the same set of nodes are complementary if two nodes i and j are joined in D by a directed edge from i to j if, and only if, they are not joined in $\bar{D}$ by a directed edge from i to j. A graph or digraph which is isomorphic to its complement is said to be *self-complementary*.

Table A shows the 10 self-complementary graphs on 8 nodes, Table B the 36 self-complementary graphs on 9 nodes, and Table C the 10 self-complementary digraphs on 4 nodes. An accompanying text describes the method used.

AUTHOR'S SUMMARY

12 [12].—BORIS BEIZER, *The Architecture and Engineering of Digital Computer Complexes*, Vols. I and II, Plenum Press, New York, 1971, li + 394 pp. (Vol. I), xix + 847 pp. (Vol. II), 24 cm. Price $22.50 each volume.

This two-volume opus is a comprehensive discussion of the varied and interrelated problems that face the designer of a large computer complex. It is impressive in its scope and in the organization of its material, covering topics ranging from the mathematics of flowchart analysis and a dissection of machine instruction types to personnel management and the difficulties of handling thick cables under a computer room floor.

The first volume is mainly concerned with the components of a computer complex. Chapter 2 provides an excellent discussion of instruction repertoires, including various approaches to addressing, indexing, and instruction modification, and a classification of the types of instructions that appear in computers. Chapter 3 discusses the structural elements of a computer complex: memories, interrupt handling, controllers, and peripheral devices. There are two chapters on programming. Chapter 4 examines the programming process, considers certain selected techniques of general applicability, and shows how tradeoffs can be applied in this area. Chapter 5 is concerned with firmware, that is, the supporting programs that are required in order to make application programs work: assemblers, loaders, compilers, and utilities. Chapter 6 is on analysis, and is followed up in the second volume. It begins with an elementary discussion of statistics, and then shows how statistical techniques can be applied to estimating the behavior of programs, in terms both of time and of space. Transformation of flowcharts are shown to be a useful analytic tool, and the behavior of various statistical measures under these transformations is developed.

The second volume deals with questions of system organization. Chapter 8 considers the partitioning of tasks among hardware and software resources, as well as some of the interconnection problems. This chapter introduces a great deal of terminology, some of which, unfortunately, is rather obscure. Chapter 9 considers the functions and organization of the system executive, while Chapter 10 considers the system nucleus, whose task it is to manage storage and input-output. Chapters 11 and 12 consider the problem of system viability—that is, how to keep the system alive despite hardware and software failures and overloads. Viability has three components: *performance*, the ability of the system to handle its appointed tasks under varying loads; *reliability*, the mean time between failures; and *maintainability*, the mean time to recover from a failure. The viability executive has the task of maintaining system viability under the assumption that any part of the system, including

the viability executive itself, can fail. I found the discussion of this problem and some of the solutions to it to be quite stimulating. Chapter 13 deals with system analysis, and considers questions such as the length of the basic system cycle and the response to various loading conditions. Chapter 14, the final chapter, is concerned with the process by which a system is implemented, from procurement through installation.

I recommend this book both for reference and for teaching. It is well written, though there are some unfortunate lapses of clarity. The author seems to be particularly influenced by the overseas AUTODIN system, a communications-handling system constructed by Philco for the Defense Communications Agency, and some of the material on viability, in particular, is directed towards systems of that genre. There are problems at the end of each chapter. Some of these are quite thought-provoking, though a few seemed ill-defined and insufficiently related to the preceding material. There are quite a few amusing anecdotal footnotes, which I found to be one of the most charming aspects of the book.

In sum, this book is a significant addition to the computer science literature. While it makes no great theoretical contributions, it is a cogent presentation of a wide range of pragmatic knowledge.

PAUL ABRAHAMS

Courant Institute of Mathematical Sciences
New York University
251 Mercer Street
New York, New York 10012