Untitled

Jamie Ash

2022-10-29

To answer this question I felt like I needed an equation that describes how each natural number is produced. I began by handwriting out by calculating the 1 through 10 of $\in N$ to see if there was a pattern. I realized that the even numbers where reproduced by summing all possible combinations of 2^n

for $\{\mathbb{N} : 2^n\}$ which looks like $\{2, 4, 8, 16...\}$ I realized you can get all even numbers between n and $n+1$ by raising all possible subsets of 1 to n by two and summing them the individual subsets. Then the odd numbers can be found by adding 2^0 to the set of even numbers, and since 2^0 is not in the set $\{\mathbb{N} : 2^n\}$ it does not break the rules of the question.

```r
raise = function(x){
  sum(2^x)
  }

# first few natural numbers
n = 1:2
# produce the power set of n
sets = powerSetCond(n, rev=TRUE)
# raise by 2 and sum each subset of n (in the powerset)
evens = lapply(sets, raise)
# Just changing the data class to vector (from a list), and sriting them
evens = unlist(evens)
sort(evens)
```

```
## [1] 0 2 4
```

```r
# retrieveing all odd numbers
sort(evens + 2^0)
```

```
## [1] 1 3 5
```

I then realized that $\mathbb{N}$ can be found by individualy summing the subsets within the power set of $\{n \in \mathbb{N} : 2^n\}$ and also including 0. This is the same as adding 2^0 to $\{n \in \mathbb{N} : 2^n\}$

```r
# first few positive integers
z = 0:3
# produce the power set of z
sets = powerSetCond(z, rev=TRUE)
# raise by 2 and sum each subset of z (in the powerset)
nat_nums = lapply(sets, raise)
# Just changing the data class to vector (from a list), and sriting them
nat_nums = unlist(nat_nums)
sort(nat_nums)
```

```
##  [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14
```

Finally, if I go into the negative and sum the first few smallest (and largest negative) values in $\mathbb{Z}$, this would be like adding fractions to the set of $\{n \in \mathbb{N} : 2^n\}$. Like $\{z \in \mathbb{Z} : 2^z\}$. I realized I start getting the set of real numbers, $\mathbb{R}$.

```r
# first few natural numbers
z = -1:1
# produce the power set of n
sets = powerSetCond(z, rev=TRUE)
# raise by 2 and sum each subset of n (in the powerset)
nat_nums = lapply(sets, raise)
# Just changing the data class to vector (from a list), and sriting them
nat_nums = unlist(nat_nums)
sort(nat_nums)
```

```
## [1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0
```

As we go more negative we achieve a higher "resolution". It seems like for each integer into the negative we go, one more decimal place is retrieved. For $z \in Z$ where $-2 \leq z \leq 1$ we get…

```r
# first few natural numbers
z = -2:1
# produce the power set of n
sets = powerSetCond(z, rev=TRUE)
# raise by 2 and sum each subset of n (in the powerset)
nat_nums = lapply(sets, raise)
# Just changing the data class to vector (from a list), and sriting them
nat_nums = unlist(nat_nums)
sort(nat_nums)
```

```
##  [1] 0.00 0.25 0.50 0.75 1.00 1.25 1.50 1.75 2.00 2.25 2.50 2.75 3.00 3.25 3.50
```

adding one more negative integer, $-3 \leq z \leq 1$ we get…

```r
# first few natural numbers
z = -3:1
# produce the power set of n
sets = powerSetCond(z, rev=TRUE)
# raise by 2 and sum each subset of n (in the powerset)
nat_nums = lapply(sets, raise)
# Just changing the data class to vector (from a list), and sriting them
nat_nums = unlist(nat_nums)
sort(nat_nums)
```

```
##  [1] 0.000 0.125 0.250 0.375 0.500 0.625 0.750 0.875 1.000 1.125 1.250 1.375
## [13] 1.500 1.625 1.750 1.875 2.000 2.125 2.250 2.375 2.500 2.625 2.750 2.875
## [25] 3.000 3.125 3.250 3.375 3.500 3.625 3.750
```

So then, is $\mathbb{R}^+$ the individual sums of the subsets of $P(\mathbb{Z})$ raised by 2? This also makes me think that the $\mathbb{R}^+$ can be achieved by individualy summing the subsets in the power set of rational numbers.

Anyway, this is not what the question asked, but I spent too long thinking about it, so I have copied an answer from the internet for the sake of time.