

AN ENVIRONMENT FOR TEACHING OF ALGORITHMS

Valentina DAGIENĖ and Gintautas GRIGAS

Institute of Mathematics and Informatics,
Lithuanian Academy of Sciences
2600 Vilnius, Akademijos St.4, Lithuania

Abstract. The role of an interface between an algorithm and its environment for school students is discussed. The motivation of the use of Pascal functions and procedures instead of commonly used programs in the first lessons of programming is presented. The main features of an interpreter of algorithms (Pascal functions and procedures) implemented on school computer are presented.

Key words: algorithm, interpreter, function, procedure, teaching of programming.

Introduction. The analysis of an existing algorithm and the development of a new one usually starts with an examination of input and output data. Thus, the interface between an algorithm and its environment is of great importance both for a reader and a writer of an algorithm. The interface must be as simple as possible and clearly defined. This is especially important during the first lessons for the beginners of programming.

The interface between an algorithm and its environment is defined by the programming language in terms of the language constructs used to describe algorithms. There is a number of such constructs namely: program, procedure, function, subroutine, module, unit, etc.

Pascal is widely used programming language for teaching purposes. The first three constructs are in Pascal. They are examined from the viewpoint of teaching in subsequent.

The analysis of Pascal constructs

PROGRAM. The only input or output data type is a file. The components of the file are not directly accessible by operators except *read* and *write* procedures. In order to operate the components of a file local variables and transfer operations of values between those variables and files are required. Read and write operations are usually split among other statements and are not incorporated into integral part of the interface. Thus, the file is too complicated data structure for the beginner. However, program is the only directly executable construct of algorithm in Pascal.

PROCEDURE. The interface may be assured by means of parameters and global objects. Those may be of any data type. It is reasonable to limit the interface only by data parameters in order to develop good programming habits among the beginners. Those are: value parameters and variable parameters. The values may be used as input data, the variables may be used both as input or output data. It is reasonable to limit the variable parameters only for output data.

FUNCTION. A function may be considered as a special case of procedure. It is convenient to use the function in the case the single scalar result is required. The output is carried out to the environment of a function by means of the function identifier. In order to exclude side effects an agreement to use only the value parameters is necessary.

Both procedures and functions are not directly executable by a computer. In order to execute them, they must be included into a program.

We may arrive at the conclusion that procedures and

functions are more suitable entities from the methodical point of view, while programs are from practical considerations.

The procedures (and functions) have been commonly used for descriptions of algorithms since the time of Fortran (as subroutines) and Algol-60. However, we don't know a textbook on programming or on algorithms starting from procedures and functions.

Our aim is to support the methodical side for teaching the algorithms by an appropriate curriculum, by a textbook of programming and by a friendly software enabling the school students to execute the Pascal procedures and functions directly by a computer. Thus, the construct of program and notion of the file becomes unnecessary and may be omitted from the curriculum.

An acquaintance with a computer begins from learning commands of its operating system. It is very desirable that the set of commands would include the commands supposed to be used in algorithms. In the case of a subsequent use of Pascal, as a media for writing algorithms, it is reasonable to include the arithmetic operators (+, -, *, /, div, mod), some standard functions (sqr, sqrt, abs, sin, etc.) and assignment statement of Pascal into this set. The above mentioned items consist of a sufficient of commands to operate a computer as an "Pascal" calculator with variables. After some experience with those commands a school student naturally wishes to widen the set of operations. It is the right time to teach the student to develop the algorithm himself (herself) and to introduce procedures and functions as an instrument to perform the job.

To support this way of teaching the interpreter of algorithms is designed and implemented on computer IBM PC XT currently being delivered for Lithuanian schools. The interpreter conceptually offers two modes of work: an environment and an algorithm. All the set of commands and the values of variables is reachable from the environment. The learner may

define integer, real and boolean variables. execute command (operators), assign values to the variables. The set of commands may be extended by procedures and functions. They may be written by school students themselves using the second mode of the interpreter.

The features common to teaching environment are included in the interpreter. Namely, a syntax-oriented editor, syntax errors checking by request during edition, execution of procedures and functions by steps, exposition of the values of variables both local defined inside an algorithm and global defined in environment and multiple copies of active function and procedure texts in the case of recursive calls, etc.

Slight differences between the procedure and the function may benefit for different ways of teaching. Let us mention two of them.

The first. A procedure is considered as the main entity to describe an algorithm with many arguments and many results in general. A function is only a special case of a procedure with many arguments and only one scalar result.

Teaching of algorithms begins from procedures. The function is introduced later and used appropriately.

The second. A function is considered as the main entity to describe algorithms. Every algorithm is considered as a device to get only one result from (possible) many arguments. It is true for functional programming. An algorithm with many results may be considered as a special case and may be described as a combination of many functional algorithms. Procedures may be introduced later or may be not used at all.

Conclusions. For the time being a text book of informatics for secondary schools (Dagienė and Grigas, 1991) is written according to the first way. The choice is determined by the traditional interpretation of the notion of the algorithm (many inputs – many outputs) inherited from the early text-book. An impact of up-to-date state of the art of program-

ming had an influence as well. However, it is more reasonable to consider an algorithm as a device to achieve only one goal. The goal may be complicated and expressed by means of a structured object in terms of programming. This is in accordance with the ideas of functional programming. Thus, the second way may be more perspective and deserves to be developed.

Acknowledgements. The authors thank to Miss. Rūta Astromskaitė for writing the interpreter of Pascal functions and procedures.

REFERENCES

Dagienė V., and G. Grigas (1991). *Informatika*, Šviesa, Kaunas.

Received April 1991

V. Dagienė graduated Vilnius State University in 1978. She is an author of four books and many papers of programming. Her research interests include programming methodology and teaching.

G. Grigas received the Degree of Candidate of Technical Sciences from the Kaunas Polytechnic Institute (Kaunas, Lithuania) in 1970. He heads the Department of Systems Programming at the Institute of Mathematics and Informatics. His research interests include abstract data types, programming methodology and teaching.