RESEARCH

Fixed point theorems in generalized metric spaces with applications to computer science

Maryam A Alghamdi[1], Naseer Shahzad[2*] and Oscar Valero[3]

*Correspondence:
nshahzad@kau.edu.sa
[2]Department of Mathematics, King
Abdulaziz University, P.O. Box 80203,
Jeddah, 21859, Saudi Arabia
Full list of author information is
available at the end of the article

Abstract

In 1994, Matthews introduced the notion of a partial metric space in order to obtain a suitable mathematical tool for program verification (Matthews in Ann. N.Y. Acad. Sci. 728:183-197, 1994). He gave an application of this new structure to formulate a suitable test for lazy data flow deadlock in Kahn's model of parallel computation by means of a partial metric version of the celebrated Banach fixed point theorem (Matthews in Theor. Comput. Sci. 151:195-205, 1995). In this paper, motivated by the utility of partial metrics in computer science, we discuss whether they are a suitable tool for asymptotic complexity analysis of algorithms. Concretely, we show that the Matthews fixed point theorem does not constitute, in principle, an appropriate implement for the aforementioned purpose. Inspired by the preceding fact, we prove two fixed point theorems which provide the mathematical basis for a new technique to carry out asymptotic complexity analysis of algorithms via partial metrics. Furthermore, in order to illustrate and to validate the developed theory, we apply our results to analyze the asymptotic complexity of two celebrated recursive algorithms.

MSC: 47H10; 54E50; 54F05; 68Q25; 68W40

Keywords: partial metric; fixed point; asymptotic complexity analysis; recurrence equation; running time of computing

1 Introduction

When a computational program uses a recursion process to find the solution to a problem, such a process is characterized by obtaining in each step of the computation an approximation to the aforementioned solution which is better than the approximations obtained in the preceding steps and, in addition, by obtaining always the final approximation to the problem solution as the 'limit' of the computing process. A mathematical model to this sort of situations is the so-called Scott model which is based on ideas from order theory and topology (see [1, 2] for a detailed account of the Scott model and its applications). In particular, the order represents some notion of information in such a way that each step of the computation is identified with an element of the mathematical model which is greater than (or equal to) the other ones associated with the preceding steps, since each approximation gives more information about the final solution than those computed before. The final output of the computational process is seen as the limit of the successive approximations. Thus the recursion processes are modeled as increasing sequences of elements of the mathematical model, which is identified with an ordered set, that converge to its least upper bound with respect to the given topology.

In 1994, Matthews introduced the notion of Scott-like topology as a mathematical framework to model increasing information content sequences in computer science in the spirit of Scott [3].

Let us recall, with the aim of reminding the concept of Scott-like topology, that a pair $(X, \leq)$ is said to be an ordered set provided that X is a nonempty set and $\leq$ is a reflexive, antisymmetric and transitive binary relation on X [2]. Given a subset $Y \subseteq X$, an upper bound for Y is an element $x \in X$ such that $y \leq x$ for all $y \in Y$. A least element for Y is an element $z \in Y$ such that $z \leq y$ for all $y \in Y$. Moreover, a sequence $(x_n)_{n \in \mathbb{N}}$ in $(X, \leq)$ is increasing if $x_n \leq x_{n+1}$ for all $n \in \mathbb{N}$.

According to Matthews, a weakly order consistent topology over an ordered set $(X, \leq)$ is a topology $\mathcal{T}$ over X such that $x \leq y \Leftrightarrow x \in \text{cl}(y)$ for all $x, y \in X$, where by $\text{cl}(y)$ we denote the closure of $\{y\}$ with respect to $\mathcal{T}$. Furthermore, a Scott-like topology over an ordered set $(X, \leq)$ is a weakly consistent topology $\mathcal{T}$ over X satisfying the following properties:

(1) every increasing sequence $(x_n)_{n \in \mathbb{N}}$ in $(X, \leq)$ has least upper bound, where $\mathbb{N}$ denotes the set of positive integer numbers;

(2) for every $O \in \mathcal{T}$ containing the least upper bound of an increasing sequence $(x_n)_{n \in \mathbb{N}}$, there exists $n_0 \in \mathbb{N}$ such that $x_n \in O$ for all $n > n_0$.

In the aforesaid reference [3], Matthews introduced also the notion of a partial metric. In order to recall this new concept, let us denote by $\mathbb{R}^+$ the set of nonnegative real numbers.

Following [3], a partial metric on a nonempty set X is a function $p : X \times X \to \mathbb{R}^+$ such that for all $x, y, z \in X$:

(i) $p(x, x) = p(x, y) = p(y, y) \Leftrightarrow x = y$;

(ii) $p(x, x) \leq p(x, y)$;

(iii) $p(x, y) = p(y, x)$;

(iv) $p(x, y) \leq p(x, z) + p(z, y) - p(z, z)$.

Of course a partial metric space is a pair (X, p) such that X is a nonempty set and p is a partial metric on X.

The concept of a partial metric, since its introduction by Matthews, has been widely accepted in computer science (see, for instance, [4–15]). This is due to the fact that partial metric spaces can be used as a mathematical tool to model computational processes in the spirit of Scott. Indeed, each partial metric p on X generates a T_0 topology $\mathcal{T}(p)$ on X which has as a base the family of open p-balls $\{B_p(x, \varepsilon) : x \in X, \varepsilon > 0\}$, where $B_p(x, \varepsilon) = \{y \in X : p(x, y) < p(x, x) + \varepsilon\}$ for all $x \in X$ and $\varepsilon > 0$. Moreover, every partial metric p induces an order $\leq_p$ on X as follows: $x \leq_p y \Leftrightarrow p(x, y) = p(x, x)$.

The next result reveals the reason for which the partial metric spaces can be used as a suitable mathematical tool to describe Scott processes (see [3] for a deeper discussion).

Proposition 1 *Let (X, p) be a complete partial metric space. Then the partial metric topology $\mathcal{T}(p)$ is a Scott-like topology over $(X, \leq_p)$, and thus every increasing sequence $(x_n)_{n \in \mathbb{N}}$ in $(X, \leq_p)$ has least upper bound and converges to it with respect to $\mathcal{T}(p)$.*

Remark 2 Note that the fact that the sequence $(x_n)_{n \in \mathbb{N}}$ is ascending with least upper bound x provides that $x_n \leq_p x$ for all $n \in \mathbb{N}$ and, thus, that $p(x_n, x) - p(x_n, x_n) = 0$. So, in the preceding proposition, we actually have the convergence of $(x_n)_{n \in \mathbb{N}}$ to x is with respect to $\mathcal{T}(p^s)$.

Inspired by the applications to program verification, Matthews extended Banach's fixed point theorem to the framework of partial metric spaces in [3], and he used it to formulate a suitable test for lazy data flow deadlock in Kahn's model of parallel computation in [4]. The aforementioned extension of Banach's fixed point theorem can be stated as follows.

Theorem 3 *Let f be a mapping of a complete partial metric space (X, p) into itself such that there is $s \in [0, 1[$ such that*

$$p\big(f(x), f(y)\big) \leq sp(x, y)$$

for all $x, y \in X$. Then f has a unique fixed point. Moreover, if $x^* \in X$ is the fixed point of f, then $p(x^*, x^*) = 0$.

According to [3], a sequence $(x_n)_{n \in \mathbb{N}}$ in a partial metric space (X, p) converges to a point $x \in X$ with respect to $\mathcal{T}(p) \Leftrightarrow p(x, x) = \lim_{n \to \infty} p(x, x_n)$. Moreover, a sequence $(x_n)_{n \in \mathbb{N}}$ in a partial metric space (X, p) is called a Cauchy sequence if $\lim_{n,m \to \infty} p(x_n, x_m)$ exists and is finite. Furthermore, a partial metric space (X, p) is said to be complete if every Cauchy sequence $(x_n)_{n \in \mathbb{N}}$ in X converges, with respect to $\mathcal{T}(p)$, to a point $x \in X$ such that $p(x, x) = \lim_{n,m \to \infty} p(x_n, x_m)$.

According to [3], every partial metric induces in a natural way a metric. Indeed, given a partial metric space (X, p), then the function $p^s : X \times X \to \mathbb{R}^+$ defined by

$$p^s(x, y) = 2p(x, y) - p(x, x) - p(y, y)$$

for all $x, y \in X$ is a metric on X. Moreover, a sequence $(x_n)_{n \in \mathbb{N}}$ in (X, p) converges to $x \in X$ with respect to $\mathcal{T}(p^s)$ if and only if $\lim_{n \to \infty} p(x, x_n) = \lim_{n \to \infty} p(x_n, x_n) = p(x, x)$. Furthermore, a sequence in (X, p) is Cauchy if and only if it is Cauchy in the metric space (X, p^s), and (X, p) is complete if and only if (X, p^s) is complete.

Since Matthews published Theorem 3, an intense research activity on fixed point results in partial metric spaces have been developed. A large number of fixed point results in the metric framework have been extended to the partial metric context in [13, 16–41].

In the light of the facts that, on the one hand, partial metric spaces are a useful tool for solving practical problems that arise in several fields of computer science and, on the other hand, the scientific community has a growing interest in partial metric fixed point theory, in this paper our goal is to show that partial metric spaces can be used satisfactorily for the asymptotic complexity analysis of algorithms by means of fixed point techniques. To this end, we discuss whether Theorem 3 can be used for such a purpose. However, we show that, in principle, this question has a negative answer and, for this reason, we delve into the study of fixed point techniques in asymptotic complexity analysis of algorithms. Thus, we present a mathematical technique for discussing the complexity of algorithms whose foundation lies in the use of two new fixed point results that we provide for partial metric spaces. In order to show the potential applicability of the developed theory and to validate our new fixed point technique, we apply it to analyze formally the asymptotic complexity of two celebrated recursive algorithms, namely, Quicksort and Hanoi.

The remainder of the paper is organized as follows. In Section 2 we prove the announced fixed point theorems for self-mappings defined on complete partial metric spaces. More-

over, in the same section, we provide examples that show that the hypothesis in the statements of our results cannot be weakened. Section 3 is devoted to introducing the reader to fixed point techniques for asymptotic complexity analysis of algorithms. Concretely, general and fundamental aspects of asymptotic complexity analysis are recalled and, in addition, the reference fixed point technique to carry out the complexity analysis of algorithms, due to Schellekens and in which our study is based, is presented in detail in order to motivate our subsequent work developed in Section 4. In the latter section, we discuss the feasibility of using the Matthews fixed point theorem as a mathematical tool for the asymptotic complexity analysis of algorithms and, in particular, we show that the aforesaid result is not, in principle, appropriate for such a purpose. Accordingly, in the same section, we introduce a new mathematical technique for the asymptotic complexity analysis of algorithms whose basis is provided by the fixed point results proved in Section 2. We end the section, and thus the paper, applying the developed fixed point method to analyze the asymptotic complexity of the aforesaid recursive algorithms.

2 The fixed point theorems

In this section we present our main results which will play a central role in the application to asymptotic complexity analysis developed in Section 4.1. To this end, let us recall that a mapping f from an ordered set $(X, \leq)$ into itself is monotone if $f(x) \leq f(y)$ whenever $x \leq y$. Moreover, according to [42], a mapping from an ordered set $(X, \leq)$ into itself is said to be $\leq$-continuous provided that the least upper bound of $(f(x_n))_{n \in \mathbb{N}}$ is $f(x^*)$ for every increasing sequence $(x_n)_{n \in \mathbb{N}}$ whose least upper bound exists and is x^*. Of course every $\leq$-continuous mapping is monotone.

Theorem 4 *Let (X, p) be a complete partial metric space and let $f : X \to X$ be a $\leq_p$-continuous mapping. If there exists $x_0 \in X$ such that $x_0 \leq_p f(x_0)$, then f has a fixed point in $\uparrow x_0 = \{x \in X : x_0 \leq_p x\}$.*

Proof Let $x_0 \in X$ such that $x_0 \leq_p f(x_0)$. Since f is monotone, we have that

$$x_0 \leq_p f(x_0) \leq_p f^2(x_0) \leq_p \cdots \leq_p f^n(x_0) \leq_p f^{n+1}(x_0) \leq_p \cdots.$$

Observe that we can assume, without loss of generality, that $x_0 \neq f(x_0)$ since otherwise we have guaranteed the existence of the fixed point in $\uparrow x_0$.

Since the sequence $(f^n(x_0))_{n \in \mathbb{N}}$ is increasing in $(X, \leq_p)$, we have, by Proposition 1 and Remark 2, that there exists $x^* \in X$ such that x^* is the least upper bound of $(f^n(x_0))_{n \in \mathbb{N}}$ and, in addition, that $\lim_{n \to \infty} f^n(x_0) = x^*$ with respect to $\mathcal{T}(p^s)$. Since f is $\leq_p$-continuous, we have that $f(x^*)$ is the least upper bound of $(f^n(x_0))_{n \in \mathbb{N}}$. Whence we immediately obtain that $f(x^*) = x^*$ and that $x^* \in \uparrow x_0$. □

Remark 5 Observe that the proof of Theorem 4 follows applying similar arguments to those given in the proof of Kleen's theorem or Tarski-Kantorovitch's theorem for mappings defined from ω-chain-complete ordered sets into itself (see [42, 43] for more details). However, we have included a detailed proof of the aforementioned result for the sake of completeness and in order to help the reader.

The next example shows that the $\leq_p$-continuity of the mapping cannot be deleted in the statement of Theorem 4.

Example 6 Let p be the partial metric space on $(0, \infty]$ given by

$$p_\infty(x, y) = \max\left\{\frac{1}{x}, \frac{1}{y}\right\}$$

for all $x, y \in (0, \infty]$, where we adopt the convention that $\frac{1}{\infty} = 0$. It is not hard to check that the partial metric space $((0, \infty], p_\infty)$ is complete and that $x \leq_{p_\infty} y \Leftrightarrow x \leq_\infty y$, where $\leq_\infty$ stands for the usual order on the extended real line. Consider the subset $X = \{1, 2\}$ of $(0, \infty]$. Then the partial metric space (X, p_∞) is also complete. Now, define the mapping $f : X \to X$ by $f(1) = 2$ and $f(2) = 1$. Clearly, $1 \leq_{p_\infty} = f(1) = 2$. Observe that f is not $\leq_p$-continuous because f is not monotone. In fact, $1 \leq_{p_\infty} 2$ and $2 = f(1) \not\leq_{p_\infty} f(2) = 1$. It is clear that f has no fixed points.

Let us recall that, given a partial metric (X, p), a mapping $f : X \to X$ is continuous provided that f is continuous from $(X, \mathcal{T}(p))$ into itself.

Remark 7 Note that every monotone and continuous mapping f from a complete partial metric space into itself is $\leq_p$-continuous. Indeed, assume that $((x_n))_{n \in \mathbb{N}}$ is an increasing sequence in $(X, \leq_p)$. Since (X, p) is complete, we have guaranteed, by Proposition 1 and Remark 2, that there exists $x^* \in X$ such that x^* is the least upper bound of $(x_n)_{n \in \mathbb{N}}$ and, in addition, that $\lim_{n \to \infty} x_n = x^*$ with respect to $\mathcal{T}(p^s)$. Since f is continuous, we have that $\lim_{n \to \infty} f(x_n) = f(x^*)$ with respect to $\mathcal{T}(p)$. The monotonicity of f provides that $f(x_n) \leq_p f(x^*)$ and, thus, that $p(f(x), f(x_n)) - p(f(x_n), f(x_n)) = 0$. Whence we deduce that $\lim_{n \to \infty} f(x_n) = f(x^*)$ with respect to $\mathcal{T}(p^s)$. Moreover, since the sequence $(f(x_n))_{n \in \mathbb{N}}$ is increasing and the partial metric space (X, p) is complete, we have guaranteed the existence of the least upper bound of $(f(x_n))_{n \in \mathbb{N}}$, say $y^* \in X$, in such a way that $\lim_{n \to \infty} f(x_n) = y^*$ with respect to $\mathcal{T}(p^s)$. Therefore $f(x^*) = y^*$, as claimed.

In the light of Theorem 4 and Remark 7, we immediately obtain the following result.

Corollary 8 *Let (X, p) be a complete partial metric space and let $f : X \to X$ be a monotone and continuous mapping. If there exists $x_0 \in X$ such that $x_0 \leq_p f(x_0)$, then f has a fixed point in $\uparrow x_0 = \{x \in X : x_0 \leq_p x\}$.*

In the following, given a partial metric space (X, p), we will say that a mapping $f : X \to X$ is p^s-continuous provided that f is continuous from $(X, \mathcal{T}(p^s))$ into itself. In our subsequent result, the p^s-continuity plays a central role.

Theorem 9 *Let (X, p) be a complete partial metric space and let $f : X \to X$ be a monotone and p^s-continuous mapping. If there exists $x_0 \in X$ such that $f(x_0) \leq_p x_0$, then f has a fixed point in $\downarrow x_0 = \{x \in X : x \leq_p x_0\}$.*

Proof Let $x_0 \in X$ such that $f(x_0) \leq_p x_0$. We assume, without loss of generality, that $x_0 \neq f(x_0)$ since otherwise we have guaranteed the existence of the fixed point in $\downarrow x_0$.

Since f is monotone, we have that

$$x_0 \geq_p f(x_0) \geq_p f^2(x_0) \geq_p \cdots \geq_p f^n(x_0) \geq_p f^{n+1}(x_0) \geq_p \cdots.$$

It follows that

$$p\big(f^n(x_0), f^n(x_0)\big) \le p\big(f^n(x_0), f^{n+1}(x_0)\big) = p\big(f^{n+1}(x_0), f^{n+1}(x_0)\big)$$

for all $n \in \mathbb{N}$. Thus the sequence $(p(f^n(x_0), f^n(x_0)))_{n\in\mathbb{N}}$ is decreasing in $\mathbb{R}^+$. So, there exists $r \in \mathbb{R}^+$ such that $\lim_{n\to\infty} p(f^n(x_0), f^n(x_0)) = r$. Then $\lim_{m,n\to\infty} p(f^m(x_0), f^n(x_0)) = r$, since $p(f^m(x_0), f^n(x_0)) = p(f^n(x_0), f^n(x_0))$ for all $m, n \in \mathbb{N}$ with $n \ge m$. It follows that the sequence $(f^n(x_0))_{n\in\mathbb{N}}$ is Cauchy. The fact that (X, p) is complete yields the existence of $x^* \in X$ such that $\lim_{n\to\infty} p(f^n(x_0), f^n(x_0)) = \lim_{n\to\infty} p(f^n(x_0), x^*) = p(x^*, x^*)$.

Next we show that $x^* \in\, \downarrow x_0$. It is clear that

$$p\big(x^*, x_0\big) - p\big(x^*, x^*\big)$$
$$\le p\big(x^*, f^n(x_0)\big) + p\big(f^n(x_0), x_0\big) - p\big(f^n(x_0), f^n(x_0)\big) - p\big(x^*, x^*\big)$$
$$= p\big(x^*, f^n(x_0)\big) - p\big(x^*, x^*\big)$$

since $p(f^n(x_0), x_0) = p(f^n(x_0), f^n(x_0))$ for all $n \in \mathbb{N}$. By the fact that $\lim_{n\to\infty} p(f^n(x_0), x^*) = p(x^*, x^*)$, we conclude that $p(x^*, x_0) = p(x^*, x^*)$ and, thus, that $x^* \le_p x_0$.

Now we prove that x^* is a fixed point of f. On the one hand, we have that

$$p\big(x^*, f(x^*)\big) - p\big(f(x^*), f(x^*)\big)$$
$$\le p\big(x^*, f^n(x_0)\big) + p\big(f^n(x_0), f(x^*)\big) - p\big(f^n(x_0), f^n(x_0)\big) - p\big(f(x^*), f(x^*)\big)$$

for all $n \in \mathbb{N}$. On the other hand, we have that $\lim_{n\to\infty} p(f^n(x_0), f^n(x_0)) = p(f^n(x_0), x^*)$ and, by the p^s-continuity of f, that $\lim_{n\to\infty} p(f^n(x_0), f(x^*)) = p(f(x^*), f(x^*))$. Thus, we deduce that $p(x^*, f(x^*)) = p(f(x^*), f(x^*))$. So, we obtain that $f(x^*) \le_p x^*$.

Moreover, we have that

$$p\big(x^*, f(x^*)\big) - p\big(x^*, x^*\big)$$
$$\le p\big(x^*, f^n(x_0)\big) + p\big(f^n(x_0), f(x^*)\big) - p\big(f^n(x_0), f^n(x_0)\big) - p\big(x^*, x^*\big)$$

for all $n \in \mathbb{N}$. Again, we have that $\lim_{n\to\infty} p(f^n(x_0), x^*) = p(x^*, x^*)$ and, by the p^s-continuity of f, that $\lim_{n\to\infty} p(f^n(x_0), f^n(x_0)) = p(f^n(x_0), f(x^*))$. Thus we deduce that $p(x^*, f(x^*)) = p(x^*, x^*)$. Whence we obtain that $x^* \le_p f(x^*)$.

We conclude from $f(x^*) \le_p x^* \le_p f(x^*)$ that $f(x^*) = x^*$. □

The next examples show that the p^s-continuity and monotonicity of the mapping cannot be deleted in the statement of Theorem 9.

Example 10 Consider the partial metric $p : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ given by $p(x, y) = \max\{x, y\}$. It is clear that the partial metric space $(\mathbb{R}^+, p)$ is complete and $p^s(x, y) = |y - x|$ for all $x, y \in \mathbb{R}^+$. Moreover, we have that $x \le_p y \Leftrightarrow y \le x$, where $\le$ stands for the usual order on $\mathbb{R}^+$. Define the mapping $f : \mathbb{R}^+ \to \mathbb{R}^+$ by

$$f(x) = \begin{cases} 2x + 1 & \text{if } x > 1, \\ x + 1 & \text{if } x \le 1 \end{cases}$$

for all $x \in \mathbb{R}^+$. Then we have that f is monotone and that $1 = f(0) \leq_p 0$. It is easy to check that f is not p^s-continuous. Moreover, it is clear that f has no fixed points.

Example 11 Let $X = \{0, 1\}$. Consider the restriction of the partial metric p defined in Example 10 to the set X. Denote the aforesaid restriction by p again. Of course the partial metric space (X, p) is complete. Now, define the mapping $f : X \to X$ by $f(0) = 1$ and $f(1) = 0$. It follows easily that f is p^s-continuous. Clearly, $1 = f(0) \leq_p 0$. Moreover, f is not monotone since $1 \leq_p 0$ and $0 = f(1) \not\leq_p f(0) = 1$. Furthermore, f has no fixed points.

3 Asymptotic complexity analysis of algorithms

3.1 Preliminaries

In computer science, the complexity analysis of an algorithm is based on determining mathematically the quantity of resources needed by the algorithm to solve the problem for which it has been designed. A typical resource, playing a central role in complexity analysis, is the execution time or running time of computing. Since there are often many algorithms to solve the same problem, one objective of the complexity analysis is to assess which of them is faster when large inputs are considered. To this end, it is necessary to compare their running time of computing. This is usually done by means of the asymptotic analysis in which the running time of an algorithm is denoted by a function $T : \mathbb{N} \to (0, \infty]$ in such a way that $T(n)$ represents the time taken by the algorithm to solve the problem under consideration when the input of the algorithm is of size n. Let us denote by $\mathcal{RT}$ the set formed by all functions from $\mathbb{N}$ into $(0, \infty]$. Of course the running time of an algorithm does not only depend on the input size n, but it depends also on the particular input of the size n (and the distribution of the data). Thus the running time of an algorithm is different when the algorithm processes certain instances of input data of the same size n. As a consequence, for the purpose of size-based comparisons, it is necessary to distinguish three possible behaviors in the complexity analysis of algorithms. These are the so-called best case, the worst case and the average case. The best case and the worst case for an input of size n are defined by the minimum and the maximum running time of computing over all inputs of the size n, respectively. The average case for an input of size n is defined by the expected value or average over all inputs of size n.

In general, to determine exactly the function which describes the running time of computing of an algorithm is an arduous task. However, in most situations it is useful to know the running time of computing of an algorithm in an 'approximate' way rather than in an exact one. For this reason, the asymptotic complexity analysis of algorithms is focused on obtaining the 'approximate' running time of computing of an algorithm, and this is done by means of the Θ-notation. Let us recall how the Θ-notation allows to achieve such a goal.

Let $f \in \mathcal{RT}$ denote the running time of computing of an algorithm, then the statement $f \in \mathcal{O}(g)$, where $g \in \mathcal{RT}$, means that there exist $n_0 \in \mathbb{N}$ and $c \in \mathbb{R}^+$ such that $f(n) \leq_\infty cg(n)$ for all $n \in \mathbb{N}$ such that $n_0 \leq n$ ($\leq_\infty$ stands for the usual order on the extended real line). So, the function g gives an asymptotic upper bound of the running time f and, thus, an 'approximate' information of the running time of the algorithm. Following the standard notation, when g is an asymptotic upper bound of f, we write $f \in \mathcal{O}(g)$.

Sometimes in the analysis of the complexity of an algorithm, it is useful to assess an asymptotic lower bound of the running time of computing. In this case, the Ω-notation

plays a central role. Thus the statement $f \in \Omega(g)$ means that there exist $n_0 \in \mathbb{N}$ and $c > 0$ such that $cg(n) \leq_\infty f(n)$ for all $n \in \mathbb{N}$ with $n_0 \leq n$. Of course, and similarly to the $\mathcal{O}$-notation case, when the time taken by the algorithm to solve the problem f is unknown, the function g yields an 'approximate' information of the running time of the algorithm in the sense that the algorithm takes a time to solve the problem bounded below by g.

It is clear that the best situation, when the complexity of an algorithm is discussed, matches up with the case in which we can find a function $g \in \mathcal{RT}$ in such a way that the running time f satisfies the condition $f \in \mathcal{O}(g) \cap \Omega(g)$, denoted by $f \in \Theta(g)$, because, in this case, we obtain a 'tight' asymptotic bound of f and, thus, a total asymptotic information about the time taken by the algorithm to solve the problem under consideration. From now on, we will say that f belongs to the asymptotic complexity class of g whenever $f \in \Theta(g)$.

In the light of the preceding discussion, to determine, from an asymptotic complexity analysis viewpoint, the running time of an algorithm consists of obtaining its asymptotic complexity class.

For a fuller treatment of asymptotic complexity analysis of algorithms, we refer the reader to [44, 45].

3.2 The complexity space approach

In 1995, Schellekens introduced a topological foundation for the asymptotic complexity analysis of algorithms [46]. The aforementioned foundation is based on the notions of quasi-metric and complexity space.

Let us recall that, following [47], a quasi-metric on a nonempty set X is a function $d : X \times X \to \mathbb{R}^+$ such that for all $x, y, z \in X$:

(i) $d(x, y) = d(y, x) = 0 \Leftrightarrow x = y$;

(ii) $d(x, y) \leq d(x, z) + d(z, y)$.

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

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

Given a quasi-metric d on X, the function $d^s : X \times X \to \mathbb{R}^+$ defined by $d^s(x, y) = \max\{d(x, y), d(y, x)\}$ is a metric.

A quasi-metric space (X, d) is called bicomplete if the metric space (X, d^s) is complete.

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

$$\mathcal{C} = \left\{ f \in \mathcal{RT} : \sum_{n=1}^{\infty} 2^{-n} \frac{1}{f(n)} < \infty \right\}$$

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

$$d_\mathcal{C}(f, g) = \sum_{n=1}^{\infty} 2^{-n} \max\left\{ \frac{1}{g(n)} - \frac{1}{f(n)}, 0 \right\}.$$

Obviously, we adopt the convention that $\frac{1}{\infty} = 0$.

According to [46], from a complexity analysis point of view, it is possible to associate a function of $\mathcal{C}$ with each algorithm in such a way that such a function represents, as a

function of the size of the input data, the running time of computing of the algorithm. Because of this, the elements of $\mathcal{C}$ are called complexity functions. Moreover, 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 a complexity function f by any program Q with a complexity function g. Therefore, if $f \neq g$, the condition $d_{\mathcal{C}}(f, g) = 0$ can be read as f is 'at least as efficient' as g on all inputs. In fact, we have that $d_{\mathcal{C}}(f, g) = 0 \Leftrightarrow f(n) \leq_{\infty} g(n)$ for all $n \in \mathbb{N}$ and, thus, the fact that $d_{\mathcal{C}}(f, g) = 0$ ($d_{\mathcal{C}}(g, f) = 0$) implies that $f \in \mathcal{O}(g)$ ($f \in \Omega(g)$).

The applicability of the complexity space to the asymptotic complexity analysis of algorithms was illustrated by Schellekens in [46].

In particular, he introduced a method, based on a fixed point theorem for functionals defined on the complexity space into itself, to provide the asymptotic upper bound of those algorithms whose running time of computing satisfies a recurrence equation of Divide and Conquer type. Let us recall the aforenamed method.

A Divide and Conquer algorithm solves a problem of size n ($n \in \mathbb{N}$) splitting it into a subproblems of size $\frac{n}{b}$, for some constants a, b with $a, b \in \mathbb{N}$ and $a, b > 1$, and solving them separately by the same algorithm. After obtaining the solution of the subproblems, the algorithm combines all subproblem solutions to give a global solution to the original problem. The recursive structure of a Divide and Conquer algorithm leads to a recurrence equation for the running time of computing. In many cases the running time of a Divide and Conquer algorithm is the solution to the Divide and Conquer recurrence equation of the form

$$
T(n) = \begin{cases} c & \text{if } n = 1, \\ aT(\frac{n}{b}) + h(n) & \text{if } n \in \mathbb{N}_b, \end{cases}
\tag{1}
$$

where $\mathbb{N}_b = \{b^k : k \in \mathbb{N}\}$, $c > 0$ denotes the complexity on the base case (*i.e.*, the problem size is small enough and the solution takes constant time), and $h(n)$ represents the time taken by the algorithm in order to divide the original problem into a subproblems and to combine all subproblems solutions into a unique one ($h \in \mathcal{C}$ and $h(n) <_{\infty} \infty$ for all $n \in \mathbb{N}$).

Notice that for Divide and Conquer algorithms with running time satisfying the recurrence equation (1), it is typically sufficient to obtain the complexity on inputs of size n, where n ranges over the set $\mathbb{N}_b$ [45, 46].

Typical examples of algorithms whose running time of computing can be obtained by means of the recurrence (1) are Quicksort (best case behavior) and Mergesort (all behaviors) (see [45] for a detailed discussion about the both aforesaid algorithms).

The method introduced by Schellekens allows to show that the Divide and Conquer recurrence equation (1) has a unique solution and, in addition, to provide an upper asymptotic complexity bound of such a solution. To this end, denote by $\mathcal{C}_{b,c}$ the subset of $\mathcal{C}$ given by

$$
\mathcal{C}_{b,c} = \left\{ f \in \mathcal{C} : f(1) = c \text{ and } f(n) = \infty \text{ for all } n \in \mathbb{N} \backslash \mathbb{N}_b \text{ with } n > 1 \right\},
$$

and by $d_{\mathcal{C}}|_{\mathcal{C}_{b,c}}$ the restriction of $d_{\mathcal{C}}$ to $\mathcal{C}_{b,c}$.

We have that the quasi-metric space $(\mathcal{C}_{b,c}, d_{\mathcal{C}}|_{\mathcal{C}_{b,c}})$ is bicomplete since the quasi-metric space $(\mathcal{C}, d_{\mathcal{C}})$ is bicomplete [48] and the set $\mathcal{C}_{b,c}$ is closed in $(\mathcal{C}, d_{\mathcal{C}}^s)$.

Next we associate a functional $\Phi_T : \mathcal{C}_{b,c} \to \mathcal{C}_{b,c}$ with the recurrence equation (1) defined as follows:

$$
\Phi_T(f)(n) = \begin{cases} c & \text{if } n = 1, \\ \infty & \text{if } n \notin \mathbb{N}_b \text{ and } n > 1, \\ af\left(\frac{n}{b}\right) + h(n) & \text{otherwise.} \end{cases} \tag{2}
$$

Of course a complexity function in $\mathcal{C}_{b,c}$ is a solution to the recurrence equation (1) if and only if it is a fixed point of the functional Φ_T. It was proved in [46] that

$$
d_{\mathcal{C}}|_{\mathcal{C}_{b,c}}\left(\Phi_T(f), \Phi_T(g)\right) \leq \frac{1}{a} d_{\mathcal{C}}|_{\mathcal{C}_{b,c}}(f, g) \tag{3}
$$

for all $f, g \in \mathcal{C}_{b,c}$. So, by Banach's fixed point theorem for metric spaces, we deduce that the functional $\Phi_T : \mathcal{C}_{b,c} \to \mathcal{C}_{b,c}$ has a unique fixed point and, thus, the recurrence equation (1) has a unique solution.

In order to obtain the asymptotic upper bound of the solution to the recurrence equation (1), Schellekens introduced a special class of functionals known as improvers.

Let $C \subseteq \mathcal{C}$, a functional $\Phi : C \to C$ is called an improver with respect to a function $f \in C$ provided that Φ is monotone and that $\Phi(f)(n) \leq_\infty f(n)$ for all $n \in \mathbb{N}$.

Taking into account the exposed facts, the following result was stated in [46].

Theorem 12 *The Divide and Conquer recurrence equation* (1) *has a unique solution* f_T *in* $\mathcal{C}_{b,c}$. *Moreover, if the monotone functional* Φ_T *associated to* (1), *and given by* (2), *is an improver with respect to some function* $g \in \mathcal{C}_{b,c}$, *then the solution to the recurrence equation satisfies that* $f_T \in \mathcal{O}(g)$.

In [46] Schellekens discussed the complexity class of Mergesort in order to illustrate the utility of Theorem 12. The Mergesort running time of computing (average case behavior) satisfies the following particular case of the recurrence equation (1):

$$
T(n) = \begin{cases} c & \text{if } n = 1, \\ 2T\left(\frac{n}{2}\right) + \frac{n}{2} & \text{if } n \in \mathbb{N}_2. \end{cases} \tag{4}
$$

Of course Theorem 12 provides that the recurrence equation (4) has a unique solution $f_T^M \in \mathcal{C}_{2,c}$. In addition, it is not hard to check that the functional Φ_T given by (2) and induced by the recurrence equation (4) is an improver with respect to the complexity function $g_k \in \mathcal{C}_{2,c}$, with $k > 0$, if and only if $k \geq \frac{1}{2}$, where

$$
g_k(n) = \begin{cases} c & \text{if } n = 1, \\ kn \log_2(n) & \text{if } n \in \mathbb{N}_2. \end{cases}
$$

Therefore, by Theorem 12, we conclude that $f_T^M \in \mathcal{O}(g_{\frac{1}{2}})$. Hence the complexity function $g_{\frac{1}{2}}$, or equivalently $\mathcal{O}(n \log_2(n))$, gives an asymptotic upper bound of the running time of computing of the aforenamed algorithm.

Furthermore, it must be stressed that Schellekens provided the asymptotic lower bound, and thus the asymptotic complexity class, of the running time of Mergesort (average

case behavior) in [46]. Concretely, it was obtained that such a running time of computing belongs to $\Omega(n\log_2(n))$. Nevertheless, the asymptotic lower bound was obtained applying standard arguments that are not based on the use of fixed point techniques. So, Schellekens proved that Mergesort running time (average case behavior) belongs to the asymptotic complexity class $\Theta(n\log_2(n))$, but the fixed point technique was used only to provide the asymptotic upper bound.

4 Asymptotic complexity analysis of algorithms and partial metric spaces

Motivated by the usefulness of partial metric spaces in program verification, we wonder if this kind of generalized metric spaces are also useful in asymptotic complexity analysis in the spirit of Schellekens. Of course it seems natural to attempt to apply Theorem 3 (in Section 1) in order to obtain a fixed point technique for asymptotic complexity analysis of algorithms based on the use of partial metric spaces. However, the following reasoning shows that the aforementioned result cannot be applied for this purpose. To this end, let us recall some additional and useful concepts about partial metrics and complexity spaces.

Following [49] the set $\mathcal{C}$ can be endowed with a partial metric $p_\mathcal{C}$ defined for all $f,g \in \mathcal{C}$ by

$$p_\mathcal{C}(f,g) = \sum_{n=1}^{\infty} 2^{-n} \max\left\{ \frac{1}{f(n)}, \frac{1}{g(n)} \right\}.$$

Obviously, we make again the assumption that $\frac{1}{\infty} = 0$.

Notice that although, in principle, several partial metrics could be defined on $\mathcal{C}$ with the aim of developing a mathematical foundation of asymptotic complexity analysis by means of partial metric spaces, it seems reasonable to consider $p_\mathcal{C}$ as a (partial metric) complexity distance because of the following reasons.

On the one hand, it allows to retrieve the Schellekens (quasi-metric) complexity distance $d_\mathcal{C}$. Indeed, the following correspondences between quasi-metric and partial metric spaces was stated in [3].

Proposition 13 *If (X,p) is a partial metric space, then the function $d_p : X \times X \to \mathbb{R}^+$ defined by $d_p(x,y) = p(x,y) - p(x,x)$ is a quasi-metric such that $\mathcal{T}(p) = \mathcal{T}(d_p)$.*

In the light of Proposition 13, we have that the partial metric $p_\mathcal{C}$ induces the quasi-metric $d_{p_\mathcal{C}}$ on $\mathcal{C}$ given by

$$d_{p_\mathcal{C}}(f,g) = p_\mathcal{C}(f,g) - p_\mathcal{C}(f,f)$$

for all $f,g \in \mathcal{C}$. Moreover, it is not hard to see that

$$p_\mathcal{C}(f,g) - p_\mathcal{C}(f,f) = d_\mathcal{C}(f,g),$$

$f,g \in \mathcal{C}$ and, thus, that $d_{p_\mathcal{C}}(f,g) = d_\mathcal{C}(f,g)$ for all $f,g \in \mathcal{C}$.

On the other hand, the order induced by the partial metric $p_\mathcal{C}$, $\leq_{p_\mathcal{C}}$, is exactly the pointwise order relation defined on $\mathcal{C}$. Indeed, given $f,g \in \mathcal{C}$, we have that

$$f \leq_{p_\mathcal{C}} g \quad \Leftrightarrow \quad p_\mathcal{C}(f,g) = p_\mathcal{C}(f,f) \quad \Leftrightarrow \quad f(n) \leq_\infty g(n)$$

for all $n \in \mathbb{N}$. Whence we obtain that $f \leq_{p_\mathcal{C}} g$ implies that $f \in \mathcal{O}(g)$. Observe that the last implication recovers the information yielded by the condition $d_\mathcal{C}(f,g) = 0$ in the Schellekens approach.

Furthermore, the completeness of the partial metric space $(\mathcal{C}, p_\mathcal{C})$ is guaranteed by the bicompleteness of the complexity space $(\mathcal{C}, d_\mathcal{C})$ and the following result which was proved in [49].

Proposition 14 *If (X,p) is a partial metric space, then the below assertions are equivalent*:

(1) (X,p) *is complete.*

(2) (X,d_p) *is bicomplete.*

Since the partial metric space $(\mathcal{C}, p_\mathcal{C})$ is complete, so is the partial metric $(\mathcal{C}_{b,c}, p_\mathcal{C}|_{\mathcal{C}_{b,c}})$.

Now, for the purpose of applying Theorem 3 to discuss the complexity of Divide and Conquer algorithms whose running time of computing satisfy the Divide and Conquer recurrence equation (1), suppose that there exists $s \in [0,1[$ such that

$$p_\mathcal{C}|_{\mathcal{C}_{b,c}}\big(\Phi_T(f), \Phi_T(g)\big) \leq s p_\mathcal{C}|_{\mathcal{C}_{b,c}}(f,g)$$

for all $f,g \in \mathcal{C}_{b,c}$, where Φ_T is the functional given by (2).

Of course, we only consider the case of $s \in {]0,1[}$ because it is evident that the case $s = 0$ gives a contradiction.

Take $f,g \in \mathcal{C}_{b,c}$ such that $f(n) = 2c$ and $g(n) = 2(c+1)$ for all $n \in \mathbb{N}_b$. It is clear that

$$p_\mathcal{C}|_{\mathcal{C}_{b,c}}\big(\Phi_T(f), \Phi_T(g)\big) = \frac{1}{2c} + \sum_{n=2}^{\infty} 2^{-b^n} \max\left\{ \frac{1}{2ac + h(b^n)}, \frac{1}{2a(c+1) + h(b^n)} \right\}.$$

Moreover,

$$p_\mathcal{C}|_{\mathcal{C}_{b,c}}(f,g) = \frac{1}{2c} + \sum_{n=2}^{\infty} 2^{-b^n} \max\left\{ \frac{1}{f(b^n)}, \frac{1}{g(b^n)} \right\} \leq \sum_{n=1}^{\infty} 2^{-n} \max\left\{ \frac{1}{2c}, \frac{1}{2(c+1)} \right\}$$

$$= \frac{1}{2c}.$$

Applying our hypothesis, we obtain that

$$\frac{1}{2c} = p_\mathcal{C}|_{\mathcal{C}_{b,c}}\big(\Phi_T(f), \Phi_T(g)\big) \leq s p_\mathcal{C}|_{\mathcal{C}_{b,c}}(f,g) = s\frac{1}{2c}.$$

As a result we deduce that $1 \leq s < 1$, which is a contradiction.

Consequently, when we consider the partial metric $p_\mathcal{C}$ as a complexity distance instead the original quasi-metric $d_\mathcal{C}$, Theorem 3 cannot be applied to the asymptotic complexity analysis of those algorithms whose running time of computing leads to the Divide and Conquer recurrence equation (1).

We want to point out that the above reasoning was first introduced in [50] in order to show the impossibility of using Theorem 3 to analyze the Divide and Conquer recurrence equations. As a consequence, in the aforesaid reference a fixed point technique, which differs from the technique that we will introduce in the remainder of this section, was introduced to discuss the complexity of algorithms via the use of partial quasi-metrics, and not partial metrics, and a few aspects of language theory.

4.1 The fixed point technique for asymptotic complexity analysis based on partial metric spaces

Inspired by the impossibility of developing a fixed point technique for the asymptotic complexity analysis of algorithms based on the use of Theorem 3, we present a new fixed point technique that respects the spirit of the original Schellekens technique and whose foundation lies in the use of Theorems 4 and 9 in Section 2.

In the complexity analysis of algorithms, Divide and Conquer algorithms belong to the wider class of recursive algorithms. In many cases the running time of imputing of the latter is the solution to the following recurrence equation:

$$
T(n) = \begin{cases} c & \text{if } n = 1, \\ aT(n-1) + h(n) & \text{if } n \geq 2, \end{cases} \tag{5}
$$

where $c > 0$, $a \geq 1$ and $h \in \mathcal{C}$ with $h(n) <_{\infty} \infty$ for all $n \in \mathbb{N}$.

Observe that the discussion of the asymptotic complexity of the Divide and Conquer algorithms introduced in Section 3.2 can be carried out from the recurrence equation (5). In fact, as discussed in Section 3.2, the running time of computing of the aforesaid algorithms is the solution to the recurrence equation

$$
T(n) = \begin{cases} c & \text{if } n = 1, \\ aT(\frac{n}{b}) + h(n) & \text{if } n \in \mathbb{N}_b. \end{cases}
$$

Clearly, the preceding Divide and Conquer recurrence equation can be retrieved as a particular case of the recurrence equation (5). Indeed, the Divide and Conquer recurrence equation can be transformed into the following one:

$$
S(m) = \begin{cases} c & \text{if } m = 1, \\ aS(m-1) + r(m) & \text{if } m > 1, \end{cases} \tag{6}
$$

where $S(m) = T(b^{m-1})$ and $r(m) = h(b^{m-1})$ for all $m \in \mathbb{N}$.

The remainder of this section is devoted to introducing, by means of the partial metric space $(\mathcal{C}, p_{\mathcal{C}})$, a new fixed point technique in the spirit of Schellekens for yielding the asymptotic complexity class of those recursive algorithms whose running time is the solution to the recurrence equation (5).

4.1.1 The existence and uniqueness of solution

Consider the subset $\mathcal{C}_c$ of $\mathcal{C}$ given by

$$
\mathcal{C}_c = \{f \in \mathcal{C} : f(1) = c\}.
$$

Define the functional $\Psi_T : \mathcal{C}_c \to \mathcal{C}_c$ by

$$
\Psi_T(f)(n) = \begin{cases} c & \text{if } n = 1, \\ af(n-1) + h(n) & \text{if } n \geq 2 \end{cases} \tag{7}
$$

for all $f \in \mathcal{C}_c$. It is clear that a complexity function in $\mathcal{C}_c$ is a solution to the recurrence equation (5) if and only if it is a fixed point of the functional Ψ_T.

In order prove the announced existence and uniqueness of solution to the recurrence equation (5), we will need the following auxiliary result.

Lemma 15 *Let $\Psi_T : \mathcal{C}_c \to \mathcal{C}_c$ be the functional given by (7) and let $f_{c,\infty} \in \mathcal{C}_c$ be the complexity function given by*

$$f_{c,\infty}(n) = \begin{cases} c & \text{if } n = 1, \\ \infty & \text{if } n \geq 2. \end{cases}$$

Then the following statements hold:

(1) Ψ_T *is monotone.*

(2) Ψ_T *is continuous and p^s-continuous.*

(3) $\Psi_T(f_{c,\infty}) \leq_{p_{\mathcal{C}_c}} f_{c,\infty}$.

Proof (1) Consider $f, g \in \mathcal{C}_c$ such that $f \leq_{p_{\mathcal{C}_c}} g$. Then $f(n) \leq_{\infty} g(n)$ for all $n \in \mathbb{N}$. Thus we have that $\Psi_T(f)(1) = \Psi_T(g)(1) = c$ and

$$\Psi_T(f)(n) = af(n-1) + h(n) \leq_{\infty} ag(n-1) + h(n) = \Psi_T(g)(n)$$

for all $n \in \mathbb{N}$ with $n \geq 2$. It follows that $\Psi_T(f) \leq_{p_{\mathcal{C}_c}} \Psi_T(g)$.

(2) First of all we prove that Ψ_T is continuous from $(X, \mathcal{T}(p))$ into itself. Indeed, for the purpose of contradiction, assume that there exists a sequence $(f_k)_{k \in \mathbb{N}}$ in $\mathcal{C}_c$ which converges to $f \in \mathcal{C}_c$ with respect to $\mathcal{T}(p_{\mathcal{C}_c})$ and, in addition, the sequence $(\Psi_T(f_k))_{k \in \mathbb{N}}$ does not converge to $\Psi_{T(f)}$ with respect to $\mathcal{T}(p_{\mathcal{C}_c})$. Then we have that there exists $\varepsilon > 0$ such that for each $n \in \mathbb{N}$ there is $m \geq n$ with

$$\varepsilon \leq p_{\mathcal{C}_c}\big(\Psi_T(f_m), \Psi_T(f)\big) - p_{\mathcal{C}_c}\big(\Psi_T(f), \Psi_T(f)\big)$$

and

$$p_{\mathcal{C}_c}(f, f_m) - p_{\mathcal{C}_c}(f, f) < \varepsilon.$$

Then we have that

$$\varepsilon \leq \sum_{n=2}^{\infty} 2^{-n} \max\left\{ \frac{1}{af_m(n-1) + h(n)}, \frac{1}{af(n-1) + h(n)} \right\} - \sum_{n=2}^{\infty} 2^{-n} \frac{1}{af(n-1) + h(n)}$$

$$\leq \frac{1}{2a}\left(\sum_{n=1}^{\infty} 2^{-n} \max\left\{ \frac{1}{f_m(n)}, \frac{1}{f(n)} \right\} - \frac{1}{f(n)} \right)$$

$$= \frac{1}{2a}\big(p_{\mathcal{C}_c}(f, f_m) - p_{\mathcal{C}_c}(f, f)\big) < \varepsilon,$$

which is a contradiction. It follows that Ψ_T is continuous from $(\mathcal{C}_c, \mathcal{T}(p_{\mathcal{C}_c}))$ into itself.

Next we prove that Ψ_T is continuous from $(\mathcal{C}_c, \mathcal{T}(p^s_{\mathcal{C}_c}))$ into itself. To this end, we assume that there exists a sequence $(f_k)_{k \in \mathbb{N}}$ in $\mathcal{C}_c$ which converges to $f \in \mathcal{C}_c$ with respect to $\mathcal{T}(p^s_{\mathcal{C}_c})$

and, in addition, the sequence $(\Psi_T(f_k))_{k\in\mathbb{N}}$ does not converge to $\Psi_T(f)$ with respect to $\mathcal{T}(p^s_{\mathcal{C}_c})$. By the continuity of Ψ_T, we deduce that necessarily there exists $\varepsilon > 0$ such that for each $n \in \mathbb{N}$ there is $m \geq n$ with

$$\varepsilon \leq p_{\mathcal{C}_c}\big(\Psi_T(f_m), \Psi_T(f)\big) - p_{\mathcal{C}_c}\big(\Psi_T(f_m), \Psi_T(f_m)\big)$$

and

$$p^s_{\mathcal{C}_c}(f, f_m) < \varepsilon.$$

If follows that

$$\varepsilon \leq \sum_{n=2}^{\infty} 2^{-n} \max\left\{ \frac{1}{af_m(n-1)+h(n)}, \frac{1}{af(n-1)+h(n)} \right\} - \sum_{n=2}^{\infty} 2^{-n} \frac{1}{af_m(n-1)+h(n)}$$

$$\leq \frac{1}{2a}\left(\sum_{n=1}^{\infty} 2^{-n} \max\left\{ \frac{1}{f_m(n)}, \frac{1}{f(n)} \right\} - \frac{1}{f_m(n)} \right)$$

$$= \frac{1}{2a}\big(p_{\mathcal{C}_c}(f, f_m) - p_{\mathcal{C}_c}(f_m, f_m) \big)$$

$$\leq \frac{1}{2a} p^s_{\mathcal{C}_c}(f, f_m) < \varepsilon,$$

which is a contradiction. Whence we conclude that Ψ_T is continuous from $(\mathcal{C}_c, \mathcal{T}(p^s_{\mathcal{C}_c}))$ into itself, *i.e.*, Ψ_T is p^s-continuous.

(3) It is clear that $\Psi_T(f_{c,\infty})(1) = c = f_{c,\infty}(1)$. Moreover,

$$\Psi_T(f_{c,\infty})(2) = af_{c,\infty}(1) + h(2) = ac + h(2) \leq_\infty f_{c,\infty}(2) = \infty.$$

Finally,

$$\Psi_T(f_{c,\infty})(n) = \infty = f_{c,\infty}(n)$$

for all $n \in \mathbb{N}$ with $n > 2$. Consequently, we obtain that $\Psi_T(f_{c,\infty})(n) \leq_\infty f_{c,\infty}(n)$ for all $n \in \mathbb{N}$. Therefore $\Psi_T(f_{c,\infty}) \leq_{p_{\mathcal{C}_c}} f_{c,\infty}$. □

Remark 16 It should be stressed that, given a partial metric space (X, p), the continuous and p^s-continuous mappings $f : X \to X$ are called properly continuous in [5]. So, the functional Ψ_T, by assertion (2) in the statement of Lemma 15, is properly continuous.

According to [51], the quasi-metric space $(\mathcal{C}_c, d_{\mathcal{C}}|_{\mathcal{C}_c})$ is bicomplete. Hence, by Proposition 14, we have that the partial metric space $(\mathcal{C}_c, p_{\mathcal{C}_c})$ is complete.

Theorem 17 *The recurrence equation* (5) *has a unique solution f_T in $\mathcal{C}_c$.*

Proof The completeness of $(\mathcal{C}_c, p_{\mathcal{C}_c})$ and Lemma 15 provide the conditions in the statement of Theorem 9. So, we have that Ψ_T has a fixed point in $\downarrow f_{c,\infty}$. Since $\downarrow f_{c,\infty} \cap \mathcal{C}_c = \mathcal{C}_c$, we obtain the existence of a fixed point of f in $\mathcal{C}_c$.

It remains to prove the uniqueness of the fixed point. To this end, assume that $f_T, g_T \in \mathcal{C}_c$ are solutions to the recurrence equation (5). Next we will prove by induction over n that $f_T = g_T$. Since f_T, g_T are solutions to the recurrence equation (5), we have that they are fixed points of the functional Ψ_T, *i.e.*, $\Psi_T(f_T) = \Psi_T(g_T)$. Hence we have that

$$f_T(1) = \Psi_T(f_T)(1) = c = \Psi_T(g_T)(1) = g_T(1).$$

Now assume that $\Psi(f_T)(n) = \Psi(g_T)(n)$ with $n > 1$. Then

$$f_T(n+1) = \Psi_T(f_T)(n+1) = af_T(n) + h(n+1)$$
$$= ag_T(n) + h(n+1) = \Psi_T(g_T)(n+1) = g_T(n+1).$$

Consequently, $f_T = g_T$ and, thus, Ψ_T has a unique fixed point in $\mathcal{C}_c$. Therefore the recurrence equation (5) has a unique solution in $\mathcal{C}_c$. □

4.1.2 The asymptotic complexity class of the solution

In the next result, we obtain the announced method to provide the complexity class of an algorithm whose running time of computing satisfies the recurrence equation (5). To this end, let us recall that, given $C \subseteq \mathcal{C}$, a monotone functional $\Phi : C \to C$ is called an improver with respect to a function $f \in C$ provided that $\Phi(f) \leq_{p_{\mathcal{C}_c}} f$ (see Section 3.2). Furthermore, on account of [51], a monotone functional $\Phi : C \to C$ is said to be a worsener with respect to a function $f \in C$ provided that $f \leq_{p_{\mathcal{C}_c}} \Phi(f)$.

Observe that the computational meaning of the improvers, as discussed in [46], can be interpreted as that this type of functionals correspond to a transformation on programs in such a way that the iterative applications of the transformation yield, from a complexity point of view, an improved program at each step of the iteration. Similarly, the worseners can be interpreted as those functionals which correspond to a transformation on programs in such a way that the iterative applications of the transformation yield a worsened, from a complexity point of view, program at each step of the iteration.

Theorem 18 *Let $f_T \in \mathcal{C}_c$ be the unique solution to the recurrence equation* (5)*. Then the following assertions hold*:

(1) *If the functional Ψ_T associated to* (5)*, and given by* (7)*, is a worsener with respect to some complexity function $g \in \mathcal{C}_c$, then $f_T \in \Omega(g)$.*

(2) *If the functional Ψ_T associated to* (5)*, and given by* (7)*, is an improver with respect to some complexity function $g \in \mathcal{C}_c$, then $f_T \in \mathcal{O}(g)$.*

Proof (1) Suppose that there exists $g \in \mathcal{C}_c$ such that $\Psi_T(g) \leq_{p_{\mathcal{C}_c}} g$. Then, by Corollary 8, we deduce the existence of a fixed point g_T of Ψ_T such that $g_T \in \uparrow g$, *i.e.*, $g_T \leq_{p_{\mathcal{C}_c}} g$ and, thus, $g_T(n) \leq_\infty g(n)$ for all $n \in \mathbb{N}$. So, $g_T \in \Omega(g)$. Since f_T is the unique fixed point of Ψ_T in $\mathcal{C}_c$, we deduce that $f_T = g_T$ and, hence, that $f_T \in \Omega(g)$.

(2) Assume that there exists $g \in \mathcal{C}_c$ such that $g \leq_{p_{\mathcal{C}_c}} \Psi_T(g)$. Then Theorem 9 gives the existence of a fixed point g_T of Ψ_T such that $g_T \in \downarrow g$, *i.e.*, $g \leq_{p_{\mathcal{C}_c}} g_T$ and hence $g(n) \leq_\infty g_T(n)$ for all $n \in \mathbb{N}$. So, $g_T \in \mathcal{O}(g)$. The uniqueness of a fixed point of Ψ_T in $\mathcal{C}_c$ allows us to deduce that $f_T = g_T$ and, thus, that $f_T \in \mathcal{O}(g)$. □

Remark 19 Notice that, indeed, Theorem 18 yields the complexity class of algorithms whose running time of computing satisfies the recurrence equation (5) because when there exist $l \in \mathcal{C}_c$, $r, t > 0$ and $n_0 \in \mathbb{N}$ such that $g(n) = rl(n)$ and $h = tl(n)$ for all $n > n_0$ and, besides, Ψ_T is an improver and a worsener with respect to g and h, respectively, then $f_T \in \Theta(l)$.

4.2 Analyzing the running time computing of two examples

The aim of this section is twofold. On the one hand, we show that the developed method in Section 4.1 is useful to analyze the asymptotic complexity of recursive algorithms. On the other hand, in order to validate the new results, we will retrieve, by means of their application, the complexity class of two well-known algorithms in the literature.

Typical examples of algorithms whose running time of computing is the solution to the recurrence equation (5) are Quicksort (worst case behavior) and Hanoi.

In particular, the running time of computing of Quicksort is the solution to the recurrence equation given exactly as follows:

$$
T(n) = \begin{cases} c & \text{if } n = 1, \\ T(n-1) + jn & \text{if } n \geq 2, \end{cases} \tag{8}
$$

with $j > 0$ and where c is the time taken by the algorithm in the base case.

Regarding Hanoi, under the uniform cost criterion assumption, its running time of computing is the solution to the recurrence equation given as follows:

$$
T(n) = \begin{cases} c & \text{if } n = 1, \\ 2T(n-1) + d & \text{if } n \geq 2, \end{cases} \tag{9}
$$

with $c, d > 0$ and where, again, c represents the time taken by the algorithm to solve the base case. Note that it does not make sense to distinguish three possible running time behaviors for Hanoi since the distribution of the input data is always the same for each size n.

For a deeper discussion about Quicksort and Hanoi, we refer the reader to [44, 45] and [44, 52], respectively.

Next we discuss the running time of computing of the aforesaid algorithms through our results.

Corollary 20 *The running time of computing of Quicksort* (*worst case behavior*) *is in the complexity class* $\Theta(n^2)$.

Proof The running time of computing of Quicksort (worst case behavior) is provided by the solution to the recurrence equation (8). Theorem 17 guarantees the existence and uniqueness of such a solution. Denote it by f_T^Q.

Consider the functional Ψ_T^Q given in (7) and induced by the recurrence equation (8), *i.e.*,

$$
\Psi_T^Q(f)(n) = \begin{cases} c & \text{if } n = 1, \\ f(n-1) + jn & \text{if } n \geq 2 \end{cases} \tag{10}
$$

for all $f \in \mathcal{C}_c$.

Next we provide the asymptotic upper bound of f_T^Q. With this aim, define the complexity function h_r as follows:

$$h_r(n) = \begin{cases} c & \text{if } n = 1, \\ rn^2 & \text{if } n \geq 2, \end{cases}$$

where $r > 0$. Then it is not hard to see that Ψ_T is an improver with respect to $h_r \in \mathcal{C}_c$ (*i.e.*, $\Psi_T^Q(h_r) \leq_{p_{\mathcal{C}_c}} h_r$) if and only if $r \geq \max\{\frac{3j}{5}, \frac{c}{4} + \frac{j}{2}\}$. It follows, by statement (2) in Theorem 18, that $f_T^Q \in \mathcal{O}(h_{\max\{\frac{3j}{5}, \frac{c}{4} + \frac{j}{2}\}})$.

In order to provide the asymptotic complexity class, it remains to yield an asymptotic lower bound of f_T^Q. Now it is a routine to check that Ψ_T^Q is a worsener with respect to the complexity function h_s (*i.e.*, $h_s \leq_{p_{\mathcal{C}_c}} \Psi_T^Q(h_s)$) if and only if $s \leq \min\{\frac{j}{2}, \frac{c}{4} + \frac{j}{2}\}$, whence we deduce, by statement (1) in Theorem 18, that $f_T^Q \in \Omega(h_{\min\{\frac{j}{2}, \frac{c}{4} + \frac{j}{2}\}})$.

Therefore we obtain that $f_T^Q \in \mathcal{O}(h_{\max\{\frac{3j}{5}, \frac{c}{4} + \frac{j}{2}\}}) \cap \Omega(h_{\min\{\frac{j}{2}, \frac{c}{4} + \frac{j}{2}\}})$. Whence we deduce, by Remark 19, that $f_T^Q \in \Theta(n^2)$, which is in accordance with the Quicksort (worst case behavior) asymptotic complexity class that can be found in the literature [44, 45]. □

Corollary 21 *The running time of computing of Hanoi, under the uniform const criterion, is in the complexity class* $\Theta(2^n)$.

Proof The running time of computing of Hanoi, under the uniform const criterion, is provided by the solution to the recurrence equation (9). Theorem 17 guarantees the existence and uniqueness of such a solution. Denote it by f_T^H.

Consider the functional Ψ_T^H given in (7) and induced by the recurrence equation (9), *i.e.*,

$$\Psi_T^H(f)(n) = \begin{cases} c & \text{if } n = 1, \\ 2f(n-1) + d & \text{if } n \geq 2 \end{cases} \tag{11}$$

for all $f \in \mathcal{C}_c$.

Next we provide the asymptotic upper bound of f_T^H. With this aim, define the complexity function h_r as follows:

$$h_r(n) = \begin{cases} c & \text{if } n = 1, \\ r(2^n - 1) & \text{if } n \geq 2, \end{cases}$$

where $r > 0$. It is not hard to check that Ψ_T^H is an improver with respect to h_r (*i.e.*, $\Psi_T^H(h_r) \leq_{p_{\mathcal{C}_c}} h_r$) if and only if $r \geq \max\{d, \frac{2c+d}{3}\}$. It follows, by statement (2) in Theorem 18, that $f_T^H \in \mathcal{O}(h_{\max\{d, \frac{2c+d}{3}\}})$.

Next we provide an asymptotic lower bound of f_T^H. It is a routine to check that Ψ_T^H is a worsener with respect to the complexity function h_s (*i.e.*, $h_s \leq_{p_{\mathcal{C}_c}} \Psi_T^H(h_s)$) if and only if $s \leq \min\{d, \frac{2c+d}{3}\}$, whence we deduce, by statement (1) in Theorem 18, that $f_T^H \in \Omega(h_{\min\{d, \frac{2c+d}{3}\}})$.

Therefore we obtain that $f_T^H \in \mathcal{O}(h_{\max\{d, \frac{2c+d}{3}\}}) \cap \Omega(h_{\min\{d, \frac{2c+d}{3}\}})$. Thus, by Remark 19, we obtain that $f_T^H \in \Theta(2^n)$, which is in accordance with the Hanoi asymptotic complexity class that can be found in the literature [44, 52]. □

Author details
[1]Department of Mathematics, Sciences Faculty for Girls, King Abdulaziz University, P.O. Box 4087, Jeddah, 21491, Saudi Arabia. [2]Department of Mathematics, King Abdulaziz University, P.O. Box 80203, Jeddah, 21859, Saudi Arabia. [3]Departamento de Ciencias Matemáticas e Informática, Universidad de las Islas Baleares, Ctra. de Valldemossa km. 7.5, Palma de Mallorca, 07122, Spain.

References
1. Gierz, G, Hofmann, KH, Keimel, K, Lawson, JD, Mislove, M, Scott, DS: Continuous Lattices and Domains. Cambridge University Press, Cambridge (2003)
2. Davey, BA, Priestley, HA: Introduction to Lattices and Order. Cambridge University Press, Cambridge (1990)
3. Matthews, SG: Partial metric topology. Ann. N.Y. Acad. Sci. **728**, 183-197 (1994)
4. Matthews, SG: An extensional treatment of lazy data flow deadlock. Theor. Comput. Sci. **151**, 195-205 (1995)
5. O'Neill, SJ: Two topologies are better than one. Technical report 283, Department of Computer Science, University of Warwick (1995)
6. O'Neill, SJ: Partial metrics, valuations, and domain theory. Ann. N.Y. Acad. Sci. **806**, 304-315 (1996)
7. Bukatin, MA, Scott, JS: Towards computing distances between programs via Scott domains. In: Logical Foundations of Computer Science. LNCS, vol. 1234, pp. 33-43 (1997)
8. Heckmann, R: Approximation of metric spaces by partial metric spaces. Appl. Categ. Struct. **7**, 71-83 (1999)
9. Schellekens, MP: A characterization of partial metrizability: domains are quantifiable. Electron. Notes Theor. Comput. Sci. **305**, 409-432 (2003)
10. Schellekens, MP: The correspondence between partial metrics and semivaluations. Theor. Comput. Sci. **315**, 135-149 (2004)
11. Romaguera, S, Valero, O: A quantitative computational model for complete partial metric spaces via formal balls. Math. Struct. Comput. Sci. **19**, 541-563 (2009)
12. Romaguera, S, Valero, O: Domain theoretic characterizations of quasi-metric completeness in terms of formal balls. Math. Struct. Comput. Sci. **20**, 453-472 (2010)
13. Seda, AK, Hitzler, P: Generalized distance functions in the theory of computation. Comput. J. **53**, 443-464 (2010)
14. Romaguera, S, Schellekens, MP, Valero, O: Complexity spaces as quantitative domains of computation. Topol. Appl. **158**, 853-860 (2011)
15. Romaguera, S, Tirado, P, Valero, O: Complete partial metric spaces have partially metrizable computational models. Int. J. Comput. Math. **89**, 284-290 (2012)
16. Abbas, M, Nazir, T, Romaguera, S: Fixed point results for generalized cyclic contraction mappings in partial metric spaces. Rev. R. Acad. Cienc. Exactas Fís. Nat., Ser. A Mat. **106**, 287-297 (2012)
17. Abdeljawad, T, Alzabut, JO, Mukheimer, A, Zaidan, Y: Banach contraction principle for cyclical mappings on partial metric spaces. arXiv:1112.5891v1 [math.GN] (2011)
18. Abdeljawad, T, Karapinar, E, Tas, K: Existence and uniqueness of a common fixed point on partial metric spaces. Appl. Math. Lett. **24**, 1900-1904 (2011)
19. Alghamdi, MA, Shahzad, N, Valero, O: On fixed point theory in partial metric spaces. Fixed Point Theory Appl. **2012**, 175 (2012). doi:10.1186/1687-1812-2012-175
20. Altun, I, Erduran, A: Fixed point theorems for monotone mappings on partial metric spaces. Fixed Point Theory Appl. **2011**, Article ID 508730 (2011)
21. Altun, I, Simsek, H: Some fixed points theorems on dualistic partial metric spaces. J. Adv. Math. Stud. **1**, 1-8 (2008)
22. Altun, I, Sadarangani, K: Corrigendum to 'Generalized contractions on partial metric spaces' [Topology Appl. 157 (2010) 2778-2785]. Topol. Appl. **158**, 1738-1740 (2011)
23. Altun, I, Sola, F, Simsek, H: Generalized contractions on partial metric spaces. Topol. Appl. **157**, 2778-2785 (2010)
24. Agarwal, RP, Alghamdi, MA, Shahzad, N: Fixed point theory for cyclic generalized contractions in partial metric spaces. Fixed Point Theory Appl. **2012**, 40 (2012). doi:10.1186/1687-1812-2012-40
25. Aydi, H, Abbas, M, Vetro, C: Partial Hausdorff metric and Nadler's fixed point theorem on partial metric spaces. Topol. Appl. **159**, 3234-3242 (2012)
26. Aydi, H, Vetro, C, Sintunavarat, W, Kumam, P: Coincidence and fixed points for contractions and cyclical contractions in partial metric spaces. Fixed Point Theory Appl. **2012**, 124 (2012). doi:10.1186/1687-1812-2012-124
27. Ćirić, L, Samet, B, Aydi, H, Vetro, C: Common fixed points of generalized contractions on partial metric spaces and an application. Appl. Math. Comput. **218**, 2398-2406 (2011)
28. Di Bari, C, Vetro, P: Common fixed points for ψ-contractions on partial metric spaces. Hacet. J. Math. Stat. (to appear)
29. Di Bari, C, Vetro, P: Fixed points for weak φ-contractions on partial metric spaces. Int. J. Eng. Contemp. Math. Sci. **1**, 5-13 (2011)
30. Ilić, D, Pavlović, V, Rakočević, V: Some new extensions of Banach's contraction principle to partial metric space. Appl. Math. Lett. **24**, 1326-1330 (2011)
31. Ilić, D, Pavlović, V, Rakočević, V: Extensions of Zamfirescu theorem to partial metric spaces. Math. Comput. Model. **55**, 801-809 (2012)

32. Karapinar, E, Erhan, IM: Fixed point theorems for operators on partial metric spaces. Appl. Math. Lett. **24**, 1894-1899 (2011)
33. Karapinar, E: Fixed point theory for cyclic weak ϕ-contraction. Appl. Math. Lett. **24**, 822-825 (2011)
34. Karapinar, E: Weak ϕ-contraction on partial metric spaces and existence of fixed points in partially ordered sets. Math. Aeterna **1**, 237-244 (2011)
35. Oltra, S, Valero, O: Banach's fixed point theorem for partial metric spaces. Rend. Ist. Mat. Univ. Trieste **36**, 17-26 (2004)
36. Romaguera, S: Matkowski's type theorems for generalized contractions on (ordered) partial metric spaces. Appl. Gen. Topol. **12**, 213-220 (2011)
37. Romaguera, S: Fixed point theorems for generalized contractions on partial metric spaces. Topol. Appl. **159**, 194-199 (2012)
38. Rus, IA: Fixed point theory in partial metric spaces. An. Univ. Vest. Timiş., Ser. Mat.-Inform. **XLVI**, 149-160 (2008)
39. Valero, O: On Banach fixed point theorems for partial metric spaces. Appl. Gen. Topol. **6**, 229-240 (2005)
40. Vetro, F, Radenović, S: Nonlinear ψ-quasi-contractions of Ćirić-type in partial metric spaces. Appl. Math. Comput. **219**, 1594-1600 (2012)
41. Paesano, D, Vetro, P: Suzuki's type characterizations of completeness for partial metric spaces and fixed points for partially ordered metric spaces. Topol. Appl. **159**, 911-920 (2012)
42. Baranga, A: The contraction principle as a particular case of Kleen's fixed point theorem. Discrete Math. **98**, 75-79 (1991)
43. Dugundji, J, Granas, A: Fixed Point Theory. Polish Sci., Warsaw (1982)
44. Aho, AV, Hopcroft, JE, Ullman, JD: The Design and Analysis of Computer Algorithms. Addison-Wesley, Reading (1974)
45. Brassard, G, Bratley, P: Algorithms: Theory and Practice. Prentice Hall, New York (1988)
46. Schellekens, MP: The Smyth completion: a common foundation for denotational semantics and complexity analysis. Electron. Notes Theor. Comput. Sci. **1**, 211-232 (1995)
47. Künzi, HPA: Nonsymmetric distances and their associated topologies: about the origins of basic ideas in the area of asymmetric topology. In: Aull, CE, Lowen, R (eds.) Handbook of the History of General Topology, vol. 3, pp. 853-968. Kluwer Academic, Dordrecht (2001)
48. Romaguera, S, Schellekens, MP: Quasi-metric properties of complexity spaces. Topol. Appl. **98**, 311-322 (1999)
49. Oltra, S, Romaguera, S, Sánchez-Pérez, EA: Bicompleting weightable quasi-metric spaces and partial metric spaces. Rend. Circ. Mat. Palermo **51**, 151-162 (2002)
50. Cerdà-Uguet, MA, Schellekens, MP, Valero, O: The Baire partial quasimetric space: a mathematical tool for the asymptotic complexity analysis in computer science. Theory Comput. Syst. **50**, 387-399 (2012)
51. Romaguera, S, Tirado, P, Valero, O: New results on mathematical foundations of asymptotic complexity analysis of algorithms via complexity spaces. Int. J. Comput. Math. **89**, 1728-1741 (2012)
52. Cull, P, Flahive, M, Robson, R: Difference Equations: From Rabbits to Chaos. Springer, New York (2005)