Hyperreal Progression Circle

by Sven Nilsen, 2020

In this paper I present a natural way of mapping hyperreal progressions to a circle.

Hyperreal, or nonstandard reals, ${}^*\mathbb{R}$, is a way of treating infinite and infinitesimal quantities.

For standard reals, the number `1` can be written in two ways:

0.9999999… = 1.0000000…

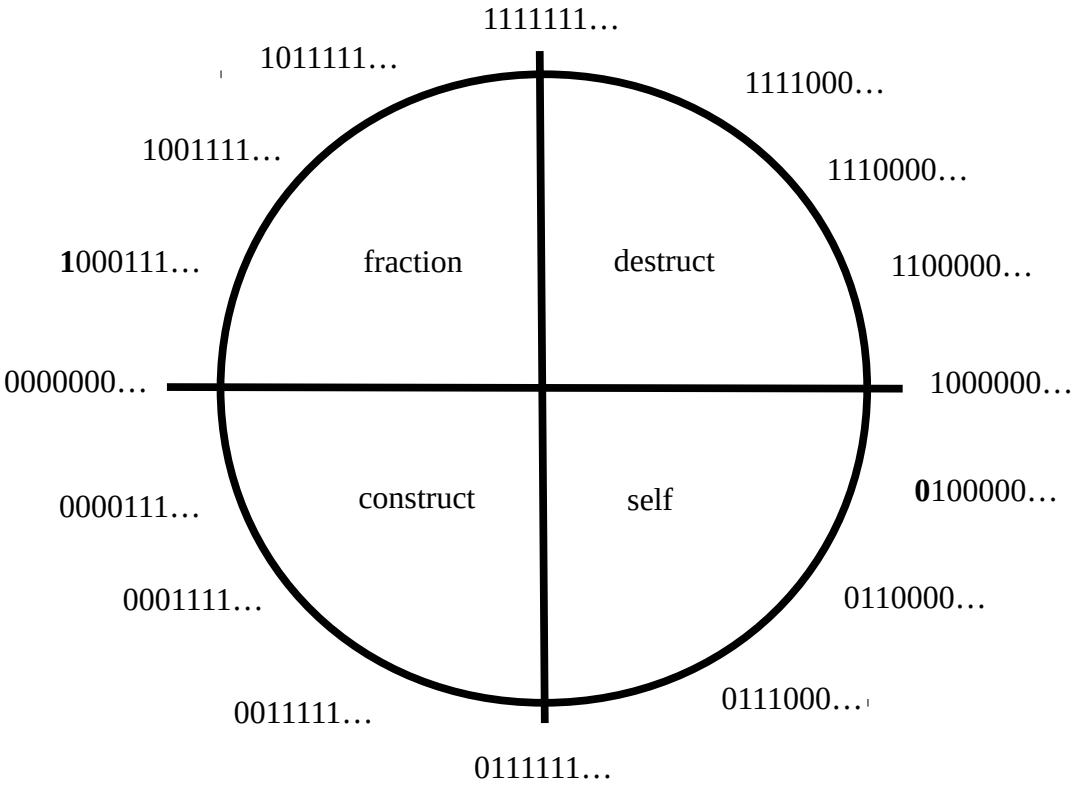
In binary base this becomes (used in the rest of this paper):

0.1111111… = 1.0000000…

A hyperreal number has the ability to distinguish between two such values.

The problem is that in some sense, these numbers are similar, but at the same time not the same.
To distinguish these numbers, one must look at the progression of constructing one number from another using sequences. These operations are hyper constructions, because they require an infinite number of steps to complete. However, since progression alone is sufficient, I call them “progressions”.

There is a natural way of thinking about such hyperreal numbers by mapping progressions to a circle:



I divide the progressions into four kinds:

- construct
- destruct
- self
- fraction

The two most intuitive progressions are:

- construct
- destruct

When you construct, you can imagine an infinite sequence of zeroes, flipped to one, one at a time:

0000000…0000000
0000000…0000001
0000000…0000011
…
0011111…1111111
0111111…1111111
1111111…1111111

Construct starts with the least significant bits and progresses toward the most significant bits.
Destruct is very similar to construct, but flipping off bits in order from significant to insignificant.

For construct and destruct, when the last bit is flipped, one jumps 180 degrees across the circle.

The intuition of jumping 180 degrees across the circle can be generalized.
One can leap any time in the middle of construct and destruct by flipping all bits at once:

0000000…0000011	doing construct twice
1111111…1111100	flip all bits to take a shortcut

In just 3 steps, consisting of 2 constructions and 1 jump, one has arrived at a number which would otherwise require an infinite number of steps in the destruct progression.

Notice that the direction of such proofs does not matter. This is why progressions do not have direction.

The self progression is a way of creating a hyperreal number from its dual representation in reals:

1.000000…0000000 = 0.111111…1111111 for standard reals

1000000…0000000	
0100000…0000000	insert zero at the beginning
0110000…0000000	flip bits…
…	
0111111…1111110	just one more bit to flip…
0111111…1111111	arrived at dual representation for standard reals

The `.` sign that separates fractions from integers is a way of interpreting an infinite sequence of bits. It means that somewhere in the infinite sequence of bit, there is a separator symbol.

The separator symbol has no proof relevance, which means that if two representations are treated as equivalent, then any interpretation with a separator symbol is also equivalent:

1000000…0000000 ~= 0111111…1111111

<=>

1.000000…0000000 ~= 0.111111…1111111

The intuition that hyper reals represents numbers, or values, is somewhat misleading because hyperreals are not limited to the semantics of numbers in ordinary sense. It is much easier to think of them as abstract infinite sequences of bits. Proofs on the hyperreals are computer programs that can finish given infinite amount of time, in the sense that no counter-proof will be found in a finite time.

One can create a hyperreal number such that it requires completing a hyper task before finding a counter-proof, but this is cheating and not sound, because a finite counter-proof exists. The proof that you can construct a such number is a proof that the hyperreal progression circle permits a finite counter-proof. Only if no finite counter-proof using all possible proofs, then the proof is sound.

I showed previously that there are finite proofs using some operations that otherwise would take an infinite amount of time. The hyperreal progression circle is a way of describing which operations a computer can do that are “shortcuts” in the space of proofs on hyperreals (creativity is the limit).

Just like destruct is construct where all bits are flipped, the fraction progression is self where all bits are flipped. The name “fraction” comes from the dual representation of reals, where:

0.1111111… = 1.0000000… for standard reals

If you look at the fractions only, then:

1111111… = 0000000… mod 1 for standard reals

This is literally true for integers, since `mod 1` only has `0`, but it also hold for fractions.

Another symmetry of the hyperreal progression circle:
Flipping the most signitificant bit gives the vertical mirror image.

A clever way of using the hyperreal progression circle to create more hyperreals,
is to associate the order of e.g. `1`s as a mask for operations on hyperreals.

For example:

- Flip every bit in a hyperreal mask: `0 = no flip`, `1 = flip`
- Shift bits accordingly to a hyperreal mask: `abc…` and `100011…` becomes `a000bc…`.
- Join two hyperreal numbers using a mask: `0 = first`, `1 = second`