# Primitive Recursive Arithmetic

Lecture 19

November 1, 2016

## 1 Topics

(1) Finishing up non-standard analysis from H.Jerome Keisler's book *Elementary Calculus* (logician's pun on "elementary," it also means "first-order" in some contexts).

(2) Introduction to the classical versus constructive mathematics "divide." This will take us through

- constructive geometry,
- constructive/computational number theory (Heyting Arithmetic),
- constructive analysis (Brouwer and Bishop), and
- constructive FOL, sometimes called iFOL .

(3) These topics will inform constructive type theory.

## 2 Calculus from H. Jerome Keisler *Elementary Calculus*, 1976 p.145

Def: $f$ is *continuous* at a point $c \in R$ $iff$

(i) $f$ is defined at $c$ and

(ii) whenever $x$ is infinitely close to $c$ ($x \approx c$), $f(x)$ is infinitely close to $f(c)$, i.e. $f(x) \approx f(c)$.

Thm 1: $f$ is continuous at $c$ $iff$ $\lim_{x \to c} f(x) = f(c)$.

Def: $L$ is the limit of $f(x)$ as $x$ approaches $c$ if whenever $x$ is infinitely close to $c$ but not equal to it, $f(x)$ is infinitely close to $L$, i.e. whenever $x \approx c$ and $x \neq c$, $f(x) \approx L$.

Note: We can have $\lim_{x \to c} f(x)$ yet $f(c)$ is not defined.

Def: (p.73) Given $f \in \mathbb{R} \to \mathbb{R}$, the *derivative* of $f$ is the function $f'$ whose value at $x$ is the *slope* of $f$ at $x$, i.e. $f'(x) = st\left(\frac{f(x + \Delta x) - f(x)}{\Delta x}\right)$ whenever the slope exists.

# 3 Constructive vs Classical Mathematics

(from Harold M. Edwards: *Essays in Constructive Mathematics*)

Preface

Mathematics came to a fork in the road around 1880

| | | |
|---|---|---|
| Dedekind, Cantor, Weierstrass<br>(accepted "transfinite functions") | vs | Kronecker, (spirit of Gauss, Dirichlet)<br>later Brouwer, Poincaré, Bishop |

"The advent of computers has had a profound impact on mathematics, directing attention to algorithms."

---

We are not justified in this course to pursue the theory of polynomials that Edwards uses to illustrate constructivity. Moreover, there are good examples closer to Computer Science (CS) that make his points.

We can rely on a simpler and more universal set of concepts arising from Euclidean Geometry, a theory with constructive aspirations in which computations are essential.

We will look at the historical progression of constructive mathematics.

I. Euclidean Geometry - naive and informal, *synthetic* geometry

| | |
|---|---|
| Euclid's *Elements* | 300 BC Alexandria, Proclus commentary |
| Appollonius | Died in 190 BC |
| Book VIII | Euclid Proposition 20 : There are infinitely many primes (prime numbers are more than any assigned magnitude of prime numbers). |

II. Descartes - *analytic geometry*

| | |
|---|---|
| *Discourse on Method* | 1637, with appendices |
| *La Geométrie* | Coordinate geometry (only book he wrote on mathematics) |
| *Rules for the Direction of Mind* | Isolated from *Discourse* |

(a) I think, therefore I am

(b) Each phenomenon must have a cause

(c) An effect cannot be greater than its cause

(d) The mind has innate (knowledge) of perfection, space, time, motion
(The idea of perfection cannot be created by the imperfect mind of man, thus God exists. God made nature to follow mathematical laws.)

III. Dedekind - (1888) Recursive definitions

IV. Peano - (1889) The principles of arithmetic presented by a new method
(He used recursive definitions and induction. Next we look at primitive recursive definitions of numerical functions.)

# 4 Examples of Primitive Recursion

$a_0(x, y) = x + 1$ $\qquad$ $s(x) = x + 1$ $\qquad$ so $\qquad$ $a_0 = s(x)$

$a_1(x, y) = add(x, y)$

$$\boxed{\begin{array}{l} add(0, y) = y \\ add(s(x), y) = s(add(x, y)) \end{array}}$$

note $\quad s(add(x, y) = a_0(a_1(x, y), y)$

$a_2 = mult(x, y)$

$$\boxed{\begin{array}{l} mult(0, y) = 0 \\ mult(s(x), y) = add(mult(x, y), y) \end{array}}$$

i.e. $(x + 1) \cdot y = x \cdot y + y$
note $\quad add(mult(x, y), y) = a_1(a_2(x, y), y)$

$a_3(x, y) = exp(x, y)$

$$\boxed{\begin{array}{l} exp(0, y) = 1 \qquad \text{i.e. } y^0 = 1 \\ exp(s(x), y) = mult(exp(x, y), y) \end{array}}$$

i.e. $y^{x+1} = y^x \cdot y$
note $\quad mult(exp(x, y), y) = a_2(a_3(x, y), y)$

$a_4(x, y) = hypexp(x, y)$

$$\boxed{\begin{array}{l} hypexp(0, y) = y \\ hypexp(s(x), y) = exp(hypexp(x, y), y) \end{array}}$$

note $\quad exp(hypexp(x, y), y) = a_3(a_4(x, y), y)$

$a_{n+1}(x, y)$

$$\boxed{\begin{array}{l} a_{n+1}(0, y) = y \\ a_{n+1}(s(x), y) = a_n(a_{n+1}(x, y), y) \end{array}}$$

$a_n(x, y)$ can be thought of as a function of $n, x, y$. This is *Ackerman's function* in one form. It is not primitive recursive.