Proposal for C23
WG14 N2879

Title: 5.2.4.2.2 cleanup (N2806 update)
Author, affiliation: C FP group
Date: 2021-12-23
Proposal category: Editorial
Reference: N2596, N2672, N2806

This document updates N2806 (approved at the November 2021 WG 14 meeting) to further clarify the definitions and requirements in 5.2.4.2.2. It includes the suggestion from Joseph Myers to place definitions in separate paragraphs. It also broadens footnote 22 to cover extreme-magnitude numbers represented with reduced precision, as occur in some double-double implementations.

Suggested changes (change marks relative to N2596)**:**

NOTE At the end we show the new changes suggested by this document relative to N2596 + N2806.

Changes for 5.2.4.2.2:

> [1] The characteristics of floating types are defined in terms of a model that describes a representation of floating-point numbers and allows other values. The characteristics ~~that~~ provide information about an implementation's floating-point arithmetic.21)
> ...
>
> [3a] Model floating-point numbers x with $f_1 > 0$ are called *normalized floating-point numbers*.
>
> [3b] Model floating-point numbers $x \neq 0$ with $f_1 = 0$ and $e = e_{\min}$ are called *subnormal floating-point numbers*.
>
> [3c] Model floating-point numbers $x \neq 0$ with $f_1 = 0$ and $e > e_{\min}$ are called *unnormalized floating-point numbers*.
>
> [3d] Model floating-point numbers x with all $f_k = 0$ are zeros.
>
> [4] Floating types shall be able to represent signed zeros or an unsigned zero ~~(all f_k == 0)~~ and all normalized floating-point numbers ~~($f_1 > 0$ and all possible k digits and e exponents result in values representable in the type)~~. In addition, floating types may be able to contain other kinds of floating-point numbers,22) such as ~~negative zero,~~ subnormal floating-point numbers ~~($x \neq 0$, $e = e_{min}$, $f_1 = 0$)~~ and unnormalized floating-point numbers ~~($x \neq 0$, $e > e_{min}$, $f_1 =$~~

~~0)~~, and values that are not floating-point numbers, such as ~~infinities and NaNs~~ NaNs and (signed or unsigned) infinities. ...

[5] ~~An implementation may give zero and values that are not floating-point numbers (such as infinities and NaNs) a sign or may leave them unsigned.~~ Wherever ~~such~~ values are unsigned, any requirement in this document to ~~retrieve~~ get the sign shall produce an unspecified sign, and any requirement to set the sign shall be ignored, unless specified otherwise.*)

22) Some implementations have types that include finite numbers with ~~extra~~ range and/or precision that are not covered by the model.

*) Bit representations of floating-point values might include a sign bit, even if the value can be regarded as unsigned. IEC 60559 NaNs are such values.

Suggested changes (change marks relative to N2596 + N2806)**:**

[1] The characteristics of floating types are defined in terms of a model that describes a representation of floating-point numbers and allows other values. The characteristics provide information about an implementation's floating-point arithmetic.21)

...

[3a] Model floating-point numbers x with $f_1 > 0$ are called *normalized floating-point numbers*.

[3b] Model floating-point numbers $x \neq 0$ with $f_1 = 0$ and $e = e_{\min}$ are called *subnormal floating-point numbers*.

[3c] Model floating-point numbers $x \neq 0$ with $f_1 = 0$ and $e > e_{\min}$ are called *unnormalized floating-point numbers*.

[3d] Model floating-point numbers x with all $f_k = 0$ are zeros.

[4] Floating types shall be able to represent signed zeros or an unsigned zero ~~(all $f_k = 0$)~~ and all normalized floating-point numbers ~~(all x with $f_1 > 0$)~~. In addition, floating types may be able to contain other kinds of floating-point numbers,22) such as subnormal floating-point numbers ~~($x \neq 0$, $e = e_{min}$, $f_1 = 0$)~~ and unnormalized floating-point numbers ~~($x \neq 0$, $e > e_{min}$, $f_1 = 0$)~~, and values that are not floating-point numbers, such as NaNs and (signed or unsigned) infinities. ...

[5] Wherever values are unsigned, any requirement in this document to get the sign shall produce an unspecified sign, and any requirement to set the sign shall be ignored, unless specified otherwise.*)

22) Some implementations have types that include finite numbers with ~~extra~~ range and/or precision that are not covered by the model.

*) Bit representations of floating-point values might include a sign bit, even if the value can be regarded as unsigned. IEC 60559 NaNs are such values.