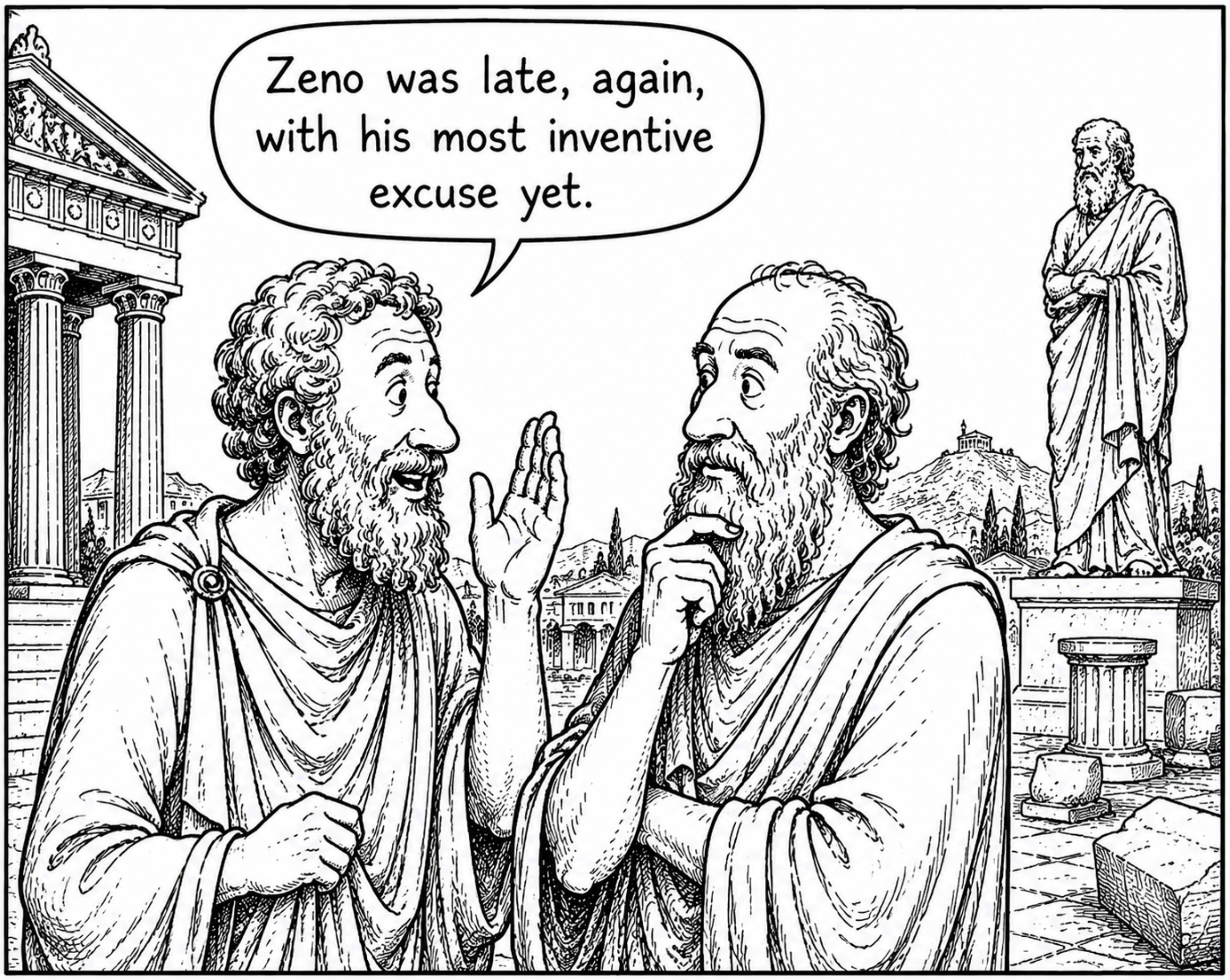
Chapter 27
“Infinity”



Zeno

Infinity has bollixed and perplexed humans for as long as we have historical records. That *which never ends* seems both daunting and impossible, yet somehow inevitable. It has been the source of much fascination, error, controversy and advancement as any other abstract concept. Let’s apply the paradigm of paradox to the challenge.

Before diving in, however, a direct line to the reader. If you have gotten this far, I consider you an Intrepid Reader (my target demographic). This is not easy stuff. If this chapter feels difficult, it’s because where infinity meets self-reference requires all the tools we’ve developed so far to be used at once. You got this.

The distinction that follows is not merely philosophical; it will determine whether self-reference must be treated as *pathology* or can be treated as *structure*.

A continuing controversy in infinity is whether infinity is an actual thing or just a potential thing. Consider the natural numbers, they go on forever, you can always add one more, but there is no largest number. Seems like in this context, infinity is a potential; one never actually gets there. Contrast that with the idea of the set of all sets. There is no more increment; it's all there, complete. Seems like in this context, infinity is an actual thing. This controversy still divides contemporary mathematicians.

We begin with the lowly NOT gate; to which we add one more, then one more, then another, forever. Like this

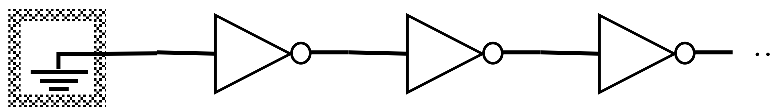


Figure 1 – **A Sequence of NOT Gates that Goes on Forever**: It's an unbalanced infinity. Given whether a gate is odd or even, counted from a known input value, the truthvalue of each output is deterministic.

If we know the input, in this case ground (ala false), then we know the value of the output of every gate. We simply count them; if odd, the output is true, if even, the output is false. But this infinity is one sided, unbalanced. What about this one

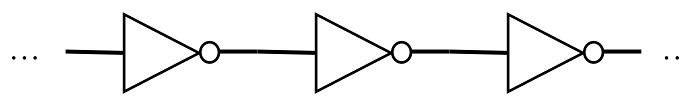


Figure 2 – **An Infinite Sequence of NOT Gates**: With the input infinitely far away, we can no longer count the gates. We get lost in the form.

Now we can't tell whether the gate is even or odd, and even if we could we have no information about the input – it is infinitely far away. Think of it this way; we are lost in the form. Hint 1: you have seen this form in this book already, a few chapters ago. The representation was different, but the form was the same.

Ok, how about this one

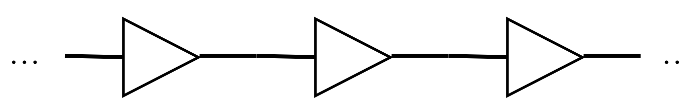


Figure 3 – **An Infinite Sequence of BUF Gates**: With the input infinitely far away, we can no longer count the gates, but it doesn't matter; their outputs will all be the same. If only we knew the input value…

Being lost is irrelevant, only the lack of knowledge about the input prevents resolution. Hint 2: you've seen this form before also.

The two preceding infinities have something in common; both are representable as a self-referential form. First the infinite series of not gates

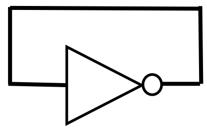


Figure 4 – **A Self-Referential NOT Gate**: The classic Liar's paradox: neither true nor false are self-consistent.

then the infinite series of buf gates

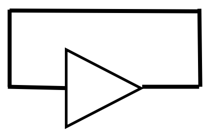


Figure 5 – **A Self-Referential BUF Gate**: The classic Circular Argument; both true and false are self-consistent.

The self-referential BUF gate is indeterminate; either true or false is self-consistent. The self-referential NOT gate is paradoxical; neither true nor false is self-consistent. Both, however, violate the design goal of classical logic, that every expression not contingent should have one and only one truthvalue. Self-reference has broken closure.

For the self-referential NOT gate, if we let the gate be clocked, we can get an interpretation; it oscillates between true and false. However, the phase of the oscillation is indeterminate. Either phase is self-consistent. We call one phase imaginary true, the other imaginary false.

Both forms are indeterminate, in the appropriate basis.

Completed Infinity

And here is the rub, neither form makes sense as a potential infinity. The self-reference implies that the infinity is actual, complete, there is no *one more* to be added. Let's make this a formal hypothesis, call it the *loop* hypothesis:

Loop hypothesis: "Any form that can be captured as a self-referential form is a *completed* infinity."

This is not an argument that all infinites are completed, perhaps some are indeed merely potential infinities, but any that can be represented in a self-referential way, those appear to have to be actual infinities. The corollary is immediate and obvious: completed infinities break closure; potential infinites may not.

Different Excursions to Infinity

Let's sharpen this idea. If we are lost in the form, say an infinite series of NOT gates, then we can envision slipping uncontrollably from gate to gate. We have no idea if we are on an odd gate or an even gate. But what if we could control how we slipped; say we only slipped by pairs, 2 gates at a time, like this

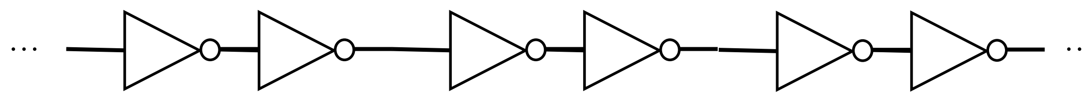


Figure 6 – **An Infinite Sequence of NOT Gate Pairs**: The spatial pairing is to suggest that while lost in the form, we are only lost in groups of two. The output of each pair is the same. Now, if only we knew the input value…

This has a different self-referential representation

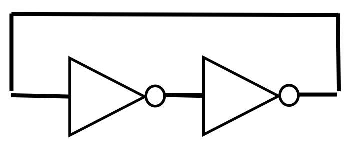


Figure 7 – **A Self-Referential Pair of NOT Gates**: The infinity above is represented differently. This is a two-stage paradox, and we should expect it might have a different evaluation than a one stage paradox. Indeed, it might not be paradoxical at all; perhaps it is indeterminate.

A self-referential pair of NOT gates is not paradoxical in the real basis, its indeterminate in the real basis; either true or false is self-consistent.

This distinction, rather subtle, is nonetheless easily captured in the notation of NLL. A simple paradox, indeterminate in i & j

$$A{:}{\sim}A \tag{1}$$

A simple indeterminacy, indeterminate in T & F

$$A{:}{\approx}A \tag{2}$$

A simple indeterminacy using a pair of combinatorial NOT gates (think of only one of them as clocked) no concurrency, also indeterminate in T & F

$$A: \sim\sim A \tag{3}$$

And finally, a pair of concurrent NOT gates, (ala Figure 7)

$$A{:}\sim B,\ B{:}\sim A \tag{4}$$

Indeterminate in…wait a minute, something's fishy. Time to bring out the truth tables, successor vectors, and attractor structures.

x	A	B	A:~B	B:~A	S(x)
0	0	0	1	1	3
1	0	1	0	1	1
2	1	0	1	0	2
3	1	1	0	0	0

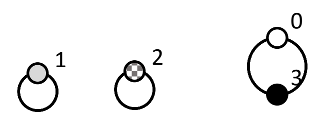


Figure 8 – **Truth Table, Successor Vector, and Attractor Structure for a Self-Referential Pair of NOT Gates**: Three pure attractors. The pair of singlets have nodes where both bits have opposite values. The doublet has nodes also where both bits have opposite values.

One of the most powerful paradigms is the principle of symmetry. Before analyzing Figure 8, let's go ahead and create Figures 9 & 10. You should be able to guess what circuit is relevant and that we'll want the corresponding truth table, successor vector and attractor structure.

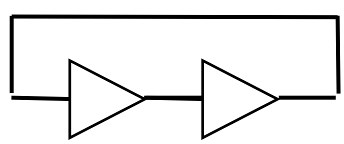


Figure 9 – **A Self-Referential Pair of BUF Gates**: This would represent an infinity of paired buf gates; a two-stage circular argument, but should we expect it to have a different evaluation than a one stage circular argument? Indeed, it might not be indeterminate at all; could it be paradoxical?

x	A	B	A:≈B	B:≈A	S(x)
0	0	0	0	0	0
1	0	1	1	0	2
2	1	0	0	1	1
3	1	1	1	1	3

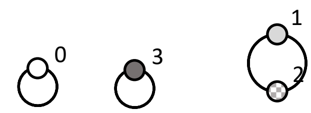


Figure 10 – **Truth Table, Successor Vector, and Attractor Structure for a Self-Referential Pair of BUF Gates**: Three pure attractors. The pair of singlets have nodes where both bits have opposite values. The doublet has nodes also where both bits have opposite values. But the nodes are swapped with those of the double NOT gate form

And, less we forget, the NLL equation

$$A{:}\approx B,\ B{:}\approx A \tag{5}$$

Now we have a complete view of the landscape. Now we can analyze. Both forms are indeterminate in both bases. In the real basis (T & F), indeterminacy takes on the form of a pair of singlet attractors whose nodes have opposite bits. In the imaginary basis (i & j), indeterminacy takes on the form of a doublet attractor whose nodes have opposite bits.

In the former case, the fixed points are random, in the later the phase is random. The symmetry is evident.

With these tools we can begin to see that our initial hypothesis that imaginary truthvalues might form a conjugate basis set to the real truthvalues has real (sorry) merit. For those still skeptical let me anticipate your witty comeback – the merit is imaginary.

Laws of Form Redux

Here is a more complicated excursion to infinity

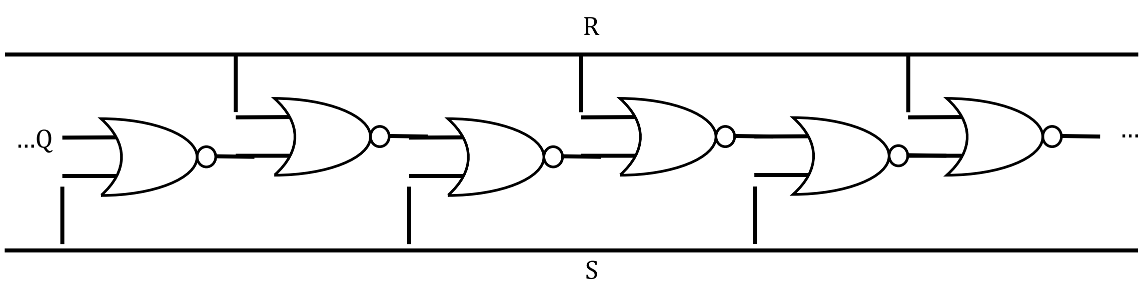


Figure 11 – **An Infinity of Alternating NOR Gates**: Two sets of NOR gates, interleaved. Both sets have one input coming from the previous gate in the other set. For the other input, the top set has input R, the bottom set has input S.

And what it looks like in its self-referential form

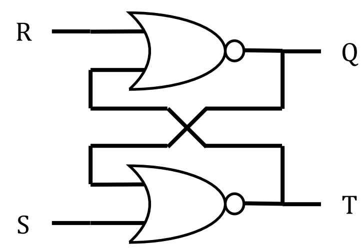


Figure 12 – **The NOR Gate SR Flip-Flop**: A simple 1-bit memory element. When R & S are false, this circuit remembers which one was last true.

You might recognize it from Chapter 25, Laws of Form; it is the recursive form G. Spencer-Brown first derives; the one which for one of its four possible input values is indeterminate, Equation (9). It is his first example of how self-reference breaks closure, where we must admit multi-valuedness into our logical systems. The Liar's Paradox, where we must introduce imaginary truthvalues, is his second example.

This form is well known, it is a staple of digital electronics. It is called the SR latch (or flip-flop). It's easy to remember the labels, they are alphabetical; QRST. Start with the output and go around counterclockwise.

Its output *T* is the output of G. Spencer-Brown's rendition, *f*. It is a memory element; stores one bit. When R and S are both false, it remembers which of R or S was last true. Think of R as *reset* and S as *set*. It is multi-valued only in the abstract sense. What it actually does is remember history.

Time has entered logic. Or perhaps, more revealing, time has arisen out of logic. It's no mean feat to *derive* time.

Truth Table

Here is the truth table for this form. It includes the NLL typographic strings and the successor vector. It also includes several more successive successor vectors to make it easier to determine the attractor structures. The horizontal lines indicate when R & S change values. Note that T is the output variable in this table (true for odd numbered nodes), which reverses the roles of R & S; R resets Q (sets T) while S sets Q (and thus resets T). Standards :(

x	R	S	Q	T	R:R	S:S	Q:R~∨T	T:S~∨Q	Sx	Sxx	Sxxx	Sxxxx	Sx?	Len
0	0	0	0	0	0	0	1	1	3	0	3	0	3	2
1	0	0	0	1	0	0	0	1	1	1	1	1	1	1
2	0	0	1	0	0	0	1	0	2	2	2	2	2	1
3	0	0	1	1	0	0	0	0	0	3	0	3	0	2
4	0	1	0	0	0	1	1	0	6	6	6	6	6	1

5	0	1	0	1	0	1	0	0	4	6	6	6	6	1
6	0	1	1	0	0	1	1	0	6	6	6	6	6	1
7	0	1	1	1	0	1	0	0	4	6	6	6	6	1
8	1	0	0	0	1	0	0	1	9	9	9	9	9	1
9	1	0	0	1	1	0	0	1	9	9	9	9	9	1
10	1	0	1	0	1	0	0	0	8	9	9	9	9	1
11	1	0	1	1	1	0	0	0	8	9	9	9	9	1
12	1	1	0	0	1	1	0	0	12	12	12	12	12	1
13	1	1	0	1	1	1	0	0	12	12	12	12	12	1
14	1	1	1	0	1	1	0	0	12	12	12	12	12	1
15	1	1	1	1	1	1	0	0	12	12	12	12	12	1

Figure 13 – **Truth Table for the NOR Gate SR Flip-Flop**: This table is taken directly from a spreadsheet which makes the analysis of self-referential forms less tedious. See the Appendix for the formulas.

Because there are four permutations of the input values, the attractor structures arrange into four groups. The first group, when R & S are both false, is the indeterminate group, the memory group if you will. The other three show brief transients as they are forced to a singlet where both Q & T are fully determined.

In the memory group you will see two singlets, where Q & T have opposite values. But you will also see a doublet, where Q & T alternate between true and false, and do so in phase. Where did this come from? The successor vector idiom is implementing concurrency; every input is sampled but the outputs are not updated until everyone is ready. This is the clocked gate metaphor. For the usual TTL (transiter-transistor logic, the digital circuits you can buy) the NOR gates are not clocked, and thus the paradox state is unstable. As a practical electronic matter, it is necessary for R & S to remain true long enough for signal propagation to get through both gates. Ultra-short true pulses will result in undefined behavior for such circuits.

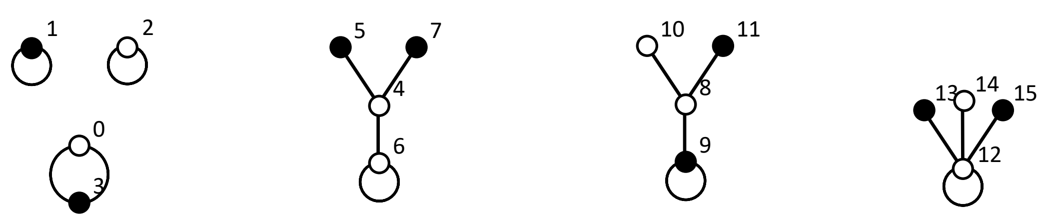


Figure 14 – **Attractor Structure for the NOR Gate SR Flip-Flop**: Four groups of attractors; the 1st group is the memory group, 2nd is the driven group when S is true, 3rd is the driven group when R is true, and 4th is the forced group when both S & R are true. Note in group 4 how the stems are shorter, the driving input doesn't have to propagate to the other gate. For clocked gates, it is also possible to store a paradox

in this memory (imaginary truthvalue, phase determines whether i or j). In non-clocked gates, this state is unstable. This form remembers which of R or S was last true; its value depends on history – time has entered logic.

NLL

And lest we forget, the NLL string (concurrency model, ala clocked gates, can store imaginary truthvalues as well). T being first implies that odd nodes are true.

$$\mathbf{T:}S\sim\vee Q,\ \mathbf{Q:}R\sim\vee T \tag{6}$$

Or this expression, which can only store real truthvalues, more typical of circuits in actual use.

$$\mathbf{T:}S\sim\vee(R\sim\vee T) \tag{7}$$

Or this one as well, but where S really is *set,* and R really is *reset*.

$$\mathbf{Q:}R\sim\vee(S\sim\vee Q) \tag{8}$$

If you've gotten this far, congratulations; this is all rather technical. But you should be starting to feel comfortable with the tools. You can represent self-referential forms not only in each of the five tools, but you can now also unroll them into an infinite recursive form, and back again.

You are now in possession of a pretty impressive toolbox.