Université de Paris **(Date: 23/09/2020)**
UFR Mathématiques

Computability and Incompleteness TD1

Professor: P. Rozière Juan Ignacio Padilla, M2 LMFI

Exercise 1. Show that the set of primitive recursive functions is countable.
Solution: We can define by induction

$$\mathcal{F}_0 = \{\lambda x.0, \lambda x.s(x)\} \cup \{p_k^i, 1 \leq i \leq k\}_{k\in\mathbb{N}}$$
$$\mathcal{F}_{n+1} = \{f \in \mathbb{N}^{\mathbb{N}^k}, \exists g, h \in \mathcal{F}_n, f = \mathrm{Rec}(g,h)\}_{n\in\mathbb{N}}$$
$$\cup\{f \in \mathbb{N}^{\mathbb{N}^k}, \exists g_1, \ldots, g_m, h \in \mathcal{F}_n, f \equiv h(g_1, \ldots, g_m)\}_{k\in\mathbb{N}}$$

Each $\mathcal{F}_n$ is countable since the operations Rec and composition only require finitely many arguments. We have that the set of primitive recursive functions is $\mathcal{F} = \bigcup_n \mathcal{F}_n$, and consequently it is countable.

Exercise 2. (examples, special cases of the primitive recursion scheme).

(1) Show that constant functions are primitive recursive. By induction: the function $\lambda x.1$ equals the composition of $s(x)$ and the zero function. Now, if $f(x) = \lambda x.k$ is primitive recursive, then $\lambda x.k+1 = s(f(x))$, which is primitive recursive by the composition scheme.
(2) Show that $x \mapsto x+2$, $x \mapsto 2x$ and $x \mapsto 2x+1$ are primitive recursive. $f(x) = \lambda x.x + 2 = s(s(p_1^1(x)))$, the doubling function is defined by primitive recursion as $g(0) = 0$ and $g(x+1) = f(p_2^1(x, g(x))$. Finally $h(x) = s(g(x))$.
(3) Show that addition, multiplication and exponentiation are primitive recursive functions.

$$+(x,0) = x = p_1^1(x)$$
$$+(x,y+1) = s(p_3^3(x,y,+(x,y)))$$
$$\times(x,0) = 0$$
$$\times(x,y+1) = +(p_3^3(x,y,\times(x,y)), p_3^1(x,y,\times(x,y))$$
$$\exp(x,0) = 1$$
$$\exp(x,y+1) = \times(p_3^3(x,y,\exp(x,y)), p_3^1(x,y,\exp(x,y))$$

(4) Show that the function sg which maps 0 to 0 and all other integers to 1, as well as the function $\bar{\mathrm{sg}}$ which maps 0 to 1 and all other integers to 0, are primitive recursive.

$$\mathrm{sg}(0) = 0$$
$$\mathrm{sg}(x+1) = \lambda xy.1(x, \mathrm{sg}(0))$$

The other case is the same.
(5) Show that the set of primitive functions is closed under the iteration definition scheme, which associates to a function g from $\mathbb{N}^p \to \mathbb{N}$ and to a function $h : \mathbb{N}^{p+1} \to \mathbb{N}$ the function $f : \mathbb{N}^p \to \mathbb{N}$ defined by:

$$f(a_1, \ldots, a_p, 0) = g(a_1, \ldots, a_p)$$
$$f(a_1, \ldots, a_p, x+1) = h(a_1, \ldots, a_p, f(a_1, \ldots, a_p, x))).$$

We can write

$$f(a_1,\dots,a_p,0) = g(a_1,\dots,a_p)$$
$$f(a_1,\dots,a_p,x+1) = h(p^1_{p+2}(\bar a,x,f(\bar a,x)),\dots,p^p_{p+2}(\bar a,x,f(\bar a,x)),p^{p+2}_{p+2}(\bar a,x,f(\bar a,x)))$$

to express f in primitive recursive form. Then show that the functions introduced so far in this exercise can be defined from the base functions and the iteration scheme. We have

$$\begin{aligned}
+(x,0) &= x\\
+(x,y+1) &= s(x,+(x,y))\\
\times(x,0) &= x\\
\times(x,y+1) &= +(x,\times(x,y))\\
\exp(x,0) &= x\\
\exp(x,y+1) &= \times(x,\exp(x,y))
\end{aligned}$$

(6) Show that the set of primitive recursive functions is closed *by case definition* on a primitive recursive predicate: if g and h are primitive recursive functions from $\mathbb{N}^p$ to $\mathbb{N}$, and P is a primitive recursive predicate on $\mathbb{N}^p$, then the function f from $\mathbb{N}^p$ to $\mathbb{N}$ defined below is primitive recursive:

$$f(a_1,\dots,a_p) = \begin{cases} g(a_1,\dots,a_p) & \text{if } P(a_1\dots,a_n)\\ h(a_1,\dots,a_n) & \text{otherwise}\end{cases}$$

We have $f(\bar a) = g(\bar a)\chi_P(\bar a) + h(\bar a)\chi_P(\bar a)$.

Exercise 3 (bounded sum and product). Show that if $f : \mathbb{N}^{p+1} \to \mathbb{N}$ is primitive recursive, the functions g and h defined by

$$g(\bar a,x) = \sum_{i=0}^{x} f(\bar a,i) \text{ and } h(\bar a,x) = \prod_{i=0}^{x} f(\bar a,i)$$

are primitive recursive.

Solution: We have

$$\begin{aligned}
g(\bar a,0) &= f(\bar a,0)\\
g(\bar a,x+1) &= g(\bar a,x) + f(\bar a,x+1)\\
h(\bar a,0) &= f(\bar a,0)\\
h(\bar a,x+1) &= h(\bar a,x)\times f(\bar a,x+1)
\end{aligned}$$

Exercise 4 (predecessor, comparison)

(1) Show that the function $\mathrm{pred}\,\mathbb{N}\to\mathbb{N}$ which equals 0 at 0 and $n-1$ at $n>0$ is primitive recursive.

$$\begin{aligned}
\mathrm{pred}(0) &= 0\\
\mathrm{pred}(n+1) &= n
\end{aligned}$$

(2) Show that $x \dot- y = x - y$ if $x \geq y$ and 0 otherwise, as well as the function $x,y \mapsto |x-y|$ are primitive recursive.

$$\begin{aligned}
x \dot- 0 &= x\\
x \dot- (y+1) &= \mathrm{pred}(x \dot- y)
\end{aligned}$$

(3) Show that the comparison predicates $\leq, \geq, <, <, =, \neq$ are primitive recursive. We have $\chi_{\leq}(x,y) = \bar{\text{sg}}(x \dot{-} y)$, $\chi_{\geq}(x,y) = \bar{\text{sg}}(y \dot{-} x)$, $\chi_{=}(x,y) = \chi_{\leq}(x,y)\chi_{\geq}(x,y)$, $\chi_{\neq}(x,y) = \bar{\text{sg}}(\chi_{=}(x,y))$, $\chi_{<}(x,y) = \chi_{\leq}(x,y)\chi_{\neq}(x,y)$, $\chi_{>}(x,y) = \chi_{\geq}(x,y)\chi_{\neq}(x,y)$.

Exercise 5 (Primitive recursive predicates, boolean operations)

(1) Show that the set of primitive recursive predicates of any arity is closed under boolean operations.
(2) Deduce that the set of primitive recursive sets is closed under union, intersection and complement.

Solution: (1) and (2) If $P[\bar{x}, \bar{y}]$ and $Q[\bar{x}', \bar{y}]$ are primitive recursive predicates, we have

$$\begin{aligned}
\chi_{P \wedge Q}(\bar{x}, \bar{x}', \bar{y}) &= \chi_P(\bar{x}, \bar{y})\chi_Q(\bar{x}', \bar{y}) \\
\chi_{P \vee Q}(\bar{x}, \bar{x}', \bar{y}) &= \text{sg}(\chi_P(\bar{x}, \bar{y}) + \chi_Q(\bar{x}', \bar{y})) \\
\chi_{\neg P}(\bar{x}, \bar{y}) &= \overline{\text{sg}}(\chi_P(\bar{x}, \bar{y}))
\end{aligned}$$

The same applies to primitive recursive sets in $\mathbb{N}^p$.

Exercise 6. Show that finite and cofinite subsets of $\mathbb{N}^p$ are primitive recursive.
Solution: If $p = 0$, $\varnothing$ has the zero function as its characteristic function. If $p > 0$ and $A \subseteq \mathbb{N}^p$ is finite, we have $A = \{\bar{a}_1, \ldots, \bar{a}_n\}$. We can write

$$\chi_A(\bar{x}) = \begin{cases} 1 & \text{if } \bigvee_{i=0}^{n} \bar{x} = \bar{a}_i \\ 0 & \text{otherwise} \end{cases}$$

By *case definition*, A is primitive recursive. Note that the predicate $P[\bar{x}] : \bar{x} = \bar{a}$ has as its characteristic function $\chi_P(\bar{x}) = \chi_{=}(\bar{x}, \bar{a})$, so it is primitive recursive. If A is cofinite, we have $\chi_A(\bar{x}) = \overline{\text{sg}}(\chi_{\mathbb{N}^p \setminus A}(\bar{x}))$.

Exercise 7 (bounded minimization) The *bounded minimization* scheme associates to a primitive recursive predicate $B \subseteq \mathbb{N}^{p+1}$ the function $f : \mathbb{N}^{p+1} \to \mathbb{N}$ defined by:

$$\begin{aligned}
f(a_1, \ldots, a_p, x) &= \text{ the smallest integer } t \leq x \text{ such that } B(\bar{a}, t) && \text{if such an integer exists} \\
f(a_1, \ldots, a_p, x) &= 0 && \text{if no such integer exists}
\end{aligned}$$

We write $f(\bar{a}, x) = \mu t \leq x B(\bar{a}, t)$.

(1) Given a primitive recursive predicate $B \subseteq \mathbb{N}^{p+1}$, show that the function $b : \mathbb{N}^{p+1} \to \mathbb{N}$ is primitive recursive, where b is defined by:

$$\begin{aligned}
b(a_1, \ldots, a_p, x) &= 0 \text{ if there exists an integer } t \leq x \text{ such that } B(\bar{a}, t) \\
b(a_1, \ldots, a_p, x) &= 1 \text{ if no such integer exists}
\end{aligned}$$

We can write

$$b(\bar{a}, x) = \overline{\text{sg}} \left(\sum_{t=0}^{x} \chi_B(\bar{a}, t) \right)$$

(2) Deduce that the set of primitive recursive functions is closed under the bounded minimization scheme. We can write, using the helper function b,

$$f(\bar{a}, x) = \sum_{t=0}^{x} b(\bar{a}, x).$$

Exercise 8 (bounded quantification). Show that the set of primitive recursive predicates is closed under bounded existential and universal quantification.
Solution: If P is a primitive recursive predicate, and we define

$$P_e[\bar{x}, y] = \exists z \leq y P[\bar{x}, z]$$
$$P_q[\bar{x}, y] = \forall z \leq y P[\bar{x}, z]$$

Then,

$$\chi_{P_e}(\bar{x}, y) = \mathrm{sg}\left(\sum_{t=0}^{y} \chi_P(\bar{x}, t)\right)$$
$$\chi_{P_q}(\bar{x}, y) = \left(\prod_{t=0}^{y} \chi_P(\bar{x}, t)\right)$$

Exercise 9 (Euclidean division). Show that the functions $q : \mathcal{N}^2 \to \mathbb{N}$ and $r : \mathbb{N}^2 \to \mathbb{N}$ where $q(n, p)$ is the quotient and $r(n, p)$ the remainder of the division of n by p are primitive recursive functions. Deduce that the binary predicate $a|b$ is primitive recursive.
Solution:

$$q(n, p) = \mu t \leq n(pt \leq n \wedge p(t + 1) > n)$$
$$r(n, p) = n \dot{-} (p \times q(n, p))$$

And we have that

$$\chi_{n|p}(n, p) = \begin{cases} 1 & \text{if } r(n, p) = 0 \\ 0 & \text{otherwise} \end{cases}$$

Exercise 10 (prime numbers). Let $p : \mathbb{N} \to \mathbb{N}$ be the function such that $p(n)$ is the $(n+1)$-th prime number.

(1) Show that the predicate "being prime" is primitive recursive.

$$p \text{ is prime iff } p > 1 \wedge \forall x \leq p(\neg x|p \vee x = 1 \vee x = p)$$

(2) Show that $p(n + 1) \leq p(n)! + 1$ and that the factorial function is primitive recursive.
Let q be prime such that $q|p(n)! + 1$, we know that $q \notin \{p(0), \ldots, p(n)\}$ (otherwise it would imply the absurdity $q|1$), this implies that $p(n + 1) \leq q \leq p(n)! + 1$. We also have

$$0! = 1$$
$$(n + 1)! = (n + 1)n!$$

This shows that $n!$ is primitive recursive.

(3) Show that the function p is primitive recursive.
Consider the primitive recursive function

$$p'(n, y_1, y_2) = \mu t \leq y_1(t \text{ is prime} \wedge y_2 \leq t).$$

Then,

$$p(0) = 2$$
$$p(n + 1) = p'(n, p(n)! + 1, p(n))$$

This shows that $p(n)$ is primitive recursive.

Exercise 11 (encoding of pairs and k-tuples). Let α be the Cantor bijection from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$, defined by

$$\alpha(n,p) = \left(\sum_{i=0}^{n+p} i\right) + p.$$

(1) Verify that α is indeed bijective and primitive recursive. Verify that α is increasing on each of its two components. If $m = n + n'$ we have, for all p

$$\alpha(m,p) = \left(\sum_{i=0}^{m+p} i\right) + p = \left(\sum_{i=0}^{n+p} i\right) + \left(\sum_{i=n+p+1}^{n+n'+p} i\right) + p \geq \left(\sum_{i=0}^{n+p} i\right) + p = \alpha(n,p).$$

If $p \leq q$ it is evident that for all n

$$\left(\sum_{i=0}^{n+p} i\right) + p \leq \left(\sum_{i=0}^{n+q} i\right) + q.$$

From ex 3. it is clear that α is primitive recursive. We show injectivity, denote $(\sum_{i=0}^{n} i) = \Delta(n)$
Let $(n,p) \neq (m,q)$, if $n + p = m + q$ then $\Delta(n+p) = \Delta(m+q)$, and if we assume $\alpha(n,p) = \alpha(m,q)$ this would imply $p = q$ and hence $n = m$, a contradiction. For surjectivity, let $m \in \mathbb{N}$, take the smallest x such that $\Delta(x) \leq m \leq \Delta(x+1)$ and take $r = m - x$. Note that $r \leq x$, otherwise we would have $r > x \Rightarrow m = \Delta(x) + r \geq \Delta(x+1)$ which contradicts the minimality of x. Then, $x = r + m$ and $m = \Delta(r+m) + r = \alpha(m,r)$

(2) Define in primitive recursive fashion the two associated projections π_2^1 and π_2^2 satisfying

$$\alpha(\pi_2^1(c), \pi_2^2(c)) = c \ , \ \pi_2^1(\alpha(n,p)) = n \ , \ \pi_2^2(\alpha(n,p)) = p.$$

It is evident that $n, p \leq \alpha(n,p)$, we can write

$$\pi_2^1(c) = (\mu z \leq c)(\exists t \leq c)(\alpha(z,t) = c)$$
$$\pi_2^2(c) = (\mu z \leq c)(\exists t \leq c)(\alpha(t,z) = c)$$

(3) We define by induction on $k \leq 1$ the functions $\alpha_k : \mathbb{N}^p \to \mathbb{N}$ by:

$$\alpha_1(n) = n$$
$$\alpha_{k+1}(n_1, \ldots, n_{k+1}) = \alpha(n_1, \alpha_k(n_2 \ldots, n_{k+1}))$$

Show that, for all $k \leq 1$, α_k is a primitive recursive bijection and define recursively the associated projections $\pi_k^i : \mathbb{N} \to \mathbb{N}$. Verify that α_k is increasing on each of its components. We also write $\langle x_1, \ldots, x_k \rangle$ for $\alpha_k(x_1, \ldots, x_k)$.
The fact that α_k is bijective and primitive recursive is easily shown by induction, because α_k is a composition of primitive recursive functions. By induction, if π_k^i are defined for $i = 1, \ldots, k$, we define

$$\pi_{k+1}^1(n_1, \ldots, n_{k+1}) = \pi_2^1(\alpha(n_1, \alpha_k(n_2, \ldots, n_{k+1})))$$
$$\pi_{k+1}^i(n_1, \ldots, n_{k+1}) = \pi_k^{i-1}(\pi_2^2(\alpha(n_1, \alpha_k(n_2, \ldots, n_{k+1})))) \ \text{ for } i \in \{2, \ldots, k+1\}$$

Exercise 12 (Definitions by mutual recursion). Use the function α_k to show that if the functions $g_1, \ldots, g_k : \mathbb{N}^n \to \mathbb{N}$ and $h_1, \ldots, h_k : \mathbb{N}^{n+k+1}$ are primitive recursive, then the functions

$f_1, \ldots, f_k$ defined below are primitive recursive

$$\begin{aligned}
&f_1(\bar{a}, 0) = g_1(\bar{a}) \\
&\vdots \\
&f_k(\bar{a}, 0) = g_k(\bar{a}) \\
&f_1(\bar{a}, x+1) = h_1(\bar{a}, x, f_1(\bar{a}, x), \ldots, f_k(\bar{a}, x)) \\
&\vdots \\
&f_k(\bar{a}, x+1) = h_k(\bar{a}, x, f_1(\bar{a}, x), \ldots, f_k(\bar{a}, x))
\end{aligned}$$

We can simply write for $i \leq i \leq k$

$$\begin{aligned}
f_i(\bar{a}, 0) &= \pi_k^i(\alpha_k(g_1(\bar{a}), \ldots, g_k(\bar{a})) \\
f_i(\bar{a}, x+1) &= \pi_k^i\Big(\alpha_k\Big(h_1(\bar{a}, x, f_1(\bar{a}, x), \ldots, f_k(\bar{a}, x)), \ldots, h_k(\bar{a}, x, f_1(\bar{a}, x), \ldots, f_k(\bar{a}, x)\Big)\Big)
\end{aligned}$$

By the composition scheme, f_i is primitive recursive.

Exercise 13 (A bijective encoding of finite sequences). We obtain the function ::

$$x :: y = 1 + \alpha_2(x, y)$$

We thus obtain a bijective primitive recursive function $\mathbb{N}^2 \to \mathbb{N}^*$. We call hd and tl the functions satisfying

$$\begin{aligned}
\mathrm{hd}(0) &= 0 & \mathrm{tl}(0) &= 0 \\
\mathrm{hd}(x :: y) &= x & \mathrm{tl}(x :: y) &= y
\end{aligned}$$

We define a function list from the set $\mathcal{S}$ of finite sequences of integers to $\mathbb{N}$ as follows (we write $[a_0; \ldots; a_n] = \mathrm{list}(a_0, \ldots, a_n))$

$$\begin{aligned}
[\ ] &= 0 \\
[a_0; \ldots; a_n] &= a_0 :: [a_1; \ldots; a_n]
\end{aligned}$$

Show that the function list is bijective, and that the functions hd and tl are primitive recursive. We have

$$\begin{aligned}
\mathrm{hd}(c) &= \pi_2^1(c \dot{-} 1) \\
\mathrm{tl}(c) &= \pi_2^2(c \dot{-} 1)
\end{aligned}$$

To show that list is injective, let $[a_0; \ldots; a_n) = [b_0; \ldots; b_{n+k}]$ for some $k \geq 0$, then

$$\begin{aligned}
a_0 :: [a_1; \ldots; a_n] &= b_0 :: [b_1; \ldots; b_{n+k}] \\
\Rightarrow a_0 = b_0 \wedge [a_1; \ldots; a_n] &= [b_1; \ldots; b_{n+k}]
\end{aligned}$$

We can repeat this argument starting from $[a_1; \ldots; a_n] = [b_1; \ldots; b_{n+k}]$ and arrive at

$$\bigwedge_{i=0}^{n} a_i = b_i \wedge [\ ] = [b_1, ; \ldots; b_{k+1}]$$

This shows that $k = 0$ and $(a_0, \ldots, a_n) = (b_0, \ldots, b_n)$. To show surjectivity, simply note that for all $m \in \mathbb{N}$, there is $k \in \mathbb{N}$ such that $\mathrm{tl}^k(m) = 0$ (because the sequence $\{\mathrm{tl}^k(m)\}_{k\in\mathbb{N}}$ is strictly decreasing), then

$$m = [\mathrm{hd}(m); \mathrm{hd}(\mathrm{tl}(m)); \ldots ; \mathrm{hd}(\mathrm{tl}^k(m))] = [\mathrm{nth}(m, 0); \ldots ; \mathrm{nth}(m, k)].$$

Exercise 14 (recursion on the sequence of values).

(1) Prove that the set of recursive functions is closed by the following scheme of recursion on the sequence of values: if $g : \mathbb{N}^p \to \mathbb{N}$ and $h : \mathbb{N}^{p+2} \to \mathbb{N}$ are primitive recursive, then $f : \mathbb{N}^{p+1} \to \mathbb{N}$ defined by

$$\begin{aligned} f(a_1, \ldots, a_p, 0) &= g(a_1, \ldots, a_p) \\ f(a_1, \ldots, a_p, x + 1) &= h(\bar{a}, x, [f(\bar{a}, x); \ldots ; f(\bar{a}, 0)]). \end{aligned}$$

It suffices to prove that the function $F(\bar{a}, x) = [f(\bar{a}, x); \ldots ; f(\bar{a}, 0]$ is primitive recursive. We have

$$\begin{aligned} F(\bar{a}, 0) &= [f(\bar{a}, 0)] = g(\bar{a})) :: 0 \\ F(\bar{a}, x + 1) &= f(\bar{a}, x + 1) :: F(\bar{a}, x) = h(\bar{a}, x, F(\bar{a}, x)) :: F(\bar{a}, x) \end{aligned}$$

We thus have $f(\bar{a}, x) = \mathrm{hd}(F(\bar{a}, x)$

(2) Show that the function $\mathrm{nthl}(l, i)$ which returns the sequence encoded by l starting from the $(i + 1)$-th element (0 otherwise), and the function $\mathrm{nth}(l, i)$ which returns the $(i + 1)$-th element of the sequence encoded by l, are primitive recursive.

$$\begin{array}{ll} \mathrm{nthl}(l, 0) = l & \mathrm{nth}(l, 0) = \mathrm{hd}(l) \\ \mathrm{nthl}(l, i + 1) = \mathrm{tl}(\mathrm{nthl}(l, i)) & \mathrm{nth}(l, i + 1) = \mathrm{hd}(\mathrm{nthl}(l, i)) \end{array}$$

(3) Show that if $g : \mathbb{N}^p \to \mathbb{N}$, $h : \mathbb{N}^{p+k+1} \to \mathbb{N}$ are primitive recursive, and if $p_1, \ldots, p_k : \mathbb{N} \to \mathbb{N}$ are primitive recursive functions each satisfying

$$\forall x \in \mathbb{N} p_i(x) \leq x$$

then $f : \mathbb{N}^{p+1} \to \mathbb{N}$ defined by

$$\begin{aligned} f(a_1, \ldots, a_p, 0) &= g(a_1, \ldots, a_p) \\ f(a_1, \ldots, a_p, x + 1) &= h(\bar{a}, x, f(\bar{a}, p_1(x)), \ldots, f(\bar{a}, p_k(x)) \end{aligned}$$

is primitive recursive. We can write

$$\begin{aligned} f(\bar{a}, x + 1) = h\Big(&\bar{a}, x, \mathrm{nth}([f(\bar{a}, 0); \ldots ; f(\bar{a}, x)], x - p_1(x)), \\ &\ldots, \mathrm{nth}([f(\bar{a}, 0); \ldots ; f(\bar{a}, x)], x - p_k(x))\Big) \end{aligned}$$

Exercise 15 (recursion on lists).

(1) Show that f is primitive recursive

$$\begin{aligned} f(\bar{a}, []) &= g(\bar{a}) \\ f(\bar{a}, x :: l) &= h(\bar{a}, x, l, f(\bar{a}, l)). \end{aligned}$$

We can write

$$f(\bar{a}, y) = h(\bar{a}, \mathrm{hd}(y), \mathrm{tl}(y), f(\bar{a}, \mathrm{tl}(y)))$$

f is well-defined since the list function is bijective.
mem

$$\begin{aligned} \mathrm{mem}(a, []) &= 0 \\ \mathrm{mem}(a, x :: l) &= \chi_=(x, a)\,\mathrm{mem}(a, l) \end{aligned}$$

@

$$\begin{aligned} @(l', []) &= l' \\ @(l', x :: l) &= x :: (l@l') \end{aligned}$$

length

$$\begin{aligned} \mathrm{lg}([]) &= 0 \\ \mathrm{lg}(x :: l) &= \mathrm{lg}(l) + 1 \end{aligned}$$

(2) Show that if f is pr, then the function $\mathrm{map}(f)$ which maps $l = [\bar{u}]$ to $[f(\bar{a}, u_1); \ldots; f(\bar{a}, u_p)]$

$$\begin{aligned} \mathrm{map}_f([]) &= 0; \\ \mathrm{map}_f(x :: l) &= f(\bar{a}, x) :: \mathrm{map}_f(l) \end{aligned}$$

(3) concat

$$\begin{aligned} \mathrm{concat}([\ ]) &= [\ ]; \\ \mathrm{concat}(x :: l) &= x :: [\mathrm{nth}(l, 0); \ldots, \mathrm{nth}(l, \mathrm{length}(l))] \end{aligned}$$

subst

$$\begin{aligned} \mathrm{subst}([\ ], k, v) &= [\ ]; \\ \mathrm{subst}(x :: l, k, v) &= \begin{cases} x :: \mathrm{subst}(l) & \text{if } x \neq v \\ \mathrm{concat}([k, \mathrm{subst}(l)] & \text{if } x = v \end{cases} \end{aligned}$$

Extra Exercise (Encoding lists by prime number decomposition) Let $\mathcal{S}$ denote the set of finite sequences of integers. The list encoding function $\mathrm{seq} : \mathcal{S} \to \mathbb{N}$ associates to each sequence $(x_1, \ldots, x_k)$ the following value

$$\mathrm{seq}(x_1, \ldots, x_k) = p_0^k p_1^{x_1} \cdots p_k^{x_k}$$

sending the empty sequence to 1.

(1) Show that this encoding is injective but not surjective. Injectivity is clear by the fundamental theorem of arithmetic. There is no sequence sent to 3, for example.

(2) Show that the function which maps (x, n) to the exponent of p_n in the prime factorization of x is primitive recursive.

$$\exp(x, n) = (\mu k \leq x)(p_n^{k+1} \nmid x)$$

(3) Deduce that
  (a) There exists a primitive recursive function that computes the n-th element of a sequence represented by x, when x represents a sequence of length greater than or equal to n. Take $\exp(x, n)$
  (b) There exists a pr function that computes the length of the sequence encoded by x. Take $l(x) = \exp(n, 0)$.
  (c) The characteristic function of the set C of sequence codes is primitive recursive. We have $x \in A$ iff $x \neq 0$ and $(x = 1 \vee 2|x)$.

(4) Show that there exists a primitive recursive function which, given two integers $n = \mathrm{seq}(x_1, \ldots, x_k)$ and $m = \mathrm{seq}(y_1, \ldots, y_h)$ encoding sequences, returns the number representing the concatenation of the two lists $\mathrm{seq}(\bar{x}, \bar{y})$.
Take $\mathrm{concat}(n, m) = \mathrm{seq}(\exp(n, 1) \ldots, \exp(n, k), \exp(m, 1), \ldots, \exp(m, h))$

Exercise 16 (recursion with parameter substitution). This is the scheme

$$\begin{aligned} f(a, 0) &= g(a) \\ f(a, x+1) &= h(a, x, f(\gamma(a), x)). \end{aligned}$$

(1) Show that the function F is PR

$$\begin{aligned} F(p, a, 0) &= g(\gamma^p(a)) \\ F(p, a, x+1) &= h(\gamma^{p-(x+1)}(a), x, F(p, a, x)). \end{aligned}$$

We see that

$$F(p, a, x+1) = h(\mathrm{nth}([a; \gamma(a), \ldots, \gamma^p(a), p - (x+1)]), x, F(p, a, x))$$

(2) Show that

$$\forall x, a, p \in \mathbb{N}(x \leq p \Rightarrow F(p, a, x) = f(\gamma^{p-x}(a), x))$$

and deduce that f is primitive recursive.
By induction on x (we assume $x \leq p$ always)

$$\begin{aligned} F(p, a, 0) &= g(\gamma^p(a)) \\ F(p, a, x+1) &= h(\gamma^{p-(x+1)}(a), x, F(p, a, x)) \\ &= h(\gamma^{p-(x+1)}(a), x, f(\gamma^{p-x}(a), x)) \\ &= f(\gamma^{p-(x+1)}(a), x+1) \end{aligned}$$

We can deduce that $f(a, x) = F(x, a, x)$.

(3) Application: show that the function $\mathrm{inc} : \mathbb{N}^2 \to \mathbb{N}$ which maps i and $l = [a_0; \ldots; a_i; \ldots; a_n]$ to $[a_0; \ldots; a_i + 1; \ldots; a_n]$, is primitive recursive.
We can write

$$\begin{aligned} f(l, 0) &= (\mathrm{hd}(l) + 1) :: \mathrm{tl}(l) \\ f(l, i+1) &= \begin{cases} f(\mathrm{tl}(l), i) & \text{if } i \leq n \\ l & \text{otherwise} \end{cases} \end{aligned}$$

Exercise 17 (double recursion without nesting). Show that the function f defined by

$$\begin{aligned} f(0, y) &= a \\ f(x+1, 0) &= b \\ f(x+1, y+1) &= h(x, y, f(x, y), f(x+1, y)). \end{aligned}$$

is primitive recursive.
We can use the encoding of pairs ($t = \langle x, y \rangle$), and the scheme of recursion on the sequence of values

to write f as follows

$$f(t)=\begin{cases} b & \text{if } \pi_2^1(t)=0\\ a & \text{otherwise and } \pi_2^2(t)=0\\ h\Bigg(\pi_2^1(t)-1,\pi_2^2(t)-1, & \\ \text{nth}\left([f(0);f(1);\dots;f(\alpha(\pi_2^1(t),\pi_2^2(t)-1)],\alpha(\pi_2^1(t)-1,\pi_2^2(t)-1)\right), & \\ \text{nth}\left([f(0);f(1);\dots;f(\alpha(\pi_2^1(t),\pi_2^2(t)-1)],\alpha(\pi_2^1(t),\pi_2^2(t)-1)\right)\Bigg), & \text{otherwise} \end{cases}$$

Ackermann Function

$$\begin{aligned} \text{Ack}(0,x)&=x+2\\ \text{Ack}(1,0)&=0\\ \text{Ack}(n+2,0)&=1\\ \text{Ack}(n+1,x+1)&=\text{Ack}(n,\text{Ack}(n+1,x)) \end{aligned}$$

Exercise 18. Show that each function $\text{Ack}_n(x)=\text{Ack}(n,m)$ is primitive recursive and strictly increasing. Make explicit Ack_n, for $n=1,2,3$. We proceed by induction on n. If $n=0,1$, it is evident that Ack_n is primitive recursive. We assume that Ack_n is primitive recursive for $n\geq 2$. We note that

$$\begin{aligned} \text{Ack}_{n+1}(0)&=1\\ \text{Ack}_{n+1}(x+1)&=\text{Ack}_n(\text{Ack}_{n+1}(x)) \end{aligned}$$

By the induction hypothesis, $\text{Ack}_{n+1}=\text{Rec}(1,\text{Ack}_n\circ\pi_2^2)\Rightarrow\text{Ack}_{n+1}$ is primitive recursive. We also have

$$\begin{aligned} \text{Ack}_1(x)&=2x\\ \text{Ack}_2(x)&=2^x\\ \text{Ack}_3(x)&=\underbrace{2\,\hat{}\cdots\hat{}\,2}_{x-\text{times}}\hat{}x \end{aligned}$$

The fact that the function is strictly increasing follows from an immediate application of induction on n.

Exercise 19 (Ackermann function)

(1) Verify that there exists exactly one function from $\mathbb{N}^2\to\mathbb{N}$ satisfying the Ackermann function equations. Consider the recurrence

$$\text{Ack}_{n+1}(x+1)=\text{Ack}_n(\text{Ack}_{n+1}(x)),$$

and note that, if $>_{lex}$ denotes the lexicographic order on $\mathbb{N}^2$, we have

$$\begin{aligned} (n+1,x+1)&>_{lex}(n+1,x)\\ (n+1,x+1)&>_{lex}(n,\text{Ack}_{n+1}(x)) \end{aligned}$$

We can deduce that, to compute the value $\text{Ack}_{n+1}(x+1)$, we need the values that Ack takes at pairs strictly smaller than $(x+1,n+1)$ (according to $>_{lex}$). Since $(\mathbb{N}^2,>_{lex})$ is a well-ordered set, it follows that the set of pairs needed to compute $\text{Ack}_{n+1}(x+1)$ is finite. Therefore, Ack is well-defined on $\mathbb{N}^2$, and it is "intuitively computable".

(2) Show that

$$\forall n \in \mathbb{N}\ \forall x > 0\, \mathrm{Ack}_{n+1}(x) = \mathrm{Ack}_n^x(\mathrm{Ack}_{n+1}(0)))$$

and verify the expressions of the functions Ack_1, Ack_2, Ack_3. By induction on x. If $x = 0$, it is trivial. For the case $x+1$, we have

$$\begin{aligned}\mathrm{Ack}_{n+1}(x+1) &= \mathrm{Ack}_n(\mathrm{Ack}_{n+1}(x)) \text{ by definition}\\ &= \mathrm{Ack}_n(\mathrm{Ack}_n^x(\mathrm{Ack}_{n+1}(0))) \text{ IH}\\ &= \mathrm{Ack}_n^{x+1}(\mathrm{Ack}_{n+1}(0)))\end{aligned}$$

(3) Verify that each of the functions Ack_n has a definition using exactly n instances of the iteration definition scheme. The explicit forms of $\mathrm{Ack}_1, \mathrm{Ack}_2, \mathrm{Ack}_3$ are in exercise 18. This follows directly from the definition, and by induction on n, noting that

$$\begin{aligned}\mathrm{Ack}_{n+1}(0) &= 1\\ \mathrm{Ack}_{n+1}(x+1) &= \mathrm{Ack}_n(\mathrm{Ack}_{n+1}(x))\end{aligned}$$

Then, if $\mathrm{Ack}_n \in \mathcal{C}_n$, $\mathrm{Ack}_{n+1} \in \mathcal{C}_{n+1}$.

(4) Show that $\mathrm{Ack}_n(x) > x$.
By induction on n. If $x > 0$,

$$\begin{aligned}\mathrm{Ack}_0(x) &= x + 2 > x\\ \mathrm{Ack}_1(x) &= 2x > x\end{aligned}$$

If $n \geq 2$ and we assume that for all $x > 0$, $\mathrm{Ack}_n(x) > x$,

$$\mathrm{Ack}_{n+1}(1) = \mathrm{Ack}_n(\mathrm{Ack}_{n+1}(0)) = \mathrm{Ack}_n(1) > 1$$

If $x > 1$, since Ack is strictly increasing, $\mathrm{Ack}_{n+1}(x) > 1 \neq 0$, and we can apply induction on x,

$$\begin{aligned}\mathrm{Ack}_{n+1}(x+1) &= \mathrm{Ack}_n(\mathrm{Ack}_{n+1}(x))\\ &\geq \mathrm{Ack}_{n+1}(x) + 1 \text{ (IH1)}\\ &> x + 1 \text{ (IH2)}\end{aligned}$$

(5) Deduce that for all integers m, Ack_m is strictly increasing.
This has already been demonstrated in the previous exercise.

(6) Deduce from question 4, that, from 2 onwards, Ack is non-decreasing on its first argument, the second being fixed:

$$\forall x \geq 2\ \forall n \in \mathbb{N}\ \ \mathrm{Ack}(n, x) \leq \mathrm{Ack}(n+1, x).$$

We have

$$\mathrm{Ack}_{n+1}(x) = \mathrm{Ack}_n(\underbrace{\mathrm{Ack}_{n+1}(x-1)}_{\geq x}) \geq \mathrm{Ack}_n(x)$$

(7) Show that $\forall k, n \in \mathbb{N}\, \mathrm{Ack}_n^k \in \mathcal{C}_n$.
This is clear in view of exercise 19.3 and since $\mathcal{C}_n$ is closed under composition.

(8) Show that $\forall k, n \in \mathbb{N}\, \mathrm{Ack}_n^k(x) \leq \mathrm{Ack}_{n+1}(x+k)$. By induction on k, the case $k=0$ is trivial, then

$$\begin{aligned}\mathrm{Ack}_n^{k+1}(x) &= \mathrm{Ack}_n(\mathrm{Ack}_n^k(x))\\ &\leq \mathrm{Ack}_n(\mathrm{Ack}_{n+1}(x+k)) \text{ IH}\\ &= \mathrm{Ack}_{n+1}(x+k+1) \text{ def}\end{aligned}$$

(9) Show by induction on the definition of the set of primitive recursive functions that if $f \in \mathcal{C}_n$, then $\exists k\, \mathrm{Ack}_n^k$ dominates f.
It is easy to see that the base functions are dominated by $\mathrm{Ack}_3(x)$.
If $h, g_1, \dots, g_m \in \mathcal{C}_n$, $h(\bar{x}) \leq \mathrm{Ack}_n^k \sup(\bar{x}, K)$ and $g_i(\bar{x}) \leq \mathrm{Ack}_n^{k_i} \sup(\bar{x}, K_i)$, we set $M = \sup(K_1, \dots, K_m, K)$, $l = \sup_i k_i$, and $M(\bar{x}) = \sup(\bar{x}, M)$.

$$\begin{aligned}h(g_1(\bar{x}), \dots, g_m(\bar{x})) &\leq \mathrm{Ack}_n^k \sup_i(g_i(\bar{x}), K)\\ &\leq \mathrm{Ack}_n^k \sup_i(\mathrm{Ack}_n^{k_i} \sup(\bar{x}, K_i), K)\\ &\leq \mathrm{Ack}_n^k \sup_i(\mathrm{Ack}_n^{k_i}(M(\bar{x})))\\ &= \mathrm{Ack}_n^k(\mathrm{Ack}_n^l(M(\bar{x})))\\ &= \mathrm{Ack}_n^{k+l} \sup(\bar{x}, M)\end{aligned}$$

Now, if $g(\bar{x}) \leq \mathrm{Ack}_n^{k_1} \sup(\bar{x}, N_1)$, and $h(\bar{x}, y, z) \leq \mathrm{Ack}_n^{k_2} \sup(\bar{x}, y, z, N_2)$, the function obtained by primitive recursion $f \in \mathcal{C}_{n+1}$ satisfies

$$f(\bar{x}, y) \leq \mathrm{Ack}_n^{k_1 + k_2 y}(\sup(\bar{x}, y, N_1, N_2)$$

We prove by induction on y,

$$\begin{aligned}f(\bar{x}, 0) &= g(\bar{x}) \leq \mathrm{Ack}_n^{k_1} \sup(\bar{x}, N_1)\\ f(\bar{x}, y+1) &= h(\bar{x}, y, f(\bar{x}, y))\\ &\leq \mathrm{Ack}_n^{k_2}(\sup(\bar{x}, y, f(\bar{x}, y), N_2)\\ &\leq \mathrm{Ack}_n^{k_2}(\sup(\bar{x}, y, \mathrm{Ack}_n^{k_1+k_2 y} \sup(\bar{x}, y, N_1, N_2), N_2)\\ &= \mathrm{Ack}_n^{k_2}(\mathrm{Ack}_n^{k_1+k_2 y} \sup(\bar{x}, y, N_1, N_2))\\ &= \mathrm{Ack}_n^{k_1+k_2(y+1)} \sup(\bar{x}, y, N_1, N_2)\\ &\leq \mathrm{Ack}_{n+1}(\sup(\bar{x}, y, N_1, N_2) + k_1 + k_2 y)\end{aligned}$$

This last function is a composition of C_{n+1} functions and therefore is dominated by some Ack_{n+1}^l.

(10) Show that Ack_n^k is dominated by Ack_{n+1}.
Note that if $y > 0$, $\mathrm{Ack}_{n+1}(y) \geq \mathrm{Ack}_1(y) = 2y$, and we can deduce that if $x > 2k$, $\mathrm{Ack}_{n+1}(x-k) \geq 2x - 2k > x$. Then, for all $x > 2k$,

$$\begin{aligned}&\mathrm{Ack}_{n+1}(x-k) > x\\ \Rightarrow &\mathrm{Ack}_n^k(\mathrm{Ack}_{n+1}(x-k)) > \mathrm{Ack}_n^k(x)\\ \Rightarrow &\qquad \mathrm{Ack}_{n+1}(x) > \mathrm{Ack}_n^k(x) \text{ (ex 19.2)}\end{aligned}$$

This shows that Ack_n^k is dominated by Ack_{n+1}.

(11) Deduce that if $f \in \mathcal{C}_n$, then Ack_{n+1} dominates f.
If $f \in \mathcal{C}_n$, $\exists k$ such that f is dominated by Ack_n^k, and by the previous exercise, Ack_n^k is dominated by Ack_{n+1}. Moreover, $\mathrm{Ack}_{n+1} \notin \mathcal{C}_n$.

(12) Deduce that the Ackermann function is not primitive recursive. Show that the diagonal function $\mathrm{Ack}(n,n)$ dominates all primitive recursive functions.
If $\mathrm{Ack}(n,n) \in \mathcal{C}_k$,

$$\exists N\ \forall n > N\ \ \mathrm{Ack}(n,n) \leq \mathrm{Ack}_k(n),$$

which is impossible if $n > N, k$. If f is primitive recursive, $f \in \mathcal{C}_n$ for some n, so using the previous exercises, except for finitely many values of $\bar{x}$

$$f(\bar{x}) \leq \mathrm{Ack}_n^k(\sup(\bar{x})) \leq \mathrm{Ack}_{n+1}(\sup(\bar{x})) \leq \mathrm{Ack}_{\sup(\bar{x})}(\sup(\bar{x}))$$