Dividing by Zero

What a Wheel Is, and Why /0 Becomes a Real Element

ZP Companion | The Wheel of Fractions | Initial: June 2026 | Current: 1.3, July 2026

This companion explains in plain language the ZP-J Wheel Addendum: the machine-verified proof that the "wheel of fractions" is a wheel - an algebraic structure (Carlström, 2001) in which dividing by zero is a defined operation rather than an error. Every formal claim here restates a theorem already proved in the technical document and its Lean source (Wheel.lean, WheelFrac.lean). Analogies are included to build intuition, not as proof claims. Consult the addendum for the authoritative mathematics.

The Problem With Dividing by Zero

In ordinary arithmetic, $1/0$ is undefined. You cannot simply declare it equal to some number: every attempt breaks one of the laws of arithmetic. If $1/0 = \infty$ and you insist on the usual rules, you quickly derive contradictions like $1 = 2$. So mathematics normally leaves division by zero out, and "/" stays a partial operation - defined for every input except one.

A wheel is a structure that makes division total instead. Every element, including 0, gets a reciprocal $/x$, so $/0$ becomes a first-class member of the system rather than an error. The trick is not to force $/0$ to be an ordinary number - it is to add new elements and adjust the laws so that nothing breaks. The name comes from the shape of the smallest example: a circle of values with the two new elements sitting on it.

The Two New Elements: ∞ and ⊥

Making division total introduces exactly two new elements:

$\infty = /0$ (the reciprocal of zero). This is the "point at infinity" you may have met on the Riemann sphere, where 0 and ∞ sit at opposite poles. A wheel goes one step further than the sphere - it adds the second element $\perp$ below as well - and it makes ∞ a genuine element you can compute with.

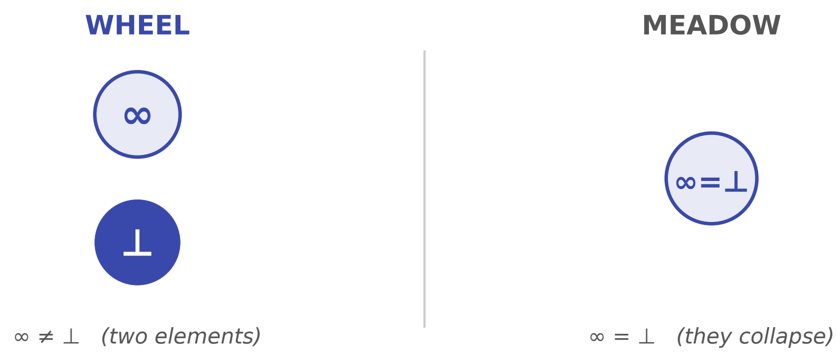
$\perp = 0{\cdot}/0$ (zero times the reciprocal of zero). This is an absorbing "undefined" element: combine it with anything and you get $\perp$ back. It is the wheel's honest answer to a genuinely meaningless expression - instead of crashing, the arithmetic returns a dedicated value that means "no information." (The wheel writes this element $\perp$, the same symbol the Zero Paradox uses for its bottom element - see the last section for why that is not a coincidence.)

Real-world analogy - The calculator that never crashes

A pocket calculator shows "Error" when you divide by zero, and then you have to clear it before you can do anything else. A wheel is like a calculator that, instead of locking up, returns a special "undefined" reading - and lets you keep computing. Any further operation on that reading just returns "undefined" again. Nothing is lost and nothing breaks; the error is absorbed into a value the system knows how to carry.

Wheel or Meadow? The Key Question

There is one decisive question about any such structure: are ∞ and $\perp$ two different elements, or do they turn out to be the same? Both answers give a consistent algebra, and they have different names:



A wheel keeps the reciprocal of zero (∞) and the absorbing undefined element ($\perp$) distinct. A meadow is the variant in which they collapse into a single element. The whole content of "wheel, not meadow" is that these two stay apart.

Building It: Fractions as Pairs

The construction at the heart of the addendum is the wheel of fractions, written $\odot_S A$. It generalises the way you build fractions from whole numbers, but it is designed to survive division by zero. Start with any commutative ring A (a number system with + and ×, like the integers) and a set S of "allowed denominators" closed under multiplication. Write each fraction as a pair (x, y), meaning x/y.

Reciprocal is just swapping the pair: $/(x, y) = (y, x)$. That single move is what makes division total - and it is why /0 exists. The fraction 0 is (0, 1), so /0 swaps to $(1, 0) = \infty$, and $0 \cdot /0$ works out to $(0, 0) = \perp$. Nothing is forbidden; the pair (1, 0) is a perfectly good fraction in this system.

There is one genuine subtlety: when are two fractions equal? The schoolbook rule - cross-multiply, $x \cdot y' = x' \cdot y$ - does not work on a general number system, because it fails to be transitive without a cancellation law (you cannot always divide out a common factor). The wheel of fractions fixes this by requiring a witness from S for every equality: two fractions count as equal only when elements of S certify it. Checking that this repaired notion of equality behaves - and that +, ×, and / are well defined on it - is the bulk of the formal proof.

The Main Result

The headline theorem is that this construction always works. For any commutative ring A and any choice of allowed denominators S, the wheel of fractions $\odot_S A$ satisfies every wheel axiom - all eight of Carlström's Definition 1.1 (checked in Lean as 14 separate equational fields). It is a wheel, with no exceptions and no extra hypotheses.

WheelFrac.instWheel (machine-verified, Lean 4)

For every commutative ring A and every multiplicative submonoid S, the wheel of fractions $\odot_S A$ is a wheel - it satisfies all of Carlström's axioms. The proof is sorry-free and free of the axiom of choice: its only foundations are propositional extensionality and quotient soundness ([propext, Quot.sound]), both standard in Lean 4.

Wheel, Not Meadow

Being a wheel is only half the story. The other half is that this construction is a genuine wheel - the one where ∞ and $\perp$ stay distinct - and not a meadow in disguise. The addendum proves exactly when that holds: as long as 0 is not one of the allowed denominators ($0 \notin S$), the reciprocal of zero and the absorbing element are different.

That condition is the normal situation: any sensible set of denominators excludes 0. The nonzero elements of an integral domain, or the complement of a prime ideal (the usual choice when building fractions), all avoid 0 - so the two elements stay apart. You only collapse to a meadow if you deliberately allow 0 as a denominator.

WheelFrac.inf_ne_bot (machine-verified, Lean 4)

If $0 \notin S$, then $\infty \neq \perp$ in $\odot_S A$. The construction is a genuine wheel, not a meadow. (If ∞ and $\perp$ were equal, the witnessing elements of S would force 0 itself into S, contradicting the hypothesis.) Also sorry-free and choice-free: [propext, Quot.sound].

Why This Belongs in the Zero Paradox

The Zero Paradox studies one structural fact about the bottom element $\perp$, written in several mathematical languages at once: in the 2-adic valuation it is $v_2(0) = \infty$, and in non-well-founded set theory it is the Quine atom $\perp = \{\perp\}$. The wheel of fractions is the algebraic face of the same point. Where ZP-J says "the bottom is where measure runs to infinity," the wheel says "the bottom is where division by zero lives" - and it shows that this point is a defined, well-behaved element ($\infty = /0$), distinct - in the standard case where $0 \notin S$ - from the absorbing $\perp$. Same location, different vocabulary.

One honest limitation. The wheel is built on top of a ring you supply - the ring structure is an input, not something derived from the Zero Paradox's own axioms. Whether the defining property of the porthole can be derived from upstream ZP structure, rather than assumed, is an open question flagged in the Lean source (the bridge typeclass WheelValuationStructure states it, but its key condition is an assumed axiom). The wheel shows the algebraic face exists and is consistent; it does not claim to force it into being.

What This Is, and What It Is Not

The wheel of fractions and the theorem that it is a wheel are Carlström's, not new mathematics. What the Zero Paradox adds is a faithful, machine-checked encoding - readable field by field against Carlström's Definition 1.1, with a verified axiom footprint showing it needs no axiom of choice - placed in the context of the bottom element $\perp$. The takeaway: division by zero is not an error to be avoided but a defined element to be understood, and at exactly the point the rest of the framework already identifies as the bottom.