Subject: Issue 211 et cetera

From: Malcolm Cohen

1. Introduction

Issue 211 takes issue with the terms nested form and flattened form, which are nowhere defined. On further examination, nested and flattened form of the intrinsic structure constructors cannot be made to work reliably automatically.

Since the natural form for type extension (as opposed to embedding) is flattened form, this paper proposes resolution of this issue in favour of the flattened form for the non-keyword form of the constructor. The user can produce an overload of the structure constructor, use keywords in the constructor, or use embedding instead of extension, if that is what he wants.

2. Example of the problem

```
TYPE,EXTENSIBLE :: listitem
        CLASS(listitem),POINTER :: next
END TYPE
TYPE,EXTENDS(listitem) :: real_listitem
        REAL value
END TYPE
TYPE(real_listitem),TARGET :: y
TYPE(listitem),TARGET :: x
x%next = null()
y = real_listitem(x,4.5)
        ! is this: y%next = x
        !               y%value = 4.5
        ! or this: y%listitem = x
        !               y%value = 4.5
        ! which is:
        !               y%next = x%next ! = null()
        !               y%value = 4.5
?
```

3. Specification/Syntax Change

Structure constructors for extended types use the defined "component order" for positional arguments. This omits the parent component, so constructors without "keyword arguments" are always in what we used to call "flattened form".

4. Edits to 00-007r3

[56:35-36] Replace with

"A *type-param-value* that is not of the same integer kind as the corresponding type parameter is converted according to the rules of intrinsic assignment (7.5.1.4)"

{We allow the kinds to be different but provide no semantics.}

[57:24-32] Replace text and J3 note with:

"If a structure constructor for an extended type specifies a value for a parent component, it shall not specify a value for any component that is associated with the parent component (4.5.3.1)."

{Disallow overlapping value specs.}

[57:34-] Insert new sentence at beginning of paragraph:

"Because no parent components appear in the defined component ordering, a value for a parent component may only be specified with a component name keyword."

[57:39-44] Replace examples with:

"
```
COLOR_POINT( point=point(1,2), color=3) ! Value for parent component
COLOR_POINT( point=PV, color=3)          ! Available even if TYPE(point) has private components
COLOR_POINT( 1, 2, 3)                   ! All components of TYPE(point) need to be accessible.
```
"

[58:30,32] After "`bounds`" insert "`(if any)`", once per line.

{Allocatable entities may be scalar and therefore not have bounds.}

[58:31] Replace "`With ... array`" with "`Otherwise`".

{Allocatable entities may be scalar and therefore need not be arrays.}