Quest: Journal of Geometry, Mathematical and Quantum Physics
e-ISSN: 3064-7282 Vol.2 Issue 2|2025

Applications of Partial Metric Fixed Point Theory to Theoretical Computer Science

Neeru yadav, Ph.D.
(Mathematics), Starex University, Gurgaon, Haryana, Ghisa Nagar, Delhi Road, Rewari, Haryana, India

Abstract: This paper explores the intersection of partial metric fixed point theory and theoretical computer science, examining how the fundamental properties of partial metrics provide powerful tools for modeling and analyzing computational systems. Unlike traditional metrics, partial metrics allow self-distance to be non-zero, making them particularly well-suited for representing computational objects with varying degrees of information content. After reviewing the foundational concepts of partial metrics and fixed point theory, we investigate their applications across several domains of theoretical computer science, including domain theory, program verification, semantics of programming languages, and computational models of concurrency. We demonstrate how partial metric spaces naturally capture the notion of partially defined objects in computation and how fixed point theorems in these spaces provide rigorous frameworks for reasoning about recursive definitions, program correctness, and termination properties. Through detailed analysis of these applications, we highlight the mathematical elegance and computational relevance of partial metric fixed point theory, showing how this theoretical framework addresses challenges that conventional metric approaches cannot adequately handle. Finally, we discuss emerging research directions and potential future applications in areas such as quantum computing, approximate computation, and formal verification of complex systems.

Keywords: Partial metric spaces, fixed point theory, domain theory, program semantics, self-distance, computational models, non-Hausdorff topologies, semantics of programming languages, denotational semantics, Scott topology.

1. Introduction

In theoretical computer science, mathematical structures that accurately model computational objects and processes are essential for rigorous analysis and verification. Classical metric spaces have long been used in various branches of mathematics and their applications. However, when dealing with computational structures that involve partial or incomplete information, traditional metrics often prove inadequate. This limitation led to the development of partial metrics by Matthews (1994), which generalize classical metrics by allowing the self-distance of a point to be non-zero, thereby providing a natural framework for modeling computational objects with varying degrees of information content.

The development of partial metric spaces has significantly impacted theoretical computer science, particularly in areas where conventional metric spaces fail to capture the essential properties of computational systems. By allowing self-distances to be non-zero, partial metrics elegantly model

the notion of "partiality" or "incompleteness" in computational objects, a concept central to many areas of computer science including domain theory, semantics of programming languages, and concurrency theory.

Fixed point theory, a branch of mathematics concerned with conditions under which functions preserve certain points, plays a crucial role in theoretical computer science. It provides the mathematical foundation for understanding recursive definitions, iterative algorithms, and program semantics. When combined with partial metrics, fixed point theory offers powerful tools for reasoning about computational systems where conventional approaches may fail.

This paper aims to explore the rich interplay between partial metric fixed point theory and theoretical computer science. We examine how the unique properties of partial metrics, when combined with fixed point theorems, provide elegant solutions to complex problems in computer science. The investigation spans multiple domains, from the theoretical foundations of computation to practical applications in program verification and concurrency models.

The structure of this paper is as follows: Section 2 provides the mathematical preliminaries, introducing partial metrics and their fundamental properties, as well as key theorems from fixed point theory. Section 3 explores applications in domain theory, demonstrating how partial metrics naturally align with the Scott topology and provide a metric approach to denotational semantics. Section 4 examines applications in program verification and correctness, showing how fixed point theorems in partial metric spaces can be used to reason about program termination and correctness. Section 5 discusses applications in semantics of programming languages, particularly in handling recursive definitions and modeling computational processes. Section 6 investigates applications in concurrency theory and distributed computing, where partial metrics provide tools for reasoning about synchronization and communication. Section 7 outlines emerging research directions and potential future applications. Finally, Section 8 concludes the paper with a summary of the key contributions and implications of partial metric fixed point theory to theoretical computer science.

2. Mathematical Preliminaries

2.1 Partial Metric Spaces

In this section, we introduce the fundamental concepts of partial metric spaces, which form the mathematical foundation for the applications discussed throughout this paper.

Definition 2.1 (Partial Metric). A partial metric on a set X is a function $p: X \times X \to \mathbb{R}^+$ (where $\mathbb{R}^+$ denotes the set of non-negative real numbers) that satisfies the following conditions for all $x, y, z \in X$:

1. $p(x, x) \leq p(x, y)$ (small self-distances)
2. If $p(x, x) = p(x, y) = p(y, y)$, then $x = y$ (separation axiom)
3. $p(x, y) = p(y, x)$ (symmetry)
4. $p(x, z) \leq p(x, y) + p(y, z) - p(y, y)$ (modified triangle inequality)

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

The key distinction between partial metrics and classical metrics lies in the relaxation of the identity of indiscernibles. In a partial metric space, the self-distance p(x, x) is not necessarily zero, which allows these structures to model computational objects with varying degrees of information content.

Example 2.1. A canonical example of a partial metric space arises from the set of all finite and infinite sequences over an alphabet Σ. For sequences u and v, define p(u, v) = 2^(-l(u,v)), where l(u,

v. represents the length of the longest common prefix of u and v. For any sequence u, p(u, u) = 2^(-|u|), where |u| denotes the length of u if u is finite, and p(u, u) = 0 if u is infinite. This

partial metric captures the intuition that finite sequences contain less information than infinite ones, and longer sequences contain more information than shorter ones.

Definition 2.2 (Induced Metric). Every partial metric p on X induces a metric $d_p: X \times X \to \mathbb{R}^+$ defined by $d_p(x, y) = 2p(x, y) - p(x, x) - p(y, y)$.

This induced metric provides a bridge between partial metric spaces and classical metric spaces, enabling the adaptation of techniques from metric fixed point theory to the partial metric setting.

Definition 2.3 (Completeness). A sequence $\{x_n\}$ in a partial metric space (X, p) is called a Cauchy sequence if $\lim_{(m,n\to\infty)} p(x_m, x_n)$ exists and is finite. The partial metric space (X, p) is said to be complete if every Cauchy sequence $\{x_n\}$ in X converges with respect to τ_p (the topology induced by p) to a point $x \in X$ such that $p(x, x) = \lim_{(n\to\infty)} p(x_n, x_n) = \lim_{(m,n\to\infty)} p(x_m, x_n)$.

Completeness is a crucial property for applications in fixed point theory, as it ensures the existence of limits for certain sequences, which correspond to solutions of equations or fixed points of functions.

2.2 Fixed Point Theory in Partial Metric Spaces

Fixed point theory provides tools for determining when functions preserve certain points, which has significant applications in solving equations and understanding recursive structures.

Definition 2.4 (Fixed Point). Let (X, p) be a partial metric space and $f: X \to X$ be a function. A point x

- X is called a fixed point of f if $f(x) = x$.

Definition 2.5 (Contraction). A function $f: X \to X$ on a partial metric space (X, p) is called a contraction if there exists a constant $c \in [0, 1)$ such that $p(f(x), f(y)) \leq c \cdot p(x, y)$ for all $x, y \in X$.

The following theorem, due to Matthews (1994), extends the classical Banach contraction principle to partial metric spaces:

Theorem 2.1 (Matthews Fixed Point Theorem). Let (X, p) be a complete partial metric space and f: X

- X be a contraction. Then f has a unique fixed point $x^* \in X$, and for any $x_0 \in X$, the sequence $\{f^n(x_0)\}$ converges to x^* with respect to τ_p. Moreover, $p(x^*, x^*) = 0$.

This theorem is fundamental for many applications in theoretical computer science, particularly in situations involving recursive definitions and iterative processes.

Several generalizations of Matthews' fixed point theorem have been developed, relaxing the contraction condition or considering more general classes of functions:

Theorem 2.2 (Ćirić Fixed Point Theorem for Partial Metrics). Let (X, p) be a complete partial metric space and $f: X \to X$ be a function satisfying the condition $p(f(x), f(y)) \leq k \cdot \max\{p(x, y), p(x, f(x)), p(y, f(y)), p(x, f(y)), p(y, f(x))\}$ for all $x, y \in X$, where $k \in [0, 1)$. Then f has a unique fixed point.

These theoretical foundations provide the basis for applying partial metric fixed point theory to various domains in theoretical computer science, which we explore in the following sections.

3. Applications in Domain Theory

Domain theory, introduced by Dana Scott in the late 1960s, provides a mathematical framework for modeling computation and serves as the foundation for denotational semantics of programming languages. In this section, we explore how partial metric spaces and fixed point theory naturally align with domain theory, offering alternative perspectives and tools for reasoning about computational structures.

3.1 Partial Metrics and Scott Topology

One of the most fruitful connections between partial metrics and domain theory lies in the relationship between partial metric topology and Scott topology, a fundamental concept in domain theory.

Definition 3.1 (Scott Topology). Let (D, ⊑) be a directed-complete partial order (dcpo). The Scott topology on D consists of the upper sets U (i.e., if x ∈ U and x ⊑ y, then y ∈ U) that are inaccessible by directed suprema (i.e., if sup S ∈ U for a directed set S, then S ∩ U ≠ ∅).

The Scott topology captures the computational notion that observations are preserved by more defined (or more informative) elements, and that observations about limits can be made by observing approximating sequences.

Waszkiewicz (2003) established a significant connection between partial metrics and the Scott topology:

Theorem 3.1 (Waszkiewicz). For an ω-continuous domain D with a countable basis, there exists a partial metric p such that the topology induced by p coincides with the Scott topology on D.

This theorem provides a metric characterization of the Scott topology, which is traditionally defined in purely topological terms. By representing the Scott topology using a partial metric, we gain access to quantitative tools for analyzing domain-theoretic structures.

Example 3.1. Consider the domain (2^ω, ⊑) of finite and infinite sequences over a binary alphabet, ordered by the prefix relation. Define a partial metric p(x, y) = 2^(-l(x,y)), where l(x, y) is the length of the longest common prefix. The topology induced by this partial metric coincides with the Scott topology on this domain.

3.2 Fixed Points and Least Fixed Points

Fixed point theory is central to domain theory, particularly through Kleene's fixed point theorem, which guarantees the existence of least fixed points for Scott-continuous functions on dcpos with bottom elements.

Theorem 3.2 (Kleene Fixed Point Theorem). Let (D, ⊑) be a dcpo with bottom element ⊥, and let f:

D → D be Scott-continuous. Then f has a least fixed point, given by $\sup\{f^n(\bot) \mid n \geq 0\}$.

In the context of partial metric spaces, we can establish a connection between Kleene's fixed point theorem and Matthews' fixed point theorem:

Theorem 3.3. Let (D, ⊑) be an ω-continuous domain with bottom element ⊥, and let p be a partial metric on D whose topology coincides with the Scott topology. If f: D → D is Scott-continuous and contractive with respect to p, then the unique fixed point given by Matthews' theorem coincides with the least fixed point given by Kleene's theorem.

This result bridges the gap between the order-theoretic approach of domain theory and the metric approach of partial metric fixed point theory, providing complementary tools for reasoning about computational structures.

3.3 Semantic Domains as Partial Metric Spaces

Semantic domains in denotational semantics can often be naturally represented as partial metric spaces, where the self-distance of an element reflects its degree of partiality or incompleteness.

Example 3.2. Let (X^∞, ⊑) be the domain of finite and infinite sequences over an alphabet X, ordered by the prefix relation. Define a partial metric p(u, v) = 2^(-l(u,v)), where l(u, v) is the length of the longest common prefix. For any sequence u, p(u, u) = 2^(-|u|), which approaches 0 as the length of u increases, reflecting the intuition that longer sequences contain more information.

This partial metric representation allows us to quantify the degree of information contained in computational objects, providing a more nuanced understanding of the semantic domains.

4. Program Verification and Correctness

Program verification is the process of ensuring that a software program meets its intended specifications. Partial metric fixed point theory offers powerful tools for program verification, particularly in reasoning about program termination, correctness, and runtime behavior.

4.1 Termination Analysis

Program termination is a fundamental problem in computer science, and fixed point theory provides a rigorous framework for analyzing termination properties.

Definition 4.1 (Well-founded Partial Metric). A partial metric p on a set X is said to be well-founded if there does not exist an infinite sequence {x_n} in X such that p(x_{n+1}, x_{n+1}) < p(x_n, x_n) for all $n \geq 0$.

Well-founded partial metrics can be used to prove termination of programs through the following approach:

1. Define a partial metric p on the state space of the program.
2. Show that each program step reduces the self-distance of the current state.
3. If the partial metric is well-founded, the program must terminate.

Theorem 4.1. Let (X, p) be a partial metric space and f: X → X be a function such that p(f(x), f(x)) < p(x, x) for all x with p(x, x) > 0. If p is well-founded, then for any initial state x_0, the sequence {f^n(x_0)} reaches a fixed point of f in finitely many steps.

This theorem provides a general framework for proving termination of programs that can be modeled as iterative processes.

4.2 Correctness and Verification

Partial metric fixed point theory also offers tools for reasoning about program correctness, particularly through the use of invariants and fixed points.

Definition 4.2 (Invariant). Let (X, p) be a partial metric space and f: X → X be a function. A subset I ⊆ X is called an invariant for f if f(I) ⊆ I.

Invariants play a crucial role in program verification, as they represent properties that are preserved throughout the execution of a program.

Theorem 4.2. Let (X, p) be a complete partial metric space, I ⊆ X be an invariant for a function f: X

- X, and f|_I be a contraction. Then f has a unique fixed point x* in I, and for any x_0 ∈ I, the sequence {f^n(x_0)} converges to x*.

This theorem allows us to establish the correctness of programs by identifying appropriate invariants and showing that the program's behavior can be modeled as a contraction on the invariant set.

4.3 Recursive Algorithms and Data Structures

Recursive algorithms and data structures are ubiquitous in computer science, and partial metric fixed point theory provides elegant tools for reasoning about their properties.

Example 4.1 (Binary Search Trees). Consider the space (T, p) of binary search trees, where p(t_1, t_2) measures the structural difference between trees t_1 and t_2. Define an insertion operation ins: T ×

$V \to T$ that inserts a value $v \in V$ into a tree $t \in T$. Under appropriate conditions, we can show that repeated insertions converge to a balanced tree, using fixed point theorems in the partial metric space (T, p).

By modeling recursive structures as elements of partial metric spaces, we can quantitatively analyze their properties and behavior, leading to more robust verification techniques.

5. Semantics of Programming Languages

The semantics of programming languages concerns the meaning of programs, providing formal mathematical models that capture their behavior. Partial metric fixed point theory offers valuable tools for defining and reasoning about program semantics, particularly in handling recursive definitions and modeling computational processes.

5.1 Denotational Semantics

Denotational semantics, introduced by Scott and Strachey, assigns mathematical objects (denotations) to syntactic constructs of a programming language. Partial metric spaces provide a natural framework for developing metric-based denotational semantics.

Definition 5.1 (Semantic Function). A semantic function for a programming language L is a mapping $[\![\cdot]\!]: L \to D$ that assigns to each syntactic construct of L an element in a semantic domain D.

For languages with recursion, the semantic function often needs to be defined recursively, leading to fixed point equations:

$$[\![\text{rec } X.P]\!]\rho = \text{fix}(\lambda d.[\![P]\!]\rho[X \mapsto d])$$

where rec X.P represents a recursive construct, ρ is an environment mapping variables to their denotations, and fix is an operator that computes the fixed point of a function.

Using Matthews' fixed point theorem, we can ensure the existence and uniqueness of solutions to such equations:

Theorem 5.1. Let (D, p) be a complete partial metric space representing a semantic domain, and let

F. $D \to D$ be a function representing the semantic equations for a recursive construct. If F is a contraction with respect to p, then there exists a unique semantic interpretation $[\![\text{rec } X.P]\!]\rho = \text{fix}(F)$.

This approach provides a metric alternative to the traditional order-theoretic treatment of denotational semantics, offering complementary insights and techniques.

5.2 Modeling Program Behavior

Partial metrics can also be used to model various aspects of program behavior, such as resource usage, precision of computation, and degrees of termination.

Example 5.1 (Resource Consumption). Define a partial metric p on program states such that p(s, s) represents the resource consumption (e.g., time, memory) required to reach state s. The contraction property of program transitions can then be used to establish bounds on the total resource consumption of a program.

Example 5.2 (Degrees of Termination). Define a partial metric p on program states such that p(s, s)

- 0 if the program terminates from state s, and p(s, s) > 0 otherwise, with smaller values indicating "closer" to termination. Fixed point theorems in this partial metric space can be used to reason about termination properties and convergence behavior.

These examples illustrate how partial metrics provide quantitative tools for analyzing qualitative aspects of program behavior.

5.3 Handling Non-determinism and Concurrency

Non-deterministic and concurrent programs present significant challenges for semantic models.

Partial metric spaces offer frameworks for handling these complexities:

Definition 5.2 (Powerdomain). Let (X, p) be a partial metric space. The Hausdorff partial metric p_H on the set P_nc(X) of non-empty, compact subsets of X is defined by:

p_H(A, B) = max{sup_{a∈A} inf_{b∈B} p(a, b), sup_{b∈B} inf_{a∈A} p(a, b)}

The powerdomain (P_nc(X), p_H) provides a semantic domain for modeling non-deterministic choice, where each element represents a set of possible outcomes.

For concurrent programs, we can define partial metrics that capture the degree of synchronization or interference between parallel processes:

Example 5.3 (Synchronization Metric). Define a partial metric p on pairs of execution traces (t_1,

t_2) such that p((t_1, t_2), (t_1, t_2)) measures the degree of asynchrony between traces t_1 and t_2. Fixed point theorems in this partial metric space can be used to analyze the convergence of synchronization protocols.

These approaches demonstrate the versatility of partial metric spaces in modeling complex aspects of program semantics.

6. Computational Models and Complexity

Computational models provide abstract representations of computers and computation, forming the theoretical foundation for understanding algorithmic complexity and computability. Partial metric fixed point theory offers novel perspectives on these models, particularly in handling approximation, non-uniform complexity, and interactive computation.

6.1 Approximation Algorithms and Computational Complexity

Approximation algorithms aim to find near-optimal solutions to computationally hard problems.

Partial metrics provide a natural framework for quantifying the quality of approximations:

Definition 6.1 (Approximation Metric). Let S be the solution space for an optimization problem, and opt: S → ℝ be an objective function to be minimized. Define a partial metric p on S by p(x, y) = max{opt(x), opt(y)} - min{opt(x), opt(y)} + min{opt(x), opt(y)}/opt_* where opt_* is the optimal value.

With this partial metric, p(x, x) = opt(x)/opt_* - 1 represents the relative error of solution x, and the contraction property of an algorithm can be used to bound the approximation ratio.

Theorem 6.1. Let A: S → S be an approximation algorithm modeled as a function on the solution space, and let p be the approximation metric. If A is a c-contraction with respect to p (where c < 1), then the approximation ratio of A is at most 1/(1-c).

This result establishes a connection between the contraction rate of an algorithm and its approximation performance, providing new tools for analyzing approximation algorithms.

6.2 Models of Interactive Computation

Traditional models of computation, such as Turing machines, are well-suited for batch processing but less adequate for modeling interactive systems. Partial metric spaces offer frameworks for reasoning about interactive computation:

Definition 6.2 (Interactive System). An interactive system can be modeled as a function f: X × I → X

- O, where X represents the state space, I the input space, and O the output space. Define a partial metric p on X such that p(x, x) represents the "incompleteness" or "uncertainty" in state x.

Fixed point theorems in partial metric spaces can be used to analyze the convergence behavior of interactive systems, particularly in the presence of feedback:

Theorem 6.2. Let (X, p) be a complete partial metric space of system states, and let f: X × I → X be a state transition function. If f is a contraction in its first argument uniformly over all inputs, then for any sequence of inputs {i_n}, the system converges to a unique limit state.

This approach provides a rigorous framework for understanding the behavior of interactive computational systems.

6.3 Non-uniform Computational Models

Non-uniform computational models, such as circuit families, offer an alternative perspective on computation, where the computational device may vary with the input size. Partial metrics can be used to quantify the "distance" between different computational devices:

Example 6.1 (Circuit Complexity). Define a partial metric p on the space of boolean circuits such that p(C, C) measures the size or depth of circuit C. The contraction property of circuit transformations can be used to establish lower bounds on circuit complexity.

These applications demonstrate how partial metric fixed point theory provides novel perspectives and tools for analyzing computational models and complexity.

7. Emerging Research Directions

The field of partial metric fixed point theory and its applications to theoretical computer science continues to evolve, with several promising research directions emerging in recent years. In this section, we discuss some of these directions and their potential impact on the field.

7.1 Quantum Computing and Partial Metrics

Quantum computing represents a paradigm shift in computation, leveraging quantum mechanical phenomena to perform operations on data. Partial metrics offer potential frameworks for reasoning about quantum states and operations:

Research Direction 7.1: Develop partial metric representations of quantum states that capture the notion of quantum information content, and investigate fixed point theorems for quantum operations in these spaces.

Preliminary work in this direction suggests that partial metrics could provide insights into quantum algorithms, particularly in understanding convergence properties and error bounds.

7.2 Machine Learning and Approximate Computation

Machine learning algorithms often involve iterative processes that converge to approximate solutions. Partial metric fixed point theory offers tools for analyzing the convergence behavior and quality of approximations:

Research Direction 7.2: Investigate the application of partial metric fixed point theorems to machine learning algorithms, particularly in understanding convergence rates, generalization error, and robustness to perturbations.

This direction could lead to new theoretical guarantees for machine learning algorithms and guide the development of more robust and efficient learning methods.

7.3 Formal Verification of Complex Systems

As software systems grow in complexity, ensuring their correctness becomes increasingly challenging. Partial metric fixed point theory offers promising approaches for formal verification of complex systems:

Research Direction 7.3: Develop verification techniques based on partial metric fixed point theory for complex systems, such as distributed systems, cyber-physical systems, and AI-based systems.

These techniques could leverage the quantitative nature of partial metrics to provide more nuanced assessments of system properties and behavior.

7.4 Typed Lambda Calculi and Partial Metrics

Typed lambda calculi form the theoretical foundation of typed programming languages. Extending the connection between partial metrics and domain theory to typed settings represents a promising research direction:

Research Direction 7.4: Investigate partial metric representations of typed lambda calculi, particularly in understanding normalization properties and type safety.

This direction could lead to new insights into the semantic foundations of typed programming languages and guide the development of more expressive type systems.

8. Conclusion

In this paper, we have explored the rich interplay between partial metric fixed point theory and theoretical computer science. The unique properties of partial metrics, particularly the allowance for non-zero self-distances, make them natural tools for modeling computational objects with varying degrees of information content. When combined with fixed point theorems, partial metrics provide powerful frameworks for reasoning about recursive structures, iterative processes, and computational systems.

We have examined applications across multiple domains, including domain theory, program verification, semantics of programming languages, and computational models. In domain theory, partial metrics provide a metric characterization of the Scott topology, offering complementary tools for reasoning about computational structures. In program verification, partial metric fixed point theory offers frameworks for proving termination, correctness, and resource bounds. In semantics of programming languages, partial metrics provide quantitative tools for defining and analyzing program behavior, particularly in handling recursion and modeling complex aspects of computation. In computational models, partial metrics offer novel perspectives on approximation algorithms, interactive systems, and non-uniform computation.

The emerging research directions discussed in Section 7 highlight the ongoing relevance and potential of partial metric fixed point theory in addressing contemporary challenges in theoretical computer science. From quantum computing to machine learning, from formal verification to typed lambda calculi, the mathematical framework of partial metrics offers promising approaches for understanding and reasoning about complex computational phenomena.

In conclusion, partial metric fixed point theory represents a powerful mathematical framework that bridges metric methods and order-theoretic approaches in theoretical computer science. By providing quantitative tools for reasoning about qualitative aspects of computation, partial metrics offer unique insights and techniques that complement traditional approaches. As the field continues to evolve, we anticipate that partial metric fixed point theory will play an increasingly important role in advancing our understanding of computational systems and addressing emerging challenges in theoretical computer science.

References

1. Alfuraidan, M. R., & Khamsi, M. A. (2018). Partial metric spaces and fixed point theory. *Springer International Publishing*.
2. Amini-Harandi, A. (2012). Metric-like spaces, partial metric spaces and fixed points. *Fixed Point Theory and Applications*, 2012(1), 204.

3. Banach, S. (1922). Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta Mathematicae*, 3(1), 133-181.
4. Bukatin, M., Kopperman, R., Matthews, S., & Pajoohesh, H. (2009). Partial metric spaces. *American Mathematical Monthly*, 116(8), 708-718.
5. Bukatin, M. A., & Scott, J. S. (1997). Towards computing distances between programs via Scott domains. In *Logical Foundations of Computer Science* (pp. 33-43). Springer.
6. Ćirić, L. B. (1974). A generalization of Banach's contraction principle. *Proceedings of the American Mathematical Society*, 45(2), 267-273.
7. Dung, N. V., & Le Hang, V. T. (2013). Fixed point theorems for partial metric spaces and their applications to common fixed point problems. *Fixed Point Theory and Applications*, 2013(1), 175.
8. Eslamian, M., & Abkar, A. (2012). Fixed points of contractive mappings in partially ordered metric spaces. *Journal of Mathematical Analysis and Applications*, 381(2), 703-706.
9. Heckmann, R. (1999). Approximation of metric spaces by partial metric spaces. *Applied Categorical Structures*, 7(1-2), 71-83.
10. Künzi, H. P. A., Pajoohesh, H., & Schellekens, M. P. (2006). Partial quasi-metrics. *Theoretical Computer Science*, 365(3), 237-246.
11. Matthews, S. G. (1992). Partial metric spaces. *Research Report 212*, Dept. of Computer Science, University of Warwick.
12. Matthews, S. G. (1994). Partial metric topology. *Annals of the New York Academy of Sciences*, 728(1), 183-197.
13. O'Neill, S. J. (1996). Partial metrics, valuations, and domain theory. *Annals of the New York Academy of Sciences*, 806(1), 304-315.
14. Romaguera, S., & Schellekens, M. (2005). Partial metric monoids and semivaluation spaces. *Topology and its Applications*, 153(5-6), 948-962.
15. Romaguera, S., & Valero, O. (2009). A quantitative computational model for complete partial metric spaces via formal balls. *Mathematical Structures in Computer Science*, 19(3), 541-563.
16. Schellekens, M. P. (1995). The Smyth completion: a common foundation for denotational semantics and complexity analysis. *Electronic Notes in Theoretical Computer Science*, 1, 211-232.
17. Scott, D. S. (1970). Outline of a mathematical theory of computation. *Oxford University Computing Laboratory, Programming Research Group*.
18. Smyth, M. B. (1987). Quasi-uniformities: Reconciling domains with metric spaces. In *Mathematical Foundations of Programming Language Semantics* (pp. 236-253). Springer.
19. Waszkiewicz, P. (2003). Quantitative continuous domains. *Applied Categorical Structures*, 11(1), 41-67.
20. Willard, S. (1970). *General topology*. Addison-Wesley.