Annex B

(informative)

Decremental features

B.1 Deleted features

The deleted features are those features of Fortran 90 that were redundant and are considered largely unused. Section 1.7.1 describes the nature of the deleted features. The Fortran 90 features that are not contained in Fortran 95 or this standard are the following:

(1) Real and double precision DO variables.

The ability present in FORTRAN 77, and for consistency also in Fortran 90, for a DO variable to be of type real or double precision in addition to type integer, has been deleted. A similar result can be achieved by using a DO construct with no loop control and the appropriate exit test.

(2) Branching to a END IF statement from outside its block.

In FORTRAN 77, and for consistency also in Fortran 90, it was possible to branch to an END IF statement from outside the IF construct; this has been deleted. A similar result can be achieved by branching to a CONTINUE statement that is immediately after the END IF statement.

(3) PAUSE statement.

The PAUSE statement, present in FORTRAN 66, FORTRAN 77 and for consistency also in Fortran 90, has been deleted. A similar result can be achieved by writing a message to the appropriate unit, followed by reading from the appropriate unit.

(4) ASSIGN and assigned GO TO statements and assigned format specifiers.

The ASSIGN statement and the related assigned GO TO statement, present in FORTRAN 66, FORTRAN 77 and for consistency also in Fortran 90, have been deleted. Further, the ability to use an assigned integer as a format, present in FORTRAN 77 and Fortran 90, has been deleted. A similar result can be achieved by using other control constructs instead of the assigned GOTO statement and by using a default character variable to hold a format specification instead of using an assigned integer.

(5) H edit descriptor.

In FORTRAN 77, and for consistency also in Fortran 90, there was an alternative form of character string edit descriptor, which had been the only such form in FORTRAN 66; this has been deleted. A similar result can be achieved by using a character string edit descriptor.

The following is a list of the previous editions of the international Fortran standard, along with their informal names:

ISO/IEC 1539:1972	FORTRAN 66
ISO/IEC 1539:1978	FORTRAN 77
ISO/IEC 1539:1991	Fortran 90
ISO/IEC 1539:1997	Fortran 95

See the Fortran 90 standard for detailed rules of how these deleted features work.

B.2 Obsolescent features

The obsolescent features are those features of Fortran 90 that were redundant and for which better methods were available in Fortran 90. Section 1.7.2 describes the nature of the obsolescent features. The obsolescent features in this standard are the following:

(1) Arithmetic IF — use the IF statement (8.1.2.4) or IF construct (8.1.2).

(2) Shared DO termination and termination on a statement other than END DO or CONTINUE — use an END DO or a CONTINUE statement for each DO statement.

(3) Alternate return — see B.2.1.

(4) Computed GO TO statement - see B.2.2.

(5) Statement functions - see B.2.3.

(6) DATA statements amongst executable statements - see B.2.4.

(7) Assumed length character functions - see B.2.5.

(8) Fixed form source - see B.2.6.

(9) CHARACTER* form of CHARACTER declaration - see B.2.7.

B.2.1 Alternate return

An alternate return introduces labels into an argument list to allow the called procedure to direct the execution of the caller upon return. The same effect can be achieved with a return code that is used in a CASE construct on return. This avoids an irregularity in the syntax and semantics of argument association. For example,

```
CALL SUBR_NAME (X, Y, Z, *100, *200, *300)
```

may be replaced by

```
CALL SUBR_NAME (X, Y, Z, RETURN_CODE)
SELECT CASE (RETURN_CODE)
   CASE (1)
      ...
   CASE (2)
      ...
   CASE (3)
      ...
   CASE DEFAULT
      ...
END SELECT
```

B.2.2 Computed GO TO statement

The computed GO TO has been superseded by the CASE construct, which is a generalized, easier to use and more efficient means of expressing the same computation.

B.2.3 Statement functions

Statement functions are subject to a number of nonintuitive restrictions and are a potential source of error since their syntax is easily confused with that of an assignment statement.

The internal function is a more generalized form of the statement function and completely supersedes it.

B.2.4 DATA statements among executables

The statement ordering rules of FORTRAN 66, and hence of FORTRAN 77 and Fortran 90 for compatibility, allowed DATA statements to appear anywhere in a program unit after the specification statements. The ability to position DATA statements amongst executable statements is very rarely used, is unnecessary and is a potential source of error.

B.2.5 Assumed character length functions

Assumed character length for functions is an irregularity in the language since elsewhere in Fortran the philosophy is that the attributes of a function result depend only on the actual arguments of the invocation and on any data accessible by the function through host or use association. Some uses of this facility can be replaced with an automatic character length function, where the length of the function result is declared in a specification expression. Other uses can be replaced by the use of a subroutine whose arguments correspond to the function result and the function arguments.

Note that dummy arguments of a function may be assumed character length.

B.2.6 Fixed form source

Fixed form source was designed when the principal machine-readable input medium for new programs was punched cards. Now that new and amended programs are generally entered via keyboards with screen displays, it is an unnecessary overhead, and is potentially error-prone, to have to locate positions 6, 7, or 72 on a line. Free form source was designed expressly for this more modern technology.

It is a simple matter for a software tool to convert from fixed to free form source.

B.2.7 CHARACTER* form of CHARACTER declaration

Fortran 90 had two different forms of specifying the length selector in CHARACTER declarations. The older form (CHARACTER*char-length) was an unnecessary redundancy.