

## **Analysis of Stacking Sigma Results from a Turing State Machine**

**Lisa Piccirillo - Solving the Conway Knot**

[https://en.wikipedia.org/wiki/Lisa\\_Piccirillo](https://en.wikipedia.org/wiki/Lisa_Piccirillo)

**Matt Parker - Stand-Up Maths**

<http://standupmaths.com/>

Hypercube and Turing Machine Videos

**Moritz Firsching - Unfolding of the Hypercube**

<https://unfolding.appperceptual.com/>

**Pascal Michel - Busy Beaver Game (Explanation)**

<https://webusers.imj-prg.fr/~pascal.michel/>

This paper was inspired by thinking about how to achieve a contra flow of instructions. In previous papers it is assumed that there is an X and Y flip somewhere to achieve the combination of data to produce the results we see.

This paper looks primarily to show how a contra flow set of state machine instructions could be built and overlayed without assuming it is present without reason.

It is recommended to reference previous papers, in particular the Analysis of a Turing Machine to understand the sigma(0,0) result set pattern.

<https://github.com/Sandcrawler/turingmachine>

### **Sigma(0,0)**

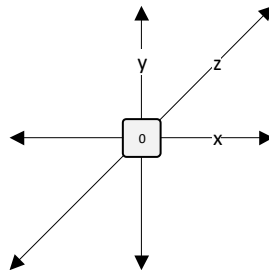
In the last paper it was observed that  $\text{sigma}(0,0)$  could be referenced as  $\text{sigma}(+0,+0)$ ,  $\text{sigma}(-0,+0)$ ,  $\text{sigma}(+0,-0)$ ,  $\text{sigma}(-0,-0)$ .

Let us extract this out further and show how the sigma can be used in a larger pattern set and how this could then allow for a contra flow set of instructions to occur.

## Quadrant Representation

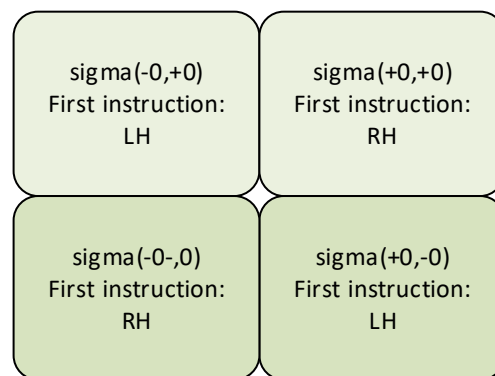
Consider one of these quadrants as what we have been working in to now (although with the assumed contra flow of instructions to manifest what the results have been).

Consider a graph with 8 quadrants.



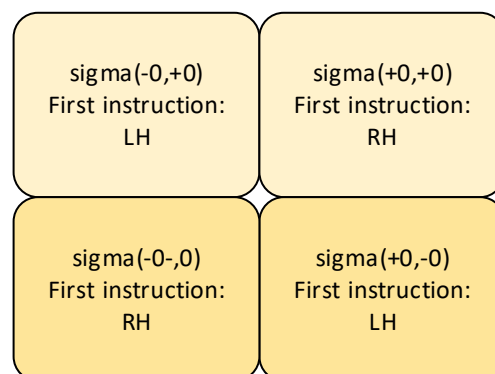
*Figure 1.1*

### Upper Quadrant



*Figure 1.2*

### Lower Quadrant



*Figure 1.3*

## Quadrant Mapping

This diagram shows a simple mapping of quadrants.

Consider 0 as 0,0,0 and each of the 8 quadrant nodes as the infinite opposite of 0,0,0 toward the opposing end of the quadrant.

The nodes will then point toward the same infinite points that 0,0,0 will along the axis to create a quadrant.

Quadrant	Axis-1	Axis-2	Axis-3
TR+	+y	+x	+z
TL+	+y	-x	+z
TR-	+y	+x	-z
TL-	+y	-x	-z
BR+	-y	+x	+z
BL+	-y	-x	+z
BR-	-y	+x	-z
BL-	-y	-x	-z

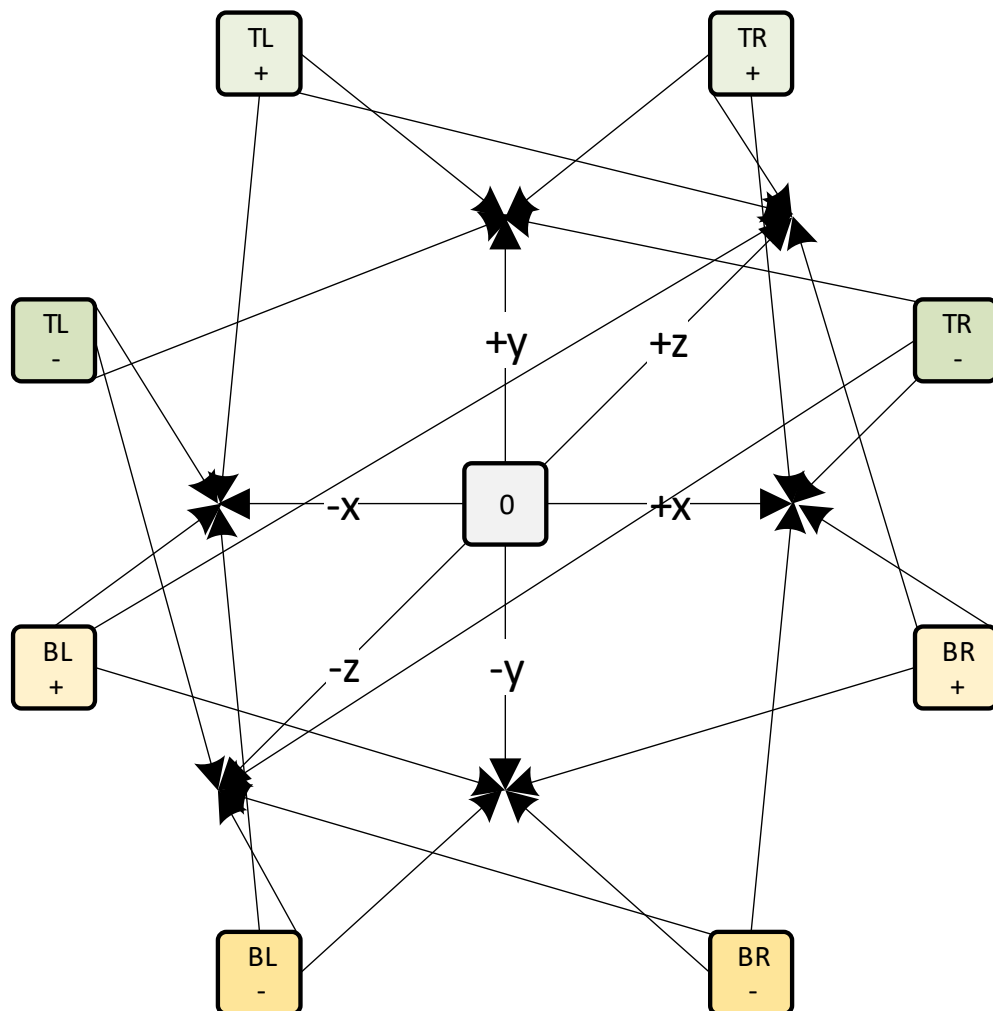


Figure 2.1

## Upper Quadrant Mapping

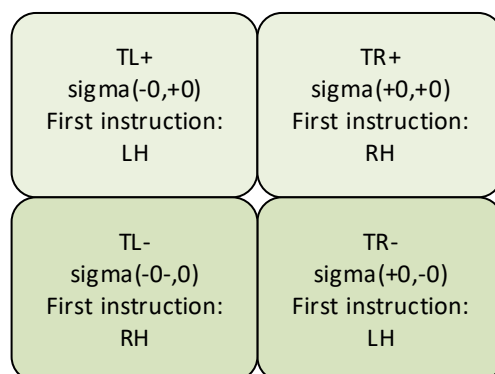


Figure 3.1

## Lower Quadrant Mapping

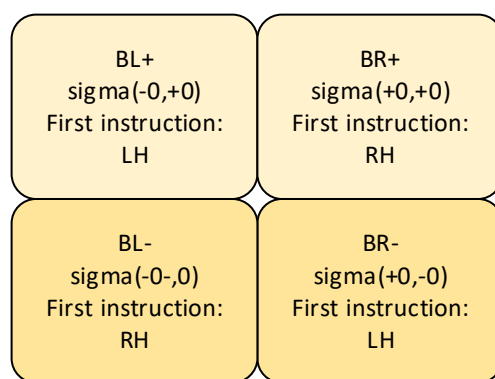


Figure 3.2

We now build the 8-quadrant structure up and overlay the structure on itself in a symmetrical manner by placing 0,0,0 at each of the 8 nodes.

What we see at this point is all 8 quadrants are all equal.

This would suggest that the inverse instruction set is equal to the forward instruction set and that everything starts at 0,0,0 and ends at 0,0,0 and every corner of the quadrant is 0,0,0.

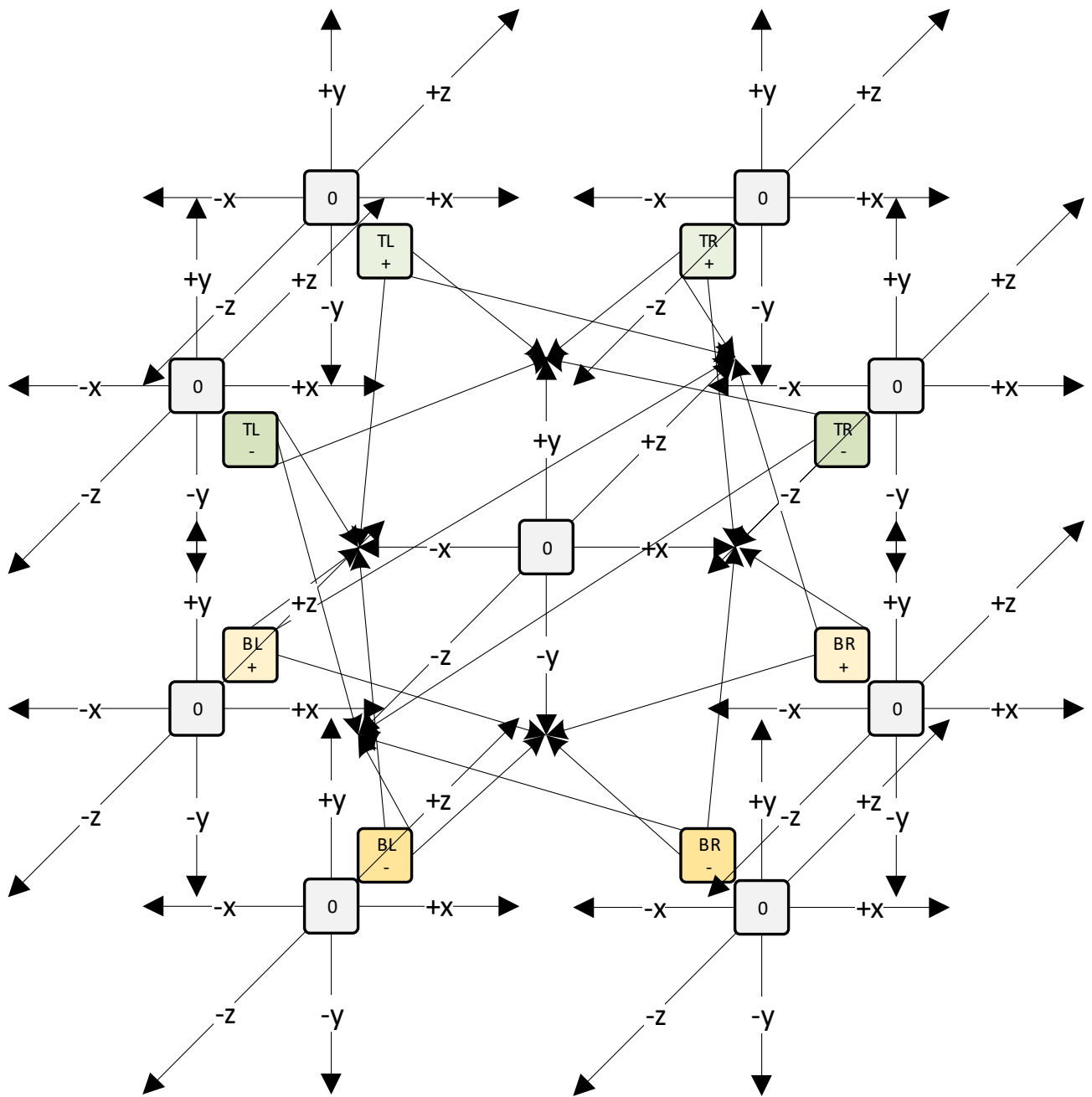


Figure 4.1

## State Machine Instruction Flow

Consider from 0,0,0 sigma(+0,+0) runs and follows a trajectory towards TR+

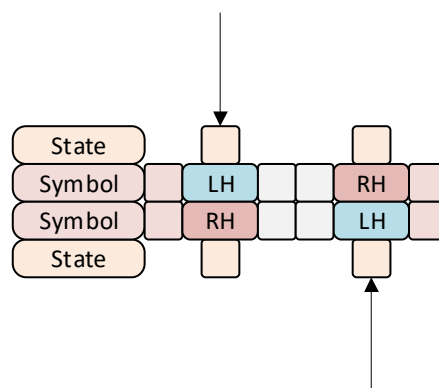
We would see from TR+ (which could be considered as 0,0,0) BL- running toward 0,0,0

This would result in instructions sets in each direction starting with the same Left or Right instruction giving a contra flow set of instructions.

In the simplest example:

LH originating from 0,0,0

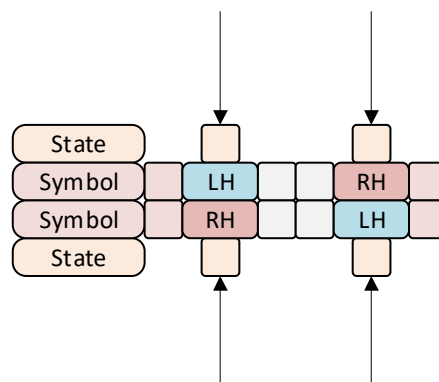
LH originating from TR+



If we consider that the quadrant is overlapped with all 8 instructions, we could also suggest the following:

RH originating from 0,0,0

RH originating from TR+



## **Observations**

### **Observation #1**

If all points at opposing ends of the hypothetical cubes (quadrant), ie all 8 corners are considered as 0,0,0. The overlapped space would have all instruction sets running within each quadrant in all directions.

You would in this case get contra flowing sets of instructions in every symmetry.

This would result in the ability to combine instruction sets that come from opposing directions as the inverse instruction would be available.