Subject: Comments on Section 7
From: Van Snyder

1 Edits

Edits refer to 02-007. 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 associated text, while a page and line number followed by + (-) indicates that associated text is to be inserted after (before) the indicated line. Remarks are noted in the margin, or appear between [and] in the text.

[Editor: Insert “, pointer assignment” after “defined assignment”.] 111:3

[Editor: Indent the last line of Note 7.3.] 112:19-

[Editor: Indent the last line in each of of Notes 7.4 and 7.5.] 113:3-,18-

[Editor: Indent the last line of Note 7.6.] 114:1-

[Editor: Indent the last five lines of Note 7.7.] 114:8-

[Upside down.] 115:1-3

A **numeric intrinsic operator** is one of +, -, *, / or //. A **numeric intrinsic operation** is an intrinsic operation for which *intrinsic-operator* is a numeric intrinsic operator.

[Editor: Insert “declared” before the first “type”.] 116:31

[Editor: Insert “; the dynamic type and type parameters are those of the variable value or function result, respectively” after “respectively”.] 116:32

[Editor: Insert “declared” before the first “type”; Insert “If the pointer is associated with a target its dynamic type and type parameters are the same as the target.” before “If”.] 117:1

[Editor: Insert “, type parameters, and shape” after the second “type”; Exchange “The type of the result of a defined ... (7.3)” and “The shape ... otherwise”.] 117:14-17

[Editor: Start a new paragraph with “For an expression...” at lines 22, 24, 26, 34 and 37. 117:18-39

[Surealy we don’t allow just any kind type parameter for the result in the case of operands having the same decimal precision. Editor: “or ... precision” ⇒ “if the decimal precisions are different; if the decimal precisions are the same, the kind type parameter of the expression is the same as that of one of the operands, but it is processor dependent which one”.] 117:33-34

[Editor: “,” ⇒ “and”.] 119:20

[Editor: Insert “of” before the first “the” in the third line of Note 7.14.] 122:4+4

[Editor: Replace “Nonallowable” in the second heading in Note 7.19 with “Forbidden”, for consistency with the text two lines above it.] 123:7+a bunch

[Editor: “generic spec” ⇒ “*generic-spec*”.] 129:5

[Specific interfaces don’t have generic specs, so the declared type can’t have a corresponding specific interface. Editor: Delete “specific”.] 129:6

[After item (1), no item other than (3) bothers to say that d_2 is a dummy argument. Editor: Delete “dummy argument”.] 129:9

[Editor: Delete “specific” for the reason described for [129:6] above.]	129:20
[The specifications in 7.1.1 seem more to be specifications than implications. Editor: “implied ... form” ⇒ “, specified”; “which ⇒ “that”; Delete the comma on line 31.]	129:30-31
[Editor: Indent the fourth nonblank line of the continuation of Note 7.32.]	131:0+5
[Now that we have a syntax term index, Note 7.35 is unnecessary. I don’t think there are other notes of the same form anyway. Editor: Delete Note 7.35.]	131:12+1-2
[Editor: “12.3.2.1” ⇒ “12.3.2.1.2”.]	132:5
[The rest of the specification of defined assignment is in 7.5.1.6, but it belongs here. Editor: Move [135:4-19] to here.	132:6+
[The description of defined assignment at [132:12-13] is inadequate. This is an example of “say it twice, get it wrong at least once.” We could duplicate the description that was in 7.5.1.6, but which the edit above moved to 7.5.1.2. Better, now that defined assignment is described completely in 7.5.1.2, we can simply say “not defined assignment”:] An **intrinsic assignment statement** is an assignment statement that is not a defined assignment statement, and in which (1) The shapes of *variable* and *expr* conform, and (2) Either (a) The types of *variable* and *expr* are intrinsic, as specified in Table 7.8, or (b) The dynamic types of *variable* and *expr* are the same derived type with the same type parameter values and *variable* is not polymorphic. [Editor: Move table 7.8 to here.]	132:8-13
An *assignment-stmt* shall meet the requirements of either a defined assignment statement or an intrinsic assignment statement.	132:24+ New ¶
[Editor: Delete “specific” for the reason described for [129:6] above. Notice that this stuff is moved by a previous edit.]	135:8
[Editor: Move to [136:0+].	136:8-9
[Editor: Move to [136:0+].	136:13-18

2 Unresolved issue 335

[Editor: Insert “, x_1 and x_2 are conformable,” after “elemental”.]	129:27
[Editor: Delete unresolved issue 335 note.]	129:28+1ff

3 Comments without edits

Could be a constraint.	111:27
“the kind type parameters shall he the same” could be a constraint.	115:13-14
“and have the same kind type parameter value” could be a constraint.	115:17
Does “they” refer to declared type and type parameters, or declared and dynamic type and type parameters?	116:14
Is this true even if the function reference is evaluated?	122:10-12

Should “variable” be “*variable*”? 132:1

Is “The evaluation of expressions within *variable* shall neither affect nor be affected by the evaluation of *expr*” a requirement on the processor or the program? 133:19-20