# COMPSCI 750: Computation Complexity
# First Assignment

Florian Suess

August 28, 2022

## 1 Program Analysis

### 1.1 Deciding if an even integer N is the sum of two primes

For clarity further along - we begin with a simple utility **isPrime** that via primitive trial division [1] method determines if an input candidate number $C$ is prime.

**Require:** $C > 1$
1: **if** $C = 2$ **then**
2: **return** 1
3: **end if**
4: $f \leftarrow C - 1$
5: **while** $f > 1$ **do**
6: **if** $f \mid C$ **then**
7: **return** 1
8: **end if**
9: $f \leftarrow f - 1$
10: **end while**
11: **return** 0

If it is not clear, we **return** to indicate the returning of a value to a potential caller. Suppose this algorithm runs on it's own, we interpret **return** as a **print** followed by a **stop**.

[1] as first described in `https://en.wikipedia.org/wiki/Liber_Abaci`

And so here is the requested algorithm that takes in even numbers greater than 2, returns "1" if it is indeed the sum of two prime numbers - "0" if not.

**Require:** $N > 2$ **and** $2 \nmid N$
1: $c \leftarrow N - 2$
2: **while** $c > 2$ **do**
3:  **if isPrime(c) then**
4:  $c' \leftarrow N - c$
5:  **if isPrime(c') then**
6:  **return** 1
7:  **end if**
8:  **end if**
9:  $c \leftarrow c - 1$
10: **end while**
11: **return** 0

## 1.2 Does the above halt?

We shall first prove **isPrime** halts on all valid inputs. Give me some $C \in \mathbb{N}$ where $C > 1$. If $C = 2$ then **isPrime(C)** halts. Let $(f_n)_{n\in\mathbb{N}}$ be a sequence into $\mathbb{N}$ representing the *potential assignments* to the variable $f$. We know $f_0 = C - 1 > 0$. We know that **isPrime** halts if for some $\theta \in \mathbb{N}$, if $f_\theta \leq 1$ thus can be viewed as a lower bound of the sequence. Looking at the set of assignment operations on $f$ we know that for any $t, v \in \mathbb{N}$ where $t < v \implies f_t < f_v$. It follows by the monotonic convergence theorem [2] (with small enough $\epsilon$) $\exists \pi \in \mathbb{N}$ such that $f_\pi = 1$ and so **isPrime** inevitably halts as we get to the $\pi^{\text{th}}$ assignment to $f$, even if $\forall x \in N, x < \pi$ we have $f_n \nmid C$.

We use this result of **isPrime** directly for the main algorithm by effectively viewing it as a bland statement. We construct an identical argument to above - so in greater strides; let $(c_n)_{n\in\mathbb{N}}$ be the sequence into $\mathbb{N}$ defined by the *potential assignments* to $c$, $c_0 = N - 2$ (for some $N > 2$). Similarly to above the sequence is also bounded below, this time by 2. Looking at all assignment statements to $c$, $(c_n)$ is indeed a monotonocally decreasing sequence. By the similar application of monotone convergence theorem, $\exists \alpha \in \mathbb{N}$ such that $c_\alpha = 2$. So this algorithm inevitably halts as we get to the $\alpha^{\text{th}}$ assignment to $c$ if this hasn't halted already earlier.

[2] `https://en.wikipedia.org/wiki/Monotone_convergence_theorem`

## 1.3 Does the proposed algorithm halt? And does it solve the Goldbach conjecture?

My algorithm being callable via **A(N)** for some $N$ [3]. Here is the caller algorithm for reference.[4]

1: $G \leftarrow 1$
2: $N \leftarrow 1$
3: **while** $A(N) = 1$ **do**
4: $\quad N \leftarrow N + 2$
5: **end while**
6: $G \leftarrow 0$
7: **stop**

Yes - we've solved the Goldbach conjecture. I'm heading to Faber and Faber to pick up my cool million dollars[5]. **I am ofcourse kidding**.

A nice property of $N$ is again it's monotonically increasing trajectory. For this algorithm to halt we need **A(N)** for some $N$ to at some point return a 0 value in order to break out of the otherwise infinite loop. Assuming $A$ is correct, for this to happen the Goldbach conjecture must be false! In which case the bound is set as the least greatest counter example and a halt is inevitable. Suppose though Goldbach's conjecture is true, in that case there exists no bound for this loop and we find that this program never halts.

[3] I inferred your intention of stepping $N$ in increments of 2 to be valid argument for $A$
[4] I adapted this to a while loop for clarity
[5] `https://www.math.tugraz.at/~elsholtz/WWW/papers/papers14faber.html`

# 2 Research Question

My interpretation of Joel David Hamkins proof via reference to John Baez; Computing the uncomputable blog post: a summation of Joel David Hamkins (and friends like Woodin) result.

## 2.1 The main result

There is **a** Turing machine $T$ that when given any function $f : N \to N$, there is a model of Peano Axioms[6] (PA), such that in this model, if we give $T$ any *standard* natural $n$ as input, it halts and outputs $f(n)$.

### 2.1.1 Preliminaries

A model of PA is a mere triple $(N, 0, S)$, where $N$ is a set, $0 \in N$ and $S : N \to N$ satisfies the PA the precise "structure" of $N$ and $S$ is up to interpretation (although it is worth noting that $N$ is necessarily infinite). The narrative goes; that we are all aware of the "standard" PA model; where the natural numbers are just "0, 1, 2, 3, ..." with the usual way of adding and multiplying them. What we need to do is speak about "non-standard" models of PA - construction originally driven by the works of Löwenheim–Skolem[7]. The existence of such models are initial segment isomorphic to the "standard" natural numbers. Initial segment[8] isomorphism[9] suggests the existence of an *upper set*[10], S, of elements;

$$S \subseteq N; \text{ if } s \in S \text{ and } n \in N \text{ where } s \leq n \implies n \in S \tag{1}$$

In John Baez's words - these are "tacked" on the end of the set of "standard" natural numbers. These "non-standard" numbers as having the property that they are strictly greater than 0 and every element of the model you can reach from 0 by applying $S$ (**"finitely"** many times). However, these elements still obey all the axioms of PA (which necessarily pushes an infinite size of this upper set also).

The reflection is necessary further on, as the core realisation is that these differing models of PA contrast against each other by differences in the ability to

[6] actually extends to models of ZFC (or more) in the original proof
[7] `https://mathworld.wolfram.com/Loewenheim-SkolemTheorem.html`
[8] initial segment: `https://proofwiki.org/wiki/Definition:Initial_Segment`
[9] Socratica explains isomorphism: `https://www.youtube.com/watch?v=BAmWgVjSosY`
[10] upper set: `https://en.wikipedia.org/wiki/Upper_set`

prove a sentence $p$ and indeed, Wether or not this sentence $p$ holds.

### 2.1.2 Notes on the construction of proof

The proof itself uses Rosser sentences[11], which themselves say: "For any proof of this sentence in a theory, there is a smaller proof of the negation of this sentence". The proof described in the original Hamkins blog derives **a single**[12] Turing machine $T$. Hamkins then continues to systematically *build a model of arithmetic* via an infinite sequence of these Rosser sentences and their negations that entirely depend on the given $f : N \to N$. The original turing machine $T$ then **via this created arithmetic model** can compute $f$ on it's standard input domain. That means that this Turing machine halts and computes $f(n)$ for all standard $n \in N$.

### 2.1.3 Contradicting the Halting Theorem?

This draws an interesting contrast to the halting theorem. Indeed the halting decider is just a mere function of shape $h : N \times N \to N$ that takes a program $p$ and input $n$ pair and returns "true" (1) or "false" (0), interpretted as if the program with input halts or not. We can systematically encode the program itself alongside it's input, and hence bijectively map this $N \times N$ input to just $N$, and so we really have this halting decider function $h : N \to N$.

Well then, here we apply this theorem provided by Hamkin and indeed use this magical Turing machine $T$, then within a model of PA, say $M = (\xi, 0, S)$, which we can compute the result of $h$ given any standard input. The turing machine halts and prints $f(n)$. Concretely we mean that the machine halts after some $\eta \in \xi$ steps. Note we cannot with certainty rule out that $M$ is indeed a "non-standard" model of PA, $\eta$ could be very well contained in the upper set $\gamma \subseteq \xi$ such that if there is no "finite set" of application of $S$ on 0 that can reach $\eta$. What interests me is the strong deduction of John Baez; that this Turing machine only ever halts in a "non-standard" amount of time.

With the halting theorem precisely saying; "There is no program solving correctly and in finite time the Halting Problem for every input pair" - it appears we do not have in front of us a counter example for the halting theorem due to this "non-standard" number of steps being relative to some model of PA. Avoiding the circular argument of finite steps, you could more reasonably assert the same principles discovered here, this idea of relativity, to argue that indeed

[11] `https://www.sciencedirect.com/science/article/pii/0003484379900172`

[12] In my opinion the most surprising result is, that there exists a natural number $p$ that codes this Turing machine

this halting theorem statement can very much still hold relative to a fixed model as we've discovered that this construction of this Turing machine requires many models in order to compute any $f : N \to N$.

### 2.1.4 Conclusion

What interests me is John Baez convictive stance around the surrounding vagueness of the definition of the "standard" model of PA. It's up to interpretation and suggests that we all could be all be hosting a "divergent" interpretation of this model. So it makes me question to what gravity does the phrase "standard model" even have?

# 3 Mathematical Modelling

## 3.1 Monochromatic arithmetic progressions

Given a infinite binary sequence $s = s_1 s_2 s_3 ...$, that is for any $\omega \in N$, we have $s_\omega = \{0, 1\}$. Given some $k > 0$, we say we can find a monochromatic arithmetic progression of length $k$, that is, $\exists i, t \geq 1$ such that $s_i s_{i+t} s_{i+2t} ... s_{i+(k-1)t}$ is $\in \{1^k, 0^k\}$.[13]

## 3.2 Does every infinite sequence contain a monochromatic arithmetic progression?

This is a vacuous application of a finding directly from our lectures; the **finite Van der Waerden theorem**[14]. Give me your infinite binary sequence $s$ and desired $k$ value. In close relation to the theorem delivery, fix $c = 2$ (colouring). By theorem, there $\exists \gamma$ such that the prefix of length $\gamma + 1$ contains this monochromatic arithmetic progression of length $k$.

## 3.3 And for ternary sequences?

Given a infinite binary sequence $s = s_1 s_2 s_3 ...$, that is for any $\omega \in N$, we have $s_\omega = \{0, 1, 2\}$. Given some $k > 0$, we say we can find a monochromatic arithmetic progression of length $k$, that is, $\exists i, t \geq 1$ such that $s_i s_{i+t} s_{i+2t} ... s_{i+(k-1)t}$ is $\in \{2^k, 1^k, 0^k\}$.

The proof of this statement; surely it's sufficiently apt to prescribe the exact same proof applied as above where instead we now fix $c = 3$.

[13] Evidently a slight amendment to the original question as it concretely seemed to have wanted me to just find "any" monochromatic arithmetic progression which would be met with me showing the 3 letter prefix contained a monochromatic arithmetic progression of length 2 by virtue of Pigeonhole Principle

[14] visually the theorem `https://www.youtube.com/watch?v=Hb35djuGMIg`