Proposal for C2Y
WG14 n3404

Title:	Improved treatment of error conditions for functions that round result to narrow type
Author, affiliation:	CFP group
Date:	2024-11-16
Proposal category:	Editorial
Reference:	N3301

This document proposes clarification of the language and unification of the requirements around some errors that arise in functions that perform arithmetic and deliver results to a narrower floating type.

Suggested change:

fadd – change to the last sentence of 7.12.15.2#2

> A domain error may occur for ~~infinite arguments~~ both arguments infinite and with opposite sign.

fsub – change to the last sentence of 7.12.15.3#2

> A domain error may occur for ~~infinite arguments~~ both arguments infinite and either with the same sign or unsigned.

fmul – change to the last sentence of 7.12.15.4#2

> A domain error ~~occurs~~ may occur for one infinite argument and one zero argument.

fdiv – change to the second to the last sentence of 7.12.15.5#2

> A domain error ~~occurs~~ may occur for either both arguments infinite or both arguments zero.

fma – change to the last sentence of 7.12.15.6#2

> A domain error may occur ~~for an infinite argument~~ if one of x and y is zero and the other is infinite. A domain error may occur if z is infinite and one of x and y is infinite while the other is nonzero finite or infinite, such that the product of x and y has the opposite sign of z.

Rationale:

This change makes explicit which cases may cause domain errors in fadd, fsub, and fma. The case of fsub is expanded to include unsigned (projective) infinity, should it be implemented, in the one case of magnitude subtraction of unsigned infinite values.

The change also brings the reporting requirements of all functions in 7.12.15 into harmony by weakening the specification for fmul, and fdiv from "error occurs" to "error may occur". The alternative suggestion, to strengthen all reporting to "error occurs", could break some existing implementations.

This change requires no implementation changes.