BSS Machines: Computability without Search Procedures

Russell Miller,

Queens College &

Graduate Center – CUNY

August 19, 2009

Effective Mathematics of the Uncountable

CUNY Graduate Center

Some of this work is joint with Wesley Calvert.

Turing-Computable Fields

Defn.: A *computable field* F is a field with domain ω, in which the field operations $+$ and $\cdot$ are (Turing-)computable.

One considers the *root set* and the *splitting set*:

$$R_F = \{p \in F[X] : (\exists a \in F)\ p(a) = 0\}$$

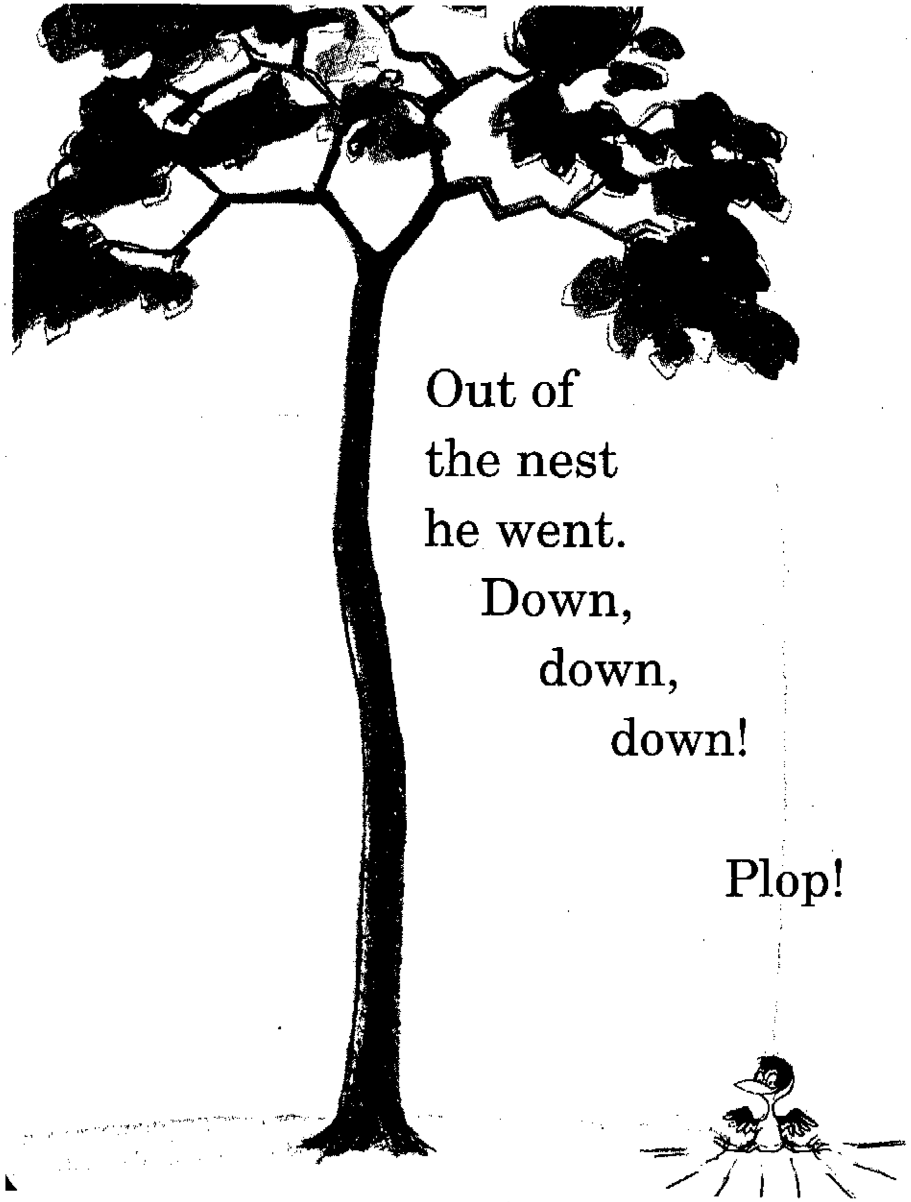
$$S_F = \{p \in F[X] : p \text{ factors properly in } F[X]\}.$$

From these sets, one can find the irreducible factors, hence the roots, of any $p \in F[X]$. Finding roots or factors requires only a simple search procedure, provided that they do exist.

BRIGHT and EARLY BOOKS

Are You My Mother?

by P. D. Eastman

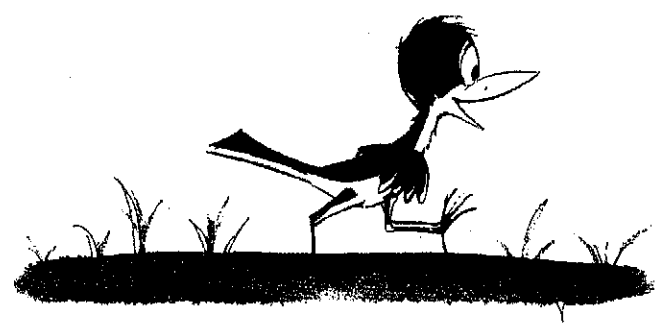


Out of
the nest
he went.
Down,
down,
down!

Plop!

The baby bird could not fly.

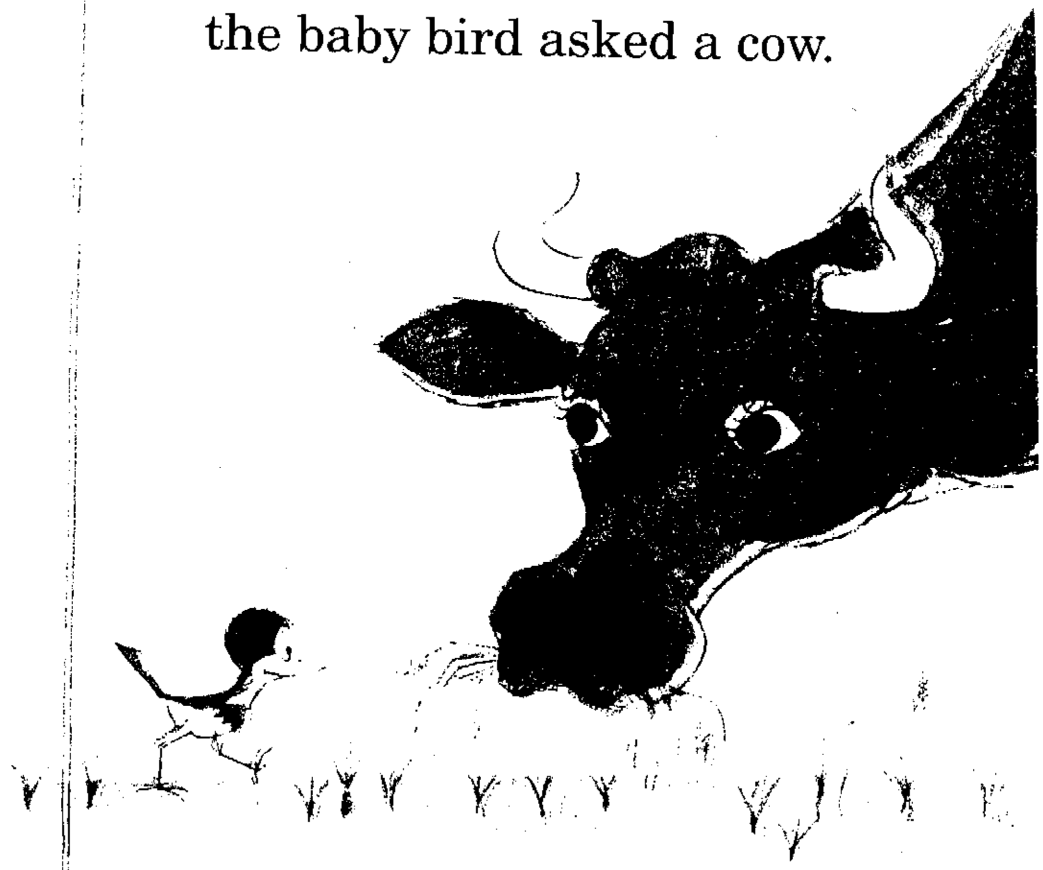
But he could walk.
"Now I will go and find
my mother," he said.



"Are you my mother?"
the baby bird asked a dog.

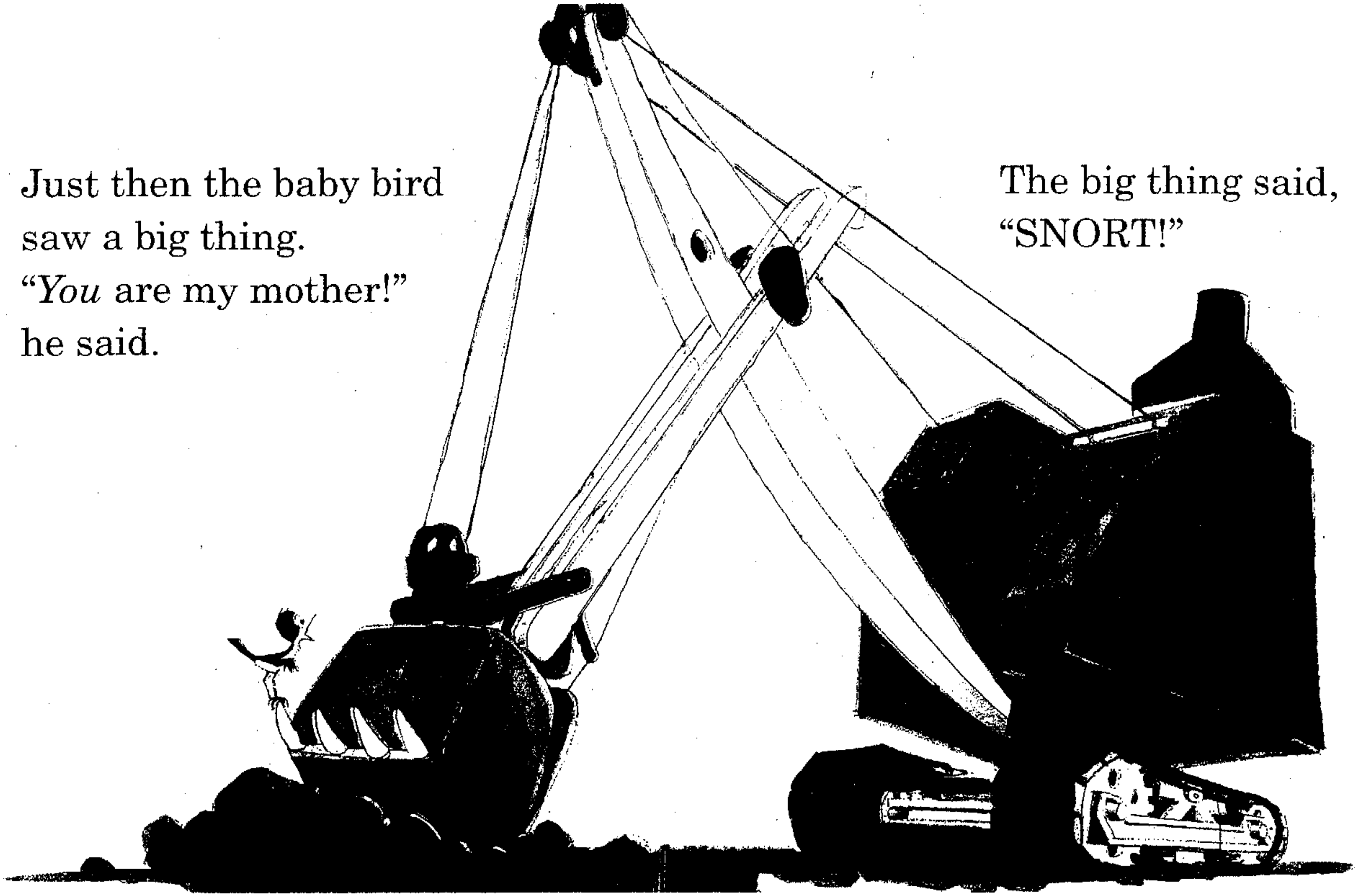
"I am not your mother.
I am a dog," said the dog.

"Are you my mother?"
the baby bird asked a cow.



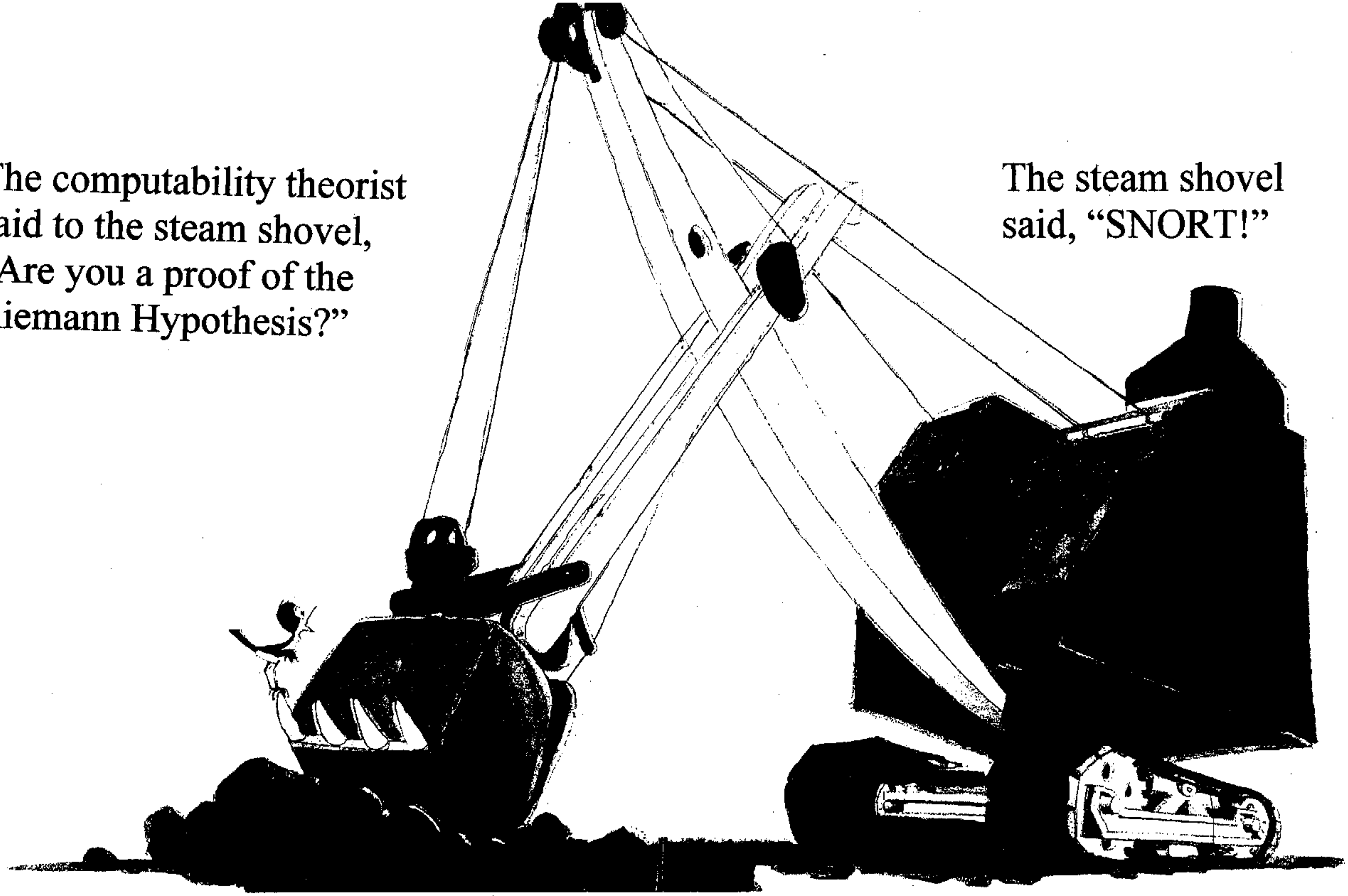
"How could I be your mother?"
said the cow. "I am a cow."

Just then the baby bird
saw a big thing.
"*You* are my mother!"
he said.



The big thing said,
"SNORT!"

The computability theorist said to the steam shovel, "Are you a proof of the Riemann Hypothesis?"



The steam shovel said, "SNORT!"

BSS Computability

Defn.: A *BSS-machine* has an infinite tape, indexed by ω. At each stage, cofinitely many cells are blank, and finitely many contain one real number each. In a single step, the machine can copy one cell into another, or perform a field operation ($+$, $-$, $\cdot$, or $\div$) on two cells, or compare any cell to 0 (using $<$ or $=$) and fork, or halt.

The machine starts with a tuple $\vec{p} \in \mathbb{R}^{<\omega}$ of real parameters in its cells, and the input consists of a tuple $\vec{x} \in \mathbb{R}^{<\omega}$, written in the cells immediately following $\vec{p}$. The machine runs according to a finite program, and if it halts within finitely many steps, the output is the tuple of reals in the cells when it halts.

BSS-Semidecidability

Defn.: A set $S \subseteq \mathbb{R}$ is:

- *BSS-decidable* if χ_S is BSS-computable;
- *BSS-enumerable* if S is the image of ω ($\subseteq \mathbb{R}$) under some partial BSS-computable function;
- *BSS-semidecidable* if S is the domain of such a function.

So
{BSS-decidable sets} $\subseteq$ {BSS-semidecidable sets}
and
{BSS-enumerable sets} $\subseteq$ {BSS-semidecidable}.
However, the set $\mathbb{A}$ of algebraic real numbers is BSS-semidecidable, but turns out not to be BSS-enumerable, nor BSS-decidable. Indeed, $\mathbb{Q}$ is not BSS-decidable. And there exist countable BSS-decidable sets which are not BSS-enumerable. (Proofs by Herman-Isard, Meer, Ziegler.)

Field Questions on $\mathbb{R}$

Lemma (Folklore): The splitting set $S_{\mathbb{R}}$ and the root set $R_{\mathbb{R}}$ are both BSS-decidable. Also, the number of real roots of $r \in \mathbb{R}[X]$ is BSS-computable.

Lemma: If $f : \mathbb{R}^m \to \mathbb{R}^n$ is BSS-computable by a machine with real parameters $\vec{p}$, then for all $\vec{x} \in \mathbb{R}^m$, $f(\vec{x})$ lies in the field $\mathbb{Q}(\vec{x}, \vec{p})$.

Corollary: $\mathbb{A}$ is not BSS-enumerable. Indeed, every BSS-enumerable set is contained in a finitely generated extension of $\mathbb{Q}$.

Corollary: No BSS-computable function can accept all inputs $q \in \mathbb{Q}[X]$ and output the real roots of each input q. (Hence neither can it output the irreducible factors of q in $\mathbb{R}[X]$.)

Intuition: finding roots of a polynomial requires an AYMM search.

Alternative Proof

Prop.: Neither $\mathbb{Q}$ nor $\mathbb{A}$ is BSS-decidable.

Proof: Suppose some BSS machine M computes a total function $H : \mathbb{R} \to \mathbb{R}$, using real parameters $\vec{p}$. Choose an input $y \in \mathbb{R}$ transcendental over $\mathbb{Q}(\vec{p})$, and run M on y. At each stage s, the n-th cell contains $f_{n,s}(y)$, for some $f_{n,s} \in \mathbb{Q}(\vec{p})(Y)$. Then there exists $\epsilon > 0$ such that when $|x - y| < \epsilon$, each step by M on input x is identical to the computation on y, with $f_{n,s}(x)$ in place of $f_{n,s}(y)$ in the n-th cell. So, on the ϵ-ball around y, M computes a $\mathbb{Q}(\vec{p})$-rational function of its input. We say that M computes a function which is *locally* $\mathbb{Q}(\vec{p})$*-rational at transcendentals* over $\vec{p}$.

If M computes the characteristic function of $S \subseteq \mathbb{R}$, then it must be constant on such ϵ-balls. So either S or $\overline{S}$ is not dense in $\mathbb{R}$.

Application to Finding Roots

Suppose that M, on every input $\langle a_0, \ldots, a_4 \rangle$, outputs a real root of $X^5 + a_4 X^4 + \cdots + a_1 X + a_0$. Choosing $\vec{a} \in \mathbb{R}^5$ algebraically independent over the parameters $\vec{p}$ of M, we would have a rational function over $\mathbb{Q}(\vec{p})$ which gives a root of each monic degree-5 polynomial in $\mathbb{R}[X]$ with coefficients within ϵ of $\vec{a}$. But then this rational function extends from this open ϵ-ball to give a general formula for such a root. By the Ruffini-Abel Theorem, this is impossible.

The same would hold even for BSS machines enhanced with the ability to find n-th roots of positive real numbers.

Algebraic Numbers of Degree d

Defn.: $\mathbb{A}_d$ is the set of all algebraic real numbers of degree $\leq d$ over $\mathbb{Q}$. $\mathbb{A}_{=d}$ is the set $(\mathbb{A}_d - \mathbb{A}_{d-1})$.

Question (Meer-Ziegler): Can a BSS machine with oracle $\mathbb{A}_d$ decide the set $\mathbb{A}_{d+1}$?

Answer (work in progress): No. So we have

$$\mathbb{Q} = \mathbb{A}_1 \prec_{BSS} \mathbb{A}_2 \prec_{BSS} \mathbb{A}_3 \prec_{BSS} \cdots \prec_{BSS} \mathbb{A}.$$

Proving $\mathbb{A}_{d+1} \not\leq \mathbb{A}_d$

A process similar to before: If M with parameters $\vec{p}$ is an oracle BSS-machine deciding $\mathbb{A}_{d+1}$ from oracle $\mathbb{A}_d$, let y be transcendental over $\mathbb{Q}(\vec{p})$. Then $M^{\mathbb{A}_d}$ on input y halts and outputs 0, with finitely many $f \in \mathbb{Q}(\vec{p})(Y)$ giving the values in its cells during the computation. We claim that $\exists x \in \mathbb{A}_{d+1}$ sufficiently close to y that $M^{\mathbb{A}_d}$ on input x mirrors this computation and also outputs 0.

Let F be the set of nonconstant $f(Y)$ used. Problem: we need to ensure $f(x) \notin \mathbb{A}_d$ for every $f \in F$.

Getting all $f(x) \notin \mathbb{A}_d$

- We may ignore any $f \in F$ in which a transcendental parameter p_i appears. So assume there is a single algebraic parameter p.
- If $f(x) = a \in \mathbb{A}_d$, and $f(X) = \frac{g(X)}{h(X)}$ with $g, h \in \mathbb{Q}(p)[X]$, then x is a root of $f_a(X) = g(X) - ah(X) \in \mathbb{A}_d(p)[X]$. Make sure that the minimal polynomial $q(X)$ of x over $\mathbb{Q}$ stays irreducible in $\mathbb{A}_d(p)[X]$, so that if $f(x) = a$, then $q(X)$ would divide $f_a(X)$.
- We can choose such $q(X)$ so that $q(X)$ does not divide any $f_a(X)$ with $a \in \mathbb{A}_d$ and $f \in F$. Indeed, with all other coefficients fixed, there are only finitely many constant terms q_0 which would allow q to divide any f_a.
- Choose q_0 so that $q(X)$ has a real root x within ϵ of y.

Summary

For the root x of $q(X)$ chosen above, we have $x \in \mathbb{A}_{d+1}$. Since $|x - y| < \epsilon$, we know that for all nonconstant $f \in F$, $f(x)$ and $f(y)$ have the same sign, with $f(x) \notin \mathbb{A}_d$ and $f(y) \notin \mathbb{A}_d$. So the computation by the BSS machine M on input x parallels that on input y, and both halt (at the same step) and output 0. Thus $M^{\mathbb{A}_d}$ does not decide the set $\mathbb{A}_{d+1}$.

More General Questions

- What other AYMM searches can be investigated by translating them into problems in $\mathbb{R}$ and trying to compute them using BSS machines?
- Can one do anything similar with Infinite Time Turing Machines?
- Is there any way to consider AYMM searches for Gödel numbers of proofs?