

# Computability via Recursive Functions

Justin Pumford

March 2020

## 1 Effective Calculability and Computability

How does one rigorously define what functions can be calculated by means of an algorithm? It is easy to see that  $f(x) = x^3 + 2$  can be calculated by a combination of algorithms for exponents and addition.

But what about  $isRing(X, +, *)$ , a function that accepts a set and two binary operators? It returns 0 if the set is not a ring under the given operations, and 1 if the set is a ring under the given operations.

Many problems in math employ creativity in solving and may not always have a step-by-step way of solving them. Is there a way to separate what can be solved via an algorithm and what cannot?

Enter the concept of effective calculability. An effectively calculable function is one for which an effective method exists to calculate the value of the function. Paraphrased from The Cambridge Dictionary of Philosophy, an effective method:

1. has a finite number of finite instructions
2. always finishes after a finite number of steps
3. always produces a correct answer
4. could be done by a human with a pencil and paper
5. requires no ingenuity to complete

Obviously this definition is not mathematically rigorous enough for us to use in a proof. We can, however, capture the essence in this definition by constructing functions. These will capture ideas familiar to computer scientists and programmers, such as function "composition" and "loops", all of which are able to be done via pencil and paper by a sufficiently patient and long-lived individual. These definitions were formalized in the early part of the 1900s.

## 2 Primitive Recursive Functions

### 2.1 Functions

For this paper,  $\mathbb{N}$  refers to the set  $\{0, 1, 2, 3, \dots\}$

**Definition.** A total function  $f$  from a set  $X$  to a set  $Y$  is a subset of the Cartesian product  $X \times Y$  so that the following hold:

$$\begin{aligned} \forall x \in X \exists y \in Y : (x, y) \in f \\ (x, y) \in f \wedge (x, z) \in f \implies y = z \end{aligned}$$

such that each  $x \in X$  only shows in exactly one element  $(x, y) \in f$ .

All functions are total unless otherwise indicated.

**Definition.** A function  $f : X \rightarrow Y$  has arity  $n$  if  $X$  is composed of  $n$ -tuples from sets  $X_1, \dots, X_n$ . If  $X$  is not composed of tuples at all, then the  $f$  is unary.

**Definition.** The following functions from  $\mathbb{N} \oplus \dots \oplus \mathbb{N}$  to  $\mathbb{N}$  are primitive recursive functions:

1. The unary constant function  $c$ :

$$c(x) = 0$$

2. The unary successor function  $s$ :

$$s(x) = x + 1$$

3. The  $n$ -ary projection function  $p$ :

$$1 \leq i \leq n$$

$$p_i(x_1, \dots, x_n) = x_i$$

4. Function composition

Let  $f$  be an  $n$ -ary primitive recursive function and  $g_1, g_2, \dots, g_n$  all be  $m$ -ary primitive recursive functions. Then the  $m$ -ary composition  $h$  of  $f$  and  $g_1, g_2, \dots, g_n$  given by

$$h(x_1, x_2, \dots, x_m) = f(g_1(x_1, x_2, \dots, x_m), \dots, g_n(x_1, x_2, \dots, x_m))$$

is a primitive recursive function

5. Primitive recursion Let  $g$  be an  $n$ -ary primitive recursive function and  $f$  be an  $(n+2)$ -ary primitive recursive function. Then the  $(n+1)$ -ary primitive recursion  $h$  of  $f$  and  $g$  given by

$$h(0, x_1, \dots, x_n) = g(x_1, \dots, x_n)$$

$$h(s(x), x_1, \dots, x_n) = f(x, h(x, x_1, \dots, x_n), x_1, \dots, x_n)$$

is a primitive recursive function

### 3 The Ackermann Function

**Definition** (The Ackermann Function). Let  $n, m \in \mathbb{N}$ . Then define  $A(n, m)$  as follows:

$$A(n, m) = \begin{cases} m + 1 & n = 0 \\ A(n - 1, 1) & n > 0 \wedge m = 0 \\ A(n - 1, A(n, m - 1)) & n > 0 \wedge m > 0 \end{cases}$$

**Theorem 1.** For any  $n, m \in \mathbb{N}$ ,  $A(n, m) \in \mathbb{N}$

*Proof.* We proceed by double induction.

Clearly  $A(0, k) \in \mathbb{N}$  for every  $k \in \mathbb{N}$  because  $k + 1 \in \mathbb{N}$ . Now assume  $A(k, m) \in \mathbb{N}$  for some  $k \in \mathbb{N}$  for every  $m \in \mathbb{N}$ . Then  $A(k + 1, 0) = A(k + 1 - 1, 1) = A(k, 1) \in \mathbb{N}$ . Now assume  $A(k + 1, j) \in \mathbb{N}$  for some  $j \in \mathbb{N}$ . Then  $A(k + 1, j + 1) = A(k + 1 - 1, A(k + 1, j + 1 - 1)) = A(k, A(k + 1, j))$ . By our second hypothesis  $A(k + 1, j) \in \mathbb{N}$ , so by our first hypothesis  $A(k, A(k + 1, j)) \in \mathbb{N}$ . Therefore  $A(k + 1, j + 1) \in \mathbb{N}$ . Then  $A(k + 1, m) \in \mathbb{N}$  for every  $m \in \mathbb{N}$ , so  $A(n, m) \in \mathbb{N}$  for all choices of  $n, m \in \mathbb{N}$   $\square$

**Theorem 2.** *A is a total function*

*Proof.* By Theorem 1, all  $n, m$  in the domain of  $A$  have an image  $A(n, m) \in \mathbb{N}$ . Therefore  $A$  is a total function.  $\square$

**Theorem 3.** *For any  $m \in \mathbb{N}$ ,  $A(1, m) = m + 2$*

*Proof.* We proceed by induction.

$A(1, 0) = A(1 - 1, 1) = A(0, 1) = 2 = 0 + 2$ . Now assume  $A(1, k) = k + 2$  for some  $k \in \mathbb{N}$ . Then  $A(1, k + 1) = A(1 - 1, A(1, k + 1 - 1)) = A(0, A(1, k)) = A(0, k + 2) = k + 3 = (k + 1) + 2$ .  $\square$

**Lemma 1.** *For every  $n, m \in \mathbb{N}$ ,  $m < A(n, m)$*

*Proof.* We proceed by double induction.  $A(0, 0) = 0 + 1 = 1 > 0$ . Clearly  $A(0, m) = m + 1 > m$  for every  $m \in \mathbb{N}$ . Now assume  $m < A(k, m)$  for some  $k \in \mathbb{N}$  for every  $m \in \mathbb{N}$ . Now we show that  $m < A(k + 1, m)$  for every  $m \in \mathbb{N}$ .  $A(k + 1, 0) = A(k, 1)$ . By our inductive hypothesis,  $A(k, 1) > 1 > 0$ . So  $A(0, 0) < A(k + 1, 0)$ . Now assume  $j < A(k + 1, j)$  for some  $j \in \mathbb{N}$ . Then  $A(k + 1, j + 1) = A(k, A(k + 1, j))$ . By our first inductive hypothesis,  $A(k + 1, j) < A(k, A(k + 1, j))$ . By our second inductive hypothesis,  $j < A(k + 1, j)$ . So  $j + 1 \leq A(k + 1, j) < A(k, A(k + 1, j)) = A(k + 1, j + 1)$ .  $\square$

**Theorem 4.** *For any  $n, m, s \in \mathbb{N}$  where  $m < s$ ,  $A(n, m) < A(n, s)$*

*Proof.* Let  $m, s, n \in \mathbb{N}$ . Since  $m < s$  and  $m, s \in \mathbb{N}$ , we have a  $d = s - m$  where  $d \in \mathbb{Z}^+$ . We will proceed by induction to first show  $A(n, m) < A(n, m + 1)$ .  $A(0, m) = m + 1 < m + 1 + 1 = A(0, m + 1)$ . Assume  $A(k, m) < A(k, m + 1)$  for some  $k \in \mathbb{N}$ . By Lemma 1,  $A(k + 1, m) < A(k, A(k + 1, m)) = A(k + 1, m + 1)$ . So  $A(k + 1, m) < A(k + 1, m + 1)$ . Therefore  $A(n, m) < A(n, m + 1)$ .

Now we use this result to construct a chain of  $d$  inequalities  $A(n, m) < A(n, m + 1) < A(n, m + 1 + 1) < \dots < A(n, m + 1 + 1 + \dots + 1) = A(n, m + d) = A(n, s)$ .  $\square$

**Theorem 5.** *For any  $n, m \in \mathbb{N}$ ,  $A(n, m + 1) \leq A(n + 1, m)$*

*Proof.* We proceed by induction.  $A(n, 0 + 1) = A(n, 1) = A(n + 1, 0)$  by the definition of the Ackermann function. Now assume  $A(n, k + 1) \leq A(n + 1, k)$  for some  $k \in \mathbb{N}$ . Note that  $k + 1 < A(n, k + 1)$  by Theorem 1. So  $k + 2 \leq A(n, k + 1)$ . Then  $A(n, k + 2) \leq A(n, A(n, k + 1))$  by Theorem 4. By our

inductive hypothesis,  $A(n, k + 1) \leq A(n + 1, k)$ , so by Theorem 4 again,  $A(n, A(n, k + 1)) \leq A(n, A(n + 1, k))$ . By the definition of the Ackermann function,  $A(n, A(n + 1, k)) = A(n + 1, k + 1)$ .  $\square$

**Theorem 6.** *For any  $n, m, s \in \mathbb{N}$  where  $n < s$ ,  $A(n, m) < A(s, m)$*

*Proof.* Let  $n, s, m \in \mathbb{N}$ . Since  $n < s$  and  $n, s \in \mathbb{N}$ , we have a  $d = s - n$  where  $d \in \mathbb{Z}^+$ . We begin by showing  $A(n, m) < A(n + 1, m)$ .  $A(n, m) < A(n, m + 1)$  by Theorem 4, and  $A(n, m + 1) \leq A(n + 1, m)$  by Theorem 5. So  $A(n, m) < A(n + 1, m)$ . Now we use this result to construct a chain of  $d$  inequalities  $A(n, m) < A(n + 1, m) < A(n + 1 + 1, m) < \dots < A(n + 1 + 1 + \dots + 1, m) = A(n + d, m) = A(s, m)$ .  $\square$

**Theorem 7.** *For any  $n, m, s \in \mathbb{N}$ ,  $A(n, A(s, m)) < A(n + s + 2, m)$*

*Proof.*  $A(n, A(s, m)) < A(n + s, A(s, m))$  by Theorem 6.  $A(n + s, A(s, m)) < A(n + s, A(s + n + 1, m))$  by Theorem 4 since, by Theorem 6,  $A(s, m) < A(s + n + 1, m)$ .  $A(n + s, A(s + n + 1, m)) = A(n + s, A(n + s + 1, m)) = A(n + s + 1, m + 1)$  by the definition of the Ackermann function. Finally,  $A(n + s + 1, m + 1) \leq A(n + s + 1 + 1, m) = A(n + s + 2, m)$  by Theorem 5.  $\square$

**Theorem 8.** *For any  $m \in \mathbb{N}$ ,  $A(2, m) = 2m + 3$*

*Proof.*  $\square$

**Definition.** *Let  $P$  be a set all primitive recursive functions  $f$  so that if  $f \in P$ , there is a  $t \in \mathbb{N}$  such that for any  $x_1, \dots, x_n \in \mathbb{N}$ ,  $f(x_1, \dots, x_n) < A(t, \max\{x_1, \dots, x_n\})$ .*

**Theorem 9.**  $c(x), s(x), p_i(x_1, x_2, \dots, x_n) \in P$

*Proof.* Let  $m = \max\{x_1, \dots, x_n\}$ . Then

$$\begin{aligned} c(x) &= 0 < x + 1 = A(0, x) \\ s(x) &= x + 1 < x + 2 = A(1, x) \\ p_i(x_1, x_2, \dots, x_n) &= x_i \leq m < m + 1 = A(0, m) \end{aligned}$$

$\square$

**Theorem 10.**  *$P$  is closed under composition*

*Proof.* Let  $f, g_1, g_2, \dots, g_k \in P$ , where  $f$  is  $k$ -ary and each  $g_i$  is  $j$ -ary. Let  $x_1, x_2, \dots, x_j \in \mathbb{N}$ . Let  $m = \max\{x_1, x_2, \dots, x_j\}$ . Let  $h$  be the  $j$ -ary primitive recursive function that results from function composition of  $f$  with  $g_1, g_2, \dots, g_k$ . Let  $g_{\max}$  be the  $g_i$  giving the maximum value in  $\max\{g_1(x_1, \dots, x_j), \dots, g_k(x_1, \dots, x_j)\}$ . Let  $m_g = g_{\max}(x_1, \dots, x_j)$ . Since each  $g_i \in P$ , there is some  $t_i$  for each  $g_i$  such that  $g_i(x_1, \dots, x_j) < A(t_i, m)$ . Take  $t_g = \max\{t_1, t_2, \dots, t_k\}$ . Note this is not dependent on  $x_1, \dots, x_j$ . Similarly since  $f \in P$ , there exists some  $t_f \in \mathbb{N}$  so that  $h(x_1, \dots, x_j) = f(g_1(x_1, \dots, x_j), \dots, g_k(x_1, \dots, x_j)) < A(t_f, m_g)$ . Note this is also not dependent on  $x_1, \dots, x_j$ . But since  $m_g < A(t_g, m)$ , by Theorem 4  $A(t_f, m_g) < A(t_f, A(t_g, m))$ . By Theorem 7,  $A(t_f, A(t_g, m)) < A(t_f + t_g + 2, m)$ . Let  $t = t_f + t_g + 2 \in \mathbb{N}$ . Then  $h(x_1, \dots, x_j) < A(t, m)$ . So  $h \in P$ .  $\square$

**Lemma 2.** *Let  $h$  be the primitive recursion of two functions  $f, g \in P$ . Then there exists a  $t \in \mathbb{N}$  such that for every  $x, x_1, \dots, x_n \in \mathbb{N}$   $h(x, x_1, \dots, x_n) < A(t, x + \max\{x_1, \dots, x_n\})$ .*

*Proof.* Since  $f, g \in P$ , there exist  $t_f$  and  $t_g$  so that  $f(x, y, x_1, \dots, x_n) < A(t_f, \max\{x, y, x_1, \dots, x_n\})$  and  $g(x_1, \dots, x_n) < A(t_g, \max\{x_1, \dots, x_n\})$  for any arguments of  $f$  and  $g$ . Let  $t = 1 + \max\{t_f, t_g\}$ . Note that  $t$  is not dependent on  $x, x_1, \dots, x_n$ . Let  $x, x_1, \dots, x_n \in \mathbb{N}$ . Let  $m = \max\{x_1, \dots, x_n\}$ . We proceed by induction.  $h(0, x_1, \dots, x_n) = g(x_1, \dots, x_n) < A(t_g, m) < A(t, m) = A(t, m + 0)$ . Now assume  $h(k, x_1, \dots, x_n) < A(t, \max\{m, k\})$  for some  $k \in \mathbb{N}$ . Let  $m_{k+1} = \max\{k, h(k, x_1, \dots, x_n), m\}$ . Then  $h(k + 1, x_1, \dots, x_n) = f(k, h(k, x_1, \dots, x_n), x_1, \dots, x_n) < A(t_f, m_{k+1})$ . Now note that  $m_{k+1} < A(t, k + m)$ :

1.  $m \leq k + m < A(t, k + m)$  by Theorem 1
2.  $k \leq k + m < A(t, k + m)$  by Theorem 1
3.  $h(k, x_1, \dots, x_n) < A(t, \max\{m, k\})$  by our induction hypothesis, and  $A(t, \max\{m, k\}) \leq A(t, k + m)$  by Theorem 4 since  $\max\{m, k\} \leq k + m$

By Theorem 4,  $A(t_f, m_{k+1}) < A(t_f, A(t, k + m))$ . Since  $t = 1 + \max\{t_f, t_g\}$ ,  $t_f \leq t - 1$ . Then by Theorem 4 and since  $t \neq 0$ ,  $A(t_f, A(t, k + m)) \leq A(t - 1, A(t, k + m))$ . By the definition of the Ackermann function,  $A(t - 1, A(t, k + m)) = A(t, k + m + 1)$ . Therefore  $h(k + 1, x_1, \dots, x_n) < A(t, (k + 1) + m)$   $\square$

**Theorem 11.**  *$P$  is closed under primitive recursion*

*Proof.* Let  $f, g \in P$  and  $h$  be the primitive recursion of  $f$  and  $g$ . Let  $x, x_1, \dots, x_n \in \mathbb{N}$ . Let  $m = \max\{x_1, \dots, x_n\}$ . Then by Lemma 2, there is a  $t \in \mathbb{N}$  so that  $h(x, x_1, \dots, x_n) < A(t, x + m)$ . Let  $m' = \max\{x, m\}$ . Since  $x + m \leq 2m'$ ,  $A(t, x + m) \leq A(t, 2m')$  by Lemma 4. Since  $2m' < 2m' + 3$ ,  $A(t, 2m') < A(t, 2m' + 3)$  also by Lemma 4. By Theorem 8,  $2m' + 3 = A(2, m')$ . Therefore  $A(t, 2m' + 3) = A(t, A(2, m'))$ . By Theorem 7,  $A(t, A(2, m')) < A(t + 2 + 2, m') = A(t + 4, m')$ . Therefore  $h(x, x_1, \dots, x_n) < A(t + 4, m')$ .  $\square$

**Theorem 12.**  *$P$  is precisely the primitive recursive functions*

*Proof.* All  $f \in P$  are primitive recursive by the definition of  $P$ . So it remains to show all primitive recursive functions are in  $P$ . Let  $f$  be a primitive recursive function. Then  $f$  can be created by applying function composition and primitive recursion to the constant, successor, and projection functions. The constant, successor, and projection functions are all in  $P$ , and  $P$  is closed under function composition and primitive recursion. Therefore  $f \in P$ .  $\square$

**Theorem 13.**  *$A(m, n)$  is not a primitive recursive function*

*Proof.* Suppose that  $A$  is a primitive recursive function. Then  $A \in P$ . Then there exists a  $t \in \mathbb{N}$  so that for any  $n, m \in \mathbb{N}$ ,  $A(n, m) < A(t, \max\{n, m\})$ . Set  $m = n = t$ . Then  $\max\{n, m\} = n = m = t$ . So  $A(n, m) = A(t, t) < A(t, t)$  which is a contradiction. Therefore  $A$  is not primitive recursive.  $\square$

## 4 General Recursive Functions

### 4.1 Partial Functions

### 4.2 Definition of General Recursive Functions

**Definition** (Minimization). *Let  $f : \mathbb{N}^n \rightarrow \mathbb{N}$  be an  $n$ -ary partial function. Then the minimization operator  $\mu$  creates a  $(n + 1)$ -ary function  $\mu(f)$  given by*

$$\mu(f)(x_1, \dots, x_n) = \begin{cases} z & f(z, x_1, \dots, x_n) = 0 \wedge f(i, x_1, \dots, x_n) > 0 \forall i \in \mathbb{N} \cap [0, z) \\ \text{undefined} & f(i, x_1, \dots, x_n) \neq 0 \forall i \in \mathbb{N} \end{cases}$$

**Definition.** *A function  $f$  is a general recursive function if it is either:*

1. *A primitive recursive function*
2. *A general recursive function under the minimization operator*