Subject: Issue 20 and related stuff
From: Van Snyder
References: wvs-151-2

1 Discussion

In issue 20 (155:20-33), the editor asks whether kind type parameters are checkable in the presence of polymorphism. As things now stand, the answer is no.

There are five possibilities:

1. Do away with user-defined type parameters.
2. Prohibit extension types from adding any type parameters. Given the decision at the Cadarache ISO/WG5 meeting that extension types will be extended from an anonymous base type, this means that no extension type can have any type parameters.
3. Do away with polymorphism.
4. Require only that the kind type parameters of a polymorphic *pointer-object* or dummy argument shall agree with corresponding kind type parameters of the target or the associated actual argument, respectively, but the target or the associated actual argument could have more kind type parameters.
5. Put restrictions on ALLOCATE, pointer assignment, and argument association, so that all of the kind type parameters agree.

This paper proposes (at least some of) the restrictions necessary for the last two alternatives. The paper wvs-151-2 proposes edits in the same area.

2 Edits

Edits refer to 99-007r2. 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 immediately following text, while a page and line number followed by + indicates that immediately following text is to be inserted after the indicated line. Remarks for the editor are noted in the margin, or appear between [and] in the text.

2.1 Fourth alternative

[Editor: Replace "kind type ... *pointer-object*" by "the corresponding kind type parameters of *pointer-object* and the target shall have the same values".] 155:18-19

[Editor: Delete.] 155:20-33

2.2 Fifth alternative

The restrictions on pointer assignment and argument association are parallel. A restriction on ALLOCATE is necessary in order for the other two to work.

Constraint: If *type-spec* appears, it shall have no more kind type parameters than the declared type of any polymorphic *allocate-obj*. 121:6+

Constraint: If *pointer-object* is polymorphic, the declared type of the target shall have no more kind type parameters than the declared type of *pointer-object*. 155:17+

Constraint: If *pointer-object* is polymorphic, the target shall have no assumed or deferred type parameters that are not parameters of the declared type of *pointer-object*.

[Editor: Delete.] 155:20-33

Constraint: If the dummy argument with which *actual-arg* is associated is polymorphic (which requires explicit interface), the declared type of *actual-arg* shall have no more kind type parameters than the declared type of the dummy argument. 273:31+