# Characterizing the Grzegorczyk hierarchy by safe recursion

**Marc Wirz**

**IAM-99-005**

**November 1999**

# Characterizing the Grzegorczyk hierarchy by safe recursion

Marc Wirz *

Institut für Informatik und angewandte Mathematik, Universität Bern
Neubrückstrasse 10, CH–3012 Bern, Switzerland
wirz@iam.unibe.ch

November 22, 1999

**Abstract**

We show how the charaterization of the polytime functions by Bellantoni and Cook [1] can be extended to characterize any stage of the Grzegorzyk hierarchy above the second, thus proposing an answer to a problem posted by Clote [3]. This is done by allowing an arbitrary fixed number of distinct positions for variables instead of only two as in the original work of Bellantoni and Cook. As turned out after writing down this paper, comparable results were also proved by Bellantoni and Niggl [2].

*Keywords:* Recursion theory, Complexity theory, Grzegorzcyk.

## 1 Introduction

Bellantoni and Cook [1] characterized the polytime functions by distinguishing between two sorts of arguments of functions, called *normal* and *safe* arguments. Recursion is only allowed over normal arguments, whereas the recursively computed values must be inserted in a safe position, and function composition is defined accordingly. Related tiering notions also appeared elsewhere, e.g. in Leivant [5] or Simmons [10]. Clote [3] gives a short review of some recent results and rises the problem of relating these concepts to the Grzegorczyk hierarchy (Problem 3.102). This paper proposes an answer to that question.
Many characterizations of the Grzegorczyk hierarchy are based on controlling the depth of nested recursions, as e.g. in [6] or [9]. Even its usual definition can be seen from this angle: If we ignore the instances of bounded recursion as negligible then the functions in the $n + 1$-st level of the Grzegorczyk hierarchy are exactly

*Research supported by the Swiss National Science Foundation.

those definable with at most $n$ relevant recursions. The ramification used in the safe recursion scheme allows to control the nesting of definitions by recursion in a similar way. As each argument of a function can be used at most once as a recursion argument, the maximal depth of nested recursions is 1. If we allow $n+1$ tiers instead of only 2 tiers, and if we formulate the recursion scheme such that the recursive values must be inserted in a tier lower than the tier of the recursion argument, then the depth of the nesting is at most $n$. Thus, one can expect that such a system will produce exactly the functions in $\mathcal{E}^{n+1}$, a conjecture which we will prove below.
However, Bellantoni and Cook characterized the polytime functions, rather than the second level $\mathcal{E}^2$ of the Grzegorczyk hierarchy which equals linear time by a result of Ritchie [7]. This is, of course, due to the fact they don't use primitive recursion, but recursion on binary notation. If their definitions are adapted for unary notation of integers, their class would correspond to $\mathcal{E}^2$, as noticed by several people, cf. [3]. As soon as the exponentiation function is available, i.e. above the third level of the Grzegorzyk hierarchy, both definitions coincide, but still unary notation is more in tune with Grzegorczyk's definitions, and it is technically simpler.

# 2 The classes $B^n$

Functions in Bellantoni's and Cook's class $B$ have two sorts of inputs, called safe and normal inputs. We generalize this definition to obtain a hierarchy $B^n$, where the functions of the $n$-th level have $n$ sorts.
To distinguish between arguments of different levels we separate them by semicolons and we use the convention that a variable $x_k$ or a sequence $\overline{x_k} = x_{k,1}, \ldots, x_{k,r_k}$ of variables is always inserted in the $k$-th level when used as an argument to such a function. This convention also applies to function symbols, or even to constants, when it is necessary to make clear in which level a certain value is inserted.

**Definition 1** *For $n \geq 0$, define $B^{n+1}$ to be the smallest set of functions containing the initial functions 1.-5. and closed under safe recursion and safe composition.*

1. *The constant (zero-ary) function* $0$.
2. **(Projection)** $\pi_{k,j}(x_{n,1}, \ldots, x_{n,r_n}; \ldots; x_{0,1}, \ldots, x_{0,r_0}) = x_{k,j}$, *for* $0 \leq k \leq n$ *and* $1 \leq j \leq r_k$.
3. **(Sucessor)** $S(x_0) = x_0 + 1$.
4. **(Predecessor)** $P(0_0) = 0$, $P(x_0 + 1) = x_0$.
5. **(Conditional)** $C(0_0, y_0, z_0) = y_0$, $C(x_0 + 1, y_0, z_0) = z_0$.

6. **(Safe Recursion)**
$f(\overline{x_n}; \ldots ; \overline{x_{k+2}}; \overline{x_{k+1}}, 0_{k+1}; \overline{x_k}) = g(\overline{x_n}; \ldots ; \overline{x_{k+2}}; \overline{x_{k+1}}; \overline{x_k})$,
$f(\overline{x_n}; \ldots ; \overline{x_{k+2}}; \overline{x_{k+1}}, v_{k+1} + 1; \overline{x_k}) =$
$h(\overline{x_n}; \ldots ; \overline{x_{k+2}}; \overline{x_{k+1}}, v_{k+1}; \overline{x_k}, f(\overline{x_n}; \ldots ; \overline{x_{k+2}}; \overline{x_{k+1}}, v_{k+1}; \overline{x_k}))$,
*where $g$ and $h$ are in $B^{n+1}$.*

7. **(Safe Composition)**
$f(\overline{x_n}; \ldots ; \overline{x_0}) = h(\overline{r_n}(\overline{x_n};); \overline{r_{n-1}}(\overline{x_n}; \overline{x_{n-1}}); \ldots ; \overline{r_0}(\overline{x_n}; \ldots ; \overline{x_0}))$,
*where $\overline{r_n}, \ldots , \overline{r_0}$ and $h$ are in $B^{n+1}$.*

Thus, Bellantoni's and Cook's original class $B$ corresponds to $B^2$ in our definition, when adapted for unary notation of integers. More explicit, one obtains $B$ from $B^2$ by replacing the unary sucessor and predecessor function by their binary couterparts and the Safe Recursion scheme by an analogous scheme for Recursion on Notation and modifying the conditional such that it checks for the last bit of the binary expansion of its first argument. We will sometimes identify fuctions from $B^n$ with functions in $B^{n+k}$ which use only arguments in levels lower than $n$. $B^n$ then obviously becomes a subset of $B^{n+k}$.
The definition of the sets $\mathcal{E}^n$ is based on the following sequence of hierarchy functions $E_n$:

**Definition 2**

$$\begin{array}{lcl} E_1(x) & = & x^2 + 2 \\ E_{n+1}(0) & = & 2 \\ E_{n+1}(x + 1) & = & E_n(E_{n+1}(x)). \end{array}$$

**Definition 3** *For $n \geq 1$, define $\mathcal{E}^{n+1}$ to be the smallest set of functions containing the zero, sucessor and projection functions and $E_n$ closed under composition and the scheme of bounded recursion:*
*If $g$, $h$ and $j$ are in $\mathcal{E}^{n+1}$ then so is $f$, where*

$$\begin{array}{lcl} f(0, \overline{x}) & = & g(\overline{x}) \\ f(y + 1, \overline{x}) & = & h(y, \overline{x}, f(y, \overline{x})), \end{array}$$

*provided that*

$$f(y, \overline{x}) \leq j(y, \overline{x}).$$

We conclude this section recalling some properties of the functions $E_n$. Proofs, when not straightforward, can be found in [8].

**Remark 1** *For all $n \geq 1$:*

*i)* $E_n(x) \geq x + 1$

*ii)* $E_n(x + 1) \geq E_n(x) + 1$

*iii)* $E_n(x) \geq 2x$

*iv)* $E_n(x) \geq x^2$

*v)* $E_n(x) + E_n(y) \leq \begin{cases} E_n(x+y), & \textit{if } x, y > 0 \\ E_n(x+y) + 2, & \textit{else} \end{cases}$

*vi) If $f$ is in $\mathcal{E}^{n+1}$ then there is an integer $m$ such that*

$$f(x_1, \ldots, x_k) \leq E_n^m(\max(x_1, \ldots, x_k)).$$

# 3 $B^n$ contains $\mathcal{E}^n$

To show that $\mathcal{E}^n$ is contained in $B^n$ we can use the techniques from [1]. The proof of the first lemma even becomes more simple by using unary notation for integers. This lemma shows how definitions by several bounded recursions can be replaced with recursion in a single variable, provided that a number is given which is big enough to comprise the complexity of the computation.

**Lemma 2** *For each $f$ in $\mathcal{E}^n$ there are a function $f'$ in $B^2$ and a monotone function $e_f$ in $\mathcal{E}^n$ such that for all integers $\overline{x_0}$ and all $w$ satisfying $w_1 \geq e_f(\overline{x_0})$*

$$f'(w_1; \overline{x_0}) = f(\overline{x_0})$$

*holds.*

Proof. We proceed by induction on the definition of $f$ as a function of $\mathcal{E}^n$. To simplify notation we omit the subscripts indicating the levels, but use the convention that arguments to the left of the semicolon are in level 1 and those to the right in level 0.
If $f$ is the zero, the sucessor or a projection function then we can define $f'$ using the corresponding initial function of $B^2$. In this case we choose $e_f = 0$.
$E_{n-1}$ has a definition by $n$ applications of bounded recursion, proceeding from the sucessor and the projection functions, where each recursion is bounded by $E_{n-1}$. Since the treatment of bounded recursion does not make use of the induction hypothesis for the bounding function, we can use this method to get functions $E'_{n-1}$ and $e_{E_{n-1}}$ with the required properties.
If $f$ is defined by composition, $f(\overline{x}) = h(\overline{g}(\overline{x}))$, then we put $f'(w; \overline{x}) = h'(w; \overline{g'}(w; \overline{x}))$, where $\overline{g'}$ and $h'$ are obtained from the induction hypothesis. The functions $\overline{g}$ are bounded by monotone functions $\overline{b_g}$ from $\mathcal{E}^n$, as every function in $\mathcal{E}^n$ is. Therefore the function $e_f(\overline{x}) = e_h(\overline{b_g}(\overline{x})) + \sum_j e_{g_j}(\overline{x})$ does the job.

Finally we consider the case that $f$ is defined by bounded recursion. Let $g'$ and $h'$ in $B^2$ be given by the induction hypothesis and define

$$\begin{array}{lcl} \hat{f}(0,w;x,\overline{y}) & = & g'(w;\overline{y}) \\ \hat{f}(v+1,w;x,\overline{y}) & = & C(;x \dot{-} (w \dot{-} (v+1)), \\ & & \quad g'(w;\overline{y}), \\ & & \quad h'(w;x \dot{-} (w \dot{-} v),\overline{y},\hat{f}(v,w;x,\overline{y}))) \\ f'(w;x,\overline{y}) & = & \hat{f}(w,w;x,\overline{y}). \end{array}$$

In this definition we used the function $W(v,w;x) = x \dot{-} (w \dot{-} v)$, which is in $B^2$ as it can be defined by

$$\begin{array}{lcl} \dot{-}(0;x) & = & x \\ \dot{-}(y+1;x) & = & P(\dot{-}(y;x)) \\ W(v,w;x) & = & \dot{-}(\dot{-}(v;w);x). \end{array}$$

We also notice that the subtraction $x \dot{-} y$ is in $B^2$ where $x$ and $y$ are arguments in the levels 0 and 1 respectively. This result will be used later. Back to the proof, we define $e_f(x,\overline{y}) = e_g(\overline{y}) + e_h(x,\overline{y},j(x,\overline{y}))$. Assuming the bounding function $j$ to be monotone, $e_f$ is monotone, too. Next we show by induction on $u$ that: whenever $w \geq e_f(x,\overline{y})$ and $w - x \leq u \leq w$, then

$$\hat{f}(u,w;x,\overline{y}) = f(x-(w-u),\overline{y}).$$

When $u = w - x$ we immediately get $\hat{f}(w-x,w;x,\overline{y}) = g'(w;\overline{y}) = g(\overline{y})$. As to the induction step, we observe that

$$\begin{array}{lcl} \hat{f}(u+1,w;x,\overline{y}) & = & h'(w;x \dot{-} (w \dot{-} u),\overline{y},\hat{f}(u,w;x,\overline{y})) \\ & = & h'(w;x-(w-u),\overline{y},f(x-(w-u),\overline{y})) \\ & = & h(x-(w-u),\overline{y},f(x-(w-u),\overline{y})) \\ & = & f(x-(w-u)+1,\overline{y}) \\ & = & f(x-(w-(u+1)),\overline{y}). \end{array}$$

Thus we have found that $f'(w;x,\overline{y}) = \hat{f}(w,w;x,\overline{y}) = f(x,\overline{y})$ for all $w \geq e_f(x,\overline{y})$, and we are done. $\square$

**Theorem 3** *If $f(x_1,\ldots,x_r)$ is in $\mathcal{E}^n$ and $n \geq 2$ then $f(x_{n-1,1},\ldots,x_{n-1,r})$ is in $B^n$.*

Proof. Let $f'$ and $e_f$ be obtained by the preceeding lemma. According to remark 1.vi) there is an integer $m$ such that $e_f(x_1,\ldots,x_r) \leq E^m_{n-1}(\max_{1\leq i\leq r}(x_i))$.
We first establish that some auxiliary functions belong to $B^2$, using the same simplified notation as in the preceeding lemma:

$$\begin{array}{lcl} +(0;y) & = & y \\ +(x+1;y) & = & S(+(x;y)) \\ \cdot(x,0;) & = & 0 \\ \cdot(x,y+1;) & = & +(x;\cdot(x,y;)) \\ \max_2(x,y;) & = & +(x \dot{-} y;y) \\ \max_{k+1}(x_1,\ldots,x_{k+1};) & = & \max_2(\max_k(x_1,\ldots,x_k;),x_{k+1};), \end{array}$$

where the subtraction function $x \dot{-} y$ is defined as in the proof of the preceeding lemma. Moreover, for each $k \leq n-1$ the function $E_k^m$ belongs to $B^n$:

$$\begin{array}{lcl} E_1(x_1) & = & S(S(\cdot(x_1, x_1; ))) \\ E_{k+1}(0_{k+1}) & = & 2 \\ E_{k+1}(x_{k+1}+1) & = & E_k(E_{k+1}(x_{k+1})) \\ E_k^0(x_k) & = & x \\ E_k^{m+1}(x_k) & = & E_k(E_k^m(x_k)). \end{array}$$

We now define

$$f(x_{n-1,1}, \ldots, x_{n-1,r}) = f'(E_{n-1}^m(\max_r(x_{n-1,1}, \ldots, x_{n-1,r})); x_{n-1,1}, \ldots, x_{n-1,r}),$$

then this function $f$ belongs to $B^n$, and it agrees with the given $f$ according to the preceeding lemma. $\square$

# 4 $\mathcal{E}^n$ contains $B^n$

To prove that every function in $B^n$ belongs to $\mathcal{E}^n$ we have to show that its restriction to the arguments in the $k$-th level is bounded by a function in $\mathcal{E}^{k+1}$. To make this statement precise we define functions $e_{n,k}$ of arity $(n-k+2)$, for all $n \geq k \geq 0$.

**Definition 4** $e_{0,0}(z, x) = x + z$
$e_{k,k}(z, x) = E_k^z(x)$, *if* $k \geq 1$
$e_{n+1,k}(z, x_{n+1}, x_n, \ldots, x_k) = e_{n,k}(E_{n+1}^z(x_{n+1}), x_n, \ldots, x_k)$.

In a more reader-friendly presentation, this means $e_{n,k}(z, x_n, \ldots, x_k) = E_k^{E_{k+1}^{\cdot^{\cdot^{E_n^z(x_n)}}}(x_{k+1})}(x_k)$ if $k \geq 1$, whereas $e_{n,0}(z, x_n, \ldots, x_1, x_0) = e_{n,1}(z, x_n, \ldots, x_1) + x_0$. Evidently, $e_{n,k}$ belongs to $\mathcal{E}^{n+2}$. Further, for each fixed integer $z$ the function $(x_n, \ldots, x_k) \mapsto e_{n,k}(z, x_n, \ldots, x_k)$ is in $\mathcal{E}^{n+1}$. We first derive some properties of these functions $e_{n,k}$.

**Lemma 4** *For all $n \geq m \geq k \geq 0$:*

*i)* $e_{n,k}$ *is strictly monotonic in each of its arguments.*

*ii)* $e_{n,k}(z, x_n, \ldots, x_m^2, \ldots, x_k) \leq e_{n,k}(z+1, x_n, \ldots, x_m, \ldots, x_k)$, *if* $m \geq 1$

*iii)* $e_{n,k}(z, x_n, \ldots, 2x_m + 2, \ldots, x_k) \leq e_{n,k}(z+2, x_n, \ldots, x_m, \ldots, x_k)$, *if* $m \geq 1$

*iv)* $2e_{n,k}(z, x_n, \ldots, x_k) \leq e_{n,k}(z+1, x_n, \ldots, x_k)$, *if* $k \geq 1$

*v)* $e_{n,k}(z, e_{n,n}(\tilde{z}, x_n), x_{n-1}, \ldots, x_k) = e_{n,k}(z + \tilde{z}, x_n, \ldots, x_k)$

*vi)* $e_{n+1,k}(z, x_{n+1}, \ldots, x_{m+2}, x_{m+1}, e_{n+1,m}(z, x_{n+1}, \ldots, x_{m+2}, \widetilde{x_{m+1}}, x_m), x_{m-1}, \ldots, x_k)$
$\leq e_{n+1,k}(z, x_{n+1}, \ldots, x_{m+2}, x_{m+1} + \widetilde{x_{m+1}} + 2, x_m, x_{m-1}, \ldots, x_k)$.

Proof.
i) follows from monontonicity of the $E_k$ and from remark 1.i).
ii) is proved by induction on $n \geq m$. If $n = m \geq 1$ we use 1.iv) to obtain

$$\begin{array}{rcl} e_{n,k}(z, x_n^2, x_{n-1}, \ldots, x_k) & = & e_{n-1,k}(E_n^z(x_n^2), x_{n-1}, \ldots, x_k) \\ & \leq & e_{n-1,k}(E_n^z(E_n(x_n)), x_{n-1}, \ldots, x_k) \\ & = & e_{n-1,k}(E_n^{z+1}(x_n), x_{n-1}, \ldots, x_k) \\ & = & e_{n,k}(z+1, x_n, x_{n-1}, \ldots, x_k). \end{array}$$

The induction step follows from the induction hypothesis and from 1.i) by computing

$$\begin{array}{rcl} e_{n+1,k}(z, x_{n+1}, x_n, \ldots, x_m^2, \ldots, x_k) & = & e_{n,k}(E_{n+1}^z(x_{n+1}), x_n, \ldots, x_m^2, \ldots, x_k) \\ & \leq & e_{n,k}(E_{n+1}^z(x_{n+1}) + 1, x_n, \ldots, x_m, \ldots, x_k) \\ & \leq & e_{n,k}(E_{n+1}^{z+1}(x_{n+1}), x_n, \ldots, x_m, \ldots, x_k) \\ & = & e_{n+1,k}(z+1, x_{n+1}, x_n, \ldots, x_m, \ldots, x_k). \end{array}$$

The proof of iii) ist almost identical, using 1.iii) and 1.i) instead of 1.iv) in the case $n = m$. iv) is proved similarily again, by induction on $n \geq k$. Here the case $n = k$ follows from 1.iii) by

$$2e_{k,k}(z, x_k) = 2E_k^z(x_k) \leq E_k^{z+1}(x_k) = e_{k,k}(z+1, x_k)$$

whereas the induction step holds by

$$\begin{array}{rcl} 2e_{n+1,k}(z, x_{n+1}, x_n, \ldots, x_k) & = & 2e_{n,k}(E_{n+1}^z(x_{n+1}), x_n, \ldots, x_k) \\ & \leq & e_{n,k}(E_{n+1}^z(x_{n+1}) + 1, x_n, \ldots, x_k) \\ & \leq & e_{n,k}(E_{n+1}^{z+1}(x_{n+1}), x_n, \ldots, x_k) \\ & = & e_{n+1,k}(z+1, x_{n+1}, x_n, \ldots, x_k). \end{array}$$

v) is shown by a simple calculation in the case $n = k$ and by

$$\begin{array}{rcl} e_{n+1,k}(z, e_{n+1,n+1}(\tilde{z}, x_{n+1}), x_n, \ldots, x_k) & = & e_{n,k}(E_{n+1}^z(E_{n+1}^{\tilde{z}}(x_{n+1})), x_n, \ldots, x_k) \\ & = & e_{n+1,k}(z + \tilde{z}, x_{n+1}, x_n, \ldots, x_k) \end{array}$$

otherwise.
vi), finally, is proved by induction on $n \geq m$. For the case $n = m$ we first need a generalization of 1.v):

$$E_n^z(x) + E_n^z(\tilde{x}) \leq \begin{cases} E_n^z(x + \tilde{x}), & \text{if } x, \tilde{x} > 0 \\ E_n^z(x + \tilde{x} + 2), & \text{else.} \end{cases} \tag{$*$}$$

Its proof is by induction on $z$. If $z = 0$ the claim is obvious. As to the induction step we notice that $E_n(x), E_n(\tilde{x}) > 0$. Thus we obtain

$$\begin{array}{rcl} E_n^{z+1}(x) + E_n^{z+1}(\tilde{x}) & = & E_n^z(E_n(x)) + E_n^z(E_n(\tilde{x})) \\ & \leq & E_n^z(E_n(x) + E_n(\tilde{x})) \\ & \leq & \begin{cases} E_n^z(E_n(x + \tilde{x})) & = E_n^{z+1}(x + \tilde{x}), \quad \text{if } x, \tilde{x} > 0 \\ E_n^z(E_n(x + \tilde{x}) + 2) & \leq E_n^{z+1}(x + \tilde{x} + 2), \quad \text{else} \end{cases} \end{array}$$

by the induction hypothesis, 1.v) and 1.ii).
To prove the case $n = m$ of vi) we use v) and $(*)$ to obtain

$$\begin{aligned}
&e_{m+1,k}(z, x_{m+1}, e_{m+1,m}(z, \widetilde{x_{m+1}}, x_m), x_{m-1}, \ldots, x_k) \\
&\qquad = e_{m,k}(E^z_{m+1}(x_{m+1}), e_{m,m}(E^z_{m+1}(\widetilde{x_{m+1}}), x_m), x_{m-1}, \ldots, x_k) \\
&\qquad = e_{m,k}(E^z_{m+1}(x_{m+1}) + E^z_{m+1}(\widetilde{x_{m+1}}), x_m, x_{m-1}, \ldots, x_k) \\
&\qquad \leq e_{m,k}(E^z_{m+1}(x_{m+1} + \widetilde{x_{m+1}} + 2), x_m, x_{m-1}, \ldots, x_k) \\
&\qquad = e_{m+1,k}(z, x_{m+1} + \widetilde{x_{m+1}} + 2, x_m, x_{m-1}, \ldots, x_k).
\end{aligned}$$

The induction step follows from the same calculation, using the induction hypothesis instead of v) and $(*)$. □

**Lemma 5** *Let $f(\overline{x_n}; \ldots; \overline{x_k})$ be in $B^{n+1}$. Then there is an integer $c_f$ such that*

$$f(\overline{x_n}; \ldots; \overline{x_k}) \leq e_{n,k}(c_f, \max_i(x_{n,i}), \ldots, \max_i(x_{k,i})).$$

Proof. We prove the claim by induction on the definition of $f$ in $B^{n+1}$. It clearly holds if $f$ is an initial function of $B^{n+1}$.
If $f$ is defined by composition let $c_h$ and $c_{r_{j,i}}$ be given by the induction hypothesis . Using the abbreviations $x_j := \max_i(x_{j,i})$, $c_j := \max_i(c_{r_{j,i}})$ and $c := \max(c_h, \max_j(c_j))$ we get

$$\begin{aligned}
f(\overline{x_n}; \ldots; \overline{x_k}) &= h(\overline{r_n}(\overline{x_n}); \ldots; \overline{r_k}(\overline{x_n}; \ldots; \overline{x_k})) \\
&\leq e_{n,k}(c_h, \max_i(r_{n,i}(\overline{x_n})), \ldots, \max_i(r_{k,i}(\overline{x_n}; \ldots; \overline{x_k}))) \\
&\leq e_{n,k}(c_h, e_{n,n}(c_n, x_n), \ldots, e_{n,k}(c_k, x_n, \ldots, x_k)) \\
&\leq e_{n,k}(c + 2, e_{n,n}(c_n, x_n), \ldots, e_{n,k+1}(c_{k+1}, x_n, \ldots, x_{k+1}), x_k) \\
&\leq e_{n,k}(c + 4, e_{n,n}(c_n, x_n), \ldots, e_{n,k+2}(c_{k+2}, x_n, \ldots, x_{k+2}), x_{k+1}, x_k) \\
&\;\vdots \\
&\leq e_{n,k}(c + 2(n-k), e_{n,n}(c_n, x_n), x_{n-1}, \ldots, x_k) \\
&= e_{n,k}(2(c + n - k), x_n, \ldots, x_k).
\end{aligned}$$

Thus we are done, defining $c_f$ to be $2(c+n-k)$. (The first inequality holds according to the induction hypothesis for $h$. In the second inequality we observe

$$\begin{aligned}
\max_i(r_{j,i}(\overline{x_n}; \ldots; \overline{x_j})) &\leq \max_i(e_{n,j}(c_{r_{j,i}}, x_n, \ldots, x_j)) \\
&= e_{n,j}(c_j, x_n, \ldots, x_j).
\end{aligned}$$

The remaining inequalities hold by 4.vi and 4.iii, with $m = k, k+1, \ldots, n-1$ respectively, and the last equality follows from 4.v. If $n = k = 0$ we may not apply 4.vi, however, in this case its applications are dropped anyway.)

If $f$ is defined by recursion we let $x_j := \max_i(\overline{x_{j,i}})$ again, and we additionally define $c := \max(c_g, c_h)$ and $p_f(x, v) := (v+1) \cdot \max(x, v) + 2v$. Then we prove

$$f(\overline{x_n}; \ldots; \overline{x_{k+1}}, v; \overline{x_k}) \leq e_{n,k}(c, x_n, \ldots, x_{k+2}, p_f(x_{k+1}, v), x_k) \qquad (*)$$

by induction on $v$. If $v = 0$ then the main induction hypothesis for $g$ yields

$$\begin{aligned} f(\overline{x_n}; \ldots; \overline{x_{k+1}}, 0; \overline{x_k}) &= g(\overline{x_n}; \ldots; \overline{x_{k+1}}; \overline{x_k}) \\ &\leq e_{n,k}(c_g, x_n, \ldots, x_{k+1}, x_k) \\ &\leq e_{n,k}(c, x_n, \ldots, x_{k+2}, p_f(x_{k+1}, 0), x_k). \end{aligned}$$

The induction step follows from the induction hypothesis and 4.vi (note that $n = 0$ is impossible) by

$$\begin{aligned} & f(\overline{x_n}; \ldots; \overline{x_{k+1}}, v+1; \overline{x_k}) \\ &= h(\overline{x_n}; \ldots; \overline{x_{k+1}}, v; \overline{x_k}, f(\overline{x_n}; \ldots; \overline{x_{k+1}}, v; \overline{x_k})) \\ &\leq e_{n,k}(c_h, x_n, \ldots, x_{k+2}, \max(x_{k+1}, v), \max(x_k, e_{n,k}(c, x_n, \ldots, x_{k+2}, p_f(x_{k+1}, v), x_k))) \\ &\leq e_{n,k}(c, x_n, \ldots, x_{k+2}, \max(x_{k+1}, v), e_{n,k}(c, x_n, \ldots, x_{k+2}, p_f(x_{k+1}, v), x_k)) \\ &\leq e_{n,k}(c, x_n, \ldots, x_{k+2}, \max(x_{k+1}, v) + p_f(x_{k+1}, v) + 2, x_k) \\ &\leq e_{n,k}(c, x_n, \ldots, x_{k+2}, p_f(x_{k+1}, v+1), x_k)). \end{aligned}$$

Thus we have proved $(*)$. Now we just have to use 4.iii and 4.ii to compute

$$\begin{aligned} f(\overline{x_n}; \ldots; \overline{x_{k+1}}, v; \overline{x_k}) &\leq e_{n,k}(c, x_n, \ldots, x_{k+2}, p_f(x_{k+1}, v), x_k) \\ &\leq e_{n,k}(c, x_n, \ldots, x_{k+2}, (2 \cdot \max(x_{k+1}, v) + 2)^2, x_k) \\ &\leq e_{n,k}(c+1, x_n, \ldots, x_{k+2}, 2 \cdot \max(x_{k+1}, v) + 2, x_k) \\ &\leq e_{n,k}(c+3, x_n, \ldots, x_{k+2}, \max(x_{k+1}, v), x_k). \qquad \square \end{aligned}$$

**Theorem 6** *Every function in $B^n$ belongs to $\mathcal{E}^n$.*

Proof. By induction on the definition of $f$ in $B^n$ again. The initial functions all belong to $\mathcal{E}^n$, and $\mathcal{E}^n$ is closed under composition. Furthermore, each instance of safe recursion is an instance of bounded recursion in $\mathcal{E}^n$ by the preceeding lemma. $\square$

# Acknowledgments

I am grateful to Gerhard Jäger who introduced me to the subject and to Thomas Strahm for helpful comments.

# References


[1] Bellantoni, S., and Cook, S. A new recursion-theoretic characterization of the polytime functions. *Computational Complexity 2* (1992), 97–110.

[2] Bellantoni, S., and Niggl, K.-H. Ranking primitive recursions: The low grzegorczyk classes revisited. *SIAM Journal of Computing* (to appear).

[3] Clote, P. Computation models and function algebras. In *Handbook of Computability Theory*, E. R. Griffor, Ed. North Holland, in preparation. `http://www.tcs.informatik.uni-muenchen.de/~clote/Survey.ps.gz`.

[4] Leivant, D. A foundational delineation of computational feasability. In *Proceedings of the Sixth IEEE Conference on Logic in Computer Science (Amsterdam)*. IEEE Computer Society Press, Washington, 1991.

[5] Leivant, D. Ramified recurrence and computational complexity I: Word recurrence and poly-time. In *Feasible Mathemathics II*, P. Clote and J. Remmel, Eds. Birkhauser-Boston, New York, 1994.

[6] Parsons, C. Hierarchies of primitive recursive functions. *Zeit. Math. Logik 14* (1968), 357–376.

[7] Ritchie, R. W. Classes of predictably computable functions. *Trans. Am. Math. Soc. 106* (1963), 139–173.

[8] Rose, H. E. *Subrecursion: functions and hierarchies*, vol. 9 of *Oxford Logic Guides*. Clarendon Press, Oxford, 1984.

[9] Schwichtenberg, H. Rekursionszahlen und die Grzegorczyk-Hierarchie. *Arch. Math. Logik 12* (1969), 85–97.

[10] Simmons, H. The realm of primitive recursion. *Archive for Mathematical Logic 27* (1988), 177–188.