FORTRAN – WATFIV

Mohammed Aslam
CIS 24
Prof. Kopec
Report 01
Date: 03/03/03

WATFIV is an improvement over WATFOR. The University of Waterloo, Ontario, Canada, also developed it. Its name comes from the fact that almost everything in computer science has a number, and that a new one of something is always given a number one greater then the previous one. Since FOR from WATFOR sounds like four, the natural next choice is WATFIV. Difference between WATFOR AND WATFIV stem mainly from the fact that more statements, and capabilities associated with statements, are available in WATFIV.

Differences between WATFIV and other Fortran versions are due mainly to the availability of additional features with WATFIV. Some are directly related to the FORTRAN language; others are related to control cards and system-dependent considerations. The language differences are listed below:

1. WATFIV allows WHILE/E ND WHILE, LOOP/END LOOP, LOOP/UNTIL, AT END, QUIT, DO CASE/CASE/IF NONE/END CASE statements; fortran-77 does not – you must use a combination of ifs and GOTOs to define the equivalent logic. DO loops may be nested to any level within the 255 DO statements per program limit under WATFIV.

2. Fortran-77 permits a negative increment, real parameters, and parameter expressions in the DO statement; WATFIV does not.
3. WATFIV tests for loop continuation after executing the body of the loop; fortran-77 tests before. Thus, a DO-loop in WATFIV will always execute the loop once whereas in fortran-77 it may not.
4. If the value of the integer variable in a computed GO TO statement is zero or negative, WATFIV transfers control to the next executable statement. This interpretation is an extension to standard FORTRAN that is also implemented in IBM 360/370 FORTRAN. In WATFOR, an error message is printed and program execution is terminated.

5. RETURN IS required in WATFIV but not in FORTRAN –77. STOP is required in WATFIV and is optional in Fortran-77.
6. WATFIV issues warning diagnostics if the proper ordering of statements is not established. Data declaration statements (type, DIMENSION, COMMON, and the like) should appear before statement function definitions, which should appear before the executable statements of a program.
7. WATFIV permits specifying the upper bound of an array dimension. The lower bound is always 1. In fortran-77, both the upper and lower bounds can be specified as in "REAL X (-5.5)". WATFIV allows actual arguments passed to subprograms to be arrays, array elements or single variables.
8. CHARACTER arrays in WATFIV require the length of each element to be declared immediately following the array name as in X*2(100). In fortran-77 the length modifier follows the right parenthesis e.g. X (100)*2
9. In WATFIV the dimension of a subprogram parameter array must be provide explicitly. In fortran-77, the use of an asterisk as the dimension bound in a subprogram array declaration causes the parameter array to have the same size as the argument array.
10. If a function subprogram has multiple entry points, WATFIV equivalences the functions and entry-point names. WATFOR does not.
11. WATFOR does not permit the declaration of variables of CHARACTER type.

12. Fortran-77 uses an open statement to specify the attributes of an external file and make its available for reading or writing. In WATFIV, operating system commands and statements prior to running a job provide any necessary file specifications. Opening a file in WATFIV is done implicitly when the first reference to the file is encountered at execution time. WATFOR does not permit direct-access I/O.
13. For sequential files in fortran-77, the parameter specifications in a READ are provided by key word = value pairs. In WATFIV, the ordering of the parameter values is significant.
14. Fortran-77 includes the format codes B (for blank suppression), S (for sign control) and a colon (for output truncation); WATFIV does not.
15. Format-free input –output in WATFIV has the form "READ, list"; in fortran-77, the form is "READ*, list". WATFOR does not permit use of the name list statement.
16. Fortran-77 has a library of eight character manipulation functions such as LEN (to obtain the length of a string) and INDEX (to find the starting position of a given sub string). WATFIV has no comparable built-in functions although essentially equivalent subprograms can be written.

EXTENTIONS:

Extensions are features that are acceptable to WATFIV but are not likely to compile correctly when compilers other than WATFIV are used. Major extensions are:

1. WATFIV format-free input/output allows the programmer to easily read input or provide output in man-readable form without referring to FORMAT statements. The forms of WATFIV format-free input/output statements are:
   READ, list
   PRINT, list
   PUNCH, list
   READ (UNIT, *[, END=label1][, ERR=label2]) list
   WRITE (unit, *) list
2. Variables can be declared to be of CHARACTER type, and the values of character variables can be moved, compared, read as input, and written as output.
3. Multiple assignment statements of the form
   V1=v2=…. =expression
   Are permissible.
4. Subscripts may be used on the right-hand side of a statement function definition, for example,
   FUNC (A, B, C)=(A+B)/ARR (1) * C ** 2
5. A value of COMPLEX, LOGICAL, O or CHARACTER type can be used as the subscript of an array.
6. Implied Dos as used in I/O lists of input/output statements are also permitted in DATA statements, for example,
   DATA (A (1), 1,20,2) 10*1.0/

RESTRICTIONS:

Restrictions are minor rules of language usage that must be followed by WATFIV users but are not generally imposed in FORTRAN. Many of them are necessary because WATFIV is a one-pass compiler: All translations is done during one scan of source-program statements; there is not first pass to reserve storage locations and determine addresses, followed by a second pass

to interpret locations and determine addresses, followed by a second pass to interpret statements with reference to tables of information already established.

1. Data declaration statements referring to variables used in NAME LIST or DEFINE FILE statements must precede the NAMELIST or DEFINE FILE statements.
2. No more than 255 DO statements can be used in one program.
3. The character sequence FORMAT is a reserved sequence when used as the first six nonblank characters of a statement. It must signal the beginning of a format statement.
4. WATFIV terminates program execution if floating-point underflow occurs.
5. WATFIV considers a program to be in error if it executes a statement of the form RETURN i in with the value of I is zero, negative undefined, or grater then the number of statement-label arguments appearing in the argument list of the activating CALL statement.
6. WATFIV evaluates all functions that require complicated approximations double precision, even if the return value of such a function is a single precision numeral.
7. Under WATFIV, REAL*4 values are printed with a maximum of seven significant digits, in correspondence with the definition of single –precision values.

Sample Program:

```
1. INTEGER I, J
2. REAL R
3. READ, I, J, R
4. PRINT, J
5. PRINT, R, I
6. STOP
7. END
```

Line 1 and 2 are declaration of variables. Line 3 reads the variables and line 4 and 5 prints them.

Efficiency comparison between WATFIV and Fortran-77:

	WATFIV	Fortran-77
Compile speed	Better	Good
Execute speed	Good	Better
Compile must precede execution	Yes	No
Diagnostic messages	Better	Good

Table of different attributes of WATFIV:

Attributes	Comments	Rating (Out of 10)
Context	Simple	7
Structured	Yes	8
Conditions statements	Yes. Supports if-then statements	10
GO TO statements	Yes	5
Looping statements	Yes. Supports DO/WHILE loops	8
Used for	Mathematical computations	10
Object oriented	No	0
Arrays	Yes	8
Nested loops	Yes. Up to 255 DO loops	5
Characters	Yes. Supports Data of character types	6
Direct I/O access	Yes. Permits direct I/O access	6
String	No. It does not have any built in functions.	0

Bibliography

Moore, John B. *WATFIV: Fortran Programming with the WATFIV Compiler*, Reston Publishing Company, Inc. Reston, Virginia, 1975

Moore, John B. *Structured Fortran with WATFIV*, Reston Publishing Company, Inc. Reston, Virginia, 1985

Parker, James L. *Fortran Programming and WATFIV*, science Research Associates, Inc. Chicago, 1973