Encodings — Sequences and Strings
Proofs of Claims

This lecture introduces a large number of functions that will be used later and claims that these functions are primitive recursive. The functions are useful because they will be needed later in the course. The proofs that they are primitive recursive are not as interesting or important — they simply involve the use of techniques that have already been introduced. Thus they are not included in the lecture slides (or, generally, discussed in lectures) — and are given, for completeness, in this supplemental document.

Encoding Ordered Pairs of Natural Numbers

Consider the ***pairing function*** $\langle \cdot, \cdot \rangle : \mathbb{N}^2 \to \mathbb{N}$ such that, for $x, y \in \mathbb{N}$,

$$\langle x, y \rangle = 2^x(2y + 1) \dot{-} 1.$$

Claim 1. *The pairing function $\langle \cdot, \cdot \rangle : \mathbb{N}^2 \to \mathbb{N}$ is primitive recursive.*

Sketch of Proof. Consider the function $g; \mathbb{N}^2 \to \mathbb{N}$ such that, for $t, y \in \mathbb{N}$,

$$g(t, y) = 2 \times y.$$

Since constant functions, the projection functions and multiplication are all primitive recursive, it can be shown (using ***composition*** that g is a primitive recursive function as well.

Next consider the function $f : \mathbb{N} \to \mathbb{N}$ obtained from g using ***primitive recursion*** (when the constant k, used to define $h(0)$, is 1): $h(0) = k = 1$, and, for every integer $t \geq 0$,

$$h(t + 1) = g(t, h(t)) = 2 \cdot h(t).$$

It is easy to show — by induction on x — that $h(x) = 2^x$ for all $x \in \mathbb{N}$. This function is, therefore, primitive recursive.

Now since constant functions, addition, monus and multiplication are also primitive recursive, the function $\langle \cdot, \cdot \rangle : \mathbb{N}^2 \to \mathbb{N}$ such that, for all $x, y \in \mathbb{N}$

$$\langle x, y \rangle = 2^x \cdot (2y + 1) \dot{-} 1$$

can be shown to be primitive recursive (and constructed using ***composition***, as claimed. □

As shown in the lecture notes, if $z \in \mathbb{N}$ then there exists *exactly* one pair of values $\ell(z) \in \mathbb{N}$ and $r(z) \in \mathbb{N}$ such that $\langle \ell(z), r(z) \rangle = z$. This can be used to define left and right *inverses* of the pairing function $\ell, r : \mathbb{N} \to \mathbb{N}$.

Claim 2. *The inverse functions $\ell, r : \mathbb{N} \to \mathbb{N}$, for the pairing function, are primitive recursive.*

Sketch of Proof. Consider the predicate $p : \mathbb{N}^2 \to \{0, 1\}$ such that, for $a, b \in \mathbb{N}$,

$$p(a, b) = \begin{cases} 1 & \text{if } a \bmod 2^{b+1} > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Using the functions that were proved to be primitive recursive in Lecture #2 — including the "remainder" function — it can be shown this is a primitive recursive predicate.

Next consider the function $f : \mathbb{N}^2 \to \mathbb{N}$ obtained from p using ***bounded minimization*** — which is a primitive recursive function. One can see by the definition of "bounded minimization" that, for $a, b \in \mathbb{N}$,

$$f(a, b) = \underset{0 \le t \le b}{\textit{first}}\ p(a, t)$$

is equal to 0 if a is divisible by 2^{t+1} for every integer t such that $0 \le t \le b$ — which is true if and only if a is divisible by 2^{b+1}. On the other hand, if a is *not* divisible by 2^{b+1} then $f(a, b)$ is the *smallest* nonnegative integer t such that a is not divisible by 2^{t+1} — which is equal to the *largest* integer t such that a *is* divisible by 2^t.

Notice next that $2^{a+1} > a$ for all $a \in \mathbb{N}$. Thus a is not divisible by 2^{a+1}, and the function $\ell : \mathbb{N} \to \mathbb{N}$ such that, for $a \in \mathbb{N}$,

$$\ell(a) = f(a, a)$$

is always the largest integer t such that a is divisible by 2^t — as in the previous definition. Thus the function $\ell : \mathbb{N} \to \mathbb{N}$ is primitive recursive.

To continue, recall that the *quotient function quo* : $\mathbb{N}^2 \to \mathbb{N}$ such that, for $x, y \in \mathbb{N}$,

$$\textit{quo}(x, y) = \begin{cases} \left\lfloor \frac{x}{y} \right\rfloor & \text{if } y > 0, \\ x & \text{otherwise} \end{cases}$$

is primitive recursive. It now suffices to note that, for $z \in \mathbb{N}$,

$$r(z) = \textit{quo}(\textit{quo}(z + 1, 2^{\ell(z)}) \dot{-} 1, 2)$$

in order to see that the function $r : \mathbb{N} \to \mathbb{N}$ is primitive recursive too. □

Encoding Finite Sequences of Natural Numbers

Claim 3. *For every positive integer k, the function $[.,.,\ldots,.]_k : \mathbb{N}^k \to \mathbb{N}$ — such that, for $x_1, x_2, \ldots, x_k \in \mathbb{N}$,*

$$[x_1, x_2, \ldots, x_k]_k = \prod_{i=1}^{k} p_i^{x_k}$$

— is a primitive recursive function.

Sketch of Proof. Recall, from Lecture #3, that if p_0 is defined to be 1 then the function mapping each number $n \in \mathbb{N}$ to the n^{th} prime number, p_n, is primitive recursive.

Consider the functions $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N}^3 \to \mathbb{N}$ such that

$$f(x_1) = 1$$

for all $x_1 \in \mathbb{N}$ and such that for all $t, y, x_1 \in \mathbb{N}$,

$$g(t, y, x_1) = y \times p_{x_1}.$$

These functions are now easily shown to be primitive recursive.

The function $h : \mathbb{N}^2 \to \mathbb{N}$ obtained from f and g by ***primitive recursion*** is, therefore, primitive recursive as well. Note that, for $x_1 \in \mathbb{N}$,

$$h(x_1, 0) = f(x_1) = 1$$

and, for every integer $t \geq 0$,

$$h(x_1, t+1) = g(t, h(x_1, t), x_1) = h(x_1, t) \times p_{x_1}.$$

It is easy to show, by induction on x_2, that

$$h(x_1, x_2) = p_{x_1}^{x_2}$$

for all $x_1, x_2 \in \mathbb{N}$.

The function $\widehat{h} : \mathbb{N} \to \mathbb{N}$ such that, for $x_1, x_2 \in \mathbb{N}$,

$$\widehat{h}(x_1, x_2) = h(u_2^2(x_1, x_2), u_1^2(x_1, x_2)) = h(x_2, x_1) = p_{x_2}^{x_1}$$

is certainly primitive recursive too.

Now let k be any *fixed positive* integer and consider the function

$$[\cdot, \cdot, \ldots, \cdot]_k : \mathbb{N}^k \to \mathbb{N}$$

such that, for all $x_1, x_2, \ldots, x_k \in \mathbb{N}$,

$$\begin{aligned}[x_1,x_2,\ldots,x_k]_k &= [x_1, x_2, \ldots, x_k] \\ &= p_1^{x_1} \times p_2^{x_2} \times \cdots \times p_k^{x_k} \\ &= \widehat{h}(x_1, 1) \times \widehat{h}(x_2, 2) \times \cdots \times \widehat{h}(x_k, k).\end{aligned}$$

This function is ***primitive recursive*** because (for fixed k) it can be obtained from $\widehat{h}$, multiplication, constant functions and the projection functions using a finite number of applications of composition — as needed to establish the claim. □

As in the lecture notes, consider functions $(\cdot)_i : \mathbb{N} \to \mathbb{N}$ that are defined as follows.

- $(x)_0 = 1$ for all $x \in \mathbb{N}$.
- For every *positive* integer i, let $(x)_i = 0$ if $x = 0$ and let $(x)_i$ be the largest integer t such that x is divisible by p_i^t, otherwise.

Claim 4.

(a) The function $g : \mathbb{N}^2 \to \mathbb{N}$ such that for all $x, i \in \mathbb{N}$, $g(x, i) = (x)_i$, is primitive recursive.

(b) For every number $i \in \mathbb{N}$ the function from $\mathbb{N}$ to $\mathbb{N}$, mapping x to $(x)_i$, is primitive recursive.

Sketch of Proof.

(a) Consider a predicate $p : \mathbb{N}^3 \to \{0, 1\}$ such that, for $a, b, c \in \mathbb{N}$,

$$p(a, b, c) = \begin{cases} 1 & \text{if } a \bmod p_b^{c+1} > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Once again, it is easily shown (using the functions that have been proved to be primitive recursive already) that this function is primitive recursive.

Consider the function $f : \mathbb{N}^3 \to \mathbb{N}$ obtained from p using ***bounded minimization*** — which is a primitive recursive function. For $a, b, c \in \mathbb{N}$,

$$f(a, b, c) = \underset{0 \le t \le c}{\textit{first}}\ p(a, b, t)$$

is equal to 0 if a is divisible by p_b^t for every integer t such that $0 \le t \le c + 1$ — which is true if and only if a is divisible by p_b^{c+1}. Thus $f(0, b, c) = 0$ for all $b, c \in \mathbb{N}$.

On the other hand, if a is *not* divisible by p_b^{c+1} then $f(a, b, c)$ is the *smallest* nonnegative integer t such that a is not divisible by p_b^{t+1}. That is, $f(a, b, c)$ is the *largest* nonnegative integer t such that a is divisible by p_b^t.

Notice that $p_b^{a+1} > a$ for all $a \in \mathbb{N}$ if $b \geq 1$ (so that $p_b \geq 2$). Thus a is not divisible by p_b^{a+1} when $b \geq 1$, and the function $g : \mathbb{N}^2 \to \mathbb{N}$ such that, for all $a, b \in \mathbb{N}$ such that $b \geq 1$,

$$g(a, b) = f(a, b, a)$$

is the largest integer t such that a is divisible by p_b^t. On the other hand, $g(a, 0) = 0$ for all $a \in \mathbb{N}$ since $p_0 = 1$ and $a \bmod 1 = 0$ for all a. Similarly, $g(0, a) = 0$ for all $a \in \mathbb{N}$ as well — because $f(0, b, 0) = 0$, as noted above. Thus this is the function defined in part (a) of the claim.

This function is certainly primitive recursive, since f is — as needed to establish part (a).

(b) Part (b) is a straightforward consequence of part (a) and the fact that the constant functions are all primitive recursive: For each number $i \in \mathbb{N}$ the function mapping each number $x \in \mathbb{N}$ to $(x)_i$ can be produced from these functions and projective functions using composition. □

As in the lecture notes, consider a function Lt : $\mathbb{N} \to \mathbb{N}$ such that $\text{Lt}(0) = \text{Lt}(1) = 1$ and, for every integer $x \geq 2$, $\text{Lt}(x)$ is the smallest integer n such that

$$[(x)_1, (x)_2, \ldots, (x)_n] = x.$$

A number n with this property always exists, so Lt : $\mathbb{N} \to \mathbb{N}$ is a well-defined total function.

Claim 5. *The function Lt* : $\mathbb{N} \to \mathbb{N}$ *is primitive recursive.*

Sketch of Proof. Consider the function $f : \mathbb{N} \to \mathbb{N}$ such that, for all $z \in \mathbb{N}$,

$$f(z) = 1.$$

This function is certainly primitive recursive, since it is one of the constant functions.

Consider, as well, a function $g : \mathbb{N}^3 \to \mathbb{N}$ such that, for $t, y, z \in \mathbb{N}$,

$$g(t, y, z) = y \times p_{t+1}^{(z)_{t+1}}.$$

This function can be obtained from primitive recursive functions that have now been identified using composition, so it is also primitive recursive.

It follows that the function $h : \mathbb{N}^2 \to \mathbb{N}$ obtained from f and g using ***primitive recursion*** is primitive recursive too. Notice that

$$h(z, 0) = f(z) = 1$$

and, for every integer t such that $t \geq 0$,

$$h(z, t + 1) = g(t, h(z, t), z)$$

$$= h(z,t) \times p_{t+1}^{(z)_{t+1}}.$$

It is easily proved by induction on t that, for every integer $t \geq 0$,

$$h(z,t) = \prod_{s=1}^{t} p_s^{(z)_s}.$$

Note that since $(0)_i = (1)_i = 0$ for every positive integer i, $h(0,t) = h(1,t) = 1$ for all $t \in \mathbb{N}$.

Next consider the predicate $p : \mathbb{N}^2 \to \{0,1\}$ such that, for $z, t \in \mathbb{N}$,

$$p(z,t) = \begin{cases} 1 & \text{if } h(z,t) = z, \\ 0 & \text{otherwise.} \end{cases}$$

Note that $p(0,t) = 0$ and $p(1,t) = 1$ for all $t \in \mathbb{N}$. Since h is a primitive recursive function, p is easily shown to be a primitive recursive predicate.

Consider the function $f : \mathbb{N}^2 \to \mathbb{N}$ obtained from p using ***bounded minimization:*** For all $z, b \in \mathbb{N}$,

$$\begin{aligned} f(z,b) &= \underset{0 \leq t \leq b}{\textit{first}}\, p(z,t) \\ &= \underset{0 \leq t \leq b}{\textit{first}}\, (h(z,t) = z) \\ &= \begin{cases} \underset{t \in \mathbb{N}}{\textit{first}} (h(z,t) = z) & \text{if } \underset{0 \leq t \leq b}{\exists} (h(z,t) = z), \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

$f(0,b) = 0$ for all $b \in \mathbb{N}$ because there is no integer $b \in \mathbb{N}$ such that $p(0,b) = 1$. On the other hand, $f(1,b) = 0$ for all $b \in \mathbb{N}$ because $p(1,0) = 1$. Finally, if $t \geq 1$ then, for $x \in \mathbb{N}$, $(x)_t \geq 1$ only if $x \geq p_t \geq t$. It follows that $\mathrm{Lt}(x) = f(x,x)$ for all $x \in \mathbb{N}$, and thus the function Lt is primitive recursive, as claimed. □

Primitive Recursive Functions of Strings

Length of a String

Let Σ be a finite nonempty alphabet.

Claim 6. *The* **length** *function $f : \Sigma^\star \to \mathbb{N}$, such that $f(\omega) = |\omega|$ for all $\omega \in \Sigma^\star$, is primitive recursive.*

Sketch of Proof. Let $n = |\Sigma|$.

- $\sum_{i=0}^{\ell-1} n^i$ is a primitive recursive function of ℓ.
- Thus the predicate $p : \mathbb{N}^2 \to \{0, 1\}$ such that, for all $m, \ell \in \mathbb{N}$,

$$p(m, \ell) = \begin{cases} 0 & \text{if } m < \sum_{i=0}^{\ell-1} n^i \\ 1 & \text{if } m \geq \sum_{i=0}^{\ell-1} n^i \end{cases}$$

  is a primitive recursive predicate.
- Note that if $m = \#(\omega)$ then either $m = |\omega| = 0$ or $m \geq |\omega| \geq 1$.
- Since every string with length ℓ has an encoding greater than or equal to $\sum_{i=0}^{\ell-1} n^i$, but less than $\sum_{i=0}^{\ell} n^i$, for all $\omega \in \Sigma^\star$, if $\#(\omega) = m \geq 1$ then

$$|\omega| = f(\omega) = \underset{t \leq m}{\textit{first}}\, p(m, t)$$

  and, since this was defined from p using bounded minimization, this is a primitive recursive function. □

Concatenation of Strings

Consider the function concat : $\Sigma^\star \times \Sigma^\star \to \Sigma^\star$ such that, for all $\alpha, \beta \in \Sigma^\star$, concat$(\alpha, \beta)$ is the ***concatenation*** of the input strings. In particular, if α has length $k \geq 0$, so that $\alpha = \alpha_1 \alpha_2 \dots \alpha_k$ for symbols $\alpha_1, \alpha_2, \dots, \alpha_k \in \Sigma$, and β has length $\ell \geq 0$, so that $\beta = \beta_1 \beta_2 \dots \beta_\ell$ for symbols $\beta_1, \beta_2, \dots, \beta_\ell \in \Sigma$, then

$$\text{concat}(\alpha, \beta) = \alpha \cdot \beta = \alpha_1 \alpha_2 \dots \alpha_k \beta_1 \beta_2 \dots \beta_\ell,$$

a string with length $k + \ell$ in $\Sigma^\star$.

Claim 7. *The above function concat* : $\Sigma^\star \to \Sigma^\star \to \Sigma^\star$ *is primitive recursive.*

Sketch of Proof. Once again, let $n = |\Sigma|$.

Consider the corresponding function $\varphi_{\text{concat}} : \mathbb{N}^2 \to \mathbb{N}$ such that, for all $\alpha, \beta \in \Sigma^\star$, if $i = \#(\alpha)$, $j = \#(\beta)$ and $k = \#(\alpha \cdot \beta) = \#(\text{concat}(\alpha, \beta))$, then $\varphi_{\text{concat}}(i, j) = k$. It is necessary and sufficient to prove that φ_{concat} is primitive recursive.

A consideration of the encoding function for strings confirms that this condition is satisfied if and only if

$$\begin{aligned} k &= \varphi_{\text{concat}}(i, j) \\ &= \#(\text{concat}(\alpha, \beta)) \end{aligned}$$

$$
\begin{aligned}
&= \#(\alpha \cdot \beta) \\
&= \#(\alpha) \times n^{|\beta|} + \#(\beta) \\
&= \#(\alpha) \times n^{\varphi_f(\#(\beta))} + \#(\beta) && \text{for the above } \textbf{\textit{length}} \text{ function } f : \Sigma^\star \to \mathbb{N} \\
&= i \times n^{\varphi_f(j)} + j.
\end{aligned}
$$

It follows by Claim 6 that the function $\varphi_f : \mathbb{N} \to \mathbb{N}$, used in the above derivation, is primitive recursive. Since n is a constant function, and addition, multiplication and exponentiation are all primitive recursive functions, it follows that the function $\varphi_{\text{concat}} : \mathbb{N}^2 \to \mathbb{N}$ is also primitive recursive, as needed to establish the claim. □

Rightmost Symbol of a String

Consider a function rtend : $\Sigma^\star \to \Sigma^\star$ such that rtend$(\lambda) = \lambda$ and such that if $\omega = \sigma_1\sigma_2 \ldots \sigma_k$ is a nonempty string in $\Sigma^\star$ with positive length k, then rtend(ω) is the rightmost symbol, σ_k, in ω.

Claim 8. *The above function rtend* : $\Sigma^\star \to \Sigma^\star$ *is primitive recursive.*

Sketch of Proof. Consider the corresponding function $\varphi_{\text{rtend}} : \mathbb{N} \to \mathbb{N}$ such that

$$\varphi_{\text{rtend}}(0) = \varphi_{\text{rtend}}(\#(\lambda)) = \#(\lambda) = 0$$

and such that, for every string $\omega = \sigma_1\sigma_2 \ldots \sigma_k \in \Sigma^\star$ with positive length k, if $\ell = \#(\omega)$ and $m = \#(\sigma_k)$ then

$$\varphi_{\text{rtend}}(\ell) = \varphi_{\text{rtend}}(\#(\omega)) = \#(\sigma_k) = m.$$

It is necessary and function to show that the function φ_{rtend} is primitive recursive. Examining the definition of the encoding function one can see that, for all $\ell \in \mathbb{N}$,

$$\varphi_{\text{rtend}}(\ell) = \begin{cases} 0 & \text{if } \ell = 0, \\ n & \text{if } \ell > 0 \text{ and } \ell \bmod n = 0, \\ \ell \bmod n & \text{otherwise.} \end{cases}$$

Since n is a positive constant, and the "mod" function is primitive recursive, it can be shown that this is a primitive recursive function (obtained using "definition by cases"), as required. □

Prefix Obtained by Deleting the Rightmost Symbol

Consider the function

$$\text{rtrunc} : \Sigma^\star \to \Sigma^\star$$

such that rtrunc$(\lambda) = \lambda$ and such that if $\omega = \sigma_1\sigma_2\ldots\sigma_k$ is a nonempty string in $\Sigma^\star$ with positive length k, then rtrunc(ω) is the string $\sigma_1\sigma_2\ldots\sigma_{k-1}$ with length $k-1$ obtained by deleting the rightmost symbol of ω.

Claim 9. *The above function rtrunc* : $\Sigma^\star \to \Sigma^\star$ *is primitive recursive.*

Sketch of Proof. Consider the corresponding function $\varphi_{\text{rtrunc}} : \mathbb{N} \to \mathbb{N}$ such that

$$\varphi_{\text{rtrunc}}(0) = \varphi_{\text{rtrunc}}(\#(\lambda)) = \#(\lambda) = 0$$

and such that, for every string $\omega = \sigma_1\sigma_2\ldots\sigma_k \in \Sigma^\star$ with positive length k, if $\ell = \#(\omega)$ and $m = \#(\sigma_1\sigma_2\ldots\sigma_{k-1})$, then

$$\varphi_{\text{rtrunc}}(\ell) = \varphi_{\text{rtrunc}}(\#(\omega)) = \#(\sigma_1\sigma_2\ldots\sigma_{k-1}) = m.$$

It is necessary and sufficient to show that the function φ_{rtrunc} is primitive recursive.

It now suffices to note that $\ell \geq \varphi_{\text{rtend}}(\ell)$ and

$$\varphi_{\text{rtrunc}}(\ell) = (\ell - \varphi_{\text{rtend}}(\ell)) \text{ div } n$$

for all $\ell \in \mathbb{N}$. Since n is a positive constant and both monus function and the "divisor" function div are primitive recursive, it follows by Claim 8 that the function φ_{rtrunc} is primitive recursive as well, as required. □

Leftmost Symbol of a String

Consider the function

$$\text{ltend} : \Sigma^\star \to \Sigma^\star$$

such that ltend$(\lambda) = \lambda$ and such that if $\omega = \sigma_1\sigma_2\ldots\sigma_k$ is a nonempty string in $\Sigma^\star$ with positive length k, then ltend(ω) is the leftmost symbol, σ_1, in ω.

Claim 10. *The above function ltend* : $\Sigma^\star \to \Sigma^\star$ *is primitive recursive.*

Sketch of Proof. Consider the corresponding function $\varphi_{\text{ltend}} : \mathbb{N} \to \mathbb{N}$ such that

$$\varphi_{\text{ltend}}(0) = \varphi_{\text{ltend}}(\#(\lambda)) = \#(\lambda) = 0$$

and such that, for every string $\omega = \sigma_1\sigma_2\ldots\sigma_k \in \Sigma^\star$ with positive length k, if $\ell = \#(\omega)$ and $m = \#(\sigma_1)$ then

$$\varphi_{\text{ltend}}(\ell) = \varphi_{\text{ltend}}(\#(\omega)) = \#(\sigma_1) = m.$$

It is necessary and function to show that the function φ_{ltend} is primitive recursive.

Consider the function $f = u_1^1 : \mathbb{N} \to \mathbb{N}$ (so that $f(\#(\omega)) = \#(\omega)$ for all $\omega \in \Sigma^\star$) and consider the function $g : \mathbb{N}^3 \to \mathbb{N}$ such that for all $t, y, s \in \mathbb{N}$,

$$g(t, y, s) = \varphi_{\mathsf{rtrunc}}(u_3^2(t, y, s)) = \varphi_{\mathsf{rtrunc}}(y).$$

Since g was obtained from primitive recursive functions, g is primitive recursive — as is f.

Now let $h_n : \mathbb{N}^2 \to \mathbb{N}$ be the function obtained from f and g by primitive recursive (so that h_n is primitive recursive too): For all $s \in \mathbb{N}$,

$$h_n(s, 0) = f(s) = s$$

and, for all $t \geq 0$,

$$h_n(s, t + 1) = g(t, h(s, t), s) = f_{\mathsf{rtrunc}}(h(s, t)).$$

It follows that

$$h_n(\#(\lambda), i) = h_n(0, i) = 0 = \#(\lambda)$$

for all $i \in \mathbb{N}$. On the other hand, it can be shown by induction on i that if $m \geq 0$ and

$$\omega = \gamma_m \gamma_{m-1} \dots \gamma_1 \gamma_0$$

then

$$h_n(\#(\omega), i) = \begin{cases} \#(\gamma_m \gamma_{m-1} \dots \gamma_i) & \text{if } 0 \leq i \leq m, \\ 0 = \#(\lambda) & \text{if } i > m. \end{cases}$$

The function $f_{\mathsf{ltend}} : \mathbb{N} \to \mathbb{N}$ such that, for all $\omega \in \Sigma^\star$,

$$f_{\mathsf{ltend}}(\#(\omega)) = \begin{cases} 0 & \text{if } \omega = \lambda, \\ h_n(\#(\omega), |\omega| - 1) & \text{otherwise} \end{cases}$$

is certainly primitive recursive too. It now suffices to note that $\varphi_{\mathsf{ltend}} = f_{\mathsf{ltend}}$ in order to complete the proof. □

Suffix Obtained by Deleting the Leftmost Symbol

Consider the function

$$\mathsf{ltrunc} : \Sigma^\star \to \Sigma^\star$$

such that $\mathsf{ltrunc}(\lambda) = \lambda$ and such that if $\omega = \sigma_1 \sigma_2 \dots \sigma_k$ is a nonempty string in $\Sigma^\star$ with positive length k, then $\mathsf{ltrunc}(\omega)$ is the string $\sigma_2 \sigma_3 \dots \sigma_k$ with length $k - 1$ obtained by deleting the leftmost symbol of ω.

Claim 11. *The above function ltrunc* : $\Sigma^\star \to \Sigma^\star$ *is primitive recursive.*

Sketch of Proof. Consider the corresponding function $\varphi_{\mathsf{ltrunc}} : \mathbb{N} \to \mathbb{N}$ such that

$$\varphi_{\mathsf{ltrunc}}(0) = \varphi_{\mathsf{ltrunc}}(\#(\lambda)) = \#(\lambda) = 0$$

and such that, for every string $\omega = \sigma_1\sigma_2\ldots\sigma_k \in \Sigma^\star$ with positive length k, if $\ell = \#(\omega)$ and $m = \#(\sigma_2\sigma_3\ldots\sigma_k)$, then

$$\varphi_{\mathsf{ltrunc}}(\ell) = \varphi_{\mathsf{ltrunc}}(\#(\omega)) = \#(\sigma_2\sigma_3\ldots\sigma_k) = m.$$

It is necessary and sufficient to show that the function $\varphi_{\mathsf{ltrunc}}$ is primitive recursive.

It now suffices to note that if

$$\omega = \sigma_1\sigma_2\ldots\sigma_k$$

is a nonempty string in $\Sigma^\star$ (so that $k > 0$), so that $\#(\omega) = \ell$ for a positive integer ℓ, then

$$\begin{aligned}\varphi_{\mathsf{ltrunc}}(\ell) &= \varphi_{\mathsf{trunc}}(\#(\omega)) \\ &= \#(\mathsf{ltrunc}(\omega)) \\ &= \#(\sigma_2\sigma_3\ldots\sigma_k) \\ &= \#(\omega) - \#(\sigma_1) \times n^{k-1} \\ &= \ell - \varphi_{\mathsf{ltend}}(\ell) \times n^{\mathsf{len}(\ell)-1}\end{aligned}$$

where $\mathsf{len} : \mathbb{N} \to \mathbb{N}$ is the function such that $\mathsf{len}(\#(\omega)) = \#(|\omega|)$ for all $\omega \in \Sigma^\star$ — and which is shown to be primitive recursive in the proof of Claim 6, above.

It follows that if $\ell \in \mathbb{N}$ then

$$\varphi_{\mathsf{ltrunc}}(\ell) = \begin{cases} 0 & \text{if } \ell = 0, \\ \ell - \varphi_{\mathsf{ltend}}(\ell) \times n^{\mathsf{len}(\ell)-1} & \text{otherwise.} \end{cases}$$

It is easily checked that $\varphi_{\mathsf{ltend}}(\ell) \times n^{\mathsf{len}(\ell)-1} \leq \ell$ whenever $\ell \geq 1$ (so that subtraction can be replaced by an application of the "monus" function in the above expression). Since the function φ_{ltend} is also primitive recursive (as shown in the proof of Claim 10), n is a constant, and since the monus, multiplication and exponentiation functions are primitive recursive, this can be used to establish that the function $\varphi_{\mathsf{ltrunc}}$ is primitive recursive, establishing the claim. □

Changing Alphabets

As in the lecture notes, suppose that Σ_1 and Σ_2 are alphabets such that

$$|\Sigma_1| = n_1 < n_2 = |\Sigma_2|.$$

Suppose, in particular, that $\Sigma_1 \subset \Sigma_2$ and that every symbol in Σ_1 has the same encoding, as an element of Σ_2, as it does as an element of Σ_1.

Consider the function upchange$_{n_1,n_2} : \mathbb{N} \to \mathbb{N}$ such that, for all $\omega \in \Sigma_1^\star$, if $\ell \in \mathbb{N}$ is the encoding of ω as a string in $\Sigma_1^\star$, then upchange$_{n_1,n_2}(\ell)$ is the encoding of ω as a string in $\Sigma_2^\star$.

- Since the encoding of λ (as a string in *any* alphabet) is 0,

$$\text{upchange}_{n_1,n_2}(0) = 0.$$

- Suppose instead that $m \geq 0$, $\gamma_m, \gamma_{m-1}, \ldots, \gamma_0 \in \Sigma_1$, and

$$\omega = \gamma_m \gamma_{m-1} \ldots \gamma_1 \gamma_0.$$

  Then the encoding of ω as a string in $\Sigma_1^\star$ is $\displaystyle\sum_{i=0}^{m} \#(\gamma_i) \times n_1^i$ while the encoding of ω as a string in $\Sigma_2^\star$ is $\displaystyle\sum_{i=0}^{m} \#(\gamma_i) \times n_2^i$.

This defines the value upchange$_{n_1,n_2}(k)$ for every non-negative integer k.

Claim 12. *If n_1 and n_2 are positive integers such that $n_1 < n_2$ then the function* upchange$_{n_1,n_2}$ *is primitive recursive.*

Sketch of Proof. Consider the primitive recursive function $h_n : \mathbb{N}^2 \to \mathbb{N}$ introduced in the proof of Claim 10: $h(0, i) = 0$ for all $i \in \mathbb{N}$. For every string $\omega \in \Sigma^\star$, where $|\Sigma| = n$, if $m \geq 0$ and

$$\omega = \gamma_m \gamma_{m-1} \ldots \gamma_1 \gamma_0$$

where $\gamma_m, \gamma_{m-1} \ldots, \gamma_1, \gamma_0 \in \Sigma$ (so that $|\omega| = m + 1$), then, for all $i \in \mathbb{N}$,

$$h_n(\#(\omega), i) = \begin{cases} \#(\gamma_m \gamma_{m-1} \ldots \gamma_i) & \text{if } 0 \leq i \leq m, \\ 0 = \#(\lambda) & \text{if } i > m. \end{cases}$$

Let $n = n_1$. The function $\widehat{h}_n : \mathbb{N}^2 \to \mathbb{N}$ such that $\widehat{h}_n(s, i) = \varphi_{\text{rtend}}(h_n(s, i))$ for all $s, i \in \mathbb{N}$ is primitive recursive, since h_n and φ_{rtend} are. A consideration of the definitions of the functions φ_{rtend} and h_n suffices to confirm that if $\omega = \gamma_m \gamma_{m-1} \ldots \gamma_1 \gamma_0$, as above, then

$$\widehat{h}_n(\#(w), i) = \begin{cases} \#(\gamma_i) & \text{if } 0 \leq i \leq m, \\ 0 & \text{otherwise.} \end{cases}$$

It follows that upchange can be defined as an iterated sum: If $s \in \mathbb{N}$ such that $s \geq 1$ (encoding a string ω as a string in $\Sigma_1^\star$, as above)

$$\text{upchange}_{n_1,n_2}(s) = \sum_{i=0}^{s} \widehat{h}_n(s,i) \times n_2^i,$$

for the primitive recursive function $\widehat{h}_n$ given above.

Thus $\text{upchange}_{n_1,n_2}$ is also a primitive recursive function, as claimed. □

For each string $\omega \in \Sigma_2^\star$, let $\phi_{n_2,n_1}(\omega)$ be the string in $\Sigma_1^\star$ obtained by removing, from ω, all the symbols that are not in Σ_1. This function could defined inductively as follows: $\phi_{n_2,n_1}(\lambda) = \lambda$ and, for all $\omega \in \Sigma_2^\star$ and $\sigma \in \Sigma_2$,

$$\phi_{n_2,n_1}(\omega \cdot \sigma) = \begin{cases} \phi_{n_2,n_1}(\omega) \cdot \sigma & \text{if } \sigma \in \Sigma_1 \text{ (that is, } \#(\sigma) \leq n_1), \\ \phi_{n_2,n_1}(\omega) & \text{otherwise.} \end{cases}$$

Now let

$$\text{downchange}_{n_2,n_1} : \mathbb{N} \to \mathbb{N}$$

such that, for all $s \in \mathbb{N}$, if s is the encoding of a string $\omega \in \Sigma_2^\star$ and $\widehat{\omega} = \phi_{n_2,n_1}(\omega)$, then $\text{downchange}_{n_2,n_1}(s)$ is the encoding of $\widehat{\omega}$, when considered as a string in $\Sigma_1^\star$.

Claim 13. *If n_1 and n_2 are positive integers such that $n_1 < n_2$ then the function* $\text{downchange}_{n_2,n_1}$ *is primitive recursive.*

Sketch of Proof. To begin, consider a function $f_{n_2,n_1} : \mathbb{N} \to \mathbb{N}$ such that $f_{n_2,n_1}(0) = 0$ and such that, if a positive integer s encodes a non-empty string

$$\omega = \gamma_m \gamma_{m-1} \ldots \gamma_1 \gamma_0 \in \Sigma_2^\star$$

(so that $m \geq 0$ and the length of ω is $m+1$) then

$$f_{n_2,n_1}(s) = \begin{cases} 0 & \text{if } \gamma_0 \notin \Sigma_2, \\ 1 & \text{otherwise.} \end{cases}$$

Since $\gamma_0 \notin \Sigma_2$ if and only if $\varphi_{\text{rtend}}(s) > n_1$, where $\varphi_{\text{rtend}} : \mathbb{N} \to \mathbb{N}$ is as given in the proof of Claim 8 — and is primitive recursive, by that claim — and since n_1 and n_2 are constants — the function f_{n_2,n_1} is also primitive recursive.

Recall as well (using $n = n_2$) the function $\widehat{h}_{n_2} : \mathbb{N}^2 \to \mathbb{N}$ that was introduced in the proof of Claim 12 such that $\widehat{h}_{n_2}(0,i) = 0$ for all $i \in \mathbb{N}$ and such that, if $s = \#(\omega)$ for a string

$$\omega = \gamma_m \gamma_{m-1} \ldots \gamma_1 \gamma_0 \in \Sigma_2^\star \tag{1}$$

and if $i \in \mathbb{N}$, then

$$\widehat{h}_{n_2}(s,i) = \begin{cases} \#(\gamma_i) & \text{if } 0 \leq i \leq m, \\ 0 & \text{otherwise.} \end{cases}$$

It was established in the proof of the above claim that this function is primitive recursive too.

The function $g_{n_2,n_1} : \mathbb{N}^3 \to \mathbb{N}$ such that for all $t, y, s \in \mathbb{N}$,

$$g_{n_2,n_1}(t,y,s) = \begin{cases} y & \text{if } \widehat{h}_{n_2}(x,t+1) = 0 \text{ or } \widehat{h}_{n_2}(x,t+1) > n_1, \\ y+1 & \text{otherwise} \end{cases}$$

is certainly primitive recursive as well. Notice that if $s = \#(\omega)$ for ω as shown at line (1), then

$$g_{n_2,n_1}(t,y,s) = \begin{cases} y & \text{if } t \geq m, \\ y & \text{if } 0 \leq t < m \text{ and } \gamma_{+1} \notin \Sigma_1, \\ y+1 & \text{otherwise.} \end{cases}$$

To continue, let $h_{\mathsf{ns}} : \mathbb{N}^2 \to \mathbb{N}$ be the function obtained from f_{n_2,n_1} and g using primitive recursion — so that h_{ns} is primitive recursive as well. Then it is not hard to prove, by induction on t, that if $s = \#(\omega)$ for ω as at line (1), then

- if $0 \leq t \leq m$ then $h_{\mathsf{ns}}(s,t)$ is the number of symbols in the sequence

  $$\gamma_0, \gamma_1, \ldots, \gamma_t$$

  that belong to Σ_1, and
- if $t \geq m$ then $h_{\mathsf{ns}}(s,t)$ is the number of symbols in the sequence

  $$\gamma_0, \gamma_1, \ldots, \gamma_m$$

  that belong to Σ_1.

Next consider the function $\overline{f} : \mathbb{N} \to \mathbb{N}$ such that $\overline{f}(0) = \overline{f}(\#(\lambda)) = 0 = \#(\lambda)$ and, if $s = \#(\omega)$ for ω as at line (1), then

$$\overline{f}(s) = \begin{cases} 0 & \text{if } \gamma_0 \notin \Sigma_1, \\ \#(\gamma_0) & \text{otherwise.} \end{cases}$$

Once again, since $\#(\gamma_0) = \varphi_{\mathsf{rtend}}(s)$, so that $\gamma_0 \notin \Sigma_1$ if and only if $\varphi_{\mathsf{rtend}}(s) > n_1$, it is not hard to see that $\overline{f}$ is primitive recursive.

Let $\overline{g} : \mathbb{N}^3 \to \mathbb{N}$ such that $\overline{g}(t, y, 0) = 0$ for all $t, y \in \mathbb{N}$ and such that, if $s = \#(\omega)$ for ω as at line (1),

$$\overline{g}(t, y, s) = \begin{cases} y & \text{if } t \geq m, \\ y & \text{if } 0 \leq t < m \text{ and } \gamma_{t+1} \notin \Sigma_1, \\ y + n_1^{h_{\mathsf{ns}}(s,t+1)} \times \#(\gamma_{t+1}) & \text{otherwise.} \end{cases}$$

Since $\#(\gamma_{t+1}) = \widehat{h}_{n_2}(s, t+1)$ if $0 \leq t < m$ it is tedious, but not difficult, to prove that $\overline{g}$ is primitive recursive too.

Let $\overline{h} : \mathbb{N}^2 \to \mathbb{N}$ be the function obtained from $\overline{f}$ and $\overline{g}$ by primitive recursion — so that $\overline{h}$ is primitive recursive too. Then it is not hard to show by induction on t that if $s = \#(\omega)$ for ω as at line (1), then

$$\overline{h}(s, t) = \begin{cases} \#(\phi_{n_2,n_1}(\gamma_t \gamma_{t-1} \ldots \gamma_1 \gamma_0)) & \text{if } 0 \leq t \leq m-1, \\ \#(\phi_{n_2,n_1}(\omega)) & \text{if } t \geq m \end{cases}$$

and where encodings as strings in $\Sigma_2^\star$ are used in the definition of the value of this function and where the function $\phi_{n_2,n_1} : \Sigma_2^\star \to \Sigma_1^\star$ is as described above, before the statement of this claim.

Since $\mathsf{downchange}_{n_2,n_1}(s) = \overline{h}(s, s)$, for s as above, it is easily argued from this that the function $\mathsf{downchange}_{n_2,n_1} : \mathbb{N} \to \mathbb{N}$ is also primitive recursive, as claimed. □