THE MYTHOLOGY OF INFINITY

Andrew Banks

Abstract

Cantor's infinity (CI) depends on the capability of completing an infinite collection of successive steps (ICSS). Otherwise, CI does not solve Zeno's dichotomy paradox and all natural numbers cannot be upward constructed using the successor function. In spite of the fact that no one has ever scientifically witnessed an ICSS nor has anyone ever furnished a direct proof demonstrating its theoretical existence, the assumed capability of completing an ICSS rests at the foundations of theoretical mathematics. As such, this article will introduce a very general type of accelerated Turing machine that can execute instructions in any given time including zero seconds. This device will demonstrate that the assumption that an ICSS can be completed contradicts the assumption that the natural numbers are unbounded (NNU). Hence, CI is not consistent since it contains both. Furthermore, using this computing device and other machinery, it will be proven that space and time are only finitely divisible.

Keywords: Cantor • Infinity • ZF • Space • Time • Law of Excluded Middle • Set Theory

1. Introduction

An idea of infinity emerged at least as far back as ancient Greek times though infinity had a different character than it has today. Infinity at that time referred to processes that continued on without end. For example, the natural numbers are unbounded. Hence, a natural number oracle that produces each natural number at a constant positive rate is an example of potential infinity. However, the exclusive use of this type of infinity to describe physical reality leads to the paradoxes of Zeno.

Georg Cantor, on the other hand, introduced the notion of actual infinity, which claims a set exists that contains every possible natural number in spite of NNU. Regarding Cantor's system of reasoning, Kunen discloses the following.

> These relative consistency results will be accomplished by completely finitistic means, whereas the consistency of ZF- will remain either an open question or an article of faith, depending upon one's philosophy.[6]

Since the consistency of ZF- is an open question then ZF- may be inconsistent. In fact, no one has ever witnessed an infinite set so there is no evidence produced thus far that would even confirm the existence of infinite sets. Kunen addresses this issue as follows.

> There is some merit in the Finitist's position, since all objects in known physical reality are finite, so that infinite sets may be discarded as figments of the mathematician's

> imagination. Unfortunately, this point of view also discards much of modern mathematics.[6]

Therefore, it is possible that ZF- is inconsistent and there is also no confirmation that infinite sets exist.

There is a claimed justification for the existence of a set containing all natural numbers. So, let's start there. First, a law of excluded middle on infinite sets (LEMI) $\exists x \neg \psi(x) \vee \forall x \psi(x)$ is introduced that is used to establish $\forall x \psi(x)$ for some predicate ψ. Kunen describes this proof technique applied to transfinite induction as follows.

> A "proof by transfinite induction on α" establishes $\forall \alpha \psi(\alpha)$ by showing, for each α, that $(\forall \beta < \alpha\, \psi(\beta)) \rightarrow \psi(\alpha)$. Then $\forall \alpha \psi(\alpha)$ follows, since $\exists \alpha \neg \psi(\alpha)$, the least α such that $\neg \psi(\alpha)$ would lead to a contradiction.[6]

So the proof methodology of LEMI assumes $\exists x \neg \psi(x)$ is true and a contradiction is demonstrated from which one concludes $\neg \exists x \neg \psi(x)$. Then from $\neg \exists x \neg \psi(x)$, $\forall x \psi(x)$ is concluded.

Kunen also illustrates a LEMI argument with his claim that any set which satisfies the axiom of infinity contains all natural numbers. The following is a slight modification of Kunen's argument. The infinity axiom is stated below for clarity.

Axiom of Infinity (INF). $\exists x(0 \in x \wedge \forall y \in x(S(y) \in x))$.

(Kunen) *If a set x satisfies the INF then x contains every natural number.*

LEMI ARGUMENT. Let the predicate $NN(y)$ mean y is a natural number. Assume $\exists n(NN(n) \wedge n \notin x)$. Form the set $X = \{y : NN(y) \wedge y \leq n \wedge y \notin x\}$. X is non-empty by assumption so choose the least $n' \in X$ then $n' \notin x$. Since $0 \in x$ by INF then $n' \neq 0$. So, there is some m' such that $NN(m')$ and $S(m') = n'$. Hence, $m' \in x$ since n' is the least element in X. But, then by INF $S(m') = n' \in x$ which contradicts $n' \notin x$. Thus, $\neg \exists n(NN(n) \wedge n \notin x)$ and then $\forall n(NN(n) \rightarrow n \in x)$. □

Note that the proof above requires a domain of discourse that contains all natural numbers for LEMI's quantified variables. In order to avoid circularity, this domain must be constructible in $ZF - INF$. Using $ZF - INF$ alone, Kunen wrote:

> Many mathematical arguments involve operations with the *set* of natural numbers, but one cannot prove on the basis of the axioms so far presented that there is such a set.[6]

Hence, this domain of discourse is not a set leaving only the possibility from $ZF - INF$ a class of all natural numbers can be constructed. So, one way or another, the successor function must be performed a countable infinite collection of times in order to produce a class containing all natural numbers. Hence, there is an undisclosed requirement for Kunen's argument above that an ICSS can be completed in $ZF - INF$.

The following definition will be used to refer to the first CI set.

Definition 1.1. *ω is the set of natural numbers.*

Therefore given an ICSS can be completed, Kunen's justification for the existence of ω is based on LEMI from which $\forall n(NN(n) \rightarrow n \in x)$ is concluded. LEMI is a fundamental principle of logic in CI. In particular, as shown above, LEMI is used by Kunen in his argument that claims INF describes a collection that contains every possible natural number[6]. It is also used by Kunen to justify transfinite recursion on ON[6]. It is used by Keisler in his proof for transfinite induction[1]. Also, it is used in Cantor's diagonal argument to claim the Cantor diagonal sequence is different from every sequence in a proposed countable infinite range. However, not everyone accepts this proof methodology as indicated by Solomon Feferman.

> "Brouwer argued that LEM for infinite sets is based on an unjustified extension of that principle from finite sets. In his doctoral dissertation of 1907, he had already insisted on the subjective origin of mathematics in human intuition, and on the necessity to restrict questions of truth in mathematics to those statements which can be verified or disproved. Of course, for a finite set A and decidable P we can verify $\exists x(P(x)) \vee \forall x \neg P(x)$ by testing each $x \in A$ in turn to see whether or not $P(x)$ holds. But in general there is no way to carry out such a verification when A is infinite, even for decidable P ."[5]

Therefore, it is Feferman's view that LEMI cannot be proven to be valid because an infinite collection of steps cannot be completed. Since, there is no direct evidence in the universe that demonstrates such an infinite collection exists then Brouwer's opinion regarding LEMI as unjustified is certainly reasonable.

As such, this article will take up the task of exploring the validity of LEMI. Specifically, CI will be cornered with its assertions that it solves Zeno's dichotomy paradox and that all natural numbers are upward constructible by the application of the successor function a countable infinite collection of times.

This examination will look at LEMI from the outside using the meta-theory. In this meta-theory a theoretical computing device is introduced that proves the combination of LEMI and NNU results in a contradiction. This proposed theoretical computing device can execute instructions in any given amount of time. According to Copeland and Shagrir, a somewhat similar machine was first proposed by Weyl.

> In 1927, Hermann Weyl considered a machine (of unspecified architecture) that is capable of completing an infinite sequence of distinct acts of decision within a finite time; say, by supplying the first result after 1/2 minute, the second after another 1/4 minute, the third 1/8 minute later than the second, etc. In this way it would be possible ... to achieve a traversal of all natural numbers and thereby a sure yes-or-no decision regarding any existential question about natural numbers.[4]

Copeland[3] also notes some objections to such devices. The relevant objections center around the fact that the material universe is finite. So, it is argued that infinite requirement computing machines cannot be physically constructed in such a universe. However, this article proposes theoretical computing machines for the purpose of studying CI objects in the meta-theory and as such there is no intention of claiming that they can be physically realized. Therefore, these objections are not applicable to the theoretical devices presented in this article.

The material below will present three such computing devices, an infinity clock, a motion simulator and the infinity machine all of which show that LEMI and NNU are incompatible. This incompatibility in the various devices will refute assertions concerning CI.

The infinity clock will be instrumental in demonstrating that time consists only of a finite collection of ticks. The motion simulator and other machinery prove that any linear interval of space can only be finitely subdivided. Finally, the infinity machine demonstrates that ZF is inconsistent.

2. Finite Time

This section will demonstrate that any interval of time consists only of a finite collection of ticks. For motivation, consider an ordered finite sequence $F=(t_0,t_1,...,t_n)$ and a theoretical clock C that can tick according to the actual progress of time. Now, assume C ticks from t_0 to t_n. With each movement of C according to the sequence F, the clock has $|F-\{t_0,...,t\}|$ elements yet to traverse where t is the current time on C. So, as C ticks to each element of F, one can witness the progress being made in traversing all elements of F since $|F-\{t_0,...,t\}|$ continues to decrease. So, one may conclude that C traverses all of F only because the cardinality of the elements traversed by C increases by one when the cardinality of the elements yet to be traversed by C decreases by one such that $|F-\{t_0,...,t\}|$ eventually becomes zero.

On the other hand, consider an increasing countable infinite convergent sequence (ICICS) in CI, $A=(t_0,t_1,...,t_n,...)$, that is contained in $[a,b]$. Using LEMI, CI claims when C ticks from a to b then C ticks to every element of A. However, in this case with each movement of C according to the elements of A, there are $|A-\{t_0,...,t\}|=\omega$ elements left to encounter. Therefore, for all $t_n \in A$, $|A-\{t_0,...,t_n\}|=\omega$. So, no progress can be made in traversing the entire list A. Hence, the collection of elements traversed by C incrementally increases by one in cardinality while the collection of elements yet to be traversed by C remains constant at an infinite cardinality. So, we do not have the necessary conditions, as in the finite case, whereby we can conclude C will traverse every element of A.

Now, the questions raised above are not sufficient yet to prove CI's assertion that C ticks to every element of A is false. However, one might wonder how CI can claim the ability to complete the task of traversing every element of A given the fact that CI cannot demonstrate any progress whatsoever toward completing that task. This dilemma will be resolved below and in fact, it will be proven CI's assertion that C ticks to every element of A results in a contradiction.

To begin, assume C is a theoretical clock that functions precisely as time does. Our task is to decode its internal operation. It will be proven that C does not operate according to the rules of CI or potential infinity by showing both result in contradictions. Therefore, any arbitrary valid interval of time contains only a finite collection of time intervals.

First, any ICICS will be eliminated as a possible collection that C encounters when time elapses from a to b. So, let $[a,b]$ be an arbitrary valid interval of time and assume $A=(t_0,t_1,...,t_n,...)$ is an arbitrary ICICS such that $t_0=a$, $t_n<b$ and $t_n<t_{n+1}$ for all $n\geq 0$ with $\lim_{n\to\infty} t_n = b$. Set $D=b-a$.

Next, a computing device is designed such that it synchronizes its loop execution with the passage of time on C according to the sequence A. Below $R0$ is an unlimited natural number register and $R1$ is a real number register (i.e. $R1$ may have a countable infinite collection of decimal digits). Consider figure 1 for the algorithm.

```
R0 = 0; R1 = t_0;
do
    R0++;
    R1 = t_{R0};
while (∃z ∈ A(R1 < z))
```

6 done:

Figure 1

The machine and algorithm will be termed the infinity clock. The variable t will refer to the current time on the C clock and it is assumed that C is at rest with the infinity clock. The variable pc refers to the instruction number of the next instruction to be executed by the CPU.

A collection of supporting definitions and lemmas now follow.

Definition 2.1. $P(A,n) \equiv t = t_n$ *with* $t_n \in A$.

Definition 2.2. $P_t(A,a,b,n)$ *is true iff* $P(A,n)$ *evaluated as true sometime between the times* a *and* b.

Definition 2.3. $L(n) \equiv pc = 2 \wedge R0 = n \wedge R1 = t_n$. *If* $L(n)$ *is true then the next execution of instructions 2 through 5 of the loop will be defined as loop iteration* n.

Definition 2.4. $L_t(a,b,n)$ *is true iff* $L(n)$ *evaluated as true sometime between the times* a *and* b.

Definition 2.5. $P \underset{[a,b]}{\rightarrow} L$ *iff if* P *is true at some instant during the time interval* $[a,b]$ *then* L *is also true at that same instant.*

The configuration $P(A,0) \wedge L(0)$ will be defined as starting the infinity clock. If $P(A,n)$ and $L(n)$ are simultaneously true then when the clock C elapses an additional $t_{n+1} - t_n$ seconds the CPU simultaneously completes the execution of loop iteration n. Otherwise, if $L(n)$ and $\neg P(A,n)$ then the CPU takes one second to complete the execution of loop iteration n.

Note CI assumes that time takes on every value of the sequence A when time elapses from a to b. Hence, $\forall n \in \omega P_t(A,a,b,n)$.

The objective below is simple. Assuming $\forall n \in \omega P_t(A,a,b,n)$, LEMI will be used to prove $\neg\exists k \in \omega \neg L_t(a,b,k)$. Therefore, if C ticks to each element of A then a loop iteration completes for each $n \in \omega$. However, it will also be proven that the CPU cannot ever exit the loop. Hence, after any time $s \geq D$, the CPU is still in the loop with more loop iterations to perform, so $\exists k \in \omega \neg L_t(a,b,k)$ is true. Therefore, $\forall n \in \omega P_t(A,a,b,n)$ results in a contradiction.

Using LEMI, the next lemma shows if the clock C ticks to each element of A then the loop remains synchronized with the passage of time on the clock C according to the elements of A.

Lemma 2.6. *Assume the infinity clock is started and C elapses time from a to b. Further assume* $\forall n \in \omega P_t(A,a,b,n)$. *Then* $\forall n \in \omega \left(P(A,n) \underset{[a,b]}{\rightarrow} L(n) \right)$.

PROOF. LEMI is applied. Assume $\exists n \in \omega \neg \left(P(A,n) \underset{[a,b]}{\rightarrow} L(n) \right)$. Form the set $B = \left\{ n \in \omega : \neg \left(P(A,n) \underset{[a,b]}{\rightarrow} L(n) \right) \right\}$ and choose the least $x \in B$ since by assumption B is non-empty. The infinity clock was started which means $P(A,0) \wedge L(0)$ was true then $P(A,0) \underset{[a,b]}{\rightarrow} L(0)$ so $x \neq 0$. Since x is the least element in B then $x - 1 \notin B$ thus $P(A,x-1) \underset{[a,b]}{\rightarrow} L(x-1)$. By assumption, $\forall n \in \omega P_t(A,a,b,n)$. Hence, there was a time between a and b in which $P(A,x-1)$ was true. Thus, $L(x-1)$ was also true at that same instant in time. Therefore $P(A,x-1)$ and $L(x-1)$ were simultaneously true. There was also a time between a and b in which $P(A,x)$ was true. Then an additional $t_x - t_{x-1}$ seconds elapsed on C from the time both $P(A,x-1)$ and $L(x-1)$ were true. Hence, given $t_x - t_{x-1}$ elapsed on C then by the construction of the infinity clock the CPU simultaneously completes the execution of instructions 2-5 of loop iteration $x-1$. After the execution of instruction 5 is completed, the CPU is located at either instruction 2 or instruction 6. But since $R1 = t_x < t_{x+1}$ at instruction 5 then $(\exists z \in A(R1 < z))$. Therefore, the exit condition was not met so, the CPU cannot be at instruction 6. Thus, the CPU is at instruction 2 with $R0 = x \wedge R1 = t_x$. Hence, $L(x)$ was true at the same instant $P(A,x)$ was true, so $P(A,x) \underset{[a,b]}{\rightarrow} L(x)$, which contradicts $x \in B$. Therefore, $\neg \exists n \in \omega \neg \left(P(A,n) \underset{[a,b]}{\rightarrow} L(n) \right)$ so, $\forall n \in \omega \left(P(A,n) \underset{[a,b]}{\rightarrow} L(n) \right)$. □

Using lemma 2.6, it can now be proven $\neg \exists k \in \omega \neg L_t(a,b,k)$. This is important because using LEMI and $\forall n \in \omega P_t(A,a,b,n)$ we can conclude the loop in figure 1 executed once for every possible $k \in \omega$ or more specifically, LEMI claims the infinity clock implements an ICSS. Therefore, we have successfully transferred the CI assertion $\forall n \in \omega P_t(A,a,b,n)$ to the infinity clock in the form of $\neg \exists k \in \omega \neg L_t(a,b,k)$. From that, it can be proven that the assertion $\forall n \in \omega P_t(A,a,b,n)$ results in a contradiction.

Lemma 2.7. *Assume the infinity clock is started, C elapses time from a to b and* $\forall n \in \omega P_t(A,a,b,n)$. *Then* $\neg \exists k \in \omega \neg L_t(a,b,k)$.

PROOF. Apply LEMI. Assume $\exists k \in \omega \neg L_t(a,b,k)$. Then $\neg L_t(a,b,k)$. By assumption, $\forall n \in \omega P_t(A,a,b,n)$. Therefore, there was a time between a and b in which $P(A,k)$ was true. By lemma 2.6 $P(A,k) \underset{[a,b]}{\rightarrow} L(k)$, therefore $L(k)$ was also true at that same instant. So, $L(k)$ was true sometime between a and b thus, $L_t(a,b,k)$, which is a contradiction. Hence by LEMI, $\neg \exists k \in \omega \neg L_t(a,b,k)$. □

The following lemma shows it is impossible for the clock C to elapse $s \geq D$ seconds since the start of the infinity clock and for the CPU to exit the loop. Since there is no dispute that C can elapse s seconds, then it is impossible for the CPU to exit the loop when s seconds elapsed. Given s is arbitrary then there is no elapsed time on C at which the CPU exits the loop.

Lemma 2.8. *Assume the infinity clock is started and* $s \geq D$ *seconds elapses on* C *since the start. Then the CPU did not exit the loop.*

PROOF. Set a breakpoint at instruction 6. Start the infinity clock and assume $s \geq D$ seconds elapsed on C. Either the CPU is in the loop or it is not. Assume the CPU exited the loop and is located at the instruction 6 breakpoint. By the definition of the algorithm, instructions 3-5 are executed just before the loop exits. Then, register $R0$ contains some natural number say k since only the successor function is applied to $R0$ and the successor function only produces natural numbers. But then, $R1 = t_k < t_{k+1}$ so the exit condition was not met and thus the CPU could not be at instruction 6 contradicting the assumption that it is. Therefore, the CPU did not exit the loop. □

The next lemma demonstrates NNU by showing if D seconds elapses since the start of the infinity clock then $\exists n \in \omega \neg L_t(a,b,n)$. Hence, regardless of the conclusions of LEMI in lemma 2.7, the loop cannot execute once for every possible natural number because there is no end to the natural numbers. Thus, there is no end to the loop iterations. Note that loop timing has no impact whatsoever on this conclusion.

Lemma 2.9. *Assume the infinity clock is started and C elapses time to b. Then* $\exists n \in \omega \neg L_t(a,b,n)$*.*

PROOF. Set a breakpoint at instruction 6 and one at instruction 2 that will only trigger at the first instant the CPU is located at instruction 2 and at least D seconds elapses on C after the infinity clock is started. Then start the infinity clock and assume C elapses time to b. Wait until a breakpoint is triggered. Then, the CPU is either in the loop or it is not. Thus, the CPU hit the breakpoint at instruction 2 or it hit the breakpoint at instruction 6. By lemma 2.8 the CPU did not exit the loop. Hence, the CPU must at the instruction 2 breakpoint.

At instruction 2, the register $R0$ contains some natural number k since only the successor function is applied to $R0$. But then by the last execution of instruction 4, $R1 = t_k$. Therefore, the CPU is located at instruction 2 with $R0 = k \wedge R1 = t_k$. Thus, $L(k)$ is true and by the definition of the infinity clock, loop iteration $k-1$ just completed execution. Loop iteration k takes either $t_{k+1} - t_k$ seconds or 1 second to complete and since at least D seconds had already elapsed when $L(k)$ is true then loop iteration k did not complete execution during the D seconds. Therefore, the set of completed loop iterations is $\{0,\ldots,k-1\}$, which of course is finite. Further, since loop iteration k did not complete then $L(k+1)$ did not evaluate as true between the times a and b. Therefore, $\neg L_t(a,b,k+1)$ and so $\exists n \in \omega \neg L_t(a,b,n)$. □

Notice in the proof of lemma 2.9 that the total loop iteration time should be $t_k - t_0 < D$ when $L(k)$ is true. However as shown in lemma 2.8, the loop cannot ever exit regardless of the actual elapsed time on C. Hence, the elapsed loop execution time must always remain below D seconds even though the elapsed time on C does not. So, it is inevitable that the infinity clock eventually becomes unsynchronised with the actual progress of time on the clock C.

Interestingly, lemma 2.8 also shows the same conclusion as Zeno's progressive dichotomy paradox. More specifically, given the infinity clock loop timing is synchronized according to the elements of A and the CPU encountered each element of A then there is no elapsed time $s \geq D$ in the universe where the infinity clock would reach the time b. Alternatively, the current time on the infinity clock, which is in the $R1$ register, would remain below b forever. So , CI did not solve Zeno's progressive dichotomy paradox.

Lemmas 2.7 and 2.9 also resolve the dilemma presented above. Lemma 2.7 used LEMI to transfer the assertion made by CI that time takes on every value of an ICICS $A \subset [a,b]$, when time elapses from a to b, to the infinity clock such that we could assert that a loop iteration occurs once for every possible natural number. In particular, LEMI asserts that the infinity clock completes the task of traversing every element of A.

Lemma 2.9 on the other hand, shows the infinity clock implements the truth of NNU by using the successor function applied to the $R0$ register. As such regardless of loop timing, given any value of $R0$, there will always be a successor to $R0$, hence another $t_{R0+1} \in A$ and another corresponding loop iteration. Thus, it is impossible for the CPU to complete a loop iteration for every possible natural number since the successor function always produces another natural number. Therefore by NNU, the infinity clock cannot ever complete the task of traversing every element of A.

By combining the truth of LEMI and NNU with the infinity clock, it is now proven that C does not take on every value of the sequence A between times a and b because that assertion results in a contradiction between LEMI and NNU.

Theorem 2.10. *Assume the infinity clock is started and C elapses time to b. Then there are no ICICSs A defined as above such that $\forall n \in \omega P_t(A,a,b,n)$.*

PROOF. Assume to the contrary that there is such an ICICS A. By lemma 2.7, $\neg\exists n \in \omega \neg L_t(a,b,n)$. By lemma 2.9, $\exists n \in \omega \neg L_t(a,b,n)$. Therefore, $\neg\exists n \in \omega \neg L_t(a,b,n)$ and $\exists n \in \omega \neg L_t(a,b,n)$, which is a contradiction. Since there is no dispute that C can elapse time from a to b then the assumption that there is a sequence A with $\forall n \in \omega P_t(A,a,b,n)$ is the cause of the contradiction. Hence, there are no ICICSs A such that $\forall n \in \omega P_t(A,a,b,n)$. □

In short, LEMI in lemma 2.7 claims the infinity clock completed an infinite collection of successive steps/loop iterations when C elapses time from a to b. However, NNU as expressed in lemma 2.9 shows the infinity clock can only complete a finite collection of loop iterations when C elapses time from a to b. Therefore, the claim that an ICSS can be completed results in a contradiction with NNU.

It is now shown C does not take on every value of an arbitrary decreasing countable infinite convergent sequence (DCICS) either. Assume Define $a < b$ and define $A' = (t_0, t_1, ..., t_n, ...)$ where $t_0 = b$, $t_n > a$, $t_{n+1} < t_n$ for all $n \geq 0$ with $\lim_{n\to\infty} t_n = a$. Now, A' will be eliminated as a possibility for the operation of C.

With sequences like A', CI claims C does not tick to a first element of A', which is a correct conclusion under their assumptions. More specifically, if C ticked first to some $t_n \in A'$ then it would have failed to tick to any $t_k \in A'$ with $k > n$. However, by continuing the argument, if C did not tick to a first $t_n \in A'$ then it did not tick to a second element of A' and so on through the elements of A'. In other words, a recursive process is not viable unless it has a valid starting point. So, if C does tick to a first element, then it does not tick through all $t_n \in A'$. If C does not tick to a first element, then it does not tick to any $t_n \in A'$ which is exactly the conclusion provided by Zeno's regressive dichotomy paradox. Consequently, C does not tick to every element of A' contradicting the CI assertion that it does.

The next theorem formalizes the argument above.

Theorem 2.11. *Assume C elapses time from a to b. Then there are no DCICSs A' as defined above such that $\forall n \in \omega P_t(A',a,b,n)$.*

PROOF. Assume to the contrary there is a DCICS A' such that $\forall n \in \omega P_t(A',a,b,n)$. Introduce the variable r and initialize it to a. Define $Q(t) \equiv t \in A' \wedge r = a$. Now assume C is at a and elapses to b. Whenever the clock changes time between a and b, test the current time t with $Q(t)$. If $Q(t)$ is true then set r to t. Note that if r is set to some element of A', it cannot be changed again because the condition $r = a$ will be false thereafter. Also note, after C reaches b, either $r = a$ or $r = t_k$ for some $t_k \in A'$.

So, assume $\exists k \in \omega(r = t_k)$ or that C does tick to a first element of A'. Then $\neg\exists x \in \omega(x > k \wedge P_t(A',a,b,x))$. Otherwise, assume $\exists x \in \omega(x > k \wedge P_t(A',a,b,x))$. Then $r = a$ or $r \neq a$ when $t = t_x$. If $r \neq a$ when $t = t_x$ then $r \neq a$ when $t = t_k$ since r cannot be changed back to a once it is set to some element of A'. But, then $Q(t)$ would be false when $t = t_k$ and hence $r \neq t_k$, a contradiction. Also, if $r = a$ when $t = t_x$ then $Q(t)$ was true when $t = t_x$. So r was set to t_x at time $t = t_x$. Hence, $r = t_x$ at $t = t_k$ making $Q(t)$ false when $t = t_k$ which contradicts $r = t_k$. Therefore, $\neg\exists x \in \omega(x > k \wedge P_t(A',a,b,x))$, which contradicts $\forall n \in \omega P_t(A',a,b,n)$, so $\neg\exists k \in \omega(r = t_k)$.

Next, assume C does not tick to a first element of A' so $\neg\exists k \in \omega(r = t_k)$. Then $r \notin A'$ hence, $r = a$ was true from a to b. Since r was not changed then $\neg\exists t(a \leq t \leq b \wedge Q(t))$. Given

$\neg\exists t(a \le t \le b \wedge Q(t))$ with $Q(t) \equiv t \in A' \wedge r = a$ and $r = a$ being true between a and b then $\neg\exists t(a \le t \le b \wedge t \in A')$. Thus, $\neg\exists n \in \omega P_t(A', a, b, n)$ which contradicts $\forall n \in \omega P_t(A', a, b, n)$.

Therefore in all cases, a contradiction results. Since there is no dispute that C can elapse time from a to b then the assertion $\forall n \in \omega P_t(A', a, b, n)$ is the cause of the contradiction.

Since A' is arbitrary and $\forall n \in \omega P_t(A', a, b, n)$ results in a contradiction then there are no DCICSs A' such that $\forall n \in \omega P_t(A', a, b, n)$. □

The next lemma is needed to prove that C does not operate according to CI and is argued in the context of CI.

Lemma 2.12. *If $[c,d]$ is a real number interval then $[c,d]$ contains an ICICS.*

PROOF. Set $a_n = d - (d-c)(1/2^n)$. Define $A = (a_0, a_1, ..., a_n, ...)$. Obviously, A is countable infinite. Since $1/2^n \to 0$ as $n \to \infty$ then the sequence A has a limit of d and is therefore convergent. Now assume $\exists n(a_n \ge a_{n+1})$. Then $1/2^{n+1} \ge 1/2^n$, which is a contradiction thus $\forall n(a_n < a_{n+1})$, so A is increasing. Hence, A is an ICICS. Finally, to show A is contained in $[c,d]$, assume $\exists n(a_n < c \vee a_n \ge d)$. If $a_n < c$ then $d - (d-c)(1/2^n) < c$ and so $1/2^n > 1$, which is a contradiction. If $a_n \ge d$ then $d - (d-c)(1/2^n) \ge d$ and so $1/2^n \le 0$, which is a contradiction. Thus, $\forall n(c \le a_n < d)$. Hence, A is an ICICS that is contained in $[c,d]$. □

Next, CI is eliminated as a valid description of time.

Theorem 2.13. *The clock C does not operate according to CI.*

PROOF. Let $[c,d]$ be an arbitrary valid interval of time such that C ticks from c to d. Assume $[c,d]$ is a real number interval. By lemma 2.12, $[c,d]$ contains an ICICS A. Then $\forall n \in \omega P_t(A, \min(A), \sup(A), n)$, which contradicts theorem 2.10. So, $[c,d]$ is not a real number interval. Assume $[c,d]$ is countable infinite. Also, since time only proceeds forward, then $[c,d]$ is a total linear order. Therefore to be infinite, $[c,d]$ contains either an ICICS, A or a DCICS, A'. If $A \subseteq [c,d]$ then $\forall n \in \omega P_t(A, \min(A), \sup(A), n)$, which contradicts theorem 2.10. If $A' \subseteq [c,d]$ then $\forall n \in \omega P_t(A', \inf(A'), \max(A'), n)$, which contradicts theorem 2.11. So, any claim that $[c,d]$ contains a CI collection results in a contradiction. Therefore, $[c,d]$ is not infinite Since $[c,d]$ is arbitrary then there are no intervals of time that contain an infinite collection of points. Hence, the clock C does not operate according to CI. □.

CI has now been completely eliminated as a possible description of the operation of C. Next, potential infinity is eliminated. Assume the sequence $A = (t_0, t_1, ..., t_n, ...)$ is a valid potential infinite collection of divisions of the time interval a through b such that $t_0 = a$ with $t_n < b$ and $t_n < t_{n+1}$ for any fixed $n \ge 0$.

Theorem 2.14. *Assume C starts at a and elapses to b. Then no increasing potential infinite sequence A as defined above can be produced such that C ticks to each element in A and also ticks to b.*

PROOF. Assume such a sequence A is produced and C ticks to each element of A and then to b. Introduce the variable r and initialize it to -1 before the clock time is a. Define $Q(t) \equiv t \in A$. Whenever the clock changes time between a and b, test the current time t with $Q(t)$. If $Q(t)$ is true, set r to t. Hence, r can only be -1 or some $t_k \in A$ when the clock reaches the time b.

Now assume C elapsed from a to b. After C reaches b, if $r=-1$ then there was no time on the clock in which $t \in A$ was true which contradicts the assumption that C ticks to each element of A. If $r=t_k$ for some $t_k \in A$ then C did not tick to any t_x with $x>k$. Otherwise, if C did tick to such a t_x then $Q(t)$ would have evaluated as true after $t=t_k$. Therefore, $r \neq t_k$, which is a contradiction. So, C did not tick to any t_x with $x>k$ which contradicts the assumption that C ticks to each element of A. So, in either case, a contradiction results by assuming C ticks to each element of A. Since there is no dispute that C can elapse time from a to b then the assumption that C ticks to each element of A is the cause of the contradiction. Hence, no such sequence A can be produced such that C ticks to each element in A and also ticks to b. □

Any given decreasing potential infinite sequence is now eliminated where the divisions begin at b and continue toward a. Assume the sequence $A'=(t_0,t_1,...,t_n,...)$ is a valid potential infinite collection of divisions of the time interval a through b such that $t_0=b$ with $t_n>a$ and $t_{n+1}<t_n$ for any fixed $n \geq 0$.

Theorem 2.15. *Assume C starts at a and elapses to b. Then no decreasing potential infinite sequence A' as defined above can be produced such that C ticks to each element in A' and also ticks to b.*

PROOF. Assume to the contrary that a decreasing potential infinite sequence A' as defined above is produced such that C ticks to each element in A' and also ticks to b. Introduce the variable r and predicate Q just as in theorem 2.11. Once C reaches b, either $r=a$ or $r=t_k$ for some $t_k \in A'$. Assume $r=t_k$. Then it was not the case that $Q(t)$ evaluated as true for any given $t_x \in A'$ with $x>k$. Otherwise, assume so, then $r=a$ or $r \neq a$ when $t=t_x$. If $r \neq a$ when $t=t_x$ then $r \neq a$ when $t=t_k$ and hence $r \neq t_k$, a contradiction. If $r=a$ when $t=t_x$ then $r=t_x$ at $t=t_k$ making $Q(t)$ false when $t=t_k$ which contradicts $r=t_k$. So if $r=t_k$ then C did not tick to t_x which contradicts the assumption that C ticks to each element of A'. Thus, $r \neq t_k$ hence, $r=a$. Since r was not changed then C did not tick to any element of A', which is a contradiction. Therefore, no such A' can be produced. □

Theorem 2.16. *Potential infinity does not describe the operation of the clock C.*

PROOF. Apply theorems 2.14 and 2.15. Then any type of produced potential infinite collection of divisions of an arbitrary valid time interval is eliminated as possible explanations for the operation of C. □

All the types of infinity have been eliminated as correct descriptions for the operation of C. Keeping in mind that the clock C functions as time does, the next theorem shows any given time interval consists only of a finite collection of divisions. Hence, time is atomic.

Theorem 2.17. *Let a and b be valid time values. Then the time interval $[a,b]$ is only finitely divisible. Hence, time is atomic.*

PROOF. By theorem 2.13, $[a,b]$ is not an infinite collection. Since time only proceeds forward then assume $[a,b]$ is a potential infinite sequence. However, that assumption contradicts theorem 2.16. Consequently by elimination, $[a,b]$ contains only a finite collection of divisions of the time. Therefore, time is atomic. □

There is no justification in assuming the finite set of divisions contained in $[a,b]$ are equal in length. Further, for any valid interval of time, there is a point at which any further division results in something that cannot be described as having the property of time. Otherwise, time is infinitely divisible and that was proven to result in a contradiction.

The next section will prove space is also atomic.

3. Finite Space

To determine the nature of space, first CI will be eliminated as a valid description of a linear interval of space. To begin, let $[a,b]$ be an arbitrary valid linear interval of space and let $A=(a_0, a_1,...,a_n,...)$ be an ICICS such that $a_0 = a$, $a_n < b$, and $a_{n+1} > a_n$ for any $n \geq 0$ with $\lim_{n\to\infty} a_n = b$. Now, if CI correctly represents space, then some point P that moves from a to b will encounter every point of the sequence A. As was proven with the infinity clock, this Cantor assumption also results in a contradiction.

Next, a computing machine is designed such that it synchronizes its execution with the motion of P. Below $R0$ is an unlimited natural number register and $R1$ is a Cantor real number register. Consider figure 2 for the algorithm.

```
1    R0 = 0; R1 = a_0;
2    do
3       R0++;
4       R1 = a_R0;
5    while (∃z ∈ A(R1 < z))
6    done:
```

Figure 2

The machine and the program in figure 2 will be called the motion simulator. Below, it is assumed that P moves at a constant velocity in its motion from a to b and continues on from b with that same motion.

We are now in a position to prove that P does not encounter all the points of A during its movement from a to b. Some definitions and lemmas follow in order to achieve this task. The variable p will refer to the current position of P.

Definition 3.1. $P(A,n) \equiv p = a_n$ *with* $a_n \in A$.

Definition 3.2. $P_t(A,a,b,n)$ *is true iff* $P(A,n)$ *evaluated as true sometime during the motion of* P *from* a *to* b.

Definition 3.3. $L(n) \equiv pc = 2 \wedge R0 = n \wedge R1 = a_n$. *If* $L(n)$ *is true then the following execution of instructions 2 through 5 of the loop will be defined as loop iteration* n.

Definition 3.4. $L_t(a,b,n)$ *is true iff* $L(n)$ *evaluated as true sometime during the motion of* P *from* a *to* b.

Definition 3.5. $Q \underset{[a,b]}{\rightarrow} L$ *iff if* Q *is true sometime during the motion of* P *on the interval* $[a,b]$ *then* L *is true simultaneously with* Q *being true.*

The configuration $P(A,0) \wedge L(0)$ will be defined as starting the motion simulator. If $P(A,n)$ and $L(n)$ are simultaneously true then when the point P reaches a_{n+1} the CPU simultaneously completes the execution of loop iteration n. Otherwise, if $L(n)$ and $\neg P(A,n)$ then the CPU takes one second to complete the execution of loop iteration n.

All the machinery that follows is mostly the same as section 2 so we will proceed at a rapid pace. Note that CI claims $\forall n \in \omega P_t(A,a,b,n)$.

Lemma 3.6. *Assume the motion simulator is started, P moves from a to b and* $\forall n \in \omega P_t(A,a,b,n)$. *Then* $\forall n \in \omega\left(P(A,n) \underset{[a,b]}{\rightarrow} L(n)\right)$.

PROOF. Apply LEMI. Assume $\exists n \in \omega \neg\left(P(A,n) \underset{[a,b]}{\rightarrow} L(n)\right)$. Form the set $B = \left\{n \in \omega : \neg\left(P(A,n) \underset{[a,b]}{\rightarrow} L(n)\right)\right\}$ and choose the least $x \in B$. By assumption the motion simulator was started thus, $P(A,0) \underset{[a,b]}{\rightarrow} L(0)$ so $x \neq 0$. Since x is the least element in B then $x-1 \notin B$ thus $P(A,x-1) \underset{[a,b]}{\rightarrow} L(x-1)$. By assumption, $\forall n \in \omega P_t(A,a,b,n)$ hence, $P_t(A,a,b,x-1)$ and $P_t(A,a,b,x)$. So, there was a time during the motion of P from a to b in which $P(A,x-1)$ was true. Therefore $L(x-1)$ was simultaneously true. Since $P_t(A,a,b,x)$, assume P moved from a_{x-1} to a_x. By assumption of the motion simulator, given that $P(A,x-1)$ and $L(x-1)$ were both simultaneously true and P moved from a_{x-1} to a_x then when P arrived at a_x the CPU simultaneously completed the execution of loop iteration $x-1$.

At instruction 5 of loop iteration $x-1$, $R1 = a_x < a_{x+1}$, hence $(\exists z \in A(R1 < z))$. So, the exit condition was not met and therefore, the CPU was located at instruction 2. Thus, the CPU was at instruction 2 with $R0 = x \wedge R1 = a_x$. So, $L(x)$ was true at the same instant $P(A,x)$ was true, hence $P(A,x) \underset{[a,b]}{\rightarrow} L(x)$, which contradicts $x \in B$. Thus by LEMI, $\neg\exists n \in \omega \neg\left(P(A,n) \underset{[a,b]}{\rightarrow} L(n)\right)$ and then $\forall n \in \omega\left(P(A,n) \underset{[a,b]}{\rightarrow} L(n)\right)$. □

Lemma 3.7. *Assume the motion simulator is started, P moves from a to b and* $\forall n \in \omega P_t(A,a,b,n)$. *Then* $\neg\exists k \in \omega \neg L_t(a,b,k)$.

PROOF. Apply LEMI. Assume $\exists k \in \omega \neg L_t(a,b,k)$ then $\neg L_t(a,b,k)$. By assumption, $\forall n \in \omega P_t(A,a,b,n)$. Hence, during the motion of P from a to b, $P(A,k)$ was true. By lemma 3.6, $L(k)$ was also true at that same instant. So, $L(k)$ was true sometime during the motion of P from a to b, hence $L_t(a,b,k)$, which is a contradiction. By LEMI, $\neg\exists k \in \omega \neg L_t(a,b,k)$. □

Lemma 3.8. *Assume the motion simulator is started and P moves from a to b. Then* $\exists n \in \omega \neg L_t(a,b,n)$.

PROOF. Set a breakpoint at instruction 6 and one at instruction 2 that will only trigger at the first instant P is located at b or beyond. Then start the motion simulator and assume P has moved at least to b. Either the CPU will hit the breakpoint at instruction 2 or it will hit the breakpoint at instruction 6. Assume the exit condition is met and the CPU is located at the instruction 6 breakpoint. By the definition of the algorithm, instructions 3-5 are executed just before the loop exits. Since only the successor function is applied to $R0$ then the register $R0$ contains some natural number say k. But, then $R1 = a_k < a_{k+1}$ so the exit condition was not met and thus the CPU could not be at instruction 6 contradicting the assumption that it is. Therefore, the CPU must be at the instruction 2 breakpoint and $R0$ must contain some natural number k. Hence, $R1 = a_k$ and then $L(k)$ and $\neg P(A,k)$. Hence, loop iteration k will take one second to complete. At that time P will be past b. So, the CPU did not complete the execution of loop iteration k during the motion of P from a to b. Hence, $\neg L_t(a,b,k+1)$ and so $\exists n \in \omega \neg L_t(a,b,n)$. □

It is now proven that the point P does not encounter every value of any ICICS A during its motion from a to b.

Lemma 3.9. Assume P *moves from* a *to* b*.* Then there are no ICICSs A such that $\forall n \in \omega P_t(A,a,b,n)$.

PROOF. Assume the motion simulator is started and P moves to b. Further assume to the contrary that there is an ICICS A such that $\forall n \in \omega P_t(A,a,b,n)$. Then by lemma 3.7, $\neg\exists n \in \omega \neg L_t(a,b,n)$. By lemma 3.8, $\exists n \in \omega \neg L_t(a,b,n)$. Therefore, $\neg\exists n \in \omega \neg L_t(a,b,n)$ and $\exists n \in \omega \neg L_t(a,b,n)$, which is a contradiction. Since there is no dispute that P can move from a to b then the assumption that there is a sequence A with $\forall n \in \omega P_t(A,a,b,n)$ is the cause of the contradiction. Hence, there are no ICICSs A such that $\forall n \in \omega P_t(A,a,b,n)$. □

Next, it is shown that P does not take on every value of an arbitrary DCICS either. So, a DCICS will be defined that begins at b and converges to a. Define $A' = (a_0, a_1, \ldots, a_n, \ldots)$ such that $a_0 = b$, $a_n > a$ and $a_{n+1} < a_n$ for all $n \geq 0$ with $\lim_{n\to\infty} a_n = a$. Now, A' will be eliminated as a possibility for describing the motion of P from a to b.

Lemma 3.10. *Assume* P *moves from* a *to* b*. Then there are no decreasing countable infinite convergent sequences* A' *such that* $\forall n \in \omega P_t(A',a,b,n)$*.*

PROOF. Assume to the contrary there is a DCICS A' with $\forall n \in \omega P_t(A',a,b,n)$. Introduce the variable r and initialize it to a. Define $Q(p) \equiv p \in A' \wedge r = a$. Now assume P is at a and moves to b. Whenever P changes position between a and b, test the current position p with $Q(p)$. If $Q(p)$ is true then set r to p. Once P moves to b, either $r = a$ or $r = a_k$ for some $a_k \in A'$.

Assume $r = a$. Then $Q(p)$ did not evaluate as true during the motion of P from a to b. Hence, $\neg\exists n \in \omega P_t(A,a,b,n)$ which contradicts $\forall n \in \omega P_t(A',a,b,n)$. Now assume $r = a_k$. Then during the motion of P from a to b, $Q(p)$ first evaluated true at $p = a_k$. Therefore, $Q(p)$ did not evaluate as true for any $a_n \in A'$ such that $n > k$ which contradicts $\forall n \in \omega P_t(A',a,b,n)$.

Hence, both $r = a$ and $r = a_k$ result in a contradiction. Since there is no dispute that P can move from a to b then the assertion $\forall n \in \omega P_t(A',a,b,n)$ is the cause of the contradiction. Since A' is arbitrary then there are no DCICSs with $\forall n \in \omega P_t(A',a,b,n)$. □

Theorem 3.11. *P does not move according to the rules CI.*

PROOF. Let $[c,d]$ be an arbitrary valid linear interval of space such that P moves with a constant velocity from c to d. Assume $[c,d]$ is a real number interval. By lemma 2.12, $[c,d]$ contains an ICICS A. Hence, $\forall n \in \omega P_t(A, \min(A), \sup(A), n)$, which contradicts lemma 3.9. So, $[c,d]$ is not a real number interval. Now assume $[c,d]$ is countable infinite. Since P moves at a constant velocity, then $[c,d]$ is a total linear order. Therefore to be infinite, $[c,d]$ contains either an ICICS, A or a DCICS, A'. If $A \subseteq [c,d]$ then $\forall n \in \omega P_t(A, \min(A), \sup(A), n)$, which contradicts lemma 3.9. If $A' \subseteq [c,d]$ then $\forall n \in \omega P_t(A', \inf(A'), \max(A'), n)$, which contradicts lemma 3.10. So, any claim that $[c,d]$ contains a CI collection results in a contradiction. Therefore, $[c,d]$ is not infinite. Since $[c,d]$ is arbitrary then there are no intervals of space that contain an infinite collection of points. Hence, P does not move according to CI. □

Next, potential infinity is eliminated. Assume $A = (a_0, a_1, ..., a_n, ...)$ is a valid potential infinite collection of divisions of the linear space interval a through b such that $a_0 = a$ with $a_n < b$ and $a_n < a_{n+1}$ for any $n \geq 0$.

Lemma 3.12. *Assume P moves from a to b. Then no increasing potential infinite sequence A as defined above can be produced such that P encounters each element of A and also moves to b.*

PROOF. Assume such a sequence A is produced such that P encounters each element of A and then moves to b. Introduce the variable r and initialize it to -1 before P is at a. Define $Q(p) \equiv p \in A$. Whenever P changes position between a and b, test the current position p with $Q(p)$. If $Q(p)$ is true, set r to p. Hence, r can only be -1 or some $a_k \in A$ when P reaches b.

Now assume P moves from a to b. After P reaches b, if $r = -1$ then there is no $x \in A$ such that $p = x$ during the motion of P from a to b which contradicts the assumption that P encounters each element of A. If $r = a_k$ for some $a_k \in A$ then P did not encounter any given a_x with $x > n$ which also contradicts the assumption that P encounters each element of A. So, in either case, a contradiction results by assuming P encounters each element of A. Since there is no dispute that P can move from a to b then the assumption that P encounters each element of A is the cause of the contradiction. Hence, no such sequence A can be produced such that P encounters each element in A and also moves to b. □

We now eliminate all potential infinite sequences where the divisions begin at b and continue toward a. Assume $A' = (a_0, a_1, ..., a_n, ...)$ is a valid potential infinite collection of divisions of the space interval a through b such that $a_0 = b$ with $a_n > a$ and $a_{n+1} < a_n$ for any $n \geq 0$.

Lemma 3.13. *Assume P moves from a to b. Then no decreasing potential infinite sequence A' as defined above can be produced such that P encounters each element of A' and also moves to b.*

PROOF. Assume to the contrary that a decreasing potential infinite sequence A' as defined above is produced such that P encounters each element in A' and also moves to b. Introduce the variable r and predicate Q just as in lemma 3.10. Once P reaches b, either $r = a$ or $r = a_k$ for some $a_k \in A'$. Assume $r = a_k$. Then it was not the case that $Q(p)$ evaluated as true for any given $a_x \in A'$ with $x > k$. Otherwise, assume so then $r = a$ or $r \neq a$ when $p = a_x$. If $r \neq a$ when $p = a_x$ then $r \neq a$ when $p = a_k$ and hence $r \neq a_k$, a contradiction. If $r = a$ when $p = a_x$ then $r = a_x$ at $p = a_k$ making $Q(p)$ false when $p = a_k$ which contradicts $r = a_k$. So if $r = a_k$ then P did not encounter a_x which contradicts the assumption that P encounters each element of A'. Thus, $r \neq a_k$ hence, $r = a$. Since r was not changed then P did not encounter any element of A', which is a contradiction. Therefore, no such A' can be produced. □

Theorem 3.14. *Potential infinity does not describe the motion of P.*

PROOF. Apply lemmas 3.12 and 3.13. Then any type of potential infinite collection of divisions of an arbitrary valid linear interval of space is eliminated as a possible explanation for the motion of P. □

We conclude this last section by proving that any valid linear interval of space is atomic.

Theorem 3.15. *Let a and b be valid space positions. Then the linear space interval $[a,b]$ is only finitely divisible. Hence, space is atomic.*

PROOF. By theorem 3.11, $[a,b]$ does not contain a CI collection of divisions. Theorem 3.14 eliminated the possibility that $[a,b]$ contains a potential infinite collection of divisions. Therefore, any arbitrary valid interval of space $[a,b]$ contains only a finite collection of divisions. Hence, space is atomic. □

4. The Inconsistency of Zermelo-Fraenkel Set Theory

The infinity machine (IM) will be introduced below as a tool to attempt to upward construct a class of all finite ordinals. Starting with the null set, the IM will iteratively apply the successor function such that there is zero seconds of execution time between the start of each successor operation.

These finite ordinals will be accumulated into a set y as they are constructed so that it can be determined whether or not an infinite collection is actually generated. Then it is assumed that the IM executes for some $t > 0$ seconds. By applying LEMI, it will be shown y satisfies INF. Hence y is infinite under ZF. On the other hand, it will be also shown that the IM implements NNU. Based on NNU, it will be proven y is finite. Therefore, at t seconds, y is both infinite and finite which is a contradiction. Since a contradiction can be deduced from ZF then ZF is inconsistent.

Consider figure 3 for the implementation of the IM. The register $R0$ is a finite ordinal set register and y is countable infinite set register. The function $AddE(y,n)$ adds the finite ordinal n as an element to the set y. The function S is the successor function. All program instructions execute in zero seconds.

Below, the natural numbers $0,1,2,...$ will be interchangeable with their associated finite ordinals $\varnothing, \{\varnothing\}, \{\varnothing, \{\varnothing\}\}, ...$. All variables below range over a DDNN of CI.

```
1   R0 = ∅;  y = ∅;
2   do
3       AddE(y, R0);
4       R0 = S(R0);
5   while (∃n(NN(n) ∧ n ∉ y));
6   done:
```

Figure 3

The following definitions provide a foundation for the machinery needed to demonstrate that ZF is inconsistent.

Definition 4.1. *Loop iteration n is defined as instructions 2 through 5 with $R0 = n$ at instruction 2.*

Definition 4.2. *The constant $t_c > 0$ is some arbitrary valid time period in seconds that elapses on the C clock of section 2.*

Definition 4.3. *The constant $t_c' > t_c$ is some valid time period in seconds that elapses on the C clock of section 2.*

Definition 4.4. *A CPU breakpoint at instruction* n *stops the CPU from any further execution such that* $pc = n$*. After the CPU breakpoint is hit, the program's registers may be examined.*

Definition 4.5. *The variable* etl *is the total loop execution time. The convention* $etl_{(n)}$ *will be used to identify the value of* etl *at instruction 2 or instruction 6 when* $R0 = n$*.*

Definition 4.6.
$L(n) \equiv (pc = 2) \wedge (R0 = n) \wedge (etl_{(n)} = 0) \wedge (n = 0 \rightarrow y = \varnothing) \wedge (n > 0 \rightarrow y = \{0,...,n-1\})$.

Definition 4.7. *Starting the IM means the clock* C *of section 2 is set to zero and also* $L(0)$ *is simultaneously true.*

Definition 4.8. $L_t(n,t)$ *is true iff* $L(n)$ *evaluated as true sometime between the start of the IM and the elapsed time of* t*.*

Definition 4.9. *A CPU interrupt,* $CPUI(t)$*, stops the CPU from any further execution at the first instruction in the loop when exactly* t *seconds has elapsed on the* C *clock of section 2 since the start of the IM. After the CPU interrupt occurs, the CPU's location, registers may be examined.*

LEMI is now applied to argue $\forall n(L_t(n,t_c))$.

Lemma 4.10. *Start the IM and assume* t_c *seconds elapses on* C *after the start. Then* $\forall n(L_t(n,t_c))$*.*

PROOF. The proof proceeds by LEMI. Assume $\exists n(\neg L_t(n,t_c))$. Form the set $X = \{k : k \le n \wedge \neg L_t(k,t_c)\}$ and choose the least $x \in X$. Then $\neg L_t(x,t_c)$. By definition, starting the IM means $L(0)$ is true so $L_t(0,t_c)$ is true by hypothesis, thus $x > 0$. Hence, $L_t(x-1,t_c)$ is true since x is the least element in X. Therefore, it is justified to assume the condition that $L(x-1)$ is true. Then at instruction 2, $(R0 = x-1) \wedge (etl_{(x-1)} = 0)$. Also, $(x-1 = 0 \rightarrow y = \varnothing) \wedge (x-1 > 0 \rightarrow y = \{0,...,x-2\})$ is true. At instruction 4, $y = \{0,...,x-1\}$ since $x-1$ was added as an element to y at instruction 3. Finally, at instruction 5 , $R0 = x$. At this point in the construction, $R0 = x \notin y = \{0,...,x-1\}$ so the exit condition is not met. Thus, the CPU proceeds back to instruction 2 with $R0 = x$ and $y = \{0,...,x-1\}$. Now, at the condition where $L(x-1)$, was true, $etl_{(x-1)} = 0$ was also true. All instructions take zero seconds to execute so $etl_{(x)} = etl_{(x-1)} + 0 = 0 < t_c$. Hence, $L(x)$ is true at that point of the execution. Since $L(x)$ was true during the t_c seconds then $L_t(x,t_c)$, which contradicts $\neg L_t(x,t_c)$. So by LEMI, $\neg\exists n(\neg L_t(n,t_c))$ and then $\forall n(L_t(n,t_c))$. □

The next lemma shows that the set y satisfies INF. So, the "rules" for constructing all the finite ordinals from ZF have been followed.

Lemma 4.11. *Start the IM and assume* t_c *seconds elapses on* C *after the start. Then* $0 \in y \wedge \forall n \in y(S(n) \in y)$*. Hence, the set* y *satisfies INF.*

PROOF. By lemma 4.10, $L_t(1,t_c)$ is true. At that point in the execution, $y = \{0\}$. Thus, $0 \in y$. Now assume $\exists n \in y(S(n) \notin y)$. This assumption is justified since by lemma 4.10, for any given n, $L_t(S(n),t_c)$ is true. Hence, at the condition in which $L(S(n))$ is true, $y = \{0,..,n\}$, so $n \in y$. Also by lemma 4.10, $L_t(S(S(n)),t_c)$ is true. At the condition where $L(S(S(n)))$ is true, $y = \{0,..,S(n)\}$. Therefore, $S(n) \in y$ which is a contradiction. Thus, $\neg\exists n \in y(S(n) \notin y)$ and then $\forall n \in y(S(n) \in y)$. Therefore, $0 \in y \wedge \forall n \in y(S(n) \in y)$. □

At this point, it has been shown that y satisfies INF so we have no choice under CI but to believe that the infinity machine inserted every possible finite ordinal into y. However, the following lemmas and theorems provide a foundation to show this conclusion permitted by LEMI is false which also causes ZF to be inconsistent.

Using lemma 4.10, it is now proven if the CPU is at any instruction in any given loop iteration n then y is finite.

Lemma 4.12. *Start the IM and assume t_c seconds elapses on C after the start. Then for any given loop iteration n, y is finite at instructions 2 through 5*

PROOF. Apply LEMI and assume there is some loop iteration n in which y is not finite. By lemma 4.10, $L(n)$ evaluated as true during the t_c seconds. At that point of the execution, $(n = 0 \to y = \varnothing) \wedge (n > 0 \to y = \{0,...,n-1\})$ so y is finite at instruction 2 and 3 of loop iteration n. Then at instruction 4, the element n was added to the finite y by the $AddE(y, R0)$ operation at instruction 3. So, y is still finite. Finally, y is not changed by instruction 4 so y is finite at instruction 5. Therefore, y is finite at instructions 2 through 5 of loop iteration n, which is a contradiction. By LEMI, y is finite at instructions 2 through 5 for any given loop iteration n. □

The next lemma demonstrates that the IM implements NNU by showing no matter how much time elapses on C since starting the IM, the CPU does not exit the loop. This is true because at instruction 5 of every loop iteration k, $R0 = S(k)$ has not yet been inserted into the set y. Therefore with $n = S(k)$, $\exists n(NN(n) \wedge n \notin y)$ is true at the end loop iteration k. So, the exit condition cannot be met for any loop iteration because of NNU.

Lemma 4.13. *Start the IM and assume some arbitrary valid $t_x > 0$ seconds elapses on C after the start. Then the CPU did not exit the loop.*

PROOF. Set a breakpoint at instruction 6 and assume the conditions of the lemma. Then after t_x seconds elapses, assume the CPU exited the loop and is located at the breakpoint at instruction 6. Since the $R0$ register was initialized to the finite ordinal $\varnothing$ and only the successor function is applied to $R0$, then $R0$ can only contain some finite ordinal k. Therefore, the CPU executed to instruction 5 of loop iteration k. By lemma 4.12, y is finite at instruction 5. Hence, $\exists n(NN(n) \wedge n \notin y)$ is true. Therefore, the exit condition was not met contradicting the assumption that the CPU is at the instruction 6 breakpoint. Therefore, the CPU did not exit the loop. Since t_x is arbitrary then there is no given time in which the CPU exits the loop. □

According to lemma 4.12, at any stage in the construction of the finite ordinals, y is finite. Then by lemma 4.13, it is impossible for the CPU to exit the construction of the finite ordinals because of NNU. So, the above decidable specification gives us a precise way to scientifically and mathematically understand the interaction of NNU finite ordinal construction with the Cantor assertion that all finite ordinals can be generated.

Under these strict guidelines, Cantor's completed infinity is not an option as an outcome of NNU finite ordinal construction regardless of the speed at which each element is generated and inserted into the accumulating set y. The set y remains finite without any possibility of changing that state of logic.

The following collection of lemmas and theorems formalize the preceding argument.

Next it is proven if the IM executes for $t_c' > t_c$ seconds then $\exists n(\neg L_t(n,t_c))$ is true. This is a direct consequence of the fact that the CPU cannot exit the loop and the set y remains finite while in the loop.

Lemma 4.14. *Start the IM and assume t_c' seconds elapses on C after the start. Then $\exists n(\neg L_t(n,t_c))$. Also, whenever $t = t_c$ on C the set y is finite.*

PROOF. Set an interrupt $CPUI(t_c')$. Then assume the conditions of the lemma. After t_c' seconds elapses, the CPU is either in the loop or it is not. By lemma 4.13, the CPU did not exit the loop. Thus, the CPU interrupt was triggered somewhere inside the loop. Then $R0$ contains some finite ordinal say $R0 = k$ since only the successor function is applied to $R0$. Thus, either the CPU was executing loop iteration k when the interrupt occurred or it just completed instruction 4 of loop iteration $k-1$.

Therefore, when the elapsed time on C changed from 0 to t_c' the CPU executed from loop iteration 0 to at most some instruction in loop iteration k. Also, since $t_c < t_c'$ is a valid time period then $t = t_c$ on C sometime during the time interval $[0, t_c']$. Since the CPU was in the loop at the first instruction where $t = t_c'$ then it was also in the loop whenever $t = t_c$. So, let F be the set of loop iterations such that $t = t_c$. Since $t = t_c$ was not true at some instruction in loop iteration k given $t_c < t_c'$ then F contains at most $k+1$ elements. Thus, F is finite. By lemma 4.12, y was finite for each loop iteration $x \in F$, so y was finite whenever $t = t_c$. Finally, let $m = \max(F)$ since it is finite. Then the CPU did not reach instruction 2 of loop iteration $m+1$ during the t_c seconds. Therefore, with $n = m+1$, $\exists n(\neg L_t(n,t_c))$. □

One might think the above proof is invalid because the CPU execution time is zero when the time on C is t_c'. However, lemma 4.13 tested the reaction of the IM by assuming some arbitrary valid time $t_x > 0$ elapsed on C. This assumption is certainly permitted because *a posteriori* reasoning dictates that time always proceeds forward. The reaction is always the same though for any valid elapsed time. The IM does not exit the loop and the total execution time is zero seconds. So, the total execution time being zero with $t = t_x$ on C is a direct consequence of the assumption that each operation in an unending collection of iterative successor operations can take zero seconds to execute. Clearly, that assumption is inconsistent with the passage of time.

The next theorem ties together lemmas 4.11 and 4.14 to show ZF is inconsistent if LEMI, with variables that range over a domain of discourse that is said to contain all natural numbers, is presumed to be a fundamental principle of logic. From above, lemma 4.11 depends on lemma 4.10 which depends on LEMI. Lemma 4.14 depends on lemma 4.13 which shows that the IM implements NNU. The combination of the two results in a contradiction. Since mathematics accepted NNU before LEMI, then the addition of LEMI caused mathematics to be inconsistent.

Theorem 4.15. *ZF is inconsistent if it includes LEMI and a DDNN.*

PROOF. Start the IM and assume t_c' seconds elapses on C after the start. By lemma 4.14, y is finite whenever $t = t_c < t_c'$. However, by lemma 4.11, y satisfies INF at $t = t_c$. Then from Kunen's LEMI argument in section one, y contains all natural numbers so, y is infinite, which is a contradiction. Since LEMI with a DDNN was used to prove lemma 4.10 which in turn was used to prove y satisfies INF then ZF is inconsistent if it includes LEMI and a DDNN. □

The last theorem shows LEMI is not a valid principle of logic because with the construction of the finite ordinals above, LEMI evaluates as false.

Theorem 4.16. *Start the IM and assume t_c' seconds elapses on C after the start. Then LEMI evaluates as false whenever $t = t_c$. Therefore, LEMI is not a valid principle of logic.*

PROOF. By lemma 4.10, $\forall n(L_t(n,t_c))$. By lemma 4.14, $\exists n(\neg L_t(n,t_c))$. Therefore, $\forall n(L_t(n,t_c)) \wedge \exists n(\neg L_t(n,t_c)) \equiv \neg(\exists n(\neg L_t(n,t_c)) \vee \forall n(L_t(n,t_c)))$. □

As was shown in theorem 4.15, ZF is inconsistent if LEMI is included as a fundamental principle of logic. However, by theorem 4.16, LEMI is not a valid principle of logic so it cannot be used as a basis for arguments in ZF in any event. Without the indirect argument style of LEMI, any hope of resurrecting CI requires one to produce a direct argument showing that an ICSS can be completed in a universe that is completely finite.

But, the pursuit of infinite collections is a meaningless exercise anyway. Lemma 4.14 demonstrated that even if each natural number was iteratively generated in zero seconds, an infinite collection of natural numbers does not evolve after any given elapsed time $t_c > 0$. Consequently, infinite collections of natural numbers do not exist even in theory.

5. Conclusions

It was proven that an interval of time does not contain any countable infinite collection of points or potential infinite collection of points. So, any valid interval of time consists of a finite collection of ticks.

Similarly, it was shown that an interval of space does not contain any countable infinite collection of points or potential infinite collection of points. Therefore, a point that travels on a linear path through space only encounters a finite collection of connected line segments.

LEMI is accepted under ZF as a fundamental principle of logic. However, by using the infinity machine in this article, it was proven that LEMI causes ZF to be inconsistent. Without LEMI, ZF is nothing more than a theory of potential infinity. But, since this article proved space and time are finite and it is already agreed matter is finite then the entire universe is finite. Assuming that the objective of mathematics is provide machinery to help facilitate a more complete understanding of the nature of the universe then math objects should only be finite in order to match its finite properties.

References

[1] Chang, C. C.; Keisler, H. J. Model theory. Third edition. Studies in Logic and the Foundations of Mathematics, 73. North-Holland Publishing Co., Amsterdam, 1990. xvi+650 pp. ISBN 0-444-88054-2

[2] Church, Alonzo, On the law of excluded middle, Bull. Amer. Math. Soc., 34 (1928), no. 1, 75--78. http://projecteuclid.org/euclid.bams/1183492543.

[3] Copeland, Jack B. (4 June 2004). Hypercomputation: philosophical issues. Theoretical Computer Science 317 (1-3): 251-267

[4] Copeland, Jack B. & Shagrir, Oron (2011). Do Accelerating Turing Machines Compute the Uncomputable? Minds and Machines 21 (2): 221-239

[5] Feferman, Solomon., The development of programs for the foundations of mathematics in the first third of the 20th century, Web Accessed 12/02/2015

[6] Kunen, Kenneth, Set Theory. An Introduction to Independence Proofs, Stud. Logic Found. Math., vol. 102, North-Holland Publishing Co, Amsterdam, 1980

Andrew Banks

AndrewBanks10@gmail.com