WG 14 N2203

TS 18661 DR 17

==

Reference Document: TS 18661-3

Subject: incommensurate arguments for comparison macros

Summary

This DR addresses a problem noted by Joseph Myers in email SC22WG14.14885:

> The usual arithmetic conversions in TS 18661-3 include "If both operands have floating types and neither of the sets of values of their corresponding real types is a subset of (or equivalent to) the other, the behavior is undefined.".
>
> Thus, for example, if neither of long double and _Float128 has a set of values that is a subset of the other, given
>
> long double a;
> _Float128 b;
>
> it's undefined to have the expression "a < b".
>
> Now what about the expression "isless (a, b)"? By analogy with the direct comparison, it would seem natural for it to be undefined. But while 18661-2 explicitly disallows using those macros with one decimal and one non-decimal argument, I see nothing to disallow the case where neither set of values is a subset of the other, and the definition of these macros doesn't actually include the usual arithmetic conversions.

It was an oversight to not disallow argument types neither of which is a subset (or equivalent to) the other.

Suggested Technical Corrigendum

In TS 18661-3, at the end of clause 12 (just before 12.1), insert:

> To 7.12.14#1, append:
>
> > If neither of the sets of values of the argument formats is a subset of (or equivalent to) the other, the behavior is undefined.