Computer Science 513

Modelling Computation of Functions II

Instructor: Wayne Eberly

Department of Computer Science
University of Calgary

Lecture #3

Goals for Today

Goals for Today:

- Introduce additional techniques that can be used to prove that functions are primitive recursive
- Introduce a larger class of recursively defined total and partial functions — which are more reasonable candidates for "computable functions"

Review: Primitive Recursive Functions

Recall that a function is ***primitive recursive*** if it can be obtained from the initial functions using a finite number of applications of composition and primitive recursion:

(a) Each of the *initial functions,* including the nullify, successor and projection functions, is a primitive recursive function.

(b) If $k, n \geq 1$, $f : \mathbb{N}^k \to \mathbb{N}$ is primitive recursive, and $g_i : \mathbb{N}^n \to \mathbb{N}$ is primitive recursive for $1 \leq i \leq k$, then the function $h : \mathbb{N}^n \to \mathbb{N}$ such that, for $x_{1,2}, \ldots, x_n \in \mathbb{N}$,

$$\begin{aligned} h(x_1, x_2, \ldots, x_n) = f(&g_1(x_1, x_2, \ldots, x_n), \\ &g_2(x_1, x_2, \ldots, x_n), \ldots, g_k(x_1, x_2, \ldots, x_n)) \end{aligned}$$

— obtained by *composition* — is primitive recursive.

Review: Primitive Recursive Functions

(c) If k is a fixed nonnegative integer and $g : \mathbb{N}^2 \to \mathbb{N}$ is a primitive recursive function then the function $h : \mathbb{N} \to \mathbb{N}$ such that

- $h(0) = k$ and,
- $h(t+1) = g(t, h(t))$ for every integer $t \geq 0$

is also primitive recursive.

(d) If n is a positive integer, $f : \mathbb{N}^n \to \mathbb{N}$ and $g : \mathbb{N}^{n+2} \to \mathbb{N}$ are both primitive recursive functions, then the function $h : \mathbb{N}^{n+1} \to \mathbb{N}$ such that, for all $x_1, x_2, \ldots, x_n \in \mathbb{N}$,

- $h(x_1, x_2, \ldots, x_n, 0) = f(x_1, x_2, \ldots, x_n)$, and
- $h(x_1, x_2, \ldots, x_n, t+1) =$
$$g(t, h(x_1, x_2, \ldots, x_n, t), x_1, x_2, \ldots, x_n)$$
for every integer $t \geq 0$

is primitive recursive too.

The last two functions, listed above, have been obtained using *primitive recursion*.

Construction: Iterated Sums

Claim #1: Let $n \in \mathbb{N}$ and let $\varphi : \mathbb{N}^{n+1} \to \mathbb{N}$ be primitive recursive. Then the function $S_\varphi : \mathbb{N}^{n+1} \to \mathbb{N}$ such that, for all $x_1, x_2, \ldots, x_{n+1}$,

$$S_\varphi(x_1, x_2, \ldots, x_n, x_{n+1}) = \sum_{i=0}^{x_{n+1}} \varphi(x_1, x_2, \ldots, x_n, i)$$

is primitive recursive as well.

A sketch of a proof (whose completion is left as an exercise) of this, and several later claims, is included in a supplemental document for this lecture.

Construction: Iterated Sums

Claim #1, and its proof, can be modified to establish the following as well.

- If the function $\varphi : \mathbb{N} \to \mathbb{N}$ is primitive recursive then the function $S_\varphi : \mathbb{N} \to \mathbb{N}$ such that

$$S_\varphi(x_1) = \sum_{i=0}^{x_1} \varphi(i),$$

for all $x_1 \in \mathbb{N}$, is primitive recursive as well.

Construction: Iterated Sums

Claim #1, and its proof, can also be modified to establish the following as well.

- If $n \in \mathbb{N}$ and the function $\varphi : \mathbb{N}^{n+1} \to \mathbb{N}$ is primitive recursive then the function $S_\varphi : \mathbb{N}^{n+1} \to \mathbb{N}$ such that

$$S_\varphi(x_1, x_2, \ldots, x_{n+1}) = \sum_{i=1}^{x_{n+1}} \varphi(x_1, x_2, \ldots, x_n, i),$$

  for all $x_1, x_2, \ldots, x_{n+1} \in \mathbb{N}$, is primitive recursive as well.

Note: The difference with S_φ is that the lower index for the summation is "$i = 1$" instead of "$i = 0$".

Application: A "Quotient" Function

Example: Consider the ***quotient*** function $quo : \mathbb{N}^2 \to \mathbb{N}$ such that, for all $x_1, x_2 \in \mathbb{N}$,

$$quo(x_1, x_2) = \begin{cases} \left\lfloor \frac{x_1}{x_2} \right\rfloor & \text{if } x_2 \geq 1, \\ x_1 & \text{if } x_2 = 0. \end{cases}$$

- Recall, from the previous lecture, that the ***remainder*** function $rem : \mathbb{N}^2 \to \mathbb{N}$ is primitive recursive where, for $x_1, x_2 \in \mathbb{N}$,

$$rem(x_1, x_2) = \begin{cases} x_1 \bmod x_2 & \text{if } x_2 \geq 1, \\ 0 & \text{if } x_2 = 0. \end{cases}$$

Application: A "Quotient" Function

- Since the constant functions $c_0 : \mathbb{N} \to \mathbb{N}$ and $c_1 : \mathbb{N} \to \mathbb{N}$ such that $c_0(x_1) = 0$ and $c_1(x_1) = 1$ for all $x_1 \in \mathbb{N}$ are primitive recursive, as is the predicate, "$x > 0$", and the projections u_1^2 and u_2^2, one can show using composition and "definition by cases" that the function $\varphi : \mathbb{N}^2 \to \mathbb{N}$ such that, for all $x_1, x_2 \in \mathbb{N}$,

$$\varphi(x_1, x_2) = \begin{cases} 1 & \text{if } rem(x_2, x_1) = 0, \\ 0 & \text{if } rem(x_2, x_1) > 0 \end{cases}$$

  is primitive recursive as well.

Application: A "Quotient" Function

- It follows that the (second) iterated sum

$$\widehat{S}_\varphi = \sum_{i=1}^{x_2} \varphi(x_1, i)$$

is primitive recursive as well — and it is reasonably easy to show, by induction on x_1, that $\widehat{S}_\varphi(x_1, x_2) = quo(x_1, x_2)$ for all $x_1, x_2 \in \mathbb{N}$ — so that the quotient function is primitive recursive, as claimed.

Construction: Iterated Products

Claim #2: Let $n \in \mathbb{N}$ and let the function $\varphi : \mathbb{N}^{n+1} \to \mathbb{N}$ be primitive recursive.

(a) The iterated product $P_\varphi : \mathbb{N}^{n+1} \to \mathbb{N}$ such that, for all $x_1, x_2, \ldots, x_{n+1} \in \mathbb{N}$,

$$P_\varphi(x_1, x_2 \ldots, x_n, x_{n+1}) = \prod_{i=0}^{x_{n+1}} \varphi(x_1, x_2, \ldots, x_n, i)$$

is primitive recursive.

(b) The (modified) inner product $\widehat{P}_\varphi : \mathbb{N}^{n+1} \to \mathbb{N}$ such that, for all $x_1, x_2, \ldots, x_{n+1} \in \mathbb{N}$,

$$\widehat{P}_\varphi(x_1, x_2, \ldots, x_n, x_{n+1}) \prod_{i=1}^{x_{n+1}} \varphi(x_1, x_2, \ldots, x_n, i)$$

is primitive recursive as well.

Construction: Bounded "Universal" Quantifiers

Let $n \in \mathbb{N}$ and let $p : \mathbb{N}^{n+1} \to \{0, 1\}$ be a predicate. ***Bounded quantifiers*** can be used to produce additional predicates:

- $\underset{0 \le i \le x_{n+1}}{\forall} p$ is a function from $\mathbb{N}^{n+1}$ to $\{0, 1\}$. For all $x_1, x_2, \ldots, x_{n+1} \in \mathbb{N}$,

$$\underset{0 \le i \le x_{n+1}}{\forall} p(x_1, x_2, \ldots, x_n, i) = 1$$

  if and only if $p(x_1, x_2, \ldots, x_n, i) = 1$ for every integer i such that $0 \le i \le x_{n+1}$.

Construction: Bounded "Universal" Quantifiers

- $\underset{1 \leq i \leq x_{n+1}}{\forall} p$ is also a function from $\mathbb{N}^{n+1}$ to $\{0, 1\}$. For all $x_1, x_2, \dots, x_{n+1} \in \mathbb{N}$,

  $$\underset{1 \leq i \leq x_{i+1}}{\forall} p(x_1, x_2, \dots, x_n, i) = 1$$

  if and only if $p(x_1, x_2, \dots, x_n, i) = 1$ for every integer i such that $1 \leq i \leq x_{n+1}$.

Note: By definition,

$$\underset{1 \leq i \leq x_{n+1}}{\forall} p(x_1, x_2, \dots, x_n, i)$$

is equal to 1, for all predicates p and for all $x_1, x_2, \dots, x_n \in \mathbb{N}$, whenever $x_{n+1} = 0$.

Construction: Bounded "Existential" Quantifiers

- $\underset{0\le i\le x_{n+1}}{\exists} p$ is a function from $\mathbb{N}^{n+1}$ to $\{0,1\}$. For all $x_1, x_2, \ldots, x_{n+1} \in \mathbb{N}$,

  $$\underset{0\le i\le x_{n+1}}{\exists} p(x_1, x_2, \ldots, x_n, i) = 1$$

  if and only if $p(x_1, x_2, \ldots, x_n, i) = 1$ for at least one integer i such that $0 \le i \le x_{n+1}$.

Bounded "Existential" Quantifiers

- $\underset{1 \leq i \leq x_{n+1}}{\exists} p$ is a function from $\mathbb{N}^{n+1}$ to $\{0, 1\}$. For all $x_1, x_2, \ldots, x_{n+1} \in \mathbb{N}$,

  $$\underset{1 \leq i \leq x_{n+1}}{\exists} p(x_1, x_2, \ldots, x_n, i) = 1$$

  if and only if $p(x_1, x_2, \ldots, x_n, i) = 1$ for at least one integer i such that $1 \leq i \leq x_{n+1}$.

Note: By definition,

$$\underset{1 \leq i \leq x_{n+1}}{\exists} p(x_1, x_2, \ldots, x_n, i)$$

is equal to 0, for all predicates p and for all $x_1, x_2, \ldots, x_n \in \mathbb{N}$, whenever $x_{n+1} = 0$.

Constructions: Bounded Quantifiers

Claim #3: Let $n \in \mathbb{N}$ and let $p : \mathbb{N}^{n+1} \to \{0, 1\}$. If p is primitive recursive then each of the predicates

$$\underset{0 \le i \le x_{n+1}}{\forall} p, \quad \underset{1 \le i \le x_{n+1}}{\forall} p, \quad \underset{0 \le i \le x_{n+1}}{\exists} p \quad \text{and} \quad \underset{1 \le i \le x_{n+1}}{\exists} p$$

is a primitive recursive predicate as well.

Application: Primality Testing

Recall that a number $n \in \mathbb{N}$ is ***prime*** if $n \geq 2$ and there does not exist any integer k such that $2 \leq k \leq n - 1$ and n is divisible by k.

- Note, as well, that if $k \geq 2$ then n is divisible by k if and only if $n \bmod k = 0$.
- Now consider the predicate $p : \mathbb{N}^2 \to \{0, 1\}$ such that, for all $x_1, x_2 \in \mathbb{N}$,

$$p(x_1, x_2) = (x_2 < 2) \vee (x_2 \geq x_1) \vee (x_1 \bmod x_2 > 0).$$

Exercise: Confirm that this is a primitive recursive predicate. The examples and techniques introduced in the previous lecture should be all you need.

Application: Primality Testing

- It follows that the predicate $\varphi : \mathbb{N}^2 \to \{0,1\}$ such that, for all $x_1, x_2 \in \mathbb{N}$,

$$\begin{aligned}\varphi(x_1, x_2) &= \underset{0 \leq i \leq x_2}{\forall} p(x_1, i) \\ &= \underset{0 \leq i \leq x_2}{\forall} ((i < 2) \vee (i \geq x_1) \vee (x_1 \mod i > 0))\end{aligned}$$

  is primitive recursive as well. Note that if $x_2 \geq x_1$ then this is equivalent to the claim that “there is no integer between 2 and $x_1 - 1$ (inclusive) that is divisible by x_1”.

- Consider the predicate *prime* : $\mathbb{N} \to \{0,1\}$ such that, for all $x_1 \in \mathbb{N}$,

$$\mathit{prime}(x_1) = (x_1 \geq 2) \wedge \varphi(x_1, x_1).$$

  This is a primitive recursive predicate such that, for all $x_1 \in \mathbb{N}$, $\mathit{prime}(x_1) = 1$ if and only if x_1 is prime.

Application: Primality Testing

- It follows that the predicate $\varphi : \mathbb{N}^2 \to \{0, 1\}$ such that, for all $x_1, x_2 \in \mathbb{N}$,

$$\begin{aligned}\varphi(x_1, x_2) &= \underset{0 \leq i \leq x_2}{\forall} p(x_1, i) \\ &= \underset{0 \leq i \leq x_2}{\forall} ((i < 2) \vee (i \geq x_1) \vee (x_1 \bmod i > 0))\end{aligned}$$

  is primitive recursive as well. Note that if $x_2 \geq x_1$ then this is equivalent to the claim that "there is no integer between 2 and $x_1 - 1$ (inclusive) that is divisible by x_1".

- Consider the predicate *prime* : $\mathbb{N} \to \{0, 1\}$ such that, for all $x_1 \in \mathbb{N}$,

$$\mathit{prime}(x_1) = (x_1 \geq 2) \wedge \varphi(x_1, x_1).$$

  This is a primitive recursive predicate such that, for all $x_1 \in \mathbb{N}$, $\mathit{prime}(x_1) = 1$ if and only if x_1 is prime.

Construction: Bounded Minimization

Let $n \in \mathbb{N}$ and let $p : \mathbb{N}^{n+1} \to \{0, 1\}$. Consider the function

$$\underset{0 \leq i \leq x_{n+1}}{\mathit{first}} \; p : \mathbb{N}^{n+1} \to \mathbb{N}$$

such that, for all $x_1, x_2, \ldots, x_n, x_{n+1} \in \mathbb{N}$, the following properties are satisfied.

- If there exist an integer i such $0 \leq i \leq x_{n+1}$ and $p(x_1, x_2, \ldots, x_n, i) = 1$, then $\underset{0 \leq i \leq x_{n+1}}{\mathit{first}} \; p(x_1, x_2, \ldots, x_n, i)$ is the smallest (that is, the ***first***) integer i for which this is true.
- On the other hand, if $p(x_1, x_2, \ldots, x_n, i) = 0$ for every integer i such that $0 \leq i \leq x_{n+1}$, then

$$\underset{0 \leq i \leq x_{n+1}}{\mathit{first}} \; p(x_1, x_2, \ldots, x_n, i) = 0.$$

This function is said to be obtained from the predicate p using ***bounded minimization***.

Construction: Bounded Minimization

Claim #4: Let $n \in \mathbb{N}$ and let $p : \mathbb{N}^{n+1} \to \{0, 1\}$. If the predicate p is primitive recursive then the function

$$\underset{0 \le i \le x_{n+1}}{\mathit{first}}\ p : \mathbb{N}^{n+1} \to \mathbb{N}$$

is primitive recursive as well.

Application: Computing the n^{th} Prime Number

Let $p_0 = 0$ and, for every positive integer n, let p_n be the n^{th} prime — so that

$$p_0 = 0,\ p_1 = 2,\ p_2 = 3,\ p_3 = 5,\ p_4 = 7,\ p_5 = 11,$$

and so on. Consider the function *prime* : $\mathbb{N} \to \mathbb{N}$ such that, for all $n \in \mathbb{N}$, $\mathit{prime}(n) = p_n$, the n^{th} prime number.

- It will be helpful to find an *upper bound* for p_n in order to show that this function is primitive recursive.

Application: Computing the n^{th} Prime Number

Claim: If $n \geq 1$ then $p_{n+1} \leq B$, where

$$B = \left(\prod_{i=1}^{n} p_i \right) + 1.$$

Proof: Notice that $B \geq 3$ and that $B \bmod p_i = 1$ for every integer i such that $1 \leq i \leq n$ — so that B is not divisible by any of the first n primes. B is therefore an upper bound for p_{n+1} because *the smallest prime divisor* of B is a prime that is greater than or equal to p_{n+1}. $\square$

Application: Computing the n^{th} Prime Number

Corollary: If $n \geq 1$ then $p_{n+1} \leq 1 + p_n!$.

Proof: This follows from the previous result because (it is easily checked that) $B \leq 1 + p_n!$. □

- Now consider the predicate $p : \mathbb{N}^2 \to \mathbb{N}$ such that, for all $y, b \in \mathbb{N}$,

$$p(y, b) = \begin{cases} 1 & \text{if } (b > y) \wedge \mathit{prime}(b), \\ 0 & \text{otherwise.} \end{cases}$$

  Since the predicate *prime* is primitive recursive, this predicate is primitive recursive too.

Application: Computing the n^{th} Prime Number

- Next let $\varphi : \mathbb{N}^2 \to \mathbb{N}$ be the function such that, for all $y, b \in \mathbb{N}$,

$$\begin{aligned} \varphi(y, b) &= \mathop{\mathit{first}}_{0 \le i \le b} p(y, i). \\ &= \mathop{\mathit{first}}_{0 \le i \le b} ((i > y) \wedge \mathit{prime}(i)). \end{aligned}$$

  Since this function was obtained from p using bounded minimization, it is a primitive recursive function.

- Note, as well, that the value of this function is 0 if there are no primes between $y + 1$ and b (inclusive). Otherwise $\varphi(y, b)$ is the *smallest* prime that is greater than y.
- It follows by the previous results that if $n \ge 1$, $y = p_n$ and $b = 1 + y! = 1 + p_n!$ then $\varphi(y, b) = p_{n+1}$.

Application: Computing the n^{th} Prime Number

Exercises:

(a) Prove that the function $F : \mathbb{N} \to \mathbb{N}$ such that $F(0) = 1$ and, for $n \geq 1$, $F(n) = n!$, is primitive recursive.

(b) Prove that the function $g : \mathbb{N}^2 \to \mathbb{N}$ such that, for $t, y \in \mathbb{N}$,

$$g(t, y) = \begin{cases} 2 & \text{if } t = 0 \\ \varphi(y, 1 + F(y)) & \text{otherwise} \end{cases}$$

is primitive recursive too.

Application: Computing the n^{th} Prime Number

Finally consider the function $h : \mathbb{N} \to \mathbb{N}$ that is obtained from g using ***primitive recursion***, using the constant $k = 0$ to define $h(0)$. This function is primitive recursive because g is.

- $h(0) = 0 = p_0$.
- For every integer t such that $t \geq 0$,

$$
\begin{aligned}
h(t+1) &= g(t, h(t)) \\
&= \begin{cases} 2 & \text{if } t = 0, \\ \varphi(h(t), 1 + F(h(t))) & \text{otherwise.} \end{cases}
\end{aligned}
$$

Application: Computing the n^{th} Prime Number

- Notice that it follows that $h(1) = 2 = p_1$ — and that $h(t+1) = \varphi(h(t), 1 + h(t)!)$ whenever $t \geq 1$.
- Using the above information, it is now possible to prove by induction on n that $h(n) = p_n$ for every integer $n \geq 0$.
- Thus this function, which maps every non-negative integer n to p_n, is primitive recursive.

Moving Beyond the Set of Primitive Recursive Functions

- Note that all primitive recursive are ***total functions***.
- The addition of the next construction — which *is not* guaranteed to generate primitive recursive functions, from primitive recursive functions, gives a way to include *partial* functions, which are not defined everywhere, in another interesting set of partial and total functions.

Unbounded Minimization

Now let $n \in \mathbb{N}$ such that $n \geq 1$, and let $p : \mathbb{N}^{n+1} \to \{0, 1\}$ be a ***partial*** predicate— so that $p(x_1, x_2, \ldots, x_n, x_{n+1})$ is not necessarily defined, for all $x_1, x_2, \ldots, x_{n+1} \in \mathbb{N}$.

Definition: The function

$$\mu x\, p(x_1, x_2, \ldots, x_n, x) : \mathbb{N}^n \to \mathbb{N}$$

is a partial function which satisfies the following properties, for all $x_1, x_2, \ldots, x_n \in \mathbb{N}$.

- If there exists an integer i such that $i \geq 0$, $p(x_1, x_2, \ldots, x_n, j)$ is defined *for every* integer j such that $0 \leq j \leq i$, and $p(x_1, x_2, \ldots, x_n, i) = 1$, then

  $$\mu x\, p(x_1, x_2, \ldots, x_n, x)$$

  is the *smallest* integer i such that these properties are satisfied.

Unbounded Minimization

- If there does not exist any integer i such that $i \geq 0$, $p(x_1, x_2, \ldots, x_n, j)$ is defined for every integer j such that $0 \leq j \leq i$, and $p(x_1, x_2, \ldots, x_n, 1) = 1$, then

$$\mu x \, p(x_1, x_2, \ldots, x_n, x)$$

is ***undefined***.

Thus $\mu x \, p(x_1, x_2, \ldots, x_n, x)$ is undefined if there exists an integer i such that $i \geq 0$, $p(x_1, x_2, \ldots, x_n, j) = 0$ for every integer j such that $0 \leq j \leq i - 1$, and $p(x_1, x_2, \ldots, x_n, i)$ is undefined.

$\mu x \, p(x_1, x_2, \ldots, x_n, x)$ is also undefined if $p(x_1, x_2, \ldots, x_n, i) = 0$ for all $i \in \mathbb{N}$.

μ-Recursive Functions
Recursive Definition

Definition: A partial or total function is ***μ-recursive*** if it can be obtained from the initial functions using a finite number of applications of composition, primitive recursion, and unbounded minimization:

(a) Each of the *initial functions* is a μ-recursive function:
   (i) The nullify function, $n : \mathbb{N} \to \mathbb{N}$, is a μ-recursive function.
   (ii) The successor function, $s : \mathbb{N} \to \mathbb{N}$, is a μ-recursive function.
   (iii) For all integers i and k such that $1 \leq i \leq k$, the projection function u_i^k is a μ-recursive function.

μ-Recursive Functions
Recursive Definition

(b) If $k, n \geq 1$, $f : \mathbb{N}^k \to \mathbb{N}$ is μ-recursive, and $g_i : \mathbb{N}^n \to \mathbb{N}$ is μ-recursive for $1 \leq i \leq k$, then the function $h : \mathbb{N}^n \to \mathbb{N}$ such that, for $x_{1,2}, \ldots, x_n \in \mathbb{N}$,

$$h(x_1, x_2, \ldots, x_n) = f(g_1(x_1, x_2, \ldots, x_n), g_2(x_1, x_2, \ldots, x_n), \ldots, g_k(x_1, x_2, \ldots, x_n))$$

— obtained by *composition* — is μ-recursive. (Note that this function is undefined if any of $g_i(x_1, x_2, \ldots, x_n)$ is undefined, for $1 \leq i \leq k$).

μ-Recursive Functions
Recursive Definition

(c) If k is a fixed nonnegative integer and $g : \mathbb{N}^2 \to \mathbb{N}$ is a μ-recursive partial or total function then the (partial or total) function $h : \mathbb{N} \to \mathbb{N}$ such that

- $h(0) = k$ and,
- $h(t + 1) = g(t, h(t))$ for every integer $t \geq 0$

— obtained using the first kind of *primitive recursion* — is also μ-recursive. (Note that $h(t + 1)$ is undefined, here, if $h(t)$ is undefined.)

μ-Recursive Functions
Recursive Definition

(d) If n is a positive integer, $f : \mathbb{N}^n \to \mathbb{N}$ and $g : \mathbb{N}^{n+2} \to \mathbb{N}$ are both μ-recursive partial or total functions, then the (partial or total) function $h : \mathbb{N}^{n+1} \to \mathbb{N}$ such that, for all $x_1, x_2, \dots, x_n \in \mathbb{N}$,

- $h(x_1, x_2, \dots, x_n, 0) = f(x_1, x_2, \dots, x_n)$, and
- $h(x_1, x_2, \dots, x_n, t+1) =$
$$g(t, h(x_1, x_2, \dots, x_n, t), x_1, x_2, \dots, x_n)$$
  for every integer $t \geq 0$

— obtained using the second kind of *primitive recursion* — is μ-recursive too. (Note that $h(x_1, x_2, \dots, x_n, t+1)$ is undefined if $h(x_1, x_2, \dots, x_n, t)$ is undefined.)

μ-Recursive Functions
Recursive Definition

(e) If n is a positive integer, and $p : \mathbb{N}^{n+1} \to \{0, 1\}$ is a μ-recursive (partial or total) predicate, the (partial or total) function

$$\mu x \, p(x_1, x_2, \ldots, x_n, x) : \mathbb{N} \to \mathbb{N}$$

— obtained from p by *unbounded minimization* — is μ-recursive.

μ-Recursive Functions

- One can show (after comparing definitions) that every ***primitive recursive*** function is also ***μ-recursive***.
- It is *not* obvious, or easy to prove, but: There exist ***μ-recursive*** total functions that *are not* ***primitive recursive***.
- This was surprising to quite a few researchers when this was discovered! The *primitive recursive* functions were, originally, just called "*recursive* functions", because it was originally believed that all total μ-recursive functions were primitive recursive.
- These functions will be considered again, after a few more technical tools are available.

The λ Calculus

- The λ ***Calculus*** was also proposed, in the 1930's, as a way to model "effective computability".
- This has applications in computer science, notably including in ***functional programming***.
- While this will not be discussed, in any detail, in CPSC 513, a supplemental document provides a bit more information about this, for interested students.