Affinely extended reals with signed zero and NaN

Bill Zorn

August 29, 2017

Abstract

We present a theory for the ideal arithmetic that underlies IEEE 754 floating point. This theory is based on "typical" real arithmetic over $\mathbb{R}$; it is affinely extended to include $-\infty$ and ∞; it represents a recoverable sign for all numbers, including 0; and it adds a special member NaN to represent values that are not a real number in the typical sense. It serves as a bridge between real arithmetic in a mathematical sense and specifications and implementaions of floating point formats, particularly IEEE 754. The theory can also specify ideal, infinite-precision results of floating point computations.

Work In Progress:
This vessel is not yet seaworthy.

1 Notation

We denote the set of mathematical real numbers as $\mathbb{R}$. The affinely extended real numbers with signed zero and NaN, the set $\mathbb{R} \cup \{-\infty, \infty, -0, \text{NaN}\}$, we denote $\mathbb{R}^*$.

Numbers in $\mathbb{R}$ we write as simple variables, a, b, x, y, while numbers in $\mathbb{R}^*$ we distinguish with an overline, $\overline{a}, \overline{b}, \overline{x}, \overline{y}$. A written a cannot be -0 or ∞, while a written $\overline{x}$ can.

We distinguish arithmetic operations in the same way we distinguish numbers. $a + b$ denotes typical addition of real numbers, while $\overline{x} \mathbin{\overline{+}} \overline{y}$ denotes addition under the arithmetic of extended real numbers with signed zero and NaN, which we define in this document. $a \mathbin{\overline{+}} y$ is well-defined (and hopefully not too different from typical addition), but $\overline{x} + \overline{y}$ is not, as the operands might not be real numbers.

2 Ordering

$\mathbb{R}^*$ is only partially ordered. For all numbers in $\mathbb{R}^*$ that are also in $\mathbb{R}$, the typical rules for comparison of real numbers apply. The following rules define

the ordering of the special members $-\infty$, ∞, and -0:

$$-\infty < \overline{x} \in \mathbb{R}^* \setminus \{-\infty, \mathrm{NaN}\} \tag{1}$$

$$\overline{x} \in \mathbb{R}^* \setminus \{\infty, \mathrm{NaN}\} < \infty \tag{2}$$

$$-0 = 0 \tag{3}$$

$$a < 0 \iff a < -0 \tag{4}$$

$$0 < a \iff -0 < a \tag{5}$$

Equality of members in $\mathbb{R}^* \setminus \{\mathrm{NaN}\}$ is reflexive, symmetric, and transitive, as is to be expected, and the comparisons $<$ and $>$ follow the usual symmetries. $-\infty = -\infty$, $a \in \mathbb{R} > 0 \iff a > -0$, and so on.

NaNis unordered with respect to all other elements of $\mathbb{R}^*$. The following rules define comparisons with it:

$$\mathrm{NaN} \not< \overline{x} \tag{6}$$

$$\mathrm{NaN} \not> \overline{x} \tag{7}$$

$$\mathrm{NaN} \neq \overline{x} \tag{8}$$

Note that equality is not reflexive ($\mathrm{NaN} \neq \mathrm{NaN}$), and that the usual symmetry of $<$ and $>$ is broken ($\overline{x} \not< \mathrm{NaN} \iff \overline{x} > \mathrm{NaN}$.

Because $\mathbb{R}^*$ is only partially ordered, it is useful to define another notion of identicallity, $\overline{x} \equiv \overline{y}$, which captures when two members of $\mathbb{R}^*$ are actually the same element. $\overline{x} \equiv \overline{x}$ with no exceptions, even when $\overline{x} \equiv \mathrm{NaN}$, and $0 \not\equiv -0$ even though $0 = -0$.

3 Primitives: negation, absolute value, and sign

All numbers in $\mathbb{R}^*$ have a recoverable sign, except for NaN, which can be thought of as not a number. Unary negation flips this sign, as follows:

$$neg(-\infty) \equiv \infty \tag{9}$$

$$neg(\infty) \equiv -\infty \tag{10}$$

$$neg(-0) \equiv 0 \tag{11}$$

$$neg(0) \equiv -0 \tag{12}$$

$$neg(a \in \mathbb{R} \setminus \{0\}) \equiv -a \tag{13}$$

$$neg(\mathrm{NaN}) \equiv \mathrm{NaN} \tag{14}$$

Absolute value forces this sign to be positive:

$$abs(-\infty) \equiv \infty \tag{15}$$

$$abs(\infty) \equiv \infty \tag{16}$$

$$abs(-0) \equiv 0 \tag{17}$$

$$abs(a) \equiv |a| \tag{18}$$

$$abs(\mathrm{NaN}) \equiv \mathrm{NaN} \tag{19}$$

We also define a sign operation to recover the sign, multiplied by 1:

$$sign(-\infty) = -1 \tag{20}$$
$$sign(\infty) = 1 \tag{21}$$
$$sign(-0) = -1 \tag{22}$$
$$sign(a) = \begin{cases} -1, a < 0 \\ 1, a = 0 \vee 0 < a \end{cases} \tag{23}$$
$$sign(\text{NaN}) = -1 \,[]\, 1 \tag{24}$$

$-1 \,[]\, 1$ denotes a nondeterministic choice between -1 and 1. The sign of NaN is not undefinied: it must be -1 or 1, though this theory does not specify which one of those it is.

Finally, we can transfer the sign from one number to another:

$$copysign(\overline{x}, \overline{y}) \equiv \overline{z} \iff (abs(\overline{x}) \equiv abs(\overline{z}) \wedge sign(\overline{y}) = sign(\overline{z})) \tag{25}$$

This definition runs into some trouble with NaN. To be faithful to the IEEE 754 standard, Equation 25 must always be satisfied, even for NaN, so the nondeterministic choice must always be made in a coherent way. This could be accomplished simply by choosing a single sign for NaN (say $sign(NaN) = 1$), or by tracking each NaN in a computation and assigning some specific sign to it.

4 Addition and subtraction

All arithmetic operations in $\mathbb{R}^*$ are closed, with a defined result in $\mathbb{R}^*$. Addition is symmetric:

$$\overline{x} \mathbin{\overline{+}} \overline{y} \equiv \overline{y} \mathbin{\overline{+}} \overline{x} \tag{26}$$

We take advantage of this to simplify the rules somewhat:

$$\text{NaN} \mathbin{\overline{+}} \overline{x} \equiv \text{NaN} \tag{27}$$
$$\infty \mathbin{\overline{+}} -\infty \equiv \text{NaN} \tag{28}$$
$$\infty \mathbin{\overline{+}} \overline{x} \in \mathbb{R}^* \setminus \{-\infty, \text{NaN}\} \equiv \infty \tag{29}$$
$$-\infty \mathbin{\overline{+}} \overline{x} \in \mathbb{R}^* \setminus \{\infty, \text{NaN}\} \equiv -\infty \tag{30}$$
$$-0 \mathbin{\overline{+}} \overline{x} \in \mathbb{R}^* \setminus \{0\} \equiv \overline{x} \tag{31}$$
$$-0 \mathbin{\overline{+}} 0 \equiv \begin{cases} -0, \text{roundTowardNegative} \\ 0, \neg\text{roundTowardNegative} \end{cases} \tag{32}$$
$$a \mathbin{\overline{+}} b \equiv \begin{cases} a + b, a + b \neq 0 \\ -0, a + b = 0 \wedge \text{roundTowardNegative} \\ 0, a + b = 0 \wedge \neg\text{roundTowardNegative} \end{cases} \tag{33}$$

roundTowardNegative is some arbitrary predicate representing the IEEE 754 rounding mode. If it is True, then addition producing 0 should produce -0 instead, in accordance with the IEEE specification for arithmetic with that rounding mode. It can be thought of as a parameter of this theory, or as a distinction between two similar theories.

Subtraction is defined in terms of addition and negation:

$$\overline{x} \mathbin{\overline{-}} \overline{y} \equiv \overline{x} \mathbin{\overline{+}} neg(\overline{y}) \tag{34}$$

5 Multiplication

Like addition, multiplication over $\mathbb{R}^*$ is symmetric:

$$\overline{x} \mathbin{\overline{\times}} \overline{y} \equiv \overline{y} \mathbin{\overline{\times}} \overline{x} \tag{35}$$

It is defined as follows:

$$\text{NaN} \mathbin{\overline{\times}} \overline{x} \equiv \text{NaN} \tag{36}$$

$$\overline{x} \in \{-0, 0\} \mathbin{\overline{\times}} \overline{y} \in \{-\infty, \infty\} \equiv \text{NaN} \tag{37}$$

$$\overline{x} \in \{-\infty, \infty\} \mathbin{\overline{\times}} \overline{y} \in \mathbb{R}^* \setminus \{-0, 0, NaN\} \equiv copysign(\infty, sign(\overline{x}) \times sign(\overline{y})) \tag{38}$$

$$\overline{x} \in \{-0, 0\} \mathbin{\overline{\times}} \overline{y} \in \mathbb{R}^* \setminus \{-\infty, \infty, NaN\} \equiv copysign(0, sign(\overline{x}) \times sign(\overline{y})) \tag{39}$$

$$a \mathbin{\overline{\times}} b \equiv a \times b \tag{40}$$

Where it produces an answer other than NaN, multiplication effectively computes the sign and magnitude of the result separately; hence the use of *copysign*. This dependence could be removed by rewriting Equations 38 and 39 as cases on the signs of the operands, though this would be much more verbose.

6 Division

Division is similar to multiplication, but it cannot be defined in terms of it. We cannot simply state some equality like $\overline{x} \mathbin{\overline{\div}} \overline{y} \equiv \overline{x} \mathbin{\overline{\times}} reciprocol(\overline{y})$. Instead, we

define it as follows:

$$\text{NaN} \mathbin{\overline{\div}} \overline{x} \equiv \text{NaN} \tag{41}$$

$$\overline{x} \mathbin{\overline{\div}} \text{NaN} \equiv \text{NaN} \tag{42}$$

$$\overline{x} \in \{-\infty, \infty\} \mathbin{\overline{\div}} \overline{y} \in \{-\infty, \infty\} \equiv \text{NaN} \tag{43}$$

$$\overline{x} \in \{-0, 0\} \mathbin{\overline{\div}} \overline{y} \in \{-0, 0\} \equiv \text{NaN} \tag{44}$$

$$\overline{x} \in \{-\infty, \infty\} \mathbin{\overline{\div}} \overline{y} \in \mathbb{R}^* \setminus \{-\infty, \infty, NaN\} \equiv copysign(\infty, sign(\overline{x}) \times sign(\overline{y})) \tag{45}$$

$$\overline{x} \in \mathbb{R}^* \setminus \{-\infty, \infty, NaN\} \mathbin{\overline{\div}} \overline{y} \in \{-\infty, \infty\} \equiv copysign(0, sign(\overline{x}) \times sign(\overline{y})) \tag{46}$$

$$\overline{x} \in \{-0, 0\} \mathbin{\overline{\div}} \overline{y} \in \mathbb{R}^* \setminus \{-0, 0, NaN\} \equiv copysign(0, sign(\overline{x}) \times sign(\overline{y})) \tag{47}$$

$$\overline{x} \in \mathbb{R}^* \setminus \{-0, 0, NaN\} \mathbin{\overline{\div}} \overline{y} \in \{-0, 0\} \equiv copysign(\infty, sign(\overline{x}) \times sign(\overline{y})) \tag{48}$$

$$a \mathbin{\overline{\div}} b \equiv a \div b \tag{49}$$

Unsurprisingly, division over $\mathbb{R}^*$ is not symmetric. Independent sign and magnitude computations occur, as with multiplication, in Equations 45 - 48.

Division can be used to compute something like the reciprocol, but there isn't a closed reciprocol in $\mathbb{R}^*$: $1 \mathbin{\overline{\div}} \infty \equiv 0$, but $\infty \mathbin{\overline{\times}} 0 \equiv \text{NaN}$. In fact, it's impossible to find any $\overline{x} \in \mathbb{R}^*$ such that $\infty \mathbin{\overline{\times}} \overline{x} \equiv 1$ or $0 \mathbin{\overline{\times}} \overline{x} \equiv 1$.

7 Integer powers

An integer power function is necessary to compute the real values of floating point numbers. The implementation should be equivalent to iterated multiplication (and optionally division for negative powers).

More details coming soon!

8 Integer roots

An integer root function is useful to complete the mandated "single-ulp" correctly rounded operations in the IEEE 754 standard, + - * / and sqrt. This has not yet been implemented for Titanic, as it is not needed to convert between floating point and real numbers.

More details coming soon!