is packed, and has a component type that either is the same type as the component type of T or else, if S contains another component conformant array schema, is a conformant type derived through the component schema from the component type of T. The bound identifiers introduced in the index type specification denote the smallest and largest values of the index type of the conformant type.

11.4. Predeclared Procedures

11.4.1. File handling procedures. There are several predeclared procedures that are specifically defined for use with textfiles. These are described in detail in Section 12. The following procedures operate on any file variable f (see Sections 6.4.2 and 7.4).

`Rewrite(f)` causes f to have an empty sequence and to be in generation mode.

`Put(f)` is an error if f is undefined or is not in generation mode, or if the buffer variable f↑ is undefined. Appends the value of f↑ to the end of the sequence of f.

`Reset(f)` causes f to be placed in inspection mode, and the position in its sequence becomes the first position. If the sequence is empty, eof(f) becomes true and f↑ becomes totally undefined; otherwise, eof(f) becomes false and f↑ takes on the value of the first component of the sequence.

`Get(f)` is an error if f is undefined or if eof(f) is true. Causes the position in the sequence to be advanced to the next component, if any, and f↑ to take on its value; if no next component exists, eof(f) becomes true and f↑ becomes totally undefined.

In each of the following definitions, all occurrences of f denote the same file non–text file variable, the symbols v,v1,...,vn represent variables, and e,e1,...,en represent expressions. Note that the variables v,v1,..., and vn are not actual variable parameters, and thus they may be components of packed arrays or records. Read and Write of textfiles are defined in Section 12.