International Journal of Enhanced Research in Management & Computer Applications
ISSN: 2319-7471, Vol. 4, Issue 8, August-2015

Comprehensive Analysis of Fixed point theorems in partial metric space applications

Ashish

Research Scholar, Dept. of Mathematics, MDU, Rohtak, Haryana

ABSTRACT

In this paper, motivated by the utility of partial metric space applications, the author discussed whether they are a appropriate tool for asymptotic complexity analysis of algorithms. Solidly, the author demonstrates that the fixed point theorem does not constitute, on a basic level, a proper implement for the previously mentioned reason.

Keywords: partial metric, fixed point, complexity analysis, theorem.

I. INTRODUCTION

At the point when a computational program utilizes a recursion procedure to discover the solution to a problem, such a procedure is described by acquiring in each progression of the calculation an approximation to the aforementioned solution which is better than the approximations obtained in the preceding steps and, moreover, by getting dependably the last guess to the issue arrangement as the 'farthest point' of the registering procedure. A matehematical model to this kind of circumstances is the alleged Scott display which depends on thoughts from arrange theorem and topology for a definite record of the Scott model and its applications). Specifically, the request speaks to some idea of data such that each progression of the calculation is related to a component of the numerical model which is more prominent than (or equivalent to) alternate ones related with the former strides, since every guess gives more data about the last arrangement than those processed previously. The last yield of the computational procedure is viewed as the point of confinement of the progressive approximations. Accordingly the recursion forms are displayed as expanding arrangements of components of the scientific model, which is related to a requested set, that focalize to its minimum upper bound as for the given topology [1].

In 1994, Matthews presented the thought of Scott-like topology as a numerical structure to demonstrate expanding data content arrangements in software engineering in the soul of Scott [2].

Let us recall, with the point of helping the idea to remember Scott-like topology, that a couple $(X,\leq)$ is said to be a requested set gave that X is a nonempty set and $\leq$ is a reflexive, antisymmetric and transitive paired connection on X [2]. Given a subset $Y\subseteq X$, an upper headed for Y is a component $x\in X$ with the end goal that $y\leq x$ for all $y\in Y$. A slightest component for Y is a component $z\in Y$ with the end goal that $z\leq y$ for all $y\in Y$. Besides, a succession $(xn)n\in N$ in $(X,\leq)$ is expanding if $xn\leq xn+1$ for all $n\in N$.

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

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

(2) for every $O\in T$ containing the least upper bound of an increasing sequence $(xn)n\in N$, there exists $n0\in N$ such that $xn\in O$ for all $n>n0$.

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

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

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

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

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

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

Partial Metric Spaces

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

The concept of a partialmetric, since its presentation by Matthews, has been generally acknowledged in software engineering. This is because of the way that halfway metric spaces can be utilized as a numerical device to display computational procedures in the soul of Scott. To be sure, every incomplete metric p on X produces a T0 topology T(p) on X which has as a base the group of open p-balls $\{Bp(x,\varepsilon): x\in X,\varepsilon>0\}$, where $Bp(x,\varepsilon)= \{y\in X: p(x,y)<p(x,x)+\varepsilon\}$ for all $x\in X$ and $\varepsilon>0$. Besides, every halfway metric p incites a request $\leq p$ on X as takes after: $x\leq py \Leftrightarrow p(x,y)=p(x,x)$.

In remaining Section, it is likewise demonstrated that the reported fixed point hypotheses for self-mappings characterized on entire halfway metric spaces. Also, in a similar area, we give cases that demonstrate that the theory in the announcements of our outcomes can't be debilitated. The paper is committed to acquaint the peruser with fixed point systems for asymptotic unpredictability examination of calculations. Solidly, general and major parts of asymptotic many-sided quality investigation are reviewed and, also, the reference fixed guide procedure toward do the unpredictability examination of calculations, because of Schellekens and in which our examination is based, is displayed in detail keeping in mind the end goal to rouse our ensuing work. In the last segment, we talk about the practicality of utilizing the Matthews fixed point theorem as a numerical device for the asymptotic multifaceted nature investigation of calculations and, specifically, we demonstrate that the previously mentioned outcome isn't, on a basic level, proper for such a reason. Likewise, in a similar segment, we present another scientific procedure for the asymptotic many-sided quality examination of calculations whose premise is given by the fixed point comes about [4].

THE FIXED POINT THEOREMS

In this section the main results are presented which will play a central role in the application to asymptotic complexity analysis. To this end, let us recall that a mapping *f* from an ordered set $(X,\leq)$ into itself is monotone if $f(x)\leq f(y)$ whenever $x\leq y$. Moreover, according to a mapping from an ordered set $(X,\leq)$ into itself is said to be $\leq$-continuous provided that the least upper bound of $(f(xn))$ $n\in N$ is $f(x*)$ for every increasing sequence (xn) $n\in N$ whose least upper bound exists and is x. Of course every $\leq$-continuous mapping is monotone.

Theorem 1: *Let* (X,p) *be a complete partial metric space and let* $f:X\rightarrow X$ *be a* $\leq p$*-continuous mapping. If there exists* $x0\in X$ *such that* $x0\leq pf(x0)$*, then f has a fixed point in* $\uparrow x0=\{x\in X:x0\leq px\}$.

Proof : Let $x0\in X$ such that $x0\leq pf(x0)$. Since *f* is monotone, we have that

$$x0\leq pf(x0)\leq pf2(x0)\leq p\cdots\leq pfn(x0)\leq pfn+1(x0)\leq p\cdots. \quad (1)$$

Observe that we can assume, without loss of generality, that $x0\neq f(x0)$ since otherwise we have guaranteed the existence of the fixed point in $\uparrow x0$.

Since the sequence $(fn(x0))n\in N$ is increasing in $(X,\leq p)$, we have, by Proposition 1 and Remark 2, that there exists $x*\in X$ such that $x*$ is the least upper bound of $(fn(x0))n\in N$ and, in addition, that $\lim n\rightarrow\infty fn(x0)=x*$ with respect to T(ps). Since *f* is $\leq p$-continuous, we have that $f(x*)$ is the least upper bound of $(fn(x0))n\in N$. Whence we immediately obtain that $f(x*)=x*$ and that $x*\in\uparrow x0$□ .

Remark 1: Observe that the proof of Theorem 1 follows applying similar arguments to those given in the proof of Kleen's theorem or Tarski-Kantorovitch's theorem for mappings defined from ω-chain-complete ordered sets into itself. However, we have included a detailed proof of the aforementioned result for the sake of completeness and in order to help the reader.

The next example shows that the ≤p-continuity of the mapping cannot be deleted in the statement of Theorem 1.

Example 1: Let *p* be the partial metric space on (0,∞] given by

p∞(x,y)=max{1x,1y} (2)

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

Let us recall that, given a partial metric (X,p), a mapping f:X→X is continuous provided that *f* is continuous from (X,T(p)) into itself.

Remark 2 Note that every monotone and continuous mapping *f* from a complete partial metric space into itself is ≤p-continuous. Indeed, assume that ((xn))n∈N is an increasing sequence in (X,≤p). Since (X,p) is complete, we have guaranteed, by Proposition 1 and Remark 2, that there exists x∗∈X such that x∗ is the least upper bound of (xn)n∈N and, in addition, that limn→∞xn=x∗ with respect to T(ps). Since *f* is continuous, we have that limn→∞f(xn)=f(x∗) with respect to T(p). The monotonicity of *f* provides that f(xn)≤pf(x∗) and, thus, that p(f(x),f(xn))−p(f(xn),f(xn))=0. Whence we deduce that limn→∞f(xn)=f(x∗) with respect to T(ps). Moreover, since the sequence (f(xn))n∈N is increasing and the partial metric space (X,p) is complete, we have guaranteed the existence of the least upper bound of (f(xn))n∈N, say y∗∈X, in such a way that limn→∞f(xn)=y∗ with respect to T(ps). Therefore f(x∗)=y∗, as claimed.

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

Corollary 1: *Let* (X,p) *be a complete partial metric space and let* f:X→X *be a monotone and continuous mapping*. *If there exists* x0∈X *such that* x0≤pf(x0), *then f has a fixed point in* ↑x0={x∈X:x0≤px}.

In the following, given a partial metric space (X,p), we will say that a mapping f:X→X is ps-continuous provided that *f* is continuous from (X,T(ps)) into itself. In our subsequent result, the ps-continuity plays a central role.

ASYMPTOTIC COMPLEXITY ANALYSIS OF ALGORITHMS

The complexity analysis of an algorithm depends on deciding scientifically the amount of assets required by the calculation to take care of the issue for which it has been outlined. A run of the mill asset, assuming a focal part in unpredictability examination, is the execution time or running time of registering. Since there are frequently numerous calculations to take care of a similar issue, one goal of the intricacy investigation is to survey which of them is quicker when vast sources of info are considered. To this end, it is important to look at their running time of figuring. This is generally done by methods for the asymptotic investigation in which the running time of a calculation is signified by a capacity T:N→(0,∞] such that T(n) speaks to the time taken by the calculation to take care of the issue under thought when the contribution of the calculation is of size n. Give us a chance to indicate by RT the set framed by all capacities from ℕ into (0,∞].

Obviously the running time of a calculation does rely upon the information estimate n, as well as on the specific contribution of the size n (and the circulation of the information). Accordingly the running time of a calculation is distinctive when the calculation forms certain examples of information of a similar size n. As a result, with the end goal of size-based examinations, it is important to recognize three conceivable practices in the intricacy investigation of calculations. These are the supposed best case, the most pessimistic scenario and the normal case. The best case and the most pessimistic scenario for a contribution of size n are characterized by the base and the greatest running time of processing over all contributions of the size n, individually. The normal case for a contribution of size n is characterized by the normal esteem or normal over all contributions of size n [6].

By and large, to decide precisely the capacity which portrays the running time of processing of a calculation is a difficult assignment. Nonetheless, much of the time it is helpful to know the running time of registering of a calculation in an 'estimated' route as opposed to in a correct one. Thus, the asymptotic unpredictability examination of calculations is centered around getting the 'inexact' running time of figuring of a calculation, and this is finished by methods for the Θ-documentation. Give us a chance to review how the Θ-documentation permits accomplishing such an objective.

The complexity space approach

Schellekens, 1995 [5], presented a topological establishment for the asymptotic complexity investigation of calculations [6]. The previously mentioned establishment depends on the ideas of semi metric and complexity space.

Let us recall that, following [7], a quasi-metric on a nonempty set *X* is a function d:X×X→R+ such that for all x,y,z∈X:

(i) d(x,y)=d(y,x)=0⇔x=y;

(ii) d(x,y)≤d(x,z)+d(z,y).

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

Each quasi-metric *d* on *X* generates a T0-topology T(d) on *X* which has as a base the family of open *d*-balls {Bd(x,ε):x∈X,ε>0}, where Bd(x,ε)={y∈X:d(x,y)<ε} for all x∈X and ε>0.
Given a quasi-metric *d* on *X*, the function ds:X×X→R+ defined by ds(x,y)=max{d(x,y),d(y,x)} is a metric.
A quasi-metric space (X,d) is called bicomplete if the metric space (X,ds) is complete [4].

Let us recall that the complexity space is the pair (C,dC), where
C={f∈RT:∑n=1∞2−n1f(n)<∞}
and dC is the bicomplete quasi-metric on C defined by
dC(f,g)=∑n=1∞2−nmax{1g(n)−1f(n),0}.
Obviously, we adopt the convention that 1∞=0.

According to [5], from a complexity analysis point of view, it is possible to associate a function of C with each algorithm in such a way that such a function represents, as a function of the size of the input data, the running time of computing of the algorithm. Because of this, the elements of C are called complexity functions. Moreover, given two functions f,g∈C, the numerical value dC(f,g) (the complexity distance from *f* to *g*) can be interpreted as the relative progress made in lowering the complexity by replacing any program *P* with a complexity function *f* by any program *Q* with a complexity function *g*. Therefore, if f≠g, the condition dC(f,g)=0 can be read as *f* is 'at least as efficient' as *g* on all inputs. In fact, we have that dC(f,g)=0⇔f(n)≤∞g(n) for all n∈N and, thus, the fact that dC(f,g)=0 (dC(g,f)=0) implies that f∈O(g) (f∈Ω(g)).

The applicability of the complexity space to the asymptotic complexity analysis of algorithms was illustrated by Schellekens in [5].

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

A Divide and Conquer algorithm solves a problem of size n (n∈N) splitting it into a subproblems of size nb, for some constants a, b with a,b∈N and a,b>1, and solving them separately by the same algorithm. After obtaining the solution of the subproblems, the algorithm combines all subproblem solutions to give a global solution to the original problem. The recursive structure of a Divide and Conquer algorithm leads to a recurrence equation for the running time of computing. In many cases the running time of a Divide and Conquer algorithm is the solution to the Divide and Conquer recurrence equation of the form [7]

T(n)={caT(nb)+h(n)if n=1,if n∈Nb, (5)

where Nb={bk:k∈N}, c>0 denotes the complexity on the base case (i.e., the problem size is small enough and the solution takes constant time), and h(n) represents the time taken by the algorithm in order to divide the original problem into a subproblems and to combine all subproblems solutions into a unique one (h∈C and h(n)<∞∞ for all n∈N).

Notice that for Divide and Conquer algorithms with running time satisfying the recurrence equation (1), it is typically sufficient to obtain the complexity on inputs of size n, where n ranges over the set Nb.
Typical examples of algorithms whose running time of computing can be obtained by means of the recurrence (1) are Quicksort (best case behavior) and Mergesort (all behaviors).

FIXED POINT TECHNIQUE IN PARTIAL METRIC SPACES

Inspired by the impossibility of developing a fixed point technique for the asymptotic complexity analysis of algorithms, a new fixed point technique has been presented that respects the spirit of the original Schellekens [5] technique and whose foundation lies in the use of Theorems 1 and 2.

In the complexity analysis of algorithms, Divide and Conquer algorithms belong to the wider class of recursive algorithms. In many cases the running time of imputing of the latter is the solution to the following recurrence equation [9]:

T(n)={caT(n−1)+h(n)if n=1,if n≥2, (6)

where c>0, a≥1 and h∈C with h(n)<∞∞ for all n∈N.

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

T(n)={caT(nb)+h(n)if n=1,if n∈Nb. , (7)

Clearly, the preceding Divide and Conquer recurrence equation can be retrieved as a particular case of the recurrence equation (5). Indeed, the Divide and Conquer recurrence equation can be transformed into the following one [10]:

S(m)={caS(m−1)+r(m)if m=1,if m>1, (8)

where S(m)=T(bm−1) and r(m)=h(bm−1) for all m∈N.

The remainder of this section is devoted to introducing, by means of the partial metric space (C,pC), a new fixed point technique in the spirit of Schellekens for yielding the asymptotic complexity class of those recursive algorithms whose running time is the solution to the recurrence equation (8).

CONCLUSION

Inspired by the former fact, the author demonstrate two fixed point theorems which provide the mathematical basis for a new technique to carry out asymptotic complexity analysis of algorithms by means of partial metrics. Moreover, in order to illustrate and to validate the developed theory, the author applied the results to analyze the asymptotic complexity of two celebrated recursive algorithms.

REFERENCES

[1]. Alghamdi MA, Shahzad N, Valero O: On fixed point theory in partial metric spaces. *Fixed Point Theory Appl.* 2012., 2012: Article ID 175. doi:10.1186/1687–1812–2012–175
[2]. Bukatin MA, Scott JS: Towards computing distances between programs via Scott domains. LNCS 1234. *Logical Foundations of Computer Science* 1997, 33–43.
[3]. Agarwal RP, Alghamdi MA, Shahzad N: Fixed point theory for cyclic generalized contractions in partial metric spaces. *Fixed Point Theory Appl.* 2012., 2012: Article ID 40. doi:10.1186/1687–1812–2012–40
[4]. Gierz G, Hofmann KH, Keimel K, Lawson JD, Mislove M, Scott DS: *Continuous Lattices and Domains*. Cambridge University Press, Cambridge; 2003
[5]. Schellekens MP: The correspondence between partial metrics and semivaluations. *Theor. Comput. Sci.* 2004, 315: 135–149. 10.1016/j.tcs.2003.11.016.
[6]. Di Bari C, Vetro P: Fixed points for weak φ -contractions on partial metric spaces. *Int. J. Eng. Contemp. Math. Sci.* 2011, 1: 5–13.
[7]. Karapinar E: Fixed point theory for cyclic weak ϕ -contraction. *Appl. Math. Lett.* 2011, 24: 822–825. 10.1016/j.aml.2010.12.016.
[8]. Davey BA, Priestley HA: *Introduction to Lattices and Order*. Cambridge University Press, Cambridge; 1990.

[9]. Matthews SG: An extensional treatment of lazy data flow deadlock. *Theor. Comput. Sci.* 1995, 151: 195–205. 10.1016/0304-3975(95)00051-W

[10]. Romaguera S, Valero O: Domain theoretic characterizations of quasi-metric completeness in terms of formal balls. *Math. Struct. Comput. Sci.* 2010, 20: 453–472. 10.1017/S0960129510000010