Subject: Constraintify intrinsic assignment, or at least clean it a bit
From: Van Snyder
Reference: 09-007r1, 09-206, 09-207

1 Discussion

Eight of the ten requirements in the list in 7.2.1.2 [155:8ff] could be constraints. I would prefer that processors be required to diagnose these failures than that they be allowed to require me to track them down with a debugger.

The style varies between the list items, sometimes using ordinary nouns and sometimes using syntax terms, sometimes using "if . . . ," and sometimes using "if . . . then".

The number of requirements could be reduced from ten to six by avoiding repetition, combining related ones, and expanding Table 7.10.

The requirement for rank equality in the case that both *expr* and *variable* are arrays is well hidden in the shape conformance requirement in item (4) in the list in 7.2.1.2. It should be explicit.

The requirement for rank equality in the case that *variable* is an unallocated allocatable variable should be in the list of requirements in 7.2.1.2, not hiding in 7.2.1.3 (but see 09-207).

The edits would be simpler if a new term **coobject** were defined to be a coarray or a coindexed object, but 7.2.1.2 would be the only subclause where the term appeared.

I found two of the requirements in the list in 7.2.1.2 to be confusing because of double negatives:

(4) The shapes of the variable and *expr* shall conform unless the variable is an allocatable array that has the same rank as *expr* and is neither a coarray nor a coindexed object,

(10) If the variable is of derived type each length type parameter of the variable shall have the same value as the corresponding type parameter of *expr* unless the variable is allocatable, is not a coarray or coindexed object, and its corresponding type parameter is deferred.

It is not intended that this paper, on its own, advocate any technical changes.

2 Edits w.r.t. 09-007r1

Underlined terms are hyperlinked references in the current text that ought to be preserved.

[155:4-5 7.2.1.1 C713+]

Editor: Replace R732 and C713:

"

R732	*assignment-stmt*	**is**	*defined-assignment-stmt*
		or	*intrinsic-assignment-stmt*

"

[155:10-26 7.2.1.2p1]

Editor: Delete "In an intrinsic assignment," because the subclause heading is "Intrinsic assignment statement", (7.2.1.1p1 requires intrinsic assignment statements to meet the requirements for intrinsic assignment statements, and after the following edits we won't need it to introduce a list) and replace the list in 7.2.1.2 by the following:

"

R732a *intrinsic-assignment-stmt* **is** *variable* = *expr*

C713a (R732a) The *variable* shall not be a whole assumed-size array.

C713b (R732a) If *expr* is an array or if *variable* is polymorphic, *variable* and *expr* shall have the same rank.

{Editor: If 09-206 passes, delete "or if *variable* is polymorphic" from C713b.}

C713c (R732a) If *variable* is polymorphic it shall be allocatable and shall not be a coarray or coindexed object.

C713d (R732a) If *variable* is a coindexed object it shall not be of a type that has an allocatable ultimate component.

C713e (R732a) The declared types and kind type parameter values of *variable* and *expr* shall conform as specified in Table 7.10.

Table 7.10: **Type and kind conformance for intrinsic assignment statements**

Type and kind type parameter values of *variable*		Type and kind type parameter values of *expr*	
Type	Kind type parameter values	Type	Kind type parameter values
numeric	any	numeric	any
character	ISO 10646, default, or ASCII	character	ISO 10646, default, or ASCII
character	neither ISO 10646, default, nor ASCII	character	same as *variable*
logical	any	logical	any
type compatible (4.3.1.3) with *expr*	same as corresponding kind type parameters of *expr*	derived type	any

If *variable* is a coarray, a coindexed object, or is not allocatable, the shapes of *variable* and *expr* shall conform (1.3.31).

If *variable* is of derived type and if it is a coarray, a coindexed object, or is not allocatable, each nondeferred length type parameter of *variable* shall have the same value as the corresponding type parameter of *expr*.

{Editor: Do not do the next one if 09-207 passes.}

If *variable* is an unallocated allocatable array, *expr* shall have the same rank as *variable*."

{It would be nice, but not necessary, to get Table 7.10 into Annex D.}

[155:28 7.2.1.2p3]

Editor: Replace "the variable" by "*variable*".

[156:10 7.2.1.3p3]

Editor: Delete the first sentence, *viz.* "If the variable ... rank." It's now in 7.2.1.2, where it belongs, or it's gone altogether if 09-207 passes.

[158:21+ 7.2.1.4p1+]

Editor: Insert the following syntax rule and constraint:

"

R732b *defined-assignment-stmt* **is** *variable* = *expr*

C713f (R732b) The *variable* shall not be a whole assumed-size array.

"

{We already had an e-mail discussion about C713f, which I believe not to be technically necessary because we do allow whole assumed-size array actual arguments to correspond to the first dummy argument of the subroutine that defines assignment in references by CALL statements. 12.4.3.4.3p2 [287:1-3] covers all the other cases, such as finalization and default initialization. Others find it desirable on stylistic grounds.}

3 First alternative edits

Same as the first edits, except do not replace R732 and C713 with new syntax rules R732a and R732b and constraints C713a and C713f, and do not refer constraints C713b-C713e to a syntax rule. If we referred them to R732, they would apply to defined assignment as well as to intrinsic assignment.

4 Second alternative edits

No new syntax rules or constraints.

[155:11-27 7.2.1.2p1-2]

Editor: Replace the list and Table 7.10 by the following.

(1) if *expr* is an array or if *variable* is polymorphic or an unallocated allocatable array, *variable* and *expr* shall have the same rank,
{Editor: If 09-206 passes, delete "polymorphic or". If 09-207 passes, delete "or an unallocated allocatable array". If they both pass, delete "or if ... array".}

(2) if *variable* is polymorphic it shall be allocatable and shall not be a coarray or coindexed object,

(3) if *variable* is a coindexed object it shall not be of a type that has an allocatable ultimate component,

(4) if *variable* is a coarray, a coindexed object, or is not allocatable, the shapes of *variable* and *expr* shall conform (1.3.31),

(5) if *variable* is of derived type and if it is a coarray, a coindexed object, or is not allocatable, each nondeferred length type parameter of *variable* shall have the same value as the corresponding type parameter of *expr*, and

(6) the declared types and kind type parameter values of *variable* and *expr* shall conform as specified in Table 7.10.

Table 7.10: **Type and kind conformance for intrinsic assignment statements**

Type and kind type parameter values of *variable*		Type and kind type parameter values of *expr*	
Type	Kind type parameter values	Type	Kind type parameter values
numeric	any	numeric	any
character	ISO 10646, default, or ASCII	character	ISO 10646, default, or ASCII
character	neither ISO 10646, default, nor ASCII	character	same as *variable*
logical	any	logical	any
type compatible (4.3.1.3) with *expr*	same as corresponding kind type parameters of *expr*	derived type	any

"

[155:28 7.2.1.2p3]
Editor: Replace "the variable" by "*variable*"

[156:10 7.2.1.3p3]
Editor: Delete the first sentence, *viz.* "If the variable ... rank." It's now part of the list in 7.2.1.2, where it belongs, or it's gone altogether if 09-207 passes.