procedure is sufficiently simple, it is stored in a single file containing its source text. Versions of a file may then be introduced by the F-V mechanism described above. Of more interest are the procedures which exhibit more complex variation. Such procedures are catalogs, their source components being found in subordinate files, as shown in Fig. 4. The HEADER file contains only the Pascal header for the procedure, defining the procedure name and the type of its parameters and result, if any. The files COMMENT and BLOCK contain a Pascal comment and the statement body of the procedure bounded by **begin-end,** respectively. Any of the files named with the reserved file names, except the HEADER and BLOCK files, may be omitted in the definition of a procedure.

Any one of the components of a procedure may be allowed to vary independently in different versions of the package. For example, the CONST file might have a version with values chosen for varying memory sizes, while the BLOCK file might have versions with specialized algorithms optimized for different performance criteria. The independence of the variation of the two components of the procedure is supported quite naturally.

Using this approach and allowing arbitrary subsets of the output devices to be configured for an arbitrary number of target systems, there would be an exponential number of versions of the BLOCK for each device-dependent procedure in Level 1. This is clearly unacceptable. For this reason a BLOCK may be a catalog containing files which form the body of the block. This facility permits a **case** statement to be separated into its syntactic components; each component being stored in its own file. For an illustration of this see Fig. 5. The PROLOG and EPILOG files immediately below the BLOCK catalog form the "prologue" and "epilogue" of a block containing a **case** statement. That is, the block PROLOG is everything from the **begin** through to the statement preceding the **case,** while the block EPILOG is everything after the **case** statement through to the closing **end.** The **case** statement itself is found under the CASE catalog. The **case** statement also has a PROLOG and EPILOG. The **case** PROLOG contains the **case** keyword, the selector expression and the **of** keyword. The **case** EPILOG contains the closing **end;** for the **case** PROLOG. The EPILOG may also contain an **else*** statement to trap selections to unconfigured devices in a particular target system. Excluding the PROLOG and EPILOG files below the CASE catalog, each of the remaining files contain a labelled statement for a particular device case. The concatenation of the PROLOG, any subset of the CASE files and the EPILOG now produces a valid **case** statement. When the block PROLOG and EPILOG are then placed around this, it yields the BLOCK for the procedure. Again, any of these files, PROLOGs, EPILOGs and cases alike, could have versions associated with them as above.

The Pascal Inclusion Builder

The principal tool needed to make the structure function is "Pinc", the Pascal Inclusion Builder. Pinc's function is to output an "inclusion file", containing the file system pathnames of a consistent set of files defining a compilable instance of the package or level of the package. This output may then be input to the Pascal compiler to generate an executable program or a semi-compiled library. Given the pathname of the root of a tree having a structure as outlined above, Pinc interrogates the file system to determine its structure and then traverses this structure "pruning" away (from an internal representation of the structure) those files which will not be needed in the compilation instance it is preparing. The decisions on where to prune are based on files of directives supplied by the user and augmented by dialogue with the user if these directives are inconsistent or incomplete.

Pinc addresses two other problems: external and forward procedure declarations. External procedure declarations are required for the kind of separate compilation supported by the

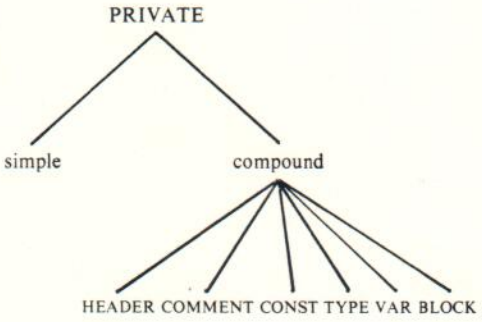

Fig. 4 Two basic procedure structures

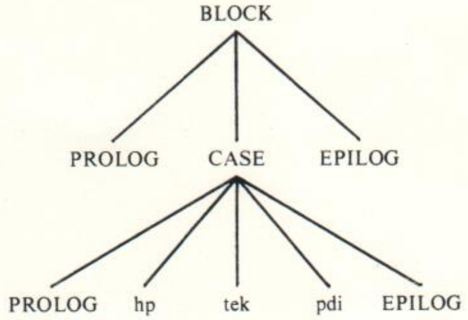

Fig. 5 Case Statement

* The **case else** or **default** clause is being considered as part of the proposed standard but it is not standard Pascal at present.