Pair-Based Subrecursion

Paul J. Voda
Institute of Informatics, Comenius University,
Bratislava Slovakia. (voda@fmph.uniba.sk)

May 1994

1 Introduction

The motivation for this research comes from our search for a good programming language. We have developed two declarative programming languages Trilogy I and Trilogy II [11, 15, 16, 1, 2] based on the domain of S-expressions of LISP with the only atom *Nil*. We call this domain *pairs* and write 0 instead of *Nil*. The operation *cons* yields the S-expression $cons(\boldsymbol{a}, \boldsymbol{b})$ from the S-expressions $\boldsymbol{a}$ and $\boldsymbol{b}$. In Trilogy we write $(\boldsymbol{a}, \boldsymbol{b})$ instead of $cons(\boldsymbol{a}, \boldsymbol{b})$. We let the operation *cons* to associate to the right: $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ stands for $\boldsymbol{a}, (\boldsymbol{b}, \boldsymbol{c})$.

Because there is only a single atom 0, every pair p can be uniquely written in the form 0 or $(p_1, p_2, \cdots, p_n, 0)$ for some $n \geq 1$. Thus every pair p is a *list* of pairs, and vice versa every list of pairs is a pair. It is well known that S-expressions (pairs) have other amazing coding properties. We have used the coding properties to introduce types such as cartesian products, (recursive) disjoint unions, etc. into both Trilogies [15].

We can present the domain of pairs as words over the alphabet $\{0, p\}$. The word 0 represents the atom 0 and the word $p\alpha\beta$ represents the pair (q, r) if α represents q and β represents r. We can then proceed in a straightforward way to develop a theory of recursive functions based on words representing pairs. Trilogy functions are functions over pairs and they are certainly computable. Thus the semantics of Trilogy can assign to every Trilogy function a word-recursive function.

The correspondence between pairs and natural numbers where the atom 0 becomes the number 0 and the function *cons* a *pairing function* P such that from $P(x, y) = P(v, w)$ follows $x = v$ and $y = w$ is obvious. The author has known about this correspondence right from the start of his work with pairs in 1984 [12, 13] (more about this can be found in section 12). In retrospective it is quite strange that it did not occur to him to identify the domain of pairs with the set of natural numbers until the year 1993. By identifying pairs with natural numbers the development of word-based theory of recursive functions over pairs becomes the development of recursive functions based on the pair notation rather than on the usual successor notation. The Trilogy programs are recursive functions. We are able to reason about Trilogy programs in the arithmetic. In other words, there are tremendous advantages of doing this.

The presented work concentrates on the development of the theory of recursive functions based on the pair notation. We stay in the *subrecursive* domain of primitive recursive functions and do not treat recursive functions. The reason for doing this should be obvious once one realizes that the class of functions which are feasibly computable, i.e. the class of functions computable by binary-coded Turing machines in polynomial time, is but a tiny subclass of the class of elementary functions. Elementary functions are at the bottom of the Grzegorczyk hierarchy of functions $\mathcal{E}^i$ [5] as the class $\mathcal{E}^3$. The union $\bigcup_i \mathcal{E}^i$ of Grzegorczyk hierarchy is the class of primitive recursive functions. A treatment of primitive recursive functions can be found in any text on computability, the good ones are [4, 6, 8]. The monograph *Subrecursion, Functions and Hierarchies* by H. Rose [10] should be consulted for an information about the Gregorczyk and Grzegorczyk-based hierarchies which is made use of here without a proof.

It is certainly necessary to develop a detailed theory of subrecursive function hierarchies based on pairs when one's goals are to provide a basis for the semantics of a programming language. When one wants to 'sell' this approach to the mathematicians not interested in computer programming one should naturally stress the beautiful coding properties of pairs. The author has often done this only to encounter a criticism that the (word) pair-based theory of recursive functions is not that surprising since it is quite straightfoward. We suppose that we could hear the same criticism when we identify pairs with natural numbers. A typical objection could certainly be that this is just another kind of recursion on notation. The objection is certainly true when one studies the class of primitive recursive functions via pairing. There are no difficulties and everything works out as expected. This is because just about any recursion-theoretic pairing function J which is primitive recursive will do. Any such function which is a bijection between $\mathbf{N}^2$ and $\mathbf{N}$, for which $J(0,0) = 0$, and x and y are less in a well-founded relation than $J(x,y)$ can be turned into a pairing function P suitable for our purposes by setting $P(x,y) = J(x,y) + 1$. This will make 0 the only number which is not in the range of P.

Once we turn our attention to the subelementary classes of primitive recursive functions, as we must because of our goal to create a formal basis for a feasible programming language, the situation is quite different. The numeric properties of the pairing function start to matter and it is usually quite tricky to analyze them. A pairing function with bad numeric properties will collapse the small classes of Grzegorczyk hierarchy. One is even for surprises as things are not as straightforward as the the sweeping statements about the straightforwardness of pair-based recursive function theory suggest.

The author has set up in 1988 a quite natural pair-based hierachy of functions, called $\mathcal{P}^i$ below, and correlated it to the Grzegorzcyk hierarchy via coding. This was before he identified pairs with numbers. After the identification of pairs with numbers the author quite naturally expected that his hierarchy will coincide with the Grzegorczyk hierarchy. This has turned out to be not the case for three reasons:

(i): The author originally used the natural standard recursion-theoretic function J (see [4]) offset by 1: $J(x, y) = ((x + y) \cdot (x + y + 1)) \div 2 + x + 1$. As already mentioned above, this function turned out to have bad numeric properties which collapsed the small classes of Grzegorczyk-based hierarchies (see the section 9). The author had to find a new suitable pairing function P. The pairing function P is a quite complicated numeric function. Consequently, the numeric analysis of function classes defined with the help of P is not easy (see the section **??**).

(ii): By trying to find a place for, what is now, the class $\mathcal{P}^{2.5}$ among the Grzegorczyk hierarchy the author has discovered a gap between the multiplicative $\mathcal{E}^2$ and exponential (elementary) $\mathcal{E}^3$ stages of the Grzegorczyk hierarchy. As it turned out, the discovery of the gap and its filling by a new class $\mathcal{E}^{2.5}$ solved a long standing open problem in small recursive classes (see [10]). The open problem was about the relation of open problems in small classes to those in the complexity theory. Until today it is not known whether certain small classes coincide. An open problem in small classes is usually a one of whether a certain class of functions is closed under certain operators. It was not clear how these questions correlate to the big open problems of complexity theory $P = NP$ and $P = PSPACE$. By discovering the missing stage in the Grzegorczyk hierarchy and by introducing a new Grzegorczyk-based *Bit size hierarchy* SSi with a 'shorter' iteration scheme than that provided by primitive recursion the author was able to show that the complexity problems are just of the same kind as the problems in the small classes. The result is that the class $SS2.5$ is closed under bounded minimum iff $P = NP$ and it is closed under limited primitive recursion iff $P = PSPACE$ (see [17]).

(iii): The presented work studies two pair-based Grzegorczyk-like hierarchies called the *Pair hierarchy* $\mathcal{P}^i$ the *Pair Order Hierarchy* $\mathcal{PO}^i$. The latter turns out to consist exactly of the unary functions of Grzegorczyk hierarchy $\mathcal{E}^i$. This was a surprise because the author has expected that $\mathcal{P}^i$ will correspond to $\mathcal{E}^i$ and introduced the hierarchy $\mathcal{PO}^i$ only after he realized that this was probably not the case. The Pair hierachy $\mathcal{P}^i$ has shorter iteration in the subelementary stages than $\mathcal{PO}^i$ and thus the small classes correspond to the hierarchy SSi. It might seem from the discussion in the preceding paragraph that the definition of the hierarchy SSi preceded that of $\mathcal{P}^i$. It was actually the other way round. The author has realized that the small classes of $\mathcal{P}^i$ are probably strictly included in the corresponding classes of $\mathcal{E}^i$ and set up the hierarchy SSi which was intended to be the same as $\mathcal{P}^i$ but it was presented by recursion on binary notation rather than by pairing. We have $\mathcal{P}^{i+3} = \mathcal{PO}^{i+3}$ and only $\mathcal{P}^i \subseteq \mathcal{PO}^i$ in the subelementary stages. The question of whether the inclusion is strict is another of those open problems in the small classes which we now know to be exactly as hard as the open problems in the complexity theory, namely $\mathcal{P}^{2.5} = \mathcal{PO}^{2.5}$ iff

$P = PSPACE$.

We now outline the presented work. In the section 2 we study the properties of suitable pairing functions. In the section 3 we define some pair-based functions and operators. In the section 4 we introduce a language for explicit definitions of functions via pairing. The section 5 introduces the Pair hierachy $\mathcal{P}^i$, while the section 6 studies some additional properties of the classes $\mathcal{P}^i$. The section section 7 studies the properties of pair-based functions which remain invariant for different pairing functions. The section 8 shows that for the standard recursion theoretic function J offset by one, the classes $\mathcal{P}^{i+3}$ are exactly the classes of unary functions of $\mathcal{E}^{i+3}$. In the section 9 we show that this is not the case for the subelementary stages and introduce a new pairing function P intended to correct the situation. In the section 10 we define the pair order hierarchy $\mathcal{PO}^i$ and investigate its relation to $\mathcal{P}^i$. The section 11 characterizes the Pair order hierarchy $\mathcal{PO}^i$ for the pairing function P as consisting exactly of the unary functions of the Grzegorczyk hierarchy $\mathcal{E}^i$. The section ?? introduces and develops the Bit-size hierachy SSi. In the section section ?? we are finally in the position to characterize the Pair order hierarchy $\mathcal{P}^i$ for the pairing function P as the same as the unary functions of SSi. At the elementary stage and beyond the classes $\mathcal{P}^{i+3}$ and $\mathcal{PO}^{i+3}$ for both pairing functions P and J coincide with the unary functions of $\mathcal{E}^{i+3} = SSi + 3$.

2 Pairing functions

We are interested mostly in functions over the set $\mathbf{N}$ of natural numbers. Thus when we say 'a function' without a further qualification we mean a function over $\mathbf{N}$. Likewise, all 'unsorted' quantifiers and variables range over $\mathbf{N}$. Pairing functions J used in recursion theory are mostly bijections from $\mathbf{N}^2$ to $\mathbf{N}$. Most of the time $J(0,0) = 0$. For our purposes we narrow the definition of pairing functions to exclude this case. The reason for our doing so will be seen at the end of this section.

Definition 2.1 (Pairing functions) The natural number x is an *atom* for a function f if x it is not in the range of f. The binary function p is a *pairing function* if

- $p(x,y) = p(v,w) \rightarrow x = v \wedge y = w$,
- 0 is the only *atom* of p,
- The relation x is an *immediate part* of y, i.e.

  $$\exists z(y = p(x,z) \vee y = p(z,x)),$$

  is well-founded.

Example 2.1 (Pairing function J) The function

$$J(x,y) = ((x+y) \cdot (x+y+1)) \div 2 + x + 1,$$

where $x \div y = [x/y]$ and $[x]$ is the *integer part* of the real number x, is a pairing function. The only difference to the standard definition (see, for instance [4, pp. 43-44]) is that the value of $J(x,y)$ is increased by 1 in order to make 0 an atom. That being an immediate part is a well-founded relation can be seen from $x < J(x,y)$ and $y < J(x,y)$.

Definition 2.2 (Projection functions) For a pairing function p we will denote by H_p and T_p the *projection functions* such that $H_p\, p(x,y) = x$, $T_p\, p(x,y) = y$, and $H_p(0) = T_p(0) = 0$. The *first projection* function H_p recovers the *head* of a pair while the *second projection* function T_p recovers the *tail* of a pair.

Example 2.2 (Projection functions for J) The projection functions for J are defined as follows:

$$\begin{aligned}
q_1(x) &= ([\sqrt{8x+1}] + 1) \div 2 \dot{-} 1 \\
q_2(x) &= 2x \dot{-} q_1(x)^2 \\
H_J(x) &= (q_2(x \dot{-} 1) \dot{-} q_1(x \dot{-} 1)) \div 2 \\
T_J(x) &= q_1(x \dot{-} 1) \dot{-} H_J(x)
\end{aligned}$$

where, as usual, $x \dot{-} y$ is $x - y$ if $x \geq y$ and 0 otherwise (see [4]).

Theorem 2.1 (Pair Representation) If p is a pairing function then every natural number x has a unique *pair representation* given by a term containing only 0's and applications of p.

Proof: By well-founded induction on the relation of being immediate part. Let us assume that the immediate parts of a number x have unique representation. We show that so does x. There are two cases. a) $x = 0$ then x cannot have a form $p(a, b)$ as it is an atom of p. Hence x has a unique representation. b) $x > 0$ then, since 0 is the only atom of p, we have $x = p(y, z)$ for some y and z. The natural numbers y and z are immediate parts of x and they have unique representations given by terms $a = y$ and $b = z$ from the assumption. Thus $x = p(a, b)$ is a pair representation of x. If there were another pair representation of x then it would have to be of the form $x = p(c, d)$ and then, since $a = c$ and $b = d$, the terms a and c as well as b and d would be identical. ■

Theorem 2.2 (Pair induction) For every pairing function p and a property $P(x)$ of natural numbers we have the *principle of p-pair induction*:

$$P(0) \wedge \forall v \forall w(P(v) \wedge P(w) \rightarrow P(p(v, w))) \rightarrow \forall x P(x).$$

Proof: This is proved by the well-founded induction on the immediate part relation in a straighforward way. ■

A note on representations of natural numbers: Let $s(x) = x+1$ denote the *successor* function. We call the representation of natural numbers consisting of terms formed from 0 by applications of the successor functin s the *successor* representation. Clearly, each natural number x has a unique representation of the form

$$x = \overbrace{s\, s\, \ldots\, s}^{x}(0).$$

If we define $s_0(x) = 2x$ and $s_1(x) = 2x + 1$ then the set of terms formed from 0 by applications of s_0 and s_1 yields the *binary* representation of natural numbers where each natural number x has a unique representation of the form

$$x = s_{i_0}\, s_{i_1}\, \ldots\, s_{i_{d(x)-2}}\, s_{i_{d(x)-1}}(0)$$

where each i_j (representing the j-th binary digit of x) is 0 or 1 and $i_{d(x)-1} = 1$. The function $d(x)$ is the *bit size* function yielding the number of digits in the binary representation of x for $x > 0$ and 0 otherwise.

Each of the three forms of representations (successor, binary, and pair) has its own recursion schema:

Definition 2.3 (Basic recursion schemas) We say that a function f is defined from the functions g, h, k, and j by *recursion*, *bit size recursion*, and *p-pair recursion* if in that order

$$f(0, \boldsymbol{x}) \quad = \quad g(\boldsymbol{x}) \tag{1}$$

$$f(y+1,\boldsymbol{x}) = h(y,\boldsymbol{x},f(y,\boldsymbol{x}))$$

$$\begin{aligned} f(0,\boldsymbol{x}) &= g(\boldsymbol{x}) \\ f(2y,\boldsymbol{x}) &= h(y,\boldsymbol{x},f(y,\boldsymbol{x})) \quad \text{if } y>0 \\ f(2y+1,\boldsymbol{x}) &= k(y,\boldsymbol{x},f(y,\boldsymbol{x})) \end{aligned} \tag{2}$$

$$\begin{aligned} f(0,\boldsymbol{x}) &= g(\boldsymbol{x}) \\ f(p(v,w),\boldsymbol{x}) &= h(v,w,\boldsymbol{x},f(v,\boldsymbol{x}),f(w,\boldsymbol{x})) \end{aligned} \tag{3}$$

where, here and below, $\boldsymbol{x}$ stands for a possibly empty sequence of variables $x_1, x_2, \ldots, x_n$.

The *p-pair size function* $|x|_p$ yielding the the number of p operations in the pair representation of x is defined by pair recursion:

$$\begin{aligned} |0|_p &= 0 \\ |p(x,y)|_p &= |x|_p + |y|_p + 1. \end{aligned} \tag{4}$$

We will write $|x|$ instead of $|x|_p$ if the pairing function p can be determined from the context. We will often speak about the *size* of a number x and mean the pair size of x for the pairing function p. This shorthand will not be used in potentially confusing contexts, such as for instance, '*the size of the function* $f(x)$' which could be understood either as the number $f(x)$ or its pair size.

If for the schemas of recursion and bit size recursion we have $f(\boldsymbol{x},y) \leq j(\boldsymbol{x},y)$ for all $\boldsymbol{x}$ and y and we have $|f(\boldsymbol{x},y)| \leq |j(\boldsymbol{x},y)|$ for the schema of pair recursion then the schemas are said to be *limited*.

To recursion operators belongs also the operator of *iteration* which yields a binary function f from a unary function g such that

$$\begin{aligned} f(x,y) &= g^x(y) \\ g^0(y) &= y \\ g^{x+1}(y) &= g\, g^x(y) \end{aligned} \tag{5}$$

If also $g^x(y) \leq j(x,y)$ then the iteration is *limited*.

The function classes we will study are closed under the operator of *composition* where the function f is defined from the functions g and and h_i such that

$$f(\boldsymbol{x}) = g(h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \ldots, h_m(\boldsymbol{x})).$$

Classes closed under composition and (limited) recursion are also closed under (limited) iteration.

We note that just as the induction principle justifies the schema of recursion, the principle of pair induction justifies the schema of pair recursion.

Definition 2.4 (Extended pairing functions) Given a pairing function p we extend the pairing to *n-ary pairing functions* p_n by defining

$$\begin{aligned} p_1(x) &= x \\ p_{n+1}(x_1, x_2, \ldots, x_{n+1}) &= p(x_1, p_n(x_2, \ldots, x_{n+1})). \end{aligned}$$

Note that $p_2 = p$. We will in the following drop the subscripts of p_n since they can be recovered from the number of arguments. In contexts where the pairing function p can be inferred we will write $\langle \boldsymbol{x} \rangle$ instead of $p(\boldsymbol{x})$.

The reason why we made 0 the only atom of our pairing functions was to get the pair representation of numbers. For a pairing function p we get from the uniqueness of the pair representation codes for finite sequences over $\mathbf{N}$ as follows. Every number a can be written in the form

$$\langle a_1, a_2, \ldots, a_m, 0 \rangle$$

for some $m \geq 0$ and numbers a_i. It is easy to see that both m as well as the numbers a_i are unique. Thus 0 codes the empty sequence and for $m > 0$ the above number codes the sequence $s = a_1, a_2, \ldots, a_m$. This establishes a bijection c from finite sequences over $\mathbf{N}$ to $\mathbf{N}$ where $c(s)$ is the code of the sequence s.

The length of the sequence s can be read off from its code by $Len\, c(s)$ where the function Len is defined by pair recursion:

$$\begin{aligned} Len_p(0) &= 0 \\ Len_p\langle v, w \rangle &= Len_p(w) + 1. \end{aligned}$$

We note that $Len(x) \leq |x|$.

The small change of increasing a pairing function by one gives us not only codes for finite sequences, called *lists* in computer science, but also a bijection from *(finite) binary trees* to their codes in $\mathbf{N}$. The code of the *empty* binary tree E is 0 and the code for the binary tree $Node(a, b)$ is $p(ca, cb)$ where ca and cb are the codes for the binary trees a and b respectively.

Both lists and binary trees are the basic data structures used in computer science. The programming language LISP was the first language permitting such a beautiful coding. The author was greatly inspired by LISP in his treatment of pairs (see the section 12).

For every number n there are only finitely many numbers x such that $|x|_p = n$ because there are finitely many different binary trees with n (internal) nodes. Hence for every n there is k such that for all $x \geq k$ we have $|x|_p > n$. The numbers $Ct(n)$ of different binary trees with n internal nodes are called *Catalan numbers* [7] and they satisfy the recurrent relation:

$$\begin{aligned} Ct(0) &= 1 \\ Ct(n+1) &= \sum_{i=0}^{n} Ct(i) \cdot Ct(n-i). \end{aligned}$$

We can get Catalan numbers by an explicit formula as

$$Ct(n) = \frac{1}{n+1} \left(\begin{array}{c} 2n \\ n \end{array} \right). \tag{6}$$

Thus there are exponentially many numbers with the size n.

3 Pair-based Functions And Operators

We now restrict our attention to unary functions defined with the help of pairing functions. All definitions are relative to some pairing function p. In order not to clutter the presentation we will generally drop the parameterization by p right after a definition.

Definition 3.1 (Some pair-based functions) Let p be a pairing function.

If $f(\boldsymbol{x})$ is an n-ary function $(n > 1)$ then its *unary equivalent*, denoted by f_p^+, is a unary function such that $f(\boldsymbol{x}) = f_p^+\langle\boldsymbol{x}\rangle$ and $f_p^+(x) = 0$ whenewer for all x_i we have $x \neq \langle\boldsymbol{x}\rangle$. For instance, if f is a binary function then $f(x,y) = f^+\langle x,y\rangle$ and $f^+(0) = 0$. For a ternary f we have also $f^+\langle x,0\rangle = 0$, etc. We extend the f^+ notation to include unary functions. We then have $f(x) = f_p^+\langle x\rangle = f_p^+(x)$, i.e. $f = f_p^+$.

The function $I_p(x) = x$ is the *identity* function, the function $Z_p(x) = 0$ is the *zero* function. Note that neither of the functions depends on p. On the other hand, the following functions depend on p.

The function D_p with the properties

$$\begin{aligned} D_p(0) &= D_p\langle v,0\rangle = 0 \\ D_p\langle 0,x,y\rangle &= x \\ D_p\langle\langle v,w\rangle,x,y\rangle &= y \end{aligned}$$

is the *conditional* function.

The function C_p, is the *converse* function:

$$\begin{aligned} C_p(0) &= 0 \\ C_p\langle v,w\rangle &= \langle w,v\rangle. \end{aligned}$$

The unary function $\oplus_p$ concatenating the codes of finite sequences is defined by pair recursion as

$$\begin{aligned} \oplus_p(0) &= 0 \\ \oplus_p\langle 0,y\rangle &= y \\ \oplus_p\langle\langle v,w\rangle,y\rangle &= \langle v,\oplus_p\langle w,y\rangle\rangle. \end{aligned}$$

We will write $x \oplus_p y$ for $\oplus_p\langle x,y\rangle$. We list some important properties of concatenation easily proved by pair induction:

$$\begin{aligned} x \oplus 0 &= x \\ x \oplus (y \oplus z) &= (x \oplus y) \oplus z \\ x \in (a \oplus b) &\leftrightarrow x \in a \lor x \in b \\ x \subset 0,0 \land y \subset 0,0 &\to x \oplus y = y \oplus x \end{aligned}$$

The relation $a \in b$ holds if a is a *member of the list* b and has the properties $a \notin 0$ and $a \in \langle x, y\rangle$ iff $a = x$ or $a \in y$. The relation $b \subseteq c$ is the *sublist* relation holding iff $\forall a(a \in b \to a \in c)$.

The unary function $Rt_p(x)$ defined by pair recursion 'flattens' the pair representation of the number number x by constructing a 'right leaning' tree of the same pair size as x:

$$\begin{aligned} Rt_p(0) &= 0 \\ Rt_p\langle v, w\rangle &= \langle 0, Rt_p(v) \oplus Rt_p(w)\rangle. \end{aligned}$$

The unary function $\otimes_p$ is defined by iterated concatenation as

$$\begin{aligned} \otimes_p(x) &= f_p\langle Rt_p\, H_p(x), T_p(x)\rangle \\ f_p(0) &= 0 \\ f_p\langle 0, y\rangle &= 0 \\ f_p\langle\langle v, w\rangle, y\rangle &= y \oplus_p f_p\langle w, y\rangle. \end{aligned}$$

We will write $x \otimes_p y$ for $\otimes_p\langle x, y\rangle$.

Unary *pair hierarchy functions* F^p_i are defined for $i = 2, 2.5, 3, 4, \ldots$ as follows.

$$\begin{aligned} F^p_2(x) &= C_p(x) \\ F^p_{2.5}(x) &= x \otimes_p x \oplus_p \langle 0, 0, 0\rangle \\ F^p_3(x) &= (F^p_{2.5})^{|x|_p}\langle 0, 0, 0\rangle \\ F^p_{i+4}(x) &= (F^p_{i+3})^{|x|_p}\langle 0, 0, 0\rangle. \end{aligned}$$

The reason for the strange numbering of pair hierarchy functions will be seen in the section 11. We have defined the functions $\oplus$, $\otimes$, and the pair hierarchy functions in such a way that we get a direct relation to addition, multiplication, and to the hierarchy functions of *Grzegorczyk hierarchy* [5] (see the section 8) as expressed by the theorem 3.1. Instead of using the original hierarchy functions of Grzegorczyk we use the functions E_i given by Rose [10]. The only change is that we increase the indices of his functions by one. The hierarchy functions are defined as follows:

$$\begin{aligned} E_0(x) &= x + 1 \\ E_1(x, y) &= x + y \\ E_2(x) &= x^2 + 2 \\ E_{i+3}(x) &= E^x_{i+2}(2). \end{aligned} \tag{7}$$

The following properties of hierarchy functions are proved in Rose [10] for $i = 2, 3, 4, \ldots$:

$$x < E_i(x) \tag{8}$$

$$
\begin{aligned}
E_i(x) &\leq E_i(x+1)\\
E_i(x) &\leq E_{i+1}(x)\\
E_i^y(x) &\leq E_{i+1}(x+y).
\end{aligned}
$$

Theorem 3.1 For a pairing function p we have

1. $Len(x \oplus y) = Len(x) + Len(y)$
2. $|x \oplus y| = |x| + |y|$
3. $Len\ Rt(x) = |Rt(x)| = |x|$
4. $|x \otimes y| = |x| \cdot |y|$
5. $|F_{2.5}^n(x)| = E_2^n\,|x|$
6. $|F_{i+3}^n(x)| = E_{i+3}^n\,|x|$.

Proof: The identities (1) through (3) are proved by straightforward pair induction on x. For the identity (4) we first prove $|f(x,y)| = Len(x) \cdot |y|$ by pair induction on x and then use the identity (3). The identity (5) is proved by induction on n. The case $n = 0$ is trivial. The case $n = 1$ is proved by a direct argument using the definition of $F_{2.5}$ and identities (2) and (4). For the case $n + 2$ we proceed as follows:

$$|F_{2.5}^{n+2}(x)| = |F_{2.5}\,F_{2.5}^{n+1}(x)|.$$

Now we use first the case $n = 1$ and then the induction hypothesis:

$$|F_{2.5}\,F_{2.5}^{n+1}(x)| = E_2\,|F_{2.5}^{n+1}(x)| = E_2\,E_2^{n+1}\,|x| = E_2^{n+2}\,|x|.$$

The proof of (6) uses double induction with the outer induction on i and the inner one on n where the steps are similar to those used in the proof of (5). ■

Definition 3.2 (Pair-based function operators) Let us assume in the following that p is a pairing function. The function f is defined by *p-pairing* from the functions g and h if

$$f(x) = \langle g(x), h(x)\rangle. \tag{9}$$

The function f is defined by *unary p-pair recursion* from the functions g and h if

$$
\begin{aligned}
f(0) &= 0 \\
f\langle 0, x\rangle &= g(x)\\
f\langle\langle v, w\rangle, x\rangle &= h\langle v, w, x, f\langle v, x\rangle, f\langle w, x\rangle\rangle.
\end{aligned}
\tag{10}
$$

If also $|f(z)|_p \leq |j(z)|_p$ then f is defined by *limited* unary p-pair recursion from g, h and j.

The function f is defined by *unary p-iteration* from the function g if

$$\begin{aligned} f(0) &= 0 \\ f\langle x, y\rangle &= g^{x}(y). \end{aligned} \tag{11}$$

If also $|g^x(y)|_p \leq |j\langle x, y\rangle|_p$ then f is defined by *limited* unary p-iteration from g and j.

Both unary pair recursion and iteration operators are the unary equivalents of pair recursion and iteration.

The function f is defined by *p-pair iteration* from the function g if

$$\begin{aligned} f(0) &= 0 \\ f\langle x, y\rangle &= g^{|x|_p}(y). \end{aligned} \tag{12}$$

If also $g^{|x|_p}(y) \leq j\langle x, y\rangle$ then f is defined by *limited* p-pair iteration from g and j.

4 Simple Functions

Classical recursion theory, as developed over years, has an established notation for the definition of recursive functions. The notation deals mainly with explicit definitions and with definition by cases. Since the development of recursion theory via pairing functions is new we devote this section to a language of explicit and conditional definitions of pair based functions by *clausal definitions*.

Definition 4.1 (Simple functions) For a pairing function p and a class of unary functions $\mathcal{F}$ the class of unary functions *p-simple in $\mathcal{F}$* is generated from the functions I_p, Z_p, D_p, and from the functions in $\mathcal{F}$ by the composition of unary functions and p-pairing. The class of *p-simple* functions is p-simple in the empty class $\emptyset$.

Until the end of this section we choose a fixed pairing function p and make all definitions and theorems relative to p without using the qualification by p.

Lemma 4.1 (Explicit definitions) The class of functions simple in $\mathcal{F}$ is closed under *explicit definitions* of the form $f(x) = \boldsymbol{a}$ where $\boldsymbol{a}$ is a term composed of the variable x, numerals, applications of extended pairing functions p_n and functions simple in $\mathcal{F}$.

Proof: We can assume without loss of generality that the term a does not contain applications of functions p_n other than $p = p_2$ as $p_1(\boldsymbol{b})$ can be replaced by $\boldsymbol{b}$ and the other extended pairing functions can be replaced by repeated applications of p. We also can eliminate all numerals other than 0 by replacing them with their pair representations.

The proof is by induction on the structure of the term $\boldsymbol{a}$. If $\boldsymbol{a}$ is the variable x then f is the identity function I. If $\boldsymbol{a}$ is the numeral 0 then f is the zero function Z. If $\boldsymbol{a}$ is of the form $\langle \boldsymbol{b}, \boldsymbol{c} \rangle$ then by induction hypothesis there are functions $g(x) = \boldsymbol{b}$ and $h(x) = \boldsymbol{c}$ and f is defined by the pairing of g and h. If $\boldsymbol{a}$ is of the form $g(\boldsymbol{b})$ then by induction hypothesis there is a function $h(x) = \boldsymbol{b}$ and f is defined by the composition of g and h. ■

We note that the first and second projections functions H and T are simple as they have explicit definitions $H(x) = D\langle 0, x\rangle$ and $T(x) = D\langle\langle 0,0\rangle, x\rangle$ respectively. We now permit definitions by case analysis based on *patterns*.

Definition 4.2 (Patterns) A *pattern* $\boldsymbol{a}$ is a term composed of variables, numerals, and applications of pairing functions p_n provided that no variable occurs more than once in $\boldsymbol{a}$.

Lemma 4.2 The class of functions simple in $\mathcal{F}$ is closed under definitions of the form

$$f(x) = \begin{cases} g(x) & \text{if } \exists \boldsymbol{x}\, x = \boldsymbol{a} \\ h(x) & \text{otherwise} \end{cases}$$

where $\boldsymbol{a}$ is a pattern in which no variables other then those in $\boldsymbol{x}$ occur and functions g and h are simple in $\mathcal{F}$.

Proof: Provided that we define a simple function t_a such that $t_a(x) = 0$ iff $\exists \boldsymbol{x}\, x = \boldsymbol{a}$ the function f has an explicit definition:

$$f(x) = D(t_a(x), g(x), h(x)).$$

The functions t_a are defined by induction on the structure of patterns $\boldsymbol{a}$. We assume again that the extended pairing functions and numerals other than 0 have been eliminated. If $\boldsymbol{a}$ is the numeral 0 we set $t_a(x) = I(x)$. If $\boldsymbol{a}$ is a variable then we set $t_a(x) = Z(x)$. If $\boldsymbol{a}$ is of the form $\langle \boldsymbol{b}, \boldsymbol{c} \rangle$ then we use the explicit definition

$$t_a(x) = D(x, 1, D(t_b\, H(x), t_c\, T(x), 1))$$

where the functions t_b and t_c are obtained from induction hypotheses. Note that we rely on the formula

$$\exists \boldsymbol{x}\boldsymbol{y}\ x = \langle \boldsymbol{c}(\boldsymbol{x}), \boldsymbol{d}(\boldsymbol{y}) \rangle \leftrightarrow x \neq 0 \wedge \exists \boldsymbol{x}\, H(x) = \boldsymbol{b}(\boldsymbol{x}) \wedge \exists \boldsymbol{y}\, T(x) = \boldsymbol{c}(\boldsymbol{y})$$

which holds only if $\boldsymbol{x}$ and $\boldsymbol{y}$ have no variables in common. ■

Definition 4.3 (Clausal definitions of functions) A *clause* is an expression of the form $f(\boldsymbol{a}) := \boldsymbol{b}$ where $\boldsymbol{a}$ is a pattern and $\boldsymbol{b}$ is a term similar to those used in explicit definitions but which may contain any number of variables provided they occur in the pattern $\boldsymbol{a}$. The above clause is *satisfied* for a number x iff $\exists \boldsymbol{x}\, x = \boldsymbol{a}$. We assume that the variable x does not occur in the pattern $\boldsymbol{a}$ and all of its variables are among the variables of $\boldsymbol{x}$. Obviously, the variables can be always systematically renamed if need be. Note that the clause is satisfied iff $t_a(x) = 0$. If the pattern $\boldsymbol{a}$ is of the form $\langle \boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n \rangle$ then we will omit the parentheses in the left-hand-side of a clause and write $f\boldsymbol{a} = \boldsymbol{b}$.

A *clausal definition of* f is a sequence of n clauses of the form $f(\boldsymbol{a}_i) = \boldsymbol{b_i}$ for $1 \leq i \leq n$. The function value $f(x)$ is defined to be the unique value y such that

$$\exists \boldsymbol{x}(x = \boldsymbol{a}_i \wedge y = \boldsymbol{b}_i)$$

for the first satisfied clause i. If no clause is satisfied for x then $f(x) = 0$.

We note that the order of clauses generally matters unless the patterns are *exclusive*, i.e. such that for any x at most one of the clauses is satisfied. A clausal definition is *complete* if for every x at least one of the clauses is satisfied. Without going into details we only state here that every clausal definition can be transformed to an equivalent complete definition with exclusive patterns. The clauses of such a definition can be taken as identities expressing properties of the defined function f.

Theorem 4.3 (Clausal definitions) The class of functions simple in $\mathcal{F}$ is closed under clausal definitions where all functions applied in the terms $\boldsymbol{b}_i$ (other than p_n) are simple in $\mathcal{F}$. Vice versa, if a class of functions $\mathcal{F}$ is closed under clausal definitions then it contains the functions I, Z, D and it is closed under the operators of composition of unary functions and pairing.

Proof: Let us assume that we have defined functions g_i simple in $\mathcal{F}$ such that

$$x = \boldsymbol{a}_i \rightarrow g_i(x) = \boldsymbol{b}_i$$

for all terms clauses of a clausal definition. Under this assumption the first part of the theorem is proved by induction on the number of clauses n. If $n = 1$ then we define f by the lemma 4.2 with the pattern $\boldsymbol{a}_1$ and functions g_1 and Z. If $n \geq 2$ then we obtain the function $h(x)$ from the second through last clauses by inductive assumption and use the lemma 4.2 with the pattern $\boldsymbol{a}_1$ and functions g_1 and h.

It remains to define the functions g_i. This is done by replacing in the term $\boldsymbol{b}_i$ every variable y by its *selector in the pattern* $\boldsymbol{a_i}$ of the form $s(x)$. We get a term $\boldsymbol{c}_i$ and use it in explicit definition $g_i(x) = \boldsymbol{c}_i$. The selector of y in $\boldsymbol{a_i}$ is a simple function $s(x) = \boldsymbol{d}$ defined by explicit definition from a term $\boldsymbol{d}$ containing only the variable x and applications of projection functions H and T. The term $\boldsymbol{d}$ is chosen in such a way that the identity $s(\boldsymbol{a}_i) = y$ holds. It should be obvious that the selector function is uniquely defined as each variable y occurring in the term $\boldsymbol{b}_i$ occurs in the pattern $\boldsymbol{a}_i$ in exactly one position. The position is reachable by a chain of projection functions.

For the proof of the second part of the theorem we observe that the functions I, Z, and D have following clausal definitions:

$$\begin{aligned} I(x) &:= x \\ Z(x) &:= 0 \\ D\langle 0, x, y\rangle &:= x \\ D\langle\langle v, w\rangle, x, y\rangle &:= y. \end{aligned}$$

If for $g, h \in \mathcal{F}$ the function f is defined by composition or pairing then it has clausal definitions $f(x) := g\,h(x)$ and $f(x) := \langle g(x), h(x)\rangle$ respectively. ∎

A unary function f yielding 0 or $\langle 0, 0\rangle$ is called the *characteristic function* of the predicate P if

$$P(x) \leftrightarrow f(x) = 0.$$

Note that 0 plays the role of *truth* and $\langle 0, 0\rangle$, which will be abbreviated to $\mathbf{1}$, of *falsehood*. We identify functions with predicates. This means that when we say 'the predicate P' we mean the 0-$\mathbf{1}$-valued function P. When we write $P(\boldsymbol{a})$ we mean $P(\boldsymbol{a}) = 0$, $\neg P(\boldsymbol{a})$ stands for $P(\boldsymbol{a}) = \mathbf{1}$.

We say that $\boldsymbol{A}$ is a *simple formula* if it is either a predicate application of the form $P(\boldsymbol{a})$ with $\boldsymbol{a}$ a term, or a *propositional formula* of one of the forms $\neg\boldsymbol{B}$, $\boldsymbol{B} \vee \boldsymbol{C}$, $\boldsymbol{B} \wedge \boldsymbol{C}$, or $\boldsymbol{B} \rightarrow \boldsymbol{C}$ for some simple formulas $\boldsymbol{B}$ and $\boldsymbol{C}$, or a *simply quantified formula* of a form $\exists\boldsymbol{x}(\boldsymbol{a} = \boldsymbol{p(x)} \wedge \boldsymbol{B})$ or $\forall\boldsymbol{x}(\boldsymbol{a} = \boldsymbol{p(x)} \rightarrow \boldsymbol{B})$ where $\boldsymbol{p(x)}$ is a pattern not containing other variables than from $\boldsymbol{x}$, $\boldsymbol{a}$ is a term not containing any variables from $\boldsymbol{x}$ and $\boldsymbol{B}$ a simple formula which may contain the variables from $\boldsymbol{x}$.

Theorem 4.4 (Explicit definitions of predicates) A class of functions $\mathcal{F}$ closed under clausal definitions is closed under *explicit definition of predicates*

$$P\langle \boldsymbol{x}\rangle \leftrightarrow \boldsymbol{A}$$

where $\boldsymbol{A}$ is a simple formula containing at most the variables $\boldsymbol{x}$ free and all of its functions and predicates are in $\mathcal{F}$.

Proof: By induction on the construction of $\boldsymbol{A}$. If $\boldsymbol{A}$ is $Q(\boldsymbol{a}(\boldsymbol{x}))$ then we set $P\langle \boldsymbol{x}\rangle := Q(\boldsymbol{a}(\boldsymbol{x}))$. If $\boldsymbol{A}$ is $\neg \boldsymbol{B}$ then by induction hypothesis there is a predicate $Q \in \mathcal{F}$ such that $Q\langle \boldsymbol{x}\rangle \leftrightarrow \boldsymbol{B}$ and we set $P\langle \boldsymbol{x}\rangle := D\langle Q\langle \boldsymbol{x}\rangle, \mathbf{1}, 0\rangle$. If $\boldsymbol{A}$ is of the form $\boldsymbol{B} \wedge \boldsymbol{C}$ we obtain the predicates Q and R from induction hypothesis and set $P\langle \boldsymbol{x}\rangle := D\langle Q\langle \boldsymbol{x}\rangle, R\langle \boldsymbol{x}\rangle, \mathbf{1}\rangle$. The other propositional formulas are treated similarly. If $\boldsymbol{A}$ is of the form $\exists \boldsymbol{y}(\boldsymbol{a}(\boldsymbol{x}) = \boldsymbol{p}(\boldsymbol{y}) \wedge \boldsymbol{B}(\boldsymbol{x}, \boldsymbol{y}))$ then we obtain the predicate $Q\langle \boldsymbol{x}, \boldsymbol{y}\rangle \leftrightarrow \boldsymbol{B}(\boldsymbol{x}, \boldsymbol{y})$ from induction hypothesis and set $P\langle \boldsymbol{x}\rangle := q\langle \boldsymbol{x}, \boldsymbol{a}(\boldsymbol{x})\rangle$ where $q\langle \boldsymbol{x}, \boldsymbol{p}(\boldsymbol{y})\rangle := Q\langle \boldsymbol{x}, \boldsymbol{y}\rangle$. The case when $\boldsymbol{A}$ is a simply universally quantified formula is treated similarly. ■

Theorem 4.5 (Generalized unary pair recursion) If the function class $\mathcal{F}$ is closed under clausal definitions and (limited) unary pair recursion (10) then $\mathcal{F}$ is closed under generalized (limited) unary pair recursion. The schema of *generalized unary pair recursion* has the form

$$\begin{aligned} f\langle 0, \boldsymbol{a}_1\rangle &:= \boldsymbol{b}_1 \\ f\langle\langle v, w\rangle, \boldsymbol{a}_2\rangle &:= \boldsymbol{b}_2(f\langle v, \boldsymbol{a}_2\rangle, f\langle w, \boldsymbol{a}_2\rangle) \end{aligned} \tag{13}$$

where $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ are patterns and $\boldsymbol{b}_1$ and $\boldsymbol{b}_2(y, x)$ are terms with variables from their corresponding patterns. The term $\boldsymbol{b}_2$ may also contain the *indicated* variables x and y. We also allow generalized *parameterless* unary pair recursion where the patterns $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ are both *empty*. A generalized schema of unary pair recursion is *limited* if also

$$|f\langle y, x\rangle| \leq |\boldsymbol{c}(y, x)|$$

for a term $\boldsymbol{c}$, or $|f(y)| \leq |\boldsymbol{c}(y)|$ if the schema is without parameters.

The notation such as $\boldsymbol{c}(y, x)$ and $\boldsymbol{b}_2(y, x)$ is an indication that the variables y and x may occur in the corresponding term. The term $\boldsymbol{b}_2(\boldsymbol{d}, \boldsymbol{e})$ then stands for the result of *substituting* in $\boldsymbol{b}_2$ of the term $\boldsymbol{d}$ for the variable y and of $\boldsymbol{e}$ for x.

Proof: The schema of generalized unary pair recursion with parameters (13) is reduced to unary pair recursion by clausal definitions:

$$\begin{aligned} f(0) &= 0 \\ f\langle 0, x\rangle &= g(x) \\ f\langle\langle v, w\rangle, x\rangle &= h\langle v, w, x, f\langle v, x\rangle, f\langle w, x\rangle\rangle \\ g(\boldsymbol{a}_1) &:= \boldsymbol{b}_1 \\ h\langle v, w, \boldsymbol{a}_2, y, x\rangle &:= \boldsymbol{b}_2(y, x) \end{aligned}$$

If the schema is limited we add the bounding conditions

$$\begin{aligned} |f(v)| &\leq |j(v)| \\ j\langle y, x\rangle &:= \boldsymbol{c}(y, x). \end{aligned}$$

Regarding the equivalence of the limiting condition we note that $|f(0)| = 0 \leq 0 = |j(0)|$ holds.

For a schema of parameterless pair recursion we explicitly define $f(y) = f_1\langle y, 0\rangle$ and then add a 'dummy' parameter x to the schema while replacing $f(\boldsymbol{a})$ by $f_1\langle \boldsymbol{a}, x\rangle$ everywhere in the schema to define the function f_1 by generalized pair recursion with a parameter. ■

For instance, the concatenation of lists is defined by the limited generalized unary pair recursion as follows:

$$\begin{aligned} 0 \oplus y &:= y \\ \langle v, w\rangle \oplus y &:= \langle v, w \oplus y\rangle \\ |x \oplus y| &= |x| + |y| \leq |\langle x, y\rangle|. \end{aligned}$$

We will often insert chains of identities and inequalities to bounding conditions to derive the limiting term as the last one in the chain.

5 Pair Hierarchy

Definition 5.1 (Pair hierarchy) For a pairing function p and a unary function f the class of unary functions $\mathcal{P}_p(f)$ is generated from the functions f, T_p, and C_p by the composition of unary functions, p-pairing, and limited p-pair iteration (12).

The *Pair hierarchy* $\mathcal{P}_p^i$ is defined as $\mathcal{P}_p^i = \mathcal{P}_p(F_i^p)$ for for $i = 2, 2.5, 3, 4, \ldots$.

Note that, although we intend the Pair hierarchy to consist of computable functions, the definition does not require that the pairing function p be recursive. The functions in Pair hierarchy are in general *recursive in p*. Until the end of this section we choose a fixed pairing function p and make all definitions and theorems relative to p without showing the parameterization.

Theorem 5.1 The classes $\mathcal{P}^i$ of Pair hierarchy are closed under clausal definitions.

Proof: Let i be one of 2, 2.5, 3, ... Since the class $\mathcal{P}^i$ is closed under unary composition and pairing it suffices to derive the functions I, Z, and D in $\mathcal{P}^i$ as then the class of functions simple in $\mathcal{P}^i$ will coincide with $\mathcal{P}^i$ and the theorem 4.3 will apply. The identity function is defined by composition as $I(x) = C\,C(x)$. The zero function is derived in $\mathcal{P}^i$ by

$$\begin{aligned} Z(x) &= f\langle I(x), I(x)\rangle \\ f(0) &= 0 \\ f\langle x, y\rangle &= T^{|x|}(y) \\ |T^{|x|}(y)| &\le |T\langle x, y\rangle|. \end{aligned}$$

Here Z is obtained first by pairing and then by composition. By induction on n we can show $T^n(0) = 0$ and use this in the proof by pair induction on x of $T^{|x|}(x) = 0$. Thus $Z(x) = f\langle x, x\rangle = T^{|x|}(x) = 0$. The conditional function D is derived in $\mathcal{P}^i$ by

$$\begin{aligned} D(x) &= H\,f(x) \\ H(x) &= T\,C(x) \\ f(0) &= 0 \\ f\langle x, y\rangle &= g^{|x|}(y) \\ |g^{|x|}(y)| &\le |g\langle x, y\rangle| \\ g(x) &= \langle T(x), T(x)\rangle. \end{aligned}$$

We note that $g^0\langle v, w\rangle = \langle v, w\rangle$ and $g^{n+1}\langle v, w\rangle = g\langle v, w\rangle = \langle w, w\rangle$. ■

We will often use in explicit definitions pair iteration of the form $g^{|\boldsymbol{a}|}(\boldsymbol{b})$ where $\boldsymbol{a}$ and $\boldsymbol{b}$ are terms as an abbreviation for $f\langle \boldsymbol{a}, \boldsymbol{b}\rangle$ where f is defined separately by (limited) pair iteration.

Lemma 5.2 We have

1. $\oplus \in \mathcal{P}^i$ for $i = 2, 2.5, 3, 4, \ldots$.
2. $\otimes, F_{2.5} \in \mathcal{P}^i$ for $i = 2.5, 3, 4, \ldots$.
3. $F_3, F_4, \ldots, F_i \in \mathcal{P}^i$ for $i = 3, 4, \ldots$.

Proof: (1): We first define an auxiliary function $\mathit{Reva} \in \mathcal{P}^i$:

$$\begin{array}{rcl} \mathit{Reva}\langle x, a\rangle & := & T\, f^{|x|}\langle x, a\rangle \\ f\langle 0, a\rangle & := & \langle 0, a\rangle \\ f\langle\langle v, w\rangle, a\rangle & := & \langle w, v, a\rangle \\ |f^{|x|}(v)| & = & v \le |v|. \end{array}$$

It is not difficult to see that we have $\mathit{Reva}\langle 0, a\rangle = a$ and

$$\mathit{Reva}\langle\langle v, w\rangle, a\rangle = \mathit{Reva}\langle w, v, a\rangle.$$

Using this in a simple pair induction proof on x we get

$$\mathit{Reva}\langle x, a \oplus b\rangle = \mathit{Reva}\langle x, a\rangle \oplus b. \tag{14}$$

The function Rev is defined in $\mathcal{P}^i$ by explicit definition $\mathit{Rev}(x) := \mathit{Reva}\langle x, 0\rangle$. Using (14) we easily derive the following properties of Rev:

$$\begin{array}{rcl} \mathit{Rev}(0) & = & 0 \\ \mathit{Rev}\langle x, y\rangle & = & \mathit{Rev}(y) \oplus \langle x, 0\rangle \\ \mathit{Reva}\langle x, a\rangle & = & \mathit{Rev}(x) \oplus a. \end{array} \tag{15}$$

We can see that $\mathit{Rev}(x)$ reverses the list x. Computer scientists will recognize in the function Reva the efficient accumulator version of the function Rev. An easy pair induction argument shows $\mathit{Rev}\ \mathit{Rev}(x) = x$. This, and (15), justify the clausal definition of $\oplus$ in $\mathcal{P}^i$:

$$x \oplus y := \mathit{Reva}\langle \mathit{Rev}(x), y\rangle.$$

(2): Functions $\otimes$ and $F_{2.5}$ are derived in the classes $\mathcal{P}^i$ for $i > 2$ as follows

$$\begin{array}{rcl} x \otimes y & := & T\, f^{|x|}\langle y, 0\rangle \\ f\langle y, a\rangle & := & \langle y, y \oplus a\rangle \\ |f^{|x|}(v)| & \le & |g\langle x, v\rangle| \\ g\langle x, y, a\rangle & := & \langle y, F_i(x \oplus y) \oplus a\rangle \\ F_{2.5}(x) & := & x \otimes x \oplus \langle 0, 0, 0\rangle. \end{array}$$

The bounding condition follows from

$$
\begin{aligned}
|f^{|x|}\langle y, a\rangle| &= |y| + 1 + |x| \cdot |y| + |a| \le |y| + 1 + |x \oplus y|^2 + |a| \le \\
&\quad |y| + 1 + E_2\,|x \oplus y| + |a| \le |y| + 1 + E_j|x \oplus y| + |a| = \\
&\quad |\langle y, F_i(x \oplus y) \oplus a\rangle|
\end{aligned}
$$

where $j = 2$ if $i = 2.5$ and $j = i$ otherwise.

(3): The functions F_k $(3 \le k \le i)$ are derived in $\mathcal{P}^i$ one after another by limited pair iteration $F_k(x) = F_j^{|x|}\langle 0, 0, 0\rangle$ where for $k > 3$ we take $j = k - 1$ and for $k = 3$ we take $j = 2.5$. The bounding condition is

$$
|F_j^{|x|}(y)| = E_{k-1}^{|x|}\,|y| \le E_k(|y| + |x|) = E_k(|y \oplus x|) \le E_i(|y \oplus x|) = |F_i(y \oplus x)|
$$

by the properties (8) of hierarchy functions E_i and the theorem 3.1. ■

Lemma 5.3 We have the following bounds on functions in the Pair hierarchy:

1. if $f \in \mathcal{P}^2$ then there are natural numbers a and b such that $|f(x)| \le a \cdot |x| + b$ for all x,

2. if $f \in \mathcal{P}^{2.5}$ then there is a polynomial $q(x)$ such that $|f(x)| \le q\,|x|$ for all x.

3. if $f \in \mathcal{P}^{i+3}$ then there is a natural number k such that $|f(x)| \le E_{i+3}^k\,|x|$ for all x.

Proof: The proof of all three properties is by induction on the formation of functions in the appropriate class $\mathcal{P}^i$. We use the properties (8) of hierarchy functions E_i and the theorem 3.1.

For the basic function C we get the bounds usable in properties (1) through (3) as follows:

$$
|C(x)| \le 1 \cdot |x| + 0 = E_{i+3}^0\,|x|.
$$

The function T has the same bounds. For the pair hierarchy functions F_i we have in the case $i = 2.5$

$$
|F_{2.5}(x)| = |x \otimes x \oplus \langle 0, 0, 0\rangle| = |x|^2 + 2 \le |x|^2 + 2
$$

and $|F_i(x)| = E_i\,|x| \le E_i^1\,|x|$ in the case $i \ge 3$.

The bounds on the function f derived by the limited pair iteration are obtained in all three properties from induction hypothesis as the bounds on the pair size of the limiting function $j(x)$.

If f is defined by composition we show the induction argument for the property (3):

$$
|f(x)| = |g\,h(x)| \le E_i^m\,|h(x)| \le E_i^m\,E_i^n\,|x| = E_i^{m+n}\,|x|
$$

where the numbers m and n are obtained from inductive hypotheses. The properties (1) and (2) are even simpler as they reduce to the composition of linear forms (polynomials of degree at most 1) for the property (1) and of polynomials for the property (2).

If f is defined by pairing we have for the property (3)

$$\begin{aligned} |f(x)| &= |\langle g(x), h(x)\rangle| = |g(x)| + |h(x)| + 1 \leq E^m_{i+3}|x| + E^n_{i+3}|x| + 1 \leq \\ &\leq E^{max(m,n)}_{i+3}|x| + E^{max(m,n)}_{i+3}|x| + 1 = 2E^{max(m,n)}_{i+3}\,|x| + 1 \end{aligned}$$

where m and n are obtained from induction hypotheses. We have

$$2a + 1 \leq a^2 + 2 = E_2(a) \leq E_{i+3}(a).$$

Hence

$$|f(x)| \leq 2E^{max(m,n)}_{i+3}\,|x| + 1 \leq E_{i+3}\,E^{max(m,n)}_{i+3}\,|x| = E^{max(m,n)+1}_{i+3}\,|x|.$$

The argument for the properties (1) and (2) goes by a similar manipulation of polynomials. ■

Theorem 5.4 The Pair hierarchy is a proper hierarchy as we have

$$\mathcal{P}^2 \subset \mathcal{P}^{2.5} \subset \mathcal{P}^3 \subset \mathcal{P}^4 \subset \dots.$$

Proof: To obtain the inclusions $\subseteq$ it is sufficient to show $F_{2.5} \in \mathcal{P}^3$ and $F_{i+3} \in \mathcal{P}^{i+4}$. This follows from the lemma 5.2. By the lemma 5.3 the pair sizes of functions of $\mathcal{P}^2$ are bounded by a linear form so the function $\otimes$ is not in $\mathcal{P}^2$ and the first inclusion is strict. The pair size of the function F_3, which is basically iterated multiplication, grows faster than any polynomial in size of x. This shows that the second inclusion is strict.

For the remaining inclusions we know from the lemma 5.3 (3) that for every function $f \in \mathcal{P}^{i+3}$ there is k such that for all x we have

$$|f(x)| \leq E^k_{i+3}\,|x| < E^k_{i+3}\,|x| + 1 \leq E_{i+3}\,E^k_{i+3}\,|x| = E^{k+1}_{i+3}\,|x|.$$

Thus for all x such that $|x| \geq k + 1$

$$|f(x)| < E^{k+1}_{i+3}\,|x| \leq E^{|x|}_{i+3}\,|x| \leq E_{i+4}\,(2|x|) = |F_{i+4}(x \oplus x)|.$$

In other words, the pair size of the function $F_{i+4}(x \oplus x)$ eventually majorizes the pair size of every $\mathcal{P}^{i+3}$ function. If $F_{i+4} \in \mathcal{P}^{i+3}$ then for $f(x) = F_{i+4}(x \oplus x) \in \mathcal{P}^{i+3}$ we would get a contradiction. Hence $\mathcal{P}^{i+3} \subset \mathcal{P}^{i+4}$. ■

6 Some Properties of Pair Hierarchy

Throughout this section we assume a fixed pairing function p and drop the parameterization by p. We have defined the Pair hierarchy by limited pair iteration although limited unary pair recursion seems more natural for this purpose. Unfortunately, it seems that we would get a larger class $\mathcal{P}^2$ had we used pair recursion instead of iteration. This would wreck the characterization of the Pair hierarchy in the section **??**.

The lemma 6.1 shows sufficient conditions for a function class to be closed under limited unary pair recursion. The theorem 6.2 then shows that pair recursion and pair iteration are equivalent in all classes of the Pair hierarchy but $\mathcal{P}^2$. For the proof of the lemma 6.1 we derive in the class $\mathcal{P}^2$ the function Rt yielding the right-leaning trees. We also need two auxiliary functions $Bsize$ and $Post$ both of which are in the class $\mathcal{P}^2$.

We derive $Rt \in \mathcal{P}^2$ as follows:

$$\begin{aligned} Rt(x) &= H\, f^{|\langle x,x\rangle|}\langle 0,x,0\rangle \\ |f^{|x|}(v)| &\leq |g\langle x,v\rangle| \\ f\langle a,\langle v,w\rangle,s\rangle &:= \langle\langle 0,a\rangle,v,w,s\rangle \\ f\langle a,0,s\rangle &:= \langle a,s\rangle \\ g\langle x,a,s\rangle &= \langle s,a,s\rangle. \end{aligned}$$

By pair induction on x we easily prove

$$\forall a \forall s\, f^{|\langle x,x\rangle|}\langle a,x,s\rangle = \langle Rt(x) \oplus a, s\rangle$$

from which we can see that the derivation of Rt is correct, i.e. that

$$H\, f^{|\langle x,x\rangle|}\langle 0,x,0\rangle = H\langle Rt(x) \oplus 0, 0\rangle = Rt(x).$$

Computer scientists will recognize in the 'second argument' s of f a *stack* holding the parts of the original argument of x as it is being broken down by the iteration. That the bounding condition holds can be seen from the property $|f\langle a,s\rangle| \leq |\langle 0,a,s\rangle|$.

The application $Bsize\langle x,b\rangle$, which is for the duration of this proof abbreviated to x_b, yields x if it is bounded by b, i.e. $x_b = x$ if $|x| \leq |b|$ and $x_b = b$ otherwise. The function $Bsize$ is derived in $\mathcal{P}^2$ as follows

$$\begin{aligned} x_b &:= D(T^{|b|}Rt(x),x,b) \\ |T^{|b|}(x)| &\leq |x|. \end{aligned}$$

The function $Post(x)$ yielding a list of 0's and $\mathbf{1}$'s corresponding to the *polish postfix notation* of x is defined by pair recursion as $Post(0) = 0,0$ and $Post\langle v,w\rangle = Post(v) \oplus Post(w) \oplus \langle \mathbf{1},0\rangle$. We have $Len\, Post(x) = |\langle x,x\rangle|$ and $|Post(x)| = 3 \cdot |x| + 1$ which can be proved by a straightforward p-pair induction.

The function $Post$ is derived in $\mathcal{P}^2$ as follows:

$$\begin{array}{rcl} Post(x) & := & H\ f^{|\langle x,x\rangle|}\langle 0,x,0\rangle \\ |f^{|y|}(x)| & \leq & |g\langle y,x\rangle| \\ f\langle a,\langle v,w\rangle,s\rangle & := & \langle\langle \mathbf{1},a\rangle,w,v,s\rangle \\ f\langle a,0,s\rangle & := & \langle\langle 0,a\rangle,s\rangle \\ g\langle y,a,s\rangle & := & \langle a,s,s,s,s,s\rangle. \end{array}$$

The iteration of f uses a stack as the second argument of f. A pair on the top of it is popped off the stack and its components are pushed back. This adds $\mathbf{1}$ to the result being accumulated in the first argument. A 0 on the top of the stack is popped from it and added to the result. This informal reasoning is formally proved by p-pair induction on x with the induction formula

$$\forall a \forall s\ f^{Len\ Post(x)}\langle a,x,s\rangle = \langle Post(x)\oplus a,s\rangle.$$

From this we can see that

$$H\ f^{|\langle x,x\rangle|}\langle 0,x,0\rangle = H\ f^{Len\ Post(x)}\langle 0,x,0\rangle = H\langle Post(x)\oplus 0,0\rangle = Post(x).$$

In order to verify the bounding condition we observe that each iteration of $f\langle a,s\rangle$ decreases the *weight* of the stack $w(s) = \sum_{v\in s}(2\cdot|v|+1)$. Thus there cannot be more than $w(s) \leq 2\cdot|s|$ iterations of f before the stack is emptied. Since $|f\langle a,s\rangle| \leq |\langle a,s\rangle| + 2$ we get

$$|f^{|y|}\langle a,s\rangle| \leq |\langle a,s\rangle| + 4\cdot|s| \leq |\langle a,s,s,s,s,s\rangle| = |g\langle y,a,s\rangle|.$$

Lemma 6.1 Assume that a class of functions $\mathcal{F}$ is closed under clausal definitions and limited pair iteration. Further assume that f is defined by limited unary pair recursion (10) from the functions $g,h,j\in\mathcal{F}$. If there is a function $m_1\in\mathcal{F}$ such that for $m(y,x) = |m_1\langle y,x\rangle|$ we have

1. $m(v,x)+m(w,x)\leq m(\langle v,w\rangle,x)$,
2. $|f\langle y,x\rangle| + |y| + 2 \leq m(y,x)$

then $f\in\mathcal{F}$.

Proof: By the theorem 4.3 $\mathcal{F}$ is closed under pairing and composition of unary functions. Futhermore, the functions T and $C\langle x,y\rangle := \langle y,x\rangle$ are in $\mathcal{F}$. Hence $\mathcal{P}^2\subseteq\mathcal{F}$.

Let f, m_1, and m be as in the assumptions of the lemma and consider the function $k\in\mathcal{F}$:

$$\begin{array}{rcl} k\langle x,b,\langle 0,p\rangle,s\rangle & := & \langle x,b,p,\langle\langle g(x),0\rangle,s\rangle_b\rangle \\ k\langle x,b,\langle \mathbf{1},p\rangle,\langle fw,w\rangle,\langle fv,v\rangle,s\rangle & := & \langle x,b,p,\langle\langle h\langle v,w,x,fv,fw\rangle,v,w\rangle,s\rangle_b\rangle. \end{array}$$

By pair induction on y one can show that

$$\forall p \forall s\, (|s| + m(y,x) \leq |b| \rightarrow \\ k^{Len\ Post(y)}\langle x, b, Post(y) \oplus p, s\rangle = \langle x, b, p, \langle f\langle y,x\rangle, y\rangle, s\rangle). \tag{16}$$

We urge the reader to do the proof as it is a nice example of, what is called in computer science, the correctness proof of k. The induction argument uses the property (2) of m both in the basis and induction steps, and the property (1) in the induction step. The iteration of k fully uses up the postfix form of y to assemble pairs of the form $\langle f\langle z,x\rangle, z\rangle$ on the stack which is the fourth argument of k. Here z is always a part of the original argument y.

The property (16) holds also without the assumption and without the bounding function s_x applied in k. The assumption on the pair size of b, together with the properties (1) and (2) of m, guarantee that the size of the stack does not exceed that of b, i.e. $s_b = s$. The iteration of k in the following clauses is bounded by the function n and the function f_1 is derived in the class $\mathcal{F}$:

$$\begin{aligned} f_1\langle y,x\rangle &:= H\,H\,T\,T\,T\,k^{|\langle y,y\rangle|}\langle x, m_1\langle y,x\rangle, Post(y), 0\rangle \\ |k^{|y|}(v)| &\leq |n\langle y,v\rangle| \\ n\langle y,x,b,p,s\rangle &:= \langle x,b,p,b\rangle. \end{aligned}$$

It remains to show that $f_1 = f$. Since $|0| + m(y,x) = m(y,x) = |m_1\langle y,x\rangle|$, the assumption of the formula (16) is satisfied for $b = m_1\langle y,x\rangle$ and we have

$$\begin{aligned} f_1\langle y,x\rangle = H\,H\,T\,T\,T\,k^{|\langle y,y\rangle|}\langle x, m_1\langle y,x\rangle, Post(y), 0\rangle = \\ H\,H\,T\,T\,T\,k^{Len\ Post(y)}\langle x, m_1\langle y,x\rangle, Post(y)\oplus 0, 0\rangle = \\ H\,H\,T\,T\,T\langle x, m_1\langle y,x\rangle, 0, \langle f\langle y,x\rangle, y\rangle, 0\rangle = f\langle y,x\rangle. \end{aligned}$$

Since also $f_1(0) = 0 = f(0)$ we have $f_1 = f$. ■

Theorem 6.2 The classes $\mathcal{P}^i$ where $i = 2.5, 3, 4, \ldots$ are generated from the functions C, T, and F_i by composition of unary functions, pairing, and limited unary pair recursion (10).

Proof: Let i be as in the assumption of the theorem and let $\mathcal{F}^i$ denote the inductively defined class of the theorem.

$\mathcal{F}^i \subseteq \mathcal{P}^i$: It is sufficient to show the class $\mathcal{P}^i$ closed under limited unary pair recursion. Let f be defined by limited unary pair recursion from $g, h, j \in \mathcal{P}^i$. Thus we have $|f\langle y,x\rangle| \leq |j\langle y,x\rangle|$. The lemma 6.1 applies provided we can find $m_1 \in \mathcal{P}^i$ such that the assumptions (1) and (2) of the lemma hold for $m(y,x) = |m_1\langle y,x\rangle|$. We can assume that the size of the bounding function j is monotone in the size of its first argument, i.e. that

$$|z| \leq |y| \rightarrow |j\langle z,x\rangle| \leq |j\langle y,x\rangle|.$$

If this is not the case for $i \geq 3$ we can always find a k such that $|j(v)| \leq E_i^k\,|v| = |F_i^k(v)|$ by the lemma 5.3 and replace j by F_i^k. The same lemma gives us for $i = 2.5$ a polynomial $q(v)$ such that $|j(v)| \leq q\,|v|$. Take the function $r(v) = q_1(v)$ where q_1 is as q but with every addition replaced by $\oplus$, multiplication by $\otimes$ and a constant c by a number c_1 such that $|c_1| = c$. Clearly $r \in \mathcal{P}^{2.5}$ and we have $|j(v)| \leq q\,|v| = |r(v)|$. Thus again j can be replaced by r if need be.

Take the $\mathcal{P}^i$ function

$$m_1\langle y, x\rangle := (j\langle y, x\rangle \oplus y \oplus \langle 0, \mathbf{1}\rangle) \otimes (y \oplus \mathbf{1}) \tag{17}$$

then for m as above we have

$$m(y, x) = (|j\langle y, x\rangle| + |y| + 2) \cdot (|y| + 1).$$

We leave it to the reader to convince himself that the assumptions (1) and (2) of the lemma 6.1 hold.

$\mathcal{P}^i \subseteq \mathcal{F}^i$: We first show the class $\mathcal{F}^i$ closed under clausal definitions. Similarly as in the proof of the theorem 5.1, we need to show $I, H, Z, D \in \mathcal{F}^i$. The first two functions are obtained as in the mentioned proof. The zero function is derived as

$$\begin{aligned}
Z(x) &= f\langle I(x), I(x)\rangle \\
f(0) &= 0 \\
f\langle 0, x\rangle &= I(x) \\
f\langle\langle v, w\rangle, x\rangle &= h\langle v, w, x, f\langle v, x\rangle, f\langle w, x\rangle\rangle \\
|f(v)| &\leq |I(v)| \\
h(z) &= T\,T\,T\,T\,T(z)
\end{aligned}$$

where Z is obtained first by pairing and then by composition while h is obtained by a repeated composition. We note that the application of h in the fourth identity yields $T\,f(w, x)$. By pair induction on y we can show that $f\langle y, x\rangle = T^{Len(y)}(x)$. By another pair induction on x we can then show $T^{Len(x)}(x) = 0$ from which we get $Z(x) = f\langle x, x\rangle = T^{Len(x)}(x) = 0$.

The reader will convince himself that the conditional D is derived in $\mathcal{F}^i$ by

$$\begin{aligned}
D(0) &= 0 \\
D\langle 0, x\rangle &= H(x) \\
D\langle\langle v, w\rangle, x\rangle &= h\langle v, w, x, D\langle v, x\rangle, D\langle w, x\rangle\rangle \\
|D(v)| &\leq |I(v)| \\
h(z) &= T\,H\,T\,T(z).
\end{aligned}$$

To finish the proof it is sufficient to show the class $\mathcal{F}^i$ closed under limited pair iteration (12). For that we derive the functions $\oplus$ and Rt in $\mathcal{F}^i$ by generalized unary pair recursion:

$$\begin{aligned}
0 \oplus y &:= y \\
\langle v, w\rangle \oplus y &:= \langle v, w \oplus y\rangle \\
|\oplus(x)| &\leq |x| \\
Rt(0) &:= 0 \\
Rt\langle v, w\rangle &:= \langle 0, Rt(v) \oplus Rt(w)\rangle \\
|Rt(x)| &\leq |x|.
\end{aligned}$$

Let f be obtained by the limited pair iteration from the functions $g, j \in \mathcal{F}^i$. We derive $f \in \mathcal{F}^i$ by generalized unary pair recursion as follows:

$$\begin{aligned}
f\langle x, y\rangle &:= f_1\langle Rt(x), y\rangle \\
f_1\langle 0, y\rangle &:= y \\
f_1\langle\langle v, w\rangle, y\rangle &:= g\, f_1\langle w, y\rangle \\
|f_1\langle x, y\rangle| &\leq |j\langle h(x), y\rangle| \\
h(0) &:= 0 \\
h\langle v, w\rangle &:= \langle 0, h(w)\rangle \\
|h(x)| &\leq |x|.
\end{aligned}$$

We note that we have $f_1\langle x, y\rangle = g^{Len(x)}(y)$ and $Len(x) = |h(x)|$. Hence

$$f\langle x, y\rangle = f_1\langle Rt(x), y\rangle = g^{Len\ Rt(x)}(y) = g^{|x|}(y)$$

and

$$|f_1\langle x, y\rangle| = |g^{Len(x)}(y)| = |g^{|h(x)|}(y)| \leq |j\langle h(x), y\rangle|. \ \blacksquare$$

The natural question to ask is why does the above argument break for $\mathcal{P}^2$? It is easy to see that the second half of the proof can be used also with $i = 2$ and thus we have $\mathcal{P}^2 \subseteq \mathcal{F}^2$. Thus pair recursion generate no fewer functions than iteration but it might well generate more. The function m_1 from the identity (17) is not in $\mathcal{P}^2$ because it grows too fast on account of $\otimes$. Can we use a different function? Let f be defined by limited unary pair recursion from $g, h, j \in \mathcal{P}^2$. By the lemma 5.3 we have for certain numbers a and b: $|f(v)| \leq |j(v)| \leq a \cdot |v| + b$. Hence

$$|f\langle y, x\rangle| + |y| + 2 \leq a \cdot |\langle y, x\rangle| + b + |y| + 2 = (a+1) \cdot |y| + a \cdot |x| + (a + b + 2).$$

Thus a function $m(y, x)$ satisfying the condition (2) of the lemma 6.1 should majorize the last expression. It is hard to see how such an m can satisfy the condition (1) of the lemma because each sum would add a factor $a \cdot |x|$. The condition (1) bounds the size of the stack during the iteration of the function k. For the argument y there can be on the order of $|y|$ values $f\langle z, x\rangle$ (z is a part of

y) on the stack. Thus a bound on the stack will have to accomodate on the order of $|y|$ factors of the form $a \cdot |x|$. This means that we cannot find a $\mathcal{P}^2$ function m_1 such that $m(y, x) = |m_1\langle y, x\rangle|$. The argument of the lemma 6.1 does not seem to be usable. It seems implausible that a derivation of f in $\mathcal{P}^2$ does exist as the problem of the factor $|y| \cdot |x|$ necessary to store the already computed values of f probably cannot be circumvened.

However, if the size of the bounding function does not depend on the parameter x, i.e. if $|j\langle y, x\rangle| \leq a \cdot |y| + b$ for some numbers a and b, then the function $m(y, x) = (a + b + 2) \cdot (|y| + 1)$ satisfies the conditions (1) and (2) of the lemma. The function $m_1\langle y, x\rangle := c \otimes (y \oplus \mathbf{1})$ where c is a number such that $|c| = a + b + 2$ has the property $m(y, x) = |m_1\langle y, x\rangle|$ and is also in $\mathcal{P}^2$ as the application of $\otimes$ can be replaced by $a + b + 2$ concatenations of $y \oplus \mathbf{1}$.

We note that the generalized limited unary pair recursion without parameter has the bounding function independent of the parameter. This proves the following theorem:

Theorem 6.3 If the function f is defined by unary pair recursion (10) from functions $g, h \in \mathcal{P}^2$ and there is a function $j \in \mathcal{P}^2$ such that $|f\langle y, x\rangle| \leq |j(y)|$ then $f \in \mathcal{P}^2$.

The class $\mathcal{P}^2$ is closed under generalized limited pair recursion without parameter (13). ■

Theorem 6.4 For a pairing function p the class $\boldsymbol{Pprec}_p$ of *primitive p-pair recursive function* is generated from the functions C_p and T_p by composition of unary functions, p-pairing, and p-pair iteration (12). We have

1. $\boldsymbol{Pprec}_p = \bigcup_i \mathcal{P}^i_p$.
2. The class $\boldsymbol{Pprec}_p$ is generated also by unary p-pair recursion (10) instead of p-pair iteration.

Proof: We assume a fixed pairing function p and drop the parameterization by p. (1): The class $\boldsymbol{Pprec}$ is closed under limited pair iteration because we can simply ignore the bounding condition. Thus the derivations in the proof of the theorem 5.1 show the class $\boldsymbol{Pprec}$ closed under clausal definitions. The functions $\oplus$, $\otimes$, and the pair hierarchy functions F_i are derived in $\boldsymbol{Pprec}$ by pair iteration exactly as in the proof of the lemma 5.2. This shows $\mathcal{P}^i \subseteq \boldsymbol{Pprec}$ for all i, which implies $\bigcup_i \mathcal{P}^i \subseteq \boldsymbol{Pprec}$.

For the converse inclusion $\boldsymbol{Pprec} \subseteq \bigcup_i \mathcal{P}^i$ it is sufficient to show the union of the Pair hierarchy closed under the operations generating the class $\boldsymbol{Pprec}$. If f is defined by the composition of g and h then for some i we have $g, h \in \mathcal{P}^i$. Hence $f \in \mathcal{P}^i \subseteq \bigcup_i \mathcal{P}^i$. The case of pairing is similar. If f is defined by the pair iteration of g then again for some i we have $g \in \mathcal{P}^{i+3}$ and for some k: $|g(y)| \leq E^k_{i+3}\,|y|$ by the lemma 5.3. A simple induction on n shows that

$|g^n(y)| \leq E_{i+3}^{k \cdot n}\,|y|$. The function g is also in the class $\mathcal{P}^{i+4}$ where we derive the function f by limited pair iteration $f\langle x, y\rangle = g^{|x|}(y)$ with the bounding condition

$$|g^{|x|}(y)| \leq E_{i+3}^{k \cdot |x|}\,|y| \leq E_{i+4}(k \cdot |x| + |y|) = |F_{i+4}(k \otimes x \oplus y)|.$$

We thus have $f \in \mathcal{P}^{i+4} \subseteq \bigcup_i \mathcal{P}^i$.

(2): Denote by $\mathcal{F}$ the class generated as ***Pprec*** but with unary pair recursion instead of pair iteration. In order to show $\mathcal{F} \subseteq$ ***Pprec*** it is sufficient to show ***Pprec*** closed under unary pair recursion. Let f be defined by unary pair recursion from primitive pair recursive g and h. The following derivation of f is taken from the proof of the lemma 6.1 where we have omitted the bounds:

$$\begin{aligned} f\langle y, x\rangle &:= H\,H\,T\,T\,k^{|\langle y,y\rangle|}\langle x, Post(y), 0\rangle \\ k\langle x, \langle 0, p\rangle, s\rangle &:= \langle x, p, \langle g(x), 0\rangle, s\rangle \\ k\langle x, \langle \mathbf{1}, p\rangle, \langle fw, w\rangle, \langle fv, v\rangle, s\rangle &:= \langle x, p, \langle h\langle v, w, x, fv, fw\rangle, v, w\rangle, s\rangle. \end{aligned}$$

By the comment in the mentioned proof we have

$$k^{Len\ Post(y)}\langle x, Post(y) \oplus p, s\rangle = \langle x, p, \langle f\langle y, x\rangle, y\rangle, s\rangle$$

which proves that f is correctly derived in ***Pprec***.

Vice versa, by repeating the derivations (without the bounding conditions) in the proof of the theorem 6.2 we show $\mathcal{F}$ first closed under clausal definitions and then under pair iteration. This shows ***Pprec*** $\subseteq \mathcal{F}$. ■

7 Pair Isomorphism

In this section we will investigate the relation of classes of functions formed in a similar way but with different pairing functions.

Definition 7.1 (Pair structures and pair morphisms) *Pair structure* is a triple $\langle A, 0, p\rangle$ where A is a set, called the *carrier*, 0 a nullary operation (constant) $0 \in A$, and p a binary operation over A satisfying the pairing conditions 2.1. It should be clear that the elements of the carrier of a pair structure can be viewed as finite binary trees. The function $\pi : A \to B$ is a *homomorphism from the pair structure* $\langle A, 0_A, p\rangle$ *to the pair structure pair structure* $\langle B, 0_B, q\rangle$ if it *preserves pairs*, i.e. if

$$\begin{aligned} \pi(0_A) &= 0_B \\ \pi\, p(x, y) &= q(\pi(x), \pi(y)). \end{aligned} \tag{18}$$

A homomorphism π is an *isomorphism* if π is a bijection. In the following we restrict our attention to pair structures with the carrier $\mathbf{N}$ and the constant 0_N the number 0. When we say that π is a *pair homomorphism (pair isomorphism) from p to q* we mean that π is a homomorphism (isomorphism) from the pair structure $\langle \mathbf{N}, 0, p\rangle$ to the pair structure $\langle \mathbf{N}, 0, q\rangle$.

Note that for a pair isomorphism π from p to q π^{-1} is a pair isomorphism from q to p.

Theorem 7.1

1. For any pairing functions p and q there is a unique pair homomorphism from p to q.
2. Every pair homomorphism is a pair isomorphism.
3. If π is a bijection such that $\pi(0) = 0$ and p a pairing function then π is a pair isomorphism from p to to q defined as

$$q(x, y) = \pi\, p(\pi^{-1}(x), \pi^{-1}(y)).$$

Proof: (1): Let p and q be two pairing functions. We define by p-pair recursion the function π to satisfy $\pi(0) = 0$ and $\pi\, p(x, y) = q(\pi(x), \pi(y))$. Clearly, π is a pair homomorphism from p to q and by a straightforward p-pair induction we show that any two pair homomorphisms must be identical.

(2): Given a pair homomorphism π from p to q we show that π is a bijection. We first show π injective by proving

$$\pi(x) = \pi(y) \to x = y$$

by p-pair induction on x. If $x = 0$ then if $\pi(0) = 0 = \pi(y)$ we must have also $y = 0$. If $x = p(v_1, w_1)$ for some v_1 and w_1 and if $\pi\, p(v_1, w_1) = q(\pi(v_1), \pi(w_1)) = \pi(y)$ then $y = p(v_2, w_2)$ for som v_2 and w_2. But then $\pi(y) = q(\pi(v_2), \pi(w_2))$ and we get $\pi(v_1) = \pi(v_2)$ and $\pi(w_1) = \pi(w_2)$ by the pairing property of q. By inductive hypotheses $v_1 = v_2$ and $w_1 = w_2$ and thus $x = y$.

By q-pair induction we show that π is onto $\mathbf{N}$, i.e. that $\exists y\, \pi(y) = x$. If $x = 0$ than take $y = 0$. If $x = q(x_1, x_2)$ for some x_1 and x_2 then by induction hypotheses we have $\pi(y_1) = x_1$ and $\pi(y_2) = x_2$ for some y_1 and y_2. Hence for $y = p(y_1, y_2)$ we have $\pi(y) = q(\pi(y_1), \pi(y_2)) = q(x_1, x_2)$.

(3): Assume the conditions of the property. We clearly have $\pi(0) = 0$ and $\pi\, p(x, y) = q(\pi(x), \pi(y))$ so it remains to show that q is a pairing function, i.e. that it satisfies the properties 2.1.

If $q(v_1, w_1) = q(v_2, w_2)$ then also

$$p(\pi^{-1}(v_1), \pi^{-1}(w_1)) = p(\pi^{-1}(v_2), \pi^{-1}(w_2)).$$

We get $v_1 = v_2$ and $w_1 = w_2$ by first using the pairing property of p and then the bijective property of π. This establishes the pairing property of q.

The number a is not an atom of q iff $a = q(x, y)$ for some x and y iff $\pi^{-1}(a) = p(\pi^{-1}(x), \pi^{-1}(y))$ for some x and y iff $\pi^{-1}(a) = p(x, y)$ for some x and y iff $\pi^{-1}(a) \neq 0$ iff $a \neq 0$.

Let $P(x)$ denote the property holding of the natural number x if every chain of immediate parts of q starting from $\pi(x)$ is finite. If $y_1, y_2, \ldots$ were an infinite chain of immediate parts of q we would have $\neg P(\pi^{-1}(y_1))$. Hence we wish to show by p-pair induction that $P(x)$ holds for all numbers x. If $x = 0$ then $\pi(x) = 0$ and the only chain of immediate parts of q starting from 0 consists just of 0. If $x = p(v, w)$ then $\pi(x) = q(\pi(v), \pi(w))$. Every chain of immediate parts of q starting from $\pi(x)$ has either $\pi(v)$ or $\pi(w)$ as the second element and must be finite by induction hypotheses. ■

Theorem 7.2 If π is a pair isomorphism from p to q then

$$\begin{aligned} |x|_p &= |\pi(x)|_q \\ Len_p(x) &= Len_q\, \pi(x). \end{aligned}$$

Proof: By p-pair induction. For $x = 0$ we have $|0|_p = 0 = |0|_q = |\pi(0)|_q$. For $x = p(v, w)$ we have

$$\begin{aligned} |p(v, w)|_p &= |v|_p + |w|_p + 1 = |\pi(v)|_q + |\pi(w)|_q + 1 = \\ & |q(\pi(v), \pi(w))|_q = |\pi\, p(v, w)|_q. \end{aligned}$$

The proof of the second property is similar. ■

The Pair hierarchy consists of inductively defined classes of functions:

Definition 7.2 (Inductively defined class of functions) A class of functions $\mathcal{F}$ is *inductively* defined if it is the smallest class closed under a set ***Ops***

of function operators. A *function operator* $\boldsymbol{Op}$ yields a function from another functions: $f = \boldsymbol{Op}(f_1, f_2, \ldots, f_n)$. We do not exclude the case when $n = 0$, i.e. when $\boldsymbol{Op}$ is a nullary operator, i.e. a specific function.

Example 7.1 (Pair Hierarchy) The class $\mathcal{P}^i_p$ is formed by three nullary operators $C_p = \boldsymbol{Conv}_p()$, $T_p = \boldsymbol{Tail}_p()$, and $F^p_i = \boldsymbol{Phier}^p_i()$, two binary operators $f = \boldsymbol{Comp}_p(g, h)$ yielding the function $f(x) = g\, h(x)$, and $f = \boldsymbol{Pair}_p(g, h)$ yielding the function $f(x) = p(g(x), h(x))$, and by a ternary operator $f = \boldsymbol{Lpiter}_p(g, h, j)$ of limited pair iteration (12).

Definition 7.3 (Pair invariants) For a unary function f and a bijection π we denote by f^π the π-*conjugate* of f which is a unary function such that $f^\pi(x) = \pi\, f\, \pi^{-1}(x)$.

An n-ary ($n \geq 0$) function operator $\boldsymbol{Op}_p$ yielding unary functions from unary functions is *pair invariant*, i.e. it *preserves pairs*, if

$$\boldsymbol{Op}_p(f_1, f_2, \ldots, f_n)^\pi = \boldsymbol{Op}_q(f_1^\pi, f_2^\pi, \ldots, f_n^\pi)$$

for any pair isomorphism π from p to q.

An inductively defined class of unary functions $\mathcal{F}_p$ is *pair invariant* if the set $\boldsymbol{Ops}$ of its function operators consists of pair invariant operators.

If $\mathcal{F}_p$ is an inductively defined class of functions and $f_p \in \mathcal{F}_p$ then we denote by f_q the function $f_q \in \mathcal{F}_q$ which is constructed in exactly the same way (by the same operator expression) as f_p except that instead of an operator $\boldsymbol{Op}_p$ used in the construction of f_p one uses $\boldsymbol{Op}_q$.

Theorem 7.3 (On pair invariants) If $\mathcal{F}_p$ is a pair invariant class and π an isomorphism from p to q then $f_q = f_p^\pi$, i.e.

$$\forall x\ \pi\, f_p(x) = f_q\, \pi(x),$$

for every function $f_p \in \mathcal{F}_p$.

Proof: By induction on the construction of functions f_p in $\mathcal{F}_p$. If $f_p = \boldsymbol{Op}_p(g_p, h_p, \ldots, k_p)$ from the functions $g_p, h_p, \ldots, k_p \in \mathcal{F}_p$ (the operator may be nullary) then by induction hypotheses we have $g_q = g_p^\pi$, $h_q = h_p^\pi$, $\ldots$, $k_p = k_q^\pi$ and we have

$$\begin{aligned} f_q = \boldsymbol{Op}_q(g_q, h_q, \ldots, k_q) = \\ \boldsymbol{Op}_q(g_p^\pi, h_p^\pi, \ldots, k_p^\pi) = \boldsymbol{Op}_p(g_p, h_p, \ldots, k_p)^\pi = f_p^\pi. \ \blacksquare \end{aligned}$$

The theorem 7.3 is not specific about which function operators are pair invariant. The question is what does it take for a pair-based operator to be pair invariant? The surprising answer is: very little!

Theorem 7.4 (Pair invariant function operators) All operators from the example 7.1 are pair invariant. In addition to this, the operators of pair iteration, unary pair recursion, and limited unary pair recursion are pair invariant.

Proof: Let us assume for this proof that π is a pair isomorphism from p to q.

To prove the nullary operator $C_p = \boldsymbol{Conv}_p()$ pair invariant it suffices to show that $C_q = C_p^\pi$. We have

$$C_q(0) = 0 = \pi(0) = \pi\, C_p(0) = \pi\, C_p\, \pi^{-1}(0) = C_p^\pi(0)$$

and also

$$\begin{aligned} C_q\, q(v,w) = q(w,v) = \pi\, p(\pi^{-1}(w), \pi^{-1}(v)) = \\ \pi\, C_p\, p(\pi^{-1}(v), \pi^{-1}(w)) = \pi\, C_p\, \pi^{-1}\, q(v,w) = C_p^\pi\, q(v,w). \end{aligned}$$

The proof of T_p is similar.

The operator of unary composition is straightforward as for $f(x) = g\,h(x)$ we have

$$f^\pi(x) = \pi\, g\, h\, \pi^{-1}(x) = \pi\, g\, \pi^{-1}\, \pi\, h\, \pi^{-1}(x) = g^\pi\, h^\pi(x).$$

This shows $\boldsymbol{Comp}(g,h)^\pi = \boldsymbol{Comp}(g^\pi, h^\pi)$ regardless of pairing functions used.

For $f = \boldsymbol{Pair}_p(g,h)$ we wish to show $f^\pi = \boldsymbol{Pair}_q(g^\pi, h^\pi)$. We have

$$\begin{aligned} f^\pi(x) = \pi\, f\, \pi^{-1}(x) = \pi\, p(g\, \pi^{-1}(x), h\, \pi^{-1}(x)) = \\ q(\pi\, g\, \pi^{-1}(x), \pi\, h\, \pi^{-1}(x)) = q(g^\pi(x), h^\pi(x)). \end{aligned}$$

For $f = \boldsymbol{Piter}_p(g)$, i.e. for f defined by pair iteration (12) where $f(0) = 0$ and $f\, p(x,y) = g^{|x|_p}(y)$, we wish to show that $f^\pi = \boldsymbol{Piter}_q(g^\pi)$. For the argument 0 we have

$$f^\pi(0) = \pi\, f\, \pi^{-1}(0) = \pi\, f(0) = \pi(0) = 0.$$

For the argument of the form $q(x,y)$ we have

$$\begin{aligned} f^\pi\, q(x,y) = \pi\, f\, \pi^{-1}\, q(x,y) = \pi\, f\, p(\pi^{-1}(x), \pi^{-1}(y)) = \\ \pi\, g^{|\pi^{-1}(x)|_p}\, \pi^{-1}(y) = \pi\, g^{|x|_q}\, \pi^{-1}(y) = (g^\pi)^{|x|_q}(y). \end{aligned}$$

The last but one step is by the theorem 7.2 and the last step follows from the property

$$\pi\, g^x\, \pi^{-1}(y) = (g^\pi)^x(y) \tag{19}$$

which is proved by the standard induction on x. For the basis step $x = 0$ we have

$$\pi\, g^0\, \pi^{-1}(y) = \pi\, \pi^{-1}(y) = y = (g^\pi)^0(y).$$

For the inductive step $x = z + 1$ we have

$$\begin{aligned} \pi\, g^{z+1}\, \pi^{-1}(y) = \pi\, g\, g^z\, \pi^{-1}(y) = \\ \pi\, g\, \pi^{-1}\, \pi\, g^z\, \pi^{-1}(y) = g^\pi\, (g^\pi)^z(y) = (g^\pi)^{z+1}(y). \end{aligned}$$

If $f = \boldsymbol{Lpiter}_p(g, j)$, i.e. if f is defined by limited pair iteration (12), then we have $|g^{|x|_p}(y)|_p \leq |j\, p(x, y)|_p$. Then by the theorem 7.2

$$|\pi\, g^{|\pi(x)|_q}(y)|_q \leq |\pi\, j\, p(x, y)|_q.$$

Substituting $\pi^{-1}(x)$ for x and $\pi^{-1}(y)$ for y we get

$$|\pi\, g^{|\pi\,\pi^{-1}(x)|_q}\, \pi^{-1}(y)|_q \leq |\pi\, j\, p(\pi^{-1}(x), \pi^{-1}(y))|_q$$

which simplifies to

$$|\pi\, g^{|x|_q}\, \pi^{-1}(y)|_q \leq |\pi\, j\, \pi^{-1}\, q(x, y)|_q.$$

From the identity (19) we get

$$|(g^{\pi})^{|x|_q}(y)|_q \leq |j^{\pi}\, q(x, y)|_q. \tag{20}$$

Since also $f = \boldsymbol{Piter}_p(q)$, we use the already proved pair invariancy of pair iteration: $f^{\pi} = \boldsymbol{Piter}_q(q^{\pi})$. This, and (20), yields $f^{\pi} = \boldsymbol{Lpiter}_q(q^{\pi}, j^{\pi})$.

The proof that unary pair recursion is pair invariant is similar to that of pair iteration. The proof is then extended to limited pair unary recursion in a similar way as above.

It remains to show that for the pair hierarchy functions we have

$$(F_i^p)^{\pi} = \boldsymbol{Phier}_i^p()^{\pi} = \boldsymbol{Phier}_i^q() = F_i^q. \tag{21}$$

Pair hierarchy functions are definable by pair iteration in the class $\boldsymbol{Pprec}$ of primitive pair recursive functions. The properties of operators proved above are sufficient for the pair invariancy of the class $\boldsymbol{Pprec_p}$. The theorem 7.3 applies from which we get (21). ■

In the view of the theorem 7.4 we should modify our question and ask when a function operator is not pair invariant? It is always the case when a function (i.e. a nullary operator) or an operator depends on numeric properties rather than on properties expressed strictly in terms of pairing. Thus neither of the size $|x|$, $Len(x)$, or successor $s(x) = x + 1$ functions is pair invariant as can be easily checked. The unary iteration operator (11), as opposed to pair iteration (12), is not pair invariant.

Theorem 7.5 If π is an isomorphism from p to q and $\mathcal{F}_p$ a pair invariant function class then

1. $\mathcal{F}_p = \mathcal{F}_q \leftrightarrow \forall p\, (f \in \mathcal{F}_p \leftrightarrow f^{\pi} \in \mathcal{F}_p)$,
2. If $\mathcal{F}_p$ is closed under unary composition then $\pi, \pi^{-1} \in \mathcal{F}_p \rightarrow \mathcal{F}_p = \mathcal{F}_q$.

Proof: $(1 \rightarrow)$: If $\mathcal{F}_p = \mathcal{F}_q$ then to every $f_p \in \mathcal{F}_p$ we have $f_q = f_p^\pi \in \mathcal{F}_q$, i.e. also in $\mathcal{F}_p$, by the lemma 7.3. By the same argument, since π^{-1} is an isomorphism from q to p, given $f^\pi \in \mathcal{F}_p$, i.e. in $\mathcal{F}_q$, we have $f = (f^\pi)^{\pi^{-1}} \in \mathcal{F}_p$. $(1 \leftarrow)$: Assume the right hand side equivalence. Then for a $f_q \in \mathcal{F}_q$ we have $f_q = f_p^\pi$ for the corresponding $f_p \in \mathcal{F}_p$, thus also $f_q \in \mathcal{F}_p$ and we have $\mathcal{F}_q \subseteq \mathcal{F}_p$. The argument for the converse inclusion is similar except that it uses π^{-1}.

(2) : If $\pi, \pi^{-1} \in \mathcal{F}_p$ then for a $f \in \mathcal{F}_p$ the function $f^\pi(x) = \pi\, f\, \pi^{-1}(x)$ is also in $\mathcal{F}_p$. Similarly, if $f^\pi \in \mathcal{F}_p$ then also $(f^\pi)^{\pi^{-1}} \in \mathcal{F}_p$ and the property (1) applies. ■

Theorem 7.6 If π is a pair isomorphism from p to q then

$$\pi, \pi^{-1} \in \mathcal{P}_p^i \rightarrow \mathcal{P}_p^i = \mathcal{P}_q^i.$$

Proof: The classes of Pair hierarchy are pair invariant and closed under unary composition so the theorem 7.5(2) applies. ■

8 Hyperelementary Classes of Pair Hierarchy

In this section we will study the relation of Pair hierachy $\mathcal{P}^i$ to the *Grzegorczyk hierarchy* $\mathcal{E}^i$ [5, 10]. The relation is not pair invariant as it must depend on numeric properties of the pairing function p. We fix for the remainder of this section the pairing function p to the function J from the example 2.1. In view of the theorem 7.6 we do not lose any generality by doing this. We remind the reader that the notation $\langle \boldsymbol{x} \rangle$ becomes an abbreviation for $J_n(\boldsymbol{x})$.

We denote by $\mathcal{F}_1$ the class of unary functions of a function class $\mathcal{F}$. We restrict ourselves in this section to the *hyperelementary* classes $\mathcal{E}^{i+3}$ of Grzegorczyk hierarchy. The reason for this will become obvious in the sections 10 and ??. The two main results of this section are given by the theorem 8.12 which says that $\mathcal{P}_J^{i+3} = \mathcal{E}_1^{i+3}$ and by the theorem 8.13 which says that the class of primitive pair recursive functions $\boldsymbol{Pprec}_J$ consists exactly of unary primitive recursive functions.

Definition 8.1 (Grzegorczyk hierarchy) The class of functions $\mathcal{E}(f_1, \cdots, f_n)$ is generated from the functions $f_1, \cdots, f_n$, zero $Z(x) = 0$, successor $s(x) = x+1$, projections $U_i^n(\boldsymbol{x}) = x_i$ $(1 \leq i \leq n)$ by composition and limited recursion (1).

The classes of *Grzegorczyk hierarchy* are defined as $\mathcal{E}^0 = \mathcal{E}(s)$, $\mathcal{E}^1 = \mathcal{E}(+)$, $\mathcal{E}^2 = \mathcal{E}(\cdot)$, and for $i \geq 3$ as $\mathcal{E}^i = \mathcal{E}(E_i, max)$.

We note here that there is an error in the definition of Grzegorczyk hierarchy in Rose [10] who defines $\mathcal{E}^i = \mathcal{E}(E_i)$. This definition bounds all functions $f \in \mathcal{E}^i$ for $i \geq 2$ to $f(x_1, \cdots, x_n) \leq E_i^k(x_j)$ for some k and j. Consequently, the addition $+$ is not in any such $\mathcal{E}^i$.

Our corrected definition is the same for $\mathcal{E}^0$ and $\mathcal{E}^1$. Our class $\mathcal{E}^2$ includes the class $\mathcal{E}(E_2, max)$ because $E_2(x) = x^2 + 2$ can be obtained from $\cdot$ and s by composition. Addition and $\dot{-}$ is definable in our $\mathcal{E}^2$ by the usual limited recursion and then we define $max(x, y) = x + (y \dot{-} x)$. Vice versa, first the addition and then the multiplication is definable by limited recursion in $\mathcal{E}(E_2, max)$ as we have $x + y \leq 2 \cdot max(x, y) \leq max(x, y)^2 + 2 = E_2\, max(x, y)$. Similar bound can be found for multiplication. Thus both classes are the same. Similar considerations hold for the higher classes.

The Gregorczyk hierarchy [5, 10] is a proper hierachy as $\mathcal{E}^i \subset \mathcal{E}^{i+1}$. The class $\mathcal{E}^2$ is the first class containing pairing functions, and it does contain our pairing function J together with its generalizations J_n. The projection functions K and L are definable already in $\mathcal{E}^0$. The class $\mathcal{E}^3$ of *elementary* functions is the first class to contain the exponential function x^y. The union of Grzegorczyk hierarchy $\bigcup_i \mathcal{E}^i$ is the class of *primitive recursive functions.* The reader will find in [10] the proofs of all properties of the classes $\mathcal{E}^i$ which we claim but do not prove here.

We start with the properties of the function J whose initial segment is tabulated in the table 1. The subscripts of the values of $J(x, y)$ give the size $|J(x, y)|$.

$J(x,y)$	0	1	2	3	4	5	6	$\cdots$
0	1_1	2_2	4_3	7_3	11_4	16_4	22_4	$\cdots$
1	3_2	5_3	8_4	12_4	17_5	23_5	30_5	$\cdots$
2	6_3	9_4	13_5	18_5	24_6	31_6	39_6	$\cdots$
3	10_3	14_4	19_5	25_5	32_6	40_6	49_6	$\cdots$
4	15_4	20_5	26_6	33_6	41_7	50_7	60_7	$\cdots$
5	21_4	27_5	34_6	42_6	51_7	61_7	72_7	$\cdots$
6	28_4	35_5	43_6	52_6	62_7	73_7	85_7	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

Table 1: Pairing function $J(x,y)$

Lemma 8.1 (Bounds on J-pair size) There is a strictly increasing elementary function $B(n)$ which is not in $\mathcal{E}^2$ such that

1. $|B(n)| = n$,
2. $|x| \leq x \leq B\,|x|$.

Proof: We first prove $|x| \leq x$ by a straightforward J-pair induction on x and then observe that $x \leq B\,|x|$ together with (1) says that $B(n)$ yields the largest number with the size n. The function B is defined by recursion as $B(0) = 0$ and $B(n+1) = \langle B(n), 0\rangle$. Since J strictly increases in both arguments B strictly increases too. Moreover, it is easily seen that (1) holds. By straightforward induction on n we can show

$$B(n) + B(m) \leq B(n+m). \tag{22}$$

We now show that $x \leq B\,|x|$ by J-pair induction. The case $x = 0$ is trivial. For $x = \langle v, w\rangle$ we have from (22) and induction hypotheses $v + w \leq B(|v| + |w|)$. Thus

$$\begin{aligned}
\langle v, w\rangle &= \frac{(v+w+1)\cdot(v+w)}{2} + v + 1 \\
&\leq \frac{(B(|v|+|w|)+1)\cdot B(|v|+|w|)}{2} + B(|v|+|w|) + 1 \\
&= \langle B(|v|+|w|), 0\rangle = B(|v|+|w|+1) = B\,|\langle v, w\rangle|.
\end{aligned}$$

Finally, by an easy induction proof we get

$$2 \cdot 5^{2^n} \leq B(n+3) \leq 10^{2^n}. \tag{23}$$

The first inequality shows that B cannot be bounded by a polynomial of n, i.e. $B \notin \mathcal{E}^2$. From the second inequality we get $B(n) \leq 10^{2^n}$ which shows that B can be defined by limited recursion in $\mathcal{E}^3$. ■

We now prove some auxiliary lemmas about the functions and operations of the classes $\mathcal{P}^{i+3}$ in the classes $\mathcal{E}^{i+3}$. The first lemma applies also to the class $\mathcal{E}^2$:

Lemma 8.2 The classes $\mathcal{E}^i$ where $i \geq 2$ are closed under clausal definitions.

Proof: The functions I, Z, H_J, and T_J are in $\mathcal{E}_1^0$ (see [10]). The conditional function D is definable already in $\mathcal{E}_1^0$ but we show it in $\mathcal{E}_1^1$:

$$D(x) = \begin{cases} H\,T(x) & \text{if } H(x) = 0 \\ T\,T(x) & \text{if } H(x) > 0. \end{cases}$$

Let $i \geq 2$. The class $\mathcal{E}^i$ is closed under composition thus $\mathcal{E}_1^i$ is closed under unary composition. If f is defined by pairing from $g, h \in \mathcal{E}_1^i$ then f is also in $\mathcal{E}_1^i$ by the explicit definition $f(x) = J(g(x), h(x))$ because $J \in \mathcal{E}^2$. Thus the functions simple in $\mathcal{E}_1^i$ are exactly the functions in $\mathcal{E}_1^i$ and by the theorem 4.3 the class $\mathcal{E}_1^i$ is closed under clausal definitions. Clausal definitions yield unary functions from unary functions and so also $\mathcal{E}^i$ is closed under clausal definitions. ■

Lemma 8.3 The pair size function $|x|$ is elementary.

Proof: The pair size function is defined by pair recursion in (4). The recursion goes from $J(x, y)$ to x and y both of which are less than $J(x, y)$. This suggests that the pair size function can be derived by *limited course of values recursion*. Course of values recursion is reducible to recursion in all classes containing the exponential function by defining a function f with the property:

$$f(x) = 2^{|0|} \cdot 3^{|1|} \cdot \ldots \cdot pr_x^{|x|}.$$

The elementary function pr_n yields the n-th prime ($pr_0 = 2$). The pair size function is then defined by $|x| = exp(f(x), x)$ where the elementary function $exp(x, n)$ yields the exponent of pr_n in the prime factorization of x.

The function f is defined elementary by limited recursion as follows

$$\begin{aligned} f(0) &= 2^0 \\ f(x+1) &= f(x) \cdot pr_{x+1}^{exp(f(x),H(x+1))+exp(f(x),T(x+1))+1} \\ f(x) &\leq \prod_{k \leq x} pr_k^k. \end{aligned}$$

The bound on f is derived from the inequality $|x| \leq x$. ■

Lemma 8.4 Every class $\mathcal{E}^{i+3}$ is closed under limited pair iteration (12) with the bounding condition in either of the forms $|g^{|x|}(y)| \leq |j\langle x, y\rangle|$ or $|g^{|x|}(y)| \leq j\langle x, y\rangle$.

Proof: Let $i \geq 3$ and f be defined by limited pair iteration from $g, j \in \mathcal{E}^i$. From the bounding condition $|g^{|x|}(y)| \leq |j\langle x, y\rangle|$ we get $|g^{|x|}(y)| \leq j\langle x, y\rangle$. Thus it is sufficient to prove the closure with the bounding condition of the second form, which can be written as $g^{|x|}(y) \leq B\, j\langle x, y\rangle$. The function f is derived in $\mathcal{E}^i$ as follows:

$$
\begin{aligned}
f(v) &= \begin{cases} f_1(|H(v)|, T(v)) & \text{if } v > 0 \\ 0 & \text{otherwise.} \end{cases} \\
f_1(0, y) &= y \\
f_1(x+1, y) &= g\, f_1(x, y) \\
f_1(x, y) &= g^x(y) = g^{|B(x)|}(y) \leq B\, j\langle B(x), y\rangle. \ \blacksquare
\end{aligned}
$$

Lemma 8.5 $\mathcal{P}^{i+3} \subseteq \mathcal{E}^{i+3}$.

Proof: Let $i \geq 3$. The closure of $\mathcal{E}^i$ under clausal definitions implies by the theorem 4.3 the closure under pairing and also that $C, T \in \mathcal{E}^i$. The class $\mathcal{E}^i$ is closed under limited pair iteration by the lemma 8.4. It remains to show $F_i \in \mathcal{E}^i$. This is done by deriving in $\mathcal{E}^i$ the functions *Reva*, *Rev*, $\oplus$, $\otimes$, $F_{2.5}$, F_3, $\ldots$, F_k, $\ldots$, F_i in that order. The derivation is exactly as in the proof of the lemma 5.2 with the only difference that whenever a pair hierarchy function is used in the bound of a bounding condition it has to be replaced by an $\mathcal{E}^i$ function. Thus the bounding condition on the iteration of the auxiliary function f which is used in the derivation of $\otimes$ is

$$
\begin{aligned}
|f^{|x|}(v)| &\leq g\langle x, v\rangle \\
g\langle x, y, a\rangle &:= (x+1) \cdot y + a + 1
\end{aligned}
$$

because

$$|f^{|x|}\langle y, a\rangle| = |y| + 1 + |x| \cdot |y| + |a| \leq y + 1 + x \cdot y + a = g\langle x, y, a\rangle.$$

The pair hierarchy functions F_k $(3 \leq k \leq i)$ are derived in $\mathcal{E}^i$ by limited pair iteration exactly as in $\mathcal{P}^i$: $F_k(x) = F_j^{|x|}\langle 0, 0, 0\rangle$ where for $k > 3$ we have $j = k - 1$ and for $k = 3$ we have $j = 2.5$. The $\mathcal{E}^i$ bounding condition is

$$|F_j^{|x|}(y)| = E_{k-1}^{|x|}\, |y| \leq E_k(|y| + |x|) \leq E_k(y + x). \ \blacksquare$$

The next set of lemmas deals with the functions and operations of classes $\mathcal{E}^{i+3}$ in the classes $\mathcal{P}^{i+3}$.

Lemma 8.6 For every n-ary function $f \in \mathcal{E}^{i+3}$ there is a function $b \in \mathcal{P}^{i+3}$ such that

$$f(|x_1|, |x_2|, \ldots, |x_n|) \leq |b\langle x_1, x_2, \ldots, x_n\rangle|.$$

Proof: For every function $f(\boldsymbol{x}) \in \mathcal{E}^{i+3}$ there is a number k such that

$$f(x_1, x_2, \ldots, x_n) \leq E^k_{i+3}(x_1 + x_2 + \ldots + x_n).$$

Thus

$$\begin{aligned} f(|x_1|, |x_2|, \ldots, |x_n|) &\leq E^k_{i+3}(|x_1| + |x_2| + \ldots + |x_n|) = \\ &E^k_{i+3}(|x_1 \oplus x_2 \oplus \ldots \oplus x_n|) = |F^k_{i+3}(x_1 \oplus x_2 \oplus \ldots \oplus x_n)|. \end{aligned}$$

Hence we can define the function $b \in \mathcal{P}^{i+3}$ as follows:

$$b\langle x_1, x_2, \ldots, x_n\rangle := F^k_{i+3}(x_1 \oplus x_2 \oplus \ldots \oplus x_n). \;\blacksquare$$

Lemma 8.7 The successor $s(x) = x + 1$ and the predecessor $pred(x) = x \dot{-} 1$ functions are in $\mathcal{P}^3$.

The inspection of the table 1 suggests the crucial recurrence relations for the successor and predecessor functions:

$$\begin{aligned} \langle x, y\rangle + 1 &= \langle x + 1, y \dot{-} 1\rangle \text{ if } y > 0 \\ \langle x, y\rangle \dot{-} 1 &= \langle x \dot{-} 1, y + 1\rangle \text{ if } x > 0. \end{aligned}$$

Hence both functions must be introduced by simultaneous pair recursion which is reduced to limited unary pair recursion by defining an auxiliary function f such that $f(x) = \langle x + 1, x \dot{-} 1\rangle$ as follows:

$$\begin{aligned} x + 1 &= K\, f(x) \\ x \dot{-} 1 &= L\, f(x) \\ f(0) &:= \langle 1, 0\rangle \\ f\langle v, w\rangle &:= g\langle v, w, f(v), f(w)\rangle \\ |f(x)| &\leq |b(x)| \\ g\langle 0, 0, \langle pv, mv\rangle, \langle pw, mw\rangle\rangle &:= \langle 2, 0\rangle \\ g\langle\langle a, b\rangle, 0, \langle pv, mv\rangle, \langle pw, mw\rangle\rangle &:= \langle\langle 0, pv\rangle, \langle mv, 1\rangle\rangle \\ g\langle 0, \langle a, b\rangle, \langle pv, mv\rangle, \langle pw, mw\rangle\rangle &:= \langle\langle 1, mw\rangle, \langle mw, 0\rangle\rangle \\ g\langle\langle a, b\rangle, \langle c, d\rangle, \langle pv, mv\rangle, \langle pw, mw\rangle\rangle &:= \langle\langle pv, mw\rangle, \langle mv, pw\rangle\rangle. \end{aligned}$$

It should be obvious that the functions behave as expected. It remains to find the bounding function b. From the lemma 8.1 we have

$$\begin{aligned} |x + 1| &\leq x + 1 \leq B\,|x| + 1 \\ |x \dot{-} 1| &\leq x \dot{-} 1 \leq B\,|x| \\ |f(x)| &= |\langle x + 1, x \dot{-} 1\rangle| = |x + 1| + |x \dot{-} 1| + 1 \leq 2 \cdot B\,|x| + 2 \leq |b(x)| \end{aligned}$$

where the function $b \in \mathcal{P}^3$ is obtained from the lemma 8.6 to bound the elementary function $k(x) = 2 \cdot B(x) + 2$. $\blacksquare$

Lemma 8.8 The classes $\mathcal{P}^{i+3}$ are closed under limited unary iteration (11).

Proof: We derive f by limited pair iteration in $\mathcal{P}^{i+3}$ as follows:

$$\begin{aligned} f\langle x,y\rangle &:= T\, h^{|b(x)|}\langle x,y\rangle \\ |h^{|n|}(z)| &\le |k\langle n,z\rangle| \\ h\langle 0,y\rangle &:= \langle 0,y\rangle \\ h\langle\langle v,w\rangle,y\rangle &:= \langle\langle v,w\rangle \dot{-} 1, g(y)\rangle \\ k\langle n,x,y\rangle &:= c\langle x,y\rangle \end{aligned} \tag{24}$$

where both functions b and c are obtained in $\mathcal{P}^{i+3}$ by the bounding lemma 8.6 as follows. For b we wish to satisfy $x \le B\,|x| \le |b(x)|$ in order to get the number of iterations of h large enough to vanish its head. Preparatory to obtaining c we observe that for a binary function m we have

$$\begin{aligned} h^n\langle x,y\rangle = \langle x \dot{-} n, g^{min(x,n)}(y)\rangle \le \\ \langle x, j\langle min(x,n),y\rangle\rangle \le \langle x, \sum_{k\le x} j\langle k,y\rangle\rangle = m(x,y). \end{aligned}$$

Here we are relying on the monotone properties of the pairing function J. Since the $\mathcal{P}^{i+3}$ function j is by the lemma 8.5 also in $\mathcal{E}^{i+3}$ the function m is also there. Hence for some k we have a monotone bound on it $m(x,y) \le E^k_{i+3}(x+y)$. The bound is used to obtain the function $c \in \mathcal{P}^{i+3}$ as follows:

$$|h^n\langle x,y\rangle| \le h^n\langle x,y\rangle \le E^k_{i+3}(x+y) \le E^k_{i+3}(B\,|x| + B\,|y|) \le |c\langle x,y\rangle|.$$

This satisfies the inequality (24) for $z = \langle x,y\rangle$. For $z = 0$ the inequality holds from the clausal definitions of h and k since $h^n(0) = k(n,0) = 0$. ■

Lemma 8.9 The unary equivalents of binary functions of addition, modified subtraction $\dot{-}$, minimum, maximum, and multiplication are in the class $\mathcal{P}^3$.

In order to improve the readability we will write applications of unary arithmetic functions in the infix form without the superscript $+$, for instance, we write $x+y$ as an abbreviation for $x^+\langle x,y\rangle$.

Proof: Unary addition is obtained by limited unary iteration of the successor function s as follows:

$$\begin{aligned} x+y &:= s^x(y) \\ s^x(y) &= x+y \le B\,|x| + B\,|y| \le |b\langle x,y\rangle| \le b\langle x,y\rangle \end{aligned}$$

where function $b \in \mathcal{P}^3$ is obtained from the bounding lemma 8.6. Unary modified subtraction is defined as $x \dot{-} y := pred^y(x)$. Unary minimum and maximum functions have clausal definitions $min^+\langle x,y\rangle := D\langle x \dot{-} y, x, y\rangle$ and $max^+\langle x,y\rangle := D\langle x \dot{-} y, y, x\rangle$ respectively. Unary multiplication is derived as $x \cdot y := T\, f^x\langle y,0\rangle$ where $f\langle y,a\rangle = \langle y, y+a\rangle$ with the bounding conditions derived similarly to that for the unary $+$. ■

Lemma 8.10 $E_{i+3} \in \mathcal{P}^{i+3}$.

Proof: By induction on i. For $i = 0$ we first observe that the function $E_2(x) = x^2 + 2$ is in $\mathcal{P}^3$ by the lemma 8.9. We derive E_3 in $\mathcal{P}^{i+3}$ by limited unary iteration: $E_3(x) = E_2^x(2)$ since

$$E_2^x(y) \leq E_3(x + y) \leq E_3(B\,|x| + B\,|y|) \leq |b\langle x, y\rangle| \leq b\langle x, y\rangle$$

where the function $b \in \mathcal{P}^{i+3}$ is obtained by the bounding lemma 8.6. The derivation of $E_{i+4} \in \mathcal{P}^{i+4}$ is by a similar iteration of E_{i+3} which is in $\mathcal{P}^{i+3} \subset \mathcal{P}^{i+4}$ by induction hypothesis. ∎

Lemma 8.11 The unary equivalent f^+ of every $\mathcal{E}^{i+3}$ function f is in the class $\mathcal{P}^{i+3}$.

Proof: By induction on construction of functions of $\mathcal{E}^{i+3}$ for a fixed i. For the basic function Z we have $Z = Z^+ \in \mathcal{P}^{i+3}$. For the successor function $s = s^+$ the lemma 8.7 applies. For a projection function U_j^n we define in $\mathcal{P}^{i+3}$

$$(U_j^n)^+\langle \boldsymbol{x}\rangle := x_j.$$

The function min^+ is in $\mathcal{E}^{i+3}$ by the lemma 8.9. For the hierarchy function $E_{i+3} = (E_{i+3})^+$ the lemma 8.10 applies.

If $f(\boldsymbol{x})$ is defined by the composition of n-ary functions h_i with m-ary function g we define in $\mathcal{P}^{i+3}$

$$f^+\langle \boldsymbol{x}\rangle := g^+\langle h_1^+\langle \boldsymbol{x}\rangle, h_2^+\langle \boldsymbol{x}\rangle, \ldots, h_m^+\langle \boldsymbol{x}\rangle\rangle.$$

Finally, let $f(y, \boldsymbol{x})$ be defined by limited recursion (1) from the functions g, h and j. By induction hypothesis the function j^+ is in $\mathcal{P}^{i+3}$ and by the lemma 8.5 also in $\mathcal{E}^{i+3}$. Thus for certain k we have $j^+(z) \leq E_{i+3}^k(z)$. For all $y_1 \leq y$ we then have

$$f(y_1, \boldsymbol{x}) \leq j(y_1, \boldsymbol{x}) = j^+\langle y_1, \boldsymbol{x}\rangle \leq E_{i+3}^k\langle y_1, \boldsymbol{x}\rangle \leq E_{i+3}^k\langle y, \boldsymbol{x}\rangle.$$

Since by the lemma 8.10 $E_{i+3} \in \mathcal{P}^{i+3}$ we use the last inequality as a bound on all recursive applications of f which are computed in the function f^+ defined by limited unary iteration in $\mathcal{P}^{i+3}$:

$$\begin{aligned}
f^+\langle y, \boldsymbol{x}\rangle &:= K\, k^y\langle g^+\langle \boldsymbol{x}\rangle, 0, E_{i+3}^k\langle y, \boldsymbol{x}\rangle, \boldsymbol{x}\rangle \\
k\langle a, y, b, \boldsymbol{x}\rangle &:= \langle min^+\langle h^+\langle y, \boldsymbol{x}, a\rangle, b\rangle, y + 1, b, \boldsymbol{x}\rangle \\
k^n(v) &\leq m\langle n, v\rangle \\
m\langle n, a, y, b, \boldsymbol{x}\rangle &:= \langle min^+\langle a, b\rangle, y + n, b, \boldsymbol{x}\rangle. \ \blacksquare
\end{aligned}$$

Theorem 8.12 $\mathcal{P}_J^{i+3} = \mathcal{E}_1^{i+3}$.

Proof: $\subseteq$: Every $f \in \mathcal{P}_J^{i+3}$ is a unary function of $\mathcal{E}^{i+3}$ by the lemma 8.5. $\supseteq$: For a unary function $f \in \mathcal{E}^{i+3}$ we have $f = f^+$ so by the lemma 8.11 $f \in \mathcal{P}_J^{i+3}$. ■

Theorem 8.13 The class $\boldsymbol{Pprec}_J$ consists exactly of unary primitive recursive functions.

Proof: A primitive pair recursive function f is by the theorem 6.4 in some $\mathcal{P}^{i+3}$, hence by the theorem 8.12 it is in $\mathcal{E}_1^{i+3}$ and so it is a unary primitive recursive function. Vice versa, every unary primitive recursive function f is by the same theorems first in some $\mathcal{E}^{i+3}$, hence also in $\mathcal{P}^{i+3}$, and thus it is primitive pair recursive. ■

9 Enumeration of Pairs

In hyperelementary function classes we have at our disposal the exponential function x^y whose powers are such that they distort the problems with the full correlation of Pair and Grzegorczyk hierarchies in the lower classes where the exponential function is not available. The problems are three: (i) the pairing function J is not usable in the lower classes, (ii) pair recursion and pair iteration do not admit sufficiently long iteration sequences, (iii) there is a missing class in the Grzegorczyk hierarchy which prevents the correlation in the lower classes.

We will correct the problem (i) in this section, the problem (ii) in the section 10 and the problem (iii) in the section 11. We fully characterize the Pair hierarchy in the section **??**. The shortness of iteration in $\mathcal{P}^i$ is a 'feature' rather than a shortcoming as it has very important consequences in the complexity theory. Nevertheless, the short iteration prevents the correlation of $\mathcal{P}^i$ to the lower classes of Grzegorczyk hierarchy.

We start with the enumeration of numbers in a pair representation.

Definition 9.1 (Standard enumeration of pairs) Let p be a pairing function. The *p-lexicographic order* $\prec^l_p$ is a relation such that $0 \prec^l_p p(a,b)$ and $p(a,b) \prec^l_p p(c,d)$ if either $a \prec^l_p c$ or else $a = c$ and $b \prec^l_p d$. The relation $\prec_p$ is such that $x \prec_p y$ if $|x|_p < |y|_p$ or $|x|_p = |y|_p$ and $x \prec^l_p y$. The relation $\prec_p$ is the *standard enumeration of pairs.*

It can be easily seen that the lexicographic order is a linear order on $\mathbf{N}$. However, it is not a well-order as we have

$$\langle\langle 0,0\rangle,0\rangle \succ^l_p \langle 0,\langle 0,0\rangle,0\rangle \succ^l_p \langle 0,0,\langle 0,0\rangle,0\rangle \succ^l_p \ldots .$$

The situation is corrected with the relation $\prec_p$ which is an order of $\mathbf{N}$ of order type ω. Thus there is a unique bijection Ord_p establishing an order isomorhism from the order structure $\langle \mathbf{N}, \prec_p\rangle$ to the order structure $\langle \mathbf{N}, <\rangle$, i.e.:

$$x \prec_p y \leftrightarrow Ord_p(x) < Ord_p(y). \tag{25}$$

The number $Ord_p(x)$ is the *ordinal* of x in the relation $\prec_p$, i.e. the total number of such y that $y \prec_p x$. Note that $Ord_p(0) = 0$, $Ord_p\, p(0,0) = 1$, $Ord_p\, p(0,0,0) = 2$, $Ord_p\, p(p(0,0),0) = 3$, etc.

Lemma 9.1 If π is a pair isomorphism from p to q then

1. $x \prec_p y \leftrightarrow \pi(x) \prec_q \pi(y)$,
2. $Ord_p(x) = Ord_q\, \pi(x)$.

Proof: The property (1) should be obvious from the definition of $\prec$ because the sizes of numbers are invariant under pair isomorphism by the theorem 7.2. The property (2) follows from the property (1) and (25). ■

For now on until the theorem 9.3 we fix the pairing function p and stop showing the parameterization by it.

The function $Succ(x)$ yields the *successor in* $\prec$ *of* x. This means that for all y we have

$$x \prec y \rightarrow Succ(x) \preceq y.$$

We will now derive the function $Succ$ in the class $\mathcal{P}^2$. We make some observations about the order $\prec$. It is easily seen that the right leaning tree $Rt(x)$ is the smallest in $\prec$ with the size $|x|$. A 'left leaning tree' $Lt(x)$ is the largest in $\prec$ with the size $|x|$:

$$Rt(x) \preceq x \preceq Lt(x) \prec \langle 0, Rt(x)\rangle = Succ\ Lt(x). \tag{26}$$

The function Lt is derived in $\mathcal{P}^2$ similarly to the derivation of Rt in the section 6:

$$\begin{aligned} Lt(x) &= H\ f^{|\langle x,x\rangle|}\langle 0, x, 0\rangle \\ |f^{|x|}(v)| &\le |g\langle x, v\rangle| \\ f\langle a, \langle v, w\rangle, s\rangle &:= \langle\langle a, 0\rangle, v, w, s\rangle \\ f\langle a, 0, s\rangle &:= \langle a, s\rangle \\ g\langle x, a, s\rangle &= \langle s, a, s\rangle. \end{aligned}$$

The function $Left(x)$ which yields 0 if x is a left leaning tree, i.e. $Lt(x) = x$, and $\mathbf{1}$ otherwise is defined in $\mathcal{P}^2$ by limited unary pair recursion without parameter (13):

$$\begin{aligned} Left(0) &= 0 \\ Left\langle v, w\rangle &= D\langle w, Left(v), \mathbf{1}\rangle \\ |Left(v)| &\le 1 = |\mathbf{1}|. \end{aligned} \tag{27}$$

The function $Left$ is the *characteristic function* of the predicate of 'being a left leaning tree'. In the following we will identify characteristic functions with predicates.

The successor in $\prec$ of a left leaning tree is given in (26). If w is not a left leaning tree, i.e. $Left(w) = \mathbf{1}$, then $Succ\langle v, w\rangle = \langle v, Succ(w)\rangle$. If v is not a left leaning tree and w is then $Succ\langle v, w\rangle = \langle Succ(v), Rt(w)\rangle$. The most complex case is when v and $\langle w, 0\rangle$ are both left leaning trees as then $Succ\langle v, w, 0\rangle = \langle\langle 0, Rt(v)\rangle, Rt(w)\rangle$, i.e. we take one from the size of tail and add it to the head. With this on mind, we derive the function $Succ$ in $\mathcal{P}^2$ by limited unary pair recursion without parameter as follows:

$$\begin{aligned} Succ(0) &:= \langle 0, 0\rangle \\ Succ\langle v, w\rangle &:= D\langle Left\langle v, w\rangle, Rt\langle 0, v, w\rangle, D\langle Left(w), \\ &\qquad D\langle Left(v), \langle\langle 0, Rt(v)\rangle, Rt\ T(w)\rangle, \langle Succ(v), Rt(w)\rangle\rangle, \\ &\quad \langle v, Succ(w)\rangle\rangle\rangle \\ |Succ(x)| &\le |x| + 1 = |\langle 0, x\rangle|. \end{aligned} \tag{28}$$

The Ord-conjugate of the function $Succ$ is the successor function, i.e.

$$s(x) = x + 1 = Succ^{Ord}(x). \tag{29}$$

This should be clear from the fact that $Succ\,Ord^{-1}(x)$ is the $\prec$ successor of $Ord^{-1}(x)$ and from (25) we get that $Ord\,Succ\,Ord^{-1}(x)$ is the successor of x. We leave it to the reader to define in $\mathcal{P}^2$ a *predecessor in* $\prec$ function $Pred$ for which we have

$$x \mathbin{\dot{-}} 1 = Pred^{Ord}(x). \tag{30}$$

We note that since the order type of $\prec$ is ω we have both the principle of (transfinite) induction on $\prec$ and *induction on Succ* where we can infer $\forall x\, P(x)$ from $P(0)$ and $\forall x(P(x) \to P(Succ(x)))$.

We will now discuss the problem (i) concerning the pairing function J. It is not difficult to see that for the $\mathcal{P}^2_J$ function Lt_J we have $Lt_J(x) = B\,|x|_J$ where B is the elementary function from the lemma 8.1. This is an unfortunate situation as we would like to characterize the class $\mathcal{P}^2_J$ by a class which is less than elementary. We recall that $|x|_J \leq x$, actually for many x the number $|x|_J$ is on the order of $log(x)$ so perhaps the detour through the elementary B can be eliminated? Unfortunately, this is not the case. We will in a moment define a function t such that $t(x) \leq 2^{|t(x)|_J}$. Then we will have by the property (23)

$$5^{t(x)} \leq 2 \cdot 5^{2^{|t(x)|_J}} \leq B(|t(x)|_J + 3) = B\,|J(0,0,0,t(x))|_J = Lt_J\,J(0,0,0,t(x)).$$

The $\mathcal{P}^2_J$ function $u(x) = Lt_J\,J(0,0,0,x)$ cannot be bounded by a polynomial of x because it majorizes an exponential fuction for infinitely many numbers $t(x)$. Thus the function u cannot be in $\mathcal{E}^2$.

The function t is defined by recursion as $t(0) = 0$ and $t(x+1) = J(t(x), t(x))$. Straightforward induction shows $|t(x)|_J = 2^x - 1$ and $t(x) + 1 \leq 2^{2^x - 1}$. Thus $t(x) \leq 2^{|x|_J}$.

The problem with the pairing function J is that it widely disperses numbers of equal J-pair size. Numbers $t(x)$ and $Lt_J\,t(x)$ have the same J-pair size but the latter is exponentially larger than the first one. We will now investigate the numeric properties of functions Ord_p and in the process of doing so derive a good pairing function P where we can get from the smallest number of a given P-pair size to the largest one by a $\mathcal{E}^2$ function.

The first set of properties concerns the ordinal of the smallest in $\prec$ number $Rt(x)$ of a pair size $|x|$:

$$\begin{aligned}
Ord(0) &= 0 \\
Ord\langle 0, Rt(x)\rangle &= Ord\,Rt(x) + Ct\,|x| \\
Ord\,Rt(x) &= Mins_1\,|x| \\
Mins_1(n) &= \sum_{i<n} Ct(i).
\end{aligned} \tag{31}$$

The first two identities give a recurrence relation which should be obvious when we realize that in the standard enumeration of pairs there are exactly $Ct\,|x|$ numbers with the pair size $|x|$ which are between the first in $\prec$ number with the pair size $|x|$ and the first in $\prec$ number with the next higher pair size $|x|+1$. The total number of numbers with the pair size $|x|$ is the number of different binary trees with $|x|$ internal nodes as given by a Catalan number introduced in the section 2. The last two identities replace the recurrences by an explicitly defined function $Mins_1(n)$ yielding the ordinal of the first in $\prec$ number with the pair size n. Note that the function $Mins_1$ does not depend on the pairing function p. The explicit definition (6) of Ct is elementary because the binomial coefficients are elementary functions. Hence the function $Mins_1$ is elementary.

We will need the operator of *bounded minimum* which yields the function $f(y,\boldsymbol{x})$ from a predicate P (i.e. from its characteristic function):

$$f(y,\boldsymbol{x}) = \mu\, z \le y\,[\, P(z,\boldsymbol{x})\,]$$

such that $f(y,\boldsymbol{x})$ is the smallest $z \le y$ such that $P(z,\boldsymbol{x})$ holds or 0 otherwise. The classes $\mathcal{E}^i$ are closed under bounded minimimum.

Combining the properties (25), (26), and (31) yields:

$$Mins_1\,|x| \le Ord(x) < Mins_1(|x|+1).$$

This suggests an elementary function Sz which is independent of p and for which we have

$$\begin{aligned} |x| &= Sz\; Ord(x) \qquad (32)\\ Sz(x) &= \mu\, n \le x\,[\, Mins_1(n) \le x < Mins_1(n+1)\,]. \end{aligned}$$

The bound on the operator of bounded minimum comes from

$$|x| \le Ord(x) \qquad (33)$$

which can be seen true by observing that the number of different left-leaning numbers y for which $Lt(y) = y \prec x$ is exactly $|x|$.

Lemma 9.2 (Pair size and the order of pairs) There is a strictly increasing elementary function A independent of the pairing function p such that

1. $|A(n)| = n$,
2. $|x| \le Ord(x) \le A\,|x|$.

Proof: The first inequality of the property (2) is (33). The second inequality of (2) together with the property (1) say that $A(n)$ is the ordinal of the largest number in $\prec$ with the pair size n. Hence we have an explicit elementary definition $A(n) = Mins_1(n+1) \dot{-} 1$. The function A is strictly increasing because the function $Mins_1$ is. ■

We note that

$$Ord\,Lt(x) = A\,|x|. \tag{34}$$

The numbers of the form $\langle Rt(v), Rt(w)\rangle$ are the first numbers in the order $\prec$ with the head size $|v|$ and tail size $|w|$. The ordinals of such numbers are given by the next set of properties

$$\begin{aligned}
Ord\langle 0, Rt(w)\rangle &= Ord\,Rt\langle 0, w\rangle = Mins_1(|w|+1) \qquad (35)\\
Ord\langle\langle 0, v\rangle, Rt(w)\rangle &= Ord\langle v, 0, Rt(w)\rangle + Ct\,|v| \cdot Ct(|w|+1)\\
Ord\langle Rt(v), Rt(w)\rangle &= Mins_2(|v|, |w|)\\
Mins_2(n, m) &= Mins_1(n+m+1) + \sum_{i<n} Ct(i) \cdot Ct(n+m-i).
\end{aligned}$$

The first two identities are recurrent and they should be obvious when we realize that the product of the two Catalan numbers gives the total of numbers with the head size $|v|$ and tail size $|w|+1$. The explicitly defined function $Mins_2(n, m)$ gives the ordinal of the first in $\prec$ number with the head size n and tail size m. The function $Mins_2$ is elementary and independent of the pairing function p.

Given a pair $\langle v, w\rangle$ we observe that its position in the standard enumeration of pairs is in the the group of numbers with the head size $|v|$ and tail size $|w|$. The group contains $Ct\,|v| \cdot Ct\,|w|$ numbers. The group can be divided into $Ct\,|v|$ segments with each segment containing $Ct\,|w|$ numbers. At the j-th position in the i-th segment we find the pair $\langle Mins_1\,|v| + i, Mins_1\,|w| + j\rangle$. Hence for the ordinal of the pair $\langle v, w\rangle$ we have:

$$\begin{aligned}
Ord\langle v, w\rangle &= Mins_2(|v|, |w|) + Ct\,|w| \cdot (Ord(v) - Mins_1\,|v|) +\\
&\quad Ord(w) - Mins_1\,|w|. \qquad (36)
\end{aligned}$$

If we replace the applications of the pair size function in the right hand side by the terms obtained from (32) we can write the identity (36) as

$$\begin{aligned}
Ord\langle v, w\rangle &= P(Ord(v), Ord(w)) \qquad (37)\\
P(x, y) &= Mins_2(Sz(x), Sz(y)) + Ct\,Sz(y) \cdot R(x) + R(y) \qquad (38)\\
R(x) &= x - Mins_1\,Sz(x). \qquad (39)
\end{aligned}$$

Elementary functions P and R are independent of the pairing function p. We have:

Theorem 9.3 The function P from (38) is a pairing function such that for any pairing function p the bijection Ord_p is a pair isomorphism from p to P and

1. $Ord_P(x) = x$,
2. $x + 1 = Succ_P(x).$,
3. $x \dot{-} 1 = Pred_P(x).$,

4. $|x|_P = Sz(x)$

5. $x \prec_P y \leftrightarrow x < y$.

Proof: We note that the identity (37) is a shorthand for $Ord_p\, p(v,w) = P(Ord_p(v), Ord_p(w))$ which can be also written in the form

$$P(x,y) = Ord_p\, p(Ord_p^{-1}(x), Ord_p^{-1}(y)).$$

Since the bijection Ord_p does not move zero we get that Ord_p is a pair isomorphism from p to P by the theorem 7.1(3).

The property (1) is obtained by observing that Ord_P is an isomorphism from P to P which must be an identity. Properties (2) and (3) follow from the property (1) and from the identities (29,30). The property (4) follows from $|x|_P = Sz\; Ord_P(x) = Sz(x)$ by the identity (32) and property (1). The property (5) is a consequence of (1) and the equivalence (25). ■

$P(x,y)$	0	1	2	3	4	5	6	$\cdots$
0	1_1	2_2	4_3	5_3	9_4	10_4	11_4	$\cdots$
1	3_2	6_3	14_4	15_4	37_5	38_5	39_5	$\cdots$
2	7_3	16_4	42_5	43_5	121_6	122_6	123_6	$\cdots$
3	8_3	17_4	44_5	45_5	126_6	127_6	128_6	$\cdots$
4	18_4	46_5	131_6	132_6	399_7	400_7	401_7	$\cdots$
5	19_4	47_5	133_6	134_6	404_7	405_7	406_7	$\cdots$
6	20_4	48_5	135_6	136_6	409_7	410_7	411_7	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

Table 2: Pairing function $P(x,y)$

The function P is tabulated in the table 2. The subscripts of the values of $P(x,y)$ give the size $|P(x,y)|$.

Until the end of the section we will write $\langle x,y\rangle$ for $P(x,y)$ and stop showing the parameterization of functions by the subscript P.

We derive some bounds on the functions Ct, $Mins_1$, and P. The explicit formula (6) for the Catalan numbers can be turned into a recurrent one by a straightforward manipulation of binomial coefficients:

$$\begin{aligned} Ct(0) &= 1 \\ Ct(n+1) &= \frac{4n+2}{n+2} \cdot Ct(n). \end{aligned} \tag{40}$$

By a simple induction proof we get from the recurrence relation exponential bounds on $Ct(n)$:

$$2^{n-1} \leq Ct(n) \leq 4^n. \tag{41}$$

Directly from the definition of $Mins_1$ we have

$$Ct(n) \leq Mins_1(n+1). \tag{42}$$

For the inequality

$$Mins_1(n) \leq Ct(n) \tag{43}$$

we observe that for $x < Mins_1(n)$ we have $|x|_P < n$ and the set of numbers

$$\{\langle Mins_1(n - |x| - 1), x\rangle \mid x < Mins_1(n)\}$$

with the pair size n has the cardinality $Mins_1(n)$ which cannot exceed $Ct(n)$. For $x \geq 1$ we have $|x| \geq 1$ and

$$2^{|x|-2} \leq Ct(|x| - 1) \leq Mins_1 \, |x| \leq x.$$

From this we get for all x

$$2^{|x|} \leq 4x + 1 \tag{44}$$

$$|x| \leq d(x) + 2. \tag{45}$$

For any x we have

$$x < Mins_1(|x| + 1) \leq Ct(|x| + 1) \leq 4^{|x|+1} = 2^{2|x|+2}. \tag{46}$$

Hence

$$d(x) \leq 2|x| + 2. \tag{47}$$

From the formulas (41,42) we get a useful inequality

$$2^{|a|} \leq Ct(|a| + 1) \leq Mins_1(|a| + 2) = Mins_1 \, |a \oplus \langle 0, \mathbf{1}\rangle| \leq a \oplus \langle 0, \mathbf{1}\rangle. \tag{48}$$

For the function Lt we have $|Lt(x)| = |x|$ and from the formulas (34,46,44) we get

$$A \, |x| = Lt(x) < 2^{2|x|+2} = (2^{|x|})^2 \cdot 2^2 \leq 4(4x + 1)^2.$$

Compare this with the superexponential bound on $B \, |x|_J = Lt_J(x) \leq 10^{2^{|x|_J}}$. The function Lt_J cannot be in $\mathcal{E}^2$, the function Lt_P can. The theorem 11.6 from the section 11 confirms this. The polynomial bound on the function Lt_P comes about because the pairing function keeps together numbers with the same pair size. The price paid for this is that P grows faster than J as the comparison of tables (1) and (2) reveals. The pairing function J is given by a polynomial of the degree 2 whereas the function P is bounded by a polynomial of the degree 4:

$$\begin{aligned}\langle x, y\rangle &< Mins_1(|x| + |y| + 2) \leq Ct(|x| + |y| + 2) \leq \\ &4^{|x|+|y|+2} \leq (2^{|x|})^2 \cdot (2^{|y|})^2 \cdot 16 \leq 16(4x + 1)^2 \cdot (4y + 1)^2.\end{aligned} \tag{49}$$

10 Pair Order Hierarchy

We now address the problem (ii) of short iteration from the section 9. The unary equivalent of limited recursion (1) gives sufficiently long iteration sequences but it is not pair invariant. We now define a pair invariant operator of recursion which has comparable iteration sequences. The operator generates a new pair based hierarchy.

Definition 10.1 (Pair order hierarchy) Let p be a pairing function. The function f is defined by *limited p-pair order recursion* from unary functions g, h, and j if

$$\begin{aligned} f(0) &= 0 \\ f\langle 0, x\rangle &= g(x) \\ f\langle Succ_p(y), x\rangle &= h\langle y, x, f\langle y, x\rangle\rangle \\ f\langle y, x\rangle &\preceq_p j\langle y, x\rangle. \end{aligned} \tag{50}$$

For a unary function f the class of functions $\mathcal{PO}_p(f)$ is generated from the functions f, $Succ_p$, C_p and T_p by composition of unary functions, p-pairing, and limited p-pair order recursion.

The *Pair order hierarchy* consists for $i = 2, 2.5, 3, 4, \ldots$ of classes

$$\mathcal{PO}^i_p = \mathcal{PO}_p(F^p_i).$$

Pair order recursion is justified by $Succ_p$-induction. Note that for the pairing function P the unary version of the operator of (limited) recursion coincides with the operator of (limited) P-pair order recursion.

Lemma 10.1 The operator of limited p-pair order recursion (50) is pair invariant.

For a pair isomorphism π from p to q we have to show that if f is derived from g, h, and j by limited p-pair order recursion then the function f^π is derived from the functions g^π, h^π, j^π by limited q-pair order recursion. Clearly, it is sufficient to show that f^π together with g^π, h^π, and j^π satisfies the formulas (50) for the pairing function q. The proof of the three identities is straightforward and uses the fact that $Succ_q = Succ^\pi_p$ which holds by the theorem 7.3 because $Succ_p \in \mathcal{P}^2_p$ and $\mathcal{P}^2_p$ is pair invariant. It remains to show the bounding condition. We have $f\,p(x, y) \preceq_p j\,p(x, y)$. Hence also $f\,p(\pi^{-1}(x), \pi^{-1}(y)) \preceq_p j\,p(\pi^{-1}(x), \pi^{-1}(y))$. Thus $f\,\pi^{-1}q(x, y) \preceq_p j\,\pi^{-1}q(x, y)$ and by the lemma 9.1 also $\pi\,f\,\pi^{-1}q(x, y) \preceq_q \pi\,j\,\pi^{-1}q(x, y)$, i.e. $f^\pi\,q(x, y) \preceq_q j^\pi\,q(x, y)$. ■

Until the end of this section we fix the pairing function p and stop showing the parameterization.

Lemma 10.2 The classes $\mathcal{PO}^i$ of Pair order hierarchy are closed under clausal definitions.

Proof: The proof is similar to the proof of the inclusion $\mathcal{P}^i \subseteq \mathcal{F}^i$ of the theorem 6.2. Let i be one of $2, 2.5, 3, \cdots$. It is clearly sufficient to derive $Z, D \in \mathcal{PO}^i$ as $I(x) = C\,C(x)$. The zero function is derived in $\mathcal{PO}^i$ as

$$\begin{aligned}
Z(x) &= f\langle I(x), I(x)\rangle \\
f(0) &= 0 \\
f\langle 0, x\rangle &= I(x) \\
f\langle Succ(y), x\rangle &= h\langle y, x, f\langle y, x\rangle\rangle \\
f\langle y, x\rangle &\preceq T\langle y, x\rangle \\
h(z) &= T\,T\,T(z).
\end{aligned}$$

As $H(x) = T\,C(x)$ the conditional function is derived in $\mathcal{PO}^i$ as

$$\begin{aligned}
D(0) &= 0 \\
D\langle 0, x\rangle &= H(x) \\
D\langle Succ(y), x\rangle &= h\langle y, x, D\langle y, x\rangle\rangle \\
D\langle y, x\rangle &\preceq T\langle y, x\rangle \\
h(z) &= T\,H\,T(z). \ \blacksquare
\end{aligned}$$

With explicit and clausal definitions available in $\mathcal{PO}^i$ we can generalize the schema (50) of (limited) pair order recursion by permitting terms in its right hand sides. The functions g, h (j) can be then explicitly defined from the terms. We will show the generalization in the definitions below by using the symbol $:=$ instead of $=$ and by omitting the first identity $f(0) = 0$ altogether. This is similar to the generalization of pair recursion in the theorem 4.5. Exactly as in the proof of this theorem we can reduce the (limited) pair order recursion without a parameter to (limited) pair order recursion.

Lemma 10.3 The classes $\mathcal{PO}^i$ of Pair order hierarchy are closed under limited pair iteration (12).

Proof: The proof is again similar to the proof of the inclusion $\mathcal{P}^i \subseteq \mathcal{F}^i$ of the theorem 6.2. Let i be one of $2, 2.5, 3, \cdots$. We first derive the predicate $\mathit{Left} \in \mathcal{PO}^i$. We observe that x is a left-leaning tree iff $Succ(x)$ is a right-leaning tree iff $x = 0 \vee x = \langle v, 0\rangle$ and $Succ(x) = \langle 0, w\rangle$ for some v and w:

$$\begin{aligned}
\mathit{Left}(x) &:= f\langle Succ(x), x\rangle \\
f\langle \mathbf{1}, 0\rangle &:= 0 \\
f\langle\langle 0, w\rangle, v, 0\rangle &:= 0 \\
g(x) &:= \mathbf{1}.
\end{aligned}$$

Let f be defined by limited pair iteration from functions $g, j \in \mathcal{PO}^i$. We derive f in $\mathcal{PO}^i$ by limited pair order recursion. For that we observe that there are exactly $|x|$ left-leaning trees preceding x in the order $\prec$:

$$\begin{array}{rcl} f\langle 0, y\rangle & := & y \\ f\langle Succ(x), y\rangle & := & D\langle Left(x), g\, f\langle x, y\rangle, f\langle x, y\rangle\rangle \\ f\langle x, y\rangle & \preceq & \langle 0, j\langle x, y\rangle\rangle. \end{array}$$

By an easy *Succ*-induction proof on x we show $f\langle x, y\rangle = g^{|x|}(y)$. From $|g^{|x|}(y)| \leq |j\langle x, y\rangle|$ we have $g^{|x|}(y) \preceq \langle 0, j\langle x, y\rangle\rangle$. This justifies the bounding condition. ■

Theorem 10.4 The Pair order hierarchy is a proper hierarchy as we have

1. $\mathcal{PO}^2 \subset \mathcal{PO}^{2.5} \subset \mathcal{PO}^3$,
2. $\mathcal{P}^i \subseteq \mathcal{PO}^i$ for $i = 2, 2.5$,
3. $\mathcal{PO}^{i+3} = \mathcal{P}^{i+3}$.

Proof: (1): The bounds on the pair size of $\mathcal{P}^i$ functions from the lemma 5.3 apply to $\mathcal{PO}^i$ functions as well because the two classes differ in the length of iteration only. This means that the function $F_{2.5}$ is not in $\mathcal{PO}^2$ and the function F_3 is not in $\mathcal{PO}^{2.5}$.

(2) + $\mathcal{P}^{i+3} \subseteq \mathcal{PO}^{i+3}$: This follows directly from the lemma 10.3.

In order to show (3) it suffices now to prove $\mathcal{PO}^{i+3} \subseteq \mathcal{P}^{i+3}$. Let $i \geq 3$. The basic function *Succ* of $\mathcal{PO}^i$ is already in $\mathcal{P}^2$ thus it is sufficient to show $\mathcal{P}^i$ closed under limited pair order iteration (50). Let us assume for a moment that we have succeeded to define in $\mathcal{P}^3$ the functions $m(x) = Ord^{-1} Len(x)$ and $n(x) = Mins_1\, Ord(x)$. We have then $Len(x) = Ord\, m(x)$, $Len\, n(x) = |n(x)| = Ord(x)$, and

$$m\, n(x) = Ord^{-1} Len\, n(x) = Ord^{-1}|n(x)| = Ord^{-1}\, Ord(x) = x. \qquad (51)$$

Let f be defined by (50) from $g, h, j \in \mathcal{P}^i$. We define f_1 by unary pair recursion:

$$\begin{array}{rcl} f_1\langle 0, x\rangle & := & g(x) \\ f_1\langle\langle v, w\rangle, x\rangle & := & h\langle m(w), x, f_1\langle w, x\rangle\rangle. \end{array}$$

By a straightforward pair induction proof on y we can show that $f_1\langle y, x\rangle = f\langle m(y), x\rangle$. Thus $f_1\langle y, x\rangle \preceq j\langle m(y), x\rangle$, i.e. $|f_1\langle y, x\rangle| \leq |j\langle m(y), x\rangle|$ and $f_1 \in \mathcal{P}^i$. The function f is then derived in $\mathcal{P}^i$ by a clasual definition $f\langle y, x\rangle := f_1\langle n(x), x\rangle$ which is justified by the equation (51).

The function m can be defined even in $\mathcal{P}^2$ by unary pair recursion without parameters:

$$\begin{array}{rcl} m(0) & := & 0 \\ m\langle v, w\rangle & := & Succ\, m(w) \\ |m(x)| & \leq & Ord\, m(x) = Len(x) \leq |x|. \end{array}$$

The function n is defined by pair iteration:

$$\begin{aligned} n(x) &:= T\, g^{|b(x)|}\langle x, 0\rangle \\ g\langle 0, a\rangle &:= \langle 0, a\rangle \\ g\langle\langle v, w\rangle, a\rangle &:= \langle Pred\langle v, w\rangle, 0, a\rangle \end{aligned}$$

where $b \in \mathcal{P}^3$ is obtained by the lemmas 9.2 and 8.6 to satisfy $Ord(x) \le A\,|x| \le |b(x)|$. Thus clearly $n(x) = Mins_1\, Ord(x)$. It remains to derive a $\mathcal{P}^3$ bounding condition for the iteration of g. We have

$$g^{|y|}\langle x, a\rangle = \langle Pred^{|y|}(x), \overbrace{0, \cdots, 0,}^{min(|y|, Ord(x))} \quad a\rangle.$$

Now $Pred^{|y|}(x) \preceq x$, i.e. $|Pred^{|y|}(x)| \le |x|$. Hence we have

$$\begin{aligned} &|g^{|y|}\langle x, a\rangle| = |Pred^{|y|}(x)| + min(|y|, Ord(x)) + |a| + 1 \le \\ &\quad |x| + Ord(x) + |a| + 1 \le |x| + A\,|x| + |a| + 1 \le |c\langle x, a\rangle| \end{aligned}$$

where the $\mathcal{P}^3$ function c is obtained by the lemma 8.6. Thus we can use the bounding condition $|g^{|y|}(z)| \le |h\langle y, z\rangle|$ where $h\langle y, x, a\rangle := c\langle x, a\rangle$. ■

Definition 10.2 The predicate P, i.e. its characteristic function, is defined from a predicate Q by *bounded existential quantification* if $P(y, \boldsymbol{x}) \leftrightarrow \exists z \le y\, Q(z, \boldsymbol{x})$. *Bounded universal quantification* is defined in a similar way.

We now define pair equivalents of the operators of bounded minimum and quantification. Let p be a pairing function. The function f is defined from a predicate P by *bounded p-pair order minimum*, in writing

$$f\langle y, x\rangle = \mu\, v \preceq_p y\, [\, P\langle v, x\rangle\,],$$

if $f\langle y, x\rangle$ yields the first number v in the order $\prec_p$ such that $Q\langle v, x\rangle$ and 0 otherwise.

The predicate P is defined from a predicate Q by *bounded p-pair order existential quantification* if $P\langle y, x\rangle \leftrightarrow \exists z \preceq_p y\, Q\langle z, x\rangle$. *Bounded p-pair order universal quantification* is defined in a similar way.

The proof of the following theorem is left to the reader.

Theorem 10.5 The operators of bounded p-order minimum and quantification are pair invariant and for the pairing function P they reduce to the unary versions of bounded minimum or of the corresponding bounded quantification respectively. ■

Theorem 10.6 The classes $\mathcal{PO}^i$ are closed under the operators of bounded pair-order minimum and quantification.

Proof: We prove the closure under bounded pair-order minimum and leave the similar proofs of the other two operators to the reader. Let i be one of $i = 2, 2.5, 3, \cdots$ and f be defined from $g \in \mathcal{PO}^i$ by bounded pair-order minimum. The function f is derived in $\mathcal{PO}^i$ by limited pair order recursion:

$$\begin{aligned} f\langle 0, x\rangle &:= 0 \\ f\langle Succ(y), x\rangle &:= D\langle g\langle f\langle y, x\rangle, x\rangle, f\langle y, x\rangle, \\ &\qquad D\langle g\langle Succ(y), x\rangle, Succ(y), 0\rangle\rangle. \\ f\langle y, x\rangle &\preceq y. \ \blacksquare \end{aligned}$$

11 Characterization of Pair Order Hierarchy

We first solve the problem (iii) from the section 9 of a gap in the subelementary part of Grzegorczyk hierarchy.

Definition 11.1 (Subelementary function classes) A class $\mathcal{F}$ of functions is *subelementary* if there is a number k such that for every n-ary $f \in \mathcal{F}$ we have $f(x_1, \ldots, x_n) \leq E_3^k(x_1 + \ldots + x_n)$.

This definition does not exclude classes which contain the exponential function but restrict its composition. Classes of Ritchie hierarchy [9], or classes of functions computable on Turing machines in exponential time using binary coding of numbers ($\boldsymbol{Exp}$) are such. The subelementary classes we are interested in here are smaller as they do not contain the exponential function. There is an important subelementary class which we have denoted by $\mathcal{E}^{2.5}$ in [17]:

Definition 11.2 (Augmented Grzegorczyk hierarchy) Define the *subexponential hierarchy function* $E_{2.5}(x, y) = x^{d(y)}$ and $\mathcal{E}^{2.5} = \mathcal{E}(E_{2.5})$. The hierarchy $\mathcal{E}^i$ for $i = 0, 1, 2, 2.5, 3, \ldots$ is the *augmented* Grzegorczyk hierarchy.

We have proved in [17] that $\mathcal{E}^2 \subset \mathcal{E}^{2.5} \subset \mathcal{E}^3$ and that for every n-ary function $f \in \mathcal{E}^{2.5}$ there is a polynomial $q(\boldsymbol{x})$ such that $f(\boldsymbol{x}) \leq 2^{q(\overline{\boldsymbol{x}})}$ for all x_i. The notation $q(\overline{\boldsymbol{x}})$ stands for $q(d(x_1), d(x_2), \ldots, d(x_n))$. Thus the exponential function is not in $\mathcal{E}^{2.5}$.

We now investigate the relation between the Pair order and augmented Grzegorczyk hierarchy. The relation is, of course, dependent on the pairing function used. In the following discussion we therefore fix the pairing function p to the function P and drop the parameterization. If we use a different pairing function than P we show it explicitly. The theorem 11.6 which fully characterizes the two hierarchies as $\mathcal{PO}^i = \mathcal{E}_1^i$ requires some lemmas. The lemma 11.1 introduces the pairing function P and its projection functions in $\mathcal{E}^2$. The next three lemmas which correspond to the lemmas 8.9, 8.10, and 8.11 from the section 8 deal with the subelementary classes. The fourth lemma 11.5 connects the hyperelementary classes $\mathcal{PO}_J^{i+3}$ and $\mathcal{PO}_P^{i+3}$.

Although the function P was introduced in (38) with the help of elementary functions the bound (49) suggests that it is in $\mathcal{E}^2$. This is because the elementary functions defining P are applied to the sizes of arguments which are logarithmic:

Lemma 11.1 The pairing function P and its projection functions H and T are in $\mathcal{E}^2$.

Proof: The detour through elementary functions applied to small arguments can be eliminated by defining *bounded versions* $f(x)_b$ of functions $f(x)$ where $f(x)_b = min(f(x), b)$. Provided that $f(x) \leq b$ we have $f(x)_b = f(x)$. The bounded functions $Ct(n)_b$ and $Mins_1(n)_b$ are derived in $\mathcal{E}^2$ by limited recursion using the recurrent relation (40) and by bounded sum respectively:

$$\begin{aligned}
Ct(0)_b &= min(1, b) \\
Ct(n+1)_b &= min(Ct(n)_b \cdot (4n+2) \div (n+2), b) \\
Ct(n)_b &\leq b \\
Mins_1(n)_b &= min(\sum_{i<n} Ct(i)_b, b).
\end{aligned}$$

The functions $|x| = Sz(x)$ and R are derived in $\mathcal{E}^2$ without bounds:

$$\begin{aligned}
|x| &= \mu\, n \leq x\,[\, Mins_1(n+1)_{4(4x+1)^2}\,] \\
R(x) &= x \dot{-} (Mins_1\, |x|)_x.
\end{aligned}$$

The bound on the second occurrence of $Mins_1$ in the definition of $|x|$ comes from the formulas (44,46) as a bound on $Ct(|x|+1)$.

The bounded version of the function $Mins_2$ is derived in $\mathcal{E}^2$ by bounded summation:

$$Mins_2(n, m)_b = min(Mins_1(n+m+1)_b + \sum_{i<n} Ct(i)_b \cdot Ct(n+m \dot{-} i)_b, b).$$

The derivation of P in $\mathcal{E}^2$ uses the bound (49):

$$\begin{aligned}
P(x, y) &= Mins_2(|x|, |y|)_{b(x,y)} + (Ct\,|y|)_{b(x,y)} \cdot R(x) + R(y) \\
b(x, y) &= 16(4x+1)^2 \cdot (4y+1)^2.
\end{aligned}$$

Finally, the projection functions are derived in $\mathcal{E}^2$ by bounded minimum:

$$\begin{aligned}
H(x) &= \mu\, v \leq x\,[\,\exists w \leq x\; P(v, w) = x\,] \\
T(x) &= \mu\, w \leq x\,[\,\exists v \leq x\; P(v, w) = x\,]. \;\blacksquare
\end{aligned}$$

We note that by putting bounds on addition, multiplication, bounded sums and by a more careful coding of the function $|x|$ we can derive the bounded functions $Ct(n)_b$, $Mins_1(n)_b$, $Mins_2(n, m)_b$, as well as the functions $|x|$ and R in $\mathcal{E}^0$. Thus we would get the bounded pairing function $P(x, y)_b$ also in $\mathcal{E}^0$. Consequently, the projection functions, but not the pairing function itself, can be derived in $\mathcal{E}^0$.

Lemma 11.2 The unary equivalents of binary functions of addition, modified subtraction $\dot{-}$, minimum min, and multiplication are in the class $\mathcal{PO}^2$.

Proof: The function $+^+$ is derived by P-pair order recursion which is, of course, the unary version of recursion in the usual way as $0 + y := y$ and $s(x) +$

$y := s(x+y)$. The $\mathcal{PO}^2$ bounding condition is obtained from the inequalities (46) using the monotone properties of the functions $Mins_1$ and Ct:

$$\begin{aligned} x+y < 2\cdot Mins_1\,|\langle x,y\rangle| \le \\ Mins_1\,|\langle x,y\rangle| + Ct\,|\langle x,y\rangle| = Mins_1\,|\langle 0,x,y\rangle| \le \langle 0,x,y.\rangle \end{aligned}$$

The modified subtraction and the function min^+ are derived in $\mathcal{PO}^2$ as follows:

$$\begin{array}{rcl} x \mathbin{\dot-} y & := & Pred^y(x) \\ Pred^y(x) & \le & x \\ min^+\langle x,y\rangle & := & D\langle x \mathbin{\dot-} y, x, y\rangle. \end{array}$$

The unary version of multiplication is obtained by the unary version of recursion in the usual way. A $\mathcal{PO}^2$ bound on the function $\cdot^+$ is obtained from the formulas (48,47) as follows:

$$x\cdot y \le 2^{d(x)+d(y)} \le 2^{2(|x|+|y|)+4} = 2^{|\langle 0,x,y,x,y\rangle|} \le \langle 0,x,y,x,y\rangle \oplus 2. \ \blacksquare$$

Lemma 11.3

1. $d \in \mathcal{PO}^2$,
2. $E^+_{2.5} \in \mathcal{PO}^{2.5}$.

Proof: We use the functions from the lemma 11.2. (1): The derivation od d needs three auxiliary functions $x \div 2$, $Even(x) = 0$ iff x is even, and $Pow(x) = 0$ iff x is a power of two.

$$\begin{array}{rcl} d(0) & := & 0 \\ d(x+1) & := & D\langle Pow(x+1), d(x)+1, d(x)\rangle \\ d(x) & \le & x \\ Pow(x) & := & D\langle x, 1, g(x,x)\rangle \\ g\langle 0,x\rangle & := & x \\ g\langle y+1,x\rangle & := & D\langle Even\, g\langle y,x\rangle, g\langle y,x\rangle \div 2, 1\rangle \\ g\langle y,x\rangle & \le & x \\ 0 \div 2 & := & 0 \\ (x+1)\div 2 & := & D\langle Even(x), x\div 2, (x\div 2)+1\rangle \\ x\div 2 & \le & x \\ Even(0) & := & 0 \\ Even(x+1) & := & D\langle Even(x), 1, 0\rangle \\ Even(x) & \le & 1. \end{array}$$

(2): The subexponential function $x^d(y)$ is obtained by the composition of the exponential function and the bit size function d. In order to eliminate the detour through $\mathcal{PO}^3$ we derive the bounded exponential $(x^y)_b = f\langle y, x, b\rangle = min(x^y, b)$ in $\mathcal{PO}^{2.5}$:

$$\begin{array}{rcl} f\langle 0, x, b\rangle & := & min^+\langle 1, b\rangle \\ f\langle y+1, x, b\rangle & := & min^+\langle x \cdot f\langle y, x, b\rangle, b\rangle \\ f\langle y, v\rangle & \leq & T(v). \end{array}$$

We obtain a $\mathcal{PO}^{2.5}$ bound on the subexponential function with the help of the formulas (47,48) as follows:

$$x^{d(y)} \leq 2^{d(x)\cdot d(y)} \leq 2^{(2|x|+2)\cdot(2|y|+2)} = 2^{|\langle 0,x,x\rangle \otimes \langle 0,y,y\rangle|} \leq \langle 0, x, x\rangle \otimes \langle 0, y, y\rangle \oplus 2$$

and set $x^{d(y)} := f\langle d(y), x, \langle 0, x, x\rangle \otimes \langle 0, y, y\rangle \oplus 2\rangle$. ■

Lemma 11.4 For $i = 2, 2.5$ the unary equivalent f^+ of every $\mathcal{E}^i$ function f is in the class $\mathcal{PO}^i$.

Proof: By induction on construction of functions of $\mathcal{E}^i$. The functions Z and s are unary and $Z = Z^+ \in \mathcal{PO}^i$ and $s = s^+ = Succ \in \mathcal{PO}^i$. The unary equivalent of the hierarchy function E_i is in $\mathcal{PO}^i$ by the lemmas (11.2,11.3). The unary equivalents of projection functions U_i^n are shown in $\mathcal{PO}^i$ exactly as in the proof of the lemma 8.11. The same applies for the inductive step when f is defined by composition.

If $f(y, \boldsymbol{x})$ is defined by limited recursion (1) from the functions g, h, and j then we use the inductive assumptions in the derivation of $f^+ \in \mathcal{PO}^i$ by limited (pair order) recursion:

$$\begin{array}{rcl} f^+\langle 0, \boldsymbol{x}\rangle & := & g^+\langle \boldsymbol{x}\rangle \\ f^+\langle y+1, \boldsymbol{x}\rangle & := & h^+\langle y, \boldsymbol{x}, f^+\langle y, \boldsymbol{x}\rangle\rangle \\ f^+\langle y, x\rangle & \leq & j^+\langle y, x\rangle. \end{array}$$ ■

Lemma 11.5 $Ord_J, Ord_J^{-1} \in \mathcal{E}^3$.

Proof: The ordinal in $\prec_J$ of x is the number of iterations of the successor function $Succ_J$ needed to reach x from 0. We have $Succ_J \in \mathcal{P}_J^2 \subseteq \mathcal{P}_J^3 = \mathcal{E}_1^3$ by the theorem 8.12. We define the elementary Ord_J as

$$Ord_J(x) = \mu\, n \leq B(x) + 1\, [\, Succ_J^n(0) = x\,].$$

The bound comes from the observation that for $y \prec_J x$ we have $|y|_J \leq |x|_J$, i.e. $y \leq B\,|x|_J \leq B(x)$. Since the number of y's preceding x in $\prec_J$ is $Ord_J(x)$ we have $Ord_J(x) \leq B(x) + 1$.

For the inverse function Ord_J^{-1} we have $Ord_J^{-1}(x) = Succ_J^x(0)$ and we can derive it by limited recursion in $\mathcal{E}^3$:

$$\begin{array}{rcl} Ord_J^{-1}(0) & = & 0 \\ Ord_J^{-1}(x+1) & = & Succ_J\, Ord_J^{-1}(x) \\ Ord_J^{-1}(x) & \leq & B\, Sz(x). \end{array}$$

The bounding condition comes from $Sz(x) = |x|_P = |Ord_J^{-1}(x)|_J$ which holds because Ord_J^{-1} is a pair isomorphism from P to J. ∎

Theorem 11.6 (Characterization of Pair order hierarchy) The classes $\mathcal{PO}^i_P$ of Pair order hierarchy are the classes of unary functions of the corresponding classes of the augmented Grzegorczyk hierarchy, i.e. we have

$$\mathcal{PO}^i_P = \mathcal{E}^i_1$$

for $i = 2, 2.5, 3, \cdots$.

Proof: We start with the subelementary classes where $i = 2$ or $i = 2.5$.

$\mathcal{PO}^i \subseteq \mathcal{E}^i$: Functions P, H, and T are in $\mathcal{E}^2$ and thus also in $\mathcal{E}^{2.5}$ by the lemma 11.1. The function C is derived in $\mathcal{E}^2$ as follows:

$$C(x) = \begin{cases} P(T(x), H(x)) & \text{if } x > 0 \\ 0 & \text{otherwise.} \end{cases}$$

The class $\mathcal{E}^i$ is closed under unary composition and also under pairing. Since $\mathcal{E}^i$ is closed under limited recursion it can be easily shown closed under the unary version of it to which (50) reduces for the pairing function P. It remains to show $F_i \in \mathcal{E}^i$. Since $F_2 = C \in \mathcal{E}^2$ we have $\mathcal{PO}^2 \subseteq \mathcal{E}^2$ and thus also $\oplus$, $\mathit{Left} \in \mathcal{E}^2$. We wish to derive the binary equivalent of the function $\otimes$ in $\mathcal{E}^{2.5}$ as we then can derive $F_{2.5}(x) = x \otimes x \oplus 2$ in $\mathcal{E}^{2.5}$ by explicit definition. Consider the function f defined by recursion

$$\begin{aligned} f(0, x) &= x \\ f(y + 1, x) &= \begin{cases} x \oplus f(y, x) & \text{if } \mathit{Left}(y) = 0 \\ f(y, x) & \text{otherwise.} \end{cases} \end{aligned}$$

Since the number of left-leaning trees less than y is exactly $|y|$ we can prove by induction on y that $f(y, x) = y \otimes x$. An $\mathcal{E}^{2.5}$ bound on f is obtained from the formulas (46,45) as follows:

$$\begin{aligned} y \otimes x &\leq 4^{|y \otimes x|+1} = 4^{|y|\cdot|x|+1} \leq \\ &4^{(d(y)+2)\cdot(d(x)+2)+1} = (4^{d(y)} \cdot 4^2)^{d(x)} \cdot 16^{d(y)} \cdot 4^5. \end{aligned} \tag{52}$$

$\mathcal{E}^i_1 \subseteq \mathcal{PO}^i$: This follows from the lemma 11.4 as for a unary $f \in \mathcal{E}^i$ we have $f = f^+ \in \mathcal{PO}^i$.

The hyperelementary cases where $i \geq 3$ are straightforward as by the theorems 11.5, 8.12, 7.6, and 10.4 we have

$$Ord_J, Ord_J^{-1} \in \mathcal{E}^i_1 = \mathcal{P}^i_J = \mathcal{P}^i_P = \mathcal{PO}^i_P. \ ∎$$

We note that the functions Ord_J and Ord_J^{-1} cannot be both in $\mathcal{PO}^{2.5}_P$ as then we would have $Lt_J \in \mathcal{PO}^{2.5}_J = \mathcal{PO}^{2.5}_P = \mathcal{E}^{2.5}$. This cannot be the case as the function $u(x) = Lt_J\, J(0, 0, 0, x)$ assumes a value larger than 5^x for infinitely many numbers x.

12 Conclusions

The importance of pair based hierachies as a basis for semantics of programming languages was discussed in the introduction. Regarding the correlation between small recursive classes and complexity classes we know on the basis of the presented work and results in [17] we know that $P = NP$ iff $\mathcal{P}^{2.5}$ is closed under bounded minimum and $P = PSPACE$ iff $\mathcal{P}^{2.5} = \mathcal{PO}^{2.5}$ iff $\mathcal{P}^{2.5}$ is closed under limited recursion. These, and all the open problems in small recursive classes are extremely hard as they have defied a solution for some 40 years since some of them were originally posed by Grzegorczyk [5] and for some 25 years in complexity theory. Our contribution lies in the discovery of the connection.

In the future work we intend to design a successor to the Trilogy programming language which should admit just the feasible functions, i.e. $\mathcal{P}^{2.5}$ functions, as programs. The language, with a working name Mini-Trilogy, should have a new form of typing which is to be supported by an intelligent interactive proof checker.

Finally, in conclusion we give a brief account of author's work on pairs because we think that it reflects on the interplay of mathematics (especially of logic and theory of recursive functions) with the design and formal semantics of declarative programming languages, and with the complexity theory. This interplay comes about because the modern computer science is gradually becoming mathematics.

The author started to use pairs in 1983 as a simplification of S-expressions of LISP where instead of denumerably many atoms he used a single atom 0. He has conceived the domain of Pairs basically as the domain of binary trees and continued to think of them in this way until the year 1993. In the retrospective it seems strange why he did not go to natural numbers ten years ago because he was right from the beginning aware of the pairing function J offset by one from the standard pairing function. Nevertheless, he axiomatized the domain of pairs by a first order Theory of pairs in [12] (see also [13, 14]). The author has chosen pairs because of their beautiful coding properties as he knew them from the programming language LISP. Pairs formed the basis for the two declarative programming languages of the author called *Trilogy I* [11, 15] designed and implemented in the years 1986-87 and *Trilogy II* [16, 1, 2] designed in 1989 and implemented by the author in the years 1989-1993. Trilogy I was given formal semantics by the Theory of pairs, while Trilogy II by the recursion theory over pairs.

The author started working on recursive theory via pairs as words in 1988 and developed the Pair hierarchy $\mathcal{P}^i$ in the present form (but containing functions over words). He used the natural numbering of hierarchy classes starting from 0. Thus $\mathcal{P}^0$ was the present $\mathcal{P}^1$ (which we did not present here), and $\mathcal{P}^2$ the present $\mathcal{P}^{2.5}$. At that time he was correlating the Pair hierarchy to Grzegorczyk hierarchy by coding.

In the spring of 1993, while working on a still unfinished text on mathematical logic and recursion theory, the author started to interprete pairs as natural numbers via pairing functions. He used the standard pairing function J. By the fall of 1993 he got to the relation between the Pair and Grzegorczyk hierarchies. He then noticed the gap in the Grzegorczyk hierarchy [17], realized that he needed a new pairing function P for the subelementary classes, and finally realized that the Pair hierachy consisted of unary functions of the Augmented bit size hierarchy.

Although the author knew about most of the complexity characterizations of pair-based hierarchies approximately from the year 1990 he then worked with binary trees (words) rather than with natural numbers. The precise connection to the complexity classes was worked out in detail only in the December of 1993 [17]. The pairing function P was shown capable of defining arithmetic operations in polynomial time and linear space only in the present work.

References

[1] P. Borovansky, P.J. Voda, M. Winczer, Design of Trilogy, Research Report, Dept. Comp. Sc. University of Bratislava, 1991.

[2] P. Borovansky, P.J. Voda, Types as Values Polymorphism, in proceedings of SOFSEM conference 1993.

[3] A. Cobham, The intrinsic computational difficulty of functions. In Proc. Int. Conf. Logic, Meth. Phil.(ed Y. Bar Hillel), 24-30, North Holland, Amsterdam, 1965.

[4] M. Davis. Computability and Unsolvability. McGraw Hill, New York. 1958.

[5] A. Grzegorczyk. Some classes of recursive functions. Rozprawy Mate. No. IV, Warsaw 1953.

[6] H. Hermes, Aufzählbarkeit Entscheidbarkeit Berechenbarkeit, Springer Verlag 1978.

[7] R.L. Graham, D.F. Knuth, O. Patashnik, Concrete Mathematics, Addison-Wesley 1989.

[8] A.I. Malcev, Algoritmy i Rekursivnye Funkcii, Nauka Moskva 1965.

[9] R.W. Ritchie, Classes of predictably computable functions. Trans. Am. Math. Soc. (106), 139-73, 1963.

[10] H.E. Rose. Subrecursion, Functions and Hierarchies. Clarendon Press, Oxford 1984.

[11] Trilogy User Manual, CLS Vancouver 1988.

[12] P.J. Voda Theory of Pairs UBC Research Report, University of B.C., Department of Computer Science, August 1984.

[13] P.J. Voda, A View of Programming Languages as Symbiosis of Meaning and Control, New Generation Computing, Vol 3. No. 1, February 1985.

[14] P.J. Voda, Computations of Logic Programs with One Variable Environments, Journal of New Generation Computing, Vol. 4, No. 2, May 1986.

[15] P. J. Voda, Types of Trilogy, Proceedings of the Fifth International Conference on Logic Programming, MIT Press, Cambridge MA, August 1988.

[16] P.J Voda, Recursion-theoretic Semantics of Trilogy, SOFSEM '90, Czechoslovakia, 1990;

[17] P.J. Voda. On a Missing Class in the Grzegorczyk Hierarchy. Submitted for publication, 1994.