The Theory of Quasi-Metric Spaces: A Topological Foundation for the Complexity of Algorithms

Fameno Rakotoniaina (fameno@aims.ac.za)
African Institute for Mathematical Sciences (AIMS)

Supervised by: Dr. Y.U. Gaba
Institut de Mathématiques et de Sciences Physiques (IMSP), Bénin
Co-supervised by: Dr. C.A. Agyingi
University of South Africa (UNISA), South Africa

14 May 2020

Submitted in partial fulfillment of a structured masters degree at AIMS South Africa

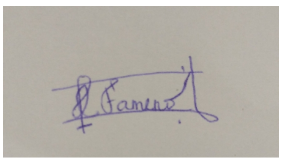
AIMS | African Institute for Mathematical Sciences SOUTH AFRICA

Abstract

In this work, motivated by the use of partial metrics in Computer Science, we analyze the asymptotic complexity of an algorithm via fixed point applied on the well-known Baire partial metric on the set of words over an alphabet.

Declaration

I, the undersigned, hereby declare that the work contained in this research project is my original work, and that any work done by others or by myself previously has been acknowledged and referenced accordingly.



Fameno Rakotoniaina, 14 May 2020

Contents

1. Introduction

In Computer Science, one interesting task is to compare running times of algorithms when the input gets larger, to find the best algorithm to solve a problem. To this end, we introduce the metric (distance) to quantify the relative progress in running time we make from an algorithm A to an algorithm B.

Question: Given two algorithms, **A** and **B**, is the progress from **A** to **B** the same as the other direction?

Definitely, NO! In terms of managing space, ordering and disordering a room are not the same. The QM ensures this property of non-symmetry. This was already introduced by Schellekens (1995) and considered as a topological foundation of asymptotic complexity analysis of algorithm. In his work, he constructed a complexity space and introduced improver. He also reproved the complexity class of Mergesort.

But before, Matthews developed the partial metric and the convenient version of Banach fixed point theorem (Matthews, 1994). He introduced PM to model computational processes where a quantitative degree of the information content of the involved element is needed. Therefore, the space is where the distance of a point to itself might not be zero.

Latter in 2012, inspired by these two notions, the Baire PQMS was proved as a suitable tool to study asymptotic complexity analysis Cerdà-Uguet, Schellekens, and Valero (2012).

In this work, we will start by analyzing the Divide and Conquer algorithm whose running time $T(n)$ satisfies:

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ aT(\frac{n}{b}) + h(n) & \text{else} \end{cases}$$

Studying the running time of such an algorithm which is provided by the solution of the (linear) recurrence equation, the extended Banach fixed point is the main focus in this work since it guarantees the existence and uniqueness of the optimal running time.

The remainder of this essay is organized as follows: Chapter 2 gives a background in metric spaces and the Banach fixed point theorem, as well as order theory and algorithm theory. We will also, introduce some definitions and the fixed point theorem in the QMS.

In Chapter 3, we discuss the motivation to PQM. Indeed, we will discuss the utility of the mathematical approach established in Schellekens (1995). Nevertheless, this approach is not valid in PMS for the complexity analysis. Moreover, the Banach fixed point theorem is stated in PQMS after defining its completeness and its equivalence to QMS.

Chapter 4 is devoted to develop how the QM and PM are unified to construct the Baire PQMS to study complexity of algorithms. This framework preserves the idea in Schellekens (1995) but, in the meantime, allows us to use the fixed point in Matthews (1994). Thereafter, we will show the existence and uniqueness of solution of the recurrence equation of Divide and Conquer algorithm as well as a more general class of recursive algorithm.

In Chapter 5, we validate the applicability of this developed theory by retrieving the complexity class of Mergesort, Quicksort, Largetwo, Hanoi, and Fibonacci.

2. Preliminaries

In this chapter, we recall certain materials that will enable us to understand the work in the write-up. We will start by some background in metric space since we need the notion of distance. Thereafter, we state the Banach fixed point theorem to ensure the optimal running time complexity of an algorithm. But in order to be able to compare running time complexity, we will introduce some definitions and properties in order theory and algorithm theory. Finally, the QM allows us to differentiate the quantitative distance in lowering the running time between two algorithms. Therefore, we will study all the materials provided in metric space in the QMS.

2.1 Metric Spaces

In this section, we will give the definition of metric space and the Banach fixed point.

2.1.1 Definition. (Sutherland, 2009) A **metric** on a non-empty set X is a function $d : X \times X \to [0, +\infty)$ satisfying:

D1) Positivity: $\forall x, y \in X,\ d(x, y) \geq 0$ and $d(x, y) = 0$ if and only if $x = y$,

D2) Symmetry: $\forall x, y \in X,\ d(x, y) = d(y, x)$,

D3) Triangle inequality: $\forall x, y, z \in X,\ d(x, y) \leq d(x, z) + d(z, y)$.

The pair (X, d) is called a **metric space**.

2.1.2 Example. (Sutherland, 2009) Let $x, y \in \mathbb{R}$.

- The usual metric on $\mathbb{R}$ is defined by $d(x, y) = |x - y|$. By definition of the absolute value, it is always positive and $\|x - y\| = 0$ if and only if $x = y$. Additionally, we have

$$d(x, y) = |x - y| = |x - z + z - y| \leq |x - z| + |z - y|.$$

- The discrete metric:

$$d(x, y) = \begin{cases} 0 & \text{if } x = y, \\ 1 & \text{otherwise.} \end{cases}$$

  Obviously, d verifies the two first conditions of a metric. For the triangle inequality, we have the following possibilities.

  Case 1: For $x = y = z$, we get: $0 = d(x, y) = d(x, z) + d(z, y) = 0 + 0 = 0$.

  Case 2: For $x = y \neq z$, we get $0 = d(x, y) < d(x, z) + d(z, y) = 1 + 1 = 2$.

  Case 3: For $x \neq y = z$, we get: $1 = d(x, y) = d(x, z) + d(z, y) = 1 + 0 = 1$.

  Case 4: For $x \neq y \neq z$, we get: $1 = d(x, y) < d(x, z) + d(z, y) = 1 + 1 = 2$.

  In any case we have that

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

Notice from Example 2.1.2 that we can define more than one metric on a given set.

2.1.3 Cauchy sequences and complete metric spaces.

The completeness of a metric space is defined by the convergence of all Cauchy sequence in this space. Before, we define the open balls in a metric space in order to define the convergence of sequence in such space. For this section, we refer the reader to Willard (2004).

The topology induced by the metric d on X has as basis the collection of open balls

$$B(a,r) = \{x \in X : d(a,x) < r\}$$

with $r > 0$.

2.1.4 Definition. Let $(x_n)_n$ be a sequence in a metric space (X,d). A point $x_0 \in X$ is called a **limit** of the sequence if for every $\epsilon > 0$, there exists a positive integer N such that $x_n \in B(x_0,\epsilon)$ for all $n \geq N$. Alternatively, $d(x_n,x_0) \leq \varepsilon$.

If the sequence has a limit x_0, we say that the sequence $(x_n)_n$ converges to x_0 and we write $x_n \to x_0$.

2.1.5 Definition. Let (X,d) be a metric space. A sequence $(x_n)_n$ of points in X is called a **Cauchy sequence** (or is called **Cauchy**) if $\forall\, \epsilon > 0$, there exists an integer $N_0 > 0$ such that $\forall\, n,m \geq N_0$, we have $d(x_n,x_m) < \epsilon$.

To avoid any ambiguity, we say "converge" when the limit belong to the considered metric space.

2.1.6 Definition. A metric space is said to be **complete** if all its Cauchy sequences converge.

2.1.7 Remark. Not every Cauchy sequence is convergent. Consider the metric space $((0,1],d)$ where d is the usual metric in $\mathbb{R}$. Let $x_n = \frac{1}{n}$ be a sequence in $((0,1],d)$. Then $(x_n)_n$ is Cauchy since

$$|x_n - x_m| < \frac{1}{\min(n,m)}.$$

The sequence $(x_n)_n$ converges to 0 but $0 \notin (0,1]$, thus $(x_n)_n$ does not converge in $(0,1]$.

However, every convergent sequence is a Cauchy sequence. Indeed, if x is the limit of the sequence $(x_n)_n$, then for any $\epsilon > 0$ there is a number $N \in \mathbb{N}$ such that $d(x_n,x) < \frac{\epsilon}{2}$ whenever $n \geq N$. For $n,m \geq N$, the triangle inequality D3 gives:

$$d(x_n,x_m) \leq d(x_n,a) + d(a,x_m) < \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon.$$

2.1.8 Example. The following are examples of complete metric spaces.

1. $\mathbb{R}$ is complete with respect to the usual metric.
2. $\mathcal{C}^0([0,1])$ is complete with respect to the metric d_∞. Indeed, let $f_n : [0,1] \to \mathbb{R}$ be a Cauchy sequence. Then
$$\forall\, \epsilon > 0, \exists\, n_0 \in \mathbb{N} : \forall\, n,m > n_0 : \sup_{x\in[0,1]} |f_n - f_m| < \epsilon.$$
So for every $x \in [0,1]$, we have $|f_n(x) - f_m(x)| < \epsilon$. Since f_n is a Cauchy sequence in $\mathbb{R}$ and $\mathbb{R}$ is complete, we have that there exists $x \in \mathbb{R}$ such that
$$\forall\, \epsilon > 0, \exists\, n_1 \in \mathbb{N} : \forall\, n > n_1,\ |f_n(x) - f(x)| < \epsilon.$$
Setting $N = max(n_0,n_1)$, we have $\forall\, n \geq N, |f_n(x) - f(x)| < \epsilon$. Since $f \in \mathcal{C}^0([0,1])$, it shows that $(\mathcal{C}^0([0,1]),d_\infty)$ is complete.

2.1.9 Banach fixed point theorem.
The Banach Fixed point theorem is an important pillar to study the complexity of an algorithm. Its first version in other kind of metric was first introduced by Matthews to study denotational semantics of programming languages, Matthews (1994).

2.1.10 Definition. (Sutherland, 2009) Let (X, d) be a metric space. A map $f : X \to X$ is said to be a **contraction** if there exists a positive real number $\lambda < 1$ such that $\forall x, y \in X$,

$$d(f(x), f(y)) \leq \lambda d(x, y).$$

The real number λ is called a **contraction factor**.

2.1.11 Theorem. *(Sutherland, 2009) Assume that (X, d) is a complete metric space and that $f : X \to X$ is a contraction. Then f has a unique fixed point a, and no matter which starting point $x_0 \in X$ we choose, the sequence*

$$x_0, x_1 = f(x_0), x_2 = f(x_1), \ldots, x_n = f(x_{n-1}), \ldots$$

converges to a

Proof. i. Uniqueness. If a and b are two fixed points, and λ is a contraction factor for f, we have

$$d(a, b) = d(f(a), f(b) \leq \lambda d(a, b).$$

Since $0 < \lambda < 1$, this is only possible when $a = b$ by D1.

ii. Existence. Fix $x_0 \in X$ and consider the sequence

$$x_0, x_1 = f(x_0), x_2 = f(x_1), \ldots, x_n = f(x_{n-1}), \ldots.$$

It only suffices to prove that this sequence is Cauchy and since (X, d) is complete, it follows that it must converge to a point x and $x = f(x)$. Thus, we have the fixed point.

Let's show that the sequence defined above is a Cauchy sequence. As f is contraction mapping, we have

$$d(x_2, x_1) \leq \lambda d(x_1, x_0)$$

and by induction

$$d(x_{n+1}, x_n) \leq \lambda^n d(x_1, x_0) \tag{2.1.1}$$

Choose two elements x_n, x_{n+k} of the sequence. By repeated use of the triangle inequality, we get

$$\begin{aligned}
d(x_n, x_{n+k}) &\leq d(x_n, x_{n+1}) + d(x_{n+1}, x_{n+2}) + \cdots + d(x_{n+k-1}, x_{n+k}) \\
&= d(f(x_{n-1}), f(x_n)) + d(f(x_n), f(x_{n+1}) + \cdots + d(f(x_{n+k-2}), f(x_{n+k-1}) \\
&\leq \lambda^n d(x_0, x_1) + \lambda^{n+1} d(x_0, x_1) + \cdots + \lambda^{n+k-1} d(x_0, x_1) \\
&= \frac{\lambda^n(1 - \lambda^k)}{1 - \lambda} d(x_0, x_1) \\
&\leq \frac{\lambda^n}{1 - \lambda} d(x_0, x_1) \quad (\text{ since } \lambda < 1).
\end{aligned}$$

In particular, choosing $n = N$ for a given $\epsilon > 0$, we have that $\forall n, m > N$,

$$d(x_n, x_m) \leq \frac{\lambda^n}{1 - \lambda} d(x_0, x_1) < \epsilon.$$

Hence the sequence is Cauchy.

□

2.2 Background from Order Theory

Latter, we will see that we have to find an upper bound. Therefore, let introduce the order. We complete this section according to Davey and Priestley (2002).

2.2.1 Definition. Let P be a set and $\mathcal{R}$ be a binary relation on P. Then $\mathcal{R}$ is a **partial order** (or order) on P if for all a, b and c in P, we have:

PO1) $a\mathcal{R}a$ (reflexivity),

PO2) if $a\mathcal{R}b$ and $b\mathcal{R}a$ then $a = b$ (antisymmetry),

PO3) if $a\mathcal{R}b$ and $b\mathcal{R}c$ then $a\mathcal{R}c$ (transitivity).

Then the pair $(P, \mathcal{R})$ is called a **partially ordered set** or **poset**. If there is no ambiguity about the order, we simply say that P is a poset. A relation $\mathcal{R}$ on a set P which is reflexive and transitive but not necessarily antisymmetric is called a **quasi-order** or a **pre-order**.

2.2.2 Example. On any set, the binary relation $=$ is an order, the discrete order.

But, for example if we say "*x is less than y*", then we have "*y is greater than x*". This notion is called the duality. Generally, it is obtained by reverting the direction, and the resulting order is called the dual. Formally, we have the following definition:

An ordered relation $\leq$ gives rise to a relation $<$ as follows: $x < y$ if and only if $x \leq y \wedge x \neq y$. We write $x||y$ when $x \nleq y$ and $y \nleq x$, i.e., x and y are not comparable.

2.2.3 Minimal and maximal elements.

2.2.4 Definition. Let Q be a subset of P with respect to the induced order. We say that $m \in Q$ is a **minimal element** of Q if there exists $x \in Q$ such that $x \leq m$, then $x = m$, that is if there is an element less than m then this element will be equal to m.

The element M is said to be **maximal** if there exists $x \in Q$ such that $M \leq x$, then $M = x$. The set of minimal element of Q is $\min Q$ and the set of maximal element of Q is $\max Q$.

A stronger notion is maximum and minimum.

2.2.5 Definition. Let $(P, \leq)$ be a poset. An element m is the **minimum** of P if $m \leq x$ for all $x \in P$. An element M is the **maximum** of P if $x \leq M$ for all $x \in P$.

2.2.6 Example. Let P be the collection $\{\{a, b\}, \{a, b, c\}, \{c, d, a, b\}, \{a, d, f\}\}$ with respect to the inclusion order $\subseteq$. $\{a, b\}$ is minimal because no element of P contains it. $\{c, d, a, b\}$ is maximal since it is not contained in any element of P. $\{a, b\} \subseteq \{a, b, c\} \subseteq \{c, d, a, b\}$, then {a,b,c} in neither minimal nor maximal. But $\{a, d, f\}$ is both minimal and maximal, and P has neither a maximum nor a minimum.

The following will lead us to an example where $\min P$ and $\max P$ has one element each and they are respectively the minimum and the maximum of an ordered set P.

2.2.7 Remark. The maximal and minimal element can also not be defined. It is the case for the set of integers $\mathbb{Z}$.

The existence of maximal elements under some conditions is stated in Zorn's lemma.

2.2.8 Order in Computer Science.
Let $\sum^{**}$ be the set of all finite and infinite binary string, i.e., the set of sequence of 0 and 1. We endow $\sum^{**}$ with the order $\leq$ such that for $u, v \in \sum^{*}, u \leq v$ if and only if $u = v$ or u is a finite initial subsequence of v (Davey and Priestley, 2002). A more general notion is given in the paper of Cerdà-Uguet, Schellekens, and Valero (2012).

Let $\sum$ be a non empty set called alphabet, endowed with some order $\preceq$. Let $\sum^{\infty}$ be set of all finite and infinite sequence over $\sum$. Let us call each element of $\sum^{\infty}$ a word and we write $x = x_0 x_1 \dots x_n$ where n is the length of x denoted $l(x)$. Then $l(x, y)$ is the length of the common prefix of x and y and is defined by $l(x, y) = sup\{n \in \mathbb{N} : x_k = y_k$ for $k \leq n\}$.

We set the prefix order $\sqsubseteq$ on $\sum^{\infty}$ such that $x \sqsubseteq y$ if x is a **prefix** of y, i.e., $l(x, y) = l(x)$ (Cerdà-Uguet, Schellekens, and Valero, 2012).

For $\sum = \{0, 1\}$ endowed with the order $\leq$ and $\sum^{**}$ the finite set of all sequence of length ≤ 2. We have: $0 \sqsubseteq 01$, $0 \sqsubseteq 00$, $1 \sqsubseteq 10$, $1 \sqsubseteq 11$.

Otherwise, we can also define another order $\sqsubseteq_{sp}$ on $\sum^{\infty}$ defined as $x \sqsubseteq_{sp} y$ if there exists n_0, such that $x_k \preceq y_k$ for all $k \leq n_0$. In this case, x is called **subprefix** of y. Observe that a prefix is also a subprefix. Thus, for the same set, we have $0 \sqsubseteq_{sp} 10$ and $0 \sqsubseteq_{sp} 11$ in addition.

2.3 Quasi-metric, their topology, specialization order and Banach Fixed Point

2.3.1 Quasi-metric spaces.
Generally, QM is a metric without the symmetry axiom. Therefore, in quasi-metric spaces some concepts that we are interested in, convergence and completeness, are slightly different from those in metric case. Its first application was introduced by Schellekens (1995) as a topological foundations for the complexity of algorithms. For further details, we refer the reader to Brattka (2003), Wilson (1931) and Romaguera and Schellekens (1999)

2.3.2 Definition. (Wilson, 1931) Let X be a non empty set. A **quasi-metric** on X is a function $q : X \times X \to [0, +\infty)$ satisfying $\forall x, y, z \in X$:

QM1) $q(x, y) \geq 0$ and $q(x, x) = 0$.

QM2) if $q(x, y) = 0 = q(y, x)$, then $x = y$.

QM3) $q(x, y) \leq q(x, z) + q(z, y)$.

The pair (X, q) is called a **quasi-metric space**.

If q does not satisfy (ii) then, q is a quasi-pseudometric.

2.3.3 Example. (Brattka, 2003) In the set of real numbers, the truncated difference u, defined by $u(x, y) = \max\{y - x, 0\}$ for $x, y \in \mathbb{R}$, is a quasi-metric .

2.3.4 Example. (Brattka, 2003) On a non-empty set, if d is a quasi-metric then, its conjugate $\bar{d}$ is a quasi-metric, which is defined as $\bar{d}(x, y) = d(y, x)$.

$$q'(x, y) = q(\phi, x) + q(x, y) - q(\phi, y).$$

2.3.5 Example. (Schellekens, 1995) The complexity space which consists of the pair $(\mathcal{C}, d_{\mathcal{C}})$ introduced by Schellekens, where

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

with $\mathcal{RT}$ the set of all functions from $\mathbb{N}$ into $[0, \infty)$, and

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

whenever $f, g \in \mathcal{C}$, is a quasi-metric space. This distance provides information about the relative progress in lowering complexity by replacing an algorithm with complexity function f with another algorithm with complexity function g. And if $d(f, g) = 0$, it means that the algorithm with complexity f is at least as efficient as the algorithm with complexity g.

Given a QM q on X, we can define a metric on X.

2.3.6 Proposition. (Romaguera and Schellekens, 1999) The function $q^s : X \times X \to \mathbb{R}, (x, y) \mapsto \max\{q(x, y), q(y, x)\}$ is a metric on X. And q^s is called the **induced metric**.

Proof. Let x, y, z be elements of X.

D1) By QM1, $q(x, y)$ and $q(y, x)$ are positive so $q^s(x, y) \geq 0$. Moreover if $x = y$, $q^s(x, y) = 0$. And if $q^s(x, y) = 0$, then $q(x, y) = 0 = q(y, x)$. Thus $x = y$.

D2)
$$\begin{aligned} q^s(x, y) &= \max\{q(x, y), q(y, x)\} \\ &= \max\{q(y, x), q(x, y)\} \\ &= q^s(y, x) \end{aligned}$$

D3) By QM3, we have $q(x, z) \leq q(x, y) + q(y, z) \leq q^s(x, y) + q^s(y, z)$ and $q(z, x) \leq q(z, y) + q(y, x) \leq q^s(z, y) + q^s(y, x)$. Since q^s is symmetric, then $q^s(x, z) = \max\{q(x, z), q(z, x)\} \leq q^s(x, y) + q^s(y, z)$.

□

2.3.7 Specialization order and topology of quasi-metric space .
A specialization order is a natural pre-order on the set of points of topological space. For a space that satisfies the T_0-separation axiom, this order is a partial order, called specialization order. Moreover, for a topological metric space, it is generated by the metric.

Let τ_q be the topology on the QMS (X, q). Each quasi-metric q on X generates a T_0-topology τ_q on X which has as basis the family of the open balls

$$\{B_q(x, r) : x \in X, r > 0\}.$$

with $B_q(x, r) = \{y \in X : q(x, y) < r\}$. Recall that a topology τ_q is T_0 if every pair of distinct points is topologically distinguishable, i.e.,if

$$\forall\, x, y \in X,\ x \neq y,\ \exists\, r > 0 : y \notin B_q(x, r).$$

This condition is the so-called T_0-separation axiom. A set together with a T_0-topology is called **Kolmogorov space**.

2.3.8 Theorem. *(Künzi, Pajoohesh, and Schellekens, 2006) Let (X, q) be quasi-metric space. Then $\leq_q$ defined as*

$$x \leq_q y \Leftrightarrow q(x, y) = 0,$$

is a partial order on X.

Proof. PO1) For every $x \in X$, $q(x, x) = 0$, so $x \leq_q x$.

PO2) If $x \leq_q y$ and $y \leq_q x$, then $q(x, y) = q(y, x) = 0$. Hence, $x = y$.

PO3) If $x \leq_q y$ and $y \leq_q z$, we have $q(x, z) \leq q(x, y) + q(y, z) = 0$. Therefore, $q(x, z) = 0$ and $x \leq_q z$.

□

Notice, since there is no notion of symmetry, we can define two T_0-topology on a set X (Brattka, 2003): the **forward** (*lower*) topology $\tau_<$ which has as basis the collection of open balls of the form

$$B_q^f(x, r) = \{y \in X : q(x, y) < r\},$$

and the **backward** (*upper*) topology $\tau_>$ which has as basis the collection of open balls of the form

$$B_q^b(x, r) = \{y \in X : q(y, x) < r\}.$$

2.3.9 Banach fixed point Theorem.
As in metric spaces, the fixed point theorem holds in a complete QMS, that we call further *bicomplete*. So, let us define a notion of completeness in quasi-metric spaces starting with the notion of convergence. Since the notion of symmetry is lost in a quasi-metric, we can thus define different types of convergence.

Since , we defined a topology on (X, q), we will define convergence and limit by the same setting as in definition 2.1.4.

2.3.10 Definition. (Wilson, 1931) A sequence (x_n) converges to x in a QM (X, q) if and only if for all $\varepsilon > 0$, there exists $N \in \mathbb{N}$ such that for every $n \geq N$, $x_n \in \mathcal{B}_q(x, \varepsilon)$.

As we have two different topologies on X, then we define different types of limit point.

- x is u-**limit** if for every $\varepsilon > 0$, there exists $N \in \mathbb{N}$ such that for every $n \geq N$, $x_n \in \mathcal{B}_q^f(x, \varepsilon)$, i.e. $q(x, x_n) < \varepsilon$.
- x is l-**limit** if for every $\varepsilon > 0$, there exists $N \in \mathbb{N}$ such that for every $n \geq N$, $x_n \in \mathcal{B}_q^b(x, \varepsilon)$, i.e. $q(x_n, x) < \varepsilon$
- x is c-**limit** if x is both u-**limit** and l-**limit**.

u-**limit** and l-**limit** point are also called collectively a **quasi-limit point**.

In this section, (X, q) is a quasi-metric space.

2.3.11 Remark. (Wilson, 1931) These limits are unique. Moreover, if a sequence (x_n) has u-limit or l-limit but does not have c-limit, then it has no l-limit, u-limit respectively.

2.3.12 Definition. A sequence (x_n) in a quasi-metric space (X, q) is a **Cauchy sequence** if it is a Cauchy sequence in the induced metric space (X, q^s).

2.3.13 Definition. (Wilson, 1931) A sequence (x_n) is **properly convergent** to $x \in X$, written $x_n \to x$ properly if (x_n) converge to x in the induced metric space (X, q^s), i.e., $x_n \to x$ if for every $\varepsilon > 0$, there exists n_0 such that for every $n \geq n_0$, $q^s(x_n, x) < \varepsilon$.

2.3.14 Proposition. Let (X, q^s) the induced metric space. Let (x_n) be a sequence in (X, q). Then (x_n) converges properly to x if and only if x is a c-limit.

Proof. (x_n) converges properly, that is

$$\forall \varepsilon > 0, \quad \exists N \in \mathbb{N}, \quad q(x_n, x) < \varepsilon \quad \text{and} \quad q(x, x_n) < \varepsilon.$$

So,

$$\forall \varepsilon > 0, \quad \exists N \in \mathbb{N}, \quad \max\{q(x_n, x), q(x, x_n)\} < \varepsilon.$$

□

In Doitchinov (1988), it said that the definition of Cauchy sequence in a QMS is equivalent to the usual definition in a metric space. Since we defined the induced metric, so(x_n) is a Cauchy sequence in (X, q) if it is a Cauchy in the induced metric space (X, q^s). That is for every $\varepsilon > 0$, there exists $N \in \mathbb{N}$ such that for every $n, m \geq N$, $q(x_n, x_m) < \varepsilon$ and $q(x_m, x_n) < \varepsilon$.

2.3.15 Definition. Then (X, q) is **bicomplete** if every Cauchy sequence in (X, q) converge in X with respect to the induced metric q^s.

In other words also, the limit of all Cauchy sequence in (X, q) is a c-limit, so it converges properly.

2.3.16 Proposition. The induced metric (X, q^s) is complete if and only if (X, q) is bicomplete.

Proof. If (x_n) is a Cauchy in (X, q), then (x_n) is a Cauchy in (X, q^s). Since (X, q^s) is complete, then there exists x such that

$$\forall \varepsilon > 0, \text{ there exists } N \in \mathbb{N}, \quad q^s(x, x_n) < \varepsilon$$

So, (x_n) converges properly. Hence, (X, q) is bicomplete.

If (x_n) is a Cauchy sequence in (X, q^s), then (x_n) is Cauchy in (X, q). And as (X, q) is complete, then (x_n) converges properly. Therefore, (x_n) converges in (X, q^s) and (X, q^s) is complete □

2.3.17 Example. (Schellekens, 1995) A well known example of bicomplete QMS and that inspired Schellekens is $((0, \infty], u_1)$ where u_1 is defined as

$$u_1(x, y) = \max\{\frac{1}{y} - \frac{1}{x}, 0\}$$

whenever $x, y \in (0, \infty]$.

Indeed,

QM1) $u_1(x, x) = 0$ and $\max\{\frac{1}{y} - \frac{1}{x}, 0\} \geq 0$ whenever $x, y \in (0, \infty]$

QM2) if $u_1(x, y) = 0 = u_1(y, x)$, then we have $\frac{1}{y} - \frac{1}{x} \leq 0$ and $\frac{1}{x} - \frac{1}{y} \leq 0$. As a result

$$\frac{1}{y} \leq \frac{1}{x} \quad \text{and} \quad \frac{1}{y} \leq \frac{1}{y}$$

Finally, $\frac{1}{y} = \frac{1}{x}$ and $x = y$

QM3) If $u_1(x,y) = 0$, then the triangular inequality always hold whenever $z \in (0,\infty]$. Suppose $u_1(x,y) = \frac{1}{y} - \frac{1}{x} > 0$. Then,

$$\begin{aligned} 0 < \frac{1}{y} - \frac{1}{x} &= \frac{1}{y} - \frac{1}{z} + \frac{1}{z} - \frac{1}{x} \quad \text{whenever } z \in (0,\infty] \\ &\leq \max\{\frac{1}{y} - \frac{1}{z}, 0\} + \max\{\frac{1}{z} - \frac{1}{x}, 0\} \\ &= u_1(z,y) + u_1(x,z) \end{aligned}$$

Hence, u_1 is quasi-metric. The induced metric is $u_1^s(x,y) = \max\{\frac{1}{y} - \frac{1}{x}, \frac{1}{x} - \frac{1}{y}, 0\}$. Since every metric is equivalent in the set of real number $\mathbb{R}$ and $(0,\infty]$ is a subset of $\mathbb{R}$. Then, $((0,\infty], u_1^s)$ is complete. Thus, $((0,\infty], u_1)$ is bicomplete.

2.3.18 Definition. (Schellekens, 1995) A self map $T : X \to X$ from a quasi-metric space (X,q) into itself is a contraction map if there exists a real constant $k \in [0,1)$ such that for every $x,y \in X$: $q(Tx,Ty) \leq kq(x,y)$.

2.3.19 Theorem. *Let (X,q) be a bicomplete quasi-metric space. If f is a contraction mapping on (X,q) to itself, then f has a unique fixed point $x^* \in X$. Furthermore, we can find x^* by starting with an arbitrary element $x_0 \in X$, and define the sequence $(x_n)_n$ by $x_n = Tx_{n-1}$, then we find that (x_n) converges to x^*. Moreover, if $x^* \in X$, then $q(x,x) = 0$.*

Proof. Let k be the contraction constant and (x_n) be a sequence in (X,q) defined as $x_n = f(x_{n-1})$. We have

$$q(x_2,x_1) \leq kq(x_1,x_0).$$

By induction, we get:

$$q(x_{n+1},x_n) \leq k^n q(x_1,x_0). \tag{2.3.1}$$

Now let $n < m \in \mathbb{N}$. Using the triangle inequality, we have:

$$\begin{aligned} q(x_m,x_n) &\leq q(x_m,x_{m-1}) + q(x_{m-1},x_{m-2}) + \ldots + q(x_{n+1},x_n) \\ &\leq k^{m-1}q(x_1,x_0) + k^{m-2}q(x_1,x_0) + \cdots + k^n q(x_1,x_0) \text{ by (2.3.1)} \\ &\leq k^n q(x_1,x_0) \sum_{i=0}^{m-n-1} k^i \\ &< k^n q(x_1,x_0) \sum_{i=0}^{\infty} k^i. \end{aligned}$$

Since $k \in [0,1)$, we get

$$q(x_m,x_n) < k^n q(x_1,x_0)\frac{1}{1-k}. \tag{2.3.2}$$

Let $\varepsilon > 0$. We can find larger $N \in \mathbb{N}$ such that:

$$k^N < \varepsilon\frac{1-k}{q(x_1,x_0)}. \tag{2.3.3}$$

Therefore by choosing $N < n < m$ and using the fact that $k \in [0,1)$, we get that $k^N > k^n$ and the inequality (2.3.2) becomes:

$$q(x_m,x_n) < \varepsilon.$$

This proves that the sequence (x_n) is Cauchy. By bicompleteness of (X, q) we get that there exists x^* in X such that x^* is the limit of (x_n). That is

$$x^* = \lim_{n\to\infty} x_n = \lim_{n\to\infty} f(x_{n-1}) = f(\lim_{n\to\infty} x_{n-1}) = f(x^*)$$

Therefore, x^* is the fixed point of the contraction mapping f. Moreover, it is unique since the limit is unique in bicomplete quasi-metric spaces. □

2.4 Theory of Algorithms: definitions and some properties

In this section, we provide some background about algorithm theory: some definitions and properties of algorithms that we shall need. The reader can refer to (Cormen et al., 2009) for these notions and more.

Informally, an algorithm is any well-defined computational procedure that takes some values (or set of values) called **input** and produces some values called **output**. Thus, we can think of it as a sequence of computational steps described unambiguously from the input to the corresponding output.

An algorithm should have liveness and correctness properties. An algorithm is said to be correct when each input gives the corresponding output. The liveness property ensures that the algorithm works and in finite time. That is, after some finite number of steps, it will stop running.

Another property, which is the main interest of this essay, is the efficiency of the algorithm. The **running time** is a quantitative way to describe it. A running time of an algorithm for a particular input is the number of steps or primitive operations executed. An algorithm is represented by a function called **complexity function**. It is a function that takes as argument the size of the input for a given algorithm and the image is the running time, time taken by the algorithm to produce the output. However, since there are several algorithms designed for the same problem, the set of these functions is called **complexity space**. Let $\mathcal{RT}$ be the set of function from $\mathbb{N}$ to $(0, \infty]$. This is a complexity space. Indeed, an algorithm has at least one step to return the output and it might be an infinite steps.

To find the explicit running time is an arduous task. Therefore, to ease the complexity analysis, we approximate it by another complexity function. This approximation are based on **asymptotic notation**. For a given algorithm whose complexity function is f, we have:

- $f(n) \in \Theta(g(n))$ if there exists $n_0 \in \mathbb{N}$, $c_1, c_2 > 0$ such that for all $n \geq n_0$,

$$c_1 g(n) \leq f(n) \leq c_2 f(n)$$

  nad g is called the **asymptotic tight bound**.

- $f(n) \in \mathcal{O}(g(n))$ if there exists $n_0 \in \mathbb{N}$, $c > 0$ such that for all $n \geq n_0$,

$$f(n) \leq cg(n)$$

  and g is called the **asymptotic upper bound**.

- $f(n) \in \Omega(g(n))$ if there exists $n_0 \in \mathbb{N}$, $c > 0$ such that for all $n \geq n_0$,

$$0 \leq cg(n) \leq f(n)$$

  and g is called the **asymptotic lower bound**.

For instance, if $f(n) = an^2+bn+c$ with $b, c \in \mathbb{R}$, $a > 0$, then $f(n) \in \mathcal{O}(n^2)$, with $n_0 = 2\max(\frac{|b|}{a}, \frac{\sqrt{|c|}}{a})$ and $c = \frac{7a}{4}$ (Cormen, Leiserson, Rivest, and Stein, 2009). In addition, for a constant function f, we have $f \in \Theta(1)$.

But usually, when an algorithm is efficient for a large input, then it is for all inputs. Thus, we will study the **asymptotic running time** complexity that we will approximate by the $\mathcal{O}$-notation. In this case, we say g **is the complexity class of** f. Thus, if there are some algorithms that solve a given problem, the $\mathcal{O}$-notation provides the (least) upper bound of the running time of the best algorithm.

But, it is necessary to distinguish three behaviour in the complexity analysis: the **worst case**, the **average case** and the **best case**. For example, the behaviour of Quicksort depends in the balance of partitioning. Recall that Quicksort is an algorithm to sort a list. Suppose n is the size of the list L and let be $L(j)$ the j-th element. This is an algorithm consisting taking the last element as pivot, finding the index j such that $L(k) < L(j)$ for all $k < j$ and $L(k) > L(j)$ for all $k > j$. As a result, we have two subarrays that we will solve by applying the same algorithm. At the end, we will combine all the solutions. The worst case occurs when the list is already sorted. The best case occurs if for each subarray, the pivot is the second minimum element. And the average case occurs when the partitioning is balanced. Furthermore, the worst case has the largest running time for any inputs while the best case has the smallest running time.

Many useful algorithm are designed with recursion: for a given problem, they call themselves one or more times inside the algorithm to solve related subproblems. For this reason, we are interested to study **Divide and Conquer** algorithm. Its approach is based on breaking the problem into subproblems similar to the original but with a smaller size, solving them recursively and combining them to get the main solution. Its running time $T(n)$ verify the recurrence equation

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ aT(\frac{n}{b}) + h(n) & \text{if} n \in \omega_b \end{cases} \tag{2.4.1}$$

where $n \in \mathbb{N}$ is the input size, $a > 0$ and $\omega_b = \{b^k : k \in \mathbb{N}\}$. $h(n)$ is the running time of dividing and combining the subsolutions while $c > 0$ is a real positif constant representing the running of the base case. We set $\mathcal{RT}_{b,c} = \{f \in \mathcal{RT} : f(1) = c \text{ and } f(n) = \infty \text{ for all } n \notin \omega_b \text{ and } n > 1\}$ as its complesity space. Quicksort in the best and average case, is an example of such an algorithm. And its running time complexity is $\mathcal{O}(\frac{1}{2}n\lg n)$ ($\lg = \log_2$). Another example is the Mergesort with $\mathcal{O}(n\lg n)$ in the worst case and $\mathcal{O}(\frac{1}{2}n\lg n)$ in the average and best case. Therefore, in this work, we will show the applicability of the developed theory to reprove these complexity class. Further, we will prove that our mathematical model can also be used to study another recurrence equation form.

3. Partial Quasi-metric

3.1 Motivation to Partial Quasi-metric Spaces:

Inspiring by the example of the bicomplete QMS in example 2.3.17, we will define a complexity space. Firstly, we will consider $(0, \infty]$ as the image of running time of complexity function. In fact, we drop out the zero since an algorithm takes time to run and some algorithms has infinite time of running. Secondly, define $d_{\mathcal{C}}$ as a QM on this space with

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

whenever $f, g \in \mathcal{C}$ and $n \in \omega_b$. As a convention , we have $2^{\infty} = 0$.

As a result, $\mathcal{C}_{b,c}$ together with the QM $d_{\mathcal{C}}$ is a bicomplete space. So we can apply the Banach fixed point theorem 2.3.19. Hence, we have a unique fixed point which is the solution of the aforementioned recurrence. Thus, the running time complexity of the Divide and Conquer algorithm. However, as we study complexity function, we introduce what we called **functional**, function which has as input a function. With in mind that we want that at each step, the running time of the algorithm approximates the solution, we introduce a particular functional called an **improver**. To clarify, a functional Φ is an improver with respect to a complexity function g, if $\Phi(g) \leq g$. Moreover, if such complexity function exists, then it is the class complexity of the runnning time solution. The functional associated to the recurrence 2.4.1 is given by:

$$\Phi_T(f)(n) = \begin{cases} c & \text{if } k = 1 \\ \infty & \text{if } k \notin \omega_k \text{ and } 2 \leq k \leq l(x) + 1 \\ af(\frac{n}{b}) + h(n) & \text{otherwise} \end{cases} \qquad (3.1.1)$$

Then, since Φ_T is a contraction mapping and $\mathcal{C}_{b,c}$ is complete, we have the following result as stated in (Schellekens, 1995):

3.1.1 Theorem. *A Divide and Conquer recurrence of form 2.4.1 has a unique solution $f_T \in \mathcal{C}_{b,c}$. Moreover, if the monotone functional Φ_T given by 3.1.1 is an improver with respect to some function $g \in \mathcal{C}_{b,c}$, then the solution to the recurrence equation satisfies that $f_T \in \mathcal{O}(g)$.*

Altogether, the QMS is a suitable mathematical tool to study running time complexity together with the functional.

In the other side, an algorithm with a recursive process has the following properties:

- Each step gives a better approximation of the solution than the previous steps.
- The final approximation is the limit of the computing process.

In a topological model, associated with an order, such successive approximation will converge to the least upper bound with respect to the topology. Alternatively, a mathematical model can be introduced to measure the information at each step of the computation. And as long as we keep computing, we approximate the solution. Thus, the computational steps can be seen as **increasing sequence**. To quantify those informations, another kind of metric is introduced, the **partial metric**, Matthews (1994). Indeed, the information involved by an element of the sequence is described by the distance of this element to itself. Apart from this, every partial metric induces a partial order. This latter will allow us to determine the least upper bound of the running time solution. So, let's recall some definitions and properties in partial metric space. These can be found in Matthews (1994).

3.1.2 Definition. A **partial metric** on a nonempty set X is a function $p : X \times X \to [0, +\infty)$ satisfying for every $x, y, z \in X$:

P1) Small self-distance: $p(x, x) \leq p(x, y)$;

P2) If $p(x, x) = p(x, y) = p(y, y)$, then $x = y$;

P3) Symmetry: $p(x, y) = p(y, x)$;

P4) Triangle inequality: $p(x, z) \leq p(x, y) + p(y, z) - p(y, y)$.

The pair (X, p) is called a partial metric space.

The open balls on (X, p) are sets of the form

$$B_p(x, r) = \{y \in X : p(x, y) - p(x, x) < \epsilon,\ \epsilon > 0\}. \tag{3.1.2}$$

3.1.3 Theorem. *Every partial metric p on X induces a partial order $\leq$ defined as*

$$x \leq y \Leftrightarrow p(x, x) = p(x, y)$$

whenever $x, y \in X$.

Proof. Let x, y, z be elements of X.

PO1) $p(x, x) = p(x, x)$, then $x \leq x$.

PO2) If we have $x \leq y$ and $y \leq x$, it implies that $p(x, x) = p(x, y)$ and $p(y, y) = p(y, x)$. By P3, we have that $p(x, x) = p(x, y) = p(y, y)$. Hence, $x = y$ by P2.

PO3) If we have $x \leq y$ and $y \leq z$, then $p(x, x) = p(x, y)$ and $p(y, y) = p(y, z)$. On the other hand, P4 tell us $p(x, z) \leq p(x, y) + p(y, z) - p(y, y) = p(x, x)$. Since, by P1, $p(x, x) \leq p(x, z)$, so , we have $p(x, x) = p(x, z)$ and $x \leq z$.

□

Inspiring by the usefulness of partial metric in computer science to analyze a program using recursion process, Matthews extended the Banach fixed point into to partial metric and we have the following result:

3.1.4 Theorem. *Let f be a mapping of complete partial metric space (X, p). If there exists $s \in \mathbb{R}^+$ such that*

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

whenever $x, y \in X$. Then f has a unique fixed point. Moreover if $x \in X$, then $p(x, x) = 0$.

In additional, the function p_q defined as $p_q(x, y) = p(x, y) - p(x, x)$ is a QM. All of this point, we will endow the complexity space $\mathcal{C}_{b,c}$ by a partial metric defined as

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

Obviously, $d_{\mathcal{C}}$ is the induced quasi-metric of $p_{\mathcal{C}}$.

But let's consider two elements of $\mathcal{C}_{b,c}$ (Schellekens, 1995):$f(n) = 2c$ and $g(n) = 2(c+1)$ for all $n \in \omega_b$. Involving the functional Φ_T, we have

$$p_{\mathcal{C}}(\Phi_T(f), \Phi_T(g) = \frac{1}{2c} + \sum_{n=2, n\in\omega_b}^{\infty} 2^{-n} \max\left\{\frac{1}{2c + h(n)}, \frac{1}{2(c+1) + h(n)}\right\}$$

and

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

Since Φ_T is a contraction mapping of contraction factor $0 < s < 1$, we have:

$$\frac{1}{2c} \leq p_{\mathcal{C}}(\Phi_T(f), \Phi_T(g) \leq s p_{\mathcal{C}}(f, g) \leq s\frac{1}{2c}$$

We conclude that $1 \leq s$ which is a contradiction. Therefore, the partial metric is not enough to study the asymptotic complexity analysis. As the QMS is a tool to study complexity of algorithm, then we will do some modifications in the partial metric inspired by the quasi-metric and some materials that we introduced in the complexity space. Hence, the partial quasi-metric space on the complexity space using the Banach fixed point theorem and the functional.

3.1.5 Definition. (Künzi, Pajoohesh, and Schellekens, 2006) A **partial quasi-metric** on a nonempty set X is a function $p : X \times X \to [0, \infty)$ satisfying for every $x, y, z \in X$:

PQ1) $p(x, x) \leq p(x, y)$;

PQ2) $p(x, x) \leq p(y, x)$;

PQ3) $p(x, z) \leq p(x, y) + p(y, z) - p(y, y)$;

PQ4) $x = y$ if and only if $(p(x, x) = p(x, y)$ and $p(y, y) = p(y, x))$.

The pair (X, p) is called a **partial quasi-metric space**.

The axioms PQ1 and PQ2 are called small-self distance axioms. PQ3 is the transitivity axiom and PQ4 is the anti-symmetry axiom.

If we add the symmetry axiom to the above axioms, then p becomes a partial metric. If p does not necessarily satisfy PQ2, then we shall speak about **lopsided** PQM. For this work, we will assume that the PQM p is lopsided.

If p satisfies PQ2, then the axiom PQ4 is also equivalent to :

PQ4') $x = y$ if and only if $(p(x, x) = p(y, x)$ and $p(y, y) = p(x, y)$.

Indeed, if we suppose that for $x, y \in X$, $p(y, x) = p(x, x)$ and $p(x, y) = p(y, y)$, then $p(x, x) - p(x, y) + p(y, y) - p(y, x) = 0$, and $p(x, x) + p(y, y) = p(x, y) + p(y, x)$. Since p satisfies PQ2, we have then that $p(x, x) = p(y, x)$ and $p(y, y) = p(x, y)$. Similarly, to prove the other implication, we will make use of PQ1 instead of PQ2.

Moreover, for $x, y \in X$, if $p(x, y) = 0$, then from PQ1, $x = y$.

3.1.6 Example. (Karapınar et al. (2013)) Let (X, p) be a PMS. Set $q_p(x, y) = p(x, y) - p(x, x)$. Then (X, q_p) is a PQMS. Indeed, $q_p(x, y) = p(x, y) - p(x, x) \neq p(x, y) - p(y, y) = q_p(y, x)$. The other axioms holds.

3.2 The Topology and Associated Order on Partial Quasi-metric Spaces

From now on, (X,p) will denote a partial quasi-metric space. Like in metric spaces, we can define a specialization order on (X,p).

3.2.1 Theorem. *For each PQM p on X, the order $\leq_p$ defined by*

$$x \leq_p y \Leftrightarrow p(x,x) = p(x,y), \tag{3.2.1}$$

whenever $x, y \in X$, is a partial order.

Proof. Let $x, y, z \in X$.

PO1) $p(x,x) = p(x,x)$, then $x \leq_p x$.

PO2) If $x \leq_p y$ and $y \leq_p x$, then $p(x,x) = p(x,y)$ and $p(y,y) = p(y,x)$. We then have directly from PQ4 that $x = y$.

PO3) If $x \leq_p y$ and $y \leq_p z$, then $p(x,x) = p(x,y)$ and $p(y,y) = p(y,z)$. Using the equalities in PQ3, we have

$$\begin{aligned} p(x,z) &\leq p(x,y) + p(y,z) - p(y,y) \\ &= p(x,x). \end{aligned}$$

By PQ1, we have $p(x,x) = p(x,z)$. Therefore, $x \leq_p z$.

□

3.2.2 Definition. (Künzi, Pajoohesh, and Schellekens, 2006) The **open balls** of a partial quasi-metric space (X,p) are sets of the form

$$B_p(x,r) = \{y \in X : p(x,y) - p(x,x) < \epsilon,\ \epsilon > 0\}. \tag{3.2.2}$$

3.2.3 Theorem. *(Künzi, Pajoohesh, and Schellekens, 2006) The set of the open balls defined in* (3.2.2) *constitute a base for a T_0-topology τ_q on X.*

Proof. 1. First, let us prove that the collection of open balls on (X,p) is a base for a topology.

For each $x \in X$, let $\epsilon > 0$. If we have $\epsilon > 0, \delta > 0$ and $x, y \in X$ such that $B_p(x,\epsilon) \cap B_p(y,\delta) \neq \emptyset$, then there exists $z \in X$ and $\rho > 0$ such that $B_p(z,\rho)$ is an open ball in the intersection of the two open balls. Indeed, let z be an element of the intersection. Since $p(x,z) - p(x,x) < \epsilon$ and $(p,z) - p(y,y) < \delta$, we can define $B_p(z,\rho)$ where $\rho = \min\{\epsilon + p(x,x) - p(x,z), \delta + p(y,y) - p(y,z)\}$. Now, let $a \in B_p(z,\rho)$. Then,

$$p(z,a) - p(z,z) < \rho \leq \epsilon + p(x,x) - p(x,z). \tag{3.2.3}$$

By PQ4, we have

$$\begin{aligned} p(x,a) &\leq p(x,z) + p(z,a) - p(z,z) \\ &< p(x,z) + \epsilon + p(x,x) - p(x,z) \\ &< p(x,a) - p(x,x) < \epsilon. \end{aligned}$$

So, $a \in B_p(x,\epsilon)$ and therefore, $B_p(z,\rho) \subset B_p(x,\epsilon)$. Similarly, in the inequality (3.2.3), if we take $\rho \leq \delta + p(y,y) - p(y,z)$ then, we will get $B_p(z,\rho) \subset B_p(y,\delta)$. Consequently, $B_p(z,\rho) \subset B_p(x,\epsilon) \cap B_p(y,\delta))$.

2. Choose $\epsilon = \frac{1}{2}(p(x,x) + p(x,y))$. Then, since $p(x,x) < p(x,y)$, $x \in B_p(x,\epsilon)$ and $y \notin B_p(x,\epsilon)$. Therefore, the collection of open ball on (X,p) is a base for a T_0-topology on X.

□

Since in a PQM p, the symmetry condition does not hold, this allows us to define a backward topology which is the set

$$B_p(x,r) = \{y \in X : p(y,x) - p(x,x) < \epsilon,\ \epsilon > 0\}. \tag{3.2.4}$$

Thus, we have two topologies induced by p on X, the forward and backward topology. The latter has as basis the collection of open balls define in 3.2.4.

3.3 Equivalence with Weighted Quasi-metric

We complete this section with a result from Künzi, Pajoohesh, and Schellekens (2006).

3.3.1 Definition. A **weighted quasi-metric** space is a triple (X,q,ω) where X is a nonempty set, q is a quasi-metric on X and ω is a weight function defined by $\omega : X \to [0,\infty)$ and satisfies:

$$\omega(y) \leq q(x,y) + \omega(x) \tag{3.3.1}$$

whenever $x,y \in X$.

3.3.2 Theorem. *To each PQM p on a set X corresponds a quasi-metric q with compatible weight ω defined by $q(x,y) = p(x,y) - \omega(x)$ whenever $x,y \in X$ where $\omega(x) = p(x,x)$.*

Proof. We shall proof that q is a quasi-metric on X. Let $x,y,z \in X$.

Q1) By the definition of a weight function, $q(x,y) \geq 0$.

Q2)
$$\begin{aligned} q(x,z) &= p(x,z) - p(x,x) \\ &\leq p(x,y) + p(y,z) - p(y,y) - p(x,x) \\ &= p(x,y) - p(x,x) + p(y,z) - p(y,y) \\ &= q(x,z) + q(y,z) \end{aligned}$$

Q3) if $p(x,y) - p(x,x) = 0$ and $p(y,x) - p(y,y) = 0$, then $x = y$ by PQ4.

□

3.3.3 Remark. If (X,q,ω) is a weighted quasi-metric space with weight ω, then the associated PQM p is a partial metric if and only if q and ω satisfy:

$$q(x,y) + \omega(x) = q(y,x) + \omega(y), \tag{3.3.2}$$

whenever $x,y \in X$.

3.4 Banach Fixed Point Theorem

Completeness in PQMS is defined in the same way as for metric spaces. Thus, the definition of convergence and Cauchy sequence are similar to that of metric spaces. We refer the reader to Karapınar, Erhan, and Öztürk (2013) for the materials in this section.

As we defined the T_0-topology on the PQMS (X,p), we can start by defining the convergence of a sequence in this space. A sequence (x_n) in a PQMS (X,p) is convergent to x if for every $\varepsilon > 0$, there exists $N \in \mathbb{N}$ such that for all $n \geq N$, $x_n \in \mathcal{B}_p(x, \varepsilon)$.

Since we have forward and backward topology on (X,p), we have that x_n converges to x if for every ε, there exists $N \in \mathbb{N}$ such that for every $n \geq N$,

$$p(x, x_n) - p(x,x) < \varepsilon \quad \text{and} \quad p(x_n, x) - p(x,x) < \varepsilon$$

Therefore,

3.4.1 Definition. A sequence (x_n) in the PQMS (X,p) **converges** to $x \in X$ if and only if $p(x,x) = \lim_{n\to\infty} p(x,x_n) = \lim_{n\to\infty} p(x_n,x)$.

Let be q the weighted QM equivalent to the PQM p as defined in theorem 3.3.2. Then a sequence (x_n) is a Cauchy sequence in the PQMS p if and only if it is Cauchy in the induced metric space (X, q^s) of the weighted QM (X,q,ω). Therefore, (x_n) is a Cauchy sequence (X,p) if and only if for every $\varepsilon > 0$, there exists $N \in \mathbb{N}$, such that for all $n, m \geq N$,

$$\max\{p(x_n, x_m) - p(x_n, x_n), p(x_m, x_n) - p(x_n, x_n)\} < \varepsilon$$

Consequently, we have also

3.4.2 Definition. Let (X,p) be a PQMS and (x_n) a sequence in X. Then (x_n) is a **Cauchy sequence** if and only if for every $\varepsilon > 0$, there exists $N \in \mathbb{N}$, such that for all $n, m \geq N$,

$$p(x_n, x_m) - p(x_n, x_n) < \varepsilon \quad \text{and} \quad p(x_m, x_n) - p(x_n, x_n) < \varepsilon$$

Compare to the definition 2.3.15 of bicompleteness of a QMS, a PQMS is complete if the induced metric space (X,q) of the weighted QMS (X,q,ω) is complete. Hence, (X,p) is complete if every Cauchy sequence converges to a point $x \in X$ such that $p(x,x) = \lim_{n,m\to\infty} p(x_n,x_m) = \lim_{n,m\to\infty} p(x_m,x_n)$.

3.4.3 Lemma. (X,p) is complete if and only if the associated quasi-metric (X,q), where q is as defined in Theorem 3.3.2 is bicomplete.

Proof. If a Cauchy sequence converges in (X,q), then it follows directly from the definition of convergence in (X,p) that it converges too in (X,p). If (x_n) is a Cauchy sequence in (X,p) and has limit x, say, that is,

$$p(x,x) = \lim_{n,m\to\infty} p(x_n,x_m) = \lim_{n,m\to\infty} p(x_m,x_n)$$

then,

$$p(x,x) - \lim_{n,m\to\infty} p(x_n,x_m) = 0 \ \text{ and } \ p(x,x) - \lim_{n,m\to\infty} p(x_m,x_n) = 0,$$

and we get $\lim_{n,m\to\infty} q(x_n,x_m) = 0$ and $\lim_{n,m\to\infty} q(x_m,x_n) = 0$ by 3.3.2. Hence, (x_n) converges in (X,q). □

The definition of the contraction mapping remains the same in PQMS. That is, f is a contraction mapping on (X,p) if there exists a real constant $k \in [0,1)$ such that :

$$p(f(x), f(y) \leq kp(x,y). \tag{3.4.1}$$

3.4.4 Theorem. *(Künzi, Pajoohesh, and Schellekens, 2006) Let f be a contraction mapping on a PQMS (X, p) into itself. Then f has a unique fixed point . Moreover, if $x \in X$ is the fixed point of f, then $p(x, x) = 0$.*

Compare to 2.3.19. Let k be the contraction constant. Let x_0 be an element of X and $\{x_n\}$ the sequence defined as $x_n = f(x_{n-1})$. Since f is a contraction, we have:

$$p(x_{n+1}, x_n) \leq k^n p(x_1, x_0). \tag{3.4.2}$$

Now let $n < m \in \mathbb{N}$. By PQ3, we have:

$$\begin{aligned} p(x_m, x_n) &\leq p(x_m, x_{m-1}) + p(x_{m-1}, x_{m-2}) - p(x_{m-1}, x_{m-1}) + \cdots + \\ &\quad + p(x_{n+2}, x_{n+1}) + p(x_{n+1}, x_n) - p(x_{n+1}, x_{n+1}) \\ &\leq p(x_m, x_{m-1}) + p(x_{m-1}, x_{m-2}) + \cdots + +p(x_{n+2}, x_{n+1}) + p(x_{n+1}, x_n) \\ &\leq (k^{m-1} + k^{m-2} + \cdots + k^n) p(x_1, x_0) \\ &\leq \frac{k^n}{1-k} p(x_1, x_0) \end{aligned}$$

For a given $\epsilon > 0$, we can find $N \in \mathbb{N}$ such that

$$k^N < \epsilon \frac{1-k}{p(x_1, x_0)}.$$

Choosing $N < n$, we have $k^n < k^N$ and

$$p(x_m, x_n) < \epsilon. \tag{3.4.3}$$

Thus, (x_n) is a Cauchy sequence and there exists x^* such that

$$p(x^*, x^*) = \lim_{n,m \to \infty} p(x_n, x_m) = \lim_{n,m \to \infty} p(x_m, x_n). \tag{3.4.4}$$

So $\lim_{n,m \to \infty} p(x_m, x_n) = 0$. On the other hand, by PQ3 we have:

$$\begin{aligned} p(f(x^*), x^*) &\leq p(f(x^*), f(x_n)) + p(f(x_n), x^*) - p(f(x_n), f(x_n)) \\ &\leq cp(x^*, x_n) + p(f(x_n), x^*) \text{ by 3.4.1} \\ &\leq cp(x^*, x^*) + p(x^*, x^*) \text{ for a larger } n. \end{aligned}$$

Therefore, $p(f(x^*), x^*) = 0$ and $f(x^*) = x^*$, by PQ1.

If we suppose that there exists $x^\star$ such that $f(x^\star) = x^\star$, then, by PQ1, and using the fact that f is a k-contraction, we have:

$$p(x^*, x^\star) \leq kp((x^*, x^\star). \tag{3.4.5}$$

Thus $p(x^*, x^\star) = 0$ and $x^* = x^\star$. □

Furthermore, under the condition in theorem 3.4.4, we have the following result:

3.4.5 Proposition. (Cerdà-Uguet, Schellekens, and Valero, 2012) If there exists $y \in X$ such that $q(f(y), y) = 0$ then $q(x, y) = 0$ where q is the QM associated to the PQM p.

Proof. x is the fixed point of f, so $q(x,x) = 0$. Suppose that $q(x,y) \neq 0$, then $q(x,y) = p(x,y) > 0$. By PQ3, we have the inequality:

$$\begin{aligned}
p(x,y) &\leq p(x,f(x)) + p(f(x),y) - p(f(x),f(x)) \\
&= p(f(x),y) && \text{since } x \text{ is the fixed point} \\
&\leq p(f(x),f(y)) + (f(y),y) - p(f(y),f(y)) && \text{by applying PQ3 again} \\
&= p(f(x),f(y)) + q(f(y),y) \\
&= p(f(x),f(y)) \leq kp(x,y)
\end{aligned}$$

which is a contradiction since $k \in [0,1)$. □

4. Divide and Conquer and Recursive Algorithm

Throughout this section, $\mathbb{R}^+$, $\mathbb{N}$ and ω will denote respectively, the set of non-negative real numbers, the set of positive natural integers and the set of nonnegative integers numbers.

Divide and Conquer algorithm is an algorithm which that the same algorithm and combine the solutions at the end. Let $T(n)$ be the running time of such an algorithm with $n \in \mathbb{N}$ is the input size. Then, $T(n)$ satisfies the following recurrence equation:

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ aT(\frac{n}{b}) + h(n) & \text{else} \end{cases} \tag{4.0.1}$$

where $h(n)$ is the time requires to combine all the solutions, a is the number of subproblems of size $\frac{n}{b}$, $b \in \mathbb{N}$. Therefore, the optimal running of this algorithm is given by the solution of this recurrence equation. That is why the use of the Banach fixed point theorem.

Now, the question is in which space should we apply this aforementioned theorem to study complexity. We will show in the following section that the Baire PQMS on a set of words is a suitable tool to this end. For more details, the reader is referred to Cerdà-Uguet, Schellekens, and Valero (2012).

4.1 Illustrations

Suppose we have a complexity space $\mathcal{RT}$ and $f \in \mathcal{RT}$ a complexity function. For each input of f we can associate to a term of sequence x_n where n is the size of the input. Therefore, f can be represented by a sequence (x_n) that we shall call **word**. Each letter of the word, or the sequence, is an element of an **alphabet**. Therefore, we will construct a PQMS of a set of words.

Let $\sum$ be a non-empty alphabet endowed with partial order $\preceq$. Let $\sum^\infty$ be the set of all finite and infinite words over $\sum$. $l(x)$ denotes the length $x \in \sum^\infty$. And $l_\preceq(x, y) = \sup\{n \in \mathbb{N} : x_k \preceq y_k \text{ for all } k \leq n\}$. Moreover, we say that x is a subprefix of y if there exists $n_0 \leq l(x)$ such that $x_k \preceq y_k$ for all $k \leq n_0$, and we denote $x \sqsubseteq_{sp} y$.

4.1.1 Proposition. Let $\sum$ be an alphabet endowed with an order $\preceq$. Then $q_B : \sum^\infty \times \sum^\infty \to \mathbb{R}^+$ defined by $q_B(x, y) = 2^{-l_\preceq(x,y)}$ is a PQM.

Proof. PQ1) Either $l(x) \leq l(y)$ or $l(y) \leq l(x)$, $l_\preceq(x, y)$ is not greater than $l(x)$. Therefore, $2^{-l_\preceq(x,x)} \leq 2^{-l_\preceq(x,y)}$.

PQ2) Similarly, $2^{-l_\preceq(x,x)} \leq 2^{-l_\preceq(y,x)}$.

PQ3) By PO3, if there exists N_0 and N_1 such that $x_k \preceq y_k$ and $y_i \preceq z_i$ for all $k \leq N_0$ and for all $i \leq N_1$, then $x_k \preceq z_k$ for all $k \leq \min\{N_0, N_1\}$. Thus, $l_\preceq(x, z) = \min\{N_0, N_1\} \min\{l_\preceq(x, y), l_\preceq(y, z)\}$.

And $2^{-l_\preceq(x,z)} \leq 2^{l_\preceq(x,y)} + 2^{l_\preceq(y,z)} - 2^{l_\preceq(y,y)}$.

□

To be able to use this set as a tool to compute the asymtotic complexity of an algorithm, we should make sure that is complete. To this end, using equivalence between PQM and QM, we have the following proposition:

4.1.2 Proposition. $(\sum^\infty, q)$ is a complete QMS, where $q(x,y) = q_B(x,y) - q_B(x,x)$.

Proof. The theorem 3.3.2 proves that q is a QM.

Let (x_n) be a Cauchy sequence in $(\sum^\infty, q)$. Then by definition 2.3.12, (x_n) is a Cauchy in $(\sum^\infty, q^s)$. That is for every $\varepsilon > 0$, there exists $N \in \mathbb{N}$ such that for every $n, m \geq N$,

$$\max\{q(x_n, x_m), q(x_m, x_n)\} < \varepsilon \tag{4.1.1}$$

$$q(x_n, x_m) < \varepsilon \text{ and } q(x_m, x_n) < \varepsilon \tag{4.1.2}$$

$$2^{-l_{\preceq}(x_n, x_m)} - 2^{-l(x_n)} < \varepsilon \text{ and } 2^{-l_{\preceq}(x_m, x_n)} - 2^{-l(x_m)} < \varepsilon \tag{4.1.3}$$

Fixing m and let $n > m$, then $y_m = 2^{-l_{\preceq}(x_n, x_m)} - 2^{-l(x_n)} = 2^{-j_m} - 2^{-j_n}$ is a Cauchy sequence in $\mathbb{R}$. So, there exists $y \in \mathbb{R}$ such that $\lim_{m\to\infty} y_m = y$. That is

$$y = \lim_{m\to\infty} 2^{-l_{\preceq}(x_n, x_m)} - 2^{-l(x_n)} \tag{4.1.4}$$

and $y \in \sum^\infty$. Therefore, since x_n is fixed, there exists x such that for every $\varepsilon_1 > 0$, there exists N_1 such that for every $m > N_1$, $q(x_m, x) < \varepsilon_1$.

On the other side, we have

$$\begin{aligned} q(x_n, x) &\leq q(x_n, x_m) + q(x_m, x) \\ &< \varepsilon + q(x_m, x) - q(x_m, x_m) \end{aligned}$$

and letting $m \to \infty$ provides

$$q(x_n, x) < \varepsilon + \varepsilon_1 \tag{4.1.5}$$

So there exists x such that for every $\varepsilon_2 = \varepsilon + \varepsilon_1 > 0$, there exists N_1 such that for every $n > N_1$, $q(x_n, x) < \varepsilon_2$. Similarly, we fix m to show that there exists x which is ***u***-limit of (x_n). Therefore, (x_n) converges in $(\sum^\infty, q)$ .Finally,$(\sum^\infty, q)$ is bicomplete. □

4.1.3 Proposition. $(\sum^\infty, q_B)$ is a complete PQMS.

Proof. Theorem 3.4.3 and the previous one tell us that $(\sum^\infty, q_B)$ is complete. □

Now, according to Schellekens (1995), we make use of a functional as a contraction on the PQMS $(\sum^\infty, q_B)$ to apply the theorem 3.4.4 in order to ensure the convergence of the recurrence equation of form 4.0.1.

4.1.4 Theorem. *Let $\sum = [0, \infty)$. Fix $c \in \sum$ and $z \in \sum^\infty$ with $l(z) = \infty$ and $z_k \neq \infty$ for all $k \in \omega_b = \{b^k : k \in \mathbb{N}\}$ with $k \geq 2$. Let $\sum^\infty_{b,c}$ be the subset of $\sum^\infty$ given by $\sum^\infty_{b,c} := \{y \in \sum^\infty : 2 \leq l(y)$ and $y_1 = c, y_k = \infty$ for all $k \notin \omega_b$ with $2 \leq k \leq l(y)\}$. Then, the mapping $\Theta^z_{b,c} : \sum^\infty_{b,c} \to \sum^\infty_{b,c}$ defined by $\Theta^z_{b,c} = x_{\Theta^z_{b,c}}$, where*

$$(x_{\Theta^z_{b,c}})_k := \begin{cases} c & \text{if } k = 1 \\ \infty & \text{if } k \notin \omega_k \text{ and } 2 \leq k \leq l(x) + 1 \\ a x_{\frac{k}{b}} + z_k & \text{if } k \in \omega_k \text{ and } \frac{k}{b} \leq l(x) \end{cases} \tag{4.1.6}$$

has a unique fixed point $v \in \sum^\infty_{b,c}$ with $l(v) = \infty$. Moreover if $u \in \sum^\infty_{b,c}$ such that $\Theta^z_{b,c}(u) \sqsubseteq_{sp} u$ then $v \sqsubseteq u$.

We will see that the last statement allow us to find the upper bound of the asymptotic running time of an algorithm.

Proof. We saw that $(\sum^\infty, q_B)$ is complete space. Since, $\sum_{b,c}^\infty$ is a subset of $\sum^\infty$, $\sum_{b,c}^\infty$ needs to be closed to be complete together with the restriction of the PQM q_B denoted as $q_B|_{\sum_{b,c}^\infty}$. Let (x_n) be a sequence in $\sum_{b,c}^\infty$ such that it converges to x in $\sum^\infty$. That is $q_B(x,x) = \lim_{n\to\infty} q_B(x_n, x) = \lim_{n\to\infty} q_B(x, x_n)$. Therefore , we have that , as $n \to \infty$, $l_\preceq(x_n, x) = l(x)$ and $l_\preceq(x, x_n) = l(x)$. So, $(x_n)^1 = x^1 = c$ and $(x_n)^k = x^k$ for all $k \leq l(x)$. Thus, (x_n) converges in $\sum_{b,c}^\infty$ and $(\sum_{b,c}^\infty, q_B|_{\sum_{b,c}^\infty})$ is a complete PQMS.

Let $x' = x_{\Theta_{b,c}^z}$ and $y' = y_{\Theta_{b,c}^z}$. If $k = 1$ or $k \notin w_b$, we always have $x' \preceq y'$. Else, if we suppose $L = l_\preceq(x,y)$, then for all $\frac{k}{b} \leq \frac{L}{b}$, we have $x' \preceq y'$. Once, the index is greater than $\frac{L}{b}$, we will always have the previous case. Therefore, $l_\preceq(x', y') = L$ and $l_\preceq(x', y') < L + 1$ which will provide that

$$2^{-l_\preceq(x',y')} \leq 2^{-l_\preceq(x,y)-1} = \frac{1}{2}2^{-l_\preceq(x,y)} \tag{4.1.7}$$

Hence, $\Theta_{b,c}^z$ is a contraction mapping with contraction factor $\frac{1}{2}$.

Since, $(\sum_{b,c}^\infty, q_B|_{\sum_{b,c}^\infty})$ is a complete PQMS, then there exists a fixed point $v \in \sum_{b,c}^\infty$ with $q_B(v,v) = 0$ implying $l_\preceq(v,v) = l(v,v) = \infty$.

Suppose there exists $u \in \sum_{b,c}^\infty$ such that $\Theta_{b,c}^z(u) \sqsubseteq_{sp} u$. So, by definition of $\Theta_{b,c}^z$, $l(u) = \infty$ and $q_B(\Theta_{b,c}^z(u), u) = 0$. If we suppose that $q(u,v) \neq 0$, then

$$\begin{aligned}
q_B(v,u) &\leq q_B(v, \Theta_{b,c}^z(v)) + q_B(\Theta_{b,c}^z(v), v) - q_B(\Theta_{b,c}^z(v), \Theta_{b,c}^z(v)) \\
&= q(\Theta_{b,c}^z(v), u) \\
&\leq q_B(\Theta_{b,c}^z(v), \Theta_{b,c}^z(u)) + q_B(\Theta_{b,c}^z(u), v) - q_B(\Theta_{b,c}^z(u), \Theta_{b,c}^z(u)) \\
&= q_B(\Theta_{b,c}^z(v), \Theta_{b,c}^z(u)) + q(\Theta_{b,c}^z(u), u) \\
&= q_B(\Theta_{b,c}^z(v), \Theta_{b,c}^z(u)) \leq \tfrac{1}{2} q_B(v,u)
\end{aligned}$$

Thus, $v \sqsubseteq u$. □

4.2 Existence of Solutions of Divide and Conquer

Let $\mathcal{RT}$ be the set of running time of algorithm. Since we set the set of words, we can identify each words as the running time given by a function $f \in \mathcal{RT}$. Indeed, for $x \in \sum^\infty$ and $f \in \mathcal{RT}$, we can set the following correspondence:

$$x_n \Leftrightarrow f(n) \text{ for all } n \in \mathbb{N}$$

Particularly, we are interested to the recurrence equation of form 4.0.1 and according to the arguments stated in the theorem 4.1.4, we define the following set:

$$\mathcal{RT}_{b,c} := \{f \in \mathcal{RT} : f(1) = c \text{ and } f(n) = \infty \text{ for all } n > 1 \notin \omega_b\}$$

Since, we have correspondence between the set of words and the set of running time functions, then we set similarly the PQM on $\mathcal{RT}_{b,c}$. Define the mapping Φ_T from $\mathcal{RT}_{b,c}$ into itself as:

$$\Phi(f)(n) = \begin{cases} c & \text{if } k = 1 \\ \infty & \text{if } k \notin \omega_k \text{ and } 2 \leq k \leq l(x) + 1 \\ af(\frac{n}{b}) + h(n) & \text{otherwise} \end{cases} \tag{4.2.1}$$

Then, we have the following result:

4.2.1 Corollary. (Cerdà-Uguet, Schellekens, and Valero, 2012) A Divide and Conquer algorithm satisfying the recurrence 4.0.1 has a unique solution $f_T \in \mathcal{RT}_{b,c}$. Moreover if there exists $g \in \mathcal{RT}_{b,c}$ such that Φ_T is an improver with respect to g, then $f_T \in \mathcal{O}(g)$

Proof. Respecting the same setting as in the previous section allows us to find directly that there exists a fixed point f_v, where v is the fixed point of the mapping defined in the theorem 4.1.4. Therefore, f_v is the running time of the Divide and Conquer algorithm.

And if there exists g such that Φ_T is an improver with respect to g, then g is the upper bound of f_v. Indeed, g is identified as u in theorem 4.1.4, so $f_v \in \mathcal{O}(g)$. □

We will show that these techniques can be applied to analyse the asymptotic complexity of other algorithm.Generally, the running time of a recursive algorithm satisfies:

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ T(n-1) + h(n) & \text{else} \end{cases} \tag{4.2.2}$$

where $h(n)$ is the running time of combining all the solutions. So this type of algorithm is not far from the Divide and Conquer. Sometimes, we might find a coefficient $a \in \mathbb{R}^+$ for the case $n \geq 2$, i.e. $aT(n-1) + h(n)$. In this work, we will take the case $a = 1$. This recurrence form can be generalized to :

$$T(n) = \begin{cases} c_n & \text{if } 1 \leq n \leq k \\ \sum_{i=1}^{k} a_i T(n-i) + h(n) & \text{else} \end{cases} \tag{4.2.3}$$

So, we can study the case $a \geq 1$ of a recursive algorithm by the use of this recurrence form with $k = 1$. Notice that the recurrence form of the Divide and Conquer algorithm can also be expressed by a similar recurrence form of recursive algorithm as following (Alghamdi, Shahzad, and Valero, 2014):

$$S(m) = \begin{cases} c & \text{if } m = 1 \\ aS(m-1) + r(m) & \text{else} \end{cases} \tag{4.2.4}$$

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

4.3 Existence of Solution of Recursive Algorithm

To study the asymptotic complexity of a recursive algorithm, we will use the same method as for Divide and Conquer. In other words, we will set the complexity space, the associated set of words, the associated functional, and finding a complexity function such that the functional is an improver with respect to this function. The computation will be shown in the next chapter for particular programs. For this section, the reader is referred to Cerdà-Uguet, Schellekens, and Valero (2012). The same result is stated in Romaguera, Tirado, and Valero (2012) by the use of quasi-metric space, an extension of the developed theory of Schellekens (Schellekens, 1995).

The complexity space associated to the recurrence 4.2.2 is the space $\mathcal{RT}_c = \{f \in \mathcal{R} : f(1) = c\}$.

The associated functional $\Gamma_T : \mathcal{RT}_c \to \mathcal{RT}_c$ is defined by:

$$\Gamma_T(f)(n) = \begin{cases} c & \text{if } n = 1 \\ f(n-1) + h(n) & \text{else} \end{cases} \tag{4.3.1}$$

for $f \in \mathcal{RT}_c$. Similarly, we identify each complexity function f by a word $x \in \sum_c^\infty$ so that $x_n = f(n)$. And the correspondent set of words is $\sum_c^\infty = \{y \in \sum^\infty : y_1 = c \text{ and } 2 \leq l(y)\}$ which is a subset of $\sum^\infty$. In the Baire PQMS on the set of words, we fix the words z representative of the function $h(n)$ such that $l(z) = \infty$ and $z_k \neq \infty$ for all $k \in \mathbb{N}$.

4.3.1 Theorem. *The mapping $\Psi^z : \sum_c^\infty \to \sum_c^\infty$ defined by $\Psi^z(x) = x_{\Psi^z}$ where*

$$(x_{\Psi^z})_k = \begin{cases} c & \text{if } k = 1 \\ x_{k-1} + z_k & \text{if } 2 \leq k \leq l(x) + 1 \end{cases}$$

has a unique fixed point $v \in \sum_c^\infty$ with $l(v) = \infty$. Moreover, if there exists $u \in \sum_c^\infty$ such that $\Psi^z(u) \sqsubseteq_{sp} u$, then $v \sqsubseteq u$.

Proof. i) Firstly, we will show that $\sum_c^\infty$ is close in $(\sum^\infty, q_B)$. Let (x_n) be a sequence in $\sum_c^\infty$ that converges to $x \in \sum^\infty$. Then we have that for every $\varepsilon > 0$, there exists $N \in \mathbb{N}$ such that for every $n \geq N$, we have $2^{-l_{\preceq}(x_n,x)} < \varepsilon$ and $2^{-l_{\preceq}(x,x_n)} < \varepsilon$. Consequently, $x_k \preceq x_n^k$ and $x_n^k \preceq x_k$ for all $k \in \mathbb{N}$, so $x_k = x_n^k$ for every $k \in \mathbb{N}$. Therefore $x_n^1 = c = x_1$ and $x \in \sum_c^\infty$. we conclude that $\sum_c^\infty$ is close in the PQMS $(\sum^\infty, q_B)$ and $(\sum_c^\infty, q_B)$ is a complete PQMS.

ii) Secondly, we will show that Ψ^z is a contraction mapping. Let $x, y \in \sum_c^\infty$ and $L = l_{\preceq}(x, y)$. By the definition of Ψ^z, $l_{\preceq}(\Psi^z(x), \Psi^z(y) = l_{\preceq}(x, y)$. Therefore, $2^{-l_{\preceq}(\Psi^z(x),\Psi^z(y)} = \frac{1}{2}2^{-l_{\preceq}(x,y)}$. Ψ^z is a contraction mapping on $\sum_c^\infty$ with $\frac{1}{2}$ as a contraction factor.

From these two points, we conclude that Ψ^z has a unique fixed point v by theorem 3.4.4.

iii) As v is a fixed point and by the first point, $v \in \sum_c^\infty$, then $q_B(v, v) = 0$ by theorem 3.4.4. Suppose there exists $u \in \sum_c^\infty$ such that $\Psi^z(u) \sqsubseteq_{sp} u$. By definition of Ψ^z, $l(\Psi^z(u)) \geq l(u)$. So, $l_{\preceq}(\Psi^z(u), u) = \infty$ and $q_B(\Psi^z(u), u) = 0$. Finally, by the proposition 3.4.5, we have that $q_B(v, u) = 0$ which implies $l_{\preceq}(v, u) = \infty$, i.e. $v_k \preceq u_k$ for all $k \in \mathbb{N}$. Hence, $v \sqsubseteq_{sp} u$.

□

Coming back to the complexity space $\mathcal{RT}_c$, this theorem 4.3.1 is stated as:

4.3.2 Corollary. A recurrence of form 4.2.2 has a unique solution $f_T \in \mathcal{RT}_c$. Moreover if there exists a complexity function $g \in \mathcal{RT}_c$ such that Ψ^z is an improver with respect to g, then $f_T \in \mathcal{O}(g)$.

Proof. For a given $f \in \mathcal{RT}_c$, we can identify it as a word $x^f \in \sum_c^\infty$ by $f(n) = x^f(n)$. Therefore, the functionnal Γ_T is just the version of the functional Ψ^z in the complexity space. Let u be the fixed point of the mapping Ψ^z by the theorem 4.3.1. We define a complexity function $f \in \mathcal{RT}_c$ as $f(n) = u_n$. Then, f is the fixed point in $\mathcal{RT}_c$, say f_T.

Assume there exists $g \in \mathcal{RT}$ such that $\Gamma_T(g) \leq g$. In the set of words, we have $\Psi^z(x^g) \sqsubseteq_{sp} x^g$ where $x^g(n) = g(n)$. Immediately, by theorem 4.3.1, $u \sqsubseteq_{sp} x^g$. Equivalently, in the complexity space, we have $f_T \in \mathcal{O}(g)$. □

4.4 Existence of Solution of a More General Recursive Algorithm

Similarly, to analyze the complexity of a non-recursive algorithm, we will set the corresponding complexity space, the associated functional, the set of words that will be endowed by a PQMS. These will guarantee the existence of the optimal running time of a non-recursive algorithm. The computation of the complexity class will be shown in the next chapter. For this section, the reader is referred to Alghamdi, Shahzad, and Valero (2014).

The complexity associated to an algorithm satisfying the recurrence form 4.2.3 is $\mathcal{RT}_{c,k} = \{f \in \mathcal{RT} : f(m) = c_m$ for all $1 \leq m \leq k\}$.

The functional associated to the recurrence 4.2.3 defined in $\mathcal{RT}_{c,k}$ into itself is given by:

$$\Xi_T(f)(n) = \begin{cases} c_n & \text{if } 1 \leq n \leq k \\ \sum_{i=1}^{k} a_i f(n-i) + h(n) & \text{else} \end{cases} \tag{4.4.1}$$

Given a complexity function $f \in \mathcal{RT}_{c,k}$, we can associate it to a word x where $x_n = f(n)$. Thus, the set of words associated to the complexity space is $\sum_{c,k}^{\infty} = \{y \in \sum^{\infty} : k \leq l(y)$ and $y_m = c_m$ for all $1 \leq m \leq k\}$. Then in $\sum_{c,k}^{\infty}$, we fix z which identifies $h(n)$ in the recurrence form 4.2.3 and $c = c_1 c_2 \ldots c_k$. The definitions of subprefix, length of words and PQM q_B yields the same as in the section of Divide and Conquer algorithm.

4.4.1 Lemma. $\sum_{c,k}^{\infty}$ is closed in $(\sum^{\infty}, d_{q_B}^s)$ where d_{q_B} is the QM associated to q_B as defined in theorem 3.3.2 and $d_{q_B}^s$ is the induced metric of the QM d_{q_B}.

Proof. Let (x_n) be a sequence in $\sum_{c,k}^{\infty}$ that converges in $(\sum^{\infty}, q_B)$. Firstly, suppose that $l(x) < k$. Then we can define $\varepsilon = 2^{-l(x)} - 2^{-k} > 0$ and there exists n_0 such that for all $n \geq n_0$, $2^{-l_{\preceq}(x_n,x)} - 2^{-l_{\preceq}(x_n,x_n)} < \varepsilon$. Alternatively, we have $k \leq l(x_n)$ as $(x_n) \in \sum_{c,k}^{\infty}$ and $l_{\preceq}(x_n, x) \leq l(x_n)$. Therefore,

$$2^{-l(w)} - 2^{-k} \leq 2^{-l_{\preceq}(x_n,x)} - 2^{-l_{\preceq}(x_n,x_n)} < \varepsilon$$

Which is a contradiction. Hence, $k \leq l(x)$.

Secondly, suppose that there exists $k_O < k$ such that $x_m = c_m$ for all $m \leq k_0$ and $x_{k_0+1} \neq c_{k_0+1}$. We have two cases:

- Case 1: $x_{k_0+1} \preceq c_{k_0+1}$. As x limit of (x_n), then $l_{\preceq}(x, x_n) = k_0$. Let $\varepsilon = 2^{-k_0} - 2^{-l(x)}$. Then, there exists $n_0 \in \mathbb{N}$ such that for every $n \geq n_0$, $2^{-l_{\preceq}(x,x_n)} - 2^{-l_{\preceq}(x,x)} < \varepsilon$. Since (x_n) converges to x,

$$2^{-k_0} - 2^{-l(x)} = 2^{-l_{\preceq}(x,x_n)} - 2^{-l(x)} < \varepsilon$$

.

- Case 2: $c_{k_0+1} \preceq x_{k_0+1}$. Let $\varepsilon = 2^{-k}$. Then there exists n_0 such that for every $n \geq n_0$, $2^{-l_{\preceq}(x_n,x)} - 2^{-l_{\preceq}(x_n,x_n)} < \varepsilon$, also

$$2^{-k_0} - 2^{-l_{\preceq}(x_n,x_n)} < 2^{-k}$$

which implies $2^{-k_0} < 2^{-k} + 2^{-l(x_n)}$. Then

$$2^{-k_0} < 2^{-k} + 2^{-l(x_n)} \leq 2 \times 2^{-k}$$

Therefore, $k + 1 \leq k_0 \leq k$ which is a contradiction.

Finally, $x_m = c_m$ for all $1 \leq m \leq k$ and $x \in \sum_{c,k}^{\infty}$. Whence, $\sum_{c,k}^{\infty}$ is closed in $(\sum^{\infty}, d_{q_B}^s)$.

□

By the lemma 3.4.3 and the proposition 4.1.3, $(\sum^\infty, d_{q_B})$ is bicomplete. By proposition 2.3.16, $(\sum^\infty, d^s_{q_B})$ is complete. Therefore $(\sum^\infty_{c,k}, d^s_{q_B})$ is complete. By the proposition 2.3.16, $(\sum^\infty_{c,k}, d_{q_B})$ is bicomplete. Hence, by the lemma 3.4.3, $(\sum^\infty_{c,k}, q_B)$ is a complete PQMS.

4.4.2 Theorem. *Under these settings, the functional* $\Lambda^z : \sum^\infty_{c,k} \to \sum^\infty_{c,k}$ *defined by* $\Lambda^z(x) = x_{\Lambda^z}$ *where*

$$(x_{\Lambda^z})_k = \begin{cases} c_m & \text{if } 1 \leq m \leq k \\ \sum_{i=1}^k a_m x_{m-i} + z_k & \text{if } k+1 \leq m \leq l(x)+1 \end{cases} \tag{4.4.2}$$

has a unique fixed point $w \in \sum^\infty_{c,k}$ *with* $l(w) = \infty$*. Moreover, if there exists* $u \in \sum^\infty_{c,k}$ *such that* $\Lambda^z(u) \sqsubseteq_{sp} u$*, then* $w \sqsubseteq_{sp} u$*.*

Proof. i) Firstly, let prove that Λ^z is a contraction mapping on the PQMS $(\sum^\infty_{c,k}, q_B)$. Let $u, v \in \sum^\infty_{c,k}$ and $L = l_\preceq(u,v)$. By definition of Λ^z, $l(\Lambda^z(u), \Lambda^z(v)) = L+1$. Therefore,

$$q_B(\Lambda^z(u), \Lambda^z(v)) = 2^{-L+1} = \frac{1}{2}2^{-L} = q_B(u,v)$$

We can write also, $q_B(\Lambda^z(u), \Lambda^z(v)) \leq \frac{1}{2} q_B(u,v)$. Whence, Λ^z is a contraction mapping with $\frac{1}{2}$ as a contraction factor.

As $(\sum^\infty_{c,k}, q_B)$ is a complete PQMS, Λ^z has a unique fixed point w by theorem 3.4.4 and $q_B(w,w) = 0$

ii) Secondly, suppose there exists u such that $\Lambda^z(u) \sqsubseteq_{sp} u$. Since $l(\Lambda^z(u)) \geq l(u)$, so $l_\preceq(\Lambda^z(u), u) = \infty$ implying $q_B(\Lambda^z(u), u) = 0$. By proposition 3.4.5, $q_B(w,u) = 0$, therefore $l_\preceq(w,u) = \infty$ and $w \sqsubseteq_{sp} u$.

□

Observe that the theorem 4.3.1 is a verxion of this theorem 4.4.2 with $a = 1$ and $k = 1$. Therefore, the Divide and Conquer algorithm is included in this recurrence form.

Coming back to the complexity space $\mathcal{RT}_{c,k}$, we have the following result:

4.4.3 Corollary. A recurrence equation of form 4.2.3 has a unique solution $f_T \in \mathcal{RT}_{c,k}$. Moreover if there exists a complexity function $g \in \mathcal{RT}_{c,k}$ such that Ξ_T is a improver with respect to g, then $f_T \in \mathcal{O}(g)$.

Proof. Since we have established a correspondence between the PQMS set of words $\sum^\infty_{c,k}$ and the complexity space $\mathcal{RT}_{c,k}$ by $x^f_n = f(n)$ for $f \in \mathcal{RT}_{c,k}$ and $x^f \in \sum^\infty_{c,k}$. Then, by the theorem 4.4.2, a unique fixed point u of the functional Λ^z exists. Therefore we can associate u to a function $f_T \in \mathcal{RT}_{c,k}$ which is the fixed point of the functional Ξ_T.

Moreover, if there exists $g \in \mathcal{RT}_{c,k}$ such that $\Xi(g) \leq g$, it can be translated in the set of words as $\Lambda^z(x^g) \sqsubseteq_{sp} x^g$ where $x^g_n = g(n)$. Then by the theorem 4.4.2, $u \sqsubseteq_{sp} x^g$ which means $f_T \in \mathcal{O}(g)$. □

5. Applications: Computing Complexity Classes

5.1 Divide and Conquer Algorihm :Mergesort and Quicksort

Now, we will show that all the previous arguments and settings are applicable to reprove asymptotic complexity analysis of programs satisfying the recurrence equation 4.0.1. Here, we will compute Mergesort and Quicksort complexity class. These examples are found in Cerdà-Uguet, Schellekens, and Valero (2012).

5.1.1 Example (Mergesort)**.** It is an example of Divide and Conquer algorithm. Indeed, it divides a given problem of size n into 2 subproblems of size $\frac{n}{2}$, and to these subproblems is applied the same algorithm. In the average and best case, its running time $T(n)$ satisfies the following recurrence:

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ 2T(\frac{n}{2}) + \frac{n}{2} & \text{if } n \in \omega_2. \end{cases} \tag{5.1.1}$$

Define the following mapping in $\mathcal{RT}_{2,c}$:

$$g^r_{\lg}(n) = \begin{cases} c & \text{if } k = 1 \\ \infty & \text{if } k \notin \omega_2 \text{ and } 2 \leq k \\ rn\lg(n) & \text{otherwise} \end{cases} \tag{5.1.2}$$

where $\lg = \log_2$.

The functional associated to the recurrence 5.1.1 is :

$$\Phi(g^r_{\lg})(n) = \begin{cases} c & \text{if } k = 1 \\ \infty & \text{if } k \notin \omega_2 \text{ and } 2 \leq k \\ 2r\frac{n}{2}\lg\frac{n}{2} + \frac{n}{2} & \text{otherwise} \end{cases} \tag{5.1.3}$$

Obviously, Φ is monotone. Indeed if $r_1 < r_2$ then $\Phi(g^{r_1}_{\lg}) < \Phi(g^{r_2}_{\lg})$. Moreover, Φ is a contraction mapping with constant contraction $\frac{1}{2}$. Now, to find the least upper bound $g^r_{\lg}$ of the running time complexity of the algorithm, we have to find r such that $\Phi(g^r_{\lg})(n) \leq g^r_{\lg}(n)$ for $n \in \omega_2$. Since $n \in \omega_2$, we have then:

$$\begin{aligned} \Phi(g^r_{\lg})(n) \leq g^r_{\lg}(n) &\Leftrightarrow 2r\frac{n}{2}\lg\frac{n}{2} + \frac{n}{2} \leq rn\lg(n) \\ &\Leftrightarrow 2r2^{k-1}(k-1) + 2^{k-1} \leq r2^k k \\ &\Leftrightarrow 2r(k-1) + 1 \leq 2rk \\ &\Leftrightarrow -2r + 1 \leq 0 \\ &\Leftrightarrow \frac{1}{2} \leq r \end{aligned}$$

Therefore, its time complexity is $\mathcal{O}(g^{\frac{1}{2}}_{\lg})$. That is $\mathcal{O}(\frac{1}{2}n\lg n)$.

5.1.2 Example (Mergesort)**.** In the worst case, the running time of `Mergesort` in the worst case satisfies the following recurrence:

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ 2T(\frac{n}{2}) + \frac{n}{2} & \text{if } n \in \omega_2. \end{cases} \tag{5.1.4}$$

Comparing to 5.1.1, we can set the same mapping as in 5.1.2 in $\mathcal{RT}_{2,c}$. And we define the following functional:

$$\Phi(g^r_{\lg})(n) = \begin{cases} c & \text{if } k = 1 \\ \infty & \text{if } k \notin \omega_2 \text{ and } 2 \leq k \\ 2r\frac{n}{2}\lg\frac{n}{2} + n - 1 & \text{otherwise} \end{cases} \tag{5.1.5}$$

Firstly, Φ is monotone since if we have $r_1 < r_2$, then $\Phi(g^{r_2}_{\lg})(n) < \Phi(g^{r_2}_{\lg})(n)$. Secondly, Φ is a contraction mapping. So, we have a fixed point $g^r_{\lg}$. Finally, the function $g^r_{\lg}$ is the least upper bound of the running time complexity (solution of the recurrence) if Φ is an improver with repsect to $g^r_{\lg}$. So, we have to find r by the following:

$$\begin{aligned} \Phi(g^r_{\lg})(n) \leq f^r_{\lg}(n) &\Leftrightarrow 2r\frac{n}{2}\lg\frac{n}{2} + n - 1 \leq rn\lg n \\ &\Leftrightarrow 2r2^{k-1}(k-1) + 2^k - 1 \leq r2^k k \\ &\Leftrightarrow r(k-1) + 1 - \frac{1}{2^k} \leq 0 \\ &\Leftrightarrow r \geq 1 - \frac{1}{2^k} \end{aligned}$$

Since we want to find the asymptotic behaviour, $r \geq 1$. Therefore, the solution is bounded by the function $g^1_{\lg}$. That is the class complexity of the running time is $\mathcal{O}(n\lg n)$.

5.1.3 Example (Quicksort in the best case behaviour)**.** Quicksort algorithm selects a pivot and divide the problems into two subproblems such that the elements of the first partition is less or equal than the pivot and the elements of the other partition are greater than the pivot. In the best case, its running time satisfies the following recurrence:

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ 2T(\frac{n}{2}) + dn & \text{if } n \in \omega_2. \end{cases} \tag{5.1.6}$$

where $d \in \mathbb{R}^+ \setminus \{0\}$.

And we define the set $\mathcal{RT}_{2,c}$ as previously. Therefore, the mapping defined in 5.1.2 still useful. Let Φ the functional:

$$\Phi(g^r_{\lg})(n) = \begin{cases} c & \text{if } k = 1 \\ \infty & \text{if } k \notin \omega_2 \text{ and } 2 \leq k \\ 2r\frac{n}{2}\lg\frac{n}{2} + dn & \text{otherwise} \end{cases} \tag{5.1.7}$$

It keeps the same properties. And to find the least upper bound of the solution of the recurrence, we have to find a function such that Φ is improver with respect to it. For $n \in \omega_2$, we have:

$$\begin{aligned} \Phi(g^r_{\lg})(n) \leq (g^r_{\lg})(n) &\Leftrightarrow 2r\frac{n}{2}\lg\frac{n}{2} + dn \leq 2rn\lg n \\ &\Leftrightarrow 2r2^{k-1}(k-1) + d2^k \leq 2r2^k k \\ &\Leftrightarrow r(k-1) + d \leq rk \\ &\Leftrightarrow -r + d \leq 0 \\ &\Leftrightarrow r \geq d \end{aligned}$$

Therefore, the least upper bound of the solution is $g^d_{\lg}$. Hence, the complexity class of the running time is $\mathcal{O}(dn\lg n)$

5.2 Recursive Algorithm: Quicksort and Largetwo

To apply concretly the corollary 4.4.2, we will give two examples of recursive algorithm whose running time satisfies the recurrence form 4.2.2. These are Quicksort in the worst case behaviour and Largetwo in the average case behaviour. They are found in Cerdà-Uguet, Schellekens, and Valero (2012).

5.2.1 Example. In the worst case behaviour, the running time $T(n)$ of Quicksort satisfies:

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ T(n-1) + jn & \text{else} \end{cases} \tag{5.2.1}$$

where $j \in \mathbb{R}^+$.

So, Quicksort in the worst case behaviour is a recursive algorithm of form **??**. So the existence and uniqueness of the running time of such recurrence form is ensured by the corollary 4.3.

Now, we will look for its complexity class. To do so, define the complexity function:

$$g_r(n) = \begin{cases} c & \text{if } n = 1 \\ rn^2 & \text{else} \end{cases}$$

And the functional associated to 5.2.1 and taking argument as g_r is given by:

$$\Gamma_T(g_r(n) = \begin{cases} c & \text{if } k = 1 \\ r(n-1)^2 + jn & \text{else} \end{cases} \tag{5.2.2}$$

g is the complexity class that we look for if $\Gamma_T(g_r)(n) \leq g(n)$ for all $n \in \mathbb{N}$. Equivalently for $n \geq 3$,

$$r(n-1)^2 + jn \leq rn^2 \Leftrightarrow \frac{jn}{2n-1} \leq r$$

But as n increases, $\frac{jn}{2n-1}$ decreases, so for $n = 3$, we have the greastest lower bound for r which is $\frac{3j}{5}$. In the other hand, we have for $n = 2$

$$c + 2j \leq 4r \Leftrightarrow \frac{c+2j}{4}$$

Therefore, the running time of Quicksort in the worst case behaviour is in the complexity class $\mathcal{O}(g_k)$ where $k = \max\{\frac{c+2j}{4}, \frac{3j}{5}\}$.

5.2.2 Example. Another example of recursive algorithm is **Largetwo** whose running time in the average case behaviour satisfies:

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ T(n-1) + 2 - \frac{1}{n} & \text{else} \end{cases} \tag{5.2.3}$$

The existence of the solution and its uniqueness is ensured by the corollary 4.3. In order to find the complexity class of the running time of Largetwo, we define:

$$g_r(n) = \begin{cases} c & \text{if } n = 1 \\ 2r(n-1) - r\lg n + d & \text{else} \end{cases}$$

Similar case as in the case of Quicksort of worst case behaviour happens. Indeed, g_r is the complexity class if

$$\Gamma(g_r)(n) \leq g_r(n) \Leftrightarrow \frac{2n-1}{2n + n\lg(n-1) - n\lg n}$$

Therefore, to find the class complexity, we are interested in the case $n = 2$ and $n = 3$. In this case, we will get the greatest lower bound for r.

If $n = 2$,

$$c + 2 - \frac{1}{2} \leq 2r - r\lg 2 + rd \Leftrightarrow \frac{2c+1}{2+2d} \leq r$$

If $n = 3$,

$$r(1+d) + 2\frac{1}{3} \leq 4r - r\lg 3 + rd \Leftrightarrow \frac{5}{3(3-\lg 3)}$$

Hence, the running time of the Largetwo in the average case is in $\mathcal{O}(g_k)$ with $k = \max\{\frac{2c+1}{2+2d}, \frac{5}{3(3-\lg 3)}\}$

5.3 Other Recursive Algorithm: Hanoi and Fibonacci

In order to show the applicability of the theorem 4.4.2, we will retrieve the well-known upper bound of running of Hanoi and Fibonacci, i.e. their complexity classes. These example are found in Alghamdi, Shahzad, and Valero (2014).

5.3.1 Example. The **Hanoi** algorithm is the algorithm to solve the Towers of Hanoi puzzle. Its running time complexity satisfies the recurrence equation:

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ 2T(n-1) + d & \text{else} \end{cases} \tag{5.3.1}$$

So, it is a particular case of 4.2.3 with $k = 1$ and $a_1 = 2$.So our complexity space is $\mathcal{RT}_{c,1}$. The functional associated to this recurrence form is given by:

$$\Xi(f)(n) = \begin{cases} c & \text{if } n = 1 \\ 2f(n-1) + d & \text{else} \end{cases} \tag{5.3.2}$$

for $f \in \mathcal{RT}_{c,1}$.

The corollary 4.4.3 ensures the existence and uniqueness of the fixed point of Ξ. Alternatively, we have the promised running time complexity of Hanoi, say f, which is the solution of the recurrence 5.3.1. And if there exists $g \in \mathcal{RT}_{c,1}$ such that $\Xi(g)(n) \leq g(n)$ for every $n \in \mathbb{N}$, then g is the upper bound of f or the class complexity of f. If $n = 1$, then $g(1) = c = \Xi(g(1))$.

If $n = 2$, then $2c + d \leq g(2)$.

If $n = 3$, then $2^2c + 2^2d - d \leq g(3)$. By induction, we have

$$2^{n-1}(d + c) - d \leq g(n)$$

.

Therefore, the running time complexity of Hanoi is in $\mathcal{O}(2^{n-1})$.

5.3.2 Example. The running time of Fibonacci satisfies the recurrence:

$$T(n) = \begin{cases} 2c & \text{if } n = 1 \\ 3c & \text{if } n = 2 \\ T(n-1) + T(n-2) + 4c & \text{else} \end{cases} \tag{5.3.3}$$

Thus, it is a particular case of recurrence of form 4.2.3 with $k = 2$, $a_1 = a_2 = 1$, $c_1 = 2c$ and $c_2 = 3c$ where $c \in \mathbb{R}^+$. The complexity space is $\mathcal{RT}_{c,2}$. The functional defiend in this complexity space and associated to the equation 4.2.3 is given by:

$$\Xi_T(f)(n) = \begin{cases} c & \text{if } n = 1 \\ 3c & \text{if } n = 2 \\ f(n-1) + f(n-2) + 4c & \text{else} \end{cases} \tag{5.3.4}$$

whenever $f \in \mathcal{RT}_{c,2}$

The corollary 4.4.3 ensures the existence and the uniqueness of the solution of the recurrence equation 5.3.3 which is provided by the fixed of the functional Ξ_T.

Now, we will look for its asymptotic upper bound or the complexity class. By the corollary 4.4.3, this upper bound is the function g that satisfies $\Xi_T(g) \leq g$. To this end, we define the complexity function:

$$g_{a,r}(n) = \begin{cases} 2c & \text{if } n = 1 \\ 3c & \text{if } n = 2 \\ ra^2 & \text{else} \end{cases} \tag{5.3.5}$$

where $a, r \in \mathbb{R}^+$ are fixed. We have

$$\Xi_T(g_{a,r}) \leq g_{a,r} \Leftrightarrow -ra^3 + ra^2 + ra + 4c \leq 0$$

Let $\phi = \frac{1+\sqrt{5}}{2}$. By solving this equation, we find that

$$r \geq \frac{4}{\phi^3 - \phi^2 - \phi} \quad \text{and} \quad a > \phi$$

Recall that $f \in \mathcal{O}(g_{a,r})$ for every $a > \phi$ and $r \geq \frac{4}{\phi^3-\phi^2-\phi}$. So, if we take $a = 1.619$, the running time complexity of Fibonacci is in $\mathcal{O}(g_{1.619,r})$ where $r \geq \frac{4}{\phi^3-\phi^2-\phi}$.

6. Conclusion

Starting by the definition stated in metric space, we defined completeness for the QMS. Then, we define the complexity analysis of an algorithm by the mean of a running time function, and we discussed why particularly, we study its asymptotic behaviour. This latter can be bounded by a running time. Hence, we use the $\mathcal{O}$-notation.

Afterwards, we define the completness of the PQMS. We saw that there is a correspondence between QMS and PQMS. The weighted QM is the key. Then, we are able to provide the extended Banach fixed point theorem in the PQMS.

Besides, we associated a running time with a word in this Baire PQMS on the set of words over an alphabet. In the meantime, we define the functional associated to the recurrence equation of a running time we want to study in this set of words. As a result, the Baire PQMS on the set of words corresponds to the complexity space associated to the algorithm, which has as running time satisfying the recurrence equation. Since we proved that the functional is a contraction mapping on this space, the existence of a fixed point is guaranteed by the fixed point theorem. Additionally, we introduced a particular functional called improver. In fact, we can find a word such that this functional is an improver with respect to this word which is an upper bound of the fixed point.

Thereafter, in the complexity space for Divide and Conquer, the fixed point provided in the Baire PQMS is nothing else that the running time of the algorithm. Moreover, the improver provides the upper bound, thus the complexity class of the running time. We have also shown the applicability of these techniques to ensures the existence of the solution for a more general class of recursive algorithm.

Finally, we reproved the well-known complexity class of Mergesort in the best and average case behaviour and Quicksort in the best case behaviour which are examples of Divide and Conquer algorithm. Furthermore, using the same developed theory, we could find the complexity class of Largetwo in the average case and Quicksort in the worst case behaviour. To find the asymptotic complexity of Hanoi and Fibonacci, this theory still valid. These latter are examples of recursive algorithm.

Acknowledgements

This period of pandemic COVID-19 is a hard time for all of us. Without the support and encouragement of beautiful persons, this essay would not have been possible; my appreciation to all of them.

I want to acknowledge AIMS and its funders for supporting this work. I would also like to thank Prof Barry Green and Dr Simukai Utete for their help and their hard work to make this research project phase complete during this breakthrough.

I owe my warmest gratitude to my supervisors, Dr Collins Amburo Agyingi from the University of South Africa and Dr Yae Ulrich Gaba from IMSP, Bénin. Without their continuous encouragement and enthusiasm, this essay would hardly have been completed.

I gratefully acknowledge the contributions of my tutor Rahinatou. Her assistance, guidance, and effort are very appreciated. In addition, I am grateful to all the tutors, particularly Prudence. I also want to thank my colleague Abraham Orinda for his frequent help.

It is a pleasure to express my gratitude to my friend Ahmed El-Mwafy for his support towards the completion of this program and for making every moment at South Africa memorable.

Special thanks to my friends Hirmans, Patrick, Sitraka, and Antsa for the wonderful time we shared, specially movie night and karaoke. And thanks to all AIMS family!

Finally, my deep and sincere gratitude to my family for their support and prayers. I would like to thank particularly my sister who has always been there as a friend. To my mother, you have made me who I am and I cannot express enough thanks to you.

Glory to God!

References

Alghamdi, M. A., Shahzad, N., and Valero, O. New results on the baire partial quasi-metric space, fixed point theory and asymptotic complexity analysis for recursive programs. *Fixed Point Theory and Applications*, 2014(1):14, 2014.

Brattka, V. Generated quasi-metric hyper and function spaces. *Topology and its Applications*, 127(3): 355–373, 2003.

Cerdà-Uguet, M., Schellekens, M. P., and Valero, O. The baire partial quasi-metric space: a mathematical tool for asymptotic complexity analysis in computer science. *Theory of Computing Systems*, 50(2):387–399, 2012.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. *Introduction to algorithms*. MIT press, 2009.

Davey, B. A. and Priestley, H. A. *Introduction to lattices and order*. Cambridge University Press, 2002.

Doitchinov, D. On completeness in quasi-metric spaces. *Topology and its Applications*, 30(2):127–148, 1988.

Karapınar, E., Erhan, I., and Öztürk, A. Fixed point theorems on quasi-partial metric spaces. *Mathematical and Computer Modelling*, 57(9-10):2442–2448, 2013.

Künzi, H.-P. A., Pajoohesh, H., and Schellekens, M. P. Partial quasi-metrics. *Theoretical Computer Science*, 365(3):237–246, 2006.

Matthews, S. G. Partial metric topology. *Annals of the New York Academy of Sciences-Paper Edition*, 728:183–197, 1994.

Romaguera, S. and Schellekens, M. Quasi-metric properties of complexity spaces. *Topology and its Applications*, 98(1-3):311–322, 1999.

Romaguera, S., Tirado, P., and Valero, O. New results on the mathematical foundations of asymptotic complexity analysis of algorithms via complexity spaces. *International Journal of Computer Mathematics*, 89(13-14):1728–1741, 2012.

Schellekens, M. The Smyth completion: a common foundation for denotational semantics and complexity analysis. *Electronic Notes in Theoretical Computer Science*, (1):535–556, 1995.

Sutherland, W. A. *Introduction to metric and topological spaces*. Oxford University Press, 2009.

Willard, S. *General topology*. Courier Corporation, 2004.

Wilson, W. On quasi-metric spaces. *American Journal of Mathematics*, 53(3):675–684, 1931.