The complexity space of partial functions: A connection between Complexity Analysis and Denotational Semantics

Salvador Romaguera[1)], M.P. Schellekens[2)], Oscar Valero[3)*]

[1)]Departamento de Matemática Aplicada, IMPA-UPV, Universidad Politécnica de Valencia, 46071 Valencia, Spain. E-mail: sromague@mat.upv.es

[2)]Center of Efficiency-Oriented Languages, Departament of Computer Science, National University of Ireland, Western Road, Cork, Ireland. E-mail: m.schellekens@cs.ucc.ie

[3)]Departamento de Ciencias Matemáticas e Informática, Universidad de las Islas Baleares, 07122, Palma de Mallorca, Spain. E-mail: o.valero@uib.es

Abstract

The study of dual complexity spaces, introduced by S. Romaguera and M. Schellekens [Topology Appl. 98 (1999), 311-322], constitutes a part of the interdisciplinary research on Computer Science and Topology. The relevance of the theory is given by the fact that it allows one to apply fixed point techniques of Denotational Semantics to Complexity Analysis. Motivated by this fact and with the intention of obtaining a mixed framework valid for both disciplines, a new complexity space has been introduced and studied, formed by partial functions [Int. J. Comput. Math. 85 (2008), 631-640]. In this paper we enter more deeply into the relationship between semantics and complexity analysis of programs. We present an application of the complexity space of partial functions via an alternative formal proof of the asymptotic upper bound for the average case analysis of Quicksort. An extension of the complexity space of partial functions is constructed in order to give a mathematical model for the validation of recursive definitions of programs. As an application of this new approach the correctness of the denotational specification of the factorial function is shown.

*The first and the third authors thank the support of the Spanish Ministry of Education and Science, and FEDER, grant MTM2006-14925-C02-01

Keywords: ordered cone, extended quasi-metric, fixed point, Complexity Analysis, Quicksort, Denotational Semantics, recursive specification.

1 Introduction

The theory of complexity spaces was introduced in [26] as a topological foundation for the complexity analysis of programs and algorithms. The basis for this theory is the notion of "complexity distance", which is a generalized metric that intuitively measures relative progress made in lowering the complexity when a program is replaced by another one. The main aim of the developed topological theory is to obtain a unified structure that allows one to apply the techniques of Denotational Semantics to the analysis of algorithmic complexity. In order to achieve this objective the notion of "complexity domain" was introduced in [27]. This generalized concept consists of an ordered structure, which satisfies the same axioms of an ordered cone except the existence of a neutral element, equipped with a quasi-metric.

Later on, a new complexity structure was introduced and studied, the so-called dual complexity space ([22, 23]). This is a quasi-metric space actually admitting the structure of an ordered cone in the sense of [8]. Furthermore, dual complexity spaces still allow one to carry out the complexity analysis of algorithms and programs. These last two facts motivate the use of dual complexity spaces instead of the original ones.

In the last years the interest in dual complexity spaces has increased and they have been studied in depth ([24], [9], [10], [11], [18], [17], [20], [16], [21]).

On the other hand, in Computer Science it is very usual to define procedures or functions as subprograms that call themself. When a programmer designs a procedure using recursion one must consider whether the mathematical specification for the procedure provides a nonterminating program i.e. the result of a computation fails to terminate or, alternatively, whether the total program takes too much running time to solve the desired problem. Furthermore, when the recursion is used by a programmer to define a function, such a recursive definition can be a semantically meaningless object if its meaning is expressed in terms of the function to define.

The analysis of the amount of running time for this kind of programs and the consistency of recursive definitions of functions is based on the theory of recurrence equations. Thus both the running of computing taken by a recursive algorithm to perform a fixed task, and the semantical meaning of a recursive denotational definition can be seen as a solution of a recurrence equation. Consequently, fixed point theory turns out central to obtain "consistent" specifications for procedures or functions. This is achieved using the *principle of fixpoint induction* ([5]), which provides the mathematical specification (a total mapping defined recursively) as a fixed point that is, at the same time, the limit of a sequence of partial mappings (also defined recursively).

Motivated by the fact that partial functions have proven to be very useful in Denotational Semantics in that they provide a basis for a mathematical model for high-level programming languages, a new (dual) complexity space was constructed in [25] using the notion of a partial function. This new complexity structure is also an ordered cone and supplies a suitable tool for the application of typical Denotational Semantics techniques in the context of Symbolic Computation ([25]).

In this paper we show that the complexity space of partial functions is a useful framework to apply the principle of fixed point induction to the complexity analysis of algorithms and to program verification. The remainder of this paper is organized as follows. Section 2 is devoted to introduce some mathematical preliminaries. A detailed description of complexity spaces, including the complexity space of partial functions, is introduced in Section 3. An application to the complexity analysis of Quicksort, in terms of complexity partial functions, is given in Section 4. In particular we give an alternative proof of the well-known fact that the running time of computing (for the average case) of a Quicksort algorithm has an asymptotic upper bound in the class $\mathcal{O}(n \log_2 n)$. In Section 5 we present an extension of the complexity space of partial functions. Furthermore, we show that this new approach, contrary to the old one, is suitable for the semantic analysis of programs. In fact it is useful to prove mathematically when a function defined recursively is consistent. As an example we give an alternative proof of the well-known fact that the factorial semantic specification is meaningful.

2 Preliminaries

Throughout this paper the letters $\mathbb{R}^+$, $\mathbb{N}$ and ω will denote the set of nonnegative real numbers, the set of natural numbers and the set of nonnegative integer numbers, respectively.

Our main references for quasi-metric spaces are [7] and [15].

Following the modern terminology, a quasi-metric on a set X is a nonnegative real-valued function d on $X \times X$ such that for all $x, y, z \in X$:(i) $d(x,y) = d(y,x) = 0 \Longleftrightarrow x = y$; (ii) $d(x,z) \leq d(x,y) + d(y,z)$.

We will also consider extended quasi-metrics. They satisfy the above axioms, except that we allow $d(x,y) = +\infty$.

An extended quasi-metric space is a pair (X,d) such that X is a (nonempty) set and d is an extended quasi-metric on X.

Each extended quasi-metric d on a set X induces a T_0 topology $\mathcal{T}(d)$ on X which has as a base the family of open d-balls $\{B_d(x,r) : x \in X,\ r > 0\}$, where $B_d(x,r) = \{y \in X : d(x,y) < r\}$ for all $x \in X$ and $r > 0$.

Given an extended quasi-metric d on X, then the function d^s defined on $X \times X$ by $d^s(x,y) = \max\{d(x,y), d(y,x)\}$ is an extended metric on X.

An extended quasi-metric d on a set X is said to be bicomplete if the extended metric d^s is complete on X.

According to [8] a cone on $\mathbb{R}^+$ (a semilinear space in [21]) is a triple $(X,+,\cdot)$ such that $(X,+)$ is an Abelian monoid, and $\cdot$ is a function from $\mathbb{R}^+ \times X$ to X such that for all $x,y \in X$ and $r,s \in \mathbb{R}^+$:(i) $r\cdot(s\cdot x) = (rs)\cdot x$; (ii) $r\cdot(x+y) = (r\cdot x)+(r\cdot y)$; (iii) $(r+s)\cdot x = (r\cdot x)+(s\cdot x)$; (iv) $1\cdot x = x$; (v) $0 \cdot x = 0$.

A cone $(X,+,\cdot)$ is called cancellative if for all $x,y,z \in X$, $x+z = y+z$ implies that $x = y$.

Similarly to [21], an extended quasi-metric d on a cone $(X,+,\cdot)$ is said to be subinvariant (respectively, invariant) if for each $x,y,z \in X$ and $r > 0$, $d(x+z,y+z) \leq d(x,y)$ (respectively, $d(x+z,y+z) = d(x,y)$) and $d(r\cdot x, r\cdot y) = rd(x,y)$, where we assume that $r\cdot(+\infty) = +\infty$ for all $r > 0$.

We briefly introduce a few notions of order theory (see [5] for a fuller treatment).

An order on a nonempty set X is a reflexive, transitive and antisymmetric binary relation $\leq$ on X. An ordered set is a pair $(X,\leq)$ such that $\leq$ is an order on X.

In case of the least element of an ordered set exists, we will say that the ordered set is pointed.

A well-known example of ordered set, which will play a central role in this work is the set of partial functions on ω, which is formally defined by

$$[\omega \to \mathbb{R}^+] = \{f : \mathrm{dom}f \to \mathbb{R}^+ \text{ with } \mathrm{dom}f \neq \emptyset \text{ and } \mathrm{dom}f \subseteq \omega\}.$$

Obviously this set becomes an ordered set when it is ordered by extension $\sqsubseteq$ (for a detailed discussion we refer the reader to [5]), i.e.

$$f \sqsubseteq g \Longleftrightarrow \mathrm{dom}f \subseteq \mathrm{dom}g \text{ and } f(n) = g(n) \text{ for all } n \in \mathrm{dom}f.$$

Let $(X,\leq)$ and $(Y,\preceq)$ be two ordered sets. A mapping $\varphi : X \to Y$ is said to be monotone if $\varphi(x) \preceq \varphi(y)$ whenever $x \leq y$.

Following [8], an ordered cone is a pair $(X,\leq)$ where X is a cone and $\leq$ is an order on X which is compatible with the cone structure, i.e. $x+y \leq v+w$ and $r\cdot x \leq r\cdot y$ whenever $x,y,v,w \in X$ with $x \leq v$, $y \leq w$ and $r \in$

$\mathbb{R}^+$. Ordered cones have proved to be useful in semantics for programming languages (see [29]).

In the sequel if A is a nonempty set, we will denote by $|A|$ its cardinality.

3 The complexity space of partial functions

In 1995, M. Schellekens introduced the theory of complexity (quasi-metric) spaces as a part of the development of a topological foundation for the complexity analysis of programs and algorithms ([26]). The applicability of this theory to the complexity analysis of Divide & Conquer algorithms was illustrated by Schellekens in the same reference. In particular, he gave a new proof, based on fixed point arguments, of the fact that the mergesort algorithm has optimal asymptotic average running time.

Let us recall that the complexity space is the pair $(\mathcal{C}, d_{\mathcal{C}})$, where

$$\mathcal{C} = \{f : \omega \to (0, +\infty] : \sum_{n=0}^{+\infty} 2^{-n} \frac{1}{f(n)} < +\infty\},$$

and $d_{\mathcal{C}}$ is the quasi-metric on $\mathcal{C}$ defined by

$$d_{\mathcal{C}}(f, g) = \sum_{n=0}^{+\infty} 2^{-n} [(\frac{1}{g(n)} - \frac{1}{f(n)}) \vee 0].$$

According to [26], given two functions $f, g \in \mathcal{C}$ the numerical value $d_{\mathcal{C}}(f, g)$ (the complexity distance from f to g) can be interpreted as the relative progress made in lowering the complexity by replacing any program P with complexity function f by any program Q with complexity function g. Therefore, if $f \neq g$, the condition $d_{\mathcal{C}}(f, g) = 0$ can be assumed as f is "more efficient" than g on all inputs.

Later on, S. Romaguera and M. Schellekens ([22, 23]) introduced the so-called dual complexity space and they obtained several quasi-metric properties of the complexity space, which are interesting from a computational point of view, via the analysis of this new complexity (quasi-metric) space. Furthermore, and contrarily to the original space, the dual complexity space can be endowed with a cancellative cone structure equipped with pointwise addition and pointwise scalar multiplication. This fact gives one more motivation for the use of this new approach instead of the original one, because of cones provide a suitable framework for an efficiency analysis of a wide class of algorithms (see [24], [11], [10], [9]).

The dual complexity space is the pair $(\mathcal{C}^*, d_{\mathcal{C}^*})$, where

$$\mathcal{C}^* = \{f : \omega \to \mathbb{R}^+ : \sum_{n=0}^{+\infty} 2^{-n} f(n) < +\infty\},$$

and $d_{\mathcal{C}^*}$ is the quasi-metric on $\mathcal{C}^*$ defined by

$$d_{\mathcal{C}^*}(f, g) = \sum_{n=0}^{+\infty} 2^{-n}[(g(n) - f(n)) \vee 0].$$

It is clear that the computational intuition behind the complexity distances between two functions in $\mathcal{C}$ can be recuperated in the following way: the numerical value $d_{\mathcal{C}^*}(f, g)$, for any $f, g \in \mathcal{C}^*$, can be interpreted as a relative measure of the progress made in lowering the complexity by replacing any program Q with complexity function g by any program P with complexity function f, whenever the complexity measure is assumed as the running time of computing. Hence $d_{\mathcal{C}^*}(f, g) = 0$ provides that g is more "efficient" than f on all inputs. However, as it happens for the distance $d_{\mathcal{C}}$, when $d_{\mathcal{C}^*}(f, g) \neq 0$ we can not establish which complexity function of the two, f or g, is more efficient. In order to avoid this disadvantage, a slight modification in the definition of the complexity distance $d_{\mathcal{C}^*}$ was introduced, and thus, a new complexity (extended quasi-metric) distance e_{C^*} was constructed and studied in [21]. Now, the distance e_{C^*} is a useful tool for the quantitative complexity analysis of algorithms for the specific complexity measure of running time of computing. An application of this new approach to the complexity analysis of Divide and Conquer algorithms, in the spirit of Schellekens, was also given in [21]. Furthermore, this new complexity structure was also applied to modeling certain processes that arise, in a natural way, in Symbolic Computation.

In particular, this new (dual) complexity space consists of the pair $(\mathcal{C}^*, e_{C^*})$ where e_{C^*} is given by

$$e_{C^*}(f, g) = \begin{cases} \sum_{n=0}^{+\infty} 2^{-n}(g(n) - f(n)) & \text{if } f(n) \leq g(n) \text{ for all } n \in \omega, \\ +\infty & \text{otherwise.} \end{cases}$$

The extended quasi-metric e_{C^*} has nice properties as, for instance and among others, invariancy, Hausdorffness and bicompleteness. (for a deeper study see [21]).

Recently, and motivated by the usefulness of partial functions in Denotational Semantics and the relationship between Denotational Semantics and Complexity Analysis (see [26, 27]), Romaguera and Valero have extended the dual complexity space $(\mathcal{C}^*, e_{\mathcal{C}^*})$ to a more general one, the so-called complexity space of partial functions $(\mathcal{C}^*_{\rightharpoondown}, e_{\mathcal{C}^*_{\rightharpoondown}})$ which is introduced in [25] as follows. Let $[(\omega \rightharpoondown \mathbb{R}^+)]$ be the set of partial functions $f \in [\omega \to \mathbb{R}^+]$ such that $\mathrm{dom} f = \{0, 1, ..., n\}$ for some $n \in \omega$, or $\mathrm{dom} f = \omega$, and let $\leq_{\rightharpoondown}$ be the order on $[(\omega \rightharpoondown \mathbb{R}^+)]$ given by

$$f \leq_{\rightharpoondown} g \Leftrightarrow \mathrm{dom} g \subseteq \mathrm{dom} f \text{ and } f(n) \leq g(n) \text{ for all } n \in \mathrm{dom} g.$$

Then we define

$$\mathcal{C}^*_{\rightharpoondown} := \{f \in [(\omega \rightharpoondown \mathbb{R}^+)] : \sum_{n \in \mathrm{dom} f} 2^{-n} f(n) < +\infty\}$$

and

$$e_{\mathcal{C}^*_{\rightharpoondown}}(f, g) = \begin{cases} \sum_{n \in \mathrm{dom} g} 2^{-n}(g(n) - f(n)) & \text{if } f \leq_{\rightharpoondown} g \\ +\infty & \text{otherwise.} \end{cases}.$$

Note that if $f \in \mathcal{C}^*$ then the unordered sum $\sum_{n \in \mathrm{dom} f} 2^{-n} f(n)$ exists and its sum is equals to $\sum_{n=0}^{\infty} 2^{-n} f(n)$ (see, for instance, Problem G (g) in [14]). Therefore $\mathcal{C}^* \subsetneq \mathcal{C}^*_{\rightharpoondown}$.

On the other hand, the set $\mathcal{C}^*_{\rightharpoondown}$ becomes a noncancellative ordered cone (Proposition 2, [25]) endowed with the operations $\oplus$ and $\odot$ defined for all $f, g \in \mathcal{C}^*_{\rightharpoondown}$ as follows:

$$\begin{aligned} (f \oplus g)(n) &= f(n) + g(n) \text{ for all } n \in \mathrm{dom}(f \oplus g) \\ (r \odot f)(n) &= r f(n) \text{ for all } n \in \mathrm{dom}(r \odot f), \end{aligned}$$

where $\mathrm{dom}(f \oplus g) = \mathrm{dom} f \cap \mathrm{dom} g$ and $\mathrm{dom}(r \odot f) = \mathrm{dom} f$.

Of course, if $f, g \in \mathcal{C}^*$ and $r \in \mathbb{R}^+$ then the operations $f \oplus g$ and $r \odot f$ coincide with the pointwise addition and scalar multiplication, respectively.

It was proved in Proposition 3 of [25] that $e_{\mathcal{C}^*_{\rightharpoondown}}$ is a bicomplete subinvariant extended quasi-metric on $\mathcal{C}^*_{\rightharpoondown}$.

The complexity space $(\mathcal{C}^*_{\rightharpoondown}, e_{\mathcal{C}^*_{\rightharpoondown}})$ constitutes, as in case of $(\mathcal{C}^*, e_{\mathcal{C}^*})$, a suitable framework to measure distances between symbolic representations of real numbers and its approximations, as it was showed in [25].

On the other hand, decreasing sequences of complexity functions play a central role in applications of complexity spaces to Computer Science. In fact, such sequences have allowed to discuss the complexity (running time of computing) of sorting program mergesort ([26]) and certain wide class of Probabilistic Divide and Conquer algorithms ([19]). Moreover, several advantages, in measuring real numbers, have been exhibited when sequences of computer numerical representations of real numbers have been identified with decreasing sequences ([21, 20, 25]).

Following [25], we will say that a sequence $(f_k)_{k\in\mathbb{N}}$ in $\mathcal{C}^*_{\rightarrow}$ is decreasing if $f_{k+1} \leq_{\rightarrow} f_k$ for all $k \in \mathbb{N}$. In this case we will denote by $\sqcap_{\downarrow} f_k$ the element of $[(\omega \rightharpoonup \mathbb{R}^+)]$ such that $\mathrm{dom}\sqcap_{\downarrow} f_k = \bigcup_{k\in\mathbb{N}} \mathrm{dom} f_k$ and

$$(\sqcap_{\downarrow} f_k)(n) = \inf_{n\in \mathrm{dom} f_k} f_k(n).$$

The following result will be useful later on.

Proposition 1. *Let $(f_k)_{k\in\mathbb{N}}$ be a decreasing sequence in $\mathcal{C}^*_{\rightarrow}$ such that $\sqcap_{\downarrow} f_k \in \mathcal{C}^*$ and $|\mathrm{dom} f_k|$ is finite for all $k \in \mathbb{N}$. If $\lim_{k\to\infty} e_{\mathcal{C}^*_{\rightarrow}}(\sqcap_{\downarrow} f_k, f_k) = 0$, then $\sqcap_{\downarrow} f_k$ is the unique $e_{\mathcal{C}^*_{\rightarrow}}$-limit point of $(f_k)_{k\in\mathbb{N}}$.*

Proof. Suppose that there is $g \in \mathcal{C}^*_{\rightarrow}$ such that $\lim_{k\to\infty} e_{\mathcal{C}^*_{\rightarrow}}(g, f_k) = 0$. Then, by construction of $e_{\mathcal{C}^*_{\rightarrow}}$ and our hypothesis that $|\mathrm{dom} f_k|$ is finite for all $k \in \mathbb{N}$ and $\sqcap_{\downarrow} f_k \in \mathcal{C}^*$, we deduce that $g \in \mathcal{C}^*$ and for each $k \in \mathbb{N}$, $g(n) \leq f_k(n)$ whenever $n \in \mathrm{dom} f_k$. Hence $g(n) \leq \sqcap_{\downarrow} f_k(n)$ for all $n \in \mathbb{N}$. Assume that there is $n_0 \in \mathbb{N}$ such that $\sqcap_{\downarrow} f_k(n_0) > g(n_0)$. Put $\sqcap_{\downarrow} f_k(n_0) = g(n_0) + \delta$. Since $e_{\mathcal{C}^*_{\rightarrow}}(g, f_k) \to 0$, there is $k \in \mathbb{N}$ such that $f_k(n_0) < g(n_0) + \delta$, so $f_k(n_0) < \sqcap_{\downarrow} f_k(n_0)$, a contradiction. Therefore $g = \sqcap_{\downarrow} f_k$, and thus $\sqcap_{\downarrow} f_k$ is the unique $e_{\mathcal{C}^*_{\rightarrow}}$-limit point of $(f_k)_{k\in\mathbb{N}}$.

The following easy example shows that condition $\lim_{k\to\infty} e_{\mathcal{C}^*_{\rightarrow}}(\sqcap_{\downarrow} f_k, f_k) = 0$, can not be deleted in the above result.

Example 1. Let $(f_k)_{k\in\mathbb{N}}$ be such that $\mathrm{dom} f_k = \{n \in \omega : n \leq k\}$ for all $k \in \mathbb{N}$, and $f_k(n) = 0$ whenever $n < k$ and $f_k(k) = 2^k$. It is clear that $(f_k)_{k\in\mathbb{N}}$ is a decreasing sequence in $\mathcal{C}^*_{\rightarrow}$ for which $\sqcap_{\downarrow} f_k \in \mathcal{C}^*$ (in fact $(\sqcap_{\downarrow} f_k)(n) = 0$ for all $n \in \omega$). However $e_{\mathcal{C}^*_{\rightarrow}}(\sqcap_{\downarrow} f_k, f_k) = 1$ for all $k \in \mathbb{N}$.

The fixed point theory provides an efficient tool in Computer Science. In particular, many applications of such a theory to denotational models of programming languages are obtained by means of order-theoretic notions (see, for instance, [5, 12, 28]). However, several applications of the Banach fixed point theorem to complexity analysis of programs and algorithms and to metric semantics for programming languages have been given in [26, 1, 2, 3, 13]. In this last case such applications are founded only on metric requirements. We end the section by presenting an easy fixed point theorem in the realm of extended quasi-metric spaces which involves also order notions. Its utility will be shown in the next section.

According to [25] (compare [26]), a monotone mapping $\phi : \mathcal{C}^*_{\rightarrow} \rightarrow \mathcal{C}^*_{\rightarrow}$ is called an improver with respect to $f \in \mathcal{C}^*_{\rightarrow}$ if $\phi(f) \leq_{\rightarrow} f$.

Theorem 1. *Let ϕ be a continuous monotone mapping from the complexity space $(\mathcal{C}^*_{\rightarrow}, e_{\mathcal{C}^*_{\rightarrow}})$ into itself. If ϕ is an improver with respect to any $f_0 \in \mathcal{C}^*_{\rightarrow}$ such that $\sqcap_{\downarrow}\phi^k f_0 \in \mathcal{C}^*$ and*

$$\lim_{k\rightarrow\infty} e_{\mathcal{C}^*_{\rightarrow}}(\sqcap_{\downarrow}\phi^k f_0, \phi^k f_0) = 0,$$

then $\sqcap_{\downarrow}\phi^k f_0$ is a fixed point of ϕ.

Proof. Since ϕ is an improver with respect to f_0 and ϕ is monotone we have that the sequence $(\phi^k f_0)_{k\in\omega}$ is decreasing in $\mathcal{C}^*_{\rightarrow}$. From continuity of ϕ we deduce

$$\lim_{n\rightarrow\infty} e_{\mathcal{C}^*_{\rightarrow}}(\phi(\sqcap_{\downarrow}\phi^k f_0), \phi^k f_0) = 0.$$

Consequently $\phi(\sqcap_{\downarrow}\phi^k f_0) = \sqcap_{\downarrow}\phi^k f_0$ because, by Proposition 1, $\sqcap_{\downarrow}\phi^k f_0$ is the unique $e_{\mathcal{C}^*_{\rightarrow}}$-limit point of $(\phi^k f_0)_{k\in\omega}$. The proof is complete.

4 An application of the space $(\mathcal{C}^*_{\rightarrow}, e_{\mathcal{C}^*_{\rightarrow}})$ to the complexity analysis of Quicksort

In this section we apply our approach to show that the recurrence induced by the particular class of comparison based sorting algorithms whose implementation follows a Quicksort schema has a unique solution, with asymptotic upper bound in the class $\mathcal{O}(n\log_2 n)$.

When discussing the complexity analysis (running time of computing) of Quicksort used by the Unix system, the following iterative recurrence is obtained for the average case (see Section 4 of [6]):

$$T(n) = \frac{n+1}{n}T(n-1) + \frac{2(n-1)}{n} \text{ for all } n \geq 2, \qquad (*)$$

where $T(1) = 0$.

Let T be a recurrence equation of type (*). We associate to T the mapping $\Phi_T : \mathcal{C}^*_{\rightarrow} \rightarrow \mathcal{C}^*_{\rightarrow}$ given by

$$(\Phi_T f)(n) = \begin{cases} 0 & n = 0, 1 \\ \frac{n+1}{n} f(n-1) + \frac{2(n-1)}{n} & \text{for all } n \in \mathrm{dom} f \backslash \{0, 1\} \end{cases}.$$

Note that $|\mathrm{dom}\Phi_T f| = |\mathrm{dom} f| + 1$, so for $f \in \mathcal{C}^*$ we have $\mathrm{dom}\Phi_T f = w$.

On the other hand, Φ_T is easily seen to be monotone, i.e. $\Phi_T f \leq_{\rightarrow} \Phi_T g$ whenever $f \leq_{\rightarrow} g$.

Next we show that the functional Φ_T is continuous. Consider a sequence $(f_k)_{k\in\mathbb{N}}$ in $\mathcal{C}^*_{\rightarrow}$ and $f \in \mathcal{C}^*_{\rightarrow}$ such that $\lim_{k\to\infty} e_{\mathcal{C}^*_{\rightarrow}}(f, f_k) = 0$. Then $f \leq_{\rightarrow} f_k$ eventually. Hence, by monotonicity of Φ_T, we have that $\Phi_T f \leq_{\rightarrow} \Phi_T f_k$ eventually. So

$$\begin{aligned}
e_{\mathcal{C}^*_{\rightarrow}}(\Phi_T f, \Phi_T f_k) &= \sum_{n\in \mathrm{dom}\Phi_T f_k} 2^{-n}(\Phi_T f_k(n) - \Phi_T f(n)) \\
&= \sum_{n\in |\mathrm{dom} f_k|+1} 2^{-n}(\Phi_T f_k(n) - \Phi_T f(n)) \\
&= \sum_{n=2}^{|\mathrm{dom} f_k|+1} 2^{-n-1}\frac{n+1}{2n}(f_k(n-1) - f(n-1)) \\
&\leq \frac{1}{2}\sum_{n\in \mathrm{dom} f_k} 2^{-n}(f_k(n) - f(n)) \\
&= \frac{1}{2} e_{\mathcal{C}^*_{\rightarrow}}(f, f_k)
\end{aligned}$$

eventually. Thus

$$\lim_{k\to\infty} e_{\mathcal{C}^*_{\rightarrow}}(\Phi_T f, \Phi_T f_k) = 0.$$

Consequently Φ_T is continuous.

Now let us denote by 0_1 the element of $\mathcal{C}^*_{\rightarrow}$ such that $\mathrm{dom}0_1 = \{0, 1\}$ and $0_1(0) = 0_1(1) = 0$. It is clear that $\Phi_T 0_1 \leq_{\rightarrow} 0_1$, and so Φ_T is an improver with respect to 0_1. It is obvious that $|\mathrm{dom}\Phi_T^k 0_1| = k+1$ and that $\Phi_T^k 0_1(n) = \Phi_T^{n-1} 0_1(n) = T(n)$ for all $n \in \mathrm{dom}\Phi_T^k 0_1$, with $n \geq 1$. Furthermore, it is not hard to check that the serie $\sum_{n=0}^{\infty} 2^{-n} T(n)$ converges. Consequently

$$\sum_{n=0}^{+\infty} 2^{-n} \sqcap_{\downarrow} \Phi_T^k 0_1(n) < +\infty$$

and thus $\sqcap_{\downarrow}\Phi_T^k 0_1 \in \mathcal{C}^*$.

On the other hand, it is clear that $e_{\mathcal{C}^*_{\rightarrow}}(\sqcap_{\downarrow}\Phi_T^k 0_1, \Phi_T^k 0_1) = 0$ for all $k \in \mathbb{N}$.

Now, applying Theorem 1 we deduce that Φ_T has as a fixed point $\sqcap_{\downarrow}\Phi_T^k 0_1$, i.e. $\Phi_T \sqcap_{\downarrow} \phi_T^k 0_1 = \sqcap_{\downarrow}\Phi_T^k 0_1$. Consequently the recurrence (*) has solution. In order to show the uniqueness we note that a solution of the recurrence (*) must be defined for all $n \in \mathbb{N}$. The desired conclusion, i.e. $\sqcap_{\downarrow}\Phi_T^k 0_1$ is the unique fixed point of Φ_T in $\mathcal{C}^*$, follows easily by induction, because $\sqcap_{\downarrow}\Phi_T^k 0_1(1) = 0$ and $g(1) = 0$ whenever g is a fixed point of Φ_T with $g \in \mathcal{C}^*$. We have shown, in the spirit of the principle of fixpoint induction, that the solution of the recurrence (*) can be seen as the limit of a sequence of approximations $(\Phi_T^k 0_1)_{k\in\omega}$.

Finally we prove that $\sqcap_{\downarrow}\Phi_T^k 0_1 \in \mathcal{O}(n\log_2 n)$. It is easy to see that the mapping Φ_T is an improver with respect to the complexity function $U \in \mathcal{C}^*$ given by $U(0) = 0$ and $U(n) = \frac{1}{2}n\log_2 n$ for $n \geq 1$. Then, by Theorem 1, we deduce that $\sqcap_{\downarrow}\Phi_T^k U$ is a fixed point of Φ_T in $\mathcal{C}^*$. Since $\sqcap_{\downarrow}\Phi_T^k 0_1$ is the unique fixed point of Φ_T in $\mathcal{C}^*$, $\sqcap_{\downarrow}\Phi_T^k 0_1 = \sqcap_{\downarrow}\Phi_T^k U \leq_{\rightarrow} U$. Whence we conclude that the running time of computing (for the average case) of a Quicksort algorithm is in $\mathcal{O}(n\log_2 n)$, as claimed.

5 Recursion in Denotational Semantics for programming languages: An extension of $(\mathcal{C}^*_{\rightarrow}, e_{\mathcal{C}^*_{\rightarrow}})$

Motivated, in part, by the work of E. A. Emerson and C. S. Jutla ([4]) about tree automata and modal logic, a general class of complexity spaces have been introduced and studied in [10, 11] to obtain an appropriate framework for efficient complexity analysis of algorithms with exponential running time of computing. By an exponential time algorithm we mean an algorithm whose running time is in the class $\mathcal{O}(2^{P(n)})$, where $P(n)$ is a polynomial such that $P(n) > 0$ for all $n \in \omega$. It is obvious that if $P(n) \geq n$ for all $n \in \omega$, and we associate the complexity of an algorithm of this type with a function f_P given by $f_P(n) = 2^{P(n)}$ for all $n \in \omega$, then $f_P \notin \mathcal{C}^*$. For this reason, fixed a polynomial $P(n)$ as before, the complexity structure presented in [11] consists of a pair $(\mathcal{C}^*_{P(n)}, d_{\mathcal{C}^*_{P(n)}})$ such that

$$\mathcal{C}^*_{P(n)} = \{f : \omega \to \mathbb{R}^+ : \sum_{n=0}^{+\infty} 2^{-P(n)} f(n) < +\infty\}$$

and

$$d_{\mathcal{C}^*_{P(n)}}(f,g) = \sum_{n=0}^{+\infty} 2^{-P(n)}[(g(n) - f(n)) \vee 0].$$

Now it is clear that $f_P \in \mathcal{C}^*_{P(n)}$. With the aim to go more deeply into the combination of the techniques of Denotational Semantics and Complexity Analysis, we construct, in this direction, a new complexity space which extends the old one $(\mathcal{C}^*_{\rightarrow}, e_{\mathcal{C}^*_{\rightarrow}})$. In order to motivate this new construction let us to show that the complexity space $(\mathcal{C}^*_{\rightarrow}, e_{\mathcal{C}^*_{\rightarrow}})$ can not be used, in general, as a mathematical model for the validation of recursive definitions of programs.

Indeed, consider the easy but representative example of a function which is given by a recursive specification, the factorial $fact$.

To implement an algorithm that computes the factorial of a nonnegative integer number it is needed the following recursive denotational specification (see, for instance, [12]):

$$fact(k) = \begin{cases} 1 & \text{if } k = 0 \\ k fact(k-1) & \text{if } k \geq 1 \end{cases} .$$

The preceding denotational specification has the drawback that the meaning of the symbol $fact$, which is given by the right hand side, is expressed again in terms of $fact$. So the symbol $fact$ can not be replaced by its meaning because the meaning also contains the symbol. Furthermore, it is obvious that the entire factorial function is not computable in a finite numbers of steps although, given $k \in \omega$, it is clear that the value $k!$ can be computed in a finite number of steps.

The usual method used to avoid this handicap is to consider a nonrecursive functional ϕ defined on the set of partial mappings as follows:

$$\phi f(k) = \begin{cases} 1 & \text{if } k = 0 \\ k f(k-1) & \text{if } k \geq 1 \text{ and } k-1 \in \mathrm{dom} f \end{cases} ,$$

and then to show that $fact$ is a fixed point of ϕ. Our purpose here is to prove that such a denotational specification is meaningful using as the support space of ϕ our complexity structure, and applying the fixed point induction. However it is evident that the function $fact$ (the solution of the recursive equation) is not in $\mathcal{C}^*_{\rightarrow}$, because $\sum_{n=0}^{+\infty} 2^{-n} n! = +\infty$. To obtain our aim we propose, similarly to [11], the following generalization of the complexity space $(\mathcal{C}^*_{\rightarrow}, e_{\mathcal{C}^*_{\rightarrow}})$.

Fixed a polynomial $P(n)$, with $P(n) > 0$ for all $n \in \omega$, set

$$\mathcal{C}^*_{\rightarrow, P(n)} = \{f \in [(\omega \rightharpoonup \mathbb{R}^+)] : \sum_{n \in \mathrm{dom} f} 2^{-P(n)} f(n) < +\infty\}.$$

Note that the partial order $\leq_{\rightarrow}$ remains valid on $\mathcal{C}^*_{\rightarrow, P(n)}$.

Define the nonnegative real valued function $e_{\mathcal{C}^*_{\rightarrow, P(n)}}$ on $\mathcal{C}^*_{\rightarrow, P(n)} \times \mathcal{C}^*_{\rightarrow, P(n)}$ given by

$$e_{\mathcal{C}^*_{\rightarrow, P(n)}}(f, g) = \begin{cases} \sum_{n \in \mathrm{dom} g} 2^{-P(n)}(g(n) - f(n)) & \text{if } f \leq_{\rightarrow} g \\ +\infty & \text{otherwise.} \end{cases} .$$

Obviously $\mathcal{C}^*_{\rightarrow} \subsetneq \mathcal{C}^*_{\rightarrow, P(n)}$, $\mathcal{C}^*_{P(n)} \subsetneq \mathcal{C}^*_{\rightarrow, P(n)}$ and $e_{\mathcal{C}^*_{\rightarrow, P(n)}}|_{\mathcal{C}^*_{\rightarrow}} = e_{\mathcal{C}^*_{\rightarrow}}$.

Denote by $0_{\mathcal{C}^*_{\rightarrow, P(n)}}(n) = 0$ for all $n \in \omega$.

Under these conditions, it is a simple matter to prove the next results.

Proposition 2. *The pair $(\mathcal{C}^*_{\rightarrow,P(n)}, \leq_{\rightarrow})$ is a pointed ordered (noncancellative) cone with bottom element $0_{\mathcal{C}^*_{\rightarrow,P(n)}}$.*

Proposition 3. *The function $e_{\mathcal{C}^*_{\rightarrow,P(n)}}$ is a bicomplete subinvariant extended quasi-metric on $\mathcal{C}^*_{\rightarrow,P(n)}$.*

Next we show that the mapping

$$\Phi : \mathcal{C}^*_{\rightarrow,P(n)} \rightarrow \mathcal{C}^*_{\rightarrow}$$

given by

$$\Phi f(n) = 2^{n-P(n)} f(n)$$

for all $f \in \mathcal{C}^*_{\rightarrow,P(n)}$ and $n \in \omega$, is an isometry from $(\mathcal{C}^*_{\rightarrow,P(n)}, e_{\mathcal{C}^*_{\rightarrow,P(n)}})$ onto $(\mathcal{C}^*_{\rightarrow}, e_{\mathcal{C}^*_{\rightarrow}})$.

Indeed, it is clear that Φ is a bijection. Moreover, $f \leq_{\rightarrow} g$ if and only if $\Phi f \leq_{\rightarrow} \Phi g$. Finally, if $f \leq_{\rightarrow} g$, we have

$$\begin{aligned} e_{\mathcal{C}^*_{\rightarrow}}(\Phi f, \Phi g) &= \sum_{n \in \mathrm{dom}\Phi g} 2^{-n}(\Phi g(n) - \Phi f(n)) \\ &= \sum_{n \in \mathrm{dom} g} 2^{-P(n)}(g(n) - f(n)) = e_{\mathcal{C}^*_{\rightarrow,P(n)}}(f, g). \end{aligned}$$

Therefore, and by adapting, in the obvious way, the notion of an improver to $(\mathcal{C}^*_{\rightarrow,P(n)}, \leq_{\rightarrow})$, Proposition 1 and Theorem 1 above, are generalized as follows.

Proposition 4. *Let $(f_k)_{k\in\mathbb{N}}$ be a decreasing sequence in $\mathcal{C}^*_{\rightarrow,P(n)}$ such that $\sqcap_{\downarrow} f_k \in \mathcal{C}^*_{P(n)}$ and $|\mathrm{dom} f_k|$ is finite for all $k \in \mathbb{N}$. If $\lim_{k\to\infty} e_{\mathcal{C}^*_{\rightarrow,P(n)}}(\sqcap_{\downarrow} f_k, f_k) = 0$, then $\sqcap_{\downarrow} f_k$ is the unique $e_{\mathcal{C}^*_{\rightarrow},P(n)}$-limit point of $(f_k)_{k\in\mathbb{N}}$.*

Theorem 2. *Let ϕ be a continuous monotone mapping from the complexity space $(\mathcal{C}^*_{\rightarrow,P(n)}, e_{\mathcal{C}^*_{\rightarrow,P(n)}})$ into itself. If ϕ is an improver with respect to any $f_0 \in \mathcal{C}^*_{\rightarrow,P(n)}$ such that $\sqcap_{\downarrow} \phi^k f_0 \in \mathcal{C}^*_{P(n)}$ and*

$$\lim_{k\to\infty} e_{\mathcal{C}^*_{\rightarrow,P(n)}}(\sqcap_{\downarrow} \phi^k f_0, \phi^k f_0) = 0,$$

then $\sqcap_{\downarrow} \phi^k f_0$ is a fixed point of ϕ.

In the rest of the section we show that this approach is suitable to prove mathematically when a function defined recursively is consistent as we announced before. In particular we give an alternative proof of the fact that the factorial semantic specification is meaningful, and we do this by means of the principle of fixed point induction showing that the factorial function (the total complexity mapping) can be considered as the limit of a sequence of approximations (complexity partial mappings) which can be computed in a finite number of steps.

From now on we consider the polynomial $P(n)$ given by $P(n) = n^2$ for all $n \in \omega$.

Denote by $1_{\mathcal{C}^*_{\to,n^2}}$ the element of $\mathcal{C}^*_{\to,n^2}$ such that $\mathrm{dom}1_{\mathcal{C}^*_{\to,n^2}} = \{0\}$ and $1_{\mathcal{C}^*_{\to,n^2}}(0) = 1$. Consider the functional $\phi : \mathcal{C}^*_{\to,n^2} \to \mathcal{C}^*_{\to,n^2}$ defined by

$$\phi f(k) = \begin{cases} 1 & \text{if } k = 0 \\ kf(k-1) & \text{if } k \geq 1 \text{ and } k-1 \in \mathrm{dom} f \end{cases}.$$

It is clear that ϕ is monotone and it is an improver with respect to $1_{\mathcal{C}^*_{\to,n^2}}$.

Next we prove that ϕ is continuous. Indeed, let $(f_k)_{k\in\mathbb{N}}$ be a sequence in $\mathcal{C}^*_{\to,n^2}$ and let $f \in \mathcal{C}^*_{\to,n^2}$ be such that $\lim_{k\to\infty} e_{\mathcal{C}^*_{\to,n^2}}(f, f_k) = 0$. Then $f \leq_{\to} f_k$ eventually. By monotonicity of ϕ we obtain $\phi f \leq_{\to} \phi f_k$ eventually. Moreover,

$$e_{\mathcal{C}^*_{\to,n^2}}(\phi f, \phi f_k) \leq |\mathrm{dom} f_k| e_{\mathcal{C}^*_{\to,n^2}}(f, f_k)$$

eventually. So $\lim_{k\to\infty} e_{\mathcal{C}^*_{\to,n^2}}(\phi f, \phi f_k) = 0$ and, thus, ϕ is continuous.

Note that $\phi^k 1_{\mathcal{C}^*_{\to,n^2}}(n) = n!$ for all $n \in \mathrm{dom}\phi^k 1_{\mathcal{C}^*_{\to,n^2}}$.

On the other hand, we have that $\mathrm{dom}\sqcap_{\downarrow}\phi^k 1_{\mathcal{C}^*_{\to,n^2}} = \omega$ and $\sqcap_{\downarrow}\phi^k 1_{\mathcal{C}^*_{\to,n^2}}(n) = n!$ for all $n \in \omega$, since $\lim_{n\to\infty}|\mathrm{dom}\phi^k 1_{\mathcal{C}^*_{\to,n^2}}| = \lim_{n\to\infty} k = +\infty$ and $\sqcap_{\downarrow}\phi^k 1_{\mathcal{C}^*_{\to,n^2}}(n) = \phi^n 1_{\mathcal{C}^*_{\to,n^2}}(n) = n!$ for all $n \in \omega$. Moreover,

$$\sum_{n=0}^{+\infty} 2^{-n^2} \sqcap_{\downarrow} \phi^k 1_{\mathcal{C}^*_{\to,n^2}}(n) = \sum_{n=1}^{+\infty} 2^{-n^2} n! < +\infty.$$

So $\sqcap_{\downarrow}\phi^k 1_{\mathcal{C}^*_{\to,n^2}} \in \mathcal{C}^*_{n^2}$. Since

$$e_{\mathcal{C}^*_{\to,n^2}}(\sqcap_{\downarrow}\phi^k 1_{\mathcal{C}^*_{\to,n^2}}, \phi^k 1_{\mathcal{C}^*_{\to,n^2}}) = 0$$

for all $k \in \omega$, we have, by Theorem 2, that $\sqcap_{\downarrow}\phi^k 1_{\mathcal{C}^*_{\to,n^2}}$ is a fixed point of ϕ. So we have obtained the factorial (the meaning of the recursive denotational definition) as the fixed point $\sqcap_{\downarrow}\phi^k 1_{\mathcal{C}^*_{\to,n^2}}$, which is the limit of the partial mappings $(\phi^k 1_{\mathcal{C}^*_{\to,n^2}})_{k\in\omega}$ that allow us to obtain each computation of the factorial in a finite number of steps.

References

[1] J. W. de Bakker and E. P. de Vink, *Control Flow Semantics,* The MIT Press, Cambridge, Massachusetts, 1996.

[2] J. W. de Bakker and E. P. de Vink, *A metric approach to control flow semantics,* in: Proc. Eleventh Summer Conference on General Topology and Applications, Ann. New York Acad. Sci. **806** (1996), 11-27.

[3] J. W. de Bakker and E. P. de Vink, *Denotational models for programming languages: applications of Banach's fixed point theorem,* Topology Appl. **85** (1998), 35–52.

[4] A. C. Emerson, C. S. Jutla, *The complexity of tree automata and logic of programs,* Siam J. Comput. **29** (1999), 132-158.

[5] B. A. Davey, H. A. Priestley, *Introduction to Lattices and Order,* Cambridge University Press, 1990.

[6] P. Flajolet, *Analytic analysis of algorithms*, in: 19th Internat. Colloq. ICALP'92, Vienna, July 1992; Automata, Languages and Programming, Lecture Notes in Computer Science 623, W. Kuich editor (1992), pp. 186–210.

[7] P. Fletcher, W.F. Lindgren, *Quasi-Uniform Spaces,* Marcel Dekker, New York, 1982.

[8] B. Fuchssteiner, W. Lusky, *Convex Cones*, North Holland, 1981.

[9] L.M. García-Raffi, S. Romaguera, E.A. Sánchez-Pérez, *Sequence spaces and asymmetric norms in the theory of computational complexity,* Math. Comput. Model. **36** (2002), 1-11.

[10] L.M. García-Raffi, S. Romaguera, E.A. Sánchez-Pérez, *The supremum asymmetric norm on sequence algebras: A general framework to measure complexity distances,* Electronic Notes Theoret. Comput. Sci. **74** (2003), URL:http//www.elsevier.nl/locate/entcs/volume74.html 12 pages.

[11] L.M. García-Raffi, S. Romaguera, E.A. Sánchez-Pérez, O. Valero, *Normed semialgebras: a mathematical model for the complexity analysis of programs and algorithms,* Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics Vol. II, 2003, pp. 55-58.

[12] C.A. Gunter, D.S. Scott, *Semantic domains,* in: Handbook of Theoretical Computer Science, Jan van Leeuwen (ed.), Elsevier Science Publishers, vol. B: Formal Models and Semantics, 633-674, 1990.

[13] J.I. Hartog, J.W. de Bakker, E.P. De Vink, *Metric semantics and full abstractness for action refinement and probabilistic choice,* Electronic Notes Theoret. Comput. Sci. **40** (2001), 28 pages.

[14] J.L. Kelley, *General Topology,* Springer-Verlag, 1955.

[15] H.P.A. Künzi, *Nonsymmetric distances and their associated topologies: About the origins of basic ideas in the area of asymmetric topology,* in: Handbook of the History of General Topology, C.E. Aull and R. Lowen (eds.), Kluwer Acad. Publ. vol. 3, 853-968, 2001.

[16] M.O. O'Keeffe, S. Romaguera, M. Schellekens, *Norm-weightable Riesz spaces and the dual complexity space,* Electronic Notes Theoret. Comput. Sci. **74** (2003), URL:http//www.elsevier.nl/locate/entcs/volume74.html 17 pages.

[17] J. Rodríguez-López, *A new approach to epiconvergence and some applications*, Southeast Asian Bulletin of Mathematics **28** (2004), 685-701.

[18] J. Rodríguez-López, S. Romaguera, O. Valero, *Asymptotic complexity of algorithms via the nonsymmetric Hausdorff distance*, Computing Letters **2** (2006), 155-161.

[19] J. Rodríguez-López, S. Romaguera, O. Valero, *Denotational semantics for programming languages, balanced quai-metrics and fixed points,* Int. J. Comput. Math **85** (2008), 623-630.

[20] S. Romaguera, E.A. Sánchez-Pérez, O. Valero, *The complexity space of a valued linearly ordered set,* Electronic Notes Theoret. Comput. Sci. **74** (2003), URL:http//www.elsevier.nl/locate/entcs/volume74.html 14 pages.

[21] S. Romaguera, E.A. Sánchez-Pérez, O. Valero, *Computing complexity distances between algorithms,* Kybernetika **39** (2003), 569-582.

[22] S. Romaguera, M. Schellekens, *Quasi-metric properties of complexity spaces,* Topology Appl. **98** (1999), 311-322.

[23] S. Romaguera, M. Schellekens, *The quasi-metric of complexity convergence,* Quaestiones Math. **23** (2000), 359-374.

[24] S. Romaguera, M. Schellekens, *Duality and quasi-normability for complexity spaces,* Appl. Gen. Topology **3** (2002), 91-112.

[25] S. Romaguera, O. Valero, *On the structure of the space of complexity partial functions,* Int. J. Comput. Math. **85** (2008), 631-640.

[26] M. Schellekens, *The Smyth completion: a common foundation for denonational semantics and complexity analysis,* In: Proc. MFPS 11, Electronic Notes Theoret. Comput. Sci. **1** (1995), 535-556.

[27] M. Schellekens, *Complexity domains, a relative view of complexity,* manuscript.

[28] R.D. Tennent, *The denotational semantics of programming languages,* Comm. ACM **19 (**1976**),** 437-453.

[29] R. Tix, K. Keimel, G. Plotkin, *Semantic domains for combining probability and non-determinism,* Electronic Notes Theoret. Comput. Sci. **129** (2005), 1-104.