Continuous Generalization of 2's Complement Arithmetic

Shivam Patel
Dept of Computer Science, Illinois Institute of Technology

November 2022

This report extends our EuroP4 paper [2] by providing foundational definitions and propositions. The report starts from first principles by describing the binary encoding of integers (§1), makes precise definitions and conjectures (§2) it uses for extending 2's complement encoding (§3) to endcode the real numbers (§4), and ends with a description of how to compute parameters (§6) to the approach described in the EuroP4 paper. The conjectures in §2 involve intuitive first-principles results about 2's complement arithmetic that can be proved rigorously in future work.

Future work includes, expanding this to an alternative field construction of $\mathbb{R}$, presenting the precision analysis and algorithm design of statistical primitives such as hypothesis testing and parameter estimation on devices with low computational capacity, with network-enabled devices being of primary interest.

1 Binary encoding of integers

We begin with some basic definitions and notation. In particular, definitions for sets that correspond to signed and unsigned bit strings, and expressions of their value through writing them as a sum of powers of 2 (which we call the *expansion form*). These definitions will support the propositions and proofs that will be presented once we move beyond the integer cases. The notation can get a bit cumbersome at times to due multiple indices, stars, asterisks and the like, but we hope that after the introductory discussion regarding Euclidean Division and the Existence of 2's Complement representations through Euclidean Division, enough familiarity will be built so that the reader will be comfortable as we increase the level of abstraction. The definitions are set up in a way that for each class of numbers we transcend through, the notation and meaning carries over, and that the "next hop" is defined in terms of the previous.

Starting at 2's Complement representation of integers, what we want for this number system and its extensions include the following. First we need a notion of a p-sized bit vector or binary representation of a given number x which we do through $(x)_p$. How does $(x)_p$ relate to x? The content of $(x)_p$ are the bits $(x)_p^n \in \{0,1\}$ for $0 \leq n \leq p-1$, such that its sum (eventually series) expansion $x_p = x$.

Note that $(x)_p^n$ denotes the n-th bit in the p-dimensional bit vector $(x)_p$. We use this notation to emulate vectors and limits. In particular, we would like to take limits of numerical sequences, but also do algebra on coordinate vectors. It is not clear that you could take (meaningful) limits on abstract bit vectors but still be able to do algebra on them when needed without these two views simultaneously. Then the question is, where do the $(x)_p$ reside? They all reside in the space of p-dimensional binary vectors $\{0,1\}^p$ and in particular one of its partitions $\{0,1\}_+^p$ or $\{0,1\}_-^p$.

Definition 1. *Bit Vectors of Dimension p*
Let $p \in \mathbb{N},\ n \in \mathbb{Z}^+$.

Define $(x)_p := (x_{p-1}, x_{p-2}, ...x_1, x_0)$ *where* $\quad 0 \leq i \leq p-1 \implies x_i \in \{0,1\}$

Define $(x)_p^n := x_n$ *if* $0 \leq n \leq p-1 \wedge x_n \in (x)_p$

Define $x_p := -1 \cdot (x)_p^{p-1} \cdot 2^{p-1} + \sum_{i=0}^{p-2} (x)_p^i \cdot 2^i$

Define $\{0,1\}_+^p := \{(x)_p : (x)_p^{p-1} = 0\}$

Define $\{0,1\}^p_- := \{(x)_p : (x)_p^{p-1} = 1\}$

Define $\{0,1\}^p := \{0,1\}^p_+ \cup \{0,1\}^p_-$

To motivate the rest of the paper and the alternative route we take to getting to the fundamental results, we need to discuss some of key results by Euclid.

Definition 2. *Integer Divisibility*

Let $a, b \in \mathbb{Z}$. *Then* $b|a$*(read as* b divides a*) if* $\exists$ a quotient $q \in \mathbb{Z}$ *such that* $a = bq$.

Definition 3. *The Set of Divisors*

Let $a \in \mathbb{Z}$ *and let* $B(a) = \{b \in \mathbb{Z} : b|a\}$. *Each* $b \in B$ *is called a* divisor *of* a, *and* $B(a)$ *is the set of all divisors of* a.

Definition 4. *Common Divisor(s) of* a, n

Let $a, n \in \mathbb{Z}$ *and* $B(a) = \{b \in \mathbb{Z} : b|a\}$ *and* $B(n) = \{b \in \mathbb{Z} : b|n\}$. *Denote* $B(a,n) = B(a) \bigcap B(n)$ *as the set of* Common Divisors *of* a *and* b. *That is,* $b \in B(a,n)$ *if and only if* $b|a$ *and* $b|n$.

Proposition 1. *Existence of Common Divisors of* a, n

Let $a, n \in \mathbb{Z}$. *Then* $B(a,n) \neq \emptyset$.

Proof. Observe $a = a \cdot 1$ and $n = n \cdot 1$. Then by the definition 2 we have that $1|a$ and $1|n$. Therefore $1 \in B(a)$ and $1 \in B(n)$, which implies $1 \in B(a) \bigcap B(n)$ which by definition 4 implies $1 \in B(a,n)$. Therefore $B(a,n) \neq \emptyset$.

□

Definition 5. *Greatest Common Divisor (GCD) of* a, n

Let $a, n \in \mathbb{Z}$ *and* $B(a,n)$ *their set of Common Divisors. Denote* $GCD(a,n) = \max(B(a,n))$ *as the "Greatest Common Divisor" of* a *and* n.

Theorem 1.1. *Euclidean Division of* a, b

Let $a, b \in \mathbb{Z}$. *Then there exists unique quotient* q *and remainder* $r \in \mathbb{Z}$ *with* $0 \leq |r| < b$ *such that* $a = b \cdot q + r$.

Definition 6. *The remainder operation* a *mod* b

Let $a, b \in \mathbb{Z}$. *Define the "modulus"* a *mod* b *as the remainder* r *obtained from performing Euclidean division of* a *by* b.

We begin with a discussion of the Euclidean Division and its relationship and to the 2's Complement representations of integers. The theorems regarding Euclidean Division's existence and uniqueness, and the existence and uniqueness of the GCD(a,n) are well known. From Definition 1.1 (using the variable n in place of b so that we can then relate Euclidean Division to the GCD), Euclidean Division is defined as follows: for any integers a and n, there exists unique integers q and r, where $a = n \cdot q + r$, and r has the property that $0 \leq r < |n|$.

Euclid proved a lemma that GCD(a,n) = GCD(n,a mod n) which provides a recursive algorithm that terminates when one of the arguments is 0, with the other one the being the value of GCD(a,n). If $r_k, r_{k-1} \in \mathbb{Z}$ and $r_k = 0$ then $\text{GCD}(r_{k-1}, 0) = r_{k-1}$ due to a simple proof that any integer divides 0. Therefore it can be shown using iterated Euclidean Division that there will be a remainder of 0, the recursion terminates, and that due to the lemma on the recursive nature of the GCD(a,n), we did indeed find the true GCD(a,n).

We do not prove the general Euclidean GCD Algorithm here, but we use its insights as the basis for our investigation into 2's Complement Arithmetic. A simple justification as to why the procedure eventually terminates is that $\mathbb{Z}_n := \{r \in \mathbb{N} : 0 \leq r < |n|\}$ has length $|n|$. We now apply this notion to show the existence and uniqueness of the 2's Complement representations of $\mathbb{N}$(up to a given bit-width p). We use this later when we define our extension of 2's Complement to $\mathbb{R}$: we will define a "Non Halting" division algorithm for a certain subset of $(0, 1)$ where all members x of that subset do not have a 2's Complement representation $(x \cdot 2^q)_{p+q}$ for any $(p, q) \in \mathbb{N}^2$. In building up to it, we will draw inspiration heavily from the Euclidean procedure below.

Lemma 1.2. *Existence and Uniqueness of 2's Complement Representation for* $\mathbb{N}$

$$\forall x \in \mathbb{N}, \exists p \in \mathbb{N} : \exists! (x)_p \in \{0, 1\}_+^p : \ x_p = x.$$

Proof. The procedure outlined here we will refer to as "The Euclidean Division Algorithm for Base 2". Recall the statement of Euclidean division: that $\forall a, b \in \mathbb{Z}, \exists! q \in \mathbb{Z}, \exists! r \in \mathbb{Z}$ such that $a = bq + r$ and $0 \leq r < |b|$. Take $a = x$ and $b = 2^{\lfloor log_2(x) \rfloor}$ and apply Euclidean division. There are unique $q_0, r_0 \in \mathbb{Z}$ such that $x = q_0 2^{\lfloor log_2(x) \rfloor} + r_0$ with $0 \leq r < 2^{\lfloor log_2(x) \rfloor}$ and refer to this notion as "the first step". To find the 2's complement representation of x, we need to show that either $r_0 = 0$ (that is, the remainder is 0 after the first step, in our case corresponds to $x = 2^s : s \in \mathbb{Z}^+$) or that repeating the procedure results in $r_{k-1} = 0$ after some k number of steps.

We will prove by contradiction that the procedure not terminating causes a contradiction. We will refer from here on out a statement implying a contradiction with $\rightarrow\perp$. Suppose the procedure never terminates (that is, we always have non zero remainder), then for any number of steps n, we always have the remainder $r_{n-1} > 0$. To build towards the contradiction, let's use the insights in our previous discussion, about the length of the set of remainders. By assumption we have that that $r_n > 0$. Then we have that $r_0 > 0$, suppose we take an additional $m = 2^{\lfloor log_2(x) \rfloor}$ steps, keep in mind this was the divisor on the first step. Since non-termination means that we have positive remainder at each step, the first step and taking an additional $m = 2^{\lfloor log_2(x) \rfloor}$ steps implies that due to the existence and uniqueness of Euclidean Division, we can write the following sequence of equalities:

$$x = q_0 2^{\lfloor log_2(x) \rfloor} + r_0,\ 0 < r_0 < 2^{\lfloor log_2(x) \rfloor}$$

$$r_0 = q_1 2^{\lfloor log_2(r_0) \rfloor} + r_1,\ 0 < r_1 < 2^{\lfloor log_2(r_0) \rfloor}$$

$$r_1 = q_1 2^{\lfloor log_2(r_1) \rfloor} + r_2,\ 0 < r_2 < 2^{\lfloor log_2(r_1) \rfloor}$$

...

$$r_{m-2} = q_{m-2} 2^{\lfloor log_2(r_{m-2}) \rfloor} + r_{m-1},\ 0 < r_{m-1} < 2^{\lfloor log_2(r_{m-2}) \rfloor}.$$
$$r_{m-1} = q_{m-1} 2^{\lfloor log_2(r_{m-1}) \rfloor} + r_m,\ 0 < r_m < 2^{\lfloor log_2(r_{m-1}) \rfloor}.$$

Let $R = \{r_0, r_1, ...r_m\}$. Observe $|R| = m + 1 = 2^{\lfloor log_2(x) \rfloor} + 1$ Its clear to see as well that 0 based indexing from $0...m$ implies the count is $m + 1$. and that due the floor function being a lower bound on its argument, and $2^{(\cdot)}$ being an increasing function the following holds: $r_0 < 2^{\lfloor log_2(x) \rfloor} \leq 2^{log_2(x)} = x$. Then by the same argument applied to $r_{k-1}, r_k : 0 < k \leq m$ it follows: $0 < r_m < r_{m-1} < ... < r_0 < 2^{\lfloor log_2(x) \rfloor}$. Therefore since $\forall r_k \in R, r_k \in \mathbb{N}$, and $0 < r_k \leq r_0$ and $r_0 \in \mathbb{Z}_{2^{\lfloor log_2(x) \rfloor}}$ it follows that all of the $r_k \in \mathbb{Z}_{2^{\lfloor log_2(x) \rfloor}}$, which implies $R \subset \mathbb{Z}_{2^{\lfloor log_2(x) \rfloor}}$.

Now, consider the monotone property of set length. That if A, B are proper subsets(denoted $\subset$) of $\mathbb{Z}$ and $A \subset B$, then $|A| < |B|$. $R \subset \mathbb{Z}_{2^{\lfloor log_2(x) \rfloor}}$ implies $|R| < |\mathbb{Z}_{2^{\lfloor log_2(x) \rfloor}}|$ which is the contradiction since $|R| = m + 1$ and $|\mathbb{Z}_{2^{\lfloor log_2(x) \rfloor}}| = m$. Which means $|R| < |\mathbb{Z}_{2^{\lfloor log_2(x) \rfloor}}| \rightarrow m + 1 < m \rightarrow 1 < 0 \rightarrow\perp$. Therefore $\exists k : 0 \leq k < m$ such that $r_k = 0$. Therefore we have shown that "The Euclidean Division Algorithm for Base 2" terminates.

Now to show that this procedure generates the bits of the 2's complement representation $(x)_p : x_p = x$ to meet the definition, we have to show that the sequence of quotients are $\{q_j\}_{j=0}^p$ are each 0 or 1, and

they are unique in terms of their allocation. Again by contradiction, suppose $q_0 \geq 2$, then $x - r_0 = q_0 2^{\lfloor log_2(x) \rfloor} \geq 2 \cdot 2^{\lfloor log_2(x) \rfloor} = 2^{\lfloor log_2(x) \rfloor + 1} > 2^{log_2(x)} = x \geq 2^{\lfloor log_2(x) \rfloor}$. Then $x - r_0 > x$ which implies $r_0 < 0$ contradicting the remainder property of Euclidean Division. Suppose $q_0 \leq 0$ then $q_0 2^{\lfloor log_2(x) \rfloor} \leq 0$ adding r_0 gives $x \leq r_0 < 2^{\lfloor log_2(x) \rfloor} \leq 2^{log_2(x)} = x \rightarrow x < x \rightarrow \perp$.

Therefore by the existence and uniqueness of the quotient in the normal Euclidean Division, the only remaining possibility is $q_0 = 1$. The same arguments hold when replacing x with r_{n-1} and r_0 with r_n for $n < k-1$ where k is the step our procedure terminates that we found from the initial proof by contradiction. This then gives us the rest of the quotients are unique as well. To show that the underlying sum is equal to x, do back substitution on the remainders, since we explicitly showed that $r_k = 0$ the sum is finite and we can write it out as $x = 2^{\lfloor log_2(x) \rfloor} + 2^{\lfloor log_2(r_0) \rfloor} + ... + 2^{\lfloor log_2(r_{k-1}) \rfloor} + 0$.

Now we have to construct the bit vector $(x)_p \in \{0,1\}_+^p$ where when we write out $(x)_p$ as a sum as per definition 1, which we call the *expansion form of* $x \in \mathbb{N}$, we have that $x_p = x$. First we have to determine p. Consider how we built the initial contradiction, it was when we took one more step than the size of $2^{\lfloor \log_2(x) \rfloor}$, and since our definition requires the leading bit(the coordinate of $(x)_p$ at $p-1$ denoted as $(x)_p^{p-1}$) has value 0, let us take $p = \lceil \log_2(|x|+1) \rceil + 1$. Furthermore, we will see in the remainder of the report that this way of defining p guarantees that p is always defined, and at worst case is equal to 1 which is needed for $0 < x < 1$ when we get to working with the Fixed-Point (Dyadic Rational) and Real numbers (Theorem 4.2).

Now we show that this choice of p is a suitable bit-width to represent x. To finish, consider the set $S = \{p-1, p-2, ...0\}$ and $R = \{r_0, r_1, ...r_{k-1}\}$. Since $x \in \mathbb{N}$ by assumption, and we have that $x = 2^{\lfloor \log_2(x) \rfloor} + 2^{\lfloor \log_2(r_0) \rfloor} + ... + 2^{\lfloor \log_2(r_{k-1}) \rfloor} + 0 = 0 \cdot (-1) \cdot 2^{\lceil \log_2(|x|+1) \rceil} + 1 \cdot 2^{\lfloor \log_2(x) \rfloor} + 1 \cdot 2^{\lfloor \log_2(r_0) \rfloor} + ... + 1 \cdot 2^{\lfloor \log_2(r_{k-1}) \rfloor}$, then since $p = \lceil \log_2(|x|+1) \rceil + 1$, we have that $p-1 = \lceil \log_2(|x|+1) \rceil$ and therefore we can take $(x)_p^{\lceil \log_2(|x|+1) \rceil} = (x)_p^{p-1} = 0$. To relate these expressions to what one is used to from computer arithmetic, we adopt a standard definition of Significant Bits (SB). Where the n-th significant bit of $(|x|)_p$ is the n-th highest power with non zero coefficient in the expansion form, which is equivalently the n-th highest index of the bit-vector form. From this, we define the Most Significant Bit (MSB) as the maximum of the significant bits. It's clear that since $x \in \mathbb{N}, x = |x|$ and therefore the MSB of x is the exponent of the $1 \cdot 2^{\lfloor \log_2(x) \rfloor}$ term. We have to show that the MSB $\lfloor \log_2(x) \rfloor \leq p-2$. This can be seen from the following sequence of implications $p = \lceil \log_2(|x|+1) \rceil + 1 \rightarrow p-1 = \lceil \log_2(|x|+1) \rceil = \lceil \log_2(x+1) \rceil \geq \lfloor log_2(x+1) \rfloor > \lfloor log_2(x) \rfloor \rightarrow \lfloor log_2(x) \rfloor < p-1 \rightarrow \lfloor log_2(x) \rfloor \leq p-2$. As we can see, $\lfloor log_2(x) \rfloor$ is not actually in our remainder set R. So we have to "set the bit" at $(x)_p^{MSB} = (x)_p^{\lfloor log_2(x) \rfloor} = 1$. This now is sufficient to establish that all of the elements of R including the MSB can be fit into S, while still maintaining our rules on the leading bit at $p-1$ being 0. This is due to the fact that prior in the proof, we showed the sequence of powers was decreasing, due to the remainders decreasing towards 0 to eventually terminate the expansion procedure. Therefore for $0 \leq i < \lfloor log_2(x) \rfloor$ take the i^{th} coordinate of the bit vector $(x)_p$, as $(x)_p^i = 0$, if $i \in S - R$ and $(x)_p^i = 1$, if $i \in R$. Since we manually evaluated the sum prior, we have that $x_p = x$. □

To describe the theory of 2's complement, we will state the behavior of some primitive operations as conjectures and assume them to be true without proof. The only conjectures we rely on in this report (that we do not directly prove from first principles in this report) are that 2's complement addition, negation, and multiplication exist and have the properties that we are accustomed to. The properties that we are referring to are that 2's complement addition, multiplication, and negation correspond to true addition, multiplication, and negation of the integers they represent(when we allow for bit-widths to be increased) and that these operations have the typical properties of commutativity, associativity, distributivity, and identity that we are used to from normal integer arithmetic.

2 Definitions and Conjectures Regarding 2's Complement Integer Arithmetic

Now we state the fundamental operations on 2's Complement representations of integers. In particular we define them through Boolean operations where we can define the value of the nth coordinate of some operation(the result of addition, negation, multiplication) as Boolean functions of the coordinates of the arguments. There is a notion of *carrying* when one performs addition where the aim is to write the outcome of an operation in a canonical form. The canonical form for base-10 requires that each coordinate in the

result is a member of $\mathbb{Z}_{10} = \{0,1,2,3,4,5,6,7,8,9\}$. So the carrying procedure we use in normal arithmetic in base 10 is that for some $\frac{\alpha}{2} \in \mathbb{Z}$(so α is upper bounded by 18) in the process of addition we rewrite $\alpha \cdot 10^k$ as $\lfloor \frac{\alpha}{10} \rfloor 10^{k+1} + (\alpha \mod 10)10^k$. For an example, suppose we use base 10 arithmetic and we want to perform 456+789. Then we can write the 2 numbers in their base 10 expansion forms, add the coordinates, and apply carrying. It is clear from integer addition that $456+789 = (4\cdot10^2+5\cdot10^1+6\cdot10^0)+(7\cdot10^2+8\cdot10^1+9\cdot10^0) = (4+7)\cdot10^2+(5+8)\cdot10^1+(6+9)\cdot10^0 = (11)\cdot10^2+(13)\cdot10^1+(15)\cdot10^0 = (11)\cdot10^2+(13+1)\cdot10^1+(5)\cdot10^0 = (11+1)\cdot10^2+(4)\cdot10^1+(5)\cdot10^0 = (1)\cdot10^3+(1)\cdot10^2+(4)\cdot10^1+(5)\cdot10^0 = 1145$. Therefore the definitions of the carry flag (Def 7) and of addition (Def 8) are framed to capture the same thing but in binary. Similarly, since multiplication can be defined through repeated addition, that is the motivation for the content of Def 10. The various conjectures on the properties of these operations are developed so that we can use 2's Complement Integer arithmetic to develop the other types of arithmetic in this report.

Definition 7. *Carry Flag*
Let $p \in \mathbb{N}, n \in \mathbb{Z}^+, (x)_p, (y)_p \in \{0,1\}^p$.

Define $\text{carry}(n, (x)_p, (y)_p)$ *where:*

$\text{carry}(-1, (x)_p, (y)_p) = 0$

$\text{carry}(0, (x)_p, (y)_p) = (x)_p^0 \wedge (y)_p^0$

$\text{carry}(n, (x)_p, (y)_p) = ((x)_p^n \wedge (y)_p^n) \vee (((x)_p^n \vee (y)_p^n) \wedge \text{carry}(n-1, (x)_p, (y)_p))$ *if* $1 \leq n \leq p-1$

Definition 8. *2's Complement Integer Addition*
Let $p \in \mathbb{N}, n \in \mathbb{Z}^+, (x)_p, (y)_p \in \{0,1\}^p$.

Define $\bigoplus : \{0,1\}^p \times \{0,1\}^p \to \{0,1\}^{p+1}$ *where:*

$((x)_p \bigoplus (y)_p)_{p+1}^n = carry(n-1, (x)_p, (y)_p)$ *if* $n = p$.

$((x)_p \bigoplus (y)_p)_{p+1}^n = (x)_p^n$ *XOR* $(y)_p^n$ *XOR* $carry(n-1, (x)_p, (y)_p)$ *if* $0 \leq n \leq p-1$.

Conjecture 1. *Properties of 2's Complement Integer Addition*
Let $p \in \mathbb{N}, n \in \mathbb{Z}^+$.

$\forall (x)_p, (y)_p, (z)_p \in \{0,1\}^p :$

(1) $(x)_p \bigoplus (y)_p = (x+y)_{p+1}$

(2) $(x)_p \bigoplus (y)_p = (y)_p \bigoplus (x)_p$

(2) $((x)_p \bigoplus (y)_p) \bigoplus (z)_p = (x)_p \bigoplus ((y)_p \bigoplus (z)_p)$

Definition 9. *2's Complement Negation*
Let $p \in \mathbb{N}, n \in \mathbb{Z}^+, (x)_p \in \{0,1\}^p$

Define $(-x)_p \in \{0,1\}^p$ *where:*

$(-x)_p = (\neg x)_p \bigoplus (1)_p$ *such that* $(\neg x)_p^n = \neg (x)_p^n$

Conjecture 2. *Properties of 2's Complement Negation*
Let $p \in \mathbb{N}, n \in \mathbb{Z}^+, (x)_p \in \{0,1\}^p_+, (y)_p \in \{0,1\}^p_-$

(i) $\forall (x)_p \in \{0,1\}^p_+, \exists! (x^*)_p \in \{0,1\}^p_-$*:*

(1) $(x^*)_p = (-x)_p$.

(2) $(x)_p \bigoplus (-x)_p = (-x)_p \bigoplus (x)_p = (0)_p$

(3) $-x_p = -x$

(ii) $\forall (y)_p \in \{0,1\}^p_-, \exists! (y^*)_p \in \{0,1\}^p_+$*:*

(1) $(y^*)_p = (-y)_p$.

(2) $(y)_p \bigoplus (-y)_p = (-y)_p \bigoplus (y)_p = (0)_p$

(3) $-y_p = -y$

Definition 10. *2's Complement Integer Multiplication*
Let $p \in \mathbb{N}, n \in \mathbb{Z}^+, (x)_p, (y)_p \in \{0,1\}^p$.

Define $\bigotimes : \{0,1\}^p \times \{0,1\}^p \to \{0,1\}^{2p}$ *where:*

$(x)_p \bigotimes (y)_p = \bigoplus_{k=0}^{p-1} (x \cdot (y)_p^k \cdot 2^k)_{p+k}$

Conjecture 3. *Properties of 2's Complement Integer Multiplication*
Let $p \in \mathbb{N}, n \in \mathbb{Z}^+$.

$\forall (x)_p, (y)_p, (z)_p \in \{0,1\}^p$:

(1) $(x)_p \bigotimes (y)_p = (x \cdot y)_{2p}$

(2) $(x)_p \bigotimes (y)_p = (y)_p \bigotimes (x)_p$

(2) $((x)_p \bigotimes (y)_p) \bigotimes (z)_{2p} = (x)_{2p} \bigotimes ((y)_p \bigotimes (z)_p)$

3 First Extension of 2's Complement: Finite Representability

We now begin the extension of integer representations of the various operations defined previously to the finite fractional and infinite fractional cases. In both instances, we will map a given number or bit string to a carefully constructed integer, and in the infinite fractional case, proceed via limits of finite operations. We will examine some properties about the subset of the rational numbers $\mathbb{Q}$ that can be written as a finite sum of powers of 2. These numbers are often called the *dyadic rationals*. We do not use that naming here because the popular algebraic convention to discuss the dyadic rationals are as $\mathbb{Z}[1/2]$, i.e. the set of finite integer linear combinations of integer powers of $\frac{1}{2}$.

The problem with this approach is, that if we use the previously mentioned framing, then defining and examining arithmetic, and eventually the convergence of sequences and series of dyadic rationals in the context of the framework we are proposing would involve an intermediate step of converting the integer coefficients to 2's Complement, scale (bit shift) the coefficients up or down based on the power of $\frac{1}{2}$ that coefficient was associated with, and then add the terms up. The key take away is that our framing and the classical framing are equivalent in the results both routes produce in the end. But the classical framing requires more complicated bit-width and coordinate management in the approach described here, due to

how we defined integer addition and multiplication in Def 8 and Def 10 to be operations between different sized spaces of bit vectors.

Therefore in the following section we propose some definitions that capture the essential properties of a rational being a finite sum of powers of 2 in definition Def 11. We then use that definition to show a necessary and sufficient condition for a rational number to be finitely representable (equivalently, a dyadic rational). We then extend the definition of the spaces of bit vectors mentioned in Def 1 that handle 2's Complement representation of integers to definition Def 13 to handle finitely representable numbers. After this, we establish a few equivalence lemmas, which establish that the existence and uniqueness of the representations of finitely representable numbers (Lemma 3.2 and Lemma 3.3), and as well that the addition and multiplication 2's Complement representations of finitely representable numbers can be defined through 2's Complement addition and multiplication of the scaled (bit shifted) integer they correspond to. Towards the end, of this section, we will have established the framework for both integers and rationals that are finite sums of powers of 2. Then we then define the "Fixed-Point" spaces in Def 14, which we use as a mechanism to take spaces of bit vectors like in Def 1 and Def 13, to then actually associate the value of the underlying number x to a bit vector$(x)_p$ or (eventually)$(x)_{p,q}$, and as well reason about spaces of different bit-widths.

Definition 11. *Finite Representability*
$x \in \mathbb{Q}$ *is* finitely representable *if* $\exists S \subset \mathbb{Z}$ *such that* S *is finite and* $|x| = \sum_{k \in S} 2^k$.

Definition 12. *The (extended) Well Ordering Principle (WOP)*
Let $S \subset \mathbb{Z}$ *such that:* $S \neq \emptyset$ *and* $|S| \in \mathbb{N}$. *Then* $\exists x, y \in S : x = \min(S)$ *and* $y = \max(S)$.

The following lemma gives a simpler condition to reason about finite representability, and contains insights that will form the basis of fractional operations. In particular, how to compute the coordinates of finitely representable rationals and (eventually) the rest of the real line. To support it, and as well as a few other propositions and lemmas that will follow that one, we will invoke a variant of the Well Ordering Principle which we define at 12. Some may refer to it casually as the contra-positive to the Induction axiom, but in its standard statement is that every non empty $S \subset \mathbb{N}$ bounded from below has a minimum element. An exercise many do when learning about the WOP involves showing that the standard statement of the WOP implies the variant we define in 12. That is if you extend it to non-empty $S \subset \mathbb{Z}$ bounded from above and below, then S has a minimum and maximum element and S contains them, which is the variant we use here.

Lemma 3.1. *Integer Conversion Criteria*
Let $x \in \mathbb{Q}$. x *is* finitely representable *iff* $\exists q \in \mathbb{Z}$ *such that* $|x \cdot 2^q| \in \mathbb{N}$.

Proof. ($\rightarrow$) Suppose $x \in \mathbb{Q}$ is finitely representable, then $\exists$ finite $S \subset \mathbb{Z} : |x| = \sum_{k \in S} 2^k$. Since S is finite let $n = |S|$, then apply the well ordering principle and let $s_0 = \min(S)$, and $s_{n-1} = \max(S)$. Due to S being finite, enumerate the elements in sorted order with the indices labeled. Then we see that $min(S) = s_0 < s_1 < ...s_{n-2} < s_{-1} = max(S)$ Since exponentiation is monotone on $\mathbb{R}$ it is on $\mathbb{Z}$ as well. Therefore $2^{min(S)} = 2^{s_0} < 2^{s_1} < ...2^{s_{n-1}} < 2^{s_{n-1}} = 2^{max(S)}$

Multiplying by 2^{-s_0} gives $2^{min(S)-s_0} = 2^{s_0-s_0} = 1 < 2^{s_1-s_0} < ...2^{s_{n-1}-s_0} < 2^{s_{n-1}-s_0} = 2^{max(S)-s_0}$ From this, we can see that for each $i : 2^{s_i-s_0} \geq 1$ and therefore is in $\mathbb{N}$. Summing up the pieces shows $\sum_{i=0}^{n} 2^{s_i-s_0} = |x| \cdot 2^{-s_0} = |x \cdot 2^{-s_0}|$.

($\leftarrow$) Suppose $|x \cdot 2^q| \in \mathbb{N}$. Apply 1.2, then we have for some $p \in \mathbb{N}$ a corresponding $(|x2^q|)_p \in \{0,1\}^p_+$ with $|x \cdot 2^q|_p = (-1) \cdot (|x2^q|)_p^{p-1} \cdot 2^{p-1} + \sum_{i=0}^{p-2} (|x2^q|)_p^i \cdot 2^i = |x \cdot 2^q|$. Now multiply by 2^{-q} and see that $|x| = |x \cdot 2^q \cdot 2^{-q}| = |x \cdot 2^q| \cdot 2^{-q} = (-1) \cdot (|x2^q|)_p^{p-1} \cdot 2^{p-1-q} + \sum_{i=0}^{p-2} (|x2^q|)_p^i \cdot 2^{i-q} = |x \cdot 2^q|_p \cdot 2^{-q}$.

Therefore $|x|$ is a finite sum of powers of 2. To construct S, extract the powers with non zero coefficients—that is, let $S = \{i - q : (0 \leq i \leq p-1) \wedge (|x2^q|)_p^i = 1\}$. This proves $|x|$ meets the definition of finite representability. □

The key insight was to exploit the fact that we can take a rational number that is a finite sum of powers of 2 (which may or may not be an integer to begin with) and use multiplication by a power of 2 (ideally we want it to be the negative of $\min(S)$) to find an integer representation for it such that the distribution of 1's and 0's in its bit vector representation are preserved. As we will see, we can use this as the basis to expand

our arithmetic that we had defined for 2's Complement representations of integers, but first we distill the Integer Conversion lemma further. We begin with some definitions and then provide some propositions.

Definition 13. *Extension to FP*
Let $p,\ q \in \mathbb{N},\ n \in \mathbb{Z}$.

Define $(x)_{p,q} := (x_{p-1}, x_{p-2}, ...x_1, x_0, x_{-1}, x_{-2}, ...x_{-q})$ *where* $-q \leq i \leq p-1 \rightarrow x_i \in \{0,1\}$

Define $(x)^n_{p,q} := x_n$ *if* $-q \leq n \leq p-1 \wedge x_n \in (x)_p$

Define $x_{p,q} := -1 \cdot (x)^{p-1}_{p,q} \cdot 2^{p-1} + \sum_{j=0}^{p-2} (x)^j_{p,q} \cdot 2^j + \sum_{i=1}^{q} (x)^{-i}_{p,q} \cdot 2^{-i}$

Define $\{0,1\}^{p,q}_{+} := \{(x)_{p,q} : (x)^{p-1}_{p,q} = 0\}$

Define $\{0,1\}^{p,q}_{-} := \{(x)_{p,q} : (x)^{p-1}_{p,q} = 1\}$

Define $\{0,1\}^{p,q} := \{0,1\}^{p,q}_{+} \bigcup \{0,1\}^{p,q}_{-}$

To shorten the proof of the following proposition, we invoke the notion of a "vacuous sum": we define $\sum_{i=n_1}^{n_2} a_i = 0$ whenever $n_2 < n_1$.

Lemma 3.2. *Bitwise Equivalence (Positive case)*
Let $x \in \mathbb{Q} - \mathbb{Z}$ *and* x *be finitely representable and positive. Then* $\exists p, q \in \mathbb{N}$ *such that* $\exists!\ (x)_{p,q} \in \{0,1\}^{p,q}_{+}$ *such that* $x_{p,q} = x$ *and for* $i : -q \leq i \leq p-1$ *we have that* $(x)^n_{p,q} = (x \cdot 2^q)^{n+q}_{p+q}$

Proof. Assume $x \in \mathbb{Q} - \mathbb{Z}$ and x is finitely representable. Let $S \subset \mathbb{Z}$ be its set of powers. By the Well Ordering Principle let $s_0 = \min(S)$ be the minimum power, $m = |S|$, be the number of elements and $s_{m-1} = \max(S)$, and $\forall i : 0 \leq i \leq n-1, s_i$ be the ith element of S in sorted order. $s_0 < 0$ since $s_0 \geq 0$ would imply $\forall i : s_i \geq 0$ which in turn would imply that $x \in \mathbb{Z}$, contradicting our assumption. Take $q = -s_0$ and observe that $s_0 < 0 \wedge s_0 \in \mathbb{Z} \rightarrow -s_0 = q \in \mathbb{N}$.

Consider the partition $S^- = \{s \in S : s < 0\}, S^+ = S - S^-$. We are guaranteed $S^- \neq \emptyset$ since $s_0 \in S \bigcap S^-$. To find p observe that if $S^+ = \emptyset$ then $S^+ \bigcup \{-1\} = \{-1\}$ i.e. non empty. Therefore to handle all relevant cases(integer part of x is 0 vs non zero), apply the Well Ordering Principle to $\max(S^+ \bigcup \{-1\})$ and take $p = \max(S^+ \bigcup \{-1\}) + 2$. Define a function $|x|(n) : \{-q, -q+1, ..., p-3, p-2, p-1\} \rightarrow \{0,1\}$, by $|x|(i) = 1$ if $i \in S, 0$ otherwise. Recall that due to the finite representability assumption we have that $|x| = \sum_{i=0}^{n-1} 2^{s_k}, 0 \leq k \leq m-1$.

We can now expand $|x|$ with respect to $|x|(\cdot)$ and see that $|x| = \sum_{i=-q}^{p-1} |x|(i) \cdot 2^i$ since by construction $|x|(i) = 1$ only when $i \in S$. Now split the finite sum into $|x| = (-1) \cdot |x|(p-1) \cdot 2^{p-1} + \sum_{i=0}^{p-2} |x|(i) \cdot 2^i + \sum_{i=-q}^{-1} |x|(i) \cdot 2^i$. Observe that in the case of $0 < |x| < 1, S^+ = \emptyset$ we have $p = 1$ and $p - 2 = -1$ hence the middle sum would equal 0 as a vacuous sum (i.e. no double counting). As well, that since $p = \max(S^+ \bigcup \{-1\}) + 2 = \max(S \bigcup \{-1\}) + 2$ we have that $p - 2 = \max(S \bigcup \{-1\})$ and consequently that $p - 1 \notin S$. Then by the definition of $|x|(\cdot)$ we have $|x|(p-1) = 0$. Now re-index the sums by $i \rightarrow j$ in the middle sum and $i \rightarrow -i$ (implied swapping sum bounds) in the third sum i.e. $|x| = (-1) \cdot |x|(p-1) \cdot 2^{p-1} + \sum_{j=0}^{p-2} |x|(j) \cdot 2^j + \sum_{i=1}^{q} |x|(-i) \cdot 2^{-i}$.

Now to pass to the integer representation, multiply by 2^q (recall $q \in \mathbb{N}$) then we have that $|x|2^q = |x \cdot 2^q| = ((-1) \cdot |x|(p-1) \cdot 2^{p-1} + \sum_{j=0}^{p-2} |x|(j) \cdot 2^j + \sum_{i=1}^{q} |x|(-i) \cdot 2^{-i}) \cdot 2^q = (-1) \cdot |x|(p-1) \cdot 2^{p+q-1} + \sum_{j=0}^{p-2} |x|(j) \cdot 2^{q+j} + \sum_{i=1}^{q} |x|(-i) \cdot 2^{q-i}$. From this, it follows that $|x \cdot 2^q|$ is a natural number that can be represented in $p+q$ bits. Therefore by 1.2 for $p+q$ fixed, $\exists!\ (|x \cdot 2^q|)_{p+q} \in \{0,1\}^{p+q}_{+}$ with $|x \cdot 2^q| = |x \cdot 2^q|_{p+q} = (-1)(|x \cdot 2^q|)^{p+q-1}_{p+q} \cdot 2^{p+q-1} + \sum_{z=0}^{z=p+q-2} (|x \cdot 2^q|)^z_{p+q} \cdot 2^z$

Again from this it now follows that $\forall n : -q \leq n \leq p-1 : |x|(n) = (|x \cdot 2^q|)^{n+q}_{p+q}$. And that it is the only solution due to the uniqueness of the ordinates of the natural number $|x2^q|$. Furthermore, since $x > 0$ we have that $x = |x|$. Therefore $(x)^n_{p,q} = (|x|)^n_{p,q} = |x|(n) = (|x \cdot 2^q|)^{n+q}_{p+q}$. □

Lemma 3.3. *Bitwise Equivalence (negative case)*
Let $x \in \mathbb{Q} - \mathbb{Z}$ and x be finitely representable and negative. Then $\exists p, q \in \mathbb{N}$ such that $\exists!$ $(x)_{p,q} \in \{0,1\}_{-}^{p,q}$ such that $x_{p,q} = x$ and for $n : -q \leq n \leq p-1$ we have that $(x)_{p,q}^{n} = (x \cdot 2^q)_{p+q}^{n+q}$

Proof. Assume x is negative, and $-x$ is positive. Apply the preceding lemma to $-x$. Now we have $\exists p, q\mathbb{N}$ such that $\exists!$ $(-x)_{p,q} \in \{0,1\}_{+}^{p,q}$ such that $-x_{p,q} = -x$ and for $n : -q \leq n \leq p-1$ we have that $(-x)_{p,q}^{n} = (-x \cdot 2^q)_{p+q}^{n+q}$. Consider its natural number representation $(-x \cdot 2^q)_{p+q}$. Based on the lemma for natural number negation, observe that since $(-x \cdot 2^q)_{p+q} \in \{0,1\}_{+}^{p+q}$, its complement $(x \cdot 2^q)_{p+q} \in \{0,1\}_{-}^{p+q}$ with $(x \cdot 2^q)_{p+q}^{n} = \neg(-x \cdot 2^q)_{p+q}^{n}$ XOR $(1)_{p+q}^{n}$ XOR $carry(n-1, \neg(-x \cdot 2^q)_{p+q}, (1)_{p+q})$ and $(-x \cdot 2^q)_{p+q} \bigoplus (x \cdot 2^q)_{p+q} = (0)_{p,q}$.

Write $x \cdot 2^q$ in its expansion form $x \cdot 2^q = x \cdot 2_{p+q}^{q} = (-1)(x \cdot 2^q)_{p+q}^{p+q-1} \cdot 2^{p+q-1} + \sum_{z=0}^{z=p+q-2} (x \cdot 2^q)_{p+q}^{z} \cdot 2^z$. Now multiply by 2^{-q} and see that $x = x \cdot 2_{p+q}^{q} \cdot 2^{-q} = (-1)(x \cdot 2^q)_{p+q}^{p+q-1} \cdot 2^{p-1} + \sum_{z=0}^{z=p+q-2} (x \cdot 2^q)_{p+q}^{z} \cdot 2^{z-q}$. Split the sum into 3 pieces, and see that $x = x \cdot 2_{p+q}^{q} \cdot 2^{-q} = (-1)(x \cdot 2^q)_{p+q}^{p+q-1} \cdot 2^{p-1} + \sum_{z=q}^{z=p+q-2} (x \cdot 2^q)_{p+q}^{z} \cdot 2^{z-q} + \sum_{z=0}^{z=q-1} (x \cdot 2^q)_{p+q}^{z} \cdot 2^{z-q}$. Re-index the sum by in the middle via $z \to j+q$, then $z = q = j+q$ which gives $j = 0$ for the lower bound. For the upper bound we have $j+q = p+q-2$ which gives $j = p-2$, for the the exponent we get $j+q-q = j$, and for the z^{th} bit we get $j+q$. For the third sum, re-index via $z \to -i+q$. Then we have for the lower bound $z = -i+q = 0$ implies $i = q$, and for the upper bound we have $-i+q = q-1$ implies $i = 1$, for the exponent we have $z-q = -i+q-q = -i$, and for the z^{th} bit we have $-i+q$. After (un)swapping the order of summation for the third sum we can see that $x = x \cdot 2_{p+q}^{q} \cdot 2^{-q} = (-1)(x \cdot 2^q)_{p+q}^{p+q-1} \cdot 2^{p-1} + \sum_{j=0}^{j=p-2} (x \cdot 2^q)_{p+q}^{j+q} \cdot 2^j + \sum_{i=1}^{i=q} (x \cdot 2^q)_{p+q}^{-i+q} \cdot 2^{-i}$.

From this it follows that taking $(x)_{p,q} \in \{0,1\}_{-}^{p,q}$ via $(x)_{p,q}^{n} = (x \cdot 2^q)_{p+q}^{n+q}$ for $n : -q \leq n \leq p-1$ is a solution, i.e. we have shown existence. To show that $(x)_{p,q} \in \{0,1\}_{-}^{p,q}$ is a true additive inverse to $(-x)_{p,q} \in \{0,1\}_{+}^{p,q}$, we can see from the extended definition of $\bigoplus$ that $((x)_{p,q} \bigoplus (-x)_{p,q})_{p,q}^{n} = ((x \cdot 2^q)_{p+q} \bigoplus (-x \cdot 2^q)_{p+q})_{p+q}^{n+q}$. So since under integer arithmetic we have that $((x \cdot 2^q)_{p+q} \bigoplus (-x \cdot 2^q)_{p+q})_{p+q} = (0)_{p+q}$, which implies that $\forall n : -q \leq n \leq p-1 \to ((x \cdot 2^q)_{p+q} \bigoplus (-x \cdot 2^q)_{p+q})_{p+q}^{n+q} = 0 = ((x)_{p,q} \bigoplus (-x)_{p,q})_{p,q}^{n}$ which implies $((x)_{p,q} \bigoplus (-x)_{p,q})_{p,q} = 0$.

Finally to show uniqueness. Suppose we have in addition to $(x)_{p,q}$ some other $(x^*)_{p,q}$ with the property that $x = x_{p,q}$ and $x = x_{p,q}^{*}$ and $((x)_{p,q} \bigoplus (-x)_{p,q})_{p,q} = 0$ and $((x^*)_{p,q} \bigoplus (-x)_{p,q})_{p,q} = 0$. Then by passing to the integer representation it follows that $((x \cdot 2^q)_{p+q} \bigoplus (-x \cdot 2^q)_{p+q})_{p+q} = (0)_{p+q}$ and $((x^* \cdot 2^q)_{p+q} \bigoplus (-x \cdot 2^q)_{p+q})_{p+q} = (0)_{p+q}$. But then by the uniqueness of integer negation proposition we have that $(x \cdot 2^q)_{p+q}^{n+q} = (x^* \cdot 2^q)_{p+q}^{n+q}$ therefore $\exists n : x_{p,q}^{n} \neq x^{*n}_{p,q} \to \perp$ □

The task at hand is to now show that for arbitrary finitely representable $x, y \in \mathbb{Q} - \mathbb{Z}$ we can find a common *bit width* so that whatever dimension their sum or product are in, we can select the dimensionality of x, y so that there is no overflow possibility. Since we've handled the existence and uniqueness of representations for a given p, q for positive and negative finitely representable numbers, we will handle both cases simultaneously for each operation. We seek to show that the definition of finitely representable addition (addition by changing representation to $\mathbb{Z}$ via multiplying 2^q, applying integer addition, and shifting back to $\mathbb{Q}$, and applying the bitwise equivalence lemmas) and the definition of finitely representable multiplication (via the same strategy) does indeed correspond to true addition and true multiplication. Once we go to the infinite case, this technique will be the basis for how we reason about arbitrary precision 2's Complement arithmetic.

A small notational remark, so as to not confuse the reader, read $x \,\mathrm{OP}\, y_{(p,q)}$ as if $z = x \,\mathrm{OP}\, y$, $z_{p,q} = \sum ....$ The notation for using subscripts to refer to the numerical expansion form that we have been using up until now, i.e. $z_{p,q}$, may cause confusion since the typesetting may make the reader think we are doing expansion on y only and not x if we do not put (p, q) in parenthesis.

Lemma 3.4. *Finitely Representable Addition*
Let $x, y \in \mathbb{Q} - \mathbb{Z}$ be finitely representable. Then $\exists p, q \in \mathbb{N} : (x+y)_{p,q} \in \{0,1\}^{p,q} \wedge (x)_{p,q} \bigoplus (y)_{p,q} = (x+y)_{p,q}$

Proof. Since $x, y \in \mathbb{Q} - \mathbb{Z}$ are finitely representable,
$\exists p_x, q_x, p_y, q_y \in \mathbb{N} : ((x)_{p_x,q_x} \in \{0,1\}^{p_x,q_x} : x = x_{p_x,qx}) \wedge ((y)_{p_y,qy} \in \{0,1\}^{p_y,q_y} : y = y_{p_y,qy})$. Take $p^* = \max(p_x, p_y), p = p^* + 2, q = \max(q_x, q_y)$. Padding the fractional parts of the representations with

0 bits does not change their values, and padding 1's or 0's on the integer parts of negative and positive numbers respectively does not change their values. Therefore $\exists\ (x)_{p,q}, (y)_{p,q} \in \{0,1\}^{p,q} : x = x_{p,q}, y = y_{p,q}$. Due to how p was constructed, we have that $-2^{p-1} \leq x \leq 2^{p-2} - 1$ and that $-2^{p-1} \leq y \leq 2^{p-2} - 1$. Then $-2^p = -2^{p-1} + -2^{p-1} \leq x + y \leq 2^{p-2} - 1 + 2^{p-2} - 1 = 2^{p-1} - 2 \leq 2^{p-1} - 1$. Since x, y are finitely representable, their sum $x + y$ is as well, and since $-2^p \leq x + y \leq 2^{p-} - 1$, we have that $\exists!(x+y)_{p,q} \in \{0,1\}^{p,q} : x + y = x + y_{(p,q)}$.
Then we have that $(x+y)^n_{p,q} = ((x+y)\cdot 2^q)^{n+q}_{p+q} = ((x\cdot 2^q)_{p+q} \bigoplus (y\cdot 2^q)_{p+q})^{n+q}_{p+q} = ((x)_{p,q} \bigoplus (y)_{p,q}))^n_{p,q}$.
Therefore $(x+y)_{p,q} = (x)_{p,q} \bigoplus (y)_{p,q}$. □

Lemma 3.5. *Finitely Representable Multiplication*
Let $x, y \in \mathbb{Q} - \mathbb{Z}$ be finitely representable. Then $\exists p, q \in \mathbb{N} : (x\cdot y)_{p,q} \in \{0,1\}^{p,q} \wedge\ (x)_{p,q} \bigotimes (y)_{p,q} = (x\cdot y)_{p,q}$

Proof. For multiplication we need to get around the possibility that in trying to verify a particular choice of p, q are sufficient to avoid overflow, we do not flip the directions of the underlying inequalities that tell us if we are representable in that configuration. Use the fact that since $x, y \in \mathbb{Q} - \mathbb{Z}$ are finitely representable, so are their absolute values $|x|$ and $|y|$, then $\exists p_x, q_x, p_y, q_y \in \mathbb{N} : ((|x|)_{p_x,q_x} \in \{0,1\}^{p_x,q_x}_+ : |x| = |x|_{p_x,qx}) \wedge ((|y|)_{p_y,qy} \in \{0,1\}^{p_y,q_y}_+ : |y| = |y|_{p_y,qy})$. Take $p^* = \max(p_x, p_y), p = 2\cdot p^* + 1, q = 2\cdot\max(q_x, q_y)$. Therefore, as with the addition lemma previous to this one, if we pad the fronts of x and y's representation so that the leading sign bits stay the same and pad 0 to the tails of the fractional bits, then $\exists\ (|x|)_{p,q}$, $(|y|)_{p,q} \in \{0,1\}^{p,q}_+ : |x| = |x|_{p,q}, |y| = |y|_{p,q}$. Due to how p^*, p were constructed, we have that $0 \leq |x| \leq 2^{p^*} - 1$ and $0 \leq |y| \leq 2^{p^*} - 1$ which implies $0 \leq |x||y| = |xy| \leq (2^{p^*} - 1)^2 \leq 2^{2\cdot p^*} - \cdot 2^{p^*+1} + 1 \leq 2^{2\cdot p^*+1} - 1 = 2^p - 1$. Which shows the choice of p was sufficient for us not to overflow when using p. Therefore since $|x|, |y|$ are finitely representable, their product $|xy|$ is as well, and since $-2^p \leq 0 \leq |xy| \leq 2^p - 1$, we have that $\exists!(|xy|)_{p,q} \in \{0,1\}^{p,q}_+ : |xy| = |xy|_{(p,q)}$, while maintaining enough padding bits to prevent overflow.

Then by the bitwise equivalence lemmas we have that $(|xy|)^n_{p,q} = (|xy\cdot 2^q|)^{n+q}_{p+q} = (|xy\cdot 2^{q^*}\cdot 2^{q^*}|)^{n+q^*+q^*}_{p+q^*+q^*} = (|x\cdot 2^{q^*} y\cdot 2^{q^*}|)^{n+q^*+q^*}_{p^*+p^*+1+q^*+q^*} = (|(x\cdot 2^{q^*} y\cdot 2^{q^*}|)^{n+q^*+q^*}_{p^*+p^*+1+q^*+q^*} = (|x\cdot 2^{q^*}||y\cdot 2^{q^*}|)^{n+2q^*}_{2p^*+2q^*+1} = (|x\cdot 2^{q^*}|) \bigotimes (|y\cdot 2^{q^*}|)^{n+2q^*}_{2p^*+2q^*+1} = ((|x|)_{p,q} \bigotimes (|y|)_{p,q}))^n_{p,q}$.
Therefore $(|xy|)_{p,q} = (|x|)_{p,q} \bigotimes (|y|)_{p,q}$. The uniqueness of negation implies that $(xy)_{p,q} = (x)_{p,q} \bigotimes (y)_{p,q}$. □

So we have just shown the fundamental definitions and propositions regarding how to extend the 2's complement representation of integers to handle a small subset of $\mathbb{Q}$ such that the standard notions of equality, addition, multiplication, 2's complement negation, and uniqueness hold. We leave multiplicative inversion (multiplication by reciprocal) for later, since the section that we develop now will be needed to "divide" properly. 2 fundamental spaces that we will define next(among others) are the sets FP of Fixed-Point(fixed-point) numbers(equivalently the set of dyadic rationals) and FP^* which are the bit vectors associated with elements of of FP. It is important to note that algebraically they are rings (addition is invert-able, but multiplication is not) and not fields (both addition and multiplication are invert-able). The reason being that a number being a finite sum of powers of 2 does not imply that its reciprocal can be written as a finite sum of powers of 2.

Definition 14. *$FP(p,q)$, $FP^*(p,q)$, FP, FP^**

Define $FP(p,q) := \{x \in \mathbb{Q} : \exists (x)_{p,q} \in FP^*(p,q) : x_{p,q} = x\}$

Define $FP^*(p,q) := \{(x)_{p,q} \in \{0,1\}^{p,q} : \exists x \in FP(p,q) : x = x_{p,q}\}$

Define $FP := \bigcup_{(p,q)\in\mathbb{N}^2} FP(p,q)$

Define $FP^* := \bigcup_{(p,q)\in\mathbb{N}^2} FP^*(p,q)$

FP^* is an Abelian group with respect to the extended $\bigoplus$ since as Lemma 3.4 shows, in FP^* we can always adjust the p, q we use to make sure we do not overflow. Similarly Lemma 3.5 shows that FP^* is a commutative ring with respect to the extended $\bigotimes$. This is not true for $FP^*(p,q)$ since there is an implicit requirement that if we are to do addition successfully, the result must land back in the same space. But

what $FP(p,q)$ and $FP^*(p,q)$ both have, even if they are not Abelian groups in the way we would like them to be, is that they are isomorphic to one another. And in turn, despite FP^* being an Abelian group, it is not (group) isomorphic to FP (we can define a numerical bijection easily, but it is not of the same power in terms of its algebraic consequences) due to the ability to do signed/unsigned extensions, i.e. that padding the fractional bits with 0, pad the leading bits with 1's and 0's for negatives and positives respectively keeps the same value. Which amounts to the existence of mappings along the lines of if $p_1 < p_2, q_1 < q_2$ $FP^*(p_1,q_1) \ni (x)_{p_1,q_1} \longrightarrow (x)_{p_2,q_2} \in FP^*(p_2,q_2)$ and that $x_{p_1,q_1} = x = x_{p_2,q_2}$.

An important theoretical question, which we do not address fully in this report, is what type of algebraic object and what type of operations solve the problem of simultaneously giving uniqueness of representations (circumventing the 0,1 pad issues), no overflow when performing 2's Complement arithmetic (we want to do the arithmetic, and not worry about the technical issues resulting bit vectors being in a different $FP^*(p,q)$ than the one it started in), and forming an algebraic object that is isomorphic to standard number systems ($\mathbb{Z}, \mathbb{Q}, \mathbb{R}$ etc) in the sense that we have bijection, and the bijection preserves the operations of addition and multiplication. The typical solution is to cut everything up into equivalence classes. That can be explored further in future work. For the time being, we proceed to the case of $\mathbb{R} - FP$.

4 Extension to Real numbers: Infinite Representability

Definition 15. *Infinite Representability*
Let $x \in \mathbb{R}$. x is infinitely representable *if $\exists$ an infinite sequence $\{s_n\}_{n=1}^{\infty} \subset \mathbb{Z}$ such that the corresponding following infinite series converges and equals $|x|$. i.e $|x| = \lim_{n\to\infty} \sum_{i=1}^{n} 2^{s_i}$.*

The above definition does not imply the existence of infinitely representable numbers. But as it unsurprisingly turns out, for most $A \subset \mathbb{Q}$ we have that $A \not\subset FP$, and as well $\forall A \subset \mathbb{R} - \mathbb{Q}$ we have that $A \not\subset FP$. It is simple to show the existence of numbers that are not finitely representable for the case of irrationals through proof by contradiction: pick an irrational number, assume it to be finitely representable, but then it is a sum of rationals and hence a rational due to $\mathbb{Q}$ being closed under finite addition, contradicting the supposed irrationality. This handles existence of non-finitely representable numbers but it is interesting to consider (which we will not do here) the case of rationals.

This does not mean that we can have a decision procedure over $\mathbb{R}$ that can determine $\forall x \in \mathbb{R}$ if x is finitely representable or not. This is interesting since its related to an "extended" division algorithm that does not terminate. Suppose E is the set of all extensions E of the Euclidean Divison Algorithm for base 2. By extension we mean that if $x \in \mathbb{Z}$ then $E(x) = (x)_p$ and E halts, if $x \in FP - \mathbb{Z}$ then $E(x) = (x)_{p,q}$ and E halts, and if $x \in \mathbb{R} - FP$ then $E(x)$ sequentially produces the terms of an infinite sequence whose corresponding infinite series converges to x. Keep in mind that the implication is that it never halts due to the meaning of $\mathbb{R} - FP$. Then it has to be the case that for all such E, $\nexists D$ such that D is a decision procedure that can determine $\forall x \in \mathbb{R}$ which of $\mathbb{Z}$, $FP - \mathbb{Z}$, $\mathbb{R} - FP$ that x is in. Since otherwise there exists and extension E of the Euclidean Divison Algorithm for base 2, such that the halting vs non halting of E is decidable by D, which contradicts the undecidability of the Halting Problem.

The focus now is to construct an explicit infinite series that absolutely converges $x \in \mathbb{R} - FP$. We call the underlying sequence the "Non Halting Division Algorithm". We define it as a partial function on $\mathbb{R}$, where it is only defined on $(\mathbb{R} - FP) \bigcap (0,1)$. Then it is defined only for the inputs where we know due to the nature of x (x being not finitely representable), the Non Halting Division Algorithm never terminates.

We will prove that claim, and as well that it can be used to extend the Euclidean Division Algorithm for base 2. Now we proceed to define it, and as well a few standard definitions that we use in the proof of the sequence generation procedure. In particular defining the Floor/Ceiling functions as the Supremum/Infimum

Definition 16. *The Non Halting Division Algorithm*
Let $n \in \mathbb{N}$ and $x \in \mathbb{R} - FP$ and $0 < x < 1$.
Then define the n^{th} step of The Non Halting Division Algorithm as the following recursive sequence:
$s_1 = \lfloor \log_2(x) \rfloor$. $s_n = \lfloor \log_2(x - \sum_{i=1}^{n-1} 2^{s_i}) \rfloor$

Definition 17. *Supremum*
Let $A \subset \mathbb{R} : A \neq \emptyset$ and A is bounded from above.
A real number $\sup(A) \in \mathbb{R}$ is the Supremum of A if the following 2 conditions hold:

(i). $\forall x \in A : x \leq \sup(A)$.
(ii). $\forall \epsilon > 0, \exists x_\epsilon \in A : \sup(A) - \epsilon < x_\epsilon$.

Definition 18. *Infimum*
Let $B \subset \mathbb{R} : B \neq \emptyset$ *and* B *is bounded from below.*
A real number $\inf(B) \in \mathbb{R}$ *is the Infimum of* B *if the following 2 conditions hold:*
(i). $\forall x \in B : \inf(B) \leq x$.
(ii). $\forall \epsilon > 0, \exists x_\epsilon \in B : x_\epsilon < \inf(B) + \epsilon$.

Definition 19. *Floor and Ceiling as Supremum and Infimum*
Define $\lfloor x \rfloor := \sup\{k \in \mathbb{Z} : k \leq x\}$
Define $\lceil x \rceil := \inf\{k \in \mathbb{Z} : k \geq x\}$

Proposition 2. *Non Halting Division is Non Halting*
Let $n \in \mathbb{N}$ *and* $x \in \mathbb{R} - FP$ *and* $0 < x < 1$. *Then the sequence given by* $s_1 = \lfloor \log_2(x) \rfloor$, *and* $s_n = \lfloor \log_2(x - \sum_{i=1}^{n-1} 2^{s_i}) \rfloor$ *has the property* $\forall n \in \mathbb{N} : 0 < \sum_{i=1}^{n} 2^{s_i} < x$.

Proof. Induction on n. Suppose $n = 1$, $0 < x < 1 \to \log_2(x) < 0$. Then $s_1 = \lfloor \log_2(x) \rfloor \leq \log_2(x) < 0$. Using the monotonicity of $2^{(\cdot)}$ on $\mathbb{R}$ gives that $2^{s_1} = 2^{\lfloor \log_2(x) \rfloor} \leq 2^{\log_2(x)} = x < 2^0 = 1$. Therefore $2^{s_1} \leq x < 1$. Since $x \notin FP$, $x = 2^{s_1} \to x \in FP(1,1) \to \perp$, so we have that $0 < 2^{s_1} < x < 1$.

Suppose $n - 1 \geq 1$ and we have that $\sum_{i=1}^{n-1} 2^{s_i} < x$. To show true for n see that the induction hypothesis implies we can subtract the finite sum and exploit that the sum is positive to get $0 < x - \sum_{i=1}^{n-1} 2^{s_i} < x < 1$. Taking logarithms give $\log_2(x - \sum_{i=1}^{n-1} 2^{s_i}) < \log_2(x) < \log_2(1) = 0$. Then $s_n = \lfloor \log_2(x - \sum_{i=1}^{n-1} 2^{s_i}) \rfloor \leq \log_2(x - \sum_{i=1}^{n-1} 2^{s_i}) < \log_2(x) < 0$. Now applying $2^{(\cdot)}$ gives $2^{s_n} \leq x - \sum_{i=1}^{n-1} 2^{s_i} < x < 1$. After adding the sum to both sides we see that $\sum_{i=1}^{n} 2^{s_i} \leq x < 1$. Finally (Again) $x \notin FP, x = \sum_{i=1}^{n} 2^{s_i} \to x \in FP(1,n) \to \perp$. Therefore we have that $0 < \sum_{i=1}^{n} 2^{s_i} < x < 1$. □

Theorem 4.1. *The Non Halting Division Algorithm Converges*
Take s_n, x *as in 2. Then* $\lim_{n\to\infty} \sum_{i=1}^{n} 2^{s_i}$ *exists and equals* x.

Proof. First we show $\forall n > 1 \in \mathbb{N} : 2^{s_n} < x - \sum_{i=1}^{n-1} 2^{s_i}$. By 2 we have that $\forall n : \sum_{i=1}^{n} 2_{s_i} < x$. Then after subtracting the first $n-1$ terms, we can see that $2_{s_n} < x - \sum_{i=1}^{n-1} 2^{s_i}$. From now on refer to the difference of x and the sum up to some k as the k^{th} remainder. We know that the $\forall i \in \mathbb{N} : s_i < 0$ otherwise $x > 1$. Therefore $s_n < 0$ and since $0 < 1$ we have that $s_n < s_n + 1$. By expanding their definitions we see that $\lfloor \log_2(x - \sum_{i=1}^{n-1} 2^{s_i}) \rfloor < \lfloor \log_2(x - \sum_{i=1}^{n-1} 2^{s_i}) \rfloor + 1$.

The reason for this level of detail is to properly exploit the sup property. We are guaranteed that $\lfloor \log_2(x - \sum_{i=1}^{n-1} 2^{s_i}) \rfloor \leq \log_2(x - \sum_{i=1}^{n-1} 2^{s_i}) < \lfloor \log_2(x - \sum_{i=1}^{n-1} 2^{s_i}) \rfloor + 1$, since by the definition of $\lfloor \cdot \rfloor := \sup\{k \in \mathbb{Z} : k \leq \cdot\}$. If this was not the case, then we would have that $\log_2(x - \sum_{i=1}^{n-1} 2^{s^i}) \geq \lfloor \log_2(x - \sum_{i=1}^{n-1} 2^{s^i}) \rfloor + 1 \to \log_2(x - \sum_{i=1}^{n-1} 2^{s^i}) \geq \lfloor \log_2(x - \sum_{i=1}^{n-1} 2^{s^i}) \rfloor \geq \lfloor \log_2(x - \sum_{i=1}^{n-1} 2^{s^i}) \rfloor + 1 \to 0 \geq 1 \to \perp$.

With $\lfloor \log_2(x - \sum_{i=1}^{n-1} 2^{s^i}) \rfloor \leq \log_2(x - \sum_{i=1}^{n-1} 2^{s^i}) < \lfloor \log_2(x - \sum_{i=1}^{n-1} 2^{s^i}) \rfloor + 1$ established, apply $2^{(\cdot)}$, again order is preserved to due its monotonicity on $\mathbb{R}$. Then $2^{\lfloor \log_2(x - \sum_{i=1}^{n-1} 2^{s^i}) \rfloor} \leq 2^{\log_2(x - \sum_{i=1}^{n-1} 2^{s^i})} < 2^{\lfloor \log_2(x - \sum_{i=1}^{n-1} 2^{s^i}) \rfloor + 1}$. Simplifying the expressions we see that $2^{s_n} \leq x - \sum_{i=1}^{n-1} 2^{s^i} < 2 \cdot 2^{s_n}$.

Now consider the geometric series $\sum_{j=|s_n|}^{\infty} 2^{-j}$. Write it as $\sum_{j=1}^{|s_n|-1} 0 \cdot 2^{-j} + \sum_{j=|s_n|}^{\infty} 2^{-j}$. From this its clear that $\sum_{j=1}^{|s_n|-1} 0 \cdot 2^{-j} + \sum_{j=|s_n|}^{\infty} 2^{-j} < \sum_{j=1}^{\infty} 2^{-j}$. Therefore by the comparison test, our geometric series converges since it is term wise bounded by a those of a convergent series. Since all the terms involved are non negative, we have absolute convergence and can perform rearrangements.

Observe that $\sum_{j=1}^{|s_n|-1} 2^{-j} + \sum_{j=|s_n|}^{\infty} 2^{-j} = \sum_{j=1}^{\infty} 2^{-j} = 1$. And subtracting the finite piece gives that $\sum_{j=|s_n|}^{\infty} 2^{-j} = 1 - \sum_{j=1}^{|s_n|-1} 2^{-j} = 1 - \sum_{j=1}^{|s_n|-1} 2^{-j} + 0$. Write 0 as $-2^0 + 1$ and merge the -2^0 into the sum and see that $\sum_{j=|s_n|}^{\infty} 2^{-j} = 1 - \sum_{j=1}^{|s_n|-1} 2^{-j} = 1 - \sum_{j=0}^{|s_n|-1} 2^{-j} + 1$. Similarly merge the $+1$ into the existing 1 we had. Then $\sum_{j=|s_n|}^{\infty} 2^{-j} = 2 - \sum_{j=0}^{|s_n|-1} 2^{-j}$. Using the finite geometric sum formula with common ratio $r = 2^{-1}$, $\sum_{j=|s_n|}^{\infty} 2^{-j} = 2 - \sum_{j=0}^{|s_n|-1} 2^{-j} = 2 - \frac{1-(2^{-1})^{|s_n|}}{1-2^{-1}} = 2 - \frac{1-(2^{-1})^{|s_n|}}{2^{-1}} = \frac{2 \cdot (2^{-1}) - 1 + 2^{-|s_n|}}{2^{-1}} = 2 \cdot 2^{-|s_n|}$.

Therefore $\sum_{j=|s_n|}^{\infty} 2^{-j} = 2 \cdot 2^{-|s_n|}$ and since $s_n < 0 \rightarrow |s_n| = -s_n \rightarrow -|s_n| = s_n$, we have that $\sum_{j=|s_n|}^{\infty} 2^{-j} = 2 \cdot 2^{s_n}$. Since $\forall n \in \mathbb{N} : s_n \in \mathbb{Z}^-$ and monotone decreasing, and unbounded from below, $-s_n$ is monotone increasing and unbounded from above in $\mathbb{N}$. That is, it diverges with $\lim_{n\rightarrow\infty} -s_n = \infty := \forall M \in \mathbb{N}, \exists N \in \mathbb{N} : n \geq N \rightarrow -s_n > M$.

Let $\epsilon > 0$ be arbitrary. Take $M = M(\epsilon) = \lceil |\log_2(\frac{1}{2}\epsilon)| \rceil + 1$. Then since $\lim_{n\rightarrow\infty} -s_n = \infty$, we have that $\exists N = N(\epsilon) : n \geq N \rightarrow -s_n > |\log_2(\frac{1}{2}\epsilon)| \rceil + 1 \rightarrow -s_n > -\log_2(\frac{1}{2}\epsilon) \rightarrow s_n < log_2(\frac{1}{2}\epsilon) \rightarrow 2^{s_n} < \frac{1}{2}\epsilon \rightarrow 2 \cdot 2^{s_n} < \epsilon \rightarrow 0 < 2^{s_n} < x - \sum_{i=1}^{n-1} 2^{s_i} < 2 \cdot 2^{s_n} < \epsilon \rightarrow |x - \sum_{i=1}^{n-1} 2^{s_i}| < \epsilon \rightarrow |x - \sum_{i=1}^{n} 2^{s_i}| < \epsilon$.

This proves $\sum_{i=1}^{n} 2^{s_i}$ (absolutely) converges, and that $x = \lim_{n\rightarrow\infty} \sum_{i=1}^{n} 2^{s_i}$ □

With the convergence of Definition 16 established, and particularly that through Theorem 4.1 it converges absolutely to our desired x, we need to establish how to actually perform representations. In particular we need some analog of bit vectors like we had for $\{0,1\}^{p,q}$, show existence and uniqueness, be able to define operations, and the like. This is almost the last step before we can define and prove various approximation properties that we want to be able to understand and tune for use in fixed point arithmetic. We will see that all this machinery pays off in that fixed point arithmetic, and its precision analysis, is a consequence of everything we have been building towards until now. To define the analog of bit vectors for the infinitely representable numbers, we use an idea from basic functional analysis. In particular, that when moving from finite dimensional vector spaces to infinite dimensional ones, at least for sequence spaces l_p it is very convenient both algebraically and analytically to view them as infinite coordinate vectors as if we were in $\mathbb{R}^n$. We will do the same here and extend Definition 13 to handle *arbitrary precision* bit vectors (Def 20). These are bit vectors with an infinite fractional part, as captured in the following definition. We keep the notation the same as in the case of when we defined $\{0,1\}^{p,q}$ in Def 13 with the change here being that since we have infinite fractional bits to use, we replace q with $\mathbb{N}$ as the second index.

Definition 20. *The Space of Arbitrary Precision Bit Vectors* $\{0,1\}^{p,\mathbb{N}}$
Let $p,\ q \in \mathbb{N},\ n \in \mathbb{Z}$.

Define $(x)_{p,\mathbb{N}} := (x_{p-1}, x_{p-2}, ...x_1, x_0, x_{-1}, ...)$ *where* $i \leq p-1 \rightarrow x_i \in \{0,1\}$

Define $(x)_{p,q}^n := x_n$ *if* $n \leq p-1 \wedge x_n \in (x)_{p,\mathbb{N}}$

Define $x_{p,\mathbb{N}} := -1 \cdot (x)_{p,\mathbb{N}}^{p-1} \cdot 2^{p-1} + \sum_{j=0}^{p-2} (x)_{p,\mathbb{N}}^{j} \cdot 2^j + \sum_{i=1}^{\infty} (x)_{p,\mathbb{N}}^{-i} \cdot 2^{-i}$

Define $\{0,1\}_{+}^{p,\mathbb{N}} := \{(x)_{p,\mathbb{N}} : (x)_{p,\mathbb{N}}^{p-1} = 0\}$

Define $\{0,1\}_{-}^{p,\mathbb{N}} := \{(x)_{p,\mathbb{N}} : (x)_{p,\mathbb{N}}^{p-1} = 1\}$

Define $\{0,1\}^{p,\mathbb{N}} := \{0,1\}_{+}^{p,\mathbb{N}} \bigcup \{0,1\}_{-}^{p,\mathbb{N}}$

Define $x_{p,\mathbb{N}}|_q := -1 \cdot (x)_{p,\mathbb{N}}^{p-1} \cdot 2^{p-1} + \sum_{j=0}^{p-2} (x)_{p,\mathbb{N}}^{j} \cdot 2^j + \sum_{i=1}^{q} (x)_{p,\mathbb{N}}^{-i} \cdot 2^{-i}$

Theorem 4.2. $\mathbb{R}$ *is Infinitely Representable.*
Let $x \in \mathbb{R}$. *Then* x *is infinitely representable. Furthermore,* $\exists p \in \mathbb{N}$ *such that* $\exists\ (x)_{p,\mathbb{N}} \in \{0,1\}^{p,\mathbb{N}}$ *such that* $x_{p,\mathbb{N}} = x$ *and* $\forall q \in \mathbb{N}, \exists! (x^*)_{p,q} \in \{0,1\}^{p,q} : (\forall i \in \mathbb{N}, -q \leq i \leq 1 \rightarrow ((x)_{p,\mathbb{N}}|_q)^i = (x^*)_{p,q}^i) \wedge x_{p,\mathbb{N}}|_q = x_{p,q}^*$

Proof. The first (integer) case will directly use the definitions. The second case ($FP - \mathbb{Z}$) will be mapping to the integer case, applying it, and then mapping back. We will skip the manual reindexing. The third case will be the real task at hand ($\mathbb{R} - FP$). Let $x \in \mathbb{R}$. Suppose $x \in FP - \mathbb{Q}$. Then $x \in \mathbb{Z}$. So we have that $\exists p \in \mathbb{N} : \exists! (x)_p \in \{0,1\}^p : x_p = x$. By definition of the expansion form for integers we have that $x = x_p = (-1) \cdot (x)_p^{p-1} \cdot 2^{p-1} + \sum_{j=0}^{p-2} (x)_p^j \cdot 2^j = (-1) \cdot (x)_p^{p-1} \cdot 2^{p-1} + \sum_{j=0}^{p-2} (x)_p^j \cdot 2^j + \sum_{i=1}^{\infty} 0 \cdot 2^{-i}$. Therefore $x \in \mathbb{Z}$ is infinitely representable. To find a representation for it in $\{0,1\}^{p,\mathbb{N}}$, assign $(x)_{p,\mathbb{N}}^n = (x)_p^n$ if $0 \leq n \leq p-1$, and 0 otherwise, i,e. $(x)_{p,\mathbb{N}} = ((x)_p^{p-1}, (x)_p^{p-2}, ..., (x)_p^1, (x)_p^0, 0, 0, 0...)$. Its clear that by the definition of $x_{p,N}$ we have that $x_{p,\mathbb{N}} = (-1) \cdot (x)_{p,\mathbb{N}}^{p-1} \cdot 2^{p-1} + \sum_{j=0}^{p-2} (x)_{p,\mathbb{N}}^{j} \cdot 2^j + \sum_{i=1}^{\infty} (x)_{p,\mathbb{N}}^{-i} \cdot 2^{-i} =$

$(-1)\cdot(x)_p^{p-1}\cdot 2^{p-1}+\sum_{j=0}^{p-2}(x)_p^j\cdot 2^j=(-1)\cdot(x)_p^{p-1}\cdot 2^{p-1}+\sum_{j=0}^{p-2}(x)_p^j\cdot 2^j+\sum_{i=1}^{\infty}0\cdot 2^{-i}=x_p=x$. Therefore $x_{p,\mathbb{N}}=x$, so $(x)_{p,\mathbb{N}}$ is a suitable representation for x in $\{0,1\}^{p,\mathbb{N}}$

For existence and uniqueness of truncation for this particular representation of x, we have that $x_{p,\mathbb{N}}|_q:=(-1)\cdot(x)_{p,\mathbb{N}}^{p-1}\cdot 2^{p-1}+\sum_{j=0}^{p-2}(x)_{p,\mathbb{N}}^j\cdot 2^j+\sum_{i=1}^{q}(x)_{p,\mathbb{N}}^{-i}\cdot 2^{-i}=x_{p,N}+\sum_{i=1}^{q}0\cdot 2^{-i}$, $\forall q\in\mathbb{N}$. Therefore since padding 0 fractional bits doesn't change the value, for each q, consider
$(x^*)_{p,q}=((x)_p^{p-1},(x)_p^{p-2},...,(x)_p^1,(x)_p^0,0,0...0)$. Then we have that $(x^*)_{p,q}$ is unique in $\{0,1\}^{p,q}$ by the uniqueness of finitely representable numbers for fixed p,q and that $x_p=x_{p,\mathbb{N}}=x_{p,\mathbb{N}}|_q=x_{p,q}^*$.

Now suppose $x\in FP-\mathbb{Z}$. So we have that $\exists p,q\in\mathbb{N}:\exists!(x)_{p,q}\in\{0,1\}^{p,q}:x_{p,q}=x$. Then multiply by 2^q and see that $x\cdot 2^q\in\mathbb{Z}$.
Apply the integer case just proven, and then reindex and scale down correspondingly via multiplication with 2^{-q}. Then we have that $(x)_{p,\mathbb{N}}=((x)_{p,q}^{p-1},(x)_{p,q}^{p-2},...,(x)_{p,q}^1,(x)_{p,q}^0,(x)_{p,q}^{-1},...(x)_{p,q}^{-q},0,0,0,...)$. As well in this case its clear from the definition of $(x)_{p,\mathbb{N}}$ and $x_{p,\mathbb{N}}$ that $x_{p,\mathbb{N}}=x$, therefore $(x)_{p,\mathbb{N}}$ is a suitable representation for x in $\{0,1\}^{p,\mathbb{N}}$.

For existence and uniqueness of truncation for this particular representation of x, use $q^*\in\mathbb{N}$ as the truncation variable. Then we have that $1\le q^*<q\to x_{p,\mathbb{N}}|_{q^*}=x_{p,q}-\sum_{i=q^*+1}^{q}(x)_{p,q}^i\cdot 2^{-i}$, in which case take $(x^*)_{p,q^*}\in\{0,1\}^{p,q^*}$ where $(x^*)_{p,q^*}^n=(x)_{p,q}^n$ if $-q^*\le n\le p-1$ and $(x^*)_{p,q^*}^n=0$ otherwise. As well $q^*\ge q\to x_{p,\mathbb{N}}|_{q^*}=x_{p,q}$ in which case take $(x^*)_{p,q^*}\in\{0,1\}^{p,q^*}$ where $(x^*)_{p,q^*}^n=(x)_{p,q}^n$ if $-q\le n\le p-1$ and $(x^*)_{p,q^*}^n=0$ otherwise. Again, we have that each of the $(x^*)_{p,q^*}$ are unique in their respective spaces $\{0,1\}^{p,q^*}$ due to the uniqeness of finite representability for fixed p,q^* which derives from bitwise equivalence with the corresponding integer $(x^*2^{q^*})_{p+q^*}$(we proved this through the bitwise equivalence lemmas, it is just being restated to enforce that the build up and formalism was with intent).

Finally, suppose $x\in\mathbb{R}-FP$. Consider $\lfloor|x|\rfloor$ and $(|x|-\lfloor|x|\rfloor)$. Since $0<\lfloor|x|\rfloor\le|x|\to 0<(|x|-\lfloor|x|\rfloor)<1$. Apply 4.1 and get the corresponding sequence s_n. Then $(|x|-\lfloor|x|\rfloor)=\lim_{n\to\infty}\sum_{i=1}^{n}2^{s_i}$. Since $\lfloor|x|\rfloor\in\mathbb{N}\bigcup\{0\}$, it is finitely representable, we can invoke the definition and find $H\subset\mathbb{N}:\lfloor|x|\rfloor=\sum_{j=0}^{|H|-1}2^j$. Then adding the 2 pieces gives that $|x|=\lfloor|x|\rfloor+(|x|-\lfloor|x|\rfloor)=\sum_{j=0}^{|H|-1}2^j+\sum_{i=1}^{\infty}2^{s_i}$. We skip reindexing since we want the seperation intact for the remainder, but its clear that there exists one by setting the first $|H|$ elements of the new one, to be the descending sorted elements of H, and the subsequent elements to be the index ordered elements of s_n. Therefore as with the other two cases, x is infinitely representable.

There exists two cases to handle to show the coordinate representation of x. Namely that if $x>0$, then $|x|=x$ and if $x<0$, $|x|=-x$. To proceed in finding a representation, we will work with just $|x|$ and then branch at appropriate moments. The route to the solution will involve techniques we developed to solve for bitwise equivalence for the negative case.

Since x is infintely representable, its clear that we can subtract finitely many pieces from $|x|$ and see that the remainder is still infinitely representable. In particular that $(|x|-\lfloor|x|\rfloor)$ is infinitely representable, using its own series expansion. Since each of $s_n<0$, we have that each of $-s_n\ge 1$. In particular that $\{-s_n\}_{n=1}^{\infty}\subset\mathbb{N}$. Furthermore it cannot be empty since $\{-s_n\}_{n=1}^{\infty}=\emptyset\to x\in\mathbb{Z}\to\perp$. And it cannot be all of $\mathbb{N}$ since $\{-s_n\}_{n=1}^{\infty}=\mathbb{N}\to\{s_n\}_{n=1}^{\infty}=\mathbb{Z}^-\to(|x|-\lfloor|x|\rfloor)=1\to\perp$. Therefore $\mathbb{Z}^--\{s_n\}_{n=1}^{\infty}\neq\emptyset$.

Since $\lfloor|x|\rfloor$ is finitely representable we have that $\exists p\in\mathbb{N}:\exists!(\lfloor|x|\rfloor)_p\in\{0,1\}^p:\lfloor|x|\rfloor_p=\lfloor|x|\rfloor$ Note $\lfloor|x|\rfloor=0$ is fine since in that case the maximum on the MSB of $(|x|-\lfloor|x|\rfloor)$ would be -1 and adding 2 (one bit for the next hop to the sign bit, and another for the actual p) would be 1. Construct $(|x|)_{p,\mathbb{N}}$ as follows. $(|x|)_{p,\mathbb{N}}^n=(\lfloor|x|\rfloor)_p^n=$ if $0\le n\le p-1$, and $(|x|)_{p,\mathbb{N}}^n=0$ if $n\in\mathbb{Z}^--\{s_k\}_{k=1}^{\infty}$. Otherwise $n\in\{s_k\}_{k=1}^{\infty}$ so take $(|x|)_{p,\mathbb{N}}^n=1$. Then by construction we have that $|x|_{p,\mathbb{N}}=(-1)\cdot(|x|)_{p,\mathbb{N}}^{p-1}\cdot 2^{p-1}+\sum_{j=0}^{p-2}(|x|)_{p,\mathbb{N}}^j\cdot 2^j+\sum_{i=1}^{\infty}(|x|)_{p,\mathbb{N}}^{-i}\cdot 2^{-i}=\lfloor|x|\rfloor+(|x|-\lfloor|x|\rfloor)=|x|$. This shows $(|x|)_{p,\mathbb{N}}\in\{0,1\}^{p,\mathbb{N}}$ is a suitable representation for $|x|$. Now we proceed with the cases. The case that $|x|=x\iff x>0$ will in part be used to handle the case that $|x|=-x\longleftrightarrow x<0$. Suppose $x>0$, then since $x=|x|=\lfloor|x|\rfloor+(|x|-\lfloor|x|\rfloor)=|x|_{p,\mathbb{N}}=x_{p,\mathbb{N}}$ we have that:
$x=(-1)\cdot(x)_{p,\mathbb{N}}^{p-1}\cdot 2^{p-1}+\sum_{j=0}^{p-2}(x)_{p,\mathbb{N}}^j\cdot 2^j+\lim_{q\to\infty}\sum_{i=1}^{q}(x)_{p,\mathbb{N}}^{-i}\cdot 2^{-i}$.

To show the truncations, consider the sequence of partial sums of the series expansion of x that we just showed. More explicitly, let $q\in\mathbb{N}$. The numerical sequence formed by $\{x_{p,N}|_q\}_{q=1}^{\infty}$ is exactly the sequence of partial sums of the series expansion of x. Define the sequence of bit vectors $\{(x^*)_{p,q}\}_{q=1}^{\infty}$ as we did in the case of $FP-\mathbb{Z}$ with $(x^*)_{p,q}\in\{0,1\}^{p,q}$ where $(x^*)_{p,q}^n=(x)_{p,\mathbb{N}}^n$ for $-q\le n\le p-1$ and $(x^*)_{p,q}^n=0$ otherwise. As with the case of $FP-\mathbb{Q}$ and $FP-\mathbb{Z}$ each of the $\{(x^*)_{p,q}\}_{q=1}^{\infty}$ are unique in their original spaces, and

$\{x_{p,N}|_q\}_{q=1}^{\infty} = \{x^*_{p,q}\}_{q=1}^{\infty}$ term-wise.

For the case of $x < 0$ we have that $|x| = -x > 0$. Using the positive case just established, we can represent it as we did above with $(-x)^n_{p,\mathbb{N}} = (|x|)^n_{p,\mathbb{N}}$. Similarly we have the existence and uniqueness of the sequence of truncations due to $-x$ being positive. So consider the corresponding 3 sequences: the first being the sequence of truncations $\{-x_{p,N}|_q\}_{q=1}^{\infty}$, the second being its corresponding sequence of numerical truncations $\{-x^*_{p,q}\}_{q=1}^{\infty}$ and the final being the sequence of unique bit vectors derived from sequence of numerical truncations $\{(-x^*)_{p,q}\}_{q=1}^{\infty}$.

Using one of our previously proven lemmas regarding bitwise equivalence for negative numbers (Lemma 3.3) we have that $\forall q \in \mathbb{N}, \exists!$ additive inverse $(x^*)_{p,q} \in \{0,1\}^{p,q}_{-}$ to $(-x^*)_{p,q} \in \{0,1\}^{p,q}_{+}$ with $(x^*)^n_{p,q} = (x^* \cdot 2^q)^n_{p+q} = \neg(-x^* \cdot 2^q)^n_{p+q}\ XOR\ (1)^n_{p+q}\ XOR\ carry(n-1, \neg(-x^* \cdot 2^q)_{p+q}, (1)_{p+q})$ and $(x^*)_{p,q} \bigoplus (-x^*)_{p,q} = (0)_{p,q}$. Form the sequence of additive inverses of $\{(-x^*)_{p,q}\}_{q=1}^{\infty}$ and write it as $\{(x^*)_{p,q}\}_{q=1}^{\infty}$.

Observe that since we have established in the positive case that the sequence of the numerical truncations is term-wise equal to the expansion forms of the bit vectors derived from the truncations, i.e. that $\{-x_{p,N}|_q\}_{q=1}^{\infty} = \{-x^*_{p,q}\}_{q=1}^{\infty}$ term-wise and that $(-x)^n_{p,N} = (-x^*)^n_{p,q}$ for $-q \leq n \leq p-1$, we can prove that taking $(x)_{p,N}$ such that $(x)^n_{p,N} = (x^*)^n_{p,q}$ for $-q \leq n \leq p-1$ is a suitable representation for x. First, the assignment(over $q \in \mathbb{N}$) is unique in the sense that $\forall q_1, q_2 \in \mathbb{N} : q_1 < q_2 \rightarrow -q_1 \leq n \leq p-1 \rightarrow (x^*)^n_{p,q_1} = (x^*)^n_{p,q_2}$. This comes from the fact that since we built the sequence $\{(x^*)_{p,q}\}_{q=1}^{\infty}$ as the unique additive inverses of the bit vectors $\{(-x^*)_{p,q}\}_{q=1}^{\infty}$ that component wise equal the fixed point representations of the truncations $\{-x_{p,N}|_q\}_{q=1}^{\infty}$ of $-x$, we have no "double" assignment.

Now we have to show that $x_{p,\mathbb{N}}$ constructed as above satisfied that $x_{p,\mathbb{N}} = x$. In particular that $x = (-1) \cdot (x)^{p-1}_{p,\mathbb{N}} \cdot 2^{p-1} + \sum_{j=0}^{p-2} (x)^j_{p,\mathbb{N}} \cdot 2^j + \lim_{q\rightarrow\infty} \sum_{i=1}^{q} (x)^{-i}_{p,\mathbb{N}} \cdot 2^{-i}$. Yet another equivalent way to phrase is that we want to show $\lim_{q\rightarrow\infty} x^*_{p,q} = x$.

Its actually quite simple using the machinery we've built. We know that $\lim_{q\rightarrow\infty} -x^*_{p,q} = -x$. We will use its convergence and kidnap its N. Recall Lemma 3.3, then due to the uniqueness of negation in $FP(p,q)$ we have that $-(-x^*_{p,q}) = (x^*_{p,q})$. Therefore by the definition of the limit of numerical sequences in $\mathbb{R}$(equivalently the limit of numerical series as limits of the sequences of their partial sums) we have that $\forall \epsilon > 0, \exists N = N(\epsilon) \in \mathbb{N} : q \geq N \rightarrow |(-x) - (-x^*_{p,q})| < \epsilon \rightarrow |(-x^*_{p,q}) - (-x)| < \epsilon \rightarrow |-1||(-x^*_{p,q}) - (-x)| < \epsilon \rightarrow |-1||(-x^*_{p,q}) + x)| < \epsilon \rightarrow |-(-x^*_{p,q}) - x| < \epsilon \rightarrow |x^*_{p,q} - x| < \epsilon \rightarrow |x - x^*_{p,q}| < \epsilon$.

Therefore $\lim_{q\rightarrow\infty} x^*_{p,q} = x$ where $x \in \mathbb{R} - FP$ and $x < 0$ which was the last case to handle. Therefore the theorem is true $\forall x \in \mathbb{R}$. □

5 Approximation of the Real Numbers and their arithmetic

Recall that $\forall x \in \mathbb{R} : \exists!(x)_{p,q} \in \{0,1\}^{p,q} : x_{p,\mathbb{N}}|_q = x_{p,q}$. This is the abstraction of $(x)_{p,q}$ being the $P.Q$ format: the Fixed Point integer approximation of $x \in \mathbb{R}$. The truncate operator $|_q$ is intuitively a projection from $\mathbb{R}$ to $FP(p,q)$, and involves truncating $(x)_{p,\mathbb{N}}$ to $(x)_{p,q}$. Furthermore, the truncation operator seems to capture some metric/topological properties as well. In particular, we can define convergence of sequences of bit vectors in FP^* and operators on them through the truncate operator. The convergence of sequences in FP^*, at least for the case of sequences of truncations with the same "limit", corresponds to the classic definition for the Cauchy criterion for the convergence of numerical series in $\mathbb{R}$. We will not state and prove what we believe to be the most abstract generalization, but our intuition is that it is a particular instantiation of one of the variants of Fourier Series, and we will investigate this intuition further in future versions of this report. In-spite of that, we hope that the technique we used to prove the infinite representability of negative reals at the end of Theorem 4.2 gives some motivation to the sequential approach we have been proposing to approximate $\mathbb{R}$ by FP and FP^*. Now to begin with a collection of useful approximation propositions. We begin with our framing of the classical approximation property of dyadic rational number theory. We call it the Point-wise Bit Lemma.

Lemma 5.1. *Point-wise Bit Lemma*
Let $x \in \mathbb{R}$ and consider its representation $(x)_{p,\mathbb{N}} \in \{0,1\}^{p,\mathbb{N}}$ and its sequence of numerical truncations $\{x_{p,q}\}_{q=1}^{\infty}$. The following error bound holds:

$$\forall x \in \mathbb{R}, \forall \epsilon > 0 : q \geq \lceil |\log_2(\tfrac{1}{\epsilon})| \rceil + 1 \rightarrow |x - x_{p,q}| < \tfrac{1}{2^q} < \epsilon$$

Proof. Suppose $q \in \mathbb{N} : q \geq \lceil |\log_2(\frac{1}{\epsilon})| \rceil + 1$. Then $q > \log_2(\frac{1}{\epsilon})$. Which implies $-q < -\log_2(\frac{1}{\epsilon})$. Apply the function $2^{(\cdot)}$. Then $2^{-q} < 2^{-\log_2(\frac{1}{\epsilon})}$. Expand the logarithm. Then $2^{-q} < 2^{-(\log_2(1)-\log_2(\epsilon)} = 2^{-(\log_2(1)+\log_2(\epsilon)} = 2^{-(\log_2(1)} \cdot 2^{\log_2(\epsilon)} = 2^{log_2(1^{-1})} \cdot 2^{\log_2(\epsilon)} = \epsilon$.

So $2^{-q} < \epsilon$. By basic properties of geometric series write 2^{-q} as $\sum_{i=q+1}^{\infty} 1 \cdot 2^{-i} = 2^{-q} < \epsilon$. Apply the Infinite Representability of $\mathbb{R}$ through Theorem 4.2 to x retrieve a representation $(x)_{p,\mathbb{N}}$. Then since $x_{p,\mathbb{N}} = x$ and $x_{p,\mathbb{N}} \geq x_{p,q}$ we have that $x - x_{p,q} \geq 0$. Finally: $x - x_{p,q} = |x - x_{p,q}| = |\sum_{i=q+1}^{\infty} (x)_{p,\mathbb{N}}^{-i} \cdot 2^{-i}| \leq \sum_{i=q+1}^{\infty} |(x)_{p,\mathbb{N}}^{-i} \cdot 2^{-i}| \leq \sum_{i=q+1}^{\infty} |1 \cdot 2^{-i}| = \sum_{i=q+1}^{\infty} 1 \cdot 2^{-i} = 2^{-q} = \frac{1}{2^q} < \epsilon$ □

We now provide some basic propositions that can be used to reason about how many bits we need to perform various operations in fixed point arithmetic. We use the Point-wise Bit Lemma as an error estimator, when really it is just an explicit statement about the property of the sequence of projections of $x \in \mathbb{R}$ onto FP^*, that by stepping through each fixed point space $FP^*(p, q)$ and adding one more fractional bit, has the effect that the subsequent terms in the sequence of projections re-accumulate enough precision so that they converge to the corresponding $(x)_{p,\mathbb{N}}$ in the closure of FP^*. Our ambition is to show that if the conjectures at the beginning of the report hold then we can generate sequences of fixed point numbers that converge to real numbers that match the intuition of taking a limit on the number fractional bits q of some fixed point approximation of some real number. And that on that basis, precision analysis on fixed point types can be reduced to reproving almost the same basic limit theorems for arithmetic on convergent sequences that are encountered early in an undergraduate real analysis course [1, Theorem 2.3.3 (Algebraic Limit Theorem)]. We turn to this next.

Proposition 3. *Addition up to ϵ*
Denote $||(x)_{p,q} \bigoplus (y)_{p,q}||$ to be the expansion form of $(x)_{p,q} \bigoplus (y)_{p,q}$ i.e. $(x)_{p,q} \bigoplus (y)_{p,q_{p,q}}$. Then the following error bound holds:

$$\forall x, y \in \mathbb{R}, \forall \epsilon > 0 : p \geq \max(\lceil \log_2(|x| + 1) \rceil, \lceil \log_2(|y| + 1) \rceil) + 2, q \geq \lceil |\log_2(\tfrac{1}{1/2\epsilon})| \rceil + 1 \rightarrow |(x + y) - ||(x)_{p,q} \bigoplus (y)_{p,q}||| < \epsilon$$

Proof. The reason for $p \geq \max(\lceil \log_2(|x| + 1) \rceil, \lceil \log_2(|y| + 1) \rceil) + 2$ that $\bigoplus$ as in Def 8 and Lemma 3.4 guarantee overflow does not occur when performing the addition by mapping $(x)_{p,q}$ and $(y)_{p,q} \in FP^*(p, q)$ to a one-higher dimensional space. In particular that $(x)_{p,q} \bigoplus (y)_{p,q} = (x + y)_{p+1,q} \in FP^*(p + 1, q)$ due to the carry flag as defined in 7. And as we have seen in 3.4, representations using one more integer bit than needed have the property that addition does not overflow padding the front by the sign bit does not change the value. So then addition corresponds to true addition. In any case, apply the Infinite Representability of $\mathbb{R}$ through Theorem 4.2 to x and y using p integer bits. Retrieve representations $(x)_{p,\mathbb{N}}$ for x and $(y)_{p,\mathbb{N}}$ for y.

For $\epsilon > 0$, take $\epsilon_0 = \frac{1}{2}\epsilon$, and apply the Point-wise Bit Lemma to x, y, and ϵ_0. Then $|x - x_{p,q}| < \frac{1}{2}\epsilon$ and $|y - y_{p,q}| < \frac{1}{2}\epsilon$. Adding the pieces together gives that $|x - x_{p,q}| + |y - y_{p,q}| < \frac{1}{2}\epsilon + \frac{1}{2}\epsilon = \epsilon$. Apply the triangle inequality. Then $|x + y - x_{p,q} - y_{p,q}| \leq |x - x_{p,q}| + |y - y_{p,q}| < \epsilon$. Factor the negatives out and see that $|(x + y) - (x_{p,q} + y_{p,q})| < \epsilon$. Apply the Finitely Representable Addition Lemma 3.4 to $(x)_{p,q}$ and $(y)_{p,q}$ and see that $x_{p,q} + y_{p,q} = ||(x)_{p,q} \bigoplus (y)_{p,q}||$. Then:
$|(x + y) - (x_{p,q} + y_{p,q})| < \epsilon \rightarrow |(x + y) - ||(x)_{p,q} \bigoplus (y)_{p,q}||| < \epsilon$ □

Proposition 4. *Subtraction up to ϵ*
Denote $||(x)_{p,q} \bigoplus (-y)_{p,q}||$ to be the expansion form of $(x)_{p,q} \bigoplus (-y)_{p,q}$ i.e. $(x)_{p,q} \bigoplus (-y)_{p,q_{p,q}}$. Then the following error bound holds:

$$\forall x, y \in \mathbb{R}, \forall \epsilon > 0 : p \geq \max(\lceil \log_2(|x| + 1) \rceil, \lceil \log_2(|y| + 1) \rceil) + 2, q \geq \lceil |\log_2(\tfrac{1}{1/2\epsilon})| \rceil + 1 \rightarrow |(x - y) - ||(x)_{p,q} \bigoplus (-y)_{p,q}||| < \epsilon$$

Proof. Since $x - y = x + (-y)$, apply Proposition 3 to x and $(-y)$. □

Proposition 5. *Multiplication up to ϵ*
Denote $||(x)_{p,q} \bigotimes (y)_{p,q}||$ to be the expansion form of $(x)_{p,q} \bigotimes (y)_{p,q}$ i.e. $(x)_{p,q} \bigotimes (y)_{p,q_{p,q}}$. for $\epsilon > 0$ and 2

real numbers x, y denote $\epsilon_0(\epsilon, x, y) = \min(\frac{\epsilon}{|x|+1}, \frac{\epsilon}{|y|+1})$ Then the following error bound holds:

$\forall x, y \in \mathbb{R}, \forall \epsilon > 0 : p \geq 2 \cdot \max(\lceil \log_2(|x|+1) \rceil, \lceil \log_2(|y|+1) \rceil) + 1, q \geq 2 \cdot (\lceil |\log_2(\frac{1}{(1/2)\epsilon_0(\epsilon,x,y)})| \rceil + 1) \rightarrow |(x+y) - ||(x)_{p,q} \bigotimes (y)_{p,q}||| < \epsilon$

Proof. We plan to set the values of p, q as doubles of what they normally should be for proving that the limit of the product of 2 convergent sequences is the product of the limits—hence the $\epsilon_0(\epsilon, x, y)$ specification—and simultaneously be able to easily invoke the Finitely Representable Multiplication Lemma 3.5. We will do the first part first and the second part second.

Using x, y, p as in the hypothesis, apply the Infinite Representability of $\mathbb{R}$ through Theorem 4.2 to x and y using p integer bits and retrieve representations $(x)_{p,\mathbb{N}}$ for x and $(y)_{p,\mathbb{N}}$ for y. For $\epsilon > 0$, take $\epsilon_1 = \frac{1}{2}\epsilon_0(\epsilon, x, y)$ where $\epsilon_0 = \min(\frac{\epsilon}{|x|+1}, \frac{\epsilon}{|y|+1})$ and apply the Point-wise Bit Lemma to x, y, and q as in the hypothesis along with ϵ_1. Since q is double the required number of fractional bits specified by the Point-wise Bit Lemma, its conclusion still holds, then we have $|x - x_{p,q}| < \frac{1}{2^q} < \epsilon_1$ and $|y - y_{p,q}| < \frac{1}{2^q} < \epsilon_1$. We now make use of a fundamental property about distance in $\mathbb{R}$ and metric spaces in general, called the *reverse triangle inequality*. The reverse triangle inequality states that $\forall a, b \in \mathbb{R} : |a| - |b| \leq ||a| - |b|| \leq |a - b|$. Apply the reverse triangle inequality to each piece, and consider the following sequence of comparisons: $|x_{p,q}| - |x| \leq ||x| - |x_{p,q}|| \leq |x - x_{p,q}| < \frac{1}{2^q} < \epsilon_1$ and $|y_{p,q}| - |y| \leq ||y| - |y_{p,q}|| \leq |y - y_{p,q}| < \frac{1}{2^q} < \epsilon_1$

Then we have that $|x_{p,q}| < |x| + \frac{1}{2^q}$ and $|y_{p,q}| < |y| + \frac{1}{2^q}$ and that since $q \in \mathbb{N}$ we have $|x_{p,q}| < |x| + \frac{1}{2^q} < |x|+1$ and $|y_{p,q}| < |y| + \frac{1}{2^q} < |y|+1$. For each inequality, divide by the right most piece, which in turn implies each of $\frac{|y_{p,q}|}{|y|+1}$ and $\frac{|x_{p,q}|}{|x|+1}$ are < 1. Now see that $|x_{p,q} \cdot y_{p,q} - xy| = |x_{p,q} \cdot y_{p,q} - (x_{p,q} \cdot y) + (x_{p,q} \cdot y) + xy| \leq |x_{p,q}||y - y_{p,q}| + |y||x - x_{p,q}| \leq |x_{p,q}| \cdot \epsilon_1 + |y| \cdot \epsilon_1 \leq \frac{1}{2}(\frac{|x_{p,q}| \cdot \epsilon}{|x|+1} + \frac{|y| \cdot \epsilon}{|y|+1}) < \frac{1}{2}(\epsilon + \epsilon) = \epsilon$. Now apply the Finitely Representable Multiplication Lemma 3.5 that since our choice of p, q match the configuration in the operation's construction—that is $p^* = \max(p_x, p_y), p = 2 \cdot p^* + 1, q = 2 \cdot \max(q_x, q_y)$—we meet the no-overflow condition and have that $||(x)_{p,q} \bigotimes (y)_{p,q}|| = x_{p,q} \cdot y_{p,q}$. Then since we have proved as well that $|x_{p,q} \cdot y_{p,q} - xy| < \epsilon$ we have that $|xy - ||(x)_{p,q} \bigotimes (y)_{p,q}||| < \epsilon$. □

Proposition 6. *Multiplicative Inversion up to ϵ*
Denote $||(x)_{p,q} \bigotimes (\frac{1}{x})_{p,q}||$ to be the expansion form of $(x)_{p,q} \bigotimes (\frac{1}{x})_{p,q}$ i.e. $(x)_{p,q} \bigotimes (\frac{1}{x})_{p,q_{p,q}}$. for a real number $x \neq 0$ and for $\epsilon > 0 : \epsilon < |\frac{1}{x}|$ denote $\epsilon_0(\epsilon, x, \frac{1}{x}) = \min(\frac{\epsilon}{|x|+1}, \frac{\epsilon}{|\frac{1}{x}|+1})$ Then the following error bound holds:

$\forall x \in \mathbb{R} - \{0\}, \forall \epsilon > 0$ *such that* $0 < \epsilon < |\frac{1}{x}| : p \geq 2 \cdot \max(\lceil \log_2(|x|+1) \rceil, \lceil \log_2(|\frac{1}{x}|+1) \rceil) + 1, q \geq 2 \cdot (\lceil |\log_2(\frac{1}{(1/2)\epsilon_0(\epsilon, x, \frac{1}{x})})| \rceil + 1) \rightarrow |1 - ||(x)_{p,q} \bigotimes (\frac{1}{x})_{p,q}||| < \epsilon$

Proof. The fundamental issue is that $FP(p, q)$ and $FP^*(p, q)$ for fixed p, q are not closed under any operation due to overflow or underflow. With multiplicative inversion, the issue is that even for simple numbers in some $F(p, q)$, they need infinite bits to be able to invert. The space we constructed $\{0, 1\}^{p,\mathbb{N}}$ can always be good enough, as we saw in the proof of $\mathbb{R}$'s Infinite Representability (Theorem 4.2).

We see that now with multiplicative inversion. The requirement $0 < \epsilon < |\frac{1}{x}|$ in the hypothesis is so that we can guarantee we never produce bit strings that are 0 and call them sufficient. Therefore take the configuration as in the hypothesis and apply the Proposition 5 on multiplication up to ϵ. □

6 Parameter selection

Now we discuss how to select the three parameters (N, p, q) to approximate intervals of the real line up to some arbitrary precision level ϵ. In particular we do it from the perspective of uniformity. In point wise convergence we have that the the number of steps N we need to take on the limit to get let $|f_N(x) - f(x)| < \epsilon$ can change depending on the argument x. In other words, if f_N is a sequence of functions that converges point wise to a limiting function f, then the convergence occurs at different rates at different parts of the domain. This is undesirable from both a pure theoretical perspective and a practical numerical algorithm design and implementation perspective. The reason is that there are basic properties we would like to be true, such as a limit of a series of continuous functions is itself a continuous function, or that the series formed by derivative/integral of a series of functions converge to the derivative/integral of the limit of the original series of functions.

As it turns out, the properties stated previously are false for point wise limits of continuous functions. With that in mind, we now set up some necessary infrastructure and then prove a uniformly convergent variant of Taylor's Theorem for $x \in FP \bigcap [a,b]$ where $[a,b]$ is a closed sub-interval of the interval of convergence of the Taylor Series of some real analytic f.

Lemma 6.1. *Uniform Bit Lemma*
Let $[a,b] \subset \mathbb{R}$.
$\forall \epsilon > 0, p \geq \max\left(\lceil log_2(|a|+1)\rceil, \lceil log_2(|b|+1)\rceil\right)+1,\ q \geq \lceil |log_2 \frac{1}{\epsilon}|\rceil + 1 : x \in [a,b] \longrightarrow |x - x_{p,q}| < \frac{1}{2^q} < \epsilon.$

Proof. Take $\epsilon > 0, x \in [a,b], p$ and q as above. Observe that $\min(|a|,|b|) \leq |x| \leq \max(|a|,|b|)$. The non decreasing nature of $|x| \to log_2(|x|+1)+1$ implies that $\min(log_2(|a|+1), log_2(|b|+1))+1 \leq log_2(|x|+1)+1 \leq \lceil log_2(|x|+1)\rceil + 1 \leq \max(log_2(|a|+1), log_2(|b|+1)) + 1 \leq p$. Therefore $\forall x \in [a,b]$, p as in the hypothesis satisfies the minimal number of bits needed to represent the integer part of x. Then applying the Point-wise Bit Lemma gives $|x - x_{p,q}| < \frac{1}{2^q}\epsilon$. □

Definition 21. *Uniform Continuity*
Let $D \subset \mathbb{R}$ *and* $f : D \longrightarrow \mathbb{R}$.
Then f *is Uniformly Continuous if* $\forall \epsilon > 0, \exists \delta > 0, \forall x, y \in D : |x - y| < \delta \to |f(x) - f(y)| < \epsilon$.

Lemma 6.2. *Modulus of Uniform Continuity*
$e_k(x) : [a,b] \longrightarrow \mathbb{R}$ *given by* $x \to x^k$ *is uniformly continuous with* $\delta = \epsilon / \sup_{z \in [a,b]} |\frac{d}{dx} e_k(z)|$

Proof. Let $\epsilon > 0$. Denote $e_k^{'}(x) = \frac{d}{dx} e_k(x)$. Since $e_k^{'}(x)$ is continuous on $[a,b]$ it obtains its supremum. Let $|x - y| < \delta$, then by the Mean Value Theorem there exists c such that the following holds: $|f(x) - f(y)| = |e_k^{'}(c)||x - y| \leq \sup_{z \in [a,b]} |e_k^{'}(z)||x - y| < \epsilon$. □

Theorem 6.3. *Uniform Taylor's Theorem for* $x \in FP$
Let f *be a real analytic function whose Taylor Series has radius of convergence* $r > 0$*, is centered at* c *and let* $[a,b] \subset (c - r, c + r)$ *that contains* c*, then* $\forall \epsilon > 0, \exists K = K(\epsilon) \in \mathbb{N}, \exists p(N) \in \mathbb{N}, \exists q(N) \in \mathbb{N}, \forall x \in [a,b] :$
$N \geq K, p \geq p(N), q \geq q(N) \to |T_N^{p,q}(x_{p,q}) - f(x)| < \epsilon$

Proof. (Sketch) For $x \in [a,b]$, $N \in \mathbb{N}$, define the Nth order Taylor polynomial of f as $T_N(x) := \sum_{k=0}^{N} \frac{f^k(c)}{k!}(x - c)^k$. When f is restricted to $[a,b]$ denote $R_N(x) = f(x) - T_N(x)$ as the Nth order remainder.

Since f is real analytic, it equals Taylor Series on the interval of convergence, therefore $\forall x \in [a,b] :$ $|f(x) - T_N(x)| = |R_N(x)| = |\sum_{k=N+1}^{\infty} \frac{f^k(c)}{k!}(x-c)^k|$. Since $T_N(x)$ converges point-wise to f on $[a,b]$, $R_N(x)$ converges point-wise to 0 on the same domain. Therefore by this useful theorem regarding the uniform convergence of point-wise convergent power series on compact sets [1, Theorem 6.5.5], since $[a,b]$ is compact, T_N, R_N converge uniformly to $f, 0$ on $[a,b]$. Therefore $\exists N, \forall x \in [a,b], |f(x) - T_N(x)| = |R_N(x)| < \epsilon/2$.

Let $\epsilon_0 = \epsilon/2(N+1)$. For $0 \leq k \leq N$, denote $\epsilon_k = \min(\epsilon_0/(2|b-a|^k + 1), \min(\epsilon_0/(2|\frac{f^k(c)}{k!}| + 1))$ and consider the polynomials $e_k(x)$ from Lemma 6.2 on our set $[a,b]$ and for each k apply Lemma 6.2 using ϵ_k and label the resulting sequence δ_k that is sufficient for $e_k(x)$'s uniform continuity. Now since N is finite, denote $\delta = \min_{0 \leq k \leq N}(\min(\delta_k, \epsilon_k))$ and apply Lemma 6.1 using $\delta/2, [a,b], \{[\frac{f^k(c)}{k!}, \frac{f^k(c)}{k!}]\}_{k=0}^{k=N}$ which produces a sequence $\{p_i\}_{i=0}^{i=N+1}$ of integer bit sizes, and $\{q_i\}_{i=0}^{i=N+1}$ of fractional bit sizes, corresponding to the sizes necessary for $\delta/2$ precision in each of our sets. Then take p, q as the respective maximums.

Therefore by Lemma 6.1 along with the construction of δ it follows that $\forall x \in [a,b], \forall k : |x - x_{p,q}| < \delta/2$, $|c - c_{p,q}| < \delta/2, |\frac{f^k(c)}{k!} - \frac{f^k(c)}{k!}_{p,q}| < \epsilon_k$. Adding the parts for x and c gives $|(x_{p,q} - c_{p,q}) - (x - c)| < \delta$ which by Lemma 6.2 implies $|(x_{p,q} - c_{p,q})^k - (x - c)^k| < \epsilon_0/(2|\frac{f^k(c)}{k!}| + 1)$.

Now multiply it by $|\frac{f^k(c)}{k!}_{p,q}|$ and $|\frac{f^k(c)}{k!} - \frac{f^k(c)}{k!}_{p,q}|\ |(x_{p,q} - c_{p,q})^k|$. Then add the 2 pieces, apply the triangle inequality and observe the following: $|\frac{f^k(c)}{k!}_{p,q}(x_{p,q} - c_{p,q})^k - \frac{f^k(c)}{k!}(x-c)^k| < \epsilon_0 |x_{p,q} - c_{p,q}|^k/(2|b-a|^k + 1)$ $+\epsilon_0 |\frac{f^k(c)}{k!}_{p,q}|/(2|\frac{f^k(c)}{k!}| + 1) < 2\epsilon_0/2 < \epsilon_0$.

To finish, since $\forall k : |\frac{f^k(c)}{k!}_{p,q}(x_{p,q} - c_{p,q})^k - \frac{f^k(c)}{k!}(x - c)^k| < \frac{\epsilon}{2(N+1)}$ observe that by addition and generalized triangle inequality, $|T_N^{p,q}(x) - T_N(x)| \leq \sum_{k=0}^{N} |\frac{f^k(c)}{k!}_{p,q}(x_{p,q} - c_{p,q})^k - \frac{f^k(c)}{k!}(x - c)^k| < (N +$

1)$\epsilon_0 = \epsilon/2$ One final time, add $|T_N^{p,q}(x_{p,q}) - T_N(x)|$ with $|T_N(x) - f(x)|$ and observe $|T_N^{p,q}(x_{p,q}) - f(x)| \leq |T_N^{p,q}(x_{p,q}) - T_N(x)| + |T_N(x) - f(x)| < \epsilon/2 + \epsilon/2 = \epsilon$. □

References

[1] Stephen Abbott. *Understanding Analysis.* Springer, New York, 2016.

[2] Shivam Patel, Rigden Atsatsang, Kenneth M. Tichauer, Michael H. L. S. Wang, James B. Kowalkowski, and Nik Sultana. In-Network Fractional Calculations using P4 for Scientific Computing workloads. In *Proceedings of the 5th P4 Workshop in Europe (To appear)*, EuroP4'22. Association for Computing Machinery, 2022.