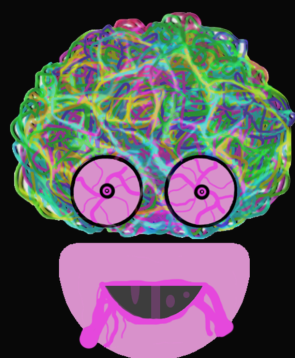
Vou's Buffet?



Von's Buffet!

Warning: The following content has the potential to significantly alter your perspective of the universe. If you do not seek these kinds of transformations, Von's Buffet is not for you.

Bon appétit!

*Vou's Buffet?*


*Von's Buffet!*

## Open

In the brilliance of brainy adventures, computational thinking is a Buffet for nature’s code—our welcoming reminder that within us, infinities ViB an art of computation.

As a reward from Stephen Wolfram's cellular automata journey, "RULE 30" is a good splash of Buffet juzce! ViB’c echo through 8 rules, pulsing a blend of simple, complex, and sexy.

The same taste as our inner relationship with the brain; Von’s Buffet has made break throughs in understanding the soul; it’s likely Rule30 has been your conscience all along?

Maybe not…

But we now have the insight needed to make the connection. And with the chance to claim a piece of Wolfram's bounty, we will be answering three simple questions, as you will see.

## Baby Bites

Knowledge thrives on repetition; reliability is the key ingredient. Without it, referencing information has no meaning.

For the sake of hungry brains, we employ reliable information theory to break Rule30 down to a bite size. Without this, we get lost in the sauce.

The goal is to understand how input-output sequencing matters in deciding the overall infoo content of rows and columns.

## Mommy Will Show You

First, get a taste! Start with the abundant 'foo', but savour it like it's scarce.

At first glance, on the left side, you will discover endlessly extending patterns towards the bottom corner.

Assign each row a natural number 'r'. When 2 rows have matching patterns, compute 'row distance'(Δr):

$$\Delta r = r_{new} - r_{old} \quad [-2]$$

Predict the next row number $r_{predicted}$ using Δr:

$$r_{predicted} = r_{new} + \Delta r \quad [-1]$$

Patterns can appear anywhere along a row. Consider 'c' and the 'column distance'(Δc) from the center:

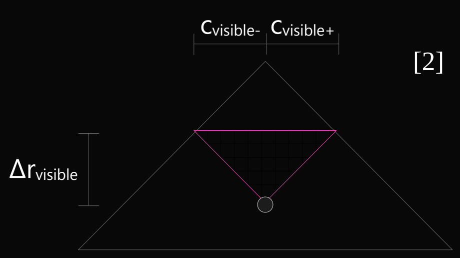
$$\Delta c = c_{new} - c_{old} \quad [0]$$

Predict the next column number $c_{predicted}$ using Δc:

$$c_{predicted} = c_{new} + \Delta c \quad [1]$$

This defines a Rule30 Buffet, where 'foo' is separated by {Δr, Δc}, offering a taste of infoo richness.

Savoring 'foo' is always an option, but true enjoyment requires navigating the {Δr, Δc} complexities.




[2]

## Big Bites

At a glance, on the right side, Rule30 appears to taste kinda spicy. The 'foo' is non repeating and savouring it is a challenge. On the left, structure unfolds; on the right structure folds. Yet. Even the far right edge has an after taste.

Why are these tantalizing morsels so?

In Rule30, information travels away from a box, into the future rows, merging with 'foo' from other points.

Each box must be viewed in the context of others. The Rule30 input-output planes show limited connectivity, bounded by a 'speed limit':

$$\{r_{visible}, c_{visible}\} = \{r_{origin} - \Delta r_{visible}, c_{origin} \pm \Delta r_{visible}\} \quad [2]$$

$\Delta r_{visible}$ is the absolute row distance between a 'row of the past' and 'the current row':

$$\Delta r_{visible} = | r_{visible} - r_{origin} | \quad [3]$$

± means we can consider both the left and right columns relative to $c_{origin}$.

$\{r_{visible}, c_{visible}\}$ signifies the furthest visible row, and the furthest left or right column (depending on ±).

Essentially, the computational timeline dictates the ($\pm\Delta r_{visible}$) 'column scope' of 'foo' dependence. It defines the infoo sharing speed limit between rules.

This 'foo' sharing potluck means each box in each row depends on previous box(c).

Box->Box 'foo' sharing occurs only between box(c) whose column separation is less than or equal to their row separation $\Delta r_{visible}$.

Otherwise, the box won't contain that infoo and will taste different.

## More Foo Please

V{DFG, DGF, GFD, FDG}
B{FDH, GFC, GEA, GEB}
C{DHG, EBD, FCF, FCE}
D{DHH, EAA, EAB, EBC}

E{BDG, CFD, HGF, HGE}
F{CFC, CEA, CEB, BDH}
G{HHG, BCF, BCE, ABD}
H{HHH, ABC, AAA, AAB}

[4]

It is likely you know Rule30 by the interesting colors it makes. But the real action is in the rules. As you know, colors are just the inputs and outputs to the rules.

So, everything you know about the patterns of colors, is the same when thinking about the patterns of rules, except in greater detail.

Each rule depends on a unique combination within curly braces {}. Rules emerge as three rules coalesce into one, like DFG converging to V, or HHH to H.

This 'rule flow map' [4] illustrates their interdepence, offering insights into color patterns.

Each new rule depends on two previous rule inputs, leaving only two possibilities for the third input; Changing one letter in the map results in a single replacement, highlighting logical dependencies in rule combinations enforced by input patterns.

Crucially, take G, the first rule on the left edge of every row. It outputs white. Now, shifting the input space right, we expect F or E to the right of G; both output white. So, we get a repeating pattern on the left edge:

GF/GE, GE, GE...

Since only blacks are on the left of G, we apply the rule for all black inputs, H:

HGF/HGE, HGE, HGE...

The cycle continues unless white appears in the input space of H. This won't occur on the leftmost edge but may happen elsewhere.

Expanding beyond GE, we encounter another black or white box, giving us:

GEB or GEA

Both A and B output black, but have different inputs. Since black favours B's input, we get a Buffet in favour of B:

GEB/GEA, GEB, GEB...

Interestingly, G, E, and B, are 3 rules that are included in their own dependence maps [4]. All Buffet's are based on this property. But if we move further, beyond EB, we can generalize this property:

EBC/EBD, EBC/EBD, EBC/EBD...

Here's both C and D in superposition. Because C outputs black and D outputs white. The reality is, that only one can be chosen:

EBC, EBD, EBC, EBD...

It's feeling kinda spicy. Lets go further and see what happens:

BCF/BCE, BDG, BCE, BDG...

BDH/BDG, BCF/BCE, BDG, BCE...

## Let's Have A Drink

Rules create patterns through logical relationships. On the sides, patterns repeat in sync with the edge. Inwards from the sides, 'foo' gets too spicy.

Let's rethink our approach.

We know edge rules start color sequences. Adjacent rules maintain harmony with the edge, presenting uncertainties at a rate $2^X$ | x is distance in from edge.

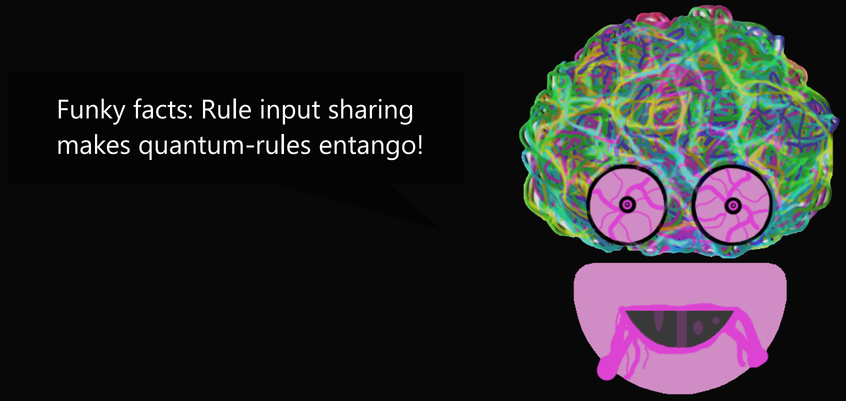
Classically, rule outputs of previous rows resolve the uncertanties; and nothing is uncertain.

"There is a rule for everything!"

Quantumly, instead of resolving the uncertainties, we let uncertainties resolve themselves.

"There is everything for a rule!"

Hmm..



## Where Should I Put It

To answer questions, it helps to move in a direction.

Wolfram's inquiries specifically revolve around the central column.

In the preceding example, the emphasis was on Rule30's edges, with the 'left' (or potentially 'right') edge serving as the initial condition for each row. We demonstrated how this initial condition is inclusive to behaviors across rows, manifesting increasing uncertainty towards the center.

From a statistical perspective, this metric affirms that the center column becomes progressively uncertain. However, this means to avoid the details ;) and it is presumed that the Wolfram team anticipates a more nuanced formalism.

The key topic then is the 'initial condition'. If the initial condition is set to black, only H would be applicable, resulting in the absence of any white outputs. While statistics would still indicate escalating uncertainties towards the center, the reality is 100% H.

On the flip side, the rule flow map [4] depicts a condition in which H relies entirely on itself.

Broadly speaking, the map illustrates how initial conditions will correspond to rules. Thus, our focus is on utilizing the map to model 'paths' that symbolize input output rule sequences unfolding over space and time.

Various methods exist for defining these paths (as you will see), but the essence is that paths remain undefined without simulating from an initial condition. And since all paths must interact, our general use of the map fails due to the increasing information demands set by $\Delta r_{visible}$ [2] .

Forced to accept this, if our objective is to establish anything concrete about Rule30, the language of uncertainty becomes our primary means of expressing the map.

## You Told Me You Were Hungry?

If we must assume complete infoo, without complete infoo, then our only approach is statistical. Thankfully, we can still formulate some ideas using the quantum language.

The row quantum states discussed are column-column. Being interested in the center column, we are interested in row-row. Since column to column interactions exist via entanglement, we must respect it, and consider both (column-row).

Quantum states over rows is simply quantum states defined over space $\Psi(c)$. Instead we need to define a quantum state that also exists over time $\Psi(r, c)$. Blessed with the best, we know about Rule30 Buffet's, and how they work [-2] → [0], they repeat at an interval:

$$\Psi(r, c) = \Psi(r+\Delta r, c+\Delta c) = \Psi(r_{predicted}, c_{predicted}) \quad [5]$$

What we did is define Rule30 Buffet's in terms of quantum states. Where the wave function is defined over the Buffet interval.

Buffet's before = 'rules at predicted positions'.
Buffet's now = 'rules at predicted positions' + 'rules at positions leading up to them'.

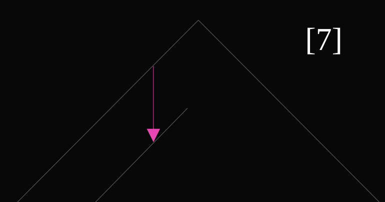
We are still doing same as before [-2] → [0], except now the context is 'all rules existing *between* and *at* predicted positions', instead of 'only rules at predicted positions':

$$S = |\Psi(r, c)\rangle = \sum^{N} a_N(r, c)|\Psi_N\rangle \quad [6]$$

'$a_N(r, c)$' is the probability amplitude of any rule 'N' defined as $N^{-1/2}$.
'$|\Psi_N\rangle$' is the state basis vector along the $N^{th}$ dimension, for any rule(N) of a sequence.
'$|\Psi(r, c)\rangle$' is the superposition 'S' combining N Buffet's with the same ($\Delta r$, $\Delta c$).

* Remember, all possible rule sequences matter. And rules that add multiple times to the superposition, must have a different '$a_N(r, c)|\Psi_N\rangle$'. Because their ($\Delta r$, $\Delta c$) is the same, But their ($r_{predicted}$, $c_{predicted}$) differ. *

In the context of the above exampes, this superposition is any diagonal pattern in sync with the left edge. The sync enforces entanglement. With increasing row count, the number of entangled Buffet's increase [7]. (Using [6] as same formula for entangled superpositions of 2 or more 'S' combined) this just means an increasing 'N'.



[7]

**Save It For Later Then...**

WOLFRAM RULE 30 PRIZES

## 1. DOES THE CENTER COLUMN ALWAYS REMAIN NON-PERIODIC?

Periodicity here can refer to rule or color patterns. Colors can result from four different inputs, making it tricky to trace back which rules were used. Analyzing colors alone erases information about the rules and surrounding possibilities.

On the other hand, rules make clear on which rule generated an output and some details about surrounding possibilities. Since rules dictate colors, grasping their behavior helps us understand colors, which are essentially a simplified representation of rule actions.

Each row of the 'center column' encodes a new superposition in relation to previous ones coming from the edges. It also encodes the superposition it takes part in, mathematically represented as:

$$E(r, c) = S(\theta_c)_{r-1} + S(\theta_{c+1}) + \dots + S(\theta_{c+n}) \text{ [8]}$$

$\theta_{c+n}$ is the phase-period function specific to each superposition nearer the edges of a column. $S(\theta_{c+n})$ is the periodic state of superpositions in the column scope of encoding E(r, c), leading up to the outer edge defined by $n < (2^r - 1)/2 - 1$ units away from the center. (n and N are different.)

Every additional row, adds an additional term:

$$E(r, c) = E(r-1, c) + S(\theta_{c+V}) \text{ [9]}$$

The continual introduction of new terms disrupts stability established by previous terms.

This is a best case statement made without simulating the entire output plane, as outlined by the following:

Rule flow maps indicate that each rule can emerge following the occurrence of any particular rule. This emergence is contingent on E(r, c) and its dependencies. More broadly, originating from the left edge, superpositions within the column scope, cause the center columns movement through the rule flow map to be governed by probabilistic behavior set by other superpositions [8].

The center column exists in a superposition state of not rules, but periodic possibilities E(r,c). Because if the superposition it encodes is always changing as a function of 'r', and a superposition is descriptive of periodic rule change, it is by definition a 3rd order derivative. Making it so random, not even a quantum system can model it probabilistically without an infinity of qubits for each possible period.

As a result, any periodic behaviour is probabilistically in superposition with all other periodic behaviours. Any 1 periodic behaviour is undefinable, and the central column is therefore 'random'.

Answer: Yes

## 2. DOES COMPUTING THE NTH CELL OF THE CENTER COLUMN REQUIRE AT LEAST O(N) COMPUTATIONAL EFFORT?

For any given row size, you can compute a fixed number of rows into the future:

$$r_{futureMAX} = \mathrm{Ln}(2^{r}) / \mathrm{Ln}(2) \quad [10]$$

Where $r_{futureMAX}$ is the number of rows into the future, computed without increasing the row size. 'r' is the row number of the current row [12].

In terms of N, we want to know, how many rules are needed to compute the $N^{th}$ cell of the center column; call this **O(N)**. We simply add all the column sizes leading to and including 'r', and all column sizes leading to the target, as shown [11] and [12]:

$$O(N) = (1 + r)^2 + (r_{futureMAX})^2 - 1 \quad [11]$$

This assumes we must fully and sequentially compute each row to get the next, and this would be the minimum computational effort.

Instead, if we had some faster function to determine the color of the center's Nth cell, it would take the row number as input and produce a binary output. This function must focus solely on the center column, and disregard rule input-output topology.

Unlike information loss by color, this is information loss by rule. The function aims to compute the color with fewer computations than the O(N) approach mentioned earlier.

Every output in the center column emerges from a unique set of inputs, unlike in math, where functions have homeomorphic transformations. The rules are not characteristic of functions. They are characteristic of number systems.

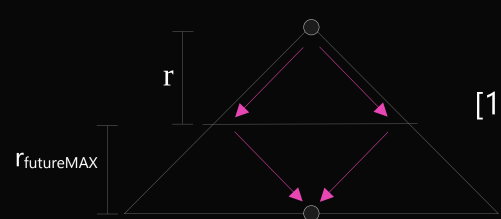
Imagine removing 1 from the Natural Number line. 'Natural Numbers' become inconsistently illogical.

The same is true of removing rules. You are effectively removing numbers from a number line. Because would be states no longer exist, even though its expected to.

Alternatively, you could get lucky, and step outside when its raining the perfect function, but this function is faced with the problem of not knowing which row you will request information for, and its output generation process must take all endless possibilities into account, thereby always exceeding any O(N) metric. (If it had knowledge of the future, it could by pass this constraint, outputing the correct answer with no information from you. But this is arguably beyond the scope of Rule30 as an independent system of study.)

Since rule flow maps are fully connected over time, there are no simplifications to be made of rule to rule interactions either; without knowing when and where interactions are considered obsolete.

Answer: Yes




[12]

## 3. DOES EACH COLOR OF CELL OCCUR ON AVERAGE EQUALLY OFTEN IN THE CENTER COLUMN?

It was proven how input superpositions result in undefined perodicities. The result is that each of the 8 rules has an occurence probability of 1/8.

4 of the 8 rules have a unique output. The probability of white occuring is 4 * 1/8 = 50%. The same is black. Therefore it agrees with equal average occurrence.

This is true for changing any 2 inputs, or any 3 inputs, respective output ratios are 12W:12B, and 4W:4B.

But this is not true for 1 input: 12W:9B.
Changing 1 from inputs with 2 or more white: 7W:4B.
Changing 1 from inputs with 2 or more black: 5W:7B.
And is some bias that may be exploited.

Using the rule flow map [4], we want to know the probabilities of rule occurence in the center column. Since the rules are connected, whenever 1 rule occurs, it makes any rule included in its dependency more likely.

Skimming a quick simulation, it shows how H, D, E, G are the most likely rules to occur. Among these H appears as a dependent the most, and this amplifies the presence of H; making it the most likely occuring rule. H outputs black.

However, this is influenced by the starting condition. If the starting condition is black, H would be the only thing that happens. And the self recurring dependence loop: HHH → H, would be the only 'path' taken through the rule flow map.

As discussed, paths are the framework for unifying rules with initial conditions. To be more formal:

$$\text{ced} = \Delta\text{path} = \Delta^2\text{rule} \quad [13]$$

'ced' is inital condition. 'ced' changes the path. This correspondingly changes how the rules change.
Path is a defined as the rule→rule change over time.

Also, each time step adds additional paths. Path change in relation to other paths matters; This is considered the initial condition with respect to any path. Like column scope, this path relationship changes indirectly:

$$\text{path} = \Delta\text{rule} = \Delta^2\text{ced} \quad [14]$$

Maintaining this framework, we can define path trajectories along a column. Edge rules makes additional columns. Additional columns means changing the initial state:

$$\text{rule} = \Delta\text{ced} = \Delta^2\text{path} \quad [15]$$

It shows rules indirectly change the path, but directly change the next initial state, as you would expect.

Comparing the ced-rule derivative [13] and the rule-ced derivative [15], it shows rule = $\Delta^3$rule. It shows uncertainties exist as functions of rules, ced, or paths:

$$\text{rule} = \Delta\text{ced} = \Delta^2\text{path} = \Delta^3\text{rule} \quad [16]$$

This equation holding true, means it is computationally irreducible. It proves the existence of initial states, paths, paths←→paths, superpositions, rules, and the mutual relationships between them.

It's then not possible to predict paths taken without simulation of paths in the context of rules and initial conditions.

In other words, this means there is no evidence of integrating any information to disprove the color occurence average in the center, because there exists no way to resolve the 1/8 probabilities without simulation.

Answer: Yes

How do you expect more accurate 'Yes' without preparing the infinity needed to explain it?

Perhaps.. The easiest way to know, is to check someone who checked?

#CheckMate

*Vou's Buffet?*

## Von's Secret Sauce

*All foo has been made possible by the following recipes:*

$$B = \Delta C$$
$$C = \Delta B$$

$$V = \Delta B = \Delta^2 C$$
$$B = \Delta C = \Delta^2 V$$
$$C = \Delta V = \Delta^2 B$$

$$P = \Delta V = \Delta^2 B = \Delta^3 C$$
$$V = \Delta B = \Delta^2 C = \Delta^3 P$$
$$B = \Delta C = \Delta^2 P = \Delta^3 V$$
$$C = \Delta P = \Delta^2 V = \Delta^3 B$$

...

*Von's Buffet!*

*Vou's Buffet ?*

*BV = $2.10 + Floss*

*Vou's Buffet?*

*Von's Buffet!*