U.P.B. Sci. Bull., Series A, Vol. 77, Iss. 2, 2015 ISSN 1223-7027

A METHOD TO REPRESENT THE SEMANTIC DESCRIPTION OF A WEB SERVICE BASED ON COMPLEXITY FUNCTIONS

Andrei-Horia MOGOS[1], Adina Magda FLOREA[2]

Semantic web services represent an important and very active research area in computer science. The semantic description of a web service has a crucial role when working with semantic web services. In this paper we propose a method for representing the semantic description of a web service using complexity functions. The main result of our paper is a theorem that links the semantic descriptions, represented as complexity functions, with several important asymptotic notations.

Keywords: semantic description of a web service, complexity function, asymptotic notation.

1. Introduction

As specified in [1], "a web service is a software system designed to support interoperable machine-to-machine interaction over a network". A web service is a software system that runs on a computer in a network (for example, in the Internet) and offers a service that can be accessed by another computer from the same network.

The Semantic Web is a machine-readable and machine-processable web. As specified in [2], "the Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation". The Semantic Web is a web that allows machines (computers) to read, 'understand' and process the information; this is made by adding machine-processable semantics to the documents.

Semantic web services are web services with a formal semantic description [3]. Semantic web services represent a technology that combines the web services and the Semantic Web. The work on semantic web services is very active and a significant number of results were obtained in the last years**.**

In the area of semantic web services, the semantic description (the semantics) of a web service has a crucial role. The main approaches proposed in the literature for the semantics of web services are the following [4]: OWL-S [5], WSDL-S [6], and WSMO [7]. According to [8], some newer approaches are SAWSDL [9] and WSMO-Lite [10]. All these approaches represent the semantic

[1] Lecturer, Computer Science Department, University POLITEHNICA of Bucharest, Romania, e-mail: andrei.mogos@cs.pub.ro
[2] Professor, Computer Science Department, University POLITEHNICA of Bucharest, Romania, e-mail: adina.florea@cs.pub.ro

description of a web service by using ontologies expressed in XML-compatible languages. Thus, these approaches represent a semantic description as a set of words.

In this paper we propose a method of representing the semantic description of a web service using a special type of function, called complexity function. This way, each semantic description will be a complexity function, in fact a sequence of positive real numbers. The most important difference between our approach and the main approaches in the literature is that we use real numbers instead of words for representing semantic descriptions. The main result of our paper is a theorem that links the semantic descriptions represented by complexity functions with the asymptotic notations.

The paper is organized as follows. Section 2 contains a brief discussion about complexity functions and asymptotic notations. In Section 3 we propose a new approach with respect to the representation of the semantic descriptions of web services. Section 4 contains the main result of the paper: we propose and prove a theorem that links the semantic descriptions, represented using our approach, with the asymptotic notations. In the end, Section 5 presents the conclusions of the paper.

2. Complexity functions and asymptotic notations

We denote by N^* the set of positive integers and by R_+^* the set of positive real numbers. A *complexity function* is a function $N^* \to R_+^*$ (see, for example, [11], [12]).

Let be $g : N^* \to R_+^*$ an arbitrary, fixed complexity function. We present five well known *asymptotic notations* (see, for example, [11], [12], [13], [14]):

$$\Theta(g(n)) = \{ f : N^* \to R_+^* \mid \exists c_1, c_2 \in R_+^*, \exists n_0 \in N^* \text{ such that } c_1 \cdot g(n) \le f(n) \le c_2 \cdot g(n), \forall n \ge n_0 \} \quad (1)$$

$$O(g(n)) = \{ f : N^* \to R_+^* \mid \exists c \in R_+^*, \exists n_0 \in N^* \text{ such that } f(n) \le c \cdot g(n), \forall n \ge n_0 \} \quad (2)$$

$$\Omega(g(n)) = \{ f : N^* \to R_+^* \mid \exists c \in R_+^*, \exists n_0 \in N^* \text{ such that } c \cdot g(n) \le f(n), \forall n \ge n_0 \} \quad (3)$$

$$o(g(n)) = \{ f : N^* \to R_+^* \mid \forall c \in R_+^*, \exists n_0 \in N^* \text{ such that } f(n) < c \cdot g(n), \forall n \ge n_0 \} \quad (4)$$

$$\omega(g(n)) = \{ f : N^* \to R_+^* \mid \forall c \in R_+^*, \exists n_0 \in N^* \text{ such that } c \cdot g(n) < f(n), \forall n \ge n_0 \} \quad (5)$$

We also present other two *asymptotic notations* (see, [11]):

$$o\Theta(g(n)) = O(g(n)) \setminus (o(g(n)) \cup \Theta(g(n))) \quad (6)$$

$$\Theta\omega(g(n)) = \Omega(g(n)) \setminus (\Theta(g(n)) \cup \omega(g(n))) \quad (7)$$

3. Representing semantic descriptions using complexity functions

In this section we present our approach with respect to the representation of the semantic descriptions of web services.

3.1. Dictionary

Consider a set of words *W*. We make the following assumptions:

1) Each word from *W* has *a single meaning*;
2) From our context we can easily obtain the answers for the following questions:
   2a) Have two words *the same meaning*?
   2b) Is a word *more general* than another one?

Definition 1. The semantic equality of two words

$$\forall w_1, w_2 \in W,\ w_1 =_s w_2 \Leftrightarrow w_1 \textit{ and } w_2 \textit{ have the same meaning}.$$

Theorem 1. The binary relation $=_s$ on *W* is an equivalence relation.

Proof. The proof follows from Definition 1.

Theorem 2. All the words with the same meaning from the set *W* form an equivalence class.

Proof. The proof follows from Definition 1 and Theorem 1.

Let be $(C_i)_{1 \le i \le NC}$ the family of equivalence classes determined by $=_s$ on *W*.

Theorem 3. The family of equivalence classes $(C_i)_{1 \le i \le NC}$ forms a partition of *W*.

Proof. The proof follows from Definition 1, Theorem 1 and Theorem 2.

Definition 2. Dictionary

The dictionary associated to *W* and $=_s$ is $D = (C_i)_{1 \le i \le NC}$.

Definition 3. The meaning function

T*he meaning function m* associated to the dictionary *D* is:

$$m : W \to \{1, 2, ..., NC\},\ m(w) = i, \text{ where } w \in C_i \quad (8)$$

We say that $m(w)$ is *the meaning of the word w*.

Theorem 4. Properties of the meaning function

a) The meaning function *m* is surjective
b) If there exists $i \in \{1, 2, ..., NC\}$ such that $card(C_i) > 1$ then the meaning function *m* is not injective
c) If $card(C_i) = 1,\ \forall i \in \{1, 2, ..., NC\}$ then the meaning function *m* is bijective

Proof. The proof follows from Definition 3.

3.2. An order relation between the equivalence classes

Definition 4. The semantic inequality of two words

$\forall w_1, w_2 \in W,\ w_1 \le_s w_2 \Leftrightarrow$

w_2 *is more general than* w_1 *in terms of meaning or* $w_1 =_s w_2$

Example 1. If we consider the words *car* and *vehicle*, then we have the relations: *car* $\le_s$ *vehicle* and *car* $\ne_s$ *vehicle*.

Theorem 5.

a) The binary relation $\le_s$ on W is a preorder relation

b) If there exists $i \in \{1, 2, ..., NC\}$ such that $card(C_i) > 1$ then the binary relation $\le_s$ on W is not an order relation

c) If $card(C_i) = 1,\ \forall i \in \{1, 2, ..., NC\}$ then the binary relation $\le_s$ on W is an order relation

Proof. The proof follows from Definition 4.

Definition 5. The representative word of an equivalence class

For each equivalence class C_i, $i \in \{1, 2, ..., NC\}$ we choose a word $\overline{w}_i \in C_i$. This word will be called *the representative word of class* C_i.

Definition 6. The semantic equality of two equivalence classes

$\forall C_i, C_j \in D,\ C_i =_{sc} C_j \Leftrightarrow i = j$

Theorem 6. The binary relation $=_{sc}$ on D is an equivalence relation.

Proof. The proof follows from Definition 6.

Remark 1. Each equivalence class determined by $=_{sc}$ on D has only one member.

Definition 7. The semantic inequality of two equivalence classes

$\forall C_i, C_j \in D,\ C_i \le_{sc} C_j \Leftrightarrow \overline{w}_i \le_s \overline{w}_j$

Theorem 7. The binary relation $\le_{sc}$ on D is an order relation.

Proof. The proof follows from Definition 7.

Remark 2. One can observe that, if $C_i, C_j \in D$ with the properties $C_i \le_{sc} C_j$ and $i \ne j$ then $C_i \ne_{sc} C_j$.

3.3. Representation of Semantic Descriptions

Consider the following semantic description of a web service: $w_{i_1}\ w_{i_2}\ ...\ w_{i_l}$, where $w_{i_1}, w_{i_2}, ..., w_{i_l} \in W$.

We want to express this semantic description as a complexity function. For this purpose, we will present several steps for transforming the initial semantic description into a complexity function.

As a first step the semantic description will be represented as follows:

$$sd1:\{1,2,...,NC\}\to\{0,1\}$$
$$sd1(n)=\begin{cases}1, & if\ \exists w\in\{w_{i_1},w_{i_2},...,w_{i_l}\}\ such\ that\ w\in C_n\\ 0, & otherwise\end{cases}\tag{9}$$

As a second step, we extend the domain of the function *sd1* to N^*. A new semantic description is obtained:

$$sd2:N^*\to\{0,1\},\quad sd2(n)=sd1(n \bmod^* NC)\tag{10}$$

where we define $\bmod^*$ as following: $n \bmod^* NC = n \bmod NC$, if $n \bmod NC \neq 0$ and $n \bmod^* NC = NC$, if $n \bmod NC = 0$.

Observe that the function $sd2$ may return the value 0 and a complexity function has only positive real values. Also, we want that the result of the function to depend on *n*. The new semantic description is the following:

Definition 8. The semantic description of a web service

The semantic description of a web service is:

$$sd:N^*\to R_+^*,\quad sd(n)=\begin{cases}n+1, & if\ sd2(n)=1\\ 1/(n+1), & if\ sd2(n)=0\end{cases}\tag{11}$$

Example 2. Consider the dictionary *D*. Suppose we have the following semantic description expressed in words: "sell white car". Also suppose that $sell\in C_{n_1}$, $white\in C_{n_2}$, and $car\in C_{n_3}$. The semantic description in form (11) is: $sd(n)=n+1, if\ (n \bmod^* NC)\in\{n_1,n_2,n_3\}$ and $sd(n)=1/(n+1), otherwise$.

Definition 9. Two approximations of a semantic description

Let be $Q_1(n,p)$ and $Q_2(n,p)$ two predicates with the following forms:

$$\begin{aligned}&Q_1(n,p)=n,p\in\{1,2,...,NC\},n\neq p,such\ that\\ &sd(n)=1/(n+1),sd(p)=p+1\ and\ C_n\leq_{sc}C_p\\ &Q_2(n,p)=n,p\in\{1,2,...,NC\},n\neq p,such\ that\\ &sd(n)=n+1,sd(p)=1/(p+1)\ and\ C_n\leq_{sc}C_p\end{aligned}\tag{12}$$

Let be $n_0,p_0\in\{1,2,...,NC\}$ such that $Q_1(n_0,p_0)$. *An approximation of type 1 of the semantic description sd with respect to* n_0 *and* p_0 is the following:

$$\overline{sd}^{(1)}_{n_0,p_0}:N^*\to R_+^*,\quad \overline{sd}^{(1)}_{n_0,p_0}(n)=\begin{cases}(n+1)^2, & if\ n=n_0\\ 1/(n+1), & if\ n=p_0\\ sd(n), & otherwise\end{cases}\tag{13}$$

Let be $n_0,p_0\in\{1,2,...,NC\}$ such that $Q_2(n_0,p_0)$. *An approximation of type 2 of the semantic description sd with respect to* n_0 *and* p_0 is the following:

$$\overline{sd}^{(2)}_{n_0,p_0} : N^* \to R^*_+, \quad \overline{sd}^{(2)}_{n_0,p_0}(n) = \begin{cases} 1/(n+1), & if\ n = n_0 \\ n+1, & if\ n = p_0 \\ sd(n), & otherwise \end{cases} \tag{14}$$

Example 3. a) Suppose we have a semantic description *sd* and two indexes $i, j \in \{1, 2, ..., NC\}, i \neq j$. Consider that $sd(i) = 1/(i+1)$ and $sd(j) = j+1$. Also consider that the class C_i corresponds to the meaning *car* and the class C_j corresponds to the meaning *vehicle*. An approximation of type 1 of *sd* represents a weaker semantic description (the one that has *car* instead of *vehicle*).

b) Suppose we have a semantic description *sd* and two indexes $i, j \in \{1, 2, ..., NC\}, i \neq j$. Consider that $sd(i) = i+1$ and $sd(j) = 1/(j+1)$. Also consider that the class C_i corresponds to the meaning *car* and the class C_j corresponds to the meaning *vehicle*. An approximation of type 2 of *sd* represents a stronger semantic description (the one that has *vehicle* instead of *car*).

3.4. The set of all semantic descriptions

Definition 10. The set of all semantic descriptions

We denote by *ASD* the set of all semantic descriptions, both exact and approximate.

Definition 11. The equality of two semantic descriptions

$$\forall sd_i, sd_j \in ASD,\ sd_i =_{sd} sd_j \Leftrightarrow sd_i = sd_j$$

Theorem 8. The binary relation $=_{sd}$ on *ASD* is an equivalence relation.

Proof. The proof follows from Definition 11.

Definition 12. The inequality of two semantic descriptions

$$\forall sd_i, sd_j \in ASD, \quad sd_i \leq_{sd} sd_j \Leftrightarrow sd_i(n) \leq sd_j(n), \forall n \in N^*$$

Theorem 9. The binary relation $\leq_{sd}$ on A*SD* is an order relation.

Proof. The proof follows from Definition 12.

Remark 4. Let be $sd_i, sd_j \in ASD$. We make the following conventions:

a) We write $sd_i <_{sd} sd_j$ if $sd_i(n) < sd_j(n), \forall n \in N^*$.

b) Another form of the relation $sd_i \leq_{sd} sd_j$ is $sd_j \geq_{sd} sd_i$.

c) Another form of the relation $sd_i <_{sd} sd_j$ is $sd_j >_{sd} sd_i$.

4. Semantic descriptions and asymptotic notations

This section contains the main result of the paper: a theorem that links the semantic descriptions and the asymptotic notations.

Theorem 10. Semantic descriptions and asymptotic notations

a) $\forall sd_i, sd_j \in ASD, \quad sd_i =_{sd} sd_j \Leftrightarrow sd_i(n) = \Theta(sd_j(n))$

b) $\forall sd_i, sd_j \in ASD, \quad sd_i \leq_{sd} sd_j \Leftrightarrow sd_i(n) = O(sd_j(n))$

c) $\forall sd_i, sd_j \in ASD, \quad sd_i \geq_{sd} sd_j \Leftrightarrow sd_i(n) = \Omega(sd_j(n))$

d) $\forall sd_i, sd_j \in ASD, \quad sd_i <_{sd} sd_j \Leftrightarrow sd_i(n) = o(sd_j(n))$

e) $\forall sd_i, sd_j \in ASD, \quad sd_i >_{sd} sd_j \Leftrightarrow sd_i(n) = \omega(sd_j(n))$

f) $\forall sd_i, sd_j \in ASD,$
$sd_i \leq_{sd} sd_j, sd_i \nless_{sd} sd_j \text{ and } sd_i \neq_{sd} sd_j \Leftrightarrow sd_i(n) = o\Theta(sd_j(n))$

g) $\forall sd_i, sd_j \in ASD,$
$sd_i \geq_{sd} sd_j, sd_i \ngtr_{sd} sd_j \text{ and } sd_i \neq_{sd} sd_j \Leftrightarrow sd_i(n) = \Theta\omega(sd_j(n))$

Proof. a) "$\Rightarrow$" If $sd_i =_{sd} sd_j$ then $sd_i = sd_j$; from the reflexivity of Θ we have that $sd_i(n) = \Theta(sd_j(n))$. "$\Leftarrow$" Let be $sd_i(n) = \Theta(sd_j(n))$. From Definition 8 and Definition 9 the values of the functions sd_i and sd_j for the argument *n* can be $(n+1)^2$, *n*+1 or 1/(*n*+1). Suppose (for a contradiction) that $\exists \overline{n} \in N^*$ such that $sd_i(\overline{n}) \neq sd_j(\overline{n})$. Consider that $(\overline{n} \bmod^* NC) = k$, where $k \in \{1, 2, ..., NC\}$.

a1) If $sd_i(\overline{n}) = 1/(\overline{n}+1)$ and $sd_j(\overline{n}) = \overline{n}+1$ then, from Definition 8 and Definition 9, we have that $sd_i(p) = 1/(p+1)$ and $sd_j(p) = p+1$, $\forall p \in N^*$ such that $(p \bmod^* NC) = k$. From $sd_i(n) = \Theta(sd_j(n))$ we have:

$$\begin{aligned} &\exists c_1, c_2 \in R_+^*, \exists n_0 \in N^* \textit{ such that} \\ &c_1 \cdot sd_j(n) \leq sd_i(n) \leq c_2 \cdot sd_j(n), \forall n \geq n_0 \end{aligned} \quad (15)$$

Let be $\overline{p} \in N^*$ with $(\overline{p} \bmod^* NC) = k$, $\overline{p} \geq n_0$ and $c_1 \cdot (\overline{p}+1)^2 > 1$. From (15) we have $c_1 \cdot (\overline{p}+1) \leq 1/(\overline{p}+1) \leq c_2 \cdot (\overline{p}+1)$. It follows that, $c_1 \cdot (\overline{p}+1)^2 \leq 1$; this contradicts the inequality $c_1 \cdot (\overline{p}+1)^2 > 1$.

a2) If $sd_i(\overline{n}) = 1/(\overline{n}+1)$ and $sd_j(\overline{n}) = (\overline{n}+1)^2$ then the result follows using the same idea used for a1).

a3) If $sd_i(\overline{n}) = \overline{n}+1$ and $sd_j(\overline{n}) = (\overline{n}+1)^2$ then the result follows using the same idea used for a1).

a4) If $sd_i(\overline{n}) = \overline{n}+1$ and $sd_j(\overline{n}) = 1/(\overline{n}+1)$, we use the symmetry of Θ: $sd_i(n) = \Theta(sd_j(n)) \Leftrightarrow sd_j(n) = \Theta(sd_i(n))$ and then the same idea used for a1).

a5) If $sd_i(\overline{n}) = (\overline{n}+1)^2$ and $sd_j(\overline{n}) = 1/(\overline{n}+1)$ then the result follows using the same idea used for a4).

a6) If $sd_i(\overline{n}) = (\overline{n}+1)^2$ and $sd_j(\overline{n}) = \overline{n}+1$ then the result follows using the same idea used for a4).

For all the six cases we found a contradiction. Consequently, $sd_i(n) = sd_j(n), \forall n \in N^*$. It follows that $sd_i =_{sd} sd_j$.

b) "$\Rightarrow$" If $sd_i \leq_{sd} sd_j$ then $sd_i(n) \leq sd_j(n), \forall n \in N^*$. It follows that

$$\begin{aligned} &\exists c = 1 \in R_+^*, \exists n_0 = 1 \in N^* \textit{ such that} \\ &sd_i(n) \leq c \cdot sd_j(n), \forall n \geq n_0 \end{aligned} \tag{16}$$

Consequently, $sd_i(n) = O(sd_j(n))$.

"$\Leftarrow$" Let be $sd_i(n) = O(sd_j(n))$. From Definition 8 and Definition 9 the values of the functions sd_i and sd_j for the argument n can be $(n+1)^2$, n+1 or 1/(n+1). Suppose (for a contradiction) that $\exists \overline{n} \in N^*$ such that $sd_i(\overline{n}) > sd_j(\overline{n})$. Consider that $(\overline{n} \bmod^* NC) = k$, where $k \in \{1, 2, ..., NC\}$.

b1) If $sd_i(\overline{n}) = \overline{n} + 1$ and $sd_j(\overline{n}) = 1/(\overline{n} + 1)$ then, from Definition 8 and Definition 9, we have that $sd_i(p) = p + 1$ and $sd_j(p) = 1/(p+1)$, $\forall p \in N^*$ such that $(p \bmod^* NC) = k$. From $sd_i(n) = O(sd_j(n))$ we have:

$$\begin{aligned} &\exists c \in R_+^*, \exists n_0 \in N^* \textit{ such that} \\ &sd_i(n) \leq c \cdot sd_j(n), \forall n \geq n_0 \end{aligned} \tag{17}$$

Let be $\overline{p} \in N^*$ with $(\overline{p} \bmod^* NC) = k$, $\overline{p} \geq n_0$ and $(\overline{p}+1)^2 > c$. From (17) we have $\overline{p} + 1 \leq c \cdot 1/(\overline{p}+1)$. It follows that $(\overline{p}+1)^2 \leq c$; this contradicts the inequality $(\overline{p}+1)^2 > c$.

b2) If $sd_i(\overline{n}) = (\overline{n}+1)^2$ and $sd_j(\overline{n}) = 1/(\overline{n}+1)$ then the result follows using the same idea used for b1).

b3) If $sd_i(\overline{n}) = (\overline{n}+1)^2$ and $sd_j(\overline{n}) = \overline{n} + 1$ then the result follows using the same idea used for b1).

Consequently, $sd_i(n) \leq sd_j(n), \forall n \in N^*$. It follows that $sd_i \leq_{sd} sd_j$.

c) The proof follows from b) and the transposed symmetry of O and Ω: $sd_j(n) = O(sd_i(n)) \Leftrightarrow sd_i(n) = \Omega(sd_j(n))$

d) "$\Rightarrow$" If $sd_i <_{sd} sd_j$ then $sd_i(n) < sd_j(n), \forall n \in N^*$. From Definition 8 and Definition 9 the values of the functions sd_i and sd_j for the argument n can be $(n+1)^2$, n+1 or 1/(n+1).

d1) If $sd_i(n) = 1/(n+1), \forall n \in N^*$ and $sd_j(n) = n+1, \forall n \in N^*$ then $sd_i(n) = o(sd_j(n))$.

d2) If $sd_i(n) = 1/(n+1), \forall n \in N^*$ and $sd_j(n) = (n+1)^2, \forall n \in N^*$ then $sd_i(n) = o(sd_j(n))$.

d3) If $sd_i(n) = n+1, \forall n \in N^*$ and $sd_j(n) = (n+1)^2, \forall n \in N^*$ then $sd_i(n) = o(sd_j(n))$.

Consequently, $sd_i(n) = o(sd_j(n))$.

"$\Leftarrow$"Let be $sd_i(n) = o(sd_j(n))$. Since $o(sd_j(n)) \subseteq O(sd_j(n))$ and using b) we have that $sd_i \leq_{sd} sd_j$; consequently, $sd_i(n) \leq sd_j(n), \forall n \in N^*$. Suppose (for a contradiction) that $\exists \overline{n} \in N^*$such that $sd_i(\overline{n}) = sd_j(\overline{n})$. Consider that $(\overline{n} \bmod^* NC) = k$, where $k \in \{1, 2, ..., NC\}$. From Definition 8 and Definition 9, we have that $sd_i(p) = sd_j(p)$, $\forall p \in N^*$ such that $(p \bmod^* NC) = k$. From $sd_i(n) = o(sd_j(n))$ we have:

$$\begin{aligned} &\forall c \in R_+^*, \exists n_0 \in N^* \textit{ such that} \\ &sd_i(n) < c \cdot sd_j(n), \forall n \geq n_0 \end{aligned} \tag{18}$$

Let be $\overline{p} \in N^*$with $(\overline{p} \bmod^* NC) = k$, $\overline{p} \geq n_0$ and let be $\overline{c} = 1 \in R_+^*$. Using (18) we have $sd_i(\overline{p}) < 1 \cdot sd_i(\overline{p})$ and this is a contradiction. Consequently, $sd_i(n) < sd_j(n), \forall n \in N^*$. It follows that $sd_i <_{sd} sd_j$.

e) The proof follows from d) and the transposed symmetry of o and ω: $sd_j(n) = o(sd_i(n)) \Leftrightarrow sd_i(n) = \omega(sd_j(n))$.

f) Using a), b), d) and (6) we have:

$$sd_i(n) = o\Theta(sd_j(n)) \Leftrightarrow \begin{cases} sd_i(n) = O(sd_j(n)) \\ sd_i(n) \neq o(sd_j(n)) \\ sd_i(n) \neq \Theta(sd_j(n)) \end{cases} \Leftrightarrow \begin{cases} sd_i \leq_{sd} sd_j \\ sd_i \not<_{sd} sd_j \\ sd_i \neq_{sd} sd_j \end{cases} \tag{19}$$

g) The proof follows from f) and the transposed symmetry of $o\Theta$ and $\Theta\omega$: $sd_j(n) = o\Theta(sd_i(n)) \Leftrightarrow sd_i(n) = \Theta\omega(sd_j(n))$.

5. Conclusions

In this paper we propose a method for representing the semantic description of a web service based on complexity functions. While the main approaches in the literature represent a semantic description as a set of words, we represent a semantic description as a complexity function, and thus as a sequence of real numbers.

The main result of the paper consists of a theorem that links the semantic descriptions expressed as complexity functions with several important asymptotic notations. This theorem shows that any two semantic descriptions (represented with our approach) can be compared using asymptotic notations.

Our method is simple and easy to use. Also, a semantic description represented as a complexity function can be easily read and interpreted by computers (software programs), since it consists of real numbers. Consequently, our approach is suitable to be used for various research topics in the area of semantic web services.

Acknowledgement

The work has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/159/1.5/S/132397.

R E F E R E N C E S

[1]. *** "Web Services Architecture" W3C Working Group Note, February 2004, see http://www.w3.org/TR/ws-arch/, last accessed June 2014

[2]. *T. Berners-Lee, J. Hendler and O. Lassila*, "The Semantic Web", in Scientific American, May 2001

[3]. *F. Lécué, A. Delteil*, "Making the Difference in Semantic Web Service Composition", in Proceedings of the 22nd Conference on Artificial Intelligence (AAAI-07), 22-26 July 2007, Vancouver, British Columbia, Canada, vol. 2, pp. 1383-1388

[4]. *J. Cardoso*, "Approaches to Developing Semantic Web Services", in International Journal of Computer Science, **vol. 1**, no. 1, 2006, pp. 8-21

[5]. *** "OWL-S: Semantic Markup for Web Services", W3C Member Submission, November 2004, see http://www.w3.org/Submission/OWL-S/, last accessed June 2014

[6]. *** "Web Service Semantics - WSDL-S", W3C Member Submission, November 2005, see http://www.w3.org/Submission/WSDL-S/, last accessed June 2014

[7]. *** "Web Service Modeling Ontology (WSMO)", W3C Member Submission, June 2005, see http://www.w3.org/Submission/WSMO/, last accessed June 2014

[8]. *M. Lanthaler, M. Granitzer and C. Gutl*, "Semantic Web Services: State of the Art", in Proceedings of IADIS International Conference on Internet Technologies & Society 2010, 29 November – 1 December 2010, Perth, Australia, pp. 107-114

[9]. *** "Semantic Annotations for WSDL and XML Schema", W3C Recommendation, August 2007, see http://www.w3.org/TR/sawsdl/, last accessed June 2014

[10]. *** "WSMO-Lite: Lightweight Semantic Descriptions for Services on the Web", W3C Member Submission, August 2010, see http://www.w3.org/Submission/WSMO-Lite/, last accessed June 2014

[11]. *A. H. Mogos, A. M. Florea*, "A Method to Compare Two Complexity Functions Using Complexity Classes", in UPB Scientific Bulletin, Series A: Applied Mathematics and Physics, **vol. 72**, iss. 2, 2010, pp. 69-84.

[12]. *C. A. Giumale*, "Introducere in Analiza Algoritmilor. Teorie si Aplicatie (Introduction to the Analysis of Algorithms. Theory and Application)", Polirom, Bucharest, Romania, 2004, pp. 32-38

[13]. *T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein*, "Introduction to Algorithms", Second edition, MIT Press, Cambridge Massachusetts, London England, USA, 2001, pp. 23-26, pp. 41-50

[14]. *A. H. Mogos*, "Three Variants of the Master Theorem", in IEEE Proceedings of CSCS 2013, the 19th International Conference on Control Systems and Computer Science, 29-31 May 2013, Bucharest, Romania, pp. 162-166