The complexity of real recursive functions[*]

Manuel Lameiras Campagnolo

D.M./I.S.A., Universidade Técnica de Lisboa,
Tapada da Ajuda, 1349-017, Lisboa, Portugal;
C.L.C./D.M./I.S.T., Universidade Técnica de Lisboa,
Av. Rovisco Pais, 1049-001, Lisboa, Portugal
mlc@math.isa.utl.pt

Abstract. Recursion theory on the reals, the analog counterpart of recursive function theory, is an approach to continuous-time computation inspired in the models of Classical Physics. In recursion theory on the reals, the discrete operations of standard recursion theory are replaced by operations on continuous functions, such as composition and various forms of differential equations as indefinite integrals, linear differential equations, and more general Cauchy problems. We define classes of real recursive functions, in a manner similar to the standard recursion theory, and we study their complexity. We consider, namely, the structural and the computational complexity of those classes. As a result, we prove both upper and lower bounds for several classes of real recursive functions, which lie inside the primitive recursive functions and, therefore, can be characterized in terms of standard computational complexity.

Key words: Continuous-time computation, differential equations, recursion theory, computational complexity.

1 Introduction

Recursive function theory provides the standard notion of computable function [Cut80,Odi89]. Moreover, many time and space complexity classes have recursive characterizations [Clo99]. As far as we know, Moore [Moo96] was the first to extend recursion theory to real valued functions. We will explore this and show that all main concepts in recursion theory like basic functions, operators, function algebras, or functionals, are indeed extendable in a natural way to real valued functions. In this paper, we define recursive classes of real valued functions analogously to the classical approach in recursion theory and we study the complexity of those classes. In recursion theory over the reals, the operations typically include composition of functions, and solutions of several forms of differential equations. On one hand, we look at the structural properties of various algebras of real functions, i.e., we explore intrinsic properties of classes of real recursive functions such as closure under iteration, bounded sums or bounded products. We investigate links between analytical and computational properties of real recursive functions. For instance, we show that a departure from analyticity to C^∞ gives closure under iteration, a fundamental property of discrete functions. On the other hand, we use standard computational complexity theory to establish upper and lower bounds on those algebras. We establish connections between subclasses of real recursive functions, which range from the functions computable in linear space to the primitive recursive functions, and subclasses of the recursive functions closed under various forms of integration. We consider, in particular, indefinite integrals, linear differential equations, and more general Cauchy problems. Finally, we describe some directions of work that suggest that the theory of real recursive functions might be fruitful in addressing open problems in computational complexity.

[*] A similar version of this paper was presented at UMC'02 [Cam02]. This paper describes join work with Félix Costa (I.S.T./Universidade Técnica de Lisboa) and Cris Moore (Santa Fe Institute and University of New Mexico).

2 Recursive functions over $\mathbb{R}$

Moore [Moo96] proposed a theory of recursive functions on the reals, which is defined in analogy with classical recursion theory. A function algebra

$$[B_1, B_2, ...; O_1, O_2, ...],$$

which we also call a computational class, is the smallest set containing basic functions $\{B_1, B_2, ...\}$ and closed under certain operations $\{O_1, O_2, ...\}$, which take one or more functions in the class and create new ones.

Although function algebras have been defined in the context of recursion theory on the integers, they are equally suitable to define classes of real valued recursive functions. As a matter of fact, if the basic functions in a function algebra are real functions and the operators map real functions into real functions, then the function algebra is a set of real functions. Furthermore, if the basic functions have a certain property (e.g. continuity or differentiability) which is preserved by the operators, then every function in the class will have that same property on its domain of definition. In recursion theory on the reals we consider operations such as the following.

COMP (Composition). Given functions $f_1, \ldots, f_p$ of arity n and g of arity p, then define h such that $h(\boldsymbol{x}) = g(f_1(\boldsymbol{x}), ..., f_p(\boldsymbol{x}))$.

$\int$ (S-integration). Given functions $f_1, \ldots, f_m$ of arity n, and $g_1, \ldots, g_m$ of arity $n+1+m$, if there is a unique set of functions $h_1, \ldots, h_m$, such that

$$\begin{aligned} \boldsymbol{h}(\boldsymbol{x}, 0) &= \boldsymbol{f}(\boldsymbol{x}), \\ \partial_y \boldsymbol{h}(\boldsymbol{x}, y) &= \boldsymbol{g}(\boldsymbol{x}, y, \boldsymbol{h}(\boldsymbol{x}, y)), \qquad \forall y \in I - S, \end{aligned} \tag{1}$$

on an interval I containing 0, where $S \subset I$ is a countable set of isolated points, and $\boldsymbol{h}$ is continuous for all $y \in I$, then $h = h_1$ is defined.

$\boldsymbol{\mu}$ (Zero-finding). Given f of arity $n+1$, then define h such that

$$h(\boldsymbol{x}) = \mu_y f(\boldsymbol{x}, y) \overset{\text{def}}{=} \begin{cases} y^- = \sup\{y \in \mathbb{R}_0^- : f(\boldsymbol{x}, y) = 0\}, \text{ if } -y^- \leq y^+ \\ y^+ = \inf\{y \in \mathbb{R}_0^+ : f(\boldsymbol{x}, y) = 0\}, \text{ if } -y^- > y^+ \end{cases}$$

whenever it is well-defined.

To match the definition in [Moo96], derivatives of functions can have singularities (we denote the set of singularities by S). The definition above allows the derivative of h to be undefined on the singularities, as long as the solution is unique and continuous on the whole domain. To illustrate the definition of the operator $\int$, let's look at the following example.

Example 1. ($\sqrt{\cdot}$) Suppose that the constant 1 and the function $g(y, z) = 1/2z$ are defined. Then, the solution of

$$\partial_y h = \frac{1}{2h} \quad \text{and} \quad h(0) = 1. \tag{2}$$

is defined on $I = [-1, 1]$. Indeed, the function $h(y) = \sqrt{y+1}$ is the unique solution of Equation (2) on $[-1, 1]$. The set of singularities is $S = \{-1\}$, so $\partial_y h(y)$ doesn't have to be defined on $y = -1$.

Clearly, the operations above are intended as a continuous analog of operators in classical recursion theory, replacing primitive recursion and zero-finding on $\mathbb{N}$ with S-integration and zero-finding on $\mathbb{R}$. Composition is a suitable operation for real valued functions and it is therefore unchanged. Then, the class of real recursive functions is defined in [Moo96] as:

Definition 1. *The real recursive functions are* $[0, 1, U; \text{COMP}, \int, \boldsymbol{\mu}]$, where 0 and 1 are simply constant functions, and U denotes the set of projections $U_i^n(x_1,\ldots,x_n) = x_i$. We also define real recursive constants as:

Definition 2. *A constant a is said to be computable if there is an unary real recursive function f such that $f(0) = a$.*

Then, if a constant a is computable, then one can also define, with composition and zero, a constant unary function g as $g(x) = f(0(x)) = a$, for all x. As we will see below, some irrational constants like e or π are real recursive, and therefore we can define a function whose value is precisely e or π. This is in contrast to the definition of real numbers computable by Turing machines, where an irrational number is said to be computable if there is a sequence of rationals that converge to it effectively.

If μ is not used at all we get M_0, the "primitive real recursive functions", i.e., $[0, 1, -1, U; \text{COMP}, \int]$. These include the differentially algebraic functions, as well as constants such as e and π. However, M_0 also includes functions with discontinuous derivatives like $|x| = \sqrt{x^2}$.

To prevent discontinuous derivatives, and to make our model more physically realistic, we may require that functions defined by integration only be defined on the largest interval containing 0 on which their derivatives are continuous. This corresponds to the physical requirement of bounded energy in an analog device. We define this in a manner similar to S-integration, but with the additional requirement of the continuity of the derivative:

$\bar{\text{I}}$ (SC^1-integration). Given functions $f_1,\ldots,f_m$ of arity n, and $g_1,\ldots,g_m$ of arity $n+1+m$, if there is a unique set of functions $h_1,\ldots,h_m$, such that

$$\begin{aligned} &\boldsymbol{h}(\boldsymbol{x},0) = \boldsymbol{f}(\boldsymbol{x}), \\ &\partial_y \boldsymbol{h}(\boldsymbol{x},y) = \boldsymbol{g}(\boldsymbol{x},y,\boldsymbol{h}(\boldsymbol{x},y)), \qquad \forall y \in I - S, \end{aligned} \tag{3}$$

on an interval I containing 0, where $S \subset I$ is a countable set of isolated points, and $\boldsymbol{h}$ and $\partial_y \boldsymbol{h}$ are both continuous for all $y \in I$, then $h = h_1$ is defined.

Then, restricting $\int$ to $\bar{\text{I}}$, we define the class $[0, 1-1, U; \text{COMP}, \bar{\text{I}}]$. It is clear that all functions in $[0, 1-1, U; \text{COMP}, \bar{\text{I}}]$ are continuously differentiable on their domains. (A question that arises naturally is if they are of class C^∞.) Therefore, $f(y) = \sqrt{y+1}$ mentioned in Example 1 cannot be defined on the interval $[-1, 1]$ in $\overline{\mathcal{D}}$ anymore, since its derivative is not continuous on that interval.

Example 2. (θ_∞) In $\overline{\mathcal{D}}$ we can define a non-analytic function θ_∞ such that $\theta_\infty(t) = \exp(-1/t)$, when $t > 0$, and $\theta_\infty(t) = 0$, when $t \leq 0$. First consider the unique solution of the initial condition problem

$$z' = \frac{1}{(t+1)^2} z \quad \text{and} \quad z(0) = \exp(-1) \tag{4}$$

with a singularity at $t = -1$. This is $z(t) = 0$ if $t \leq -1$, and $z(t) = \exp(-\frac{1}{t+1})$ if $z > -1$. Then $\theta_\infty(t) = z(t-1)$. The function θ_∞ can be though as a C^∞ version of the Heaviside function θ, defined by $\theta(x) = 0$ when $x < 0$ and $\theta(x) = 1$ when $x \geq 0$.

We can restrict the integration operation even more, if we don't allow singularities for the derivatives in the domain of existence of the solution. Formally, we say that a functions is defined by proper integration if it is defined with the following operator:

I (Proper integration). Given functions $f_1,\ldots,f_m$ of arity n, and $g_1,\ldots,g_m$ of arity $n+1+m$, if there is a unique set of continuous functions $h_1,\ldots,h_m$, such that

$$\begin{aligned} &\boldsymbol{h}(\boldsymbol{x},0) = \boldsymbol{f}(\boldsymbol{x}), \\ &\partial_y \boldsymbol{h}(\boldsymbol{x},y) = \boldsymbol{g}(\boldsymbol{x},y,\boldsymbol{h}(\boldsymbol{x},y)), \qquad \forall y \in I, \end{aligned} \tag{5}$$

on an interval I containing 0, then $h = h_1$ is defined.

This proper form of integration preserves analyticity [Arn96]. Moreover, if the functions $f_1, \ldots, f_m$ and $g_1, \ldots, g_m$ are of class C^k, then h is also of class C^k on its domain of existence (cf. [Har82, 5.4.1]). Since constants and projections are analytic and composition and integration preserve analyticity, then:

Proposition 1. *All functions in* $[0, 1, -1, U; \mathrm{COMP}, \mathrm{I}]$ *are analytic on their domains.*

Similarly, one proves that functions of one variable in $[0, 1, -1, U; \mathrm{COMP}, \mathrm{I}]$ are precisely the differentially algebraic functions [Moo96,GC]. This means that the Gamma function, for instance, is not in the class $[0, 1, -1, U; \mathrm{COMP}, \mathrm{I}]$. Next, we give some examples of functions that do belong to that class.

Proposition 2. *The functions* $+$, $-$, $\times$, $\exp$, $\exp^{[m]}$, *defined as* $\exp^{[0]}(x) = 1$ *and* $\exp^{[n+1]}(x) = \exp(\exp^{[n]}(x))$ *for any integer* n, $\sin$, $\cos$, $1/x$, $\log$, *and* $\arctan$ *belong to* $[0, 1, -1, U; \mathrm{COMP}, \mathrm{I}]$.

To further explore the theory of real recursive functions, we restrict the integration operator to solving time-varying linear differential equations, i.e.,

LI *Linear integration.* Given $f_1, \ldots, f_m$ of arity n and $g_{11}, \ldots, g_{mm}$ of arity $n+1$, then define the function $h = h_1$ of arity $n+1$, where $\boldsymbol{h} = (h_1, \ldots, h_m)$ satisfies the equations $\boldsymbol{h}(\boldsymbol{x}, 0) = \boldsymbol{f}(\boldsymbol{x})$ and $\partial_y \boldsymbol{h}(\boldsymbol{x}, y) = \boldsymbol{g}(\boldsymbol{x}, y)\boldsymbol{h}(\boldsymbol{x}, y)$.

As in classical recursion theory, we define new classes by restricting some operations but adding to the class certain basic functions which are needed for technical reasons. A typical example is the integer function called cut-off subtraction, defined by $x \dot{-} y = x - y$ if $x \geq y$, and $x \dot{-} y = 0$ otherwise. In some real recursive classes we include, instead, a basic function we denote by θ_k and is defined by $\theta_k(x) = 0$ if $x \leq 0$, and $\theta_k(x) = x^k$ if $x > 0$. Clearly, θ_0 is an extension to the reals of the Heaviside function, and $\theta_1(x)$ is an extension to the reals of $x \dot{-} 0$. In general, θ_k is of class C^{k-1}.

For example, we explore the class $[0, 1, -1, \pi, \theta_k, U; \mathrm{COMP}, \mathrm{LI}]$ for some fixed k. Since, unlike solving more general differential equations, linear integration can only produce total functions, then:

Proposition 3. *For any integer* $k > 0$, *if* $f \in [0, 1, -1, \pi, \theta_k, U; \mathrm{COMP}, \mathrm{LI}]$, *then* f *is defined everywhere and belongs to class* C^{k-1}.

We will also consider an even more restricted form of integration, which is just the indefinite integral. Formally, this is defined by:

INT *Indefinite integral.* Given $f_1, \ldots, f_m$ of arity n and $g_1, \ldots, g_m$ of arity $n+1$, then define the function $h = h_1$ of arity $n+1$, where $\boldsymbol{h} = (h_1, \ldots, h_m)$ satisfies the equations $\boldsymbol{h}(\boldsymbol{x}, 0) = \boldsymbol{f}(\boldsymbol{x})$ and $\partial_y \boldsymbol{h}(\boldsymbol{x}, y) = \boldsymbol{g}(\boldsymbol{x}, y)$.

3 Structural complexity

In this section, we ask questions about *intrinsic* properties of classes of real recursive functions such as closure under certain operations. We will see that some intriguing connections exist among closure properties and analytical properties of the classes we consider.

Closure under iteration is a basic operation in recursion theory. If a function f is computable, so is $F(x, t) = f^{[t]}(x)$, the t'th iterate of f on x. We ask whether these analog classes are closed under iteration, in the sense that if f is in the class, then so is some $F(x, t)$ that equals $f^{[t]}(x)$ when t is restricted to the natural numbers.[1]

Proposition 4. $[0, 1, -1, U; \mathrm{COMP}, \overline{\mathrm{I}}]$ *is closed under iteration.*

[1] In [CMC00] we answer this question for Shannon's General Purpose Analog Computer. For connections between real recursion theory and Shannon's model see [GC].

Proof. (Sketch) Let's denote $[0, 1, -1, U; \mathrm{COMP}, \bar{\mathrm{I}}]$ bu $\overline{\mathcal{D}}$. Given f, we can define in $\overline{\mathcal{D}}$ the differential equation

$$\begin{aligned}\big(\theta_\infty(\cos\pi t) + \theta_\infty(-\cos\pi t)\big)\,\partial_t y_1 &= -(y_1 - f(y_2))\,\theta_\infty(\sin 2\pi t)\\ \big(\theta_\infty(\sin\pi t) + \theta_\infty(-\sin\pi t)\big)\,\partial_t y_2 &= -(y_2 - y_1)\,\theta_\infty(-\sin 2\pi t)\end{aligned} \tag{6}$$

with initial condition $y_1(x,0) = y_2(x,0) = x$, where θ_∞ is the function defined in Example 2. We claim that the solution satisfies $y_1(x,t) = f^{[t]}(x)$, for all integer $t \geq 0$. On the interval $[0, \frac{1}{2}]$, $y_2'(x,t) = 0$ because $\theta_\infty(-\sin 2\pi t) = 0$. Therefore, y_2 remains constant with value x, and $f(y_2) = f(x)$. The solution for y_1 on $[0, \frac{1}{2}]$ is then given by

$$\exp\big(\tfrac{1}{\sin(2\pi t)} - \tfrac{1}{\cos(\pi t)}\big)\, y_1' = -(y_1 - f(x)),$$

which we rewrite as $\epsilon\, y_1' = -(y_1 - f(x))$. Note that $\epsilon \to 0^+$ when $t \to 1/2$. Integrating the equation above we obtain

$$y_1 - f(x) = \exp(-\frac{1}{\epsilon}t),$$

where the right hand side goes to 0 when t approaches $1/2$. Therefore, $y_1(x, 1/2) = f(x)$. A similar argument for y_2 on $[\frac{1}{2}, 1]$ shows that $y_2(x,1) = y_1(x,1) = f(x)$, and so on for y_1 and y_2 on subsequent intervals. The set of singularities of Equation (6) is $\{n/2, n \in \mathbb{N}\}$. □

However, if we replace $\mathrm{S}C^1$-integration by proper integration, which preserves analyticity, then the resulting class is no longer closed under iteration. More precisely,

Proposition 5. $[0, 1, -1, U; \mathrm{COMP}, \mathrm{I}]$ *is not closed under iteration.*

Proof. (Sketch) We denote $[0, 1, -1, U; \mathrm{COMP}, \mathrm{I}]$ by $\mathcal{D}$. Let's suppose that $\mathcal{D}$ is closed under iteration. Since $\exp \in \mathcal{D}$, then there is a function F in $\mathcal{D}$ such that $F(x,t) = \exp^{[t]}(x)$ for all $t \in \mathbb{N}$ and all $x \in \mathbb{R}$. Therefore, F has a finite description in $\mathcal{D}$ with a certain fixed number of uses of the I operation. However, it is known that functions of one variable in $\mathcal{D}$ are differentially algebraic [Moo96], that is, they satisfy a polynomial differential equation of finite order. So, for any fixed t, F is differentially algebraic in x. But, from a result of Babakhanian [Bab73], we know that $\exp^{[t]}$ satisfies no non-trivial polynomial differential equation of order less than t. This means that the number of integrations that are necessary to define $\exp^{[t]}$ has to grow with t, which creates a contradiction. □

Since $[0, 1, -1, U; \mathrm{COMP}, \bar{\mathrm{I}}]$ contains non-analytic functions while all functions in $[0, 1, -1, U; \mathrm{COMP}, \mathrm{I}]$ are analytic, one could ask if there is a connexion between those two structural properties of real recursive classes. We believe that closure under iteration and analyticity are related in the following sense:

Conjecture 1. Any non trivial real recursive class which is closed under iteration must contain non-analytic functions.

As a matter of fact, even if it is known that the transition function of a Turing machine can be encapsulated in an analytic function [KM99,Moo98], no analytic form of an iteration function is known.

Next we consider restricted operations as bounded sums and bounded products and we ask which real recursive classes are closed under those operations. We say that an analog class is closed under bounded sums (resp. products) if given any f in the class, there is some g also in the class that equals $\sum_{n<t} f(x,n)$ (resp. $\prod_{n<t} f(x,n)$) when t is restricted to the natural numbers.

Let's see how to define bounded sums in a real recursive class. Not surprisingly, we find that this is related to indefinite integrals. We first define a step function F which matches f on the integers, and whose values are constant on the interval $[j, j+1/2]$ for integer j. F can be defined as $F(t) = f(s(t))$, where s is a continuous step function that matches the identity over the integers. This can be defined with the indefinite integral $s(0) = 0$ and $s'(x) = c_k\theta_k(-\sin 2\pi x)$.[2] Then $s(t) = j$, and $F(t) = f(s(t)) = f(j)$, whenever $t \in [j, j+1/2]$ for integer j. The bounded sum of f is then given by g, such that $g(0) = 0$ and $g'(t) = c_k F(t)\,\theta_k(\sin 2\pi t)$. Then $g(t) = \sum_{z<n} f(z)$ whenever $t \in [n-1, n-1/2]$. So, we can define bounded sums with the constant π, a periodic function like sin, θ_k, and the operation of indefinite integrals. More precisely,

[2] The constant c_k is a rational or a rational multiplied by π.

Proposition 6. *For all* $k \in \mathbb{N}$*,* $[0, 1, -1, \pi, \theta_k, \sin, U; \mathrm{COMP}, \mathrm{INT}]$ *is closed under bounded sums. Moreover, any real recursive class which is closed under composition and indefinite integrals and contains the functions* $0, 1, -1, \pi, \theta_k, \sin, U$ *is closed under bounded sums.*

If a class is closed under bounded products and it contains, for instance, the identity function, then it has to contain functions that grow faster than polynomials. For instance, the class $[0, 1, -1, \pi, \theta_k, \sin, U; \mathrm{COMP}, \mathrm{INT}]$ cannot be closed under bounded products. What can we say if the analog class is closed under linear integration, instead of just indefinite integrals? We conjecture that the answer is still negative, since we believe that the simulation of bounded products would have to rely on a technique similar to Proposition 4 using synchronized clock functions, although we have no proof of this.

Let's then consider the following weaker property. We say that a class is closed under bounded products in a *weak sense* if, given any f in the class which has integer values for integer arguments (i.e., f is an extension to the reals of some $\tilde{f} : \mathbb{R} \times \mathbb{N} \to \mathbb{N}$), there is a g in the class such that $g(x, t) = \prod_{n<t} f(x, n)$ when t is restricted to the natural numbers. Then, in the presence of some appropriate non-analytic function like θ_k, proper integration and even linear integration are sufficient to simulate bounded products. In particular,

Proposition 7. *For all* $k \in \mathbb{N}$*,* $[0, 1, -1, \pi, \theta_k, U; \mathrm{COMP}, \mathrm{LI}]$ *is closed under bounded products in a weak sense.*

Proof. (Sketch) Let f be a function on $\mathbb{N}$ and g be a the function on $\mathbb{N}$ defined from f by bounded product. We show that if f has an extension to the reals in $[0, 1, -1, \pi, \theta_k, U; \mathrm{COMP}, \mathrm{LI}]$ then g does also. First, set $g_n = \prod_{j<n} f_j$. We can approximate the iteration $g_{j+1} = g_j f_j$ using synchronized clock functions as in proof of Proposition 4. However, since we only allow linear integration, the simulated functions cannot coincide exactly with the bounded product. Nevertheless, we can define a sufficiently close approximation because f and g have bounded growth (we can show that any function in $[0, 1, -1, \pi, \theta_k, U; \mathrm{COMP}, \mathrm{LI}]$ is bounded by the iterated exponential $\exp^{[m]}$ for some m).

Let's define a two-component function $\boldsymbol{y}(\tau, t)$ where $y_1(\tau, 0) = y_2(\tau, 0) = 1$,

$$\begin{aligned} \partial_t y_1 &= (y_2 F(t) - y_1)\, c_k \theta_k(\sin 2\pi t)\, \beta(\tau) \\ \partial_t y_2 &= (y_1 - y_2)\, c_k \theta_k(-\sin 2\pi t)\, \beta(\tau) \end{aligned} \tag{7}$$

$\beta(\tau)$ is an increasing function of τ, F is defined as $f \circ s$ as in the proof of Proposition 6. We can show that if β grows fast enough (roughly as fast as $\exp^{[m]}$), then by setting $\tau = n$ we can make the approximation error $|y_1(n, n) - g_n|$ as small as we like. Since g has integer values, the accumulated error on $[0, n]$ resulting from this approximation can be removed with a suitable continuous step function that matches the identity over integers. Note that the Equation (7) is linear in y_1 and y_2.

We illustrate this construction in Figure 1. We approximate the bounded product of the identity function, i.e. the factorial $(n-1)! = \prod_{j<n} j$. We numerically integrated Equation (7) using a standard package.

We can also show that a class is closed under the iteration of extensions to the reals of integer valued functions, as long as it is closed under proper integration, and it contains the non-analytic function θ_k or θ_∞. We call this property closure under iteration in a weak sense. For instance, it can be shown, using a technique similar to [Bra95], that

Proposition 8. $[0, 1, -1, \theta_\infty, U; \mathrm{COMP}, \mathrm{I}]$ *is closed under iteration in a weak sense.*

4 Computational complexity

In this section we explore connections among real recursive classes and standard recursive classes. Since we are interested in classes below the primitive recursive functions, we can characterize them in terms of standard space or time complexity, and consider the Turing machine as the underlying computational model. This approach differs from others, namely BSS-machines [BSS89] or information-based complexity

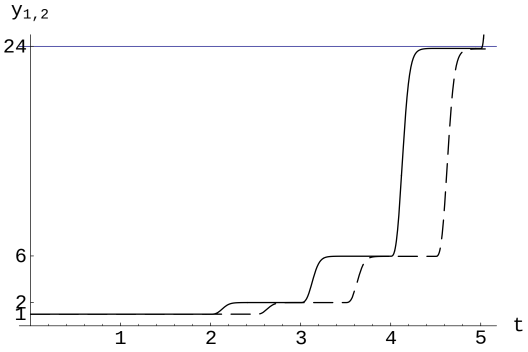

Fig. 1. A numerical integration of Equation (7), where f is a $\mathcal{L}$ function such that $f(0) = 1$ and $f(x) = x$ for $x \geq 1$. Here, $k = 2$. We obtain an approximation of an extension to the reals of the factorial function. In this example, where we chose a small $\tau < 4$, the approximation is just sufficient to remove the error with ϕ and obtain exactly $\prod_{n<5} n = 4! = \phi(y_1(5))$.

[TW98], since it focus on *effective* computability and complexity. There are two main reasons to this. First, the Turing machine model allows us to represent the concept of Cauchy sequences and, therefore, supports a very natural theory of computable analysis. Second, we aim to use the theory of real recursive functions to address problems in standard computational complexity. This would be difficult to achieve with an intrinsically analog theory like the BSS-machines over $\mathbb{R}$.

To compare the computational complexity of real recursive classes and standard recursive classes we have to set some conventions. On one hand, we follow a straightforward approach to associate a class of integer functions to a real recursive class. We simply consider the *discretization* of a real recursive class, i.e., the subset of functions with integer values for integer arguments. More precisely,

Definition 3. *Given a real recursive class $\mathcal{C}$, $\mathcal{F}_{\mathbb{N}}(\mathcal{C}) = \{f : \mathbb{N}^n \to \mathbb{N}$ s.t. f has an extension to the reals in $\mathcal{C}\}$.*

If $\mathcal{F}_{\mathbb{N}}(\mathcal{C})$ contains a certain complexity class $\mathcal{C}'$, this means that $\mathcal{C}$ has at least the computational power of $\mathcal{C}'$, i.e., we can consider $\mathcal{C}'$ as a lower bound for $\mathcal{C}$.

On the other hand, we consider the computational complexity of real functions. We use the notion of [Ko91], which is equivalent to the one proposed by Grzegorczyk [Grz55], and whose underlying computational model is the function-oracle Turing machine. Intuitively, the time (resp. space) complexity of f is the number of moves (resp. the amount of tape) required by a function-oracle Turing machine to approximate the value of $f(x)$ within an error bound 2^{-n}, as a function of the input x and the precision of the approximation n.

Let's briefly recall what a function-oracle Turing machine is (we give an informal description: details can be found in [HU79,Ko91]). For any x in the domain of f, the oracle is a computable sequence ϕ such that for all $n \in \mathbb{N}$, $|\phi(n) - x| < 2^{-n}$. The machine is a Turing machine equiped with an additional query tape, and two additional states. When the machine enters in the query state, it replaces the current string s in the query tape by the string $\phi(s)$, where ϕ is the oracle, moves the head to the first cell of the query tape, and switches to the answer state. This is done in one step of the computation. We say that the time (resp. space) complexity of f on its domain is bounded by a function b if there is a function-oracle Turing machine which, for any x in the domain of f and an oracle ϕ that converges to x, computes an approximation of $f(x)$ with precision 2^{-n} in a number of steps (resp. amount of tape) bounded by $b(x, n)$. Then, for space complexity we define:

Definition 4. *Given a set of functions S, $\mathcal{F}_{\mathbb{R}}\mathrm{SPACE}(S) = \{f : \mathbb{R}^n \to \mathbb{R}$ s.t. the space complexity of f is bounded by some function in $S\}$.*

Therefore, if a real recursive class $\mathcal{C}$ is contained in $\mathcal{F}_{\mathbb{R}}\mathrm{SPACE}(S)$, then S can be considered a space complexity upper bound for $\mathcal{C}$.

Suppose that a function f can be successively approximated within an error 2^{-n} in a certain amount of space. Then, if f is integer, it just has to be approximated to an error less than $1/2$ to know its value exactly. Therefore, if a real recursive class $\mathcal{C}$ is computable in space bounded in $\mathcal{S}$, then the discretization of $\mathcal{C}$ is also computable in space bounded in S. Formally,

Proposition 9. *Let $\mathcal{C}$ be a real recursive class. If $\mathcal{C} \subset \mathcal{F}_{\mathbb{R}}\mathrm{SPACE}(S)$, then $\mathcal{F}_{\mathbb{N}}(\mathcal{C}) \subset \mathcal{F}\mathrm{SPACE}(S)$.*

Given the two conventions established in Definition 3 and Definition 4, we will show upper and lower bounds on some real recursive classes. We can use the closure properties described in the last section to compare discretizations of real recursive classes with standard recursive classes. For instance, since $[0, 1, -1, U; \mathrm{COMP}, \bar{\mathrm{I}}]$ contains extensions of the zero function, successor, projections, and the cut-off function, and is closed under and composition and iteration, then we have the following upper bound for the primitive recursive functions (see [CMC00]):

Proposition 10. $\mathcal{PR} \subset \mathcal{F}_{\mathbb{N}}([0, 1, -1, U; \mathrm{COMP}, \bar{\mathrm{I}}])$.

Note that the same inductive proof works for $[0, 1, -1, \theta_k, U; \mathrm{COMP}, \mathrm{I}]$. Therefore, $\mathcal{PR} \subset \mathcal{F}_{\mathbb{N}}([0, 1, -1, \theta_k, U; \mathrm{COMP}, \mathrm{I}])$.

The elementary functions $\mathcal{E}$, which are closed under bounded sums and products, are a well-known class in recursion theory. All elementary functions are computable in elementary time or space, i.e., in time or space bounded by a tower of exponentials. As a matter of fact, the elementary functions are the smallest known class closed under time or space complexity [Odi00]. From Propositions 6 and 7 it follows that

Proposition 11. *For all $k \geq 0$, $\mathcal{E} \subset \mathcal{F}_{\mathbb{N}}([0, 1, -1, \theta_k, U; \mathrm{COMP}, \mathrm{LI}])$.*

In addition, all functions in $[0, 1, -1, \theta_k, U; \mathrm{COMP}, \mathrm{LI}]$ are computable in elementary space (or time) [CMC02]. Formally,

Proposition 12. *For all $k > 1$, $[0, 1, -1, \theta_k, U; \mathrm{COMP}, \mathrm{LI}] \subset \mathcal{F}_{\mathbb{R}}\mathrm{SPACE}(\mathcal{E})$.*

Combining this with Proposition 9 and Proposition 11, we conclude that:

Proposition 13. *For all $k > 1$, $\mathcal{E} = \mathcal{F}_{\mathbb{N}}([0, 1, -1, \theta_k, U; \mathrm{COMP}, \mathrm{LI}])$.*

which gives an analog characterization of the elementary functions. It is interesting that linear integration alone gives extensions to the reals of all elementary functions, since these are all the functions that can be computed by any practically conceivable digital device. Notice that the above results on $\mathcal{E}$ can be generalized to the levels $\mathcal{E} = \mathcal{E}^3$, $\mathcal{E}^4$, ... of the Grzegorczyk hierarchy if we include in our model a non linear differential operator that generates total functions [CMC02].

In recursion theory, several forms of bounded recursion are widely used, namely to obtain characterization of low complexity classes [Clo99]. In bounded recursion, an *a priori* bound is imposed on the function to be defined with the recursion scheme. Similarly, we can consider the following operator on real functions:

BI (Bounded integration). Given functions $f_1, \ldots, f_m$ of arity n, $g_1, \ldots, g_m$ of arity $n + 1 + m$, and b of arity $n + 1$, if $(h_1, \ldots, h_m)$ is the unique function that satisfies the equations $\boldsymbol{h}(\boldsymbol{x}, y) = \boldsymbol{f}(\boldsymbol{x})$, $\partial_y \boldsymbol{h}(\boldsymbol{x}, y) = \boldsymbol{g}(\boldsymbol{x}, y, \boldsymbol{h}(\boldsymbol{x}, y))$, and $||\boldsymbol{h}(\boldsymbol{x}, y)|| \leq b(\boldsymbol{x}, y)$ on $\mathbb{R}^{n+1}$, then $h = h_1$ of arity $n + 1$ is defined.

Let's consider the class $[0, 1, -1, \theta_k, \times, U; \mathrm{COMP}, \mathrm{BI}]$.[3] All its functions are defined everywhere since this is true for the basic functions and its operators preserve that property. The *a priori* bound on the integration operation strongly restricts this class. All functions in the class $[0, 1, -1, \theta_k, \times, U; \mathrm{COMP}, \mathrm{BI}]$ and its derivatives (for $k > 1$) are bounded by polynomials. Moreover, all functions computable in linear space have extensions in that class:

[3] Given an appropriate bound, the binary product $h(x, y) = xy$ could be easily defined with bounded integration: $h(x, 0) = 0$, and $\partial_y h(x, y) = U_1^2(x, y) = x$. However, no other basic function grows as fast as the binary product, so this needs to be included explicitly in the class.

Proposition 14. *For all $k \geq 0$, $\mathcal{F}$LINSPACE $\subset \mathcal{F}_{\mathbb{N}}([0, 1, -1, \theta_k, \times, U; \mathrm{COMP}, \mathrm{BI}])$.*

Proof. (Sketch) Let's denote $[0, 1, -1, \theta_k, \times, U; \mathrm{COMP}, \mathrm{BI}]$ by $\mathcal{B}_0$. Ritchie [Rit63] proved that the set of integer functions computable in linear space is the function algebra $[0, S, U, \times; \mathrm{COMP}, \mathrm{BREC}]$, usually denoted by $\mathcal{E}^2$, where BREC is bounded recursion. It is easy to verify that $\mathcal{B}_0$ contains extensions to the reals of zero, successor, projections, and binary product. Since $\mathcal{B}_0$ is closed under composition, we just have to verify that $\mathcal{B}_0$ is closed under bounded recursion in a weak sense. But since all functions in $\mathcal{B}_0$ have polynomials bounds, then this can be done with techniques similar to [Bra95] using bounded integration instead of integration. Details can be found in [Cam01]. □

The Ritchie hierarchy [Rit63] is one of the first attempts to classify recursive functions in terms of computational complexity. The Ritchie classes, which range from $\mathcal{F}$LINSPACE to the elementary functions, are the sets of functions computable in space bounded by a tower of exponentials of fixed height. Next we describe a hierarchy of real recursive classes where the first level is $[0, 1, -1, \theta_k, \times, U; \mathrm{COMP}, \mathrm{BI}]$ (see above), and the n-th level is defined by allowing n nested applications of the linear integration operator. In each level of the hierarchy, indefinite integrals are freely used. As in the Ritchie hierarchy, composition is restricted. In [Rit63], the arguments of each recursive function are of two possible types: free and multiplicative. If f is multiplicative in the argument x, then f grows at most polynomially with x. The restricted form of composition forbids composition on two free arguments. For instance, if $2^x + y$ is free in x and multiplicative in y, then the composition $z = 2^x + y$ with $x(t) = 2^t$, which is free in t, is not allowed while the composition $z = 2^x + y$ with $y(t) = 2^t$ is. We denote this restricted composition by RCOMP and define the following hierarchy of real recursive classes (see [Cam01] for details):

Definition 5. *(The hierarchy $\mathcal{S}_n$) For all $n \geq 0$, $\mathcal{S}_n = [\mathcal{B}_0; \mathrm{RCOMP}, \mathrm{INT}, n \cdot \mathrm{LI}]$, where $\mathcal{B}_0 = [0, 1, -1, \theta_k, \times, U; \mathrm{COMP}, \mathrm{BI}]$ for any fixed integer $k > 2$, and where the notation $n \cdot$ LI means that the operator LI can be nested up to n times.*

A few remarks are in order. First, all the arguments of a function h defined with linear integration, from any functions f, g of appropriate arities, are free. For instance, we are not allowed to compose the exponential function with itself, since its argument is free. Second, since solutions of linear differential equations $y'(t) = g(t)y(t)$ are always bounded by an exponential in g and t, and at most n nested applications of linear integration are allowed, then all functions in $\mathcal{S}_n$ have bounds of the form $\exp^{[n]}(p(x))$, where p is a polynomial. Even if the composition $\exp(\exp(x))$ is not permitted, towers of exponentials $\exp^{[n]} = \exp \circ \ldots \circ \exp$ can be defined in $\mathcal{S}_n$:

Example 3. ($\exp^{[n]} \circ p \in \mathcal{S}_n$). Let $u_i(\boldsymbol{x}, y) = \exp^{[i]}(p(\boldsymbol{x}, y))$ for $i = 1, \ldots, n$, where p is a polynomial. Then, the functions u_i are defined by the set of linear differential equations

$$\partial_y u_1 = u_1 \cdot \partial_y p \qquad \ldots \qquad \partial_y u_n = u_n \cdot u_{n-1} \cdots u_1 \cdot \partial_y p$$

with appropriate initial conditions. Thus u_n can be defined with up to n nested applications of LI and, therefore, $\exp^{[n]} \circ p \in \mathcal{S}_n$.

Next we relate the $\mathcal{S}_n$ hierarchy to the exponential space hierarchy (details of the proofs can be found in [Cam01]). Consider the following set of bounding functions:

$$2^{[n]} = \{b_k : \mathbb{N} \to \mathbb{N} \text{ s.t. } k > 0, b_k(m) = 2^{[n]}(km) \text{ for all } m\}.$$

On one hand, $\mathcal{S}_n$ has the following upper bound:

Proposition 15. *For all $n \geq 0$, $\mathcal{S}_n \subset \mathcal{F}_{\mathbb{R}}\mathrm{SPACE}(2^{[n+1]})$.*

Proof. (Sketch) All functions in $\mathcal{S}_n$, and its first and second derivatives, are bounded by $2^{[n]} \circ p$, where p is some polynomial. This follows from the fact that all basic functions in $\mathcal{S}_n$ have such property (this is why we restrict k in the Definition 5) and the operators of $\mathcal{S}_n$ preserve it. Then, using numerical techniques we show how to approximate a function defined by composition or bounded integration in $\mathcal{S}_0 = \mathcal{B}_0$. Given the bounds on the functions in $\mathcal{S}_0$ and their first derivative, composition can be computed in a straightforward manner, without increasing the space bounds. The major difficulty has to do with integration. We have to use an exponential amount of space to achieve a sufficiently good approximation. In fact, the standard techniques for numerical integration (Euler's method) require a number of steps which is exponential in the bounds on the derivatives of the functions we want to approximate [Hen62]. Since the bounds for functions in $\mathcal{S}_0$ are polynomial, the required number of steps N in the numerical integration is exponential. Thus all functions in $\mathcal{S}_0$ can be approximated in exponential space. Finally, we follow the same approach for other levels of the $\mathcal{S}_n$ hierarchy, where restricted composition replaces composition, and linear integration replaces bounded integration. □

On the other hand, all functions computable in space bounded by $2^{[n-1]}$ have extensions in $\mathcal{S}_n$. Formally,

Proposition 16. *For all $n \geq 1$, $\mathcal{F}\mathrm{SPACE}(2^{[n-1]}) \subset \mathcal{F}_{\mathbb{N}}(\mathcal{S}_n)$.*

Proof. (Sketch) As in [Rit63], we show that $\mathcal{F}\mathrm{SPACE}(2^{[n-1]})$ has a recursive definition, using restricted composition and a restricted form of bounded recursion. The following step is to define this restricted form of bounded recursion with bounded sums. Let's suppose that $f \in \mathcal{F}\mathrm{SPACE}(2^{[n-1]})$ is defined by bounded recursion. Then, we can encode the finite sequence $\{f(1), \ldots, f(n)\}$ as an integer (using for instance prime factorization), and replace bounded recursion by a bounded quantification over those encodings.[4] We use the fact that bounded quantifiers can be defined with bounded sums and cut-off subtraction. However, the bound on the encoding of the sequence $\{f(1), \ldots, f(n)\}$ is exponential on the bound on f. Therefore, we need an additional level of exponentials to replace bounded recursion by bounded sums. Finally, we know from Proposition 6 that $\mathcal{S}_n$ is closed under bounded sums, and contains cut-off subtraction as well. □

Unfortunately, we were not able to eliminate bounded integration from the definition of $\mathcal{B}_0$, neither were we able to show that $\mathcal{F}\mathrm{SPACE}(2^{[n]})$ is precisely $\mathcal{F}_{\mathbb{N}}(\mathcal{S}_n)$. We believe those issues are related with the open problem:

$$\mathcal{L}^2 \overset{?}{=} \mathcal{E}^2,$$

where $\mathcal{L}^2 = [0, S, U, \dot{-}; \mathrm{COMP}, \mathrm{BSUM}]$ is defined with bounded sums and is known as Skolem's lower elementary functions.[5] We consider instead the following problem:

$$\mathcal{F}_{\mathbb{N}}([0, 1, -1, \theta_k, +, U; \mathrm{COMP}, \mathrm{INT}]) \overset{?}{=} \mathcal{F}_{\mathbb{N}}([0, 1, -1, \theta_k, \times, U; \mathrm{COMP}, \mathrm{BI}]).$$

At first sight, it seems that the equality above is false, since bounded integration is more general than indefinite integrals. However, the problem only concerns the discretizations of the analog classes. One could try to use results of the theory of differential equations to show directly that bounded integration is reducible, up to a certain error, to a finite sequence of indefinite integrals. It is known that solutions of general differential equations, $y'(t) = f(t, y)$ and $y(0) = y_0$, can be uniformly approximated by sequences of integrals, given some broad conditions that guarantee existence and uniqueness [Arn96,Har82]. However, that result, which is based on Picard's successive approximations, requires a sequence of integrals whose length increases with t. Since all functions in $[0, 1, -1, \theta_k, \times, U; \mathrm{COMP}, \mathrm{BI}]$ and its derivatives are polynomially bounded, it might be possible to find a finite approximation for bounded integration, which would be sufficient to approximate functions which range on the integers. Notice that the standard numerical techniques (Euler's method) to approximate the solution of $y'(t) = f(t, y)$ and $y(0) = y_0$ require a number of approximation steps which are exponential in the bounds on the derivative, while Picard's method only needs a polynomially long sequence of indefinite integrals, if the bounds on the derivatives are polynomial.

If the equality above is true, and if $\mathcal{F}_{\mathbb{N}}([0, 1, -1, \theta_k, +, U; \mathrm{COMP}, \mathrm{INT}]) \subset \mathcal{L}^2$, then we would obtain a chain of inclusions that would show that $\mathcal{L}^2 = \mathcal{E}^2$. These remarks above establish a connection between the theory of real recursive functions and computational complexity that would be interesting to explore.

[4] We follow a known technique in recursion theory (see [Ros84]).

[5] Recall that $\mathcal{E}^2 = [0, S, U, \times; \mathrm{COMP}, \mathrm{BREC}]$ and is precisely $\mathcal{F}\mathrm{LINSPACE}$. Notice that $\mathcal{L}^2 \subset \mathcal{E}^2$.

5 Final remarks

We described some results on real recursive functions and we listed some open problems and directions for further research. We believe that recursion theory over the reals is not only an interesting area of research by itself, but it is also related to other areas such as computational complexity, numerical analysis or dynamical systems.

We mentioned possible links to computational complexity in the last section. It would be interesting to look at real recursive classes related to low time complexity classes. For instance, it is unlikely that the class in Proposition 6 is contained in $\mathcal{F}_{\mathbb{R}}\mathrm{TIME}(P)$, where P is the set of polynomials, since if $\mathcal{F}_{\mathbb{R}}\mathrm{TIME}(P)$ is closed under INT, then $\#P = \mathcal{F}\mathrm{PTIME}$ [Ko91]. Therefore, schemes of integration other than the ones we described in this paper have to be explored to find analogues to $\mathcal{F}$PTIME or other low time complexity classes.

We would like to clarify the connections between real recursive functions and dynamical systems. It is known that the unary functions in $[0, 1, -1, U; \mathrm{COMP}, \mathrm{I}]$ are precisely the solutions of equations $\boldsymbol{y}' = p(\boldsymbol{y}, x)$, where p is a polynomial [GC]. We conjecture that $[0, 1, -1, U; \mathrm{COMP}, \mathrm{LI}]$ corresponds to the family of dynamical systems $\boldsymbol{y}' = f(\boldsymbol{y}, x)$, where each f_i is linear and depends at most on $x, y_1, \ldots, y_i$. Given such canonical representations of classes of real recursive functions, one could investigate their dynamical properties.

Acknowledgements. I thank Félix Costa for inviting me to participate in the Advanced School and Workshop on Soft Computing and Complex Systems. I also thank Cristian Calude, Michael Dinneen and Ferdinand Peper for inviting me to UMC'02, where I first presented the contents of this paper, and Amílcar Sernadas, Daniel Graça, Yuzuru Sato for their valuable suggestions. This work was supported by FCT via the Laboratório de Modelos e Arquitecturas Computacionais, grant PRAXIS XXI/BD/18304/98, and more recently the Center for Logic and Computation within the project ConTComp. I also thank the Instituto Superior de Agronomia, the Fundação Luso-Americana para o Desenvolvimento and the Santa Fe Institute for their support.

References

[Arn96] V. I. Arnold. *Equations Différentielles Ordinaires.* Editions Mir, 5 ème edition, 1996.

[Bab73] A. Babakhanian. Exponentials in differentially algebraic extension fields. *Duke Math. J.*, 40:455–458, 1973.

[Bra95] M.S. Branicky. Universal computation and other capabilities of hybrid and continuous dynamical systems. *Theoretical Computer Science*, 138(1):67–100, 1995.

[BSS89] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers: NP-completness, recursive functions and universal machines. *Bull. Amer. Math. Soc.*, 21:1–46, 1989.

[Cam01] M.L. Campagnolo. *Computational complexity of real recursive functions and analog circuits.* PhD thesis, Instituto Superior Técnico, 2001.

[Cam02] M.L. Campagnolo. The complexity of real recursive functions. In C.S. Calude, M.J. Dinneen, and F. Peper, editors, *Unconventional Models of Computation, UMC'02*, number 2509 in LNCS, pages 1–14. Springer-Verlag, 2002.

[Clo99] P. Clote. Computational models and function algebras. In E.R. Griffor, editor, *Handbook of Computability Theory*, pages 589–681. Elsevier, 1999.

[CMC00] M.L. Campagnolo, C. Moore, and J.F. Costa. Iteration, inequalities, and differentiability in analog computers. *Journal of Complexity*, 16(4):642–660, 2000.

[CMC02] M.L. Campagnolo, C. Moore, and J.F. Costa. An analog characterization of the Grzegorczyk hierarchy. *Journal of Complexity*, 4(18):977–1000, 2002.

[Cut80] N. J. Cutland. *Computability: An Introduction to Recursive Function Theory.* Cambridge University Press, 1980.

[GC] D. S. Graça and J. F. Costa. Analog computers and recursive functions over the reals. To appear in the *Journal of Complexity.*

[Grz55] A. Grzegorczyk. Computable functionals. *Fund. Math.*, 42:168–202, 1955.

[Har82] P. Hartman. *Ordinary Differential Equations.* Birkhauser, 2nd edition, 1982.

[Hen62] P. Henrici. *Discrete Variable Methods in Ordinary Differential Equations.* Wiley, New York, 1962.

[HU79] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addisson-Wesley, 1979.

[KM99] P. Koiran and C. Moore. Closed-form analytic maps in one or two dimensions can simulate Turing machines. *Theoretical Computer Science*, 210:217–223, 1999.

[Ko91] K.-I. Ko. *Complexity Theory of Real Functions.* Birkhauser, 1991.

[Moo96] C. Moore. Recursion theory on the reals and continuous-time computation. *Theoretical Computer Science*, 162:23–44, 1996.

[Moo98] C. Moore. Finite-dimensional analog computers: flows, maps, and recurrent neural networks. In C.S. Calude, J. Casti, and M.J. Dinneen, editors, *Unconventional Models of Computation*, DMTCS. Springer-Verlag, 1998.

[Odi89] P. Odifreddi. *Classical Recursion Theory.* Elsevier, 1989.

[Odi00] P. Odifreddi. *Classical Recursion Theory II.* Elsevier, 2000.

[Rit63] R.W. Ritchie. Classes of predictably computable functions. *Transactions Amer. Math. Soc.*, 106:139–173, 1963.

[Ros84] H.E. Rose. *Subrecursion: Functions and Hierarchies.* Clarendon Press, 1984.

[TW98] J. Traub and A.G. Werschulz. *Complexity and Information.* Cambridge University Press, 1998.