Date: 8th December 2000
To: J3
From: Malcolm Cohen
Subject: Altering pointers

1. Introduction

The effects of INTENT(IN) on pointers are
(a) poorly described
and
(b) maybe not as intended anyway.

There is also a question of interpretation of F95.

1.1 Desired INTENT(IN) semantics

The effects of INTENT(IN) on pointers is defined as "its association shall not be changed" ([73:4-6]). This is unfortunate wording, since if the target of a pointer goes away (e.g. it is an object that gets deallocated) the association status of the pointer changes to "undefined association status".

I believe that we intended INTENT(IN) for pointers to work the same as for nonpointers, i.e. that one cannot change the (underlying representation of) the pointer.

1.2 Constraint wording

Furthermore, the constraint against an INTENT(IN) pointer appearing in an inappropriate context is strangely worded: it only applies to "A pointer dummy argument", so it seems ok to let pointer components of INTENT(IN) dummies appear in NULLIFY statements etc.! I don't think so.

E.g.
```
  TYPE T
    REAL,POINTER :: X(:)
  END TYPE
  ...
  SUBROUTINE S(D)
    TYPE(T),INTENT(IN) :: D
    NULLIFY(D%X)
```

The first constraint does not help because D%X is not a "nonpointer object with the INTENT(IN) attribute", and the "subobject of" text from F95 has been removed.

We also need to fix this constraint to make the effects of INTENT(INOUT) for pointers the same as for nonpointers.

1.3 A random comment

If we had the pointer-equivalent of our nonpointer terms "define" and "definable" (say, "alter" and "alterable") it would be easier to describe what

we mean. However, this paper attempts to fix the problems without introducing new terminology.

1.4 F95 interpretation

Consider

```
  TYPE T
    REAL,POINTER :: X(:)
  END TYPE
  ...
  SUBROUTINE S(D)
    TYPE(T),INTENT(IN) :: D
    REAL,POINTER :: Q(:)
    Q => D%X
    DEALLOCATE(Q)
```

Is the DEALLOCATE standard-conforming?
(This paper makes the answer clearly Yes in F2002.)

2. Edits to 00-007r3

[72:40] Change "dummy argument" to "object".
{Constrain direct pointer components of intent(in) dummies.}

[73:5-6] Change "its association ... procedure" to:
"during the execution of the procedure its association status shall not be changed except that it may become undefined if the target is deallocated other than through the pointer (14.6.2.1.3)"
{Allow intent(in) pointers and pointer components of intent(in) objects to become undefined as a side-effect of their target disappearing.}