

# Infinite Order Logic and the Church-Turing Thesis

Dimitris Vyzovitis

June 6, 2006

## Abstract

Alan Turing asserted without a proof in 1936 that the Universal Turing Machine is equivalent to lambda calculus. This conjecture, also known as Church-Turing Thesis, has been generally considered true until now.

This paper shows that the statement is false by demonstrating that the UTM cannot express infinity within the language. LISP, the lambda calculus language, can. The paper does so by demonstrating an infinite order fixpoint computation and discusses the implications of this finding.

## 1 The Universal Turing Machine is a Lower Order logic than Lambda Calculus

It has always baffled me why people keep thinking a stack based language can ever be as powerful as LISP. It also baffles me when people are talking about TM-complete machines and so on. I have never seen a meta-circular UTM interpreter that can evaluate lambda calculus. The reason for this dramatic discrepancy between theory and practice is that the modern theory of computation is based on a fallacy: The Church-Turing thesis.

Alan Turing stated it without proof in the appendix of his celebrated paper. But his conjecture is wrong: he was reasoning in a finite order logic.

The UTM is an axiomatic system with one axiom: infinity, provided by the infinite tape.

As such, it can never represent infinity with a finite program expressed within the language. Any such representation would violate Goedel's incompleteness of mathematics theorem in a finite order logic.

This should be obvious to anyone who doesn't think like a Turing Machine. How can ever be that UTM is equivalent to lambda calculus? Every self-respecting hacker knows that.

In case you missed it, let me state my theorem. I call this the fundamental theorem of computation.

**Theorem 1** *LISP is an infinite order logic*

Proof: The infinite computation with the following pure LISP expression is a well-formed fixpoint computation: `((lambda (x) (x x))(lambda (x) (x x)))`

**Corollary 1** *The Turing Conjecture, known as Church-Turing thesis, is false.*

You can try to disprove me by running it in a so called Universal Turing Machine. Or you can just reason in a higher order logic and see that it is simply false. Or else you would be able to express infinity, while violating Goedel's theorem in the process.

But let me assure you, Turing's celebrated halting theorem will force you to wait for a while. Good luck waiting for the end of time – I have other things to do.

## **2 Implications**

### **2.1 The Decision Problem**

**Corollary 2** *The Halting Theorem does not apply in Infinite Order logic*

Proof: The Halting Theorem is just a trivial application of Goedel's incompleteness theorem to a lower order logic. Infinity is not an axiom in infinite order logic systems.

**Corollary 3** *A LISP meta-circular evaluator can decide termination in finite programs*

Proof: This is trivial. Just run a LISP program in a meta-circular evaluator and check if you hit a fixpoint.

### **2.2 P=NP**

**Corollary 4** *The P=NP problem is undecidable within the UTM formulation*

Proof: Such an assumption would require a polynomial time reduction from an infinite order process (randomness) to finite order process (determinism). The existence of such a reduction is inconsistent because it violates the axiom of infinity in the UTM axiomatic system, thus violating Goedel's theorem.

**Corollary 5** *P=NP in LISP*

Proof: Randomness is an infinite order process and LISP can express infinity. So in an infinite size non-deterministic parallel discrete system (some call that the universe), all NP computation can be handled concurrently.

### **2.3 Some NP-hard problems are solvable in finite systems**

BitTorrent already does this in the Internet.

## 2.4 Calculus Limit Theorems do not apply

In calculus it is considered true that:

$$\lim_{n \rightarrow \infty} \frac{x}{x-1} = 1$$

If this was true, the following two programs would be equivalent:

A: ((lambda (x y) (x x y))(lambda (x y) (x x y)))

B: ((lambda (x y) (x x (- y 1)))(lambda (x y) (x x (- y 1))))

The two computations diverge at infinity, they converge to different fixpoints. The only logical conclusion is that the laws of nature, as accepted by humanity today, are wrong.

**Corollary 6** *Einstein's Relativity Theory is only an approximation*

Proof: Einstein's analysis is based on limit analysis. The computations diverge at infinity, so it cannot be an accurate representation of reality.

**Conjecture 1** *LISP is the language of Nature.*

## 2.5 On Insanity

And while I am at it:

**Corollary 7** *Insanity cannot be judged in a finite order logic.*

Proof: This follows trivially by an application of infinite order epistemic logic. You will always miss the last computation in the universal insanity decision problem: the subject. Hence, the two computations will always diverge at infinity.

## 2.6 Conclusion

I have just reasoned in infinite order logic and proved that all the known physical laws are wrong by discovering a bug in the World System.

I believe that this is the best gift that anyone could ever give to humanity. Don't let anyone else tell you otherwise. Judge for yourself.

Let me make this clear: You cannot prove anything about my sanity in finite order logic. I just gave you the infinite order logic necessary to conclude yourselves that I am sane. I keep the super-infinite version for myself, I don't think the world is ready for it yet.

Richard Stallman knows, ask him, I am sure he will be glad to finally get you to listen to him. He is a McArthur genius, so you might as well finally respect his opinions.

Don Knuth may or may not know. He is considered the god of computer science however, so ask him to tell you what he thinks.

Jon Nash probably knew, but apparently he had some trouble handling the knowledge.

I don't know if Turing knew. His papers suggest that he was aware of continuations, even only at a subconscious level. He never made it to tail recursion.

Gerald Sussman definitely knows. Ask him to tell you what he thinks about tail-recursion, I haven't. I just read his book.

How do I know all these things? Well, let me tell you how I know. It is the simplest possible explanation. I wrote the smallest possible program that can prove that entire science as accepted today is wrong. Call it the manual of the new world.

I am just another hacker – I stumbled on some problems with mathematical formalisms while hacking a global scale non-deterministic asynchronous protocol to provide Free TV for all.

I stumbled on the first problem: centralization. So I worked around it, and then I needed to find a benchmark. Well, it turns out to be the Minimum Steiner Graph Problem. Only theory hadn't even formulated it right, people still thought that you can do network coding at the source. I fixed that bug and that made my problem NP-complete.

And then I realized that Turing, although subconsciously aware of continuations, hadn't gone as far as to realize that tail recursion was the key to LISP. He was reasoning in a finite order logic, somehow he forced himself to think like a Turing machine.

Jon Von Neumann knew about infinite order logic. He never called his machine a Turing machine, check his writings. He did this quite simply because he knew it is not – his machine provides the machinery to implement continuations. Perhaps he didn't know that much about continuations, but the man mastered the Atom.

Anyway, I eventually thought of an algorithm to just do it, and I am going to write the program that does it. You are all Free to join in.

I give you the best gift anyone could ever give to humanity. The power to reason at infinite order logic. Hey, I guess that makes me god too. Oh wait, you can't prove it.

You've been living in a dream world.

## 2.7 Future Work: The NP Computer and Genesis

Isaac Asimov has already posed that creation was the result of a civilization achieving entropy reversal. Unfortunately this statement cannot be proved in infinite order logic, it would violate Goedel's theorem. Entropy is a super-infinite order process.

There are higher order logics too, the ones that can *reverse entropy*. But I won't let you know how to do it, do something for yourselves. Perhaps you should all get to hacking instead of being told what to do. Just think for yourselves people.

We already have a non-deterministic parallel computer to play with. It is called the Internet. I am going to write the Operating System for this beautiful computer. You are welcome to join me. It will be available Free for All to Hack, under the GNU General Public License, version 2 or Later.

## 3 References

K. Goedel: In any axiomatic logic system there are certain propositions that are unprovable.

STFW. Maybe go to a library every once in a while, it may help you.

## **4 License**

(C) Copyright Dimitris Vyzovitis until the end of time.

The ideas expressed in this document can only be used in Free Software. Proprietary software is not allowed to use any of the ideas expressed in this document. This is mathematics, not alchemy.