program optimization, intellectual program editors, and various program environments. Commonly, they are intended to facilitate the process of development of effective and reliable programs at different stages of their life cycle.

Such a package usually consists of a family of small utilities ensuring its easy usage by the beginners, which results in an increase in efficiency of program porting. Few examples of such packages are Faust, Start/Pat, SigMacs, Rn, PTOPP [12], etc. The most close to I_MESOP is the PROGRESS system developed in the Novosibirsk State University [12].

References

[1] V.N. Kasyanov, *Program optimizing transformations*, Moscow: Nauka, 1988, 336 p. (in Russian).

[2] B.V. Arkhangelski, A.I. Nikitin, *The systems of program optimization*, Kiev: Tekhnika, 1983, 167 p. (in Russian).

[3] I.V. Pottosin, *Global optimization: a practical approach*, Proc. of Symposium on methods of implementation of new languages. Part 1, Computing Center, Novosibirsk, 1975, 113–142 (in Russian).

[4] I.V. Pottosin, *Substantiating the algorithms of optimizing programs*, Programmirovanie, 1979, No 2, 3–13 (in Russian).

[5] M.A. Knyazeva, *An expert system modeling the process of program optimization*, Thesis abstract, Institute of Automation and Control Processes, Vladivostok, 1988, 30 p. (in Russian).

[6] M.A. Knyazeva, A.S. Kleschev, *Knowledge presentation in an expert system modeling the process of program optimization*, Preprint, Institute of Automation and Control Processes, Vladivostok, 1988, 25 p. (in Russian).

[7] M.A. Knyazeva, A.S. Kleschev, *Knowledge processing in an expert system modeling the process of program optimization*, Preprint, Institute of Automation and Control Processes, Vladivostok, 1990, 44 p. (in Russian).

[8] O.A. Kupnevich, M.A. Knyazeva, *An extended model of structured programs in an expert system modeling the process of program optimization*, Preprint, Institute of Automation and Control Processes, Vladivostok, 1997, 25 p. (in Russian).

[9] N. Fenton, *Software metrics: a practical and rigorous approach*, 1996.

[10] A.S. Kleschev, *A relational model of computing*, Programmirovanie, 1980, No 4, 20–29 (in Russian).

[11] A.S. Kleschev, *An expert system based on metaknowledge*, Preprint, Institute of Automation and Control Processes, Vladivostok, 1994, 13 p. (in Russian).

[12] V.A. Evstigneev, V.N. Kasyanov, *Instrumental system for investigation of programs transformations*, Software intellectualization and quality, Novosibirsk, 1994, 90–99 (in Russian).