View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by UPCommons. Portal del coneixement obert de la UPC

Mathware & Soft Computing 15 (2008) 251-261

A Connection Between Computer Science and Fuzzy Theory: Midpoints and Running Time of Computing

J. Casasnovas and O. Valero*
Dept. of Math. and Computer Science,
University of the Balearic Islands,
07122 Palma de Mallorca (Spain)
jaume.casasnovas@uib.es, *o.valero@uib.es*

Abstract

Following the mathematical formalism introduced by M. Schellekens [Electronic Notes in Theoret. Comput. Sci. 1 (1995), 211-232] in order to give a common foundation for Denotational Semantics and Complexity Analysis, we obtain an application of the theory of midpoints for asymmetric distances defined between fuzzy sets to the complexity analysis of algorithms and programs. In particular we show that the average running time for the algorithm known as Largetwo is exactly a midpoint between the best and the worst case running time of computing.

Keywords: Fuzzy set, asymmetric distance, midpoint, complexity analysis, running time of computing.

1 Introduction and preliminaries

Throughout this paper we shall use the letters $\mathbb{R}$, $\mathbb{R}^+$, ω and $\mathbb{N}$ to denote the set of real numbers, the set of nonnegative real numbers, the set of nonnegative integer numbers and the set of positive integers numbers, respectively.

In order to fix the terminology let us recall a few concepts about asymmetric distances.

In our context by an *asymmetric distance* (quasi-metric in [14]) on a (nonempty) set X we mean a nonnegative real-valued function d on $X \times X$ such that for all $x, y, z \in X$:

(i) $d(x,y) = d(y,x) = 0$ if and only if $x = y$.
(ii) $d(x,z) \leq d(x,y) + d(y,z)$.

*The second author thanks the support of the Spanish Ministry of Education and Science, and FEDER, grant MTM2006-14925-C02-01

Note that a *distance* (metric) on a set X is an asymmetric distance d on X satisfying in addition the following two conditions for all $x, y \in X$:

(i') $d(x, y) = 0$ if and only if $x = y$.

(iii) $d(x, y) = d(y, x)$.

Our main references for asymmetric distances are [5] and [14].

An *asymmetric distance space* (quasi-metric space in [14]) is a pair (X, d) such that X is a (nonempty) set and d is an asymmetric distance on X.

If d is an asymmetric distance on X, then the nonnegative real valued function d^{-1} defined on $X \times X$ by $d^{-1}(x, y) = d(y, x)$ is again an asymmetric distance, called the conjugate of d. Thus the conjugate of an asymmetric distance space (X, d) is the pair (X, d^{-1}). Note that each asymmetric distance d induces, in a natural way, a distance (metric) D on X as follows: $D = d + d^{-1}$, i.e.

$$D(x, y) = d(x, y) + d(y, x)$$

for all $x, y \in X$.

A well-known and useful example of asymmetric distance space, which will play a crucial role in our work, it is the so-called upper quasi-metric space, which consists of the pair $(\mathbb{R}, u)$, where $u(x, y) = (y - x) \vee 0$ for all $x, y \in \mathbb{R}$ (see [5] or [14]). Then $u^{-1}(x, y) = (x - y) \vee 0$ is the conjugate of the upper quasi-metric on $\mathbb{R}$. Furthermore, the distance induced by u is exactly the Euclidean metric $|\cdot|$ on $\mathbb{R}$, i.e.

$$|y - x| = u(x, y) + u^{-1}(x, y)$$

for all $x, y \in \mathbb{R}$. In the sequel the restriction of the asymmetric distance u to $[0, 1]$ will be also denoted by u.

Let us fix a finite set

$$X = \{x_1, ..., x_n\},$$

and let $\mathcal{FP}(X)$ denote the set of its $[0, 1]$-valued fuzzy subsets. To simplify the notations, given a fuzzy subset of X, we shall write μ_i instead of $\mu(x_i)$.

In [13] B. Kosko introduced a fuzzy subset $\mu \in \mathcal{FP}(X)$ as a vector $(\mu_1, ..., \mu_n) \in [0, 1]^n$. This is possible because the mapping sending every $\mu \in \mathcal{FP}(X)$ to the vector $(\mu_1, ..., \mu_n) \in [0, 1]^n$ is a bijection and, thus, we can identify, in a one-to-one way, every fuzzy subset of X with a point of Kosko's n-dimensional hypercube $[0, 1]^n$. For a deeper treatment of the hypercubical calculus we refer the reader to [28]. Kosko's hypercube has a wide number of applications to engineering, artificial intelligence, medicine and biology. Some of these applications are based on the notions of segment joining two given fuzzy subsets and set of midpoints between two fuzzy subsets. Such ideas were introduced by J.J. Nieto and A. Torres in [17]. They studied the properties and the relations between the mentioned concepts and gave several applications of the obtained results to study real medical data. In particular given a distance d on $\mathcal{FP}(X)$ and $\mu, \nu \in \mathcal{FP}(X)$ the segment and the midset between the fuzzy subsets μ and ν are exactly the sets

$$\begin{aligned} \mathrm{seg}_d(\mu, \nu) &= \{\zeta \in \mathcal{FP}(X) : d(\mu, \nu) = d(\mu, \zeta) + d(\zeta, \nu)\} \\ \mathrm{mid}_d(\mu, \nu) &= \{\zeta \in \mathcal{FP}(X) : \frac{d(\mu, \nu)}{2} = d(\mu, \zeta) = d(\zeta, \nu)\}. \qquad (1) \end{aligned}$$

It is clear that, given $\mu, \nu \in \mathcal{FP}(X)$, we can denote by $\frac{\mu+\nu}{2}$ the fuzzy subset of X defined by

$$\left(\frac{\mu+\nu}{2}\right)_i = \frac{\mu_i+\nu_i}{2}$$

for every $i = 1, ..., n$. Moreover, if we consider the Euclidean distance d_2 on $\mathcal{FP}(X)$ defined by

$$d_2(\mu, \nu) = \sqrt{\sum_{i=1}^{n} (\nu_i - \mu_i)^2},$$

then it is a simpler matter to see, from the classical Euclidean geometry, that the unique midpoint between μ and ν it is the fuzzy subset $\frac{\mu+\nu}{2}$. Contrary to the Euclidean case J. Nieto and A. Torres showed that, for the Hamming distance, in general there is not a unique midpoint between two fuzzy sets ([17]).

Let us recall that the Hamming distance on $\mathcal{FP}(X)$ is defined, for every $\mu, \nu \in \mathcal{FP}(X)$, by

$$d_H(\mu, \nu) = \sum_{i=1}^{n} |\mu_i - \nu_i|.$$

In most cases for the Hamming distance the set of midpoints is an infinite set. From an applied knowledge point of view, the existence of many midpoints is in agreement with the nature of many real problems where the solution can be represented as a midpoint between two given "positions" which are identified with two fuzzy subsets, and where the solution fails to be the canonical middle between the mentioned positions. In fact the solution of the practical problem can be associated with a range of "middle ways"between the given positions. So, from practical point of view, to consider the Euclidean distance to compute midpoints is not two much useful. Since the set of midpoints depends on the chosen distance, to solve this type of practical problems consists of establishing the departure positions and what is the suitable distance to find the right midway as the working representation position (i.e. as the solution).

Recently, motivated by the applications to bioinformatics, J. Casasnovas and F. Roselló have generalized the previous work of Nieto and Torres. In fact they have computed segments and midpoints for several distances between fuzzy sets. In particular they gave a concise description of the midpoints between fuzzy sets for, among others, the weighted Hamming distance and obtained applications of their results to medicine and to comparison of biological sequences (see [1] and [2]).

Let us recall that given $w = (w_1, ..., w_n) \in (\mathbb{R}^+)^n$ (any vector of positive weights) the w-weighted Hamming distance on $\mathcal{FP}(X)$ is defined, for every $\mu, \nu \in \mathcal{FP}(X)$, by

$$d_{H,w}(\mu, \nu) = \sum_{i=1}^{n} w_i |\mu_i - \nu_i|.$$

Most recently, the notion of segment and midpoint between fuzzy sets has been generalized by Casasnovas and O. Valero to the context of asymmetric distances in [3]. This generalization consists of replacing in definitions (1) the distance by an asymmetric one. Moreover, in the same reference they defined the so-called w-weighted upper Hamming distance on $\mathcal{FP}(X)$ by

$$u_{H,w}(\mu,\nu) \quad = \quad \sum_{i=1}^{n} w_i u(\mu_i,\nu_i) = \sum_{i=1}^{n} w_i[(\nu_i-\mu_i)\vee 0].$$

Again, a precise description of segments and midsets was obtained for the above distance (see Lemma 5, Theorem 6, Corollary 7 and 8 in [3] for more details). The motivation for the study of midpoints between fuzzy subsets for an asymmetric distance is given by the following facts:

On the one hand, the w-weighted Hamming distance can be retrieved from the w-weighted upper Hamming distance because

$$d_{H,w}(\mu,\nu) = u_{H,w}(\mu,\nu) + u_{H,w}^{-1}(\mu,\nu)$$

for every $\mu,\nu \in \mathcal{FP}(X)$. So it seems natural to investigate the relationship between segments and midpoints for the symmetric and asymmetric weighted Hamming distance. In this case, one can find surprising results as the following one for the segments:

$$\mathrm{seg}_{d_{H,w}}(\mu,\nu) = \mathrm{seg}_{u_{H,w}}(\mu,\nu) = \mathrm{seg}_{u_{H,w}^{-1}}(\mu,\nu)$$

for all $\mu,\nu \in \mathcal{FP}(X)$.

On the other hand, A. Stojmirović has proved a natural correspondence between similarity measures on biological (nucleotide or protein) sequences and asymmetric distances, giving practice applications to search in DNA and protein datasets (see [26] and [27]). The obtained deep connection between asymmetric distances and similarity measures on biological sequences has motivated new directions of research in the realm of life sciences and, as a consequence, the importance of the asymmetric distances and its presence have been increased in bioinformatics.

In addition to the mentioned motivations it is important to recall that asymmetric distances play a crucial role in Theoretical Computer Science. In the last years metric tools based on asymmetric distances have been introduced and developed in order to provide an efficient framework to model processes, for instance, in complexity analysis of algorithms and programs ([25], [22], [6], [7], [8], [20], [21], [19]), logic programming ([23], [24], [11]), approximate reasoning ([9], [10], [11], [12]), and in program verification and denotational semantics ([15], [16], [18]).

In this paper, taking advantage of all theory carried out so far in the field of formal methods in complexity analysis of algorithms we give in Section 2 an application of the theory of midpoints for asymmetric distances to Computer Science. In particular we prove that, for the Largetwo algorithm, the average running time is a midpoint between the running time of computing of the best and the worst case by means of several connections between the w-weighted upper Hamming distance and the complexity measure introduced by M. Schellekens in [25]. As a consequence, a whole range of potential applications from midpoint theory of fuzzy sets to many fields in Computer Science and Artificial Intelligence is open.

2 The average running time of computing of Largetwo as a midpoint

In 1995, M. Schellekens introduced the theory of complexity (asymmetric distance) spaces as a part of the development of a topological foundation for the complexity analysis of programs and algorithms ([25]). In the same reference he applied the theory to the complexity analysis of algorithms giving an alternative proof of the well-known fact that the mergesort program based on a linear average time merge algorithm has optimal asymptotic average running time.

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

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

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

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

Obviously it is adopted the convention that $\frac{1}{+\infty} = 0$. The elements of $\mathcal{C}$ are called complexity functions.

According to [25], from a complexity analysis point of view, it is possible to associate each algorithm with a function of $\mathcal{C}$ in such a way that its computational cost, as a function of the size of the input data, is represented by any function f in $\mathcal{C}$. On the other hand, given two functions $f, g \in \mathcal{C}$ the numerical value $d_{\mathcal{C}}(f, g)$ (the complexity distance from f to g) can be interpreted as the relative progress made in lowering the complexity by replacing any program P with complexity function f by any program Q with complexity function g. Therefore, if $f \neq g$, the condition $d_{\mathcal{C}}(f, g) = 0$ can be assumed as f is "more efficient" than g on all inputs (i.e. $f(n) \leq g(n)$ for all $n \in \omega$) or equivalently f is more efficient than g asymptoticly. Note that this is consistent with the idea that when we replace the program P by the program Q, being the program Q less efficient on all inputs, we obtain that there is an increase in complexity. Furthermore, the asymmetry of the complexity distance plays a crucial role in this analysis because of a symmetric distance provides information about the increase of complexity but it can not indicate which program is more efficient. So this fact, among others, is a motivation for the use of asymmetric distances in formal methods for computing in general, and in complexity analysis of algorithms in particular.

Later on, S. Romaguera and M. Schellekens ([22]) introduced the so-called dual complexity space with the aim of studying several quasi-metric properties of the complexity space, which are interesting from a computational point of view, via the analysis of this new complexity (asymmetric distance) space.

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

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

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

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

This new complexity structure also allows to carry out the complexity analysis of algorithms when the complexity measure is the running time of computing.

Now, in the dual context the value $d_{\mathcal{C}^*}(f,g)$ can be interpreted as a numerical measure of the efficiency gained when the algorithm Q, whose running time of computing is represented by g, is substituted by the algorithm P whose running time of computing is represented by f. Hence, if $f\neq g$, $d_{\mathcal{C}^*}(f,g)=0$ provides that g is more "efficient" than f on all inputs.

Notice that the above structures provide mathematical tools to decide when a given program is better (from a complexity point of view) than another one in a uniform way (asymptotic way), i. e. the efficiency of both algorithms is calculated for all possible input sizes. However, in many practice situations it is interesting to measure relative progress made in lowering of complexity when an algorithm is replaced by another one, and both are evaluated on input data of size under a fixed upper bound. Motivated, in part, for this reason L.M. García-Raffi, Romaguera and E.A. Sánchez-Pérez have introduced in [6] a new criteria (a finite criteria or non-asymptotic criteria) for computing the gained efficiency when the evaluated programs run for a fixed finite subset of input sizes. In particular, if we compare two algorithms, with associated complexity funtions f and g in $\mathcal{C}^*$, for inputs whose size is lower than (or equal to) a bound, say $m\in\mathbb{N}$, then the relative progress made in lowering of complexity is given by the numerical value

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

It is clear that the equality $d_{\mathcal{C}^*_m}(f,g)=0$ can be interpreted, if $f\neq g$, as g is more "efficient" than f on all considered inputs of size m at most.

Obviously there are a lot of programs which can not be identified with a function in $\mathcal{C}^*$, since when one consider their running time of computing as a function of the size of the input data f then such a function does not belong to $\mathcal{C}^*$. Of course an example of such programs is given by the World Series odds problem (when this is solved by a recursive algorithm), where the running time of computing is in the class $\mathcal{O}(\frac{2^n}{\sqrt{n}})$ (i.e. the running time is given by the function $f:\omega\rightarrow\mathbb{R}^+$ defined by $f(0)=0$ and $f(n)=\frac{2^n}{\sqrt{n}}$ for all $n\in\mathbb{N}$). Consequently the analysis of the progress made in lowering the running time of computing when the problem is solved applaying the dynamic programming approach, instead of the recursive one, can not be made in the context of the dual complexity space $(\mathcal{C}^*,d_{\mathcal{C}^*})$. This handicap was avoided elegantly by García-Raffi, Romaguera and Sánchez-Pérez constructing several extensions of the dual complexity structure (see [8] and [7] for more details). From this point of view the old complexity space $(\mathcal{C},d_{\mathcal{C}})$ presents an advantage with respect the dual one, because of exponential time algorithms, as the World Series odds problem algorithm, has associated complexity function inside the set $\mathcal{C}$. So we can perform the complexity analysis of such algorithms using the original complexity space without the need of any added extension. For this reason we adapt the non-asymptotic criteria to the context of the original complexity space. To this end, fixed a input size bound $m \in \mathbb{N}$, we compute the relative progress made in lowering the complexity by replacing any program P with complexity function f by any program Q with complexity function g through the numerical value

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

Obviously, and similarly to the dual case, the equality $d_{\mathcal{C}_m}(f,g) = 0$ can be interpreted as f is more “efficient” than g on all considered inputs of size m at most, when $f \neq g$. Thus there is not progress, from a complexity point of view, when the program with complexity function f is replaced by the other one with complexity function g.

In the remainder of this section we are interesting in applying the fuzzy midpoint theory to the complexity analysis of algorithms. With this aim let us recall some basic aspects of the complexity analysis of a well-known algorithm which is called Largetwo.

The Largetwo algorithm finds the two largest entries in one-dimensional array and assigns these values to the variables FIRST and SEC. The pseudocode description of the algorithms is the following:

```
PROCEDURE Largetwo(C)
FIRST:=C[1]
SEC:=C[2]
FOR $I$=2  TO $n$ DO
    IF C[I] > FIRST
        THEN  SEC:=FIRST; FIRST:=C[$I$]
        ELSE  IF C[$I$] > SEC
                 THEN  SEC:=C[$I$]
```

Obviously the algorithm assumes that the arrray has at least two components (i.e. $n \geq 2$). We structure the analysis of the running time of this algorithm in three cases:

The best case: The data is arranged in order of increasing coordinates, i.e. C[1]<C[2]<$\cdots$ <C[n]. Then the algorithm performs exactly $n-1$ comparisons, and thus the best case running time is given by the complexity function f_B defined by

$$f_B(n) = \begin{cases} +\infty & n = 0,1 \\ n-1 & n \geq 2 \end{cases}.$$

The worst case: The data is given in such a way that C[1] is the largest entry. Then the algorithm makes $2(n-1)$ comparisons, and thus the worst case running time of computing is given by the complexity function f_W defined by

$$f_W(n) = \left\{ \begin{array}{cc} +\infty & n = 0, 1 \\ 2(n-1) & n \geq 2 \end{array} \right. .$$

The average case: The running time is calculated as the average of the running time of Largetwo over all inputs of the same size. For simplicity it is assumed that, fixed a size, each input is equally likely to occur. Then the number of comparisons used on average by the algorithm is given by the complexity function f_A defined by

$$f_A(n) = \left\{ \begin{array}{cc} +\infty & n = 0, 1 \\ 2(n-1) - \sum_{i=2}^{n} \frac{1}{i} & n \geq 2 \end{array} \right. .$$

For a detailed discussion of the running time of the Largetwo algorithm we refer the reader to [4]. In the same reference it is noticed that the average case running time f_A is more close to the worst case running time than to the best case running time, and thus the average case running time is not the numerical average of the worst case and the best case running time (i.e. the average running time does not match up with the midpoint for the Euclidean distance). Note that the average of the worst and the best case running time coincides with the function $\frac{f_W + f_B}{2}$ defined by

$$\frac{f_W + f_B}{2}(n) = \left\{ \begin{array}{cc} +\infty & n = 0, 1 \\ \frac{3}{2}(n-1) & n \geq 2 \end{array} \right. ,$$

which is, in fact, more close to the best case from a complexity point of view.

Next we connect the theory of midpoints between fuzzy sets showing that the average case running time is a midpoint between the best and the worst case running time for the asymmetric complexity distance $d_{\mathcal{C}}$ (i.e. asymptoticly) using as a tool the weighted upper Hamming distance (i.e. employing the non-asymptotic criteria). To this end we proceed as follows:

Fix $m \in \omega$ $(m \geq 2)$ and let $X = \{0, 1, ..., m\}$.

Take the vector of weights $w = (w_0, w_1, ..., w_m) \in (\mathbb{R}^+)^{m+1}$ such that $w_n = 2^{-n}$ for all $n = 0, ..., m$.

It is evident that $\frac{1}{f_W(n)}, \frac{1}{f_B(n)}, \frac{1}{f_A(n)} \in [0, 1]$ for all $n = 0, ..., m$. Recall that we adopt the convention that $\frac{1}{+\infty} = 0$.

From now on we will denote by $(\frac{1}{f_W})_m, (\frac{1}{f_B})_m, (\frac{1}{f_A})_m$ the fuzzy sets corresponding to the points

$$(0, 0, \frac{1}{f_W(2)}, \ldots, \frac{1}{f_W(m)}),\ (0, 0, \frac{1}{f_B(2)}, ..., \frac{1}{f_B(m)}) \text{ and } (0, 0, \frac{1}{f_A(2)}, ..., \frac{1}{f_A(m)})$$

of the Kosko hypercube $[0, 1]^{m+1}$, respectively. It follows that $(\frac{1}{f_W})_m, (\frac{1}{f_B})_m, (\frac{1}{f_A})_m \in \mathcal{FP}(X)$.

Under these conditions we can obtain the next results.

Proposition 1. $(\frac{1}{f_A})_m \in \text{mid}_{u_{H,w}}((\frac{1}{f_B})_m, (\frac{1}{f_W})_m)$.

Proof. It is clear that

$$\begin{aligned} d_{\mathcal{C}_{m,}}(f_B, f_W) &= \sum_{n=2}^{m} 2^{-n}\left((\frac{1}{f_W(n)} - \frac{1}{f_B(n)}) \vee 0 \right) \\ &= u_{H,w}((\frac{1}{f_B})_m, (\frac{1}{f_W})_m). \end{aligned}$$

On the other hand

$$u_{H,w}((\frac{1}{f_B})_m, (\frac{1}{f_W})_m) = \sum_{n=2}^{m} 2^{-n}\left((\frac{1}{2(n-1)} - \frac{1}{n-1}) \vee 0 \right) = 0,$$

since $\frac{1}{2(n-1)} \leq \frac{1}{n-1}$ for all $n = 2, ..., m$.

Moreover,

$$\begin{aligned} d_{\mathcal{C}_{m,}}(f_B, f_A) &= \sum_{n=2}^{m} 2^{-n}\left((\frac{1}{f_A(n)} - \frac{1}{f_B(n)}) \vee 0 \right) = u_{H,w}((\frac{1}{f_B})_m, (\frac{1}{f_A})_m) \\ &= \sum_{n=2}^{m} 2^{-n}\left((\frac{1}{2(n-1) - \sum_{i=2}^{n} \frac{1}{i}} - \frac{1}{(n-1)}) \vee 0 \right) = 0, \end{aligned}$$

since $\frac{1}{2(n-1) - \sum_{i=2}^{n} \frac{1}{i}} \leq \frac{1}{(n-1)}$ for all $n = 2, ..., m$.

Furthermore,

$$\begin{aligned} d_{\mathcal{C}_{m,}}(f_A, f_W) &= \sum_{n=2}^{m} 2^{-n}\left((\frac{1}{f_W(n)} - \frac{1}{f_A(n)}) \vee 0 \right) = u_{H,w}((\frac{1}{f_A})_m, (\frac{1}{f_W})_m) \\ &= \sum_{n=2}^{m} 2^{-n}\left((\frac{1}{2(n-1)} - \frac{1}{2(n-1) - \sum_{i=2}^{n} \frac{1}{i}}) \vee 0 \right) = 0, \end{aligned}$$

since $\frac{1}{2(n-1)} \leq \frac{1}{2(n-1) - \sum_{i=2}^{n} \frac{1}{i}}$ for all $n = 2, ..., m$.

Therefore

$$\frac{u_{H,w}((\frac{1}{f_B})_m, (\frac{1}{f_W})_m)}{2} = u_{H,w}((\frac{1}{f_B})_m, (\frac{1}{f_A})_m) = u_{H,w}((\frac{1}{f_A})_m, (\frac{1}{f_W})_m). \qquad (2)$$

Whence we conclude that $(\frac{1}{f_A})_m \in \mathrm{mid}_{u_{H,w}}((\frac{1}{f_B})_m, (\frac{1}{f_W})_m)$. ■

Corollary 2. $f_A \in \mathrm{mid}_{d_{\mathcal{C}}}(f_B, f_W)$.

Proof. Since the equality (2) is satisfied for all $m \in \omega$, we obtain that

$$\frac{d_{\mathcal{C}}(f_B, f_W)}{2} = d_{\mathcal{C}}(f_B, f_A) = d_{\mathcal{C}}(f_A, f_W).$$

Therefore

$$f_A \in \mathrm{mid}_{d_{\mathcal{C}}}(f_B, f_W),$$

as we claim. ■

References

[1] J. Casasnovas, F. Roselló, *Averaging fuzzy biopolymers,* Fuzzy Sets and Systems **152** (2005), 139-158.

[2] J. Casasnovas, F. Roselló, *Midpoints as average representations of pairs of descriptions by means of fuzzy subsets,* in Proc. Information Processing and Management of Uncertainty in Knowledge-based Systems international Conference (2005), 2157-2164..

[3] J. Casasnovas, O. Valero, *On midpoints for the weighted upper Hamming distance betwen fuzzy sets,* in Proc. Information Processing and Management of Uncertainty in Knowledge-based Systems international Conference (2006), 114-120.

[4] P. Cull, M. Flahive, R. Robson, *Difference equations: from rabbits to chaos,* Springer, 2005 .

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

[6] LM García-Raffi, S. Romaguera, E.A. Sánchez-Pérez, *Weak topologies on asymmetric normed linear spaces and non-asymptotic criteria in the theory of complexity analysis of algorithms,* Journal of Analysis and Applications **3** (2004), 125-138.

[7] L.M García-Raffi, S. Romaguera, E.A. Sánchez-Pérez, *The supremum asymmetric norm on sequence algebras: a general framework to measure complexity spaces,* Electronic Notes in Theoret. Comput. Sci. **74** (2003), 12 pages.

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

[9] G. Gerla, *Representation theorems for fuzzy orders and quasi-metrics,* Soft Computing **8** (2004), 571-580.

[10] G. Gerla, C. Crisconio, *Fuzzy orders in approximate reasoning,* unpublished paper.

[11] G. Gerla, C. Crisconio, *Similarities and fuzzy orders in approximate reasoning,* in New Logics for the New economy, G. Zollo (ed.), Ed. Scientifiche Italiane, 165-168, 2001.

[12] G. Gerla, *Fuzzy submonoids, fuzzy preorders and quasi-metrics,* Fuzzy Sets Syst. **157** (2006), 2356-2370.

[13] B. Kosko, *Neural networks and fuzzy systems,* Prentice-Hall, 1992.

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

[15] S.G. Matthews, *Partial metric topology,* in: Proc. 8th Summer Conference on General Topology and Applications, Ann. New York Acad. Sci. **728** (1994), 183-197.

[16] S.G. Matthews, *An extensional treatment of lazy data flow deadlok,* Theoret. Comput. Sci., **151** (1995), 195-205.

[17] J. Nieto, A. Torres, *Midpoints for fuzzy sets and their application in medicine,* Artif. Intell. Med. **27** (2003), 81-101.

[18] J. Rodríguez-López, S. Romaguera, O. Valero, *Denotational semantics for programming languages, balanced quai-metrics and fixed points,* Int. J. Comput. Math, to appear.

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

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

[21] S. Romaguera, E. A. Sánchez-Pérez, O. Valero, *The complexity space of a valued linearly ordered set,* Electronic Notes in Theoret. Comput. Sci., **74 (**2003**),** 14 pages.

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

[23] A. K. Seda, *Quasi-metrics and the semantics of logic programs,* Fundamenta Informaticae, **29** (1997), 97-117.

[24] A. K. Seda, *Some Issues Concerning Fixed Points in Computational Logic: Quasi-Metrics, Multivalued Mappings and the Knaster-Tarski Theorem*, in Proc. 14th Summer Conference on Topology and its Applications, Topology Proc. **24** (1999), 223-250.

[25] M. Schellekens, *The Smyth completion: a common foundation for denonational semantics and complexity analysis,* in Proc. MFPS 11, Electronic Notes in Theoret. Comput. Sci. **1** (1995), 211-232.

[26] A. Stojmirović, *Quasi-metric spaces with measure*, Topology Proc. **28** (2004), 655-671.

[27] A. Stojmirović, *Quasi-metrics, similarities and searches: aspects of geometry of protein datasets*, Ph. D. Th., Victoria University of Wellington, Wellington (2005).

[28] M. Zaus, *Crisp and soft computing with hypercubical calculus,* Physica-Verlag, 1999.