

Subject: A few small problems with procedure pointers
From: Van Snyder
References: 00-245

1 Introduction

Maybe I should stop reading 7.5.2. Every time I do, I find problems with pointer assignment – this time mostly with the constraints. The general problem is that “type” is used without qualifying it to the case of data pointers. The term “type” is also apparently intended to refer to the result type specified in the interface for procedure pointers or procedures, but that’s not the same as the type of the pointer – procedure pointers don’t have types. The constraints would also be easier to understand, and could be simplified slightly, if they were grouped according to whether they apply to all pointer assignments, only for data, or only for procedures.

2 Edits

Edits refer to 00-007r3. Page and line numbers are displayed in the margin. Absent other instructions, a page and line number or line number range implies all of the indicated text is to be replaced by immediately following text, while a page and line number followed by + (-) indicates that immediately following text is to be inserted after (before) the indicated line. Remarks for the editor are noted in the margin, or appear between [and] in the text.

[Editor: Before “an” insert “is a data object that is”. This edit makes the constraint more precise; it may not be *necessary*, because a procedure pointer, even one that returns an array, is not an array. Nonetheless, it is helpful.] 139:10

[Editor: Move the constraint at [139:23] to here.] 139:10+

The following constraints apply only if *pointer-object* is not a procedure pointer:

The next constraint is intended to replace the one at [139:32-33], which is defective in not including “procedure pointer” in the conditions for <i>target</i> .)	<i>Note to J3</i> <i>Not an edit</i>
---	---

Constraint: The *target* shall not be a *procedure-name*, a procedure pointer, or an *expr* the result of which is a procedure pointer.

[Editor: Replace “If ... object,” by “The”.] 139:13

[Editor: Replace “If ... it” by “The *pointer-object*” and insert “the” before “*target*”.] 139:15

[Editor: Delete – Replaced in the edit for [139:32-33] below.] 139:19-20

The following constraints apply only if *pointer-object* is a procedure pointer: 139:22

[Editor: After making the changes indicated for [139:28] below, move the constraint at [139:28-31] to here.]

[Editor: “pointer” ⇒ “POINTER” (cf. [139:13])] 139:23

[Editor: Remove “is ... that”. Before “the” insert “*target* shall have an explicit interface, and”.] 139:24

[Editor: Replace “If ... pointer,” by “The”. Replace “an” by “a nonelemental”.] 139:28

[Editor: The constraint at this place is superseded by one introduced by the edit for [139:10+] above. Replace it by the following:] 139:32-33

Constraint: If *pointer-object* is a subroutine procedure pointer, *target* shall not be a function, a function procedure pointer, or an *expr* the result of which is a function procedure pointer.

Constraint: If *pointer-object* is a function procedure pointer, *target* shall not be a subroutine, a subroutine procedure pointer, or an *expr* the result of which is a subroutine procedure pointer.

It would at first seem that the above two upside-down and inside-out constraints could be simpler, but we have procedure pointers and external entities that are neither known to be subroutines nor known to be functions. The constraints are also not covered by the one at [139:24-27] because that one applies only if the <i>pointer-object</i> has explicit interface.

Note to J3
Not an edit

Constraint: If *pointer-object* is a function procedure pointer and a result type is specified for *target*, the characteristics of the result types (12.2.2), other than the values of nonkind type parameters, shall be the same.

[Editor: After “*pointer-object*” insert “is a data object that” twice.]

139:41,43

[Editor: Replace “If *target* is a disassociated pointer” by:]

139:44

If *pointer-object* is a data object or a function procedure pointer, the following conditions apply. In the case of a function procedure pointer, the term “type” applies to the result type of the function.

- When *target* is a disassociated pointer or an expression the result of which is a disassociated pointer

[Editor: Replace “Otherwise” by:]

139:46

- When *target* is an associated pointer or an expression the result of which is an associated pointer

[Editor: Put a bullet in front of “If”.]

140:3

[Editor: An *expr* doesn’t have an association status. After “status” insert “nor shall the result of *expr* have undefined association status”.]

140:4

In addition to pointer assignment, a data object with the POINTER attribute becomes associate with a target by execution of an ALLOCATE statement (6.3.1), and becomes disassociated from a target by execution of a NULLIFY statement (6.3.2) or a DEALLOCATE statement (6.3.3).

140:19-20