

Subject: Comments on Section 13

From: Van Snyder

1 Introduction

Here are several things that may or may not need attention. I don't even offer edits (well, sometimes I offer crappy ones). If they need attention, we can develop edits at the meeting, if we have time, or insert unresolved issue notes.

There is more gratuitous repetition in section 13 than in other sections. The following subclauses should be deleted because they contribute nothing, either in information content or organization, that is not done better elsewhere: 13.4, 13.5 except for 13.5.7, 13.6, 13.7 except for 13.7.2, 13.8 except for 13.8.0, 13.8.1 and 13.8.2, 13.9, 13.10, 13.11, 13.12 and 13.13 except for 13.13.0. There is some value in 13.14 summarizing by category, but it would not be a tragedy to delete it, too. Page and line numbers refer to 00-007r3.

| | |
|---|-------------------------|
| Does the "or argument keywords" part include the MAX and MIN functions? The problem with MAX and MIN is that they can have an unlimited number of arguments, but the standard only specifies the names of three of them. | 279:28 |
| "called" ⇒ "named". | 280:8 |
| After "and" insert "its value". | 280:9 |
| There is a Frame-ism (724) here. | 280:13 |
| The phrase "or character operations" is out-of-place in a subclause entitled "Numeric functions". Delete it. See the remark for [280:27] below. | 280:22 |
| After "ADJUSTR" insert ", MAX, MIN". | 280:27 |
| After "name" insert "that does not have deferred length". | 280:33 |
| It is always allowed for a function to determine its result without examining its argument. By saying "It is not necessary ... otherwise" at these two places, it brings the general rule into question at places where it's not mentioned. | 280:33-35, 281:25-26 |
| "which" ⇒ "that". | 281:39 |
| The left-hand side needs an x . | 282:10 |
| The word "geometric" is confusing here. Maybe "value" would be better. | 283:4 |
| The type of every argument is specified in 13.17. Why do we do it again here? | 283:20 |
| The sentence is about SPREAD. Delete "(SPREAD does this" and ")". | 284:5-6 |
| The "same elements" phrase could be interpreted "all of the elements are the same," but that's probably not right. After "elements" insert "as its SOURCE argument". | 284:9 |
| "two-dimensional" ⇒ "rank-two" (compare to [144:23]). | 284:12 |
| After "whether" insert "the dynamic type of". | 284:29 |
| Subclause 13.14.0 is about intrinsic functions. The word "generic" in its title doesn't contribute anything, and is confusing and misleading. Some of subclause 13.14 duplicates 13.3. The rest belongs there. Move [285:27-28, 32-42] to [280:13+], where arguments of intrinsics are addressed, delete [285:39-31] because it duplicates [279:35-38]. Then either (1) delete the subclause title at [285:26] and the rest of subclause 13.14 and all of subclause 13.15 [286:1-289:39] because they | 285:26- 289:39 |

say nothing that's not said in 13.17, or change the subclause title at [285:26] to "Summary of intrinsic functions by category".

| | |
|---|------------------|
| The word "Logical" could easily be confused with the type of the same name, but that would be wrong here. "Logical" \Rightarrow "bitwise" thrice. But don't bother if the result of the remark for [285:26-289:39] is to delete the passages addressed here. | 287:38- 288:2 |
| Replace "Character ... KIND('A')" by "Default character of length one". | 292:9 |
| Although everybody probably knows what is intended here, the text doesn't precisely say it. Before "KIND" insert "the value of" because this is an exception to the general rule that it's the kind of an argument, not its value, that disambiguates genericity. | 293:15 |
| Here and at numerous subsequent places, square brackets are used for arrays. Does the notation need explanation? | 294:1-3 |
| Delete "or not" because it doesn't add anything. | 294:5 |
| Before "KIND" insert "the value of" because it's usually the kind of an argument, not its value, that disambiguates genericity. See remarks for [293:15] above. | 294:20 |
| Delete "or not" because it doesn't add anything. | 295:17-18 |
| At the end add "If either POINTER or TARGET is disassociated the result is false." | 295:36 |
| "points to" \Rightarrow "is associated with". | 296:13 |
| Before "KIND" insert "the value of" because it's usually the kind of an argument, not its value, that disambiguates genericity. See remarks for [293:15] above. | 297:36 |
| Before "KIND" insert "the value of" because it's usually the kind of an argument, not its value, that disambiguates genericity. See remarks for [293:15] above. | 298:27 |
| Because of the sentence at [298:23-24], the "Y is absent and" part is vacuous; Delete it. | 298:30 |
| The word "on" is confusing. Replace it by "from" or "about". | 301:31 |
| This is too nebulous to be useful. Replace it with "Returns the smallest positive model number ϵ such that in the arithmetic of the model representing numbers of the same type and kind type parameters of the argument $1 + \epsilon \neq 1$." The phrase "in the arithmetic of the model" is important, because otherwise the result could depend on the processor rounding mode. On the other hand, the result clause is specific and definitive. Maybe it's OK for the description to be fuzzy, so long as it's not contradictory. | 305:17-19 |
| Before "KIND" insert "the value of" because it's usually the kind of an argument, not its value, that disambiguates genericity. See remarks for [293:15] above. | 306:27 |
| Insert a blank after "("). | 307:3 |
| At the end, add "Inverse of ACHAR." | 309:13 |
| The word "logical" could easily be confused with the type of the same name, but that would be wrong here. "logical" \Rightarrow "bitwise". | 309:26 |
| At the end, add "Inverse of CHAR." | 310:40 |
| Before "KIND" insert "the value of" because it's usually the kind of an argument, not its value, that disambiguates genericity. See remarks for [293:15] above. | 312:13 |
| "an" \Rightarrow "a Boolean". | 312:23 |
| "IOR(5,3) has the value 7" would be a more interesting example. | 312:37 |

| | |
|---|------------|
| The word “logical” could easily be confused with the type of the same name, but that would be wrong here. “logical” \Rightarrow “bitwise”. | 312:39 |
| Delete “If ... length” because that’s already said in 6.1.3. | 314:22-23 |
| Before “KIND” insert “the value of” because it’s usually the kind of an argument, not its value, that disambiguates genericity. See remarks for [293:15] above. | 317:8 |
| What is the name of the 97’th argument? | 318:14 |
| These two paragraphs could be combined. | 319:13-28 |
| Delete “there are no such elements, (that is,” and “)”. That is, there’s no need to say it twice, or even two times. Compare to [322:29-31]. | 319:26-28 |
| See remarks for [318:14] above. | 321:14-21 |
| See remarks for [319:13-28] above. | 322:1ff |
| Before “KIND” insert “the value of” because it’s usually the kind of an argument, not its value, that disambiguates genericity. See remarks for [293:15] above. | 325:32 |
| The word “logical” could easily be confused with the type of the same name, but that would be wrong here. “logical” \Rightarrow “bitwise”. | 325:37 |
| Replace “INT ... (13.7.1)” with “INT(LOG10(HUGE(X)))”. | 330:1-3 |
| Replace “INT ... (13.7.1)” with INT(MIN(LOG10(HUGE(X)), -LOG10(TINY(X))))”. | 330:5-7 |
| Before “KIND” insert “the value of” twice because it’s usually the kind of an argument, not its value, that disambiguates genericity. See remarks for [293:15] above. | 330:18, 22 |
| “be array values” \Rightarrow “not be scalar” twice. | 331:5, 13 |
| What does “constant size” mean? It’s SIZE (or SHAPE) shall be an initialization expression? | 331:9 |
| Need to define the result in case an argument is unlimited polymorphic but disassociated or not allocated. | 332:3-4 |

2 Not simply editorial

These would be spec changes, but they would be “regularizations” in the sense of removing unnecessary restrictions. They would, however, be changes that would require implementors to “do something.” On the other hand, responding to several would consist of taking out tests. Although this has a short-term cost, it has a long-term benefit: reduction of maintenance costs.

| | |
|---|------------|
| There is no problem with MASK being scalar. Delete “It shall be scalar” at [293:24] and at [293:29] replace “one” by “less than two”. | 293:24, 29 |
| See remarks for [293:24, 29] above. | 294:28, 33 |
| See remarks for [293:24, 29] above. | 299:29, 34 |
| The shape of a scalar is defined to be a zero-size array. There should be no problem allowing the ARRAY argument to be scalar. | 313:40 |
| What is the problem with mixed kind and type? The usual rules of mixed-mode arithmetic could apply. | 318:17-20 |

| | |
|---|-----------|
| If there is no problem with size-zero or size-1 arrays, there is no problem with a scalar ARRAY argument. | 319:3 |
| <hr/> | |
| If there is no problem with size-zero or size-1 arrays, there is no problem with scalar ARRAY arguments. | 320:6 |
| <hr/> | |
| See remarks for [318:17-20] above. | 321:14-21 |
| <hr/> | |
| See remarks for [319:3] above. | 322:1ff |
| <hr/> | |
| See remarks for and [320:6] above. | 323:10 |
| <hr/> | |
| If there is no problem with size-zero or size-1 arrays, there is no problem with a scalar ARRAY argument. | 326:25 |
| <hr/> | |
| See remarks for [319:3] above. | 327:27ff |
| <hr/> | |
| Why not COMPLEX? I.e. "X shall be of numeric type." | 328:26 |
| <hr/> | |
| See remarks for [319:3] above. | 337:6ff |