Math
Darrin Taylor
July 30, 2019

Collatz 3n+1 conjecture

A new mathematics for the 3n+1 problem

Initial definitions needed to proceed

The Collatz conjecture states

Take any integer A0

If even divide by 2

If odd multiply by 3 and add 1

And it will always decay into the loop 1->4->2->1

There are two necessary steps to prove the conjecture:

1) Prove that no other loops are possible
2) Prove that once no other loops are possible that the sequence cannot infinitely ascend.

The approach here is to apply base 3 mathematics on partially known numbers to prove the first step. So the math side of it is a little different and it reads a bit like a logic puzzle from computer science instead of normal mathematics.

First lets define a starting number:

A0

They key to this approach is to take A0 and convert it into base 3 number.

Base 3 is like binary but instead of 0 and 1 it has 0,1,2 and each digit is 3 times the next lower digit.

Forward Vs Backwards

First consider a number

We will use A0=308 base 10 to make a point about forward and backwards.

308 base 10 is expressed as 102102 base 3

This number is even so we divide to
A1=154 base 10 expressed as 012201 base 3

The forward direction is from A0->A1 and divides by 2; the backwards direction is from A1->A0 and multiplies by 2. Some things are easier to see in the backwards direction other things are easier to see in the forwards direction. This proof will use the words forward and backwards to mean forwards in time and backwards in time using the known rules of divide by 2 and 3n+1 in the forward direction and multiply by 2 and (n-1)/3 in the backwards direction.

When using partially known numbers which are the Most significant digits a branching occurs backwards because there could be a carry up from unknown digits of +1 or +0. When using the least significant digits in the backwards direction this does not occur.

Why base 3?

We have chosen base 3 because it vastly simplifies the 3n+1 and inverse (n-1)/3 that are used moving forwards or backwards through a collatz sequence or collatz loop.

SHIFTS DO NOT CHANGE ANY DIGIT VALUE EXCEPT TO SHIFT THEM EXCEPT LEADING ZERO AND LEAST SIGNIFICANT DIGIT.

Simple Rules of base 3:

1) Even or odd is determined by an even number of 1s as both 0 and 2 are even.
2) Multiply by 3 is a shift to the left.
3) 3n+1 is a shift to the left and the least significant bit becomes 1
4) Divide by 3 is a shift to the right.
5) (n-1)/3 is a shift to the right that destroys the least significant bit (which MUST always be a 1 because we are just doing the reverse of step 3 above)

Importance of base 3 digit length

There are only 2 operations that change the total number of base 3 digits in the forward direction.

1) 3n+1 shifts the digits up 1 and adds a 1 in the least significant digit and this increases the total number of digits by +1
2) Dividing a number that has a leading 1 decreases the total digits by -1

The important aspect to realize here is that if you count the number of divides of numbers that start with leading 1s and you count around the entire loop you will have the correct total number of shifts around the whole loop. This is because the loop must return to the starting integer and to the starting integer's total digit length. So the change must be 0 so total shifts must equal total leading 1s divided.

Because the shifts 3n+1 or (n-1)/3 are difficult to predict it is sometimes easier to never write them down and recover the information from counting the leading 1s.

So a technique exists where we move backwards in time and just have a sequence of multiply by 2 and know the number of shifts by virtue of counting the number of leading 1s then we can say things about the loop ONLY from the leading digits for each number in the sequence.

Although there is no clear pattern regarding when shifts occur, there is a VERY clear pattern regarding when leading 1s occur and how often digits can be lost moving forward from dividing leading 1s.

Moving backwards the same logic can apply provided the numbers we are dealing with are large enough that the total digits used stays below a total number magnitude of the 2^60s that have already been numerically eliminated in previously published attempts at this problem. We will need less than 30 base 3 digits so this will not be a problem.

Pre-View of Proof

Proof that there are no loops except 1 -> 4 -> 2 -> (1) base 10

1) Moving Backwards using base 3 consider the most significant digit 1# or 2# and prove that there are only 2 patterns possible that when converted to their Forwards form are:

Segment type A

1# -> 2# -> (1#)

The (1#) is in the next segment shown because of the "->" multiply by 2 after 2#.

There is one leading 1 so there are one non local "shifts" created by this segment.

This represents 2 divides (->) and 1 shift (3n+1 or (n-1)/3)

and

Segment type B

1# -> 2# -> #1 -> (1#)

The (1#) is the next segment shown to show 3 divides (->) in-between numbers

There are two leading 1 so there are two non local "shifts" created by this segment.

This represents 3 divides (->) and 2 shift (3n+1)

2) Show that A always has B on both sides of it. This means that the number of Bs between As forms a king of super segment. AB(A) being 1 B, ABB(A) being 2 Bs in between etc.

3) Show that ABB(A) and ABBB(A) are the only possible super segments. and they restrict the loop length to 13x +18y

Argument that infinite ascent is unlikely (starts on page ~57)

1) Show for non trivial sequences an ending of #2 or #01 always occurs

2) Show that once an ending of #2 or #01 is reached that there are only 16 segments of possible branching and all return to #2 or #01.

3) Show that 15 of the 16 pathways descend and characterize the only one that doesn't descend.

4) Show that in Base 4 the average expected value before a 3n+1 to the next 3n+1 is less than the starting value and that although the rising segment can repeat the falling segments can also repeat and the pattern revolves so that each ending of base 4 double digits will tend to be cycled through without an obvious bias to overcome the ~factor of 4-5 bias toward descent.

5) Show that in Base 4 if the next digit up is always even that it always descends. The leading 0s above the number are infinitely even. If each digit pair can be shown to relate to the pair above it in a way similar to 3n+1 or 3n-1 then we can show that eventually the lowest digit pair will behave as if the digit directly above it is always or usually even and thus descend in a predictable knowable bases because the leading 0s are infinitely even and that "leaks" down to be represented at the lowest digits eventually.

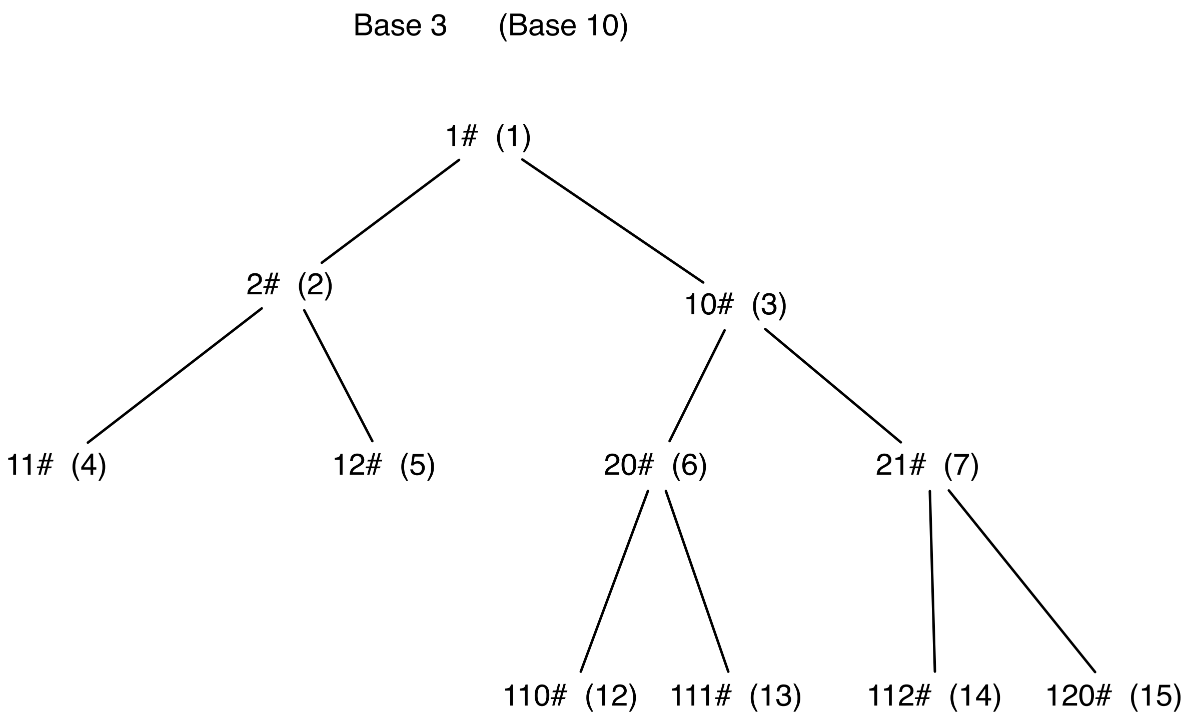
Proving only 2 patterns

We will start by considering a base 3 positive integer number with a start digit of 1 and unknown digits and unknown digit length denoted as #. Because numerical methods have been used to show that all numbers smaller than 2^60 return to 1 we will not worry about running out of digits and emptying #. We will only need 20 or 30 digits to make our case.

Moving backwards in time divide by 2 becomes multiply by 2. In base 3 the 3n+1 shift just shifts up the digits adds a 1 to the unknown number's least significant digit. But this doesn't change the order or content of the "known" digits. So we will only multiply by 2 and use the leading digit to infer how many shifts have occurred and will completely forget about the (n-1)/3 operation as it can be recovered by using the leading 1s.

This is possible because a Collatz loop sequence of numbers in base 3 has a starting position that returns to itself and therefore a loop must have increased digits (shifts= +1 digit) that are offset by decreased digits via leading 1s divides (= -1 digit). Shifts are the only thing that increases digits and it is always by a single digit in base 3. Divide by 2 of a leading 1 is the only thing that decreases digits and it is also always a single digit. Because we are only considering multiplies and leaving all the shifts un-shown there is therefore a 1 to 1 correlation between a leading 1 occurring and a shift occurring somewhere in a loop. This logic ONLY works for sequences that start and end with the same number of digits. We will need different logic when we talk about non loops or segments that change total number of digits.

The backwards multiplication tree in base 3.



Start with 1# and multiply it by 2 to find the previous first digit.

There are two possibilities:

2*1# and 2*1# + carry up from unknown number.

****

We pause for a moment to consider the least significant 2 digits which we will multiply by 2 and see if they carry up more than 1. I will show the numbers as numbers in base 3 but larger than 2 so that you can easily see what the value is and pay careful attention to the carry process which must bring each digit down to 2 or smaller by carrying up a full next most significant digit which is 3 base 10 or 10 base 3.

#22 *2 = #44

carry up the least significant 4 by subtracting 3 to get 1 and add 1 to the next digit.

#22 *2 = #51

the 5 is smaller than 6 so only a single carry up exits in the numbers we can see

#22 *2 = #21 + single carry up of #100 added to the unknown 3rd least significant digit.

*****

Returning to:

2*1#   and 2*1# + single (1) carry up from unknown number.

we get 2 possible values in base 3

2# and 10#

Here 1#*2+1#=3# and 3# in base 3 is 10# and the unknown part of the number remains so 10#.

This Tree shows the possible values in base 3.

Notice that all the numbers on the bottom line follow the pattern of the top 1# if we just forget their values except for the first digit and start over at the top.

There are only 2 types of backwards segments in a loop if we sort by most significant digits.

1#->2#->(1#) from the left side of the tree

and

1#->1#->2#->(1#) from the right side of the tree

Here the (1#) represents the first value of the next segment.

-> represents a multiply operation or a divide operation if moving forward

Leading 1s represent a shift that must occur to balance out the loss of digit in the forward direction.

So loops are built in the backward direction by either

1 shift to 2 Divides.

or

2 shifts to 3 Divides.

If you wonder what happens if you start with 2#.

If you start with 2# you can multiply to get 11# or 12#. Then you simply reposition your point of view and start with the 1# using the diagram above. The leading digit is always either a 1 or a 2 as a leading 0 can be ignored. Because we are using a loop we will loop around to the 2 later.

Forwards vs Backwards

The tree generated is moving backwards but we are interested in the tree moving forwards. We can get the forward loop by starting at the bottom of the tree and moving towards the top. So we must reverse the segments to get:

1#->2#->1#->(1#) (Represents the next segment starting with a 1#)

and

1#->2#->(1#) (Represents the next segment starting with a 1#)

Minimum Forward ascending segments.

Lets name Segment A as:

1#->2#->(1#)

Notice 2 divides and 1 shift caused by 1 leading 1# because (1#) is in the next segment.

Lets name Segment B as:

1#->2#->1#->(1#)

Notice 3 divides and 2 shifts caused by 2 leading 1# because (1#) is in the next segment.

Substitutions for 3n+1.

Because we are looking at either ascent or descent and never directly at equivalence the true value of 3n+1 is not required. Instead when we claim something ascends we will use a smaller analog of 3n+1 to prove that even when 3n is used it still ascends so that adding an extra +1 makes it ascend even more.

When Considering descending things get a little more complex.

A great many numerical n values have been computed and shown to descend to 1. The sequence at the time of this proof was n < 2^60 had been shown to descend to 1. When considering descend we will restrict ourselves to n > 1000. The reason is that 3.001n can be substituted for 3n+1 and 3.001n is always larger than 3n+1 provided n>1000.

Because both of these substitutions for 3n+1 replace a function with a multiplication we are able to move the contribution of our 3n+1 analog to be part of which ever segment reduced the digit that the shift is restoring. This allows segments to be evaluated accurately as ascending or descending without knowledge of other segments.

Evaluating specific sequences A.

The ratio for segment A has 2 Divides and 1 Shifts and descends.

This can be clearly seen if we consider dividing by a factor of 2^2=4 and ascending is only by a factor of 3 or 3.001.

Evaluating specific sequences B.

The ratio for Segment B is 3 Divides and 2 Shifts and ascends.

This can be clearly seen if we consider dividing by a factor of 2^3=8 and ascending is by a factor of 9 or slightly greater.

Evaluating specific sequences AB.

Now consider the sequence AB is the most ascending sequence we can create without putting 2 Bs in a row. Order doesn't matter ABAB… or BABA… they all create the same result 1 Bs in between every pair of As.

The ratio for AB is
3 Divides and 2 Shifts From Segment B
plus
2 Divides and 1 Shifts From Segment A

Which is 5 Divides and 3 Shifts

This gives
divide by 2^5= 32 and
multiply by 3.001^3= ~27.027 and descends in the forward direction.

Evaluating specific sequences ABB.

Now consider the sequence ABB..ABB…ABB is the most ascending sequence we can create without putting 3 Bs in a row. Order doesn't matter ABB…ABB or BAB…BAB or BBA…BBA they all create the same result 2 Bs in between every pair of As.

The ratio for ABB is
3 Divides and 2 Shifts From Segment B
plus
3 Divides and 2 Shifts From Segment B
plus
2 Divides and 1 Shifts From Segment A

Which is 8 Divides and 5 Shifts

This gives
divide by 2^8= 256 and
multiply by 3.001^5 = 243.405 and descends in the forward direction.

Adding an extra B to ABB…ABB… must produce at least one BBB. For n above 1000, any loop that does not contain at least one stretch of segments of BBB must descend.

The Challenge

We now face a challenge. Below is ABBB and we must figure out additional digits from the known information which is only the most significant digit from the segments A(1# at 1/1, 2# at 2/2) , B(1# at 2/3, 2# at 3/4, 1# at 3/5), B(1# at 4/6, 2# at 5/7, 1# at 5/8), B(1# at 6/9, 2# at 7/10, 1# at 7/11) and the next segment will always have 1# (at 8/12) ->2# (at 9/13)->?

Notice the digit loss after every 1 is divided.

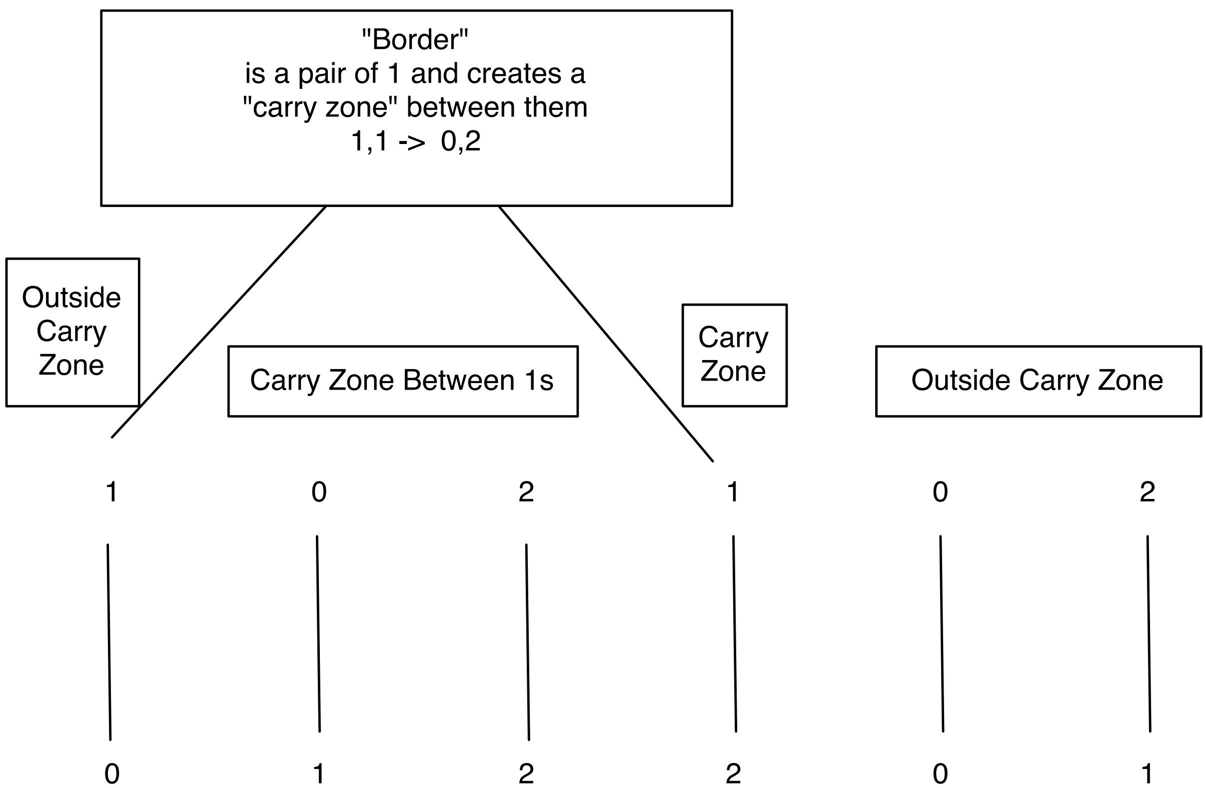
This requires a set of 14 rules derived from CLOSE and CAREFUL consideration of the divide by 2 operation in base 3. So we need to stop and study the divide by 2 operation carefully.

Column	0	1	2	3	4	5	6	7	8	9
Row 0	0 or 1	#	#	#	#	#	#	#	#	#
Row 1		1	#	#	#	#	#	#	#	#
Row 2			2	#	#	#	#	#	#	#
Row 3			1	#	#	#	#	#	#	#
Row 4				2	#	#	#	#	#	#
Row 5				1	#	#	#	#	#	#
Row 6					1	#	#	#	#	#
Row 7						2	#	#	#	#
Row 8						1	#	#	#	#
Row 9							1	#	#	#
Row 10								2	#	#
Row 11								1	#	#
Row 12									1	#
Row 13										2

Graphically showing base 3 divide by 2 and multiply by 2 relationships

Here is a graph showing top before and bottom afterwards in the Forward Direction.

1 divided by 2 is 0 remainder 1 which starts a carry zone by carrying the remainder down. (0+3)/2=1 and carried down. (2+3)/2=2 and carries down. 1+3 divided by 2 is 2 but ends a carry zone and does not carry down. 0 divided by 2 is 0 without carry down. And 2 divided by 2 is 1 without Carry down.



In 102102 Pay Careful attention to the definition of Carry Zone 1**021**02 and non Carry Zone **1**021**02.** Next page lists the Rules and then we will review the base 3 math.

Prime Rule

- No value can be written into the table unless it is the ONLY value that can be written in that cell because all other values cause a violation disproving the possibility of the sequence existing.

- The exception is Forcing statements where we try (unsuccessfully) and force something that will be proven False for example trying to force the first A in AABBB to exist only to show it is impossible. This will be explained more later.

- “If” statements are used when a single Assign is used and there after If# is filled in instead of just # and once falsified all the If#s are deleted and the first single cell usually only has 2 possible values and 1 has been disqualified via the IF process so the other can be “chosen” as it is the only legal move remaining. This will be explained more later.

Rule 0

- When dividing a known top Digit number in a known Carry / Not Carry zone state just use the truth table. Top->Bottom
- Carry 0->1, 1->2, 2->2
- Non-Carry Zone 0->0, 1->0, 2->1

Rule 1

- The 1 below a carry zone is always under a 0

Rule 2

- The 1 below a Non Carry zone is always under a 2

Rule 3

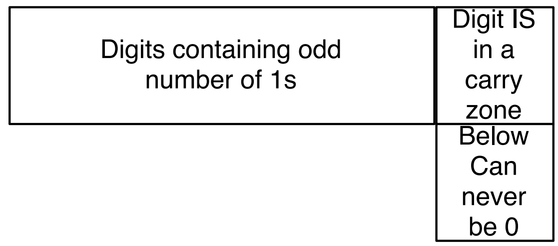
- Carry Zone cannot produce / have a 0 under it.
- 0 becomes 1 in next number
- 2 becomes 2 in next number
- 1 becomes 2 in next number
- NOTHING in a Carry Zone produces 0 in next number

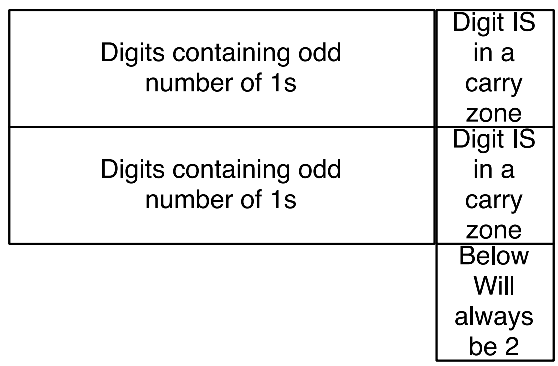
Rule 4

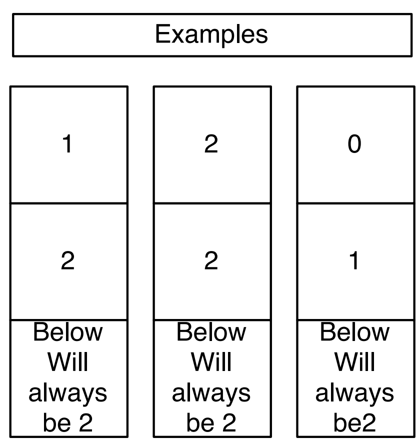
- Non-Carry Zone can never have a 2 under it
- 2 becomes 1 in next number
- 0 becomes 0 in next number
- 1 becomes 0 in next number
- NOTHING in a Non-Carry Zone produces 2 in next number

Rule 5

- Under 2 Carry zones is always 2
- Consider Each case 1,2,0
- Top (1+3)/2 =Middle=2
- Middle (2+3)/2 =Bottom=2
-
- Top (2+3)/2 =Middle=2
- Middle (2+3)/2 =Bottom=2
-
- Top (0+3)/2 =Middle=1
- Middle (1+3)/2 =Bottom=2

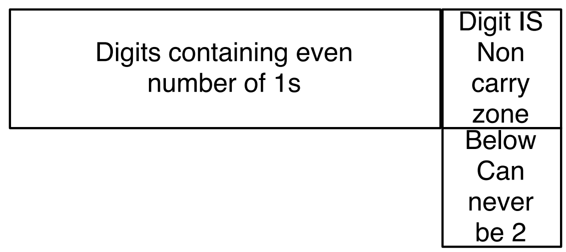


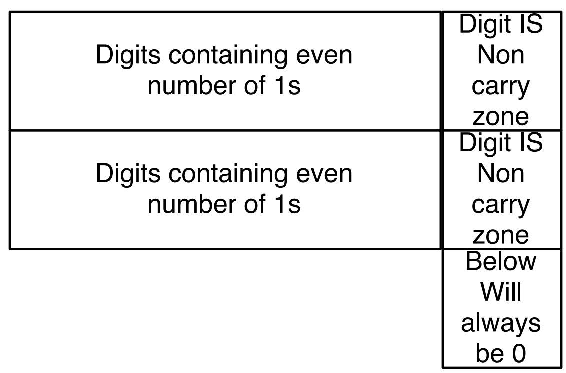


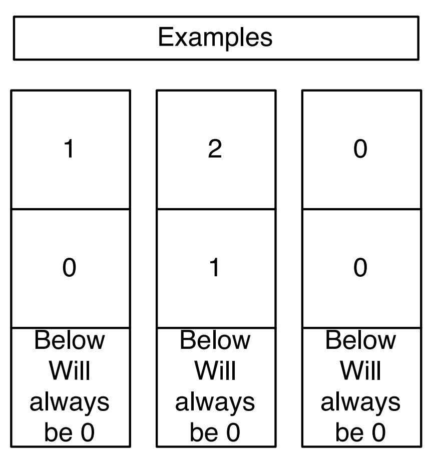


Rule 6

- Under 2 Non-Carry Zones is always 0
- Consider Each case 1,2,0
- Top (1+0)/2 =Middle=0
- Middle (0+0)/2 =Bottom=0
- Top (2+3)/2 =Middle=2
- Middle (2+3)/2 =Bottom=2
- Top (0+3)/2 =Middle=1
- Middle (1+3)/2 =Bottom=2







Rule 7

- If two 0s under a Non-Carry zone then the left most 0 has a 0 above it.

- The square above the left 0 could be 0 or 1 ordinarily

- But if it was a 1 it would make a Carry Zone and the right 0 would be under a Carry Zone violating Rule 1

IF 1 above first 0 it would make carry zone that is violated by being above 2nd 0

Digits containing even number of 1s	Must be 0	Could be 1 or 0
	If 0	And 0

Rule 8

If two 2s under a Carry zone then the left most 2 has a 2 above it.

IF 1 above first 2 it would make non carry zone that is violated by being above 2nd 2

Digits containing odd number of 1s	Must be 2	Could be 1 or 2
	If 2	And 2

The square above the left 2 could be 1 or 2 ordinarily

But if it was a 1 it would make a Non-Carry Zone and the right 2 would be under a Non Carry Zone violating Rule 2

Rule 9

			Proof		
Digits containing even number of 1s AKA Non-Carry Zone	Then 2		1	2	0
Digits containing odd number of 1s AKA Carry Zone	Then 1		0	1	0
	If 2		1	2	1

- If a 2 is directly under a Carry Zone that is Under a Non Carry Zone then the Carry Zone must be 1 and the Non Carry Zone 2
- This is because if top was 0
- 0 divided by 2 is 0 and 0+3 divided by 2 is 1 not 2
- This is because if top was 1
- 1 divided by 2 is 0 and 0+3 divided by 2 is 1 not 2

Rule 10

		Examples		
Digits containing Odd number of 1s AKA Carry Zone	Then 0	1	2	0
Digits containing even number of 1s AKA Non-Carry Zone	Then 1	2	2	1
	If 0	1	1	0

- If a 0 is directly under a Non-Carry Zone that is Under a Carry Zone then the Non-Carry Zone must be 1 and the Carry Zone 0

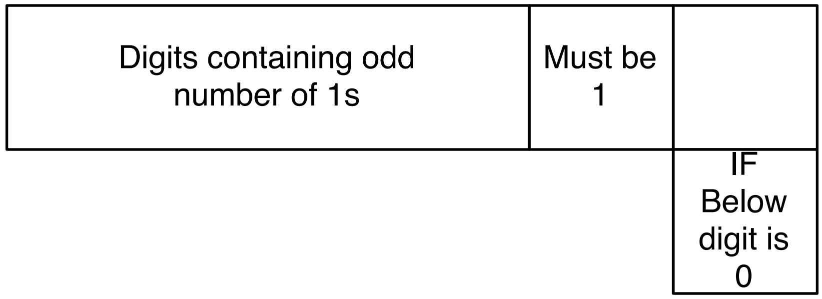
- This is because if top was 1

- 1+3 divided by 2 is 2 and 2+0 divided by 2 is 1 not 0

- This is because if top was 2

- 2+3 divided by 2 is 2 and 2+0 divided by 2 is 1 not 0

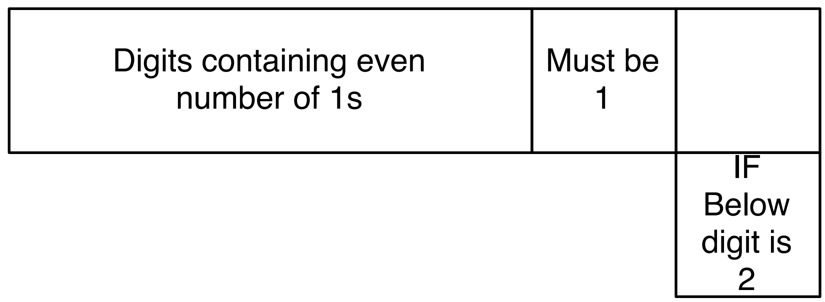
Rule 11



- If a Carry zone has 2 empty spaces and a 0 is below the second one the only viable move is to put 1 into the space before the one above the 0 to “Block” the violation of Rule 1 by ending the Carry zone

- We must always move to avoid violations

Rule 12



- If a Non-Carry zone has 2 empty spaces and a 2 is below the second one the only viable move is to put 1 into the space before the one above the 2 to “Block” the violation of Rule 2 by ending the Non-Carry zone

- We must always move to avoid violations

Rule 13

Digits containing odd number of 1s	Cannot Exit Non Carry	Cannot Exit Non Carry	May or May not exit Carry Zone
	Given 1	Given 1	Given 2

Digits containing odd number of 1s	Violation of Rule 1
	Given 0

- Exiting of Carry zone can only be achieved if the square below it is empty or a 2.
- If it is full this indicates that some other move to avoid Violation required that value

Rule 14

Digits containing even number of 1s	Cannot Exit Non Carry	Cannot Exit Non Carry	May or May not exit Carry Zone
	Given 1	Given 1	Given 0

Digits containing even number of 1s	Violation of Rule 2
	Given 2

- Exiting of Non-Carry zone can only be achieved if the square below it is empty or a 0.

- If it is full this indicates that some other move to avoid Violation required that value

Rule 15

Digits containing odd number of 1s	Can be 0 or 2	Must be 1	Violation of Rule 1 unless past block is 1
	Given 2 is not possible	Given Possibility of being 2 Then MUST be 2	Given 0

- Non-Localized Blocks occur when 1 or more blocks are unknown but the underblocks are known to NOT be 2 in some place(s) and then to potentially be 2 and then Definitely 0

Rule 16

Digits containing even number of 1s	Must be 2	Must be 1	Violation of Rule 2 unless past block is 1
	Given 0 is not possible	Given Possibility of being 0 Then MUST be 0	Given 2

- Non-Localized Blocks occur when 1 or more blocks are unknown but the underblocks are known to NOT be 0 in some place(s) and then to potentially be 0 and then Definitely 2

Rule 17

Digits containing odd number of 1s	Violation of Rule 1 unless under block is 1
Digits containing even number of 1s	Must be 1
	Given 0

- The cell under a Carry zone must be 1 if there is a non-Carry zone under that cell with the value 0 in it.

Rule 18

Digits containing even number of 1s	Violation of Rule 2 unless under block is 1
Digits containing odd number of 1s	Must be 1
	Given 2

- The cell under a Non-Carry zone must be 1 if there is a Carry zone under that cell with the value 2 in it.

Base 3 Math Review

Because the shift operations does not change any digits except the leading 0 and least significant digit it allows us to do math even if we only have partial knowledge regarding the number we are doing math on.

Now we will look at the multiply by 2 and divide by 2 tables in base 3 and review to help internalize the Rules.

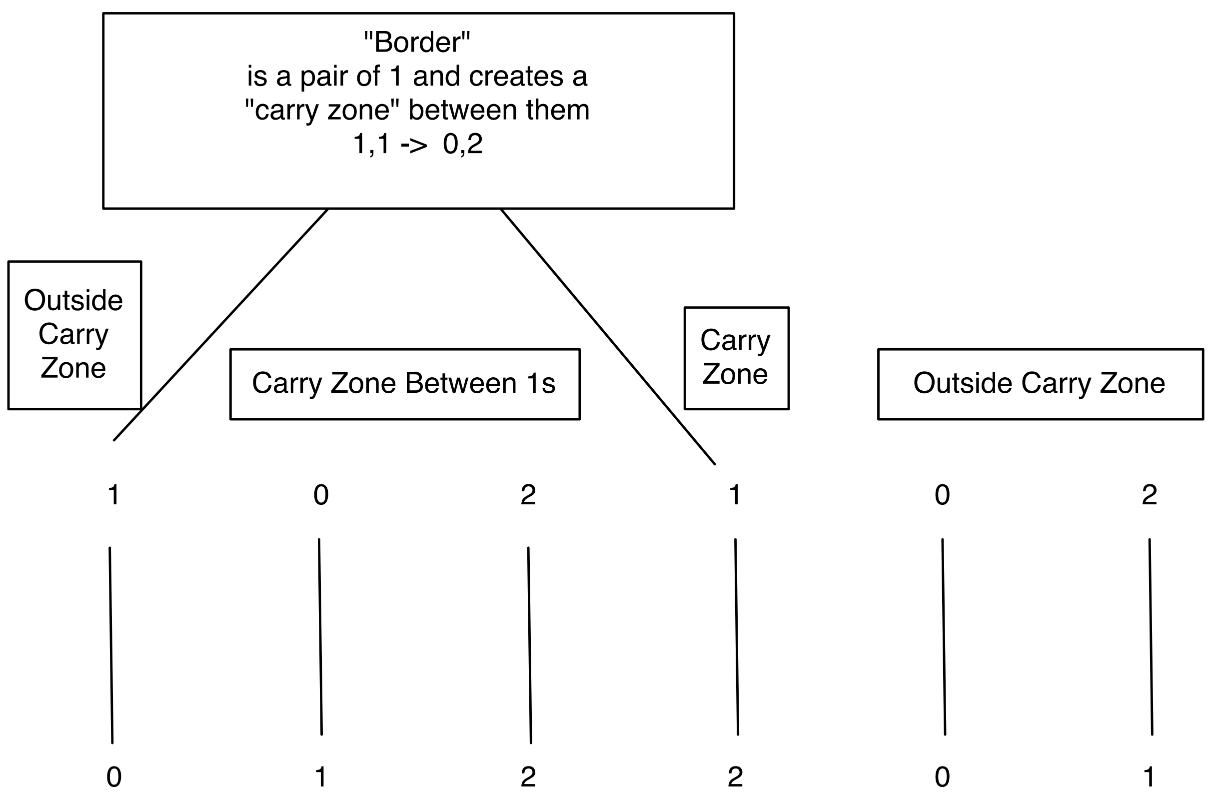
Dividing by 2	**In base 3**			
Line	Value of remainder from Digit one higher carries in and adds 3	Digit in location A at T=0	Digit in location A at T=1	Remainder
1	0	0	0	0
2	0	1	0	1
3	0	2	1	0
4	3	0	1	1
5	3	1	2	0
6	3	2	2	1

Multiplying by 2	**In Base 3**			
Line	Value if remainder from Digit one lower carries in and adds 3 which is a 1 to this digit	Digit in location A at T=1	Digit in location A at T=0	Carry up to next significant digit
1	0	0	0	0
2	0	1	2	0
3	0	2	1	3
4	1	0	1	0
5	1	1	0	3
6	1	2	2	3

Base 3 relationships

Graphically showing base 3 divide by 2 and multiply by 2 relationships

Here is a graph showing top before and bottom afterwards. Starting left 1 divided by 2 is 0 remainder 1 which starts a carry zone by carrying the remainder down. (0+3)/2=1 and carried down. (2+3)/2=2 and carries down. 1+3 divided by 2 is 2 but ends a carry zone. 0 divided by 2 is 0. 2 divided by 2 is 1. These are the only possible digit operations.



Method of disproving specific sequences

Below we will take long sequences that are identified only by the first digit for each step. This will be shown in the forward direction from top to bottom. All shifts will be hidden from view inside the # which starts off to the right of the tables shown. The reason for this is that when a sequence is divided it is MUCH easier and more natural to keep the digits in the correct relative positions

```
  121#######
  022#######
```

VS

```
  121#######
022#######*
```

*=shift

In the first case you can clearly see that the leading 1 became a 0 and the new leading digit is the 2 to the right accordingly. The lines of causation are straight up and down.

In the second case the same lines of causation connect the same digits but the shift that occurred has made the lines of causation diagonal and it is much harder to understand what is going on after a few iterations.

As a more practical consideration we do not know where the shifts are and could not place them because of this. Leading 1s do predict how many shifts occur but they do not predict where the shifts occur. So please find the digits arranged in the tables below so that the lines of causation run straight up and down. Also note that to do this digits need to proceed from top left to bottom right as divides decrease the digit count among the knowable digits.

First prove AA is impossible in forward direction

Consider AAX where A= (1#, 2#) and X = (1#, 2#, 1#) and X could be another A and the start of another segment or X could be a full B but either way the sequence 1#->2#->1# will occur directly after AA

The top is Earlier and we divide by 2 moving from row x -> row x+1

Column	0	1	2	3	4	5	6	7	8
Row 0									
Row 1	(start A)	1							
Row 2	(end A)		2						
Row 3	(Start A)		1						
Row 4	(End A)			2					
Row 5	(Start X)			1					
Row 6	(middle X)				2				
Row 7	(End X)				1				
Row 8									
Row 9									
Row 10									
Row 11									
Row 12									

Location Column / Row

Column	0	1	2	3	4	5	6	7	8
Row 0									
Row 1	(start A)	1							
Row 2	(end A)		2	2					
Row 3	(Start A)		1	1					
Row 4	(End A)			2	2				
Row 5	(Start X)			1	1				
Row 6	(middle X)				2				
Row 7	(End X)				1				
Row 8									
Row 9									
Row 10									
Row 11									
Row 12									

Location Column / Row

3/3=1 and 3/2=2 because of Rule 9

4/5=1 and 4/4=2 because of Rule 9

Notice that cell 4/3 is a non Carry zone because it has zero 1s to the left of it and 0 is even.

Cell 4/2 is a non carry zone above a 2 which is illegal.

We have violated Rule 4 and this configuration is impossible.

QED segments AA can never occur in order.

Contradiction???

The only known loop is 1->4->2->1 in base 10 math.

In Base 3 this becomes

1->11->2->1

And if we write it without the shift operation it becomes

11->2->11->2->11

This is in fact an A segment in a loop and one may argue that this disproves the proof.

But what has happened is that the # has been emptied in 1->4->2->1 and this is expressly outside of the envelop we have taken for this proof.

This proof only works for:
n>1000
sufficient to hold up to 30-40 digits so that Shifts cannot affect the leading number digits within any of the calculations.

In the known loop we clearly “gained” information in the forward direction when we divided from 2 and got 11 instead of 1. This was only possible because the # overflowed. In the backwards direction we multiplied 11 by 2 and got 02 again an operation that can only work if the effect of the shift has affected the sequence which violates the rules we are using and the cases we are considering. Our cases are sufficient to include ALL unknown loops because all known loops have large n values.

Why is that???

There are two reasons:

1) No number in any possible future undiscovered loop is smaller than 2^60

The reason for this is that any number smaller than 2^60 descends to 1 and is disqualified as per past proofs (insert references to research here). So no matter what operation we do on a loop candidate we can never reach a number smaller than 2^60 because those numbers are ALL disqualified from being in a loop by virtue of having already been numerically proven to descend to 1.

2) Because we are only relying on n>1000 we can put the remainder of the digits into the # in 1# or 2#

So we will not burn through the # in any of our ~20-25 divide long operations. At the end of that operation we will still ALWAYS have an n above 2^60. So we can do multiple sequences that are each 20-25 operations long but we must “forget” the number and start over from the leading digits when we do that. The “forgotten” digit will again be larger than 2^60 and we can proceed to do another 30-40 operations safely and we can maintain the leading digit because it have not been affected. But we must forget the discovered digits if we perform more than some number of digits larger than 30.

Implications…..

Segment A and Segment B are distinguished in that between 2s there can be one or two 1s. If there is one 1 between 2s the Segment is A. If there are two 1s between 2s the Segment is B.

But now we have just proven that there must be some number of Segment Bs in-between a pair of As. This forms the same type of pattern and if we can constrict the length of Bs possible it would greatly limit the sequences of As and Bs possible.

Now consider a sequence of As and Bs. At the end of this sequence there will ALWAYs be 1#->2#->1# which I call X. We do not know if X is a whole B segment or a whole A segment with the start of either an A or B segment attached. But regardless of our segment there will always be X of 1#->2#->1# at the end. If the digit after X is 1 then we know X was a complete B segment if the digit after X is 2 then we know X was A plus the leading 1 of a A segment.

For some super segments we will consider putting an X before the segment and looking at only the last digit of X because if it is 1 then X is a B segment and if it is 0 then the right most digit above the leading 1 must be a 2 and X is an A segment.

I have labeled A- for the start of A and A+ for the end of an A segment so the segment is easier for new mathematicians to read and the poof is easier to follow.

Prove BBBB impossible

Column	0	1	2	3	4	5	6	7	8	9	10	11
Row 0												
Row 1	B-	1										
Row 2			2									
Row 3	B+		1									
Row 4	B-			1								
Row 5					2							
Row 6	B+				1							
Row 7	B-					1						
Row 8							2					
Row 9	B+						1					
Row 10	B-							1				
Row 11									2			
Row 12	B+								1			
Row 13	X-									1		
Row 14											2	
Row 15	X+										1	

Location Column / Row

Column	0	1	2	3	4	5	6	7	8	9	10	11
Row 0												
Row 1	B-	1										
Row 2			2									
Row 3	B+		1	0								
Row 4	B-			1								
Row 5					2							
Row 6	B+				1	0						
Row 7	B-					1						
Row 8							2					
Row 9	B+						1	0				
Row 10	B-							1				
Row 11									2			
Row 12	B+								1	0		
Row 13	X-									1		
Row 14											2	
Row 15	X+										1	

Location Column / Row

3/3=0, 5/6=0, 7/9=0, 9/12=0 All because of Rule 1

Column	0	1	2	3	4	5	6	7	8	9	10	11
Row 0												
Row 1	B-	1										
Row 2			2									
Row 3	B+		1	0								
Row 4	B-			1								
Row 5					2							
Row 6	B+				1	0						
Row 7	B-					1						
Row 8							2					
Row 9	B+						1	0	If 0	If 0	X	
10	B-							1	IF 1	If 1	IF 0	
11									2	If 0	If 1	
12	B+								1	0	IF 0	
13	X-									1	IF 1	
14											2	
15	X+										1	

Location Column / Row

If 10/13=1, Then 10/12=0 Rule 1, 9/11=0 Rule 7, 8/10=1 Rule 11, 8/9=0 Rule 1, 9/10=1 Rule 11, 9/9=0 Rule 1, 10/11=1 Rule 17, 10/10=0 Rule 1, Violation at 10/9 of Rule 3 so 10/13!=2 and Using Rule 0 we know it must be 1 if it isn't 2.

Column	0	1	2	3	4	5	6	7	8	9	10	11
Row 0												
Row 1	B-	1										
Row 2			2									
Row 3	B+		1	0								
Row 4	B-			1								
Row 5					2							
Row 6	B+				1	0	IF 0	If 0	X			
Row 7	B-					1	If 1	IF 1	IF 0			
Row 8							2	If 0	IF 1			
Row 9	B+						1	0	IF 0			
10	B-							1	IF 1			
11									2			
12	B+								1	0		
13	X-									1	2	
14											2	
15	X+										1	

Location Column / Row

10/13=2 from last slide;

If 8/10=1 then 8/9=0 Rule 1, 7/8=0 Rule 7, 6/7=1 Rule 11, 6/6=0 Rule 1, 7/7=1 Rule 11,7/6=0 Rule 1, 8/8=1 Rule 17, 8/7=0 Rule 1, Violation at 8/6 so 8/10=2

Column	0	1	2	3	4	5	6	7	8	9	10	11
Row 0												
Row 1	B-	1										
Row 2			2									
Row 3	B+		1	0	If 0	If 0	X					
Row 4	B-			1	If 1	If 1	IF 0					
Row 5					2	IF 0	If 1					
Row 6	B+				1	0	IF 0					
Row 7	B-					1	IF 1					
Row 8							2					
Row 9	B+						1	0				
10	B-							1	2			
11									2			
12	B+								1	0		
13	X-									1	2	
14											2	
15	X+										1	

Location Column / Row

8/10=2 from last slide;

If 6/7=1 then 6/6=0 Rule 1, 5/5=0 Rule 7, 4/4=1 Rule 11, 5/4=1 Rule 1, 6/5=1 Rule 17, 6/4=0 Rule 1, 4/3=0 Rule 1, 5/3=0 Rule 1, Violation at 6/3 of Rule 3

so 6/7!=1 and 6/7=2 from Rule 0

Column	0	1	2	3	4	5	6	7	8	9	10	11
Row 0												
Row 1	B-	1										
Row 2			2									
Row 3	B+		1	0								
Row 4	B-			1								
Row 5					2							
Row 6	B+				1	0						
Row 7	B-					1	2	0				
Row 8							2	1				
Row 9	B+						1	0				
10	B-							1	2	0		
11									2	1		
12	B+								1	0		
13	X-									1	2	
14											2	
15	X+										1	

Location Column / Row

6/7=2 from last slide;

7/8=1 Rule 17,

7/7=0 Rule 1

9/11=1 Rule 17,

9/10=0 Rule 1

Colu mn	0	1	2	3	4	5	6	7	8	9	10	11
Row 0												
Row 1	B-	1										
Row 2			2									
Row 3	B+		1	0								
Row 4	B-			1								
Row 5					2							
Row 6	B+				1	0						
Row 7	B-					1	2	0	X			
Row 8							2	1	0			
Row 9	B+						1	0	1			
10	B-							1	2	0		
11									2	1		
12	B+								1	0		
13	X-									1	2	
14											2	
15	X+										1	

Location Column / Row

8/9=1 Rule 11,

8/8=0 Rule 1 Causing Violation at 8/7 of Rule 3

This violation means that BBBB cannot exist because it requires a “move” that is mathematically impossible. Carry zones cannot produce 0 in the next iteration.

Prove ABA impossible

Colu mn	0	1	2	3	4	5	6	7	8	9	10	11
Row 0												
Row 1	A-				1							
Row 2	A+					2						
Row 3	B-					1						
Row 4							2					
Row 5	B+						1					
Row 6	A-							1				
Row 7	A+								2			
Row 8	X-								1			
Row 9										2		
10	X+									1		
11												
12												
13												
14												
15												

Column	0	1	2	3	4	5	6	7	8	9	10	11
Row 0												
Row 1	A-				1							
Row 2	A+					2	2					
Row 3	B-					1	1					
Row 4							2					
Row 5	B+						1	0				
Row 6	A-							1				
Row 7	A+								2	2		
Row 8	X-								1	1		
Row 9										2		
10	X+									1		
11												
12												
13												
14												
15												

Location Column / Row

6/2=2 and 6/3=1 from Rule 9

9/7=2 and 9/8=1 from Rule 9

7/5=0 from Rule 1

Column	0	1	2	3	4	5	6	7	8	9	10	11
Row 0												
Row 1	A-				1							
Row 2	A+					2	2					
Row 3	B-					1	1					
Row 4							2	0	2			
Row 5	B+						1	0	1			
Row 6	A-							1	2			
Row 7	A+								2	2		
Row 8	X-								1	1		
Row 9										2		
10	X+									1		
11												
12												
13												
14												
15												

Location Column / Row

8/6=2 Because Rule 8

7/4=0 Because Rule 6

8/5=1 Because Rule 18

8/4=2 Because Rule 2

Column	0	1	2	3	4	5	6	7	8	9	10	11
Row 0												
Row 1	A-				1							
Row 2	A+					2	2					
Row 3	B-					1	1					
Row 4							2	0	2	X		
Row 5	B+						1	0	1	2		
Row 6	A-							1	2	1		
Row 7	A+								2	2		
Row 8	X-								1	1		
Row 9										2		
10	X+									1		
11												
12												
13												
14												
15												

Location Column / Row

9/6=1 because Rule 18

9/5=2 because Rule 2

Violation at 9/4 because of Rule 4

Progress so far…

At this point we could define super segments that must make up a loop and it would look like.

We proved that Segments

1

2

are the only possible leading digit

Then proved that

A=1->2

B=1->2->1

Are the only two segments that can be in any composite segment loop.

Then we proved that the only segments that can be in a composite loop are

ABB(A)

ABBB(A)

This means that according the loop length must be

ABB=1->2->1->2->1->1->2->1->

Which is 8 Divides "->"

And 5 Non local shifts "1"

Total Length of 13

ABBB=1->2->1->2->1->1->2->1->1->2->1->

Which is 11 Divides “->”

And 7 Non local shifts “1”

Total Length of 18

Total loop true length is then:

13x+18y

Consider #01 and #2

In a loop there must always be a #2 number or a #01 number somewhere in the loop if it has 3 steps or more. Here # represents the larger digits and 2 is the least significant digit not to be confused with 2# where # was the smaller digits.

Sub-Proof

Because of the 3n+1 and divide by 2 the right most digit can never be 0. This is because after a 3n+1 moving forwards the digit is always 1. Dividing the 1 produces a 2 and dividing the 2 again provides a 1. There is no pathway back to a 0. Because the sequence is infinite just advance past the first shift and name that new number the "start" of the sequence.

QED #0 base 3 cannot be the right most digit.

Now lets eliminate all #2 numbers as they satisfy our claim already. Consider all possible 2 digit rightmost digits remaining after removing #01 which satisfies our claim:

#11 and #21 remain.

Consider #11 base 3:

If the next operation moving backwards is multiply then #11->#22 satisfies the #2 case.

If the next operation is unshift then the move afterwards must be a multiply.

#11->#1->#2 Which satisfies the case #2.

Consider #21 base 3:

If the next operation moving backwards is multiply then #21->#12 satisfies the #2 case.

If the next operation is unshift then #21->#2 satisfies the case #2.

QED a loop of more than 3 steps always contains either #2 or #01.

Branching Backwards

In considering an unknown number we don't know if the number is even or not and the expectation is that every step forward will branch into 2 possible numbers due to uncertainty on wether the number is even or odd. Surprisingly taking steps backwards in time branches far less frequently. This becomes clear when we consider things that are impossible moving backwards.

First

#0

Is an impossible value except for the initial seed number that starts a sequence. We have agreed to advance past the first 3n+1 and call this the seed number to eliminate #0 as a possibility.

Second

#N1

Unshifts to

#N

But

#N2

Cannot unshift because shifts always put a 1 in the right most digit and a 2 means the previous operation must have been a multiply/divide.

Third

#N01

Cannot unshift because to do so would put a 0 in the right most position.

Again this is true for loops and all but the first number of a infinite sequence(which can be converted via stepping past the first 3n+1 into sequence that always obeys this rule.

Fourth

It is impossible to have two shifts back to back. This is because 3n+1 is always even so moving forward the next operation must be a divide and because 3n+1 is not a divide the operation before it cannot be 3n+1.

Examples:

#02

Must be a reverse divide because it cannot be a shift as right most value is not 1.

Because we are doing a reverse division we can multiply from the right by 2

2*2=4 which is 11 in base 3.

So we can see the previous value was

#11

Consider

#101

Here we have a 1 on the right most digit and it may look like we can reverse shift but the 0 makes that impossible because we would have a 0 in the right most digit. So we must multiply by 2.

#202

Because of the right most 2 we cannot unshift and must multiply by 2

#111

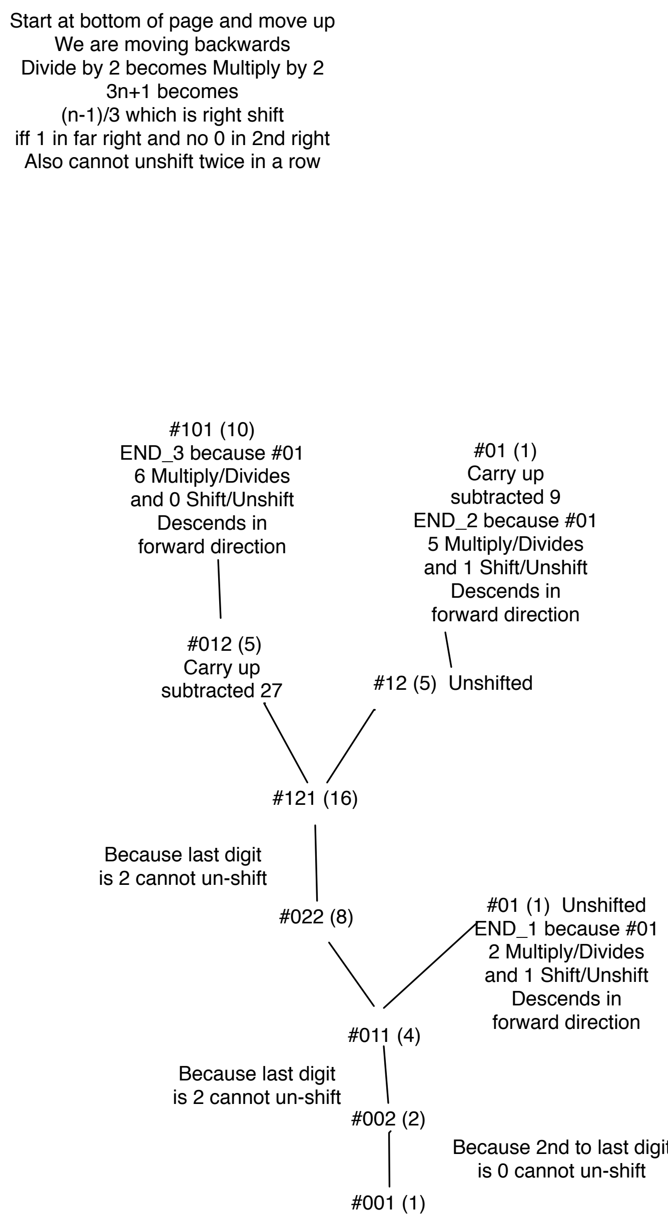
This is the end of examples

All but One Path Descend

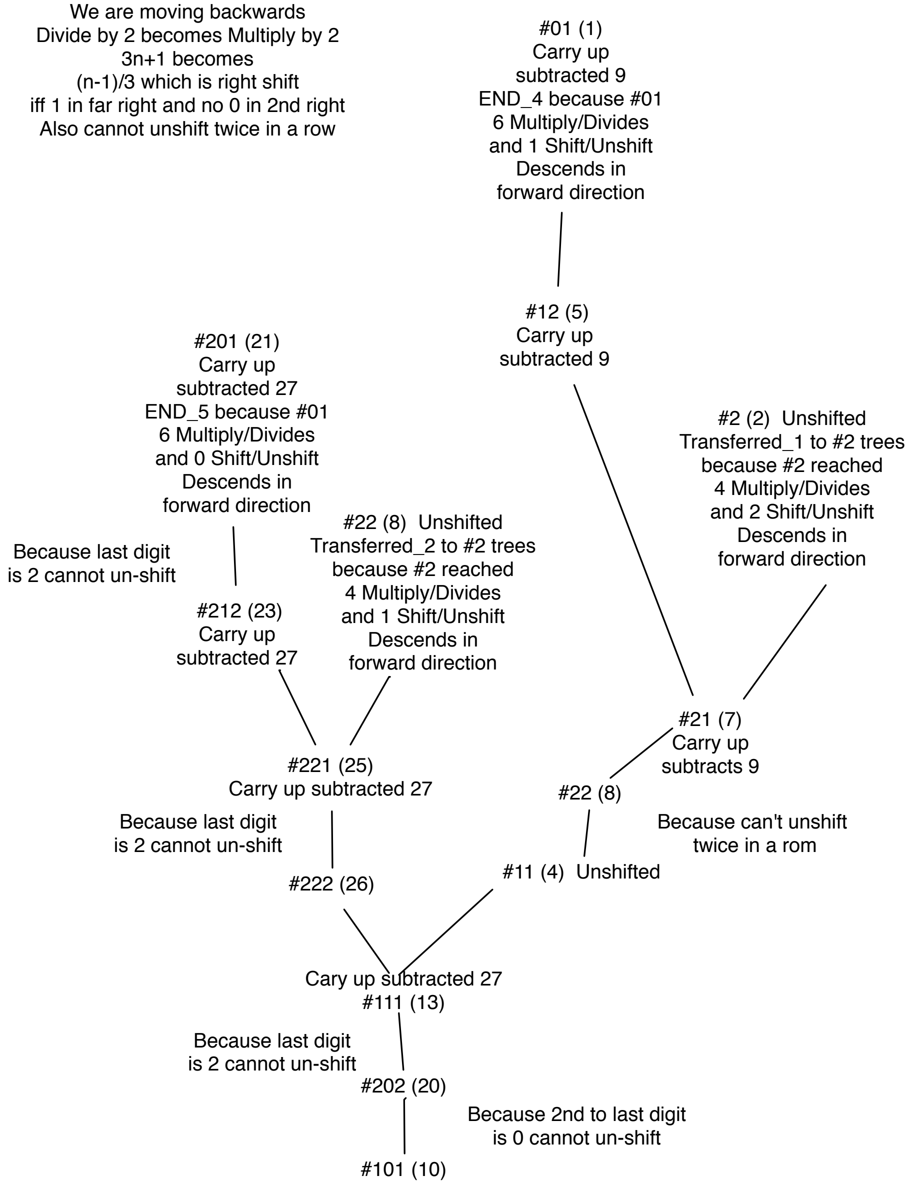
Consider the examples below were all the pathways except One descend when stepping in the positive direction. #01 is replaced by #001, #101 or #201 and #2 is replaced by #02, #12, or #22.

The trees however are created by stepping in the negative time direction so that multiply by 2 and (n-1)/3 are used along with the logic described in branching backwards.

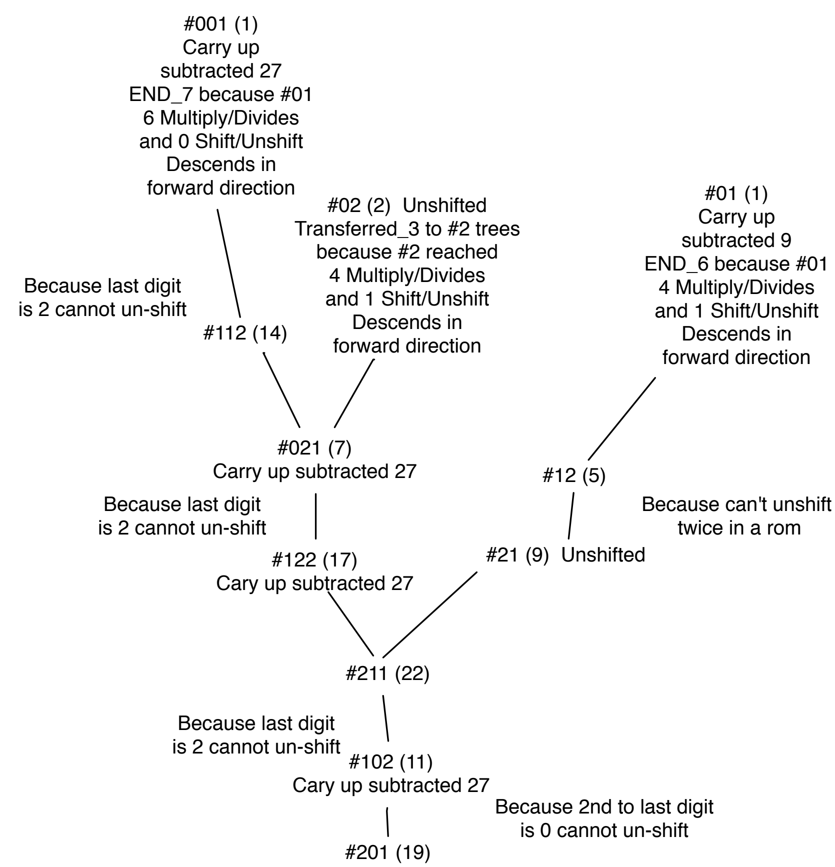
To help show this the move trees will start at the bottom and extend backwards so that they can easily be checked by stepping forwards from top to bottom.



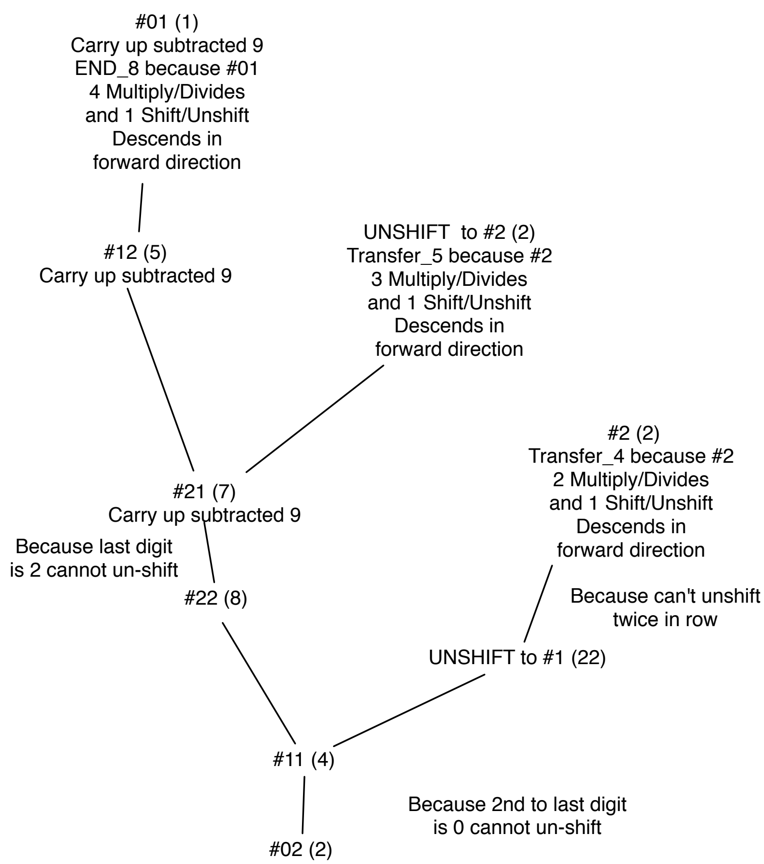
Start at bottom of page and move up
We are moving backwards
Divide by 2 becomes Multiply by 2
3n+1 becomes
(n-1)/3 which is right shift
iff 1 in far right and no 0 in 2nd right
Also cannot unshift twice in a row



Start at bottom of page and move up
We are moving backwards
Divide by 2 becomes Multiply by 2
3n+1 becomes
(n-1)/3 which is right shift
iff 1 in far right and no 0 in 2nd right
Also cannot unshift twice in a row



Start at bottom of page and move up
We are moving backwards
Divide by 2 becomes Multiply by 2
3n+1 becomes
(n-1)/3 which is right shift
iff 1 in far right and no 0 in 2nd right
Also cannot unshift twice in a row



Start at bottom of page and move up
We are moving backwards
Divide by 2 becomes Multiply by 2
3n+1 becomes
(n-1)/3 which is right shift
iff 1 in far right and no 0 in 2nd right
Also cannot unshift twice in a row

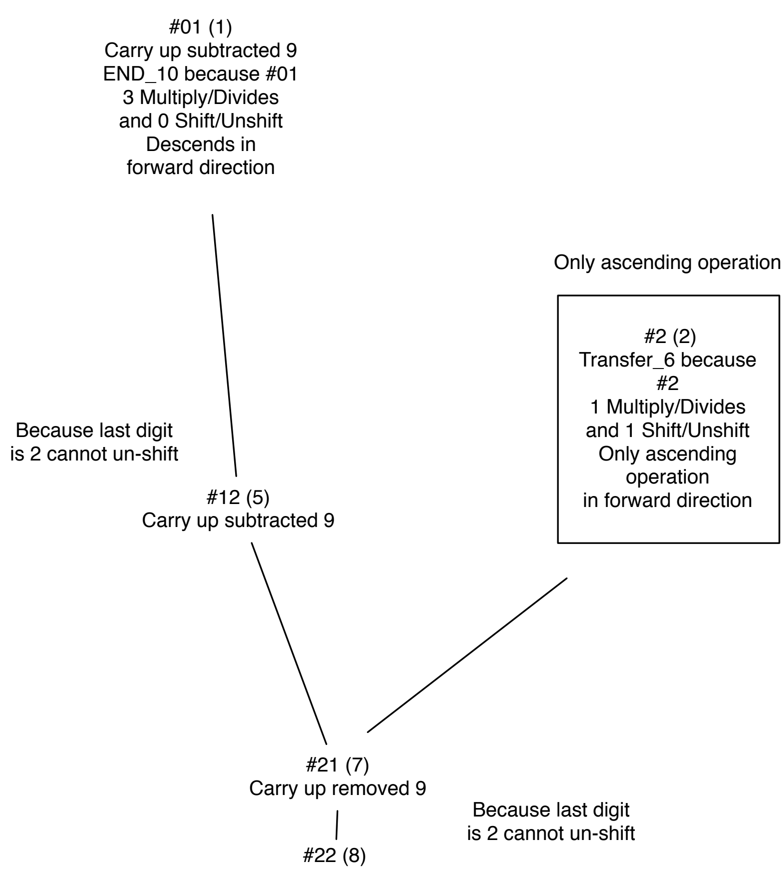
#01 (1)
Carry up subtracted 9
END_9 because #01
1 Multiply/Divides
and 0 Shift/Unshift
Descends in
forward direction

|

#12 (5)

Because last digit
is 2 cannot un-shift

Start at bottom of page and move up
We are moving backwards
Divide by 2 becomes Multiply by 2
3n+1 becomes
(n-1)/3 which is right shift
iff 1 in far right and no 0 in 2nd right
Also cannot unshift twice in a row



Using this method we can see that

#2->#1->#11->#22 only works if the start is 3#+2 and is the only sequence that ascends.

Lets consider this from a different angle

ConsiderBase 4

We will consider all the odd numbers just before 3n+1 for 2 digits of base 4 numbers each digit consisting of 0,1,2,3. Because the number is odd 1 or 3 must be the least significant digit as 0 and 2 are even.

Seed number in base 4	Carry up	Result of first 3n+1	Next seed number	Descends	Fed by
#01	0	#10	#01 or #11 or #21 or #31	Yes unconditionally	#01 *1 of 4, #31*1 of 4, #23*1 of 2, #11 1 of 8
#11	1	#00	Next 2 digits if even more divides will occur until odd	Yes unconditionally	#01 *1 of 4, #31*1 of 4, #03*1 of 2, #11 1 of 8

Seed number in base 4	Carry up	Result of first 3n+1	Next seed number	Descends	Fed by
#21	1	#30	#03, #13, #23 or #33	Yes unconditionally	#01 *1 of 4, #31*1 of 4, #23*1 of 2, #11 1 of 8
#31	2	#20	#01, #11, #21, or #31	Yes unconditionally	#01 *1 of 4, #31*1 of 4, #03*1 of 2, #11 1 of 8
#03	0	#22	#11 or #31 if above odd	Yes unconditionally when the next shift is considered	#21 *1 of 4, #13*1 of 2, #11 1 of 8
#13	1	#12	#03 or #23 if above even	Only if the number above is Odd prior to the shift	#21 *1 of 4, #03*1 of 2, #11 1 of 8
#23	2	#02	#01 or #21 if above odd	Only if the number above is even prior to the shift	#21 *1 of 4, #13*1 of 2, #11 1 of 8
#33	2	#32	#13 or #33 if above odd	No Unconditionally	#21 *1 of 4, #33*1 of 2, #11 1 of 8

Lets consider a few things:

1) #01, #11, #21, #31 all descend because 1 shift and 2, 4, 2, 3 divides respectively.

2) The average of all 8 possibilities descends as there are 8 shifts and at least 15 Divides. Using 3.001 for 3n+1 this is multiply by 6,578.52 vs divide by 32,768.

3) #03->#11->#00 or #03->#31->#20 both descend unconditionally with 4 or 5 divides and 2 shifts .

4) #13->#03->#11->#00 or #13->#03->#31->#20 descends conditionally if #03 path is taken with 5 or 6 divides and 3 shifts

5) #23->#01->#01, #11, #21 or #31->#X0 descends conditional if the #01 path is taken with 5 or more divides and 3 shifts

6) 5/8 pathways unconditionally descend, 2/8 pathways conditionally descend based on the number above being even or odd. The average descends. The numbers seem to rotate so that even though #33 does loop to itself in a stronger way that #01 loops to itself the net effect of #01 and #31 both looping to themselves half as often as #33 may even it out.

7) The only unconditionally ascending pathway is dependent on the polarity of the number above to maintain being #33 and when this is lost does not quickly return to this pattern but appears to visit many if not all of the other branches first.

This is not a proof that infinite ascent is impossible. Such a proof may not exist. But once the loop option has been removed it appears that infinite ascent is exceedingly unlikely.