partial recursive function

Contents

Context

Constructivism, Realizability, Computability

Deduction and induction

Foundations

1. Idea

A ***partial recursive function*** (often "computable function", but see there for disambiguation) is a partial function of natural numbers which can be defined by an algorithm or computer program (e.g., a Turing machine), taking finitely many natural numbers as inputs, and which on input may run forever, but otherwise eventually halts and returns a natural number as output.

This idea as described is vague until it is circumscribed by a specific notion of computer program (Turing machines, register machines, abaci, etc.). There is a standard article of faith called the "Church-Turing thesis", identifying functions on natural numbers that are algorithmically computable with those that are computable using a Turing machine (or some variant class of machines that is Turing-complete).

A purely mathematical definition of the intended class of functions is given below.

2. Definition

Definition 2.1. A *partial recursive function* is a partial function from $\mathbb{N}^k$ to $\mathbb{N}$ (where $\mathbb{N}$ is the set of natural numbers and $k \geq 0$ is finite) that belongs to the smallest class $\mathcal{C}$ of partial functions that

- includes all constant functions $1 \to \mathbb{N}$;
- includes all projection maps $\pi_i \colon \mathbb{N}^k \to \mathbb{N}$, $i = 1, \ldots, k$;
- includes the successor function $s \colon \mathbb{N} \to \mathbb{N}$;
- is closed under composition: if $f_1 \colon \mathbb{N}^k \to \mathbb{N}, \ldots, f_n \colon \mathbb{N}^k \to \mathbb{N}$ and $g \colon \mathbb{N}^n \to \mathbb{N}$ belong to $\mathcal{C}$, then so does $g \circ (f_1, \ldots, f_n) \colon \mathbb{N}^k \to \mathbb{N}$;
- is closed under primitive recursion: if $g \colon \mathbb{N}^k \to \mathbb{N}$ and $h \colon \mathbb{N}^{k+2} \to \mathbb{N}$ belong to $\mathcal{C}$, then the function $f \colon \mathbb{N}^{k+1} \to \mathbb{N}$ defined recursively by the equations (for $y \in \mathbb{N}$ and $x \in \mathbb{N}^k$)

$$f(0, x) = g(x)$$

$$f(y + 1, x) = h(y, f(y, x), x)$$

  also belongs to $\mathcal{C}$;
- is closed under minimization: given any *total* function $f \colon \mathbb{N}^{k+1} \to \mathbb{N}$ belonging to $\mathcal{C}$, the partial function $g \colon \mathbb{N}^k \to \mathbb{N}$, defined by $g(x) = c$ iff $f(c, x) = 0$ and $f(d, x) > 0$ whenever $0 \leq d \leq c - 1$, also belongs to $\mathcal{C}$.

Definition 2.2. A *primitive recursive function* is a function that belongs to the smallest class of functions of the form $\mathbb{N}^k \to \mathbb{N}$ that contains constants, projection maps, the successor map, is closed under composition, and is closed under primitive recursion.

Clearly the primitive recursive functions are a subclass of partial recursive functions. Notice that primitive recursive functions are not merely partial functions, but actual (*total*) functions.

Remark 2.3. Both the class of partial recursive functions and the class of primitive recursive functions yield Lawvere theories Th(Comp) and Th(PrimRec), where a morphism $k \to 1$ of the Lawvere theory is a function $\mathbb{N}^k \to \mathbb{N}^1$ belonging to the class. It is straightforward to check that in either case, the generating object 1 of the Lawvere theory (or $\mathbb{N}$ if you prefer) is a parametrized NNO in that category: this is the essence of primitive recursion.

Definition 2.4. A *recursive relation* (or what is more usual nowadays, a *computable relation*) is a subset of $\mathbb{N}^k$ whose characteristic function $\chi\colon \mathbb{N}^k \to \{0, 1\}$ is recursive. Similarly, a primitive recursive relation is a relation whose characteristic function is primitive recursive.

3. Properties

We may build up a stock of functions and relations that are primitive recursive as follows. (All closure properties mentioned will clearly also apply to partial recursive functions and relations.)

1. Addition, multiplication, and exponentiation on $\mathbb{N}$. The factorial function $n \mapsto n!$ is primitive recursive. Binomial coefficients.
2. The predecessor function $\mathrm{Pred}\colon \mathbb{N} \to \mathbb{N}$, defined by the recursion $\mathrm{Pred}(0) = 0$, $\mathrm{Pred}(n + 1) = n$.
3. *Truncated subtraction*, where $x \dot{-} y = x - y$ if $x \geq y$, else 0, is primitive recursive. It is defined by the recursion $x \dot{-} 0 = x$, $x \dot{-} (y + 1) = \mathrm{Pred}(x \dot{-} y)$.
4. The distance function $|x - y| = (x \dot{-} y) + (y \dot{-} x)$. The function $\alpha(x) = 1 \dot{-} x$; this is the characteristic function of the unary relation or predicate “equals zero”. So “equals zero” is a computable relation. So is “doesn’t equal zero”, since this has characteristic function $\alpha(\alpha(x))$.
5. Therefore the equality relation $|x - y| = 0$ is a primitive recursive relation. So is $x < y$, being the relation “ $y \dot{-} x$ doesn’t equal zero”. If f is a (primitive) recursive function, then its graph defined by the relation $y = f(x)$ is a (primitive) recursive relation.
6. (Boolean combinations) Similarly, if R is a primitive recursive relation (so that its characteristic function χ_R is primitive recursive), then so is its negation $\neg R$, since $\chi_{\neg R} = \alpha(\chi_R)$. If P, Q are primitive recursive relations, so

is $P \wedge Q$, since $\chi_{P \wedge Q} = \chi_P \cdot \chi_Q$. Hence Boolean combinations of primitive recursive relations are primitive recursive.

7. If $f(x, y, z)$ and $h(y)$ are primitive recursive, so is

$$g(y, z) \coloneqq \sum_{x=0}^{h(y)} f(x, y, z)$$

and similarly with the sum replaced by a product. It follows that primitive recursive predicates are closed under *bounded quantification*; e.g., if $R(x, y, z)$ is a primitive recursive predicate and h is a primitive recursive function, then the predicate Q defined by

$$Q(y, z) = \forall_{0 \leq x < y} R(x, y, z) \coloneqq \bigwedge_{x=0}^{y-1} R(x, y, z)$$

is primitive recursive since $\chi_Q(y, z) = \prod_{x=0}^{y-1} \chi_R(x, y, z)$.

8. (Bounded least choice operator) If $R(x, y, z)$ is a primitive recursive relation and h is a primitive recursive function, then the function $f(y, z)$ defined to be "the least $x \leq h(y)$ such that $R(x, y, z)$ if such x exists, else y" is primitive recursive. Indeed,

$$f(y, z) = [\sum_{x=0}^{h(y)} (x \cdot \chi_R(x, y, z) \cdot (\prod_{w=0}^{x-1} \chi_{\neg R}(w, y, z))] + y \cdot \prod_{x=0}^{h(y)} \chi_{\neg R}(x, y, z)\,.$$

9. (Pairing and unpairing) There is an isomorphism $\mathbb{N}^2 \to \mathbb{N}$ in the Lawvere theory Th(PrimRec), i.e., there is a primitive recursive function $f\colon \mathbb{N}^2 \to \mathbb{N}$ with a primitive recursive inverse $g\colon \mathbb{N} \to \mathbb{N}^2$. For example, take f to be the function

$$f\colon (m, n) \mapsto \binom{m + n + 1}{2} + n$$

Its inverse g takes m to the pair

$$(a \dot{-} (m - \binom{a \dot{-} 1}{2} + 2), m - \binom{a \dot{-} 1}{2})$$

where a is the least element less than $m + 2$ such that $\binom{a}{2} > m$. By the aforementioned properties, this g is manifestly primitive recursive.

As a consequence of this last property, there exist primitive recursive isomorphisms between $\mathbb{N}$ and $\mathbb{N}^k$ for any $k > 0$. Since partial/primitive recursive functions are closed under composition, it is sufficient (and sometimes convenient) to consider only partial/primitive recursive functions on $\mathbb{N}$ itself. (The exception is the case $k = 0$, but this is trivial, since every partial function $1 \to \mathbb{N}$ is recursive.)

Remark 3.1. To show that $\mathbb{N}^2 \cong \mathbb{N}$ in the category of primitive recursive functions, it is not enough just to exhibit a primitive recursive bijection $f: \mathbb{N}^2 \to \mathbb{N}$, because it is not true that every primitive recursive bijection possesses a primitive recursive inverse. In other words, it is not true that the forgetful functor Th(PrimRec) → Set (see Remark 2.3) reflects isomorphisms. See Example 4.4 below.

Remark 3.2. Similarly, it is not always true that if a graph of a function is a primitive recursive relation, then the function itself is primitive recursive. (For example, the graph of the Ackermann function, discussed below, is a primitive recursive relation.) However, we do have a sample theorem as follows.

Theorem 3.3. *If a graph of a function f is a primitive recursive relation, and if $f \leq g$ for some primitive recursive g, then f is itself primitive recursive.*

The proof can be roughly expressed as follows: if $R(x, y)$ is the functional computable relation, then let $f(x)$ be the least $y \leq g(x)$ such that $R(x, y)$. The bounded least choice property shows that $f(x)$ is primitive recursive.

The point hovering in the background is that there are some computable functions which grow faster (in fact, much much much faster) than any primitive recursive function. This underscores the important role of the minimization axiom for partial recursive functions, which allows unboundedly large searches to take place. Indeed, we have the following crucial fact:

Theorem 3.4. *If $R \subseteq N^2$ is a graph of a function f and is a recursive relation, then f is a (total) recursive function. (The converse holds by one of the properties listed above.)*

Proof. According to the minimization axiom in Definition 2.1, given that $\chi_R: \mathbb{N}^2 \to \{0, 1\}$ is a recursive function, the function that takes x to the least $y \in \mathbb{N}$ such that $1 - \chi_R(x, y) = 0$ is also recursive. But this function is simply f. This completes the proof. ∎

Corollary. The inverse of a recursive bijection f is also recursive.

Proof. The graph of f is a recursive relation, and so is the opposite graph, which is the graph of the function f^{-1}. Apply the preceding theorem. ∎

Before passing on to general recursive functions, it is good to have some idea of the scope of primitive recursive functions. Some examples:

- The function $f(n) = p_n$, the n^{th} prime, is primitive recursive.

4. The Ackermann function

Definition 4.1. For each n, we define a function $A_n: \mathbb{N} \to \mathbb{N}$ by

- $A_0(m) = 2m$;
- $A_{n+1}(m) = (A_n)^m(1)$

where f^m denotes the composition of m copies of f. The ***Ackermann function*** $A: \mathbb{N} \to \mathbb{N}$ is defined by $A(m) = A_m(m)$.

(The function is named after Wilhelm Ackermann. There are several variations of this function around, and one common version actually puts $A_0(m) = m + 1$ and $A_{n+1}(m) = (A_n)^m(f(1))$.)

We show that while each A_n is primitive recursive, the function A grows faster than any primitive recursive function on $\mathbb{N}$, hence is not itself primitive recursive. It does however belong to the class of partial recursive functions.

By property 1 above, A_0 is primitive recursive. Supposing that A_n is primitive recursive, $\phi = A_{n+1}$ is also primitive recursive because it satisfies the recursion $\phi(0) = 1, \phi(m+1) = A_n(\phi(m))$.

By induction one may easily show $A_n(1) = 2$ for all n and $A_n(2) = 4$ for all n. We have $A_{n+1}(3) = A_n(A_n(A_n(1))) = A_n(A_n(2)) = A_n(4)$ for all n. The function A_1 gives n^{th} powers of 2, the function A_2 gives tetrations (iterated exponentials stacked n high) of 2, the function A_3 gives iterated tetrations, and so on.

Some routine inductions establish the following facts:

- For all n, the function A_n is strictly increasing, and with the sole exception of $A_0(0) = 0$, we have $m < A_n(m)$ for all m, i.e. A_n is strictly inflationary in arguments $m > 0$. We also have $n < A_n(3)$ for all n.

Lemma 4.2. *If $f\colon \mathbb{N}^k \to \mathbb{N}$ is primitive recursive, then there exists n such that $f(x_1, \ldots, x_k) \leq A_n(\max\{3, x_1, \ldots, x_k\})$ for all $x_1, \ldots, x_k$. (We say f is **dominated** by A_n, for short.)*

Proof. In the case where f is constant with value m, take $n = m$. For f the successor, use $n = 0$. For each projection $\pi_i\colon \mathbb{N}^k \to \mathbb{N}$, again use $n = 0$:

$$x_i = \pi_i(x_1, \ldots, x_k) \leq A_0(\max\{3, x_1, \ldots, x_k\}) .$$

Now proceed by induction on the construction of primitive recursive functions. Given that $f\colon \mathbb{N}^k$ is dominated by A_n and $g_1, \ldots, g_k\colon \mathbb{N}^m \to \mathbb{N}$ are dominated by A_{n+1}, we calculate that $h = f \circ (g_1, \ldots, g_k)\colon \mathbb{N}^m \to \mathbb{N}$ is dominated by A_{n+2}:

$$\begin{aligned} h(x_1, \ldots, x_m) &= f(g_1(x_1, \ldots, x_m), \ldots, g_k(x_1, \ldots, x_m)) \\ &\leq A_n(\max\{3, g_1(x_1, \ldots, x_m), \ldots, g_k(x_1, \ldots, x_m)\}) \\ &\leq A_n(A_{n+1}(\max\{3, x_1, \ldots, x_m\})) \\ &= A_{n+1}(\max\{3, x_1, \ldots, x_m\} + 1) \\ &\leq A_{n+2}(\max\{3, x_1, \ldots, x_m\}) . \end{aligned}$$

And given that $g\colon \mathbb{N}^k \to \mathbb{N}$ is dominated by A_{n+1} and $h\colon \mathbb{N}^{k+2} \to \mathbb{N}$ is dominated by A_n, if we define $f\colon \mathbb{N}^{k+1} \to \mathbb{N}$ by recursion by $f(0, x_1, \ldots, x_k) = g(x_1, \ldots, x_k)$ and $f(y+1, x_1, \ldots, x_k) = h(y, f(y, x_1, \ldots, x_k), x_1, \ldots, x_k)$, we calculate that f is dominated by A_{n+3}, in two steps. First we claim

$$f(y, x_1, \ldots, x_k) \leq A_{n+1}(y + \max\{3, x_1, \ldots, x_k\})\,.$$

Indeed, this is true by assumption for $y = 0$. And then

$$\begin{aligned} f(y+1, x_1, \ldots, x_k) &= h(y, f(y, x_1, \ldots, x_k), x_1, \ldots, x_k) \\ &\leq A_n(\max\{3, y, f(y, x_1, \ldots, x_k), x_1, \ldots, x_k)\}) \\ &\leq A_n(\max\{3, y, A_{n+1}(y + \max\{3, x_1, \ldots, x_k\}), x_1, \ldots, x_k\}) \\ &\leq A_n(A_{n+1}(y + \max\{3, x_1, \ldots, x_k\})) \\ &= A_{n+1}(y + 1 + \max\{3, x_1, \ldots, x_k\}) \end{aligned}$$

which justifies the claim. Finally, we have

$$\begin{aligned} f(y, x_1, \ldots, x_k) &\leq A_{n+1}(y + \max\{3, x_1, \ldots, x_k\}) \\ &\leq A_{n+1}(2\max\{3, y, x_1, \ldots, x_k\}) \\ &\leq A_{n+1}(A_{n+2}(\max\{3, y, x_1, \ldots, x_k\})) \\ &= A_{n+2}(\max\{3, y, x_1, \ldots, x_k\} + 1) \\ &\leq A_{n+3}(\max\{y, x_1, \ldots, x_k\}) \end{aligned}$$

so that f is dominated by A_{n+3}. This completes the proof. ∎

Corollary. The Ackermann function A is not primitive recursive.

Proof. If $A\colon x \mapsto A_x(x)$ were recursive, then so would be the function $\phi\colon x \mapsto A_x(x) + 1$. In that case, ϕ is dominated by A_n for some $n \geq 3$. We then arrive at the contradiction

$$A_n(n) + 1 = \phi(n) \leq A_n(\max\{3, n\}) = A_n(n)\,.$$

■

Proposition 4.3. *The graph of the Ackermann function is a primitive recursive relation.*

Proof. The rough idea is to let z bound the search for solutions (x, y) to $A_x(y) = z$. For $x, y > 0$ we have $A_x(y) = A_{x-1}(A_x(y-1)) = A_x(y')$ where $y' \coloneqq A_x(y-1)$. We have $y' < A_{x-1}(y') = z$. Starting with $y_0 = y < z$, iterate this procedure so that

$$z = A_x(y_0) = A_{x-1}(y_1) = A_{x-2}(y_2) = \ldots = A_0(y_x) = 2y_x$$

with each y_i less than z. (Explicitly, the iteration is $y_{i+1} \coloneqq A_{x-i}(y_i - 1)$.)

Thus, after disposing of some trivial low number cases, WLOG we may take $z > 4$, where the ternary predicate

$$(z > 4) \wedge (A_x(y) = z)$$

is equivalent to

$$(x > 0) \wedge (y > 2) \wedge (x < z) \wedge \exists_{y_0, y_1, \ldots y_x < z} (y = y_0) \wedge (\forall_{i < x} y_{i+1} = A_{x-i}(y_i - 1)) \wedge (2y_x = z)$$

where the quantifications are manifestly bounded by z. This shows that the ternary predicate $A_x(y) = z$ is primitive recursive.

From this it follows that the binary relation $A_x(x) = z$ is also primitive recursive, as claimed. ■

Combining this result with Theorem 3.4, we have

Corollary. The Ackermann function $A: x \mapsto A_x(x)$ is recursive.

Example 4.4. Here we exhibit a primitive recursive bijection whose inverse is not primitive recursive. The graph $y = A_x(x)$ of the Ackermann function is primitive recursive binary predicate, and the image $I = \{y: \exists_x y = A_x(x)\}$ is a primitive recursive unary predicate, because the existential quantifier is a bounded quantifier applied to a primitive recursive relation:

$$I = \bigvee_{x=0}^{y-1} [A_x(x) = y] .$$

Observe that both I and its complement $\neg I$ in $\mathbb{N}$ are infinite.

For any infinite set $J \subseteq \mathbb{N}$, let $p_J : \mathbb{N} \to \mathbb{N}$ be the function taking n to the n^{th} element of J (with respect to the usual order $<$). For example, $p_I(x) = A_x(x)$. Now define a function $f : \mathbb{N} \to \mathbb{N}$ by

$$\begin{aligned} f(m) &= 2p_I^{-1}(m) && \text{if} m \in I \\ &= 2p_{\neg I}^{-1}(m) + 1 && \text{if} m \in \neg I \end{aligned}$$

Then $f(m) \leq 2m + 1$ and the graph of f is primitive recursive, so by Theorem 3.3, f is a primitive recursive function. It is a bijection by construction. But f^{-1} is not primitive recursive, because $f^{-1}(2x) = p_I(x) = A(x)$, and A is not primitive recursive.

5. Busy Beaver function

6. Related concepts

- recursive subset

computability

	type I computability	**type II computability**
typical domain	natural numbers $\mathbb{N}$	Baire space of infinite sequences $\mathbb{B} = \mathbb{N}^{\mathbb{N}}$
computable functions	partial recursive function	computable function (analysis)

	type I computability	**type II computability**
type of computable mathematics	recursive mathematics	computable analysis, Type Two Theory of Effectivity
type of realizability	number realizability	function realizability
partial combinatory algebra	Kleene's first partial combinatory algebra	Kleene's second partial combinatory algebra

7. References

A standard textbook in recursive functions is

- Hartley Rogers, Jr., *Theory of recursive functions and effective computability*, McGraw-Hill, 1967.

Quite a bit of the arrangement and proofs above were taken from clear and useful lecture notes of Stephen Simpson,

- Stephen G. Simpson, *Foundations of Mathematics* (Lecture Notes), October 1, 2009. (pdf)

Of course, there is always good old Wikipedia:

- Wikipedia, *Computable function*

Last revised on May 9, 2022 at 20:42:36. See the history of this page for a list of all contributions to it.

Edit Discuss Previous revision Changes from previous revision History (20 revisions) Cite Print Source