An Algebraic Valuation of Words

Adrian ATANASIU   Victor MITRANA

University of Bucharest, Faculty of Mathematics
Str. Academiei 14, 70109, Bucharest, Romania
email: *aadrian@pcnet.ro*, *mitrana@funinf.math.unibuc.ro*

Abstract. This paper proposes an algebraic way of sentence valuation in a semiring. Actually, throughout the paper only valuations in the ring of integers with usual addition and multiplication are considered. These valuations take into consideration both words and their positions within sentences. Two synonymy relations, with respect to a given valuation, are introduced. All sentences that are synonynous form a synonymy class. Some basic problems regarding the synonymy classes, which are actually formal languages, are formulated and studied. Some of them are completely solved whereas partial answers are provided for others.

1 Introduction

A series of paper [1], [2], [4], [5] have dealt with homomorphisms h from a free generated monoid M to the monoid $((0, \infty), \cdot, 1)$, so that the sum of all homomorphical images of generators of M equals 1, called Bernoulli homomorphisms (distributions, measures). Besides being homomorphisms, Bernoulli homomorphisms may be viewed as probability measures on the family of all languages over a given alphabet. Furthermore, they played an important role in developing the theory of codes [1]. Some authors discarded the homomorphism property keeping the probability measure property as done in [4], [5] whilst others proceeded vice versa [3], calling them valuations.

In this paper, we introduce a generalization of the aforementioned valuations in the following sense. The value of a word depends not only on its letters but also on their positions within the word. Furthermore, the valuation is computed in a richer structure that of a semiring instead of a monoid.

2 Definitions and examples

An *alphabet* is a finite nonempty set whose elements are called *letters (symbols)*; if $V = \{a_1, a_2, \ldots, a_n\}$ is an alphabet, then any sequence $w = a_{i_1}a_{i_2}\ldots a_{i_k}, 1 \leq i_j \leq n, 1 \leq j \leq k$, is called *word (string)* over V. The length of the aforementioned word w is denoted by $lg(w)$ and equals k. The *empty word* is denoted by $\varepsilon, lg(\varepsilon) = 0$. Moreover, $(x)_U$ delivers the string obtained from x by removing all letters not in U. The set of all words over V is denoted by V^* and $V^+ = V^* - \{\varepsilon\}$. For all notions and notations in formal language theory that are not defined here the reader may consult [7].

A structure $(A, +, \cdot, 0, 1)$ is called a *semiring* iff the following conditions are satisfied for all $a, b, c \in A$:

(i) $(A, +, 0)$ is a commutative monoid,

(ii) $(A, \cdot, 1)$ is a monoid,

(iii) $a \cdot (b + c) = a \cdot b + a \cdot c,\ (a + b) \cdot c = a \cdot c + b \cdot c$,

(iv) $0 \cdot a = a \cdot 0 = 0$.

The semiring $(A, +, \cdot, 0, 1)$ is said to be commutative iff $(A, \cdot, 1)$ is a commutative monoid. For further notions we refer to [6].

Let V be an alphabet and $(A, +, \cdot, 0, 1)$ be a commutative semiring. A *valuating system* of V^* is a pair of mappings

$$\phi = (\alpha, \beta),$$

where

- $\alpha : V \longrightarrow A$, (the letter attributing function; $\alpha(a)$ is said to be the attribute of a),
- $\beta : \mathcal{N} \longrightarrow A$, (the position attributing function; $\beta(n)$ is the attribute of the position n).

Given a valuating system ϕ as above and a string $x = a_1a_2\ldots a_n \in V^*, a_i \in V,\ 1 \leq i \leq n$, we define

$$val_\phi(x) = \sum_{i=1}^{n} \alpha(a_i) \cdot \beta(i).$$

Moreover, $val_\phi(\varepsilon)$ delivers always 0, for any valuating system ϕ.

Two words x, y are equivalent with respect to the valuation system ϕ, written as $x \sim_\phi y$, iff $val_\phi(x) = val_\phi(y)$. The equivalence class of x is defined as

$$[x]_\phi = \{y \in V^* | x \sim_\phi y\}.$$

Example 1 *Let us consider the semiring $\mathcal{Z}[X]$ of all polynomials with only one indeterminate and coefficients in $\mathcal{Z}$ together with addition and multiplication. We consider the valuating system of the alphabet $V = \{a, b, c, d\}$ over $\mathcal{Z}[X]$, $\phi = (\alpha, \beta)$ defined as follows:*

α	a	b	c	d
	$2X^2$	$X^2 - 1$	1	$2X^2 - 1$

$$\beta(i) = X + i, \ \ i \in \mathcal{N}.$$

It is easy to note that

$$val_\phi(dacb) = val_\phi(aba) = 5X^3 + 10X^2 - X - 1$$

which implies $dacb \sim_\phi aba$.

Example 2 *Take $V = \{a, b, c\}$ and the valuating system of V^* over $(\mathcal{Q}, +, \cdot, 0, 1)$ $\phi = (\alpha, \beta)$ with*

$$\begin{aligned} \alpha(a) &= \frac{1}{2}, \ \alpha(b) = \frac{1}{3}, \ \alpha(c) = -\frac{1}{6} \\ \beta(n) &= 5. \end{aligned}$$

The reader may easily verify that

$$[\varepsilon]_f = L_1 \cup L_2 \cup L_3,$$

where

$$\begin{aligned} L_1 &= \{x | 3lg((x)_a) = lg((x)_c), lg((x)_b) = 0\}, \\ L_2 &= \{x | 2lg((x)_b) = lg((x)_c), lg((x)_a) = 0\}, \\ L_3 &= \{x | 3lg((x)_a) + 2lg((x)_b) = lg((x)_c)\}. \end{aligned}$$

Note that this language is not context-free.

We are going to investigate mainly the equivalence classes. A natural problem concerns the finiteness of these sets as well as the possibility to decide on this problem. Furthermore, we are concerning with the problem of finding appropriate devices (automata, grammars, etc.) which characterize the equivalence classes. To this end, in this paper we shall only consider the valuation systems over the ring of integers $\mathcal{Z}$ with the usual addition and multiplication. The absolute value of an integer x is denoted by $|x|$. In the sequel, we shall foccus our attention on valuating systems whose function β is either polynomial or exponential.

3 The polynomial case

Let $x = a_1 a_2 \ldots a_n$ be a string in V^* and $\phi = (\alpha, \beta)$ be a valuating system of V over $\mathcal{Z}$ with

$$\beta(n) = c_0 n^k + c_1 n^{k-1} + \ldots + c_k.$$

Denote by

$$\phi_i = (\alpha, n^i),\ 0 \leq i \leq k.$$

Clearly,

$$val_\phi(x) = \sum_{i=0}^{k} c_i \cdot val_{\phi_{k-i}}(x). \tag{1}$$

The relation stated by the next lemma will be useful in the sequel.

Lemma 1 $val_{\phi_k}(xy) = val_{\phi_k}(x) + \sum_{i=0}^{k} \binom{i}{k} lg(x)^i val_{\phi_{k-i}}(y)$.

Proof. Assume that $y = y_1 y_2 \ldots y_p$, $y_j \in V, 1 \leq j \leq p$. Starting from

$$val_{\phi_k}(xy) = val_{\phi_k}(x) + \sum_{j=1}^{p} (lg(x) + j)^k \alpha(y_j)$$

by a direct calculation one gets the desired equation. □

Now, we restrict our investigation to two polynomials only: constant and linear.

3.1 The constant polynomial

As $[x]_\phi = V^*$, for all $x \in V^*$, iff β is the null function, we shall consider only non-zero position attributing functions. Note that the relations stated by Lemma 1 and (1), respectively, may be combined in

$$val_\phi(xy) = val_\phi(x) + val_\phi(y), \tag{2}$$

relation that the next theorem is based on.

Theorem 1 *Let $\phi = (\alpha, \beta)$ be a valuating system of V^* over $\mathcal{Z}$.*

1. *For each x, $[x]_\phi$ is finite if and only if $\alpha(a) \cdot \alpha(b) > 0$, for any $a, b \in V$.*
2. *Given a string $x \in V^*$, one can decide the finiteness of $[x]_\phi$.*

Proof. Obviously, the latter assertion immediately follows from the first one. Assume that $\alpha(a) > 0, \alpha(b) > 0$, for any $a, b \in V$; the reasoning is the same when all attributes of the letters in V are negative. As β is a constant function, say k, for each $x \in V^+$ $val_\phi(x)$ is either positive or negative depending on the sign of k. We shall only analyse the case when $k > 0$; the case $k < 0$ may be treated similarly. We claim that $[x]_\phi$ cannot contain strings of length bigger than $val_\phi(x)$. Indeed, if

$lg(y) > val_\phi(x)$, then $val_\phi(y) > k \cdot val_\phi(x)$, hence $y \notin [x]_\phi$. Consequently, $[x]_\phi$ is finite.

Now, let us consider that exist $a, b \in V$ such that $\alpha(a) \cdot \alpha(b) \leq 0$. We claim that exists $y \in V^+$ such that $val_\phi(y) = 0$. Clearly, y exists if $\alpha(a) = 0$, for some $a \in V$. Therefore, it suffices to assume that $\alpha(a) \cdot \alpha(b) < 0$. One may takes the string $y = a^{|\alpha(b)|} b^{|\alpha(a)|}$ for which

$$val_\phi(y) = k \cdot (\alpha(a) \cdot |\alpha(b)| + \alpha(b) \cdot |\alpha(a)|) = 0.$$

In conclusion, by equation 2, for each x, all strings $xy^m, m \geq 0$, are in $[x]_\phi$, which ends the proof. □

We recall now an operation on words that will turn out to be very useful for our investigation regarding the type of languages $[x]_\phi$. This operation, called *shuffle* is a well-known operation in formal language theory and in parallel programming theory. It is defined, for the strings $x, y \in V^*$, as follows

$$\begin{aligned} Shuf(x,y) \;\; = \;\; & \{x_1 y_1 x_2 y_2 \ldots x_p y_p \mid x = x_1 \ldots x_p, y = y_1 \ldots y_p, \\ & p \geq 1, x_i, y_i \in V^*, 1 \leq i \leq p\}. \end{aligned}$$

A shuffle of two strings is an arbitrary interleaving of the substrings of the original strings For two languages $L_1, L_2 \subseteq V^*$, we define

$$Shuf(L_1, L_2) = \bigcup_{x \in L_1, y \in L_2} Shuf(x, y).$$

Let $\phi = (\alpha, \beta)$ be a valuating system of V^* and U be a subset of V. For each $x \in V^*$ we denote by

$$M([x]_\phi, U) = \{y \in [x]_\phi | val_\phi((y)_U) = val_\phi((x)_U)\}.$$

Obviously, if $\bar{V} = \{a \in V | \alpha(a) = 0\}$, then

$$[x]_\phi = Shuf(M([x]_\phi, V - \bar{V}), \bar{V}^*). \tag{3}$$

Theorem 2 *Let $\phi = (\alpha, \beta)$ be a valuating system of V^* and $\bar{V} = \{a \in V | \alpha(a) = 0\}$.*

1. If $card(V - \bar{V}) \leq 1$, then $[x]_\phi$ is regular, for any $x \in V^*$.

2. If $card(V - \bar{V}) = 2$, then $[x]_\phi$ is context-free, for any $x \in V^*$. Furthermore, one can decide when $[x]_\phi$ is regular.

3. If $card(V - \bar{V}) = 3$, then $[x]_\phi$ is context sensitive, for any $x \in V^*$. Furthermore, $[x]_\phi$ is context-free iff it is regular and this can be algorithmically decided.

Proof. 1. Obviously, when $card(V - \bar{V}) = 0$ there exists just one equivalence class that is V^*. If $V - \bar{V} = \{a\}$, then $M([x]_\phi)$ is finite, consequently, by equation 3, the language $[x]_\phi$ is regular.

2. Let $V - \bar{V} = \{a, b\}$. If $\alpha(a) \cdot \alpha(b) > 0$, then $M([x]_\phi)$ is finite (Theorem 1), hence $[x]_\phi$ is regular. If $\alpha(a) \cdot \alpha(b) < 0$, then

$$M([x]_\phi) = \{z \in V^* | lg((z)_a) \cdot \alpha(a) + lg((z)_b) \cdot \alpha(b) = val_\phi(x)\}$$

which is a context-free language but not regular. As shuffling a context-free language with a regular language one gets a context-free language, the second assertion is completely proved.

3. By a similar reasoning to the previous one, we have

- There exist $a, b \in V - \bar{V}$ such that $\alpha(a) \cdot \alpha(b) < 0$; in conclusion, $M([x]_\phi)$ is a context sensitive language that is not context-free.
- For each pair $a, b \in V - \bar{V}$, $\alpha(a) \cdot \alpha(b) > 0$ holds; consequently, $M([x]_\phi)$ is finite.

□

3.2 The linear polynomial

Let $\phi = (\alpha, \beta)$ be a valuation system with $\beta(n) = kn + p$. By Lemma 1 and relation 1, one may write also

$$val_\phi(xy) = val_\phi(x) + val_\phi(y) + k \cdot lg(x) \cdot val_{\phi_0}(y). \quad (4)$$

Having the aforementioned relation, one may claim that Theorem 1 remains valid for the linear case as well.

Theorem 3 *Let $\phi = (\alpha, \beta)$ be a valuating system of V^* over $\mathcal{Z}$.*

1. For each x, $[x]_\phi$ is finite if and only if $\alpha(a) \cdot \alpha(b) > 0$, for any $a, b \in V$.

2. Given a string $x \in V^*$, one can decide the finiteness of $[x]_\phi$.

Proof. The proof foccuses only on the first assertion, the latter one obviously follows from the first one. Assume that $\alpha(a) > 0, \alpha(b) > 0$, for any $a, b \in V$; the case $\alpha(a) < 0, \alpha(b) < 0$ may be treated analogously. Moreover, assume that $\beta(n) = kn + p, n \in \mathcal{N}$. We analyse what happens when $k < 0$; the reasoning may be carried over the case $k > 0$ with minor changes. Clearly, there is $n_0 \in \mathcal{N}$ such that $val_\phi(x) < 0$, for all strings in V^* longer than n_0. Let x be such a word. We claim that $val_\phi(y) < val_\phi(x)$, for all $y \in V^*$ such that $lg(y) \geq lg(x) \cdot |val_\phi(x)|$. Due to the length of y, one infers that

$$val_\phi(y) \leq |val_\phi(x)| \cdot val_\phi(y_1 y_2 \ldots y_{lg(x)}).$$

But, $|val_\phi(x)| \cdot val_\phi(y_1 y_2 \ldots y_{lg(x)}) < val_\phi(x)$ because $val_\phi(y_1 y_2 \ldots y_{lg(x)})$ is negative too. Consequently, $[x]_\phi$ is finite for all $x \in V^*$.

Let us consider that exist $a, b \in V$ such that $\alpha(a) \cdot \alpha(b) < 0$. As in the proof of Theorem 1, the word $y = a^{|\alpha(b)|}b^{|\alpha(a)|}$ satisfies $val_{\phi_0}(y) = 0$. Denoting $\phi_1(\alpha, \beta_1)$ the valuating system with $\beta_1(n) = n$, we claim that $val_{\phi_1}(zz^R) = 0$, for every $z \in V^*$ with $val_{\phi_0}(z) = 0$. Here z^R denotes the mirror image of z. Indeed, if $z = z_1z_2\ldots z_m,\ z_i \in V$,

$$val_{\phi_1}(zz^R) = (2m+1)val_{\phi_0}(z) = 0.$$

Note also that $val_{\phi_0}(zz^R) = 0$, too.

Now, as

$$val_\phi(zz^R) = k \cdot val_{\phi_1}(zz^R) + p \cdot val_{\phi_0}(zz^R)$$

one gets $val_\phi(zz^R) = 0$. Due to the relation

$$val_\phi(xy) = val_\phi(x) + val_\phi(y) + k \cdot lg(x) \cdot val_{\phi_0}(y)$$

one concludes that all strings $x(zz^R)^q,\ q \geq 0$, with z as above, belong to $[x]_\phi$. $\square$

As far as the position of languages $[x]_\phi$ in the Chomsky hierarchy is concerned, we are not able to provide an exhaustive characterization like in the constant polynomial case.

Theorem 4 *Let $\phi = (\alpha, \beta)$ be a valuating system of V^*. The language $[x]_\phi$ is context-sensitive, for any $x \in V^*$.*

Proof. Let us suppose that $\beta(n) = kn + p$. For each $x \in V^*$ with $val_\phi(x) \geq 0$ one construct a phrase-structure grammar $G_x = (N, V, S, P)$ which works accordingly with the next nondeterministic procedure:

1. For each string $a_1a_2\ldots a_n \in V^+$ the grammar generates the sentential form $Xa_1a_2\ldots a_nYZ,\ X, Y, Z \in N$.
2. While the current sentential form contains letters in V and no trap symbol do
   - If the suffix of the current sentential form is $Y(-1)^qZ$, for some $q \geq 0$, then
     - if no position in between X and Y is occupied by a letter $a \in V$ with $\alpha(a) \geq 0$, then block the derivation by a trap symbol;
     - otherwise, choose a position i in between X and Y occupied by a_i with $\alpha(a_i) \geq 0$ and
       * transform a_i into a new symbol b_i, not in V,
       * write $1^{\alpha(a_i)(ki+p)}$ before Z,
       * remove each pair of consecutive symbols $-1, 1$, in between Y and Z.

- If the suffix of the current sentential form is $Y1^qZ$, for some $q \geq 0$, then
  - if no position in between X and Y is occupied by a letter $a \in V$ with $\alpha(a) < 0$, then
    - if $q > val_\phi(x)$, then block the derivation;
    - else choose a position i between X and Y occupied by a_i with $\alpha(a_i) \geq 0$ and
      - transform a_i into a new symbol b_i, not in V,
      - write $1^{\alpha(a_i)(ki+p)}$ before Z;
  - otherwise, choose a position i between X and Y occupied by a_i with $\alpha(a_i) < 0$ and
    - transform a_i into a new symbol b_i, not in V,
    - write $(-1)^{\alpha(a_i)(ki+p)}$ after Y,
    - remove each pair of consecutive symbols $-1, 1$, in between Y and Z.

3. If the current sentential form does not contain any trap symbol, check whether its suffix is $Y1^{val_\phi(x)}Z$. In the affirmative, remove all symbols 1 and X, Y, Z, otherwise block the derivation by a trap symbol.

Clearly, G_x generates a string in V^+ if and only if its valuation with respect to ϕ equals the valuation of x. Moreover, we should add the empty string to $L(G_x)$ when $val_\phi(x) = 0$.

Note that the working space [7] of each $z \in L(G_x)$ is bounded as follows

$$WS(z, G_x) \leq max(lg(z) + 4 + 2t(k \cdot lg(z) + p), lg(z) + 4 + 2 \cdot val_\phi(x)),$$

where $t = max\{|\alpha(a)| \ : a \in V\}$. Finally, it is enough to notice that the proof may be carried over the case when $val_\phi(x) < 0$, with the appropriate changes. □

4 The exponential case

The subject of investigation in this section is the valuation systems class whose position attributing function is exponential. To this end, let $\phi = (\alpha, \beta)$ be a valuating system with $\beta(n) = a^n$, $n \geq 1, a \in \mathcal{Z}\backslash\{0, 1\}$.

Clearly,

$$val_\phi(xy) = val_\phi(x) + a^{lg(x)} val_\phi(y). \tag{5}$$

As we have seen in the previous sections, a crucial point in proofs was the decidability of the problem concerning the existence of a string $w \in V^+$ such that $val_\phi(w) = 0$. This problem is still decidable in our case.

Theorem 5 *Let $\phi = (\alpha, \beta)$ be a valuation system of V^*, $\beta(i) = a^i, i \in \mathcal{N}$.*

1. If $a = -1$, there always exist strings $x \in V^+$ such that $val_\phi(x) = 0$.

2. If $a \in \mathcal{Z} \backslash \{0, -1\}$, one can decide whether or not exists a string $w \in V^+$ such that $val_\phi(w) = 0$ in $O(np^2)$, where $n = card(V)$ and $p = max\{|\alpha(x)| \mid x \in V\}$.

Proof. 1. If $a = -1$, then the valuation of each string $a_i^{2p},\ 1 \leq i \leq n,\ p \geq 1$, equals 0.

2. Assume that $V = \{a_1, a_2, \ldots, a_n\}$ and $\beta(n) = a^n, n \geq 1$. It is easy to notice that exists a string $w \in V^+$ such that $val_\phi(w) = 0$ if and only if there exists a polynomial P whose coefficients are in the set $C = \{\alpha(x) | x \in V\}$ such that $P(a) = 0$. Suppose that $w = x_1 x_2 \ldots x_m, x_i \in V, 1 \leq i \leq m$. For

$$val_\phi(w) = a(\alpha(x_1) + a\alpha(x_2) + \ldots a^{m-1}\alpha(x_m))$$

and $a \neq 0$, it follows that the required polynomial P is

$$P(X) = \alpha(x_1) + X\alpha(x_2) + \ldots X^{m-1}\alpha(x_m).$$

One can distinguish two cases: $a = 1$ and $|a| \geq 2$.

If $a = 1$, we are dealing with a valuation system whose position attributing function is constant; this situation has been treated in the proof of Theorem 1.

Assume now that $|a| \geq 2$. Let $p = max\{|x| \mid x \in C\}$. The following algorithm decides, for any given $|a| \geq 2$, whether there is a polynomial with coefficients in C that has the zero a.

Algorithm 1.
INPUT: $\phi = (\alpha, \beta)$
OUTPUT: YES, if the polynomial exists
NO, otherwise.

begin
$C := \{\alpha(x) | x \in V\}$;
$D_1 := \{0\}; D := \emptyset$;
repeat
$D := D \cup D_1$; $R := \emptyset$;
for *each* $u \in D$ **do**
for *each* $v \in C$ **do**
if *u+v mod a=0* **then** $R := R \cup \{u + v \ div\ a\}$;
if $0 \in R$ **then** *"YES"*; **halt**;
else $D_1 := D_1 \cup R$;
until $D = D_1$;
"NO";
end.

In order to finish the proof, we need a reasoning for the correctness of the above algorithm.

Termination. We claim that at each step when a number $u + v\ div\ a$, $u \in D$ and $v \in C$, is added to R, this number is between $-p$ and p. Indeed, initially the assertion is valid. Assume that at an arbitrary moment, when entering the **repeat...until** loop, all elements of D are bounded by $-p$ and p, respectively. For $|a| \geq 2$, every multiple of a of the form $u + v$, $u \in D, v \in C$, is in the interval $[-2p, 2p]$, hence $u + v\ div\ a$ is in $[-p, p]$. Consequently, either $0 \in R$, during the loop or $D = D_1$ after this loop has been performed at most $2p$ times.

Correctness. Assume that the algorithm provides 0 in R at some step. This implies the existence of some $k \geq 1$ such that

$$\alpha(a_{i_1}) + \frac{1}{a}(\alpha(a_{i_2}) + \frac{1}{a}(\ldots + \frac{1}{a}(\alpha(a_{i_{k-1}}) + \frac{1}{a}\alpha(a_{i_k}))\ldots)) = 0$$

or equivalently

$$\alpha(a_{i_1})a^{k-1} + \alpha(a_{i_2})a^{k-2} + \ldots + \alpha(a_{i_k}) = 0.$$

It follows that $val_\phi(a_{i_k}a_{i_{k-1}}\ldots a_{i_1}) = 0$. Obviously, if the algorithm ends with $D = D_1$, then there is no string y such that $val_\phi(y) = 0$.

Finally, one may easily notice that the algorithm requires $O(np^2)$ time. $\Box$

This result was used in the precedent cases for deciding the finiteness of $[x]$, provided that x is a given string in V^*. However, we are not able to settle this problem for the exponential case. The last theorem only allows us to point out:

Corollary 1 *Let ϕ be a valuating system of V^* and x be a given string.*

1. *If $a = -1$, then $[x]_\phi$ is always infinite.*
2. *If $a = 1$, then one can decide whether $[x]_\phi$ is infinite.*
3. *$[x]_\phi$ is infinite if the aforementioned algorithm provides a solution.*

5 Final remarks

We briefly discuss here some considerations that seem to be in order.

- In the present paper one has considered a commutative semiring; it is natural to study valuating systems over other (non-commutative) semirings.
- Our approach tries to valuate all strings over an alphabet. A more natural idea is to valuate just those strings which belong to a given language. An attractive class of languages could be the context-free one.
- All valuating systems presented in this paper may be viewed as deterministic systems since each of their functions associates exactly one attribute to every argument. It appears to be more natural to associate a finite set of attributes

to every argument. Thus, the valuation of a given word is actually a set of attributes. Of course, the relation $\sim_\phi$, defined as

$$x \sim_\phi y \text{ iff } val_\phi(x) \cap val_\phi(y) \neq \emptyset$$

is not transitive anymore; however this relation might be called "synonymy".

We hope to return extensively on these remarks in a forthcoming paper.

References

[1] J. Berstel and D. Perrin, *Theory of Codes*, Pure and Applied Mathematics, Academic Press, 1985.

[2] J. Berstel and C. Reutenauer, *Rational Series and Their Languages*, EATCS Monographs on Theoretical Computer Science, vol. 12, Springer, Berlin, 1988.

[3] H. Fernau, Valuation of languages, with applications to fractal geometry, *Theoret. Comput. Sci.* **137** (1995) 177–217.

[4] G. Hansel and D. Perrin, Codes and Bernoulli partitions, *Math. Systems Theory* **16** (1983) 133–157.

[5] G. Hansel and D. Perrin, Rational probability measures, *Theoret. Comput. Sci.* **65** (1989) 171–188.

[6] W. Kuich and A. Salomaa, *Semirings, Automata, Languages*, EATCS Monographs on Theoretical Computer Science, vol. 5, Springer-Verlag, 1986.

[7] A. Salomaa, *Formal Languages*, Academic Press, 1973.