

To: J3
 From: Malcolm Cohen
 Subject: Unresolved issue 328
 Date: 7th June 2001

1. Introduction

Issue 328, says, apropos our attempts to have the scope of type-bound operators do "the right thing",

"This sounds like a pretty subtle ... way to say that type-bound operators and assignment "stick" with data objects of the type. ... And for the assignment symbol and intrinsic operators, this sounds like it contradicts 14.4 and 14.5, which say those are global"

Yes, on reflection I agree that this is too subtle and could use more explanation.

On the other hand, it is obviously nonsense for the assignment symbol to be "global", since overloading assignment locally acts as if it were a local entity (the overloading does not happen magically across the entire program). Ditto for intrinsic operators.

Additionally, there is no indication as to the scope of a *dtio-generic-spec*. Obviously these ought to be ordinary local entities.

2. Proposal

That the scope of the type-bound operators et al be the derived type definition, and that the "lookup" procedures for assignment, operators and dtio be changed to spell out explicitly that the type-bound operators are looked for before looking for interface blocks etc. Much of this has already been done. Additionally, the ambiguity rules should say explicitly that they include the type-bound operators.

While we're at it, we may as well fix up the scope of the assignment symbol and intrinsic operators. And add the missing scope definition for dtio generic specs.

3. Edits to 01-007r1

[48:41-42] Delete.

{We'll leave the scope as per normal.}

[49:1-10] Delete J3 note 328.

[130:35] After "A" insert

"type-bound generic binding (4.5.1.5) in the dynamic type of x_2 with a *generic-spec* of OPERATOR(*op*) specifies the function, and there is a corresponding specific interface in the declared type of x_2 ; or a"

[130:35] Delete "4.5.1.5,".

{Explicitly specify how we get to type-bound operators.}

[130:40-41] Replace with

" (a) The rank of x_2 matches the rank of d_2 or
 (b) The function is elemental and there is no other function that defines the operation.

If d_2 is an array, the shape of x_2 shall match the shape of d_2 ."

{EDITOR: Note that the last line is outwith both lists.}

{Fix ambiguity over whether to reference an elemental vs. a function with an explicit rank, and make it compile-time decidable. Probably ought to be an F95 interp, ho hum.}

[131:5] After "A" insert

"type-bound generic binding (4.5.1.5) in the dynamic type of $x1$ or $x2$ with a *generic-spec* of OPERATOR(op) specifies the function, and there is a corresponding specific interface in the corresponding declared type; or a"

[131:5] Delete "4.5.1.5,".

{Explicitly specify how we get to type-bound operators.}

[131:11-13] Replace with

- " (a) The ranks of $x1$ and $x2$ match those of $d1$ and $d2$, or
 (b) The function is elemental and there is no other function that defines the operation.

If $d1$ or $d2$ is an array, the shapes of $x1$ and $x2$ shall match the shapes of $d1$ and $d2$, respectively."

{EDITOR: Note that the last line is outwith both lists.}

{Fix ambiguity over whether to reference an elemental vs. a function with explicit ranks, and make it compile-time decidable. Probably ought to be an F95 interp, ho hum.}

[136:36] After "A" insert

"type-bound generic binding (4.5.1.5) in the dynamic type of $x1$ or $x2$ with a *generic-spec* of ASSIGNMENT(=) specifies the function, and there is a corresponding specific interface in the corresponding declared type; or a"

[136:36] Delete "4.5.1.5,".

{Explicitly look up and dispatch tbps.}

[136:41-44] Replace with

- " (a) The ranks of $x1$ and $x2$ match those of $d1$ and $d2$, or
 (b) The subroutine is elemental and there is no other subroutine that defines the assignment.

If $d1$ or $d2$ is an array, the shapes of $x1$ and $x2$ shall match the shapes of $d1$ and $d2$, respectively."

{EDITOR: Note that the last line is outwith both lists.}

{Fix ambiguity over whether to reference an elemental vs. a subroutine with explicit ranks, and make it compile-time decidable. Probably ought to be an F95 interp, ho hum.}

[343:34] Append

"These rules also apply between generic interface blocks accessible in a scope and type-bound generic bindings other than names, for all types that are accessible or have accessible objects in that scope."

{Specify that the unambiguity rules always apply.}

[347:26] After "if" insert

"the declared type of the effective item has a suitable type-bound generic binding or"

{Explicitly look for a type-bound binding.}

[347:27] Replace "If ... type-bound" with "If a type-bound binding is found".

{Reword. This edit will be subsumed by 01-250.}

[349:17-19] Replace with "An operator is a local entity.".

[349:21-23] Replace with "The assignment symbol is a local entity."

{These now include type-bound things.}

[349:23+] Insert

"14.5a Scope of derived-type input-output generic specifiers

A derived-type input-output generic specifier is a local entity."

{Specify the scope of dtio generic specs.}