FINITE STATE REPRESENTATION OF INTERACTIVE LANGUAGES

Introduction:

Description of data may go beyond the definition of its structure, logical or physical, and the nature of its use by the procedure that generated it. If the data is intended as input to a program and there is more than one possible input, part of the data definition must include some equivalent of an input number.

Sometimes the concept of an input number becomes useless because the amount of input data read is itself data dependent; input number n might properly be read as input $n+3$, input $n-20$, or never read at all. In such cases the concept of a program state replaces that of indexed data. The state may be defined by a range of values for a set of internal variables or the name of an input statement. Another way to describe the state would be by including a set of inputs necessary to bring the program, from the time its execution begins, into a state ready to process the data.

In case the progarm is a programming system, the state definition accompanying the data may be the only possible way to identify it. For instance, when sending a message to an interactive system, the needed state accompanying the data may be compared by a monitor with the present state of the system and the necessary commands then issued to update the state. To do this one needs a state translator (which in some cases reduces to a command translator) between systems, and transition diagrams for each system.

The following article, by Dr. Richard See of the National Library of Medicine, includes this information for two systems. Even though the systems are quite similar, there exist problems of command translation and state transition.

It would be helpful for the computer user of one system, and may be necessary to the user on an interactive network, to have these diagrams for the systems he may use. And unless he wants to learn the details of all the systems he wishes to query, it will be necessary for some program residing on the network to have this information in some equivalent form. There is some of this information in the CODASYL 'Survey of Generalized Data Base Management Systems,' but not so clearly schematized. Something equivalent to Dr. See's approach will be a worthwhile task for CODASYL or a similar group to begin for those languages which perform information retrieval or data base management. One possible similar group would be SICFIDET.

Don Hatfield