

Ma/CS 117a Handout # 9
The Parametrization and Recursion Theorems

1 The Parametrization Theorem

Theorem 1 (The Parametrization or S-M-N Theorem) For all \( m \) and \( n \) there is a primitive recursive function \( s^n_m \) such that for all \( x_1, \ldots, x_n \) and \( y_1, \ldots, y_m \)

\[
\varphi^{n+m}_i(x_1, \ldots, x_n, y_1, \ldots, y_m) = \varphi^{m}_s(i, x_1, \ldots, x_n)(y_1, \ldots, y_m)
\]

Proof: We take cases on whether \( i \in \text{TM} \). If not, we set \( s^n_m(i, x_1, \ldots, x_n) = i_0 \) where \( i_0 \) is some fixed index which codes the empty function. Otherwise, informally, our function decodes \( i \) into a machine \( M \), and produces a new machine where we insert states at the beginning to write out \( x_1B \cdots Bx_nB \) at the beginning of the tape, followed by the original tape, and then behaves like \( M \).

Corollary 2 Composition is effective, i.e. for all \( n \) and \( p \) there is a primitive recursive function \( \text{Comp}(i_1, \ldots, i_n, j) \) such that if \( j, i_1, \ldots, i_n \) are indices for the \( n \)-ary function \( g \) and the \( p \)-ary functions \( h_1, \ldots, h_n \), then \( \text{Comp}(i_1, \ldots, i_n, j) \) is an index for the \( p \)-ary composition \( f = g(h_1, \ldots, h_n) \).

Proof: Consider the function \( \beta(i_1, \ldots, i_n, j, x_1, \ldots, x_p) = \varphi^n_j(\varphi^n_{i_1}(x_1, \ldots, x_p), \ldots, \varphi^n_{i_n}(x_1, \ldots, x_p)) \). Let \( k \) be an index for this function, \( \beta = \varphi^{n+1+p}_k \). Then, by the s-m-n theorem,

\[
\beta(i_1, \ldots, i_n, j, x_1, \ldots, x_p) = \varphi^{p}_{s^n_{k+1}(k, i_1, \ldots, i_n, j)}(x_1, \ldots, x_p)
\]

so we can set \( \text{Comp}(i_1, \ldots, i_n, j) = s^n_{k+1}(k, i_1, \ldots, i_n, j) \).

Definition 3 A set \( A \subseteq \mathbb{N} \) is an index set if it is induced by a collection of (unary) partial recursive functions, i.e. there is a collection \( \chi \) of partial recursive functions such that

\[
A = \{ e : \exists f \in \chi(\varphi_e = f) \}
\]

Equivalently, \( A \) is an index set if it is invariant under functional equivalence, i.e. if \( e \in A \) and \( \varphi_e = \varphi_i \), then \( i \in A \) as well.

Theorem 4 (Rice’s Theorem) If \( A \) is an index set such that \( A \neq \emptyset \) and \( A \neq \mathbb{N} \), then \( A \) is not recursive.

Proof: Let \( \theta_0 \) be the function with empty domain. We may assume that the indices for \( \theta_0 \) are in \( A \). Let \( b \) be some index not in \( A \). Define the function:

\[
\psi(x, y, z) = \varphi_0(z) + \varphi_2(z)\varphi_2(y)
\]


and set $\psi_{x,y}(z) = \psi(x, y, z)$. Then, if $\varphi_x(y) \downarrow$ we have $\psi_{x,y} = \varphi_b$, and if $\varphi_x(y) \uparrow$ we have $\psi_{x,y} = \theta_0$; hence, the index of $\psi_{x,y}$ is in $A$ iff $\varphi_x(y) \uparrow$. Let $k$ be an index for $\psi$ so that

$$
\psi_{x,y}(z) = \psi(x, y, z) = \varphi^2_k(x, y, z) = \varphi_{s^1_1(k,x,y)}(z)
$$

Hence, setting $f(x,y) = s^1_1(k,x,y)$ we have $(x,y) \in H$ iff $f(x,y) \notin A$, showing that the complement of $A$ (and hence $A$) is not recursive.

This immediately gives the following three corollaries:

**Corollary 5** The set of indices for a given partial recursive function $f$ is not recursive.

**Corollary 6** The set $\{(i,j) : \varphi^n_i = \varphi^n_j\}$ is not recursive.

**Corollary 7** The set $\text{Tot} = \{e : \varphi_e$ is a total function$\}$ is not recursive, hence there is no effective enumeration of the (total) recursive functions.

We can use the following lemma to strengthen the s-m-n theorem.

**Lemma 8** (The Padding Lemma) For all $n$ there is a primitive recursive function $\alpha_n$ such that for all $i$ and $k$ we have $\varphi^n_i = \varphi^n_{\alpha(i,k)}$, and for $i$ fixed, $\alpha_n$ is an increasing function of $k$.

**Proof:** Informally, $\alpha_n$ encodes $i$ into a machine, and then proceeds by course-of-values recursion to find a machine with many irrelevant states which computes the same function, and such that the code of this new machine is bigger than $\alpha_n(i,j)$ for all $j < k$.

Then, by padding the $s^m_n$ functions, we get:

**Corollary 9** In the s-m-n theorem, we may assume that for $x_1, \ldots, x_n$ fixed, the function $s^m_n$ is an increasing function of $i$.

## 2 The Recursion Theorem

We give three versions of the Recursion Theorem, or Fixed Point Theorem, due to Kleene.

**Theorem 10** (Ver. 1) For $p \geq 1$ and any total recursive unary function $\alpha$ there is an index $i$ such that $\varphi_i = \varphi_{\alpha(i)}$.

**Proof:** The idea is to restrict ourselves to machines which know their indices, in the sense that they are equivalent to machines with an additional input being run with that input set equal to the code of the machine itself. Consider the function

$$
\beta(y, x_1, \ldots, x_p) = \varphi^p_{\alpha(s^r_1(y,y))}(x_1, \ldots, x_p)
$$

and let $e$ be an index for $\beta$. Then

$$
\beta(y, x_1, \ldots, x_p) = \varphi^p_e(y, x_1, \ldots, x_p) = \varphi^p_{s^1_1(e,y)}(x_1, \ldots, x_p)
$$

so for all $y$ we have $\varphi^p_{\alpha(s^r_1(y,y))} = \varphi^p_{s^1_1(e,y)}$. Hence, setting $y = e$ and letting $i = s^1_1(e, e)$ we have $\varphi^p_{\alpha(i)} = \varphi_i$.

Note that we can find a fixed point $i$ in a primitive recursive way from an index for $\alpha$:
Theorem 11 (Ver. 2) For \( p \geq 1 \) there is a primitive recursive function \( h_p \) such that if \( j \) is an index for a total recursive function \( \alpha \) then \( h_p(j) \) is a fixed point for \( \alpha \), i.e. \( \varphi^p_{h_p(j)} = \varphi^p_{\alpha(h_p(j))} \).

Proof: Let \( b \) be an index for the function

\[
\beta(j,y,x_1,\ldots,x_p) = \varphi^p_{b}(s_p^j(y,y))(x)
\]

Then

\[
\varphi^{p+2}_b(j,y,x) = \varphi^{p+1}_{s^j_{p+1}(b,j)}(y,x) = \varphi^p_{s^j_{p+1}(b,j),y}(x)
\]

so if we let \( y = s^{p+1}_1(b,j) \) and take \( h_p(j) = s^p_1(s^{p+1}_1(b,j),s^{p+1}_1(b,j)) \) we have \( \varphi^p_{h_p(j)} = \varphi^p_{\alpha(h_p(j))} \).

Theorem 12 (Ver. 3) Let \( \alpha \) be a \((p+1)\)-ary total function, and \( n \geq 1 \). Then there is a primitive recursive \( p \)-ary function \( h \) such that for all \( x_1,\ldots,x_p \)

\[
\varphi^p_{\alpha(x_1,\ldots,x_p,h(x_1,\ldots,x_p))} = \varphi^{p+n}_{h(x_1,\ldots,x_p)}
\]

Proof: Let \( b \) be an index for the function

\[
\beta(z,x_1,\ldots,x_p,y_1,\ldots,y_n) = \varphi^n_{\alpha(x_1,\ldots,x_p,s^n_{p+1}(z,z,x_1,\ldots,x_p))}(y_1,\ldots,y_n)
\]

Then

\[
\varphi^{n+p+1}_b(z,x_1,\ldots,x_p,y_1,\ldots,y_n) = \varphi^n_{s^n_{p+1}(b,z,x_1,\ldots,x_p)}(y_1,\ldots,y_n)
\]

Setting \( z = b \) and taking \( h(x_1,\ldots,x_p) = s^n_{p+1}(b,b,x_1,\ldots,x_p) \) works.

We can combine ver. 2 and ver. 3 to get an appropriate ver. 4 which we will not state.

Corollary 13 Primitive recursion is effective, i.e. for each \( n \) there is a primitive recursive function \( \text{Prim}(i,j) \) such that if \( i \) is an index for the \( n \)-ary function \( g \) and \( j \) is an index for the \((n+2)\)-ary function \( h \), then \( \text{Prim}(i,j) \) is an index for the \((n+1)\)-ary function \( f \) obtained by primitive recursion,

\[
\begin{align*}
  f(x_1,\ldots,x_n,0) &= g(x_1,\ldots,x_n) \\
  f(x_1,\ldots,x_n,y+1) &= h(x_1,\ldots,x_n,y,f(x_1,\ldots,x_n,y))
\end{align*}
\]

Proof: The idea is to consider an operation which starts with a partial approximation to \( f \) and produces a better approximation with larger domain; we then use the recursion theorem to find a fixed point of this operation, which will be the function \( f \).

Consider the mapping of partial functions \( \psi \mapsto \psi^* \) given by

\[
\psi^*(x_1,\ldots,x_n,y) = \begin{cases} 
  g(x) & \text{if } y = 0 \\
  h(x_1,\ldots,x_n,y-1,\psi(x_1,\ldots,x_n,y-1)) & \text{if } y > 0
\end{cases}
\]

By induction on \( y \) it is straightforward to show that if \( \psi \) is a fixed point for this operation, i.e. \( \psi^* = \psi \), then \( \psi = f \). So we want to effectively find a fixed point for this operation.
Let $b$ be an index for the function

\[
k(i_1, i_2, i_3, x_1, \ldots, x_n, y) = \begin{cases} 
\varphi_n^i(x_1, \ldots, x_n) & \text{if } y = 0 \\
\varphi_n^{i_2}(x_1, \ldots, x_n, y^{-1}, \varphi_{i_3}^{n+1}(x_1, \ldots, x_n, y^{-1})) & \text{if } y > 0
\end{cases}
\]

Hence if $i_1, i_2, i_3$ are set equal to indices for $g, h, \psi$, then this function computes $\psi^*$. Now,

\[
\varphi_{b}^{n+4}(i_1, i_2, i_3, x_1, \ldots, x_n, y) = \varphi_{s_3+1}^{n+1}(b, i_1, i_2, i_3)(x_1, \ldots, x_n, y)
\]

so if we let $\alpha(i_1, i_2, i_3) = s_{3}^{n+1}(b, i_1, i_2, i_3)$ then $(\varphi_{i_3}^{n+1})^* = \varphi_{\alpha(i_1, i_2, i_3)}^{n+1}$. Applying version 3 of the recursion theorem to this $\alpha$ we get a primitive recursive function $h(i, j)$ such that for all $i, j$,

\[
\varphi_{\alpha(i_1, i_2, i_3)}^{n+1} = \varphi_{h(i, j)}^{n+1}.
\]

Hence $(\varphi_{h(i, j)}^{n+1})^* = \varphi_{h(i, j)}^{n+1}$, so if we set $\text{Prim}(i, j) = h(i, j)$ then $f = \varphi_{\text{Prim}(i, j)}^{n+1}$.

We can also use a similar technique to show that minimization is effective.