Fundamenta Informaticae XX (2010) 1–9
IOS Press

Andrzej Grzegorczyk's Contribution to Computer Science

Andrzej Salwicki *

National Institute of Telecomunication

Szachowa 1, 04-894 Warsaw

Abstract. The paper of Andrzej Grzegorczyk [19] on hierachy of the primitive recursive classes dates for 1953. This is the most frequently cited article of Polish author in computer science literature, having some hundreds citations. Moreover, many citations are in the papers of eminent computer scientists e.g. [10, 24, 33, 34, 31], sometimes the laureates of prestigious awards. In this paper we make an attempt to present Andrzej Grzegorczyk as a computer scientist.

1. Introduction

Andrzej Grzegorczyk does not consider himself as a computer scientist, c.f. bibliography of his works in this volume [25]. However, his results taught us something essential about computability, and several generations of computer scientists have been inspired by them. The most influential work by Andrzej Grzegorczyk is the landmark paper [19] introducing a hierarchy of primitive recursive functions, which is by now broadly known as the Grzegorczyk hierarchy. Apart of the *Grzegorczyk hierarchy* one may encounter in the literature: *Grzegorczyk axiom in modal logic* and *Grzegorczyk logic* [30], *Grzegorczyk induction* [11], *Grzegorczyk rule* [36], and *Grzegorczyk axiom in a logic stronger than intuitionistic logic* [21, 22, 16, 40, 13, 14].

It was astonishing to find a paper [36] on *object oriented language* for modeling of information systems where Grzegorczyk's rule is presented. There is also a paper which cites results of Grzegorczyk in studies of data bases [4].

His newest result on undecidability without arithmetics [23] is probably too new to be appreciated by computer scientists, note however the paper of Švejdar [43] in this volume. The theorem sounds natural for computer scientists. I believe that soon we shall see papers in computer science making

*I still remember the following scene: year 1959, personae: A. Mostowski–then the vice-director of Mathematical Institute, a class of students and a tall person; MOSTOWSKI: Let me introduce Docent Andrzej Grzegorczyk, he just completed his work on a textbook "Mathematical Logic", and wishes to experiment teaching with it. It turned out one of the most inspiring lectures I have attended. Thank you Professor Grzegorczyk!

use of this result. In programming, people create specifications of data structures. These specifications are frequently presented as axiomatic theories. Some of these theories are decidable, e.g. first order specifications of stacks or queues. In most cases the theories of data structures are undecidable. The new technique of [23] may occur very useful in analysing decidability problem of these specifications.

The impact of results of Andrzej Grzegorczyk is immense. There are papers which present and explain the Grzegorczyk hierarchy, other papers apply the idea of subrecursive hierarchy to some new domains, there are papers in complexity of programs which are related to the hierachy, some papers use the ideas of Grzegorczyk to define new programming languages, etc.

2. Grzegorczyk Hierarchy

The result on hierarchy of primitive recursive functions was published in 1953. Today, after more than 50 years this result is still cited and inspires many researchers in theoretical computer science.

How to explain the Grzegorczyk hierachy? Roughly speaking it displays the structure of complexity classes of the set of primitive recursive functions. The set $\mathcal{RP}$ of primitive recursive functions is the least set containing the constant 0, the successor function, projections, and closed under substitution and primitive recursion (see Definitions 2.1-2.2. below).

The results of paper [19] can be resumed as follow: There is an increasing sequence of sets of primitive recursive functions

$$\mathcal{E}^0 \subsetneq \mathcal{E}^1 \subsetneq \mathcal{E}^2 \subsetneq \mathcal{E}^3 \subsetneq \ldots \subsetneq \mathcal{E}^n \ldots$$

and such that

$$\bigcup_{n\in N} \mathcal{E}^n = \mathcal{RP}.$$

Each set $\mathcal{E}^n$ is closed with respect to the operations of substitution and of limited primitive recursion. The initial functions are primitive recursive ones. We present the idea of the proof using the expository papers of [39, 15].

Definition 2.1. The set of *basic computable functions* is the set

$$B = \{Z, S\} \cup \{p_i^n :\ 1 \leq i \leq n,\ n \in N\},$$

where

- $Z :\to N$, the constant *zero*, a *zero*-ary function,
- $S : N \to N$, the *successor* function,
- for each $n \in N$ and for each $i \in \{1, ..., n\}$
  $p_i^n : N^n \to N$, the *projection* functions $p_i^n(x_1, \ldots, x_n) = x_i$.

□

We shall consider various sets of functions closed with respect to two operations: *Substitution* and *Limited Primitive Recursion*.

Definition 2.2. *Substitution*. Let $h_1, ..., h_n$ be functions of r arguments ($r \geq 1,\ n \geq 0$) and g an n-ary function. If for arbitrary arguments $\overline{x} = (x_1, \ldots, x_r))$ we have that the following equality holds

$$f(\overline{x}) = g(h_1(\overline{x}),\ \ldots\ , h_n(\overline{x})$$

then we say that function f is obtained from function g by the substitution of functions $h_1, ..., h_n$.
Primitive Recursion. Let g be a function of r arguments and let h be a function ot $r + 2$ arguments. If for arbitrary arguments $\overline{x} = (x_1, \ldots, x_r))$ we have

$$\begin{cases} f(\overline{x}, 0) = g(\overline{x}) \\ f(\overline{x}, t + 1) = h(\overline{x}, t, f(\overline{x}, t)) \end{cases}$$

then we say that function f is obtained from the functions g and h by the scheme of primitive recursion.
Limited Primitive Recursion. Consider a triplet of functions $\langle g, h, j\rangle$, where g and h are as above and the function j has $r+1$ arguments. A function is defined by the limited primitive recursion if it is defined by primitive recursion and is bounded by the function j, i.e. $f(\overline{x}, t) \leq j(\overline{x}, t)$. □

Let G be a function which has the following property

$$G(0) \geq 2 \wedge (\forall t \in N)\ t < G(t) < G(t + 1).$$

Then the Grzegorczyk classes may be defined with the help of a sequence $\{G_i\}$ of functions which fulfill the following conditions:

(*i*) $G_1(t) = G(t)$,

(*ii*) $G_n(t) = G^t_{n-1}(2)$.

Let us define a following sequence of functions

$$E_1(t) \overset{df}{=} t^2 + 2,$$
$$E_n(t) \overset{df}{=} E^t_{n-1}(2) \qquad \text{for all } n \geq 2.$$

One may check that the sequence satisfies the conditions (*i*) and (*ii*) mentioned above.
Observe the following properties of functions E_n.

Lemma 2.1. For all n, and for all t the following inequalities hold:
$t + 1 \leq E_n(t)$,
$E_n(t) \leq E_{n+1}(t)$,
$E_n(t) \leq E_n(t + 1)$,
$(\forall m)\ E^m_n(t) \leq E_{n+1}(t + m)$. □

In the sequel we shall also need the addition $E_0(t_1, t_2) \overset{df}{=} t_1 + t_2$.
Now, we define the Grzegorczyk classes $\mathcal{E}^n$.

Definition 2.3. The classes $\mathcal{E}^n$ are defined as follow:
$\mathcal{E}^0 \stackrel{df}{=} \langle B; Substitution, Limited\ Primitive\ Recusion\rangle$,
$\mathcal{E}^{n+1} \stackrel{df}{=} \langle B \cup \{E_0\} \cup \{E_n\}; Substitution, Limited\ Primitive\ Recursion\rangle$, for every n.

□

Basing on the previous lemma one can prove the following:

Lemma 2.2. (*The Growth Lemma*)
Let $\bar{t} = (t_1, \ldots t_n) denote a tuple of arguments.$

- The functions in $\mathcal{E}^0$ are bounded by low growth functions $t_i + c_f$:
  $(\forall f \in \mathcal{E}^0)(\exists i, c_f \in N)(\forall \bar{t})\ f(\bar{t}) \leq t_i + c_f$,
- The functions in $\mathcal{E}^1$ are bounded by linear functions:
  $(\forall f \in \mathcal{E}^1)(\exists c_0, c_1, \ldots, c_n \in N)(\forall \bar{t})\ f(\bar{t}) \leq c_0 + c_1 \cdot t_1 + \ \cdots \ + c_n \cdot t_n$,
- For $n \geq 2$, the iterations of growth function E_n bound the functions in $\mathcal{E}^n$:
  $(\forall n \geq 2)(\forall f \in \mathcal{E}^n)(\exists m_f)(\forall \bar{t})\ f(\bar{t}) \leq E_{n-1}^{m_f}(max(\bar{t}))$.

□

Using a diagonalization argument, one can prove that $\mathcal{E}^n \neq \mathcal{E}^m$, for $n \neq m$. Proof uses the fact that each class contains a function whose growth rate increases with the index n of the class concerned. Hence, we arrive to the fundamental result:

Theorem 2.1. There is a growing sequence of sets of primitive recursive functions

$$\mathcal{E}^0 \subsetneq \mathcal{E}^1 \subsetneq \mathcal{E}^2 \subsetneq \mathcal{E}^3 \subsetneq \ldots \subsetneq \mathcal{E}^n \ldots$$

such that

$$\bigcup_{n \in N} \mathcal{E}^n = \mathcal{RP}.$$

Each set $\mathcal{E}^n$ is closed with respect to the operations of substitution and of limited primitive recursion. The initial functions of each class are primitive recursive functions. □

The above result was earlier than the papers on the computational complexity, c.f. Cook [10].
While the argument about rate of growth leads to Theorem 2.1, it is unclear whether these classes define different classes of sets of natural numbers. A set is called to belong to the class $\mathcal{E}_i^*$, if its characteristic function is in $\mathcal{E}^i$. For $i \geq 2$ there is a strict hierarchy of classes $\mathcal{E}_i^*$ [19]. The following problem[1] are these inclusions strict?

$$\Delta_0^N \subsetneq \mathcal{E}_0^* \subsetneq \mathcal{E}_1^* \subsetneq \mathcal{E}_2^*$$

remains one of the most challenging and intriguing questions of recursion theory for already more than 50 years.
Only a limited progress in understanding this problem has been reached: Bel'tyukov [3] showed that $\mathcal{E}_1^* \subsetneq \mathcal{E}_2^*$ implies that $\mathcal{E}_0^* \subsetneq \mathcal{E}_1^*$. Kutyłowski [28] proved that a slight change in the definition of the classes (replacing limited primitive recursion to bounded iteration) leads to equality of the first two classes of relations, i.e. $I_*^0 = I_*^1$. He also showed that simultaneous recursion might be used to refine further the hierarchy between $\mathcal{E}_0^*$ and $\mathcal{E}_2^*$. Further results of this kind have been obtained in [29].

[1] Δ_0^N denotes the class of rudimentary relations

3. Related results

In this section we quote some of related results.

3.1. Complexity of programs

For a computer scientist theorem 2.1 is important, however this importance becomes more evident in the light of work of R.W. Ritchie [37], A. Meyer [32] and Meyer and D. Ritchie [33]. The result of [33] may be abbreviated as follow:
A class of programs, called "Loop programs" is described. Each Loop program consists only of assignment statements and iteration (loop) statements, the latter resembling the FOR statements of the form **for** i:=1 **to** m **do** <statements> **done**, where the value of variable m is calculated earlier and does not change through the loop. The bound on the running time of a Loop program is determined essentially by the length of the program and the depth of nesting of its loops. See also a section on Grzegorczyk Hierarchy in the textbook by Brainerd and Landweber [6].

It is wortwhile mentioning the contributions of M.Kutyłowski [27]. The author recalls the notion of the generalized Grzegorczyk hierarchy and he discusses some problems connected with its initial classes. He establishes equalities between initial generalized Grzegorczyk classes and some Turing machine complexity classes such as P and P*LINSPACE. Using the means of Grzegorczyk classes he proves a certain hierarchy theorem for the class P*LINSPACE. Stack Turing machines are introduced for better description of low complexity classes.

3.2. Presentation of the Grzegorczyk hierarchy

There are several textbooks and articles where one can find an exposition of Grzegorczyk hierarchy: Rose [39], K. Wagner and G. Wechsung [44], Gakwaya [15], Rogers well known monograph [38].

3.3. Applications of the Grzegorczyk hierarchy

Shelah uses the Grzegorczyk hierarchy in order to obtain a better estimation of cost of evaluating van Waerden numbers [42]. Grozea found that well known NP problems like SAT and Hamiltonian Cycle problem are in very low class $\mathcal{E}^{*0}$ [17][2].

3.4. Articles and books on recursion theory

There is a rich set of papers and books where the Grzegorczyk hierarchy is mentioned, we shall quote just a few: Schwichtenberg [41], Wainer [45], Axt [1], Muchnick [34], Cichon & Wainer [8], Bellantoni [2], Mehlhorn [31].

3.5. Extensions of the Grzegorczyk Hierarchy

There are several papers presenting various extensions of the Grzegorczyk hierarchy: Muchnick studied vectorized Grzegorczyk hierarchies [34]. See also: Wainer and Cichon [45, 8], a nonpublished paper

[2] It seems that the result concerns a parallel Grzegorczyk hierarchy, namely, the hierarchy of classes $\mathcal{E}^{*n}$ of arithmetic relations whose characteristic functions belong to corresponding class $\mathcal{E}^n$ of Grzegorczyk Hierarchy.

by Weiermann [46]. Kristiansen and Barra define small Grzegorczyk classes for typed lambda calculus [26].

3.6. Analog Computing and Computing over Real Numbers

Several authors made attempt to move the Grzegorczyk hierarchy into model of analog computing or real number computing. It seems that the pioneer in this direction was the paper by Grzegorczyk [20]. In [5] an analog and machine independent algebraic characterization of elementarily computable functions over the real numbers in the sense of recursive analysis is presented. Authors prove that they correspond to the smallest class of functions that contains some basic functions, and closed by composition, linear integration, and a simple limit schema. They generalize this result to all higher levels of the Grzegorczyk Hierarchy. Mycka and da Costa [35] proved the undecidability of computations over continuous time. See also papers [7], [12].

3.7. Grzegorczyk Rule

The results of Grzegorczyk research on modal and intuistionistic logic are briefly exposed in [30].

In a paper on object oriented language of modeling information systems [36] one finds an application of the following Grzegorczyk's rule(Grz).

$$\frac{\begin{array}{c}\Gamma, [\, G(p)] \\ \vdots\, \pi \\ \text{OR}\{C(p)\ B\}\end{array}}{\text{OR}\{\ \text{FOR}\{\tau x \mid\ G(x) : C(x)\ \}B\}}$$

I am not certain how to explain the origins of the rule. Probably it originates from the papers of Grzegorczyk [21, 22]. Grzegorczyk considered an intermediate logic which results when one adds the following formula

$$\forall x(P \vee Q(x)) \Rightarrow (P \vee \forall x Q(x))$$

to the Heytings axioms of intuitionistic logic. S. Görnemann [16] called it the Grzegorczyk axiom (of an extension of intuitionistic logic) and proved that adding it allows to characterize the class of all Kripke structures of fixed domains, in the following sense: the formulas deducible in this extended intuistionistic logic are exactly the formulas that are valid in all Kripke structures of fixed domains. The name of the axiom was accepted. For example, Rauszer and Sabalski use it in their interesting paper [40].

3.8. Grzegorczyk's Results in Topology applied in Computer Science

In 1951 Grzegorczyk published a result [18] on undecidability of some topological theories. Quite recently this result was quoted in two papers: on geoinformatics [9] and on Knowledge Representation and its qualitative spatio-temporal representation and reasoning [47].

3.9. Statistics

It is difficult to estimate how often the papers of Grzegorczyk were cited in computer science papers. We asked Google Scholar for the query "Grzegorczyk hierarchy". More than 600 answers was found. Even if one deletes some less relevant positions, still more than 400 citations, extensions, presentations and applications of the Grzegorczyk hierarchy remains valid. Beside Scholar one can ask some other computer science bibliographical data base and find some new citations of Grzegorczyk's papers.

Acknowledgment. The author thanks Andrzej Skowron, Paweł Urzyczyn, Damian Niwiński, Grażyna Mirkowska, Mirosław Kutyłowski and Jurek Tomasik for their sugestions and comments. All the errors are caused by the author, who hopes that another full treatment of Grzegorczyk results will appear elsewhere.

References

[1] Axt, P.: Enumeration and the Grzegorczyk hierarchy, *Z. Math. Logik Grundlagen Math*, 1963.

[2] Bellantoni, S. J.: *Ranking Primitive Recursions: the Low Grzegorczyk Classes Revisited*, Technical Report 96-50, DIMACS, November 7, 1996,

[3] Beltyukov, A.: A computer description and a hierarchy of initial Grzegorczyk classes, *J*ournal of Soviet Mathematics, **20**,1982, 2280 - 2289, Translation from Zap. Nauk. Sem. Lening. Otd. Mat. Inst., V. A. Steklova AN SSSR, Vol. 88,1979, pp. 30 – 46, .

[4] Bonner, A.J. and Mecca, G.: Querying sequence databases with transducers, *Acta Informatica*, **36**, 2000, 511-544.

[5] Bournez, O., Hainry, E.: An analog characterization of elementarily computable functions over the real numbers, *ICALP 2004 : international colloquium on automata, languages and programming* (J. Diaz, J. Karhumaki, A. Lepistö, D. Sannella, Eds.), LNCS 3142, 2004.

[6] Brainerd, W., Landweber, L.: *Theory of Computation*, J. Wiley, New York, NY, USA, 1974.

[7] Campagnolo, M., Moore, C., Costa, J.: An analog characterization of the Grzegorczyk hierarchy, *Journal of Complexity*, 2002, 977–1000.

[8] Cichon, E.A., Wainer, S.: The Slow-Growing and the Grzegorczyk Hierarchies, *The Journal of Symbolic Logic*, 1983, 399–408.

[9] Cohn, A. G., Bennett, B., Gooday, J., Gotts, N. M.: Qualitative Spatial Representation and Reasoning with the Region Connection Calculus, *GeoInformatica*, 10, 1997, 275–317.

[10] Cook, S. A.: An overview of computational complexity, *Commun. ACM*, **26**(6), 1983, 400–408.

[11] Cornaros,C.: On Grzegorczyk Induction, *Annals of Pure and Applied Logic*, **74**, 1, 1995, 1–21.

[12] Downey, R., Hirschfeldt, D.: *Algorithmic Randomness and Complexity*, Springer, Berlin, January 2008.

[13] Gabbay, D.M.: On 2nd order intuitionistic propositional calculus with full comprehension, *Annals of Mathematical Logic*, **16**, 1974, 177–186

[14] Gabbay D.M.: Semantical Investigations in Heyting's Intuitionistic Logic, *Synthese Library*,**148**, D. Reidel,1981, Dordrecht.

[15] Gakwaya, J.-S.: Extended Grzegorczyk Hierarchy in the BSS Model of Computability, in: *Foundations of Computational Mathematics* (Cucker, Shub, Eds.), Springer, Berlin, 1997, 127–151.

[16] Görnemann, S.: A logic stronger than intuitionism, Journal of Symbolic Logic, **36**, 2, 1971, 249–261.

[17] Grozea, C.: NP predicates computable in the weakest level of the Grzegorczyck hierarchy, *Journal of Automata, Languages and Combinatorics*, **9**(2-3), September 2004, 269 – 279.

[18] Grzegorczyk, A.: Undecidability of some topological theories, *Fundamenta Mathematicae*, **38**, 1951, 137–152.

[19] Grzegorczyk, A.: Some classes of recursive functions, *Rozprawy Matematyczne*, **4**, 1953, 3–45. `url:http://matwbn.icm.edu.pl/ksiazki/rm/rm04/rm0401.pdf`

[20] Grzegorczyk, A.: On the definitions of computable real continuous functions, *Fundamenta Mathematicae*, **XLIV**, 1957, 61–71

[21] Grzegorczyk, A.: Recursive objects in all finite types, *Fundamenta Mathematicae*, **LIV**, 1964, 73–93

[22] Grzegorczyk, A.: A philosophically plausible formal iterpretation of intuitionistic logic, *Indagationes Mathematicae*, **26**, 1964, 596–601

[23] Grzegorczyk, A.: Undecidability without Arithmetization, *Studia Logica*, **79**(2), 2005, 163–230.

[24] Hartmanis, J., Hopcroft, J. E.: An Overview of the Theory of Computational Complexity, *J. ACM*, **18**(3), 1971, 444–475.

[25] Krajewski, S., Woleński, J.: Andrzej Grzegorczyk: Logic and Philosophy, *Fundamenta Informaticae*, **this volume**, 2007.

[26] Kristiansen, L., Barra, M.: The Small Grzegorczyk Classes and the Typed lambda-Calculus, *CiE* (S. B. Cooper, B. Löwe, L. Torenvliet, Eds.), LNCS 3526, Springer, 2005, 252–262.

[27] Kutyłowski, M.: A generalized Grzegorczyk hierarchy and low complexity classes, *Information and Computation*, **72**(2), 1987, 133–149.

[28] Kutyłowski, M.: Small Grzegorczyk Classes, *Journal of the London Mathematical Society*,**2**(2), 1987, 193.

[29] Kutyłowski, M. and Loryś, K.: A note about $E^0_* = E^2_*$? Problem, *Zt. math. Logik und Grundlag. Math.*, **33**, 1987, 115-121,

[30] Maksimova, L.: On modal Grzegorczyk logic, *Fundamenta Informaticae*, **this volume**, 2007.

[31] Mehlhorn, K.: Polynomial and abstract subrecursive classes, *Proceedings of the sixth annual ACM symposium on Theory of computing*, Annual ACM Symposium on Theory of Computing, ACM Press, Seattle, WA, 1974, see also J. Comp.Syst Sci. **12**, 1976, 147–178

[32] Meyer, A. R.: Depth of nesting and the Grzegorczyk hierarchy, *Notices Amer. Math. Soc.*, 1965, 622–656.

[33] Meyer, A. R., Ritchie, D. M.: The complexity of loop programs, *Proceedings of the 1967 22nd national ACM conference*, ACM Press, New York, NY, USA, 1967, 465–469.

[34] Muchnick, S.: The vectorized Grzegorczyk hierarchy, *Z. Math. Logik Grundlag. Math*,**22**, 1976, 441–480.

[35] Mycka, J., Costa, J.: Undecidability over Continuous Time, *Logic Journal of IGPL*, **14**, 5, 2006, 469.

[36] Ornaghi, M., Benini, M., Ferrari, M., Fiorentini, C., Momigliano, A.: A Constructive Object Oriented Modeling Language for Information Systems, *Electr. Notes Theor. Comput. Sci.*, **153**(1), 2006, 55–75.

[37] Ritchie, R.W.: Classes of predictably computable functions, *Trans. Amer. Math. Soc.*, **106**, 1963, 139–179.

[38] Rogers Jr, H.: *Theory of recursive functions and effective computability*, MIT Press Cambridge, MA, USA, 1987.

[39] Rose, H.: *Subrecursion, Functions and Hierarchies*, School of mathematics, University of Bristol edition, Clarendon Press, Oxford, 1984.

[40] Rauszer,C. and Sabalski,B.: Notes on Rasiowa-Sikorski Lemma, *Studia Logica*, **XXXIV**, 3, 1975, 265–268.

[41] Schwichtenberg, H.: *Classifying recursive functions*, Handbook of Recursion Theory, North Holland Elsevier, Amsterdam, 1997.

[42] Shelah, S.: Primitive Recursive Bounds for Van Der Waerden Numbers, *Journal of the American Mathematical Society*, **1**(3), 1988, 683–697.

[43] Švejdar, V.: An Interpretation of Robinson Arithmetic in its Grzegorczyk's Weaker Variant, *Fundamenta Informaticae*, **this volume**.

[44] Wagner, K., Wechsung, G.: *Computational Complexity*, Springer, 2001.

[45] Wainer, S.: Ordinal Recursion, and a Refinement of the Extended Grzegorczyk Hierarchy, *The Journal of Symbolic Logic*, **37**(2), June 1972, 281–292.

[46] Weiermann, A.: *Rewriting theory for the Hydra battle and the extended Grzegorczyk hierarchy*, Technical report, 1995.

[47] Wolter, F., Zakharyaschev, M.: Qualitative spatio-temporal representation and reasoning: a computational perspective, in: *Exploring Artificial Intelligence in the New Millenium* (G. Lakemeyer, B. Nebel, Eds.), Morgan Kaufmann, 2002.