To: J3
From: Malcolm Cohen
Subject: Unresolved issues 321 and 322
Date: 6th June 2001

1. Introduction

The editor says
  "I would have thought that a procedure pointer was a data object (and a variable)"

He might think that, but the standard does not.

Way back when deciding how to implement the "pointers to procedures" requirement, there were two possible options:
(1) procedure variables: these are variables whose "value" is a "procedure".
(2) procedure pointers: these are not variables, but sort-of-like-dummy procedures which may be pointer-associated with "real" procedures.

We took option 2, not option 1. If "procedure things" were variables, we would not need to fuss with the pointer attribute at all. We are far too far down the line to start reconsidering the approach.

Now admittedly section 2.4 is written assuming that things like procedure pointers do not even exist. However, things like
•   "A data entity has a data type" (subroutine procedure pointers do not)
and
•   "A **variable** may have a value and may be defined and redefined during execution" (procedure pointers do not have a value and cannot be defined or redefined)
make it very difficult to consider procedure pointers as being anything like data objects.

This paper assumes that we will continue to treat procedure pointers as being procedures, not as being variables.

2. More Introduction

Issue 321 points out that procedure pointers are not properly catered for in several places, in particular:
•   SAVE statement
•   POINTER statement
•   COMMON blocks
•   R1217/1218/1219

Not to mention the definition of pointers in 2.4! Also, pointer assignment is pretty dodgy, but we'll clean that up in another paper.

Do we really want to allow procedure pointers in COMMON? This seems pretty futile. Oh well, I suppose we must.

Also, in pointer assignment the first constraint, at [137:20], duplicates the constraint on all *pointer-object*s (see [108:10] where this constraint follows the BNF definition of pointer-object) and should be removed.

3. Discussion

The previously decided treatment of "procedure things" as "procedures with the pointer attribute" will not work if the pointer attribute "leaks through" from the result variable to the function (in Fortran 77, and thus presumably in Fortran 90, the data type leaks through; but it is unclear as to whether any other attributes leak through). We may need to do some work here.

4. Edits to 01-007r1

[18:24] After "variable" insert "or procedure".

{Include procedure pointers in the definition of pointer.}

[19:40-41] Replace sentence with
  "A **procedure reference** is the appearance of the procedure designator, operator symbol, or assignment symbol in a context requiring execution of the procedure at that point."
{Current sentence is badly formed (three-element list with no comma) and broken with respect to procedure pointers.}

[79:5] Before "pointer" insert "data".
{This is untrue for procedure pointers.}

[79:6+] Insert new paragraph
"A procedure pointer shall not be referenced unless it is pointer associated with a target procedure."
{We should say something about procedure pointers here.}

[79:21+] Insert new paragraph
"A procedure pointer with the SAVE attribute retains its association status when execution of an instance of any subprogram completes."
{Say something about procedure pointers. Saved procedure pointers only change status through NULLIFY and pointer assignment.}

[85:23-25] Replace with
"R545 *pointer-stmt* **is** POINTER [::] *pointer-decl-list*

 R545a *pointer-decl* **is** *object-name* [(*deferred-shape-spec-list*)]
            or *proc-entity-name*

 Constraint: A *proc-entity-name* shall also be declared in a *procedure-declaration-stmt*."

 This statement specifies the POINTER attribute (5.1.2.11) for a list of objects and procedure entities."
{The funny-looking constraint is because that is the only way we allow a procedure pointer to be declared.}

[85:32+] Insert "**or** *proc-pointer-name"*

[86:1+] Insert
"R547a *proc-pointer-name* **is** *name*
 Constraint: A *proc-pointer-name* shall be the name of a procedure pointer."
{Insert new syntax term with required constraint. Maybe this should be done somewhere else? (though this is the first use of it).}

[86:2-14] Delete J3 note 321.

[86:18] Change "objects" to "entities".
{Procedure pointers are not objects but are entities.}

[92:40+] Insert "**or** *proc-pointer-name*".
{Allow procedure pointers in COMMON.}

[93:7] After "*variable-name*" insert "or *proc-pointer-name*".

[94:18] Insert "data" before "pointer", twice.
{Type matching needs to be done differently for procedure pointers.}

[94:18] Delete ", type parameters,".

[94:19] Append
  "Data pointers that are storage associated shall have deferred the same type parameters; corresponding nondeferred type parameters shall have the same value."
{Type parameter matching needs to take account of deferred type parameters.}

[94:19+] Append to paragraph
"A procedure pointer shall become storage associated only with another procedure pointer and either both interfaces are explicit or both interfaces are implicit. If the interfaces are explicit, the characteristics shall be the same. If the interfaces are implicit, either both shall be subroutines or both shall be functions with the same type and type parameters."
{Rules for matching procedure pointers.}

[137:20] Delete constraint.
{It duplicates the one at [108:10], which is still in force.}

[254:14-34] Replace with
"R1217 *function-reference* **is** *procedure-designator* ([*actual-arg-spec-list*])

 Constraint: The *procedure-designator* shall designate a function.
 Constraint: The *actual-arg-spec-list* shall not contain an *alt-return-spec*.

 R1218 *subroutine-reference* **is** CALL *procedure-designator* [([*actual-arg-spec-list*])]

 Constraint: The *procedure-designator* shall designate a subroutine.

 R1218a *procedure-designator* **is** *procedure-name*
                    or *data-ref* % *procedure-component-name*
                    or *data-ref* % *binding-name*

 Constraint: A *procedure-name* shall be the name of a procedure or procedure pointer.
 Constraint: A *procedure-component-name* shall be the name of a procedure pointer component of the declared type of *data-ref*.
 Constraint: A *binding-name* shall be the name of a procedure binding (4.5.1.5) of the declared type of *data-ref*.
"
{Replace procedure reference syntax, factoring out how we got to the procedure.
 Also: delete redundant constraint at [254:17-18].}