

FINITE STATE REPRESENTATION OF INTERACTIVE LANGUAGES

By Richard See

1. The remote user of on-line time-sharing systems now has a wealth of choices with respect to: computer language, cost profile for various aspects of use, and quality of performance, including reliability. In order to optimize with respect to one or more of these aspects, the user may choose to switch frequently from one language, or from one computer system, to another. Carried too far, however, this effort can be self-defeating, if the operator begins to become confused by the multitude of petty procedural differences found among the various systems today. This problem exists even for the different versions of the "same language" offered by several different systems. The purpose of this paper is to offer a formalism, the finite state diagram, as a means of representing variants of the same language in a form permitting the user to program himself, in effect, for interaction with a particular computer language dialect.

2. The Lister Hill National Center for Biomedical Communications is the component of the National Library of Medicine concerned with the development of a Biomedical Communications Network and with the development of improved biomedical information systems. In order to evaluate the utility of on-line systems in the biomedical communications context, the Center has been experimenting with about two dozen different systems, including four which offer the QED language. The QED language is of interest because it is easily learned by people with no previous experience with computers and because it permits the development, manipulation, and search of small files of unformatted information.

3. The QED language was originally developed at the University of California, Berkeley, by the Project GENIE Time-Sharing Development Group. Four variants, or dialects, of QED have been used and examined by the Center, thus far, with respect to costs and ease of use. In the course of this examination it was discovered that the small dialect differences could cause considerable confusion and irritation when one switched back and forth among the dialects. The manuals for QED usually run to about 50 pages and are difficult to use quickly when one is at the terminal and wishes to issue the correct command. As an aid, therefore, the author compressed the salient features of several QED variants to a one-page diagram, which is basically a finite state diagram with a few embellishments.

4. Figure 1 is a partial diagram for one version of the QED language. Each box represents a state in which the terminal user finds himself and each directed line between boxes represents a possible change from one state to another. The QED commands and interactions typed by the user appear to the right (below) the directed lines. The QED system messages provided by the computer are shown to the left (above). Thus, the commands and system messages of the QED language represent the changes between states, while the states themselves are modes of operations of the system permitted by, or resulting from, the commands given. Thus, for example, the command APPEND puts the system in the state of permitting direct input at the terminal, but presupposes that one was already in the neutral QED state, with or without pre-existing files or text in the main text buffer, i. e., the work space.

5. Certain global properties of the system can be seen at a glance from the diagram. First, the left half of the diagram represents a collection of states in which the user has no files, while the right half consists of those states in which the user does have a file or files in the system. Next, it can be seen at once that the only way to get into a state with a file is to get into the QED+TEXT state and then WRITE the text into a named file, going into the QED+TEXT+FILE state. Furthermore, it is also apparent that the only way to get back into the states without a file is to get back into the EXECUTIVE+FILE state and thence to the EXECUTIVE (without FILE) state and back into QED. In addition to these more general properties a great many more specific details can be gotten readily from the diagram.

6. When similar diagrams are made for other QED systems two facts emerge. First, the structure of the various diagrams for the dialects of QED are very similar to the diagram in FIGURE 1. Second, many of the commands are different, even for parts of the state diagram which are identical. For example, to return from the EXECUTIVE from QED+TEXT some of the commands used are: QUIT, G<sup>C</sup>, and ESC (twice).

7. The exercise of compiling the diagram in FIGURE 1, proved worthwhile in itself, as a means of clarifying the details and the structure of the QED language. Secondly, the author has found that after having become familiar with the main features of the language the diagrams provide a very useful and rapid memory aid. Finally, the diagrams have proved helpful as instructional aids for those with little or no computer experience.

14 July 1969

