DUALITY OF FUNCTIONS AND DATA
IN ALGORITHMS DESCRIPTION

A.D. Astakhov
Institute for Systems Studies
Academy of Sciences of the USSR
9, Prospect 60 Let Oktyabrya
117312 Moscow, USSR

Representation of algorithms in terms of graphs with two kinds of vertices, namely, vertices-functions and vertices-data is under consideration. Two alternative methods for algorithm description as graphs are analysed, one being based on vertices-functions and another - on vertices-data. A graph model with equivalent role of these ways of descriptions is proposed.

Various computing models used in computation theory are known. These models have different purposes, methods of interpretation etc. However, there are only two essentially different kinds of elements used in every model: functions (operators, subroutines, procedures) and data (variables, records, entries). As a rule, there are two kinds of relations between these elements: branches, i.e. relations determining sequences of implemented functions, and informational relations determining information transform between functions' outputs-inputs (results-arguments).

Usually, an algorithm is constructed on the base of one kind of these relations. Programmer describes either a sequence of functions' implementation and, separately, the data used, or a sequence of branches from one data to another and, separately, functions needed when the branches are processed. The first way is used for computational problems which contain complex computations and relatively simple structures of the data used.

Traditional algorithmic programming languages are oriented on this type of problems. The second way is used for information-logical problems concerned with information store and retrieval, i.e. complex from the point of data structure and relatively simple from the viewpoint of function structure used. Data management tools are oriented on this type of problems. In the first case the structure of functions determines the algorithm, and the graph of bran-
ches is primary to the information one. In the second case the information graph is the primary one.

However, one can construct a model in which functions and data play equal roles. The model is a graph with both kinds of vertices and both kinds of relations. This graph is assumed to allow an asynchronous parallel processing. This means that the graph contains all possible branches (permitted by algorithm) and all possible information relations. One can obtain this graph, for example, by accomplishing the desequency of a graph for sequential algorithm [1] or by constructing a net of achievable elements - functions (transitions) and data (positions) based on Petri nets [2].

When the model is represented one can describe it by any of the ways pointed out, i.e., either defining a sequence of functions with data in additional definition, or defining sequences of branches from one data to another with functions in additional definition. Thus, functions and data play equal roles in this representation and one can choose the most convenient kind of definition depending on the particular problem.

The mentioned feature of functions and data allows to make some conclusions practically useful. In particular, the "duality" validates a general approach to operators and data organization in programs and a possibility to use traditional tools of data organization for operators, and vice versa.

E.g., extraction of subprogram structure definition (description of its identifiers, interface format, usage characteristics etc.) into a separate set usual for data (data base schemes) allows to design programs explicitly and to solve a set of problems concerning multiaspect subprograms usage, support the entity for large programming systems and simplify design work for new programs on the basis of available ones.

The approach described has already been used in the system for automatic maintenance and run of software complexes, in which both information describing programs (functions) together with technologies maintained, i.e., programs sequences and their run conditions, and data needed to generate appropriate job descriptions are constructed in a united functions and data base [3]. The system allows to unite jobs subnets in net structural elements and to manipulate aggregates as original elements. Besides, if any modification, for instance a change of some data set name or jobs sequence, is carried out, this change is automatically reflected in all jobs descriptions using this data set and in all technological nets using this sequence.