Document: N1513
Date: 2010/09/22
Author: Jim Thomas and Fred Tydeman
Subject: Followup on N1181

N1181 proposed text that was more specific about requirements for math functions with respect to exactness and the rounding direction mode. At the 2006 Portland meeting, the proposed changes were approved, though only for Annex F.

This is a proposal to address a few items (in Annex F) that were missed or for which the change was incorrect.

Item 1. The following was missed:

F.10.3.13 The scalbn and scalbln functions: Append paragraph

> If the calculation does not overflow or underflow, the returned value is exact and independent of the current rounding direction mode.

Item 2. The following two are currently incorrect because ceil and floor are allowed, but not required to, raise the inexact floating-point exception (as reflected in the sample implementation):

F.10.6.1 The ceil functions #2: Change

> The returned value is exact and independent of the current rounding direction mode.

to

> The returned value is independent of the current rounding direction mode.

Add a paragraph following the example:

> The ceil functions may, but are not required to, raise the “inexact” floating-point exception for finite non-integer arguments, as this implementation does.

F.10.6.2 The floor functions #2: Change

> The returned value is exact and independent of the current rounding direction mode.

to

> The returned value is independent of the current rounding direction mode.

Append to #3

> The floor functions may, but are not required to, raise the “inexact” floating-point exception for finite non-integer arguments, as this implementation does.

Item 3. The following was missed.

F.10.6.6 The round function: Insert after #1

> The returned value is independent of the current rounding direction mode.

Item 4. The following is currently incorrect because “numeric” includes infinities, which do not cause the inexact floating-point exception.

F.10.6.6 The round functions #2: In the sentence following the example change

> The round functions may, but are not required to, raise the ‘‘inexact’’ floating-point exception for non-integer numeric arguments, as this implementation does.

to

> The round functions may, but are not required to, raise the ‘‘inexact’’ floating-point exception for finite non-integer arguments, as this implementation does.

Item 5. The following was missed.

F.10.6.8 The trunc functions: Append a paragraph

> The returned value is independent of the current rounding direction mode. The trunc functions may, but are not required to, raise the “inexact” floating-point exception for finite non-integer arguments.

Item 6. The following is another case of the aforementioned misuse of “numeric” (unrelated to N1181). Note correct use in 5.2.4.2.2 #3.

5.2.4.2.2 #4: Change

> An implementation may give zero and non-numeric values (such as infinities and NaNs) …

to

> An implementation may give zero and values that are not floating-point numbers (such as infinities and NaNs) …