The Realization Problems Related to Weighted Transducers over Strong Bimonoids

Ping Li, Yongming Li and Shengling Geng

Abstract— In this paper, the concepts of weighted transducers over strong bimonoids and their input-output-functions are introduced. Further more, the input-functions and outputfunctions induced by the input-output-functions of weighted transducers over strong bimonoids are given. It is the most important that the input-functions and output-functions of weighted transducers over strong bimonoids can be realized by weighted finite automata over strong bimonoids, and the realization does not depend on the distributive law, which also embodies the applications of weighted finite automata over strong bimonoids.

I. INTRODUCTION

EIGHTED FINITE AUTOMATA over strong bi-monoids have been Proposed in [1] and [2]. Weighted finite automata over strong bimonoids are also called fuzzy transducers valued in strong bimonoids. Strong bimonoids can be viewed as semirings which might lack distributivity. Semirings, complete (orthomedular) lattices [3] and lattice ordered QMV algebras [4] are all the special cases of strong bimonoids. Therefore, weighted automata is one of the most extensive uncertain computing models at present. We know that weighted finite automata and weighted transducers (weighted finite automata with output) valued in semirings have both a well elaborated theory as well as practical applications [5]. It is well known that completed residuated lattice-valued [6], lattice-ordered monoids [7]-[10] are also the special cases of semirings. Particularly, the properties of weighted transducers and their applications at speech processing are studied in many other papers, such as [11]-[15].

It is the goal of this paper to study the properties and applications of weighted transducers over strong bimonoids (fuzzy automata with output valued in strong bimonoids) or fuzzy transducers valued in strong bimonoids). Concretely speaking, for some works related to weighted automata (without output or with output) over strong bimonoids which can be solved through using weighted transducers or weighted finite automata over strong bimonoids. In this paper, we mainly discuss some realization problems related to weighted transducers over strong bimonoids. That is, the input-functions and output-functions induced by the inputoutput functions of weighted transducers over strong bimonoids can be recognized by weighted automata (without output) over strong bimonoids.

The rest of the paper is arranged as follows. In Section 2, we recall some basic notions about strong bimonoids, give the definition of formal power series over strong bimonoids and some properties of them are studied. In Section 3, the definition of weighted transducers over strong bimonoids and their input-output-functions are given. In Section 4, we proposed the notions of the input-functions and output-functions induced by the input-output-functions of weighted transducers over strong bimonoids. And the realizations of the output-functions and input-functions are given (see Theorem 1 and Theorem 2). That is, for a strong bimonoid P, we write the input-function f_i and outputfunction f_o induced by the input-output-function of P-WT. Then we constructed a finite weighted automaton \mathcal{I} over strong bimonoid $(P\langle \langle \Sigma^* \rangle \rangle, +, \odot, \mathbf{0}, \varepsilon)$ such that $\mathcal{R}_{\mathcal{I}} = f_i$ and a finite weighted automaton O over strong bimonoid $(P\langle \langle \Omega^* \rangle \rangle, +, \odot, \mathbf{0}, \varepsilon)$ such that $\mathcal{R}_{\mathcal{O}} = f_o$. And an example is given for displaying the applications of the conclusions. Section 5 is a summary of this paper, and in this section, we made some further work prospects.

II. ALGEBRAIC NOTIONS

Here we collect standard definitions concerning strong bimonoids, semirings, lattice ordered QMV algebras and so on. For a more detailed introduction to these concepts we refer the reader to [1]-[4]

Definition 1: A strong bimonoid P is a set together with two binary operations + and \bullet , and two constant elements 0 and 1 such that:

(i) (P, +, 0) is a commutative monoid;

(ii) $(P, \bullet, 1)$ is a monoid;

(iii) $0 \bullet a = a \bullet 0 = 0$ for any $a \in P$.

As usual, we identify the structure $(P, +, \bullet, 0, 1)$ with its carrier set P.

A strong bimonoid P is called commutative if $a \bullet b = b \bullet a$ for any $a, b \in P$.

A strong bimonoid P is called left distributive (right distributive, resp.) if $a \bullet (b+c) = a \bullet b + a \bullet c$ ($(a+b) \bullet c = a \bullet c + b \bullet c$, resp.) for any $a, b, c \in P$.

A semiring is a strong bimonoid which is left and right distributive.

A strong bimonoid P is called additively idempotent (multiplicatively idempotent, resp.) if a + a = a ($a \bullet a = a$ resp.) for any $a \in P$.

Definition 2: Let (P, \leqslant) be a partially ordered set , and $(P, +, \bullet, 0, 1)$ be a strong bimonoid. If P satisfies the

Ping Li is with the College of Mathematics and Information Science, Shaanxi Normal University, Xi'an,China (email: liping@snnu.edu.cn).

Yongming Li is with the College of Computer Science, Shaanxi Normal University, Xi'an, China (email:liyongm@snnu.edu.cn).

Shengling Geng is with the College of Computer, Qinghai Normal University, Xining, China (email: qhnugsl@gmail.com).

This work was supported by National Science Foundation of China under Project Codes 11301321,11271237,61261047 and 11201279

following conditions,

(i) $a \leq b \Longrightarrow a + x \leq b + x$ for any $x \in P$;

(ii) $a \leq b \Longrightarrow a \bullet x \leq b \bullet x$ and $x \bullet a \leq x \bullet b$ for any $x \in P$ with $0 \leq x$.

then P is called an ordered strong bimonoid.

Moreover, P is called a positive-ordered strong bimonoid if $0 \leq a$ for every $a \in P$.

Next, we give some important algebras which are the special cases of strong bimonoids. The more specific examples of strong bimonoids cf. [1] and [2].

Example 1: (1) a semiring $(S, +, \bullet, 0, 1)$ is a strong bimonoid which is left and right distributive (cf.[1]).

(2) A complete (orthomedular) lattice $(L, \lor, \land, 0, 1)$ is a strong bimonoid which is additively idempotent and multiplicatively idempotent (cf.[3]).

(3) A lattice ordered QMV algebra $\mathcal{E} = (E, \wedge, \oplus, \mathbf{1}, \mathbf{0})($ where, we only care for the operations \wedge and \oplus) which is additively idempotent. Where, \wedge is induced by the two operations \oplus and \prime of QMV algebra $\mathcal{E} = (E, \oplus, \prime, \mathbf{0}, \mathbf{1})(cf.[4]).$

We noted that, complete (orthomedular) lattices and lattice ordered QMV algebra are all positive-ordered strong bimonoids. Therefore, strong bimonoids are more general algebraic structures.

Let U be an nonempty set, a P-valued subset on U is a mapping $A: U \longrightarrow P$, where, for every $u \in U, A(u)$ is called the *weight* of u. A P-valued subset on U is often written as $A = \sum_{u \in U} \frac{A(u)}{u}$, if U is a finite set. We write P^U for the set of all P-valued subsets on U, that is $P^U = \{A|A: U \longrightarrow P\}$.

Let Σ be an alphabet, Σ^* denote the set of all words of finite length over Σ and ε denotes the empty word, and Pbe a strong bimonoid. A formal power series over Σ and Pis a mappings $r : \Sigma^* \to P$. $\forall s \in \Sigma^*$, it is usual to write (r, s) for r(s) and r itself is written as a formal sum

$$r = \sum_{s \in \Sigma^*} (r, s) s.$$

where, (r, s)s is called a term of r and (r, s) the coefficient of (r, s)s. In general, if (r, s) = 0, then (r, s)s will be omit. The collection of all formal power series over Σ and P is denoted by $P\langle \langle \Sigma^* \rangle \rangle$.

Given $r \in P\langle \langle \Sigma^* \rangle \rangle$, the support of r is the set

$$supp(r) = \{ s \in \Sigma^* \mid r(s) \neq 0 \}.$$

A series $r \in P\langle\langle \Sigma^* \rangle\rangle$ where every coefficient equals 0 or 1 is termed the characteristic series of its support L, in symbols, r = char(L) or $r = 1_L$. The subset of $P\langle\langle \Sigma^* \rangle\rangle$ consisting of all series with a finite support is denoted by $P\langle \Sigma^* \rangle$ are referred to as polynomials. It will be convenient to use the notations $P\langle \Sigma \cup \{\varepsilon\}\rangle$, $P\langle \Sigma \rangle$ for the collection of polynomials having their supports in $\Sigma \cup \{\varepsilon\}$, Σ .

Examples of polynomials belong to $P\langle \Sigma^* \rangle$ are **0** and *as*, where $a \in P$ and $s \in \Sigma^*$, defined by

$$\begin{aligned} (\mathbf{0},s) &= 0, \forall s \in \Sigma^*. \\ (as,s') &= \left\{ \begin{array}{ll} a, & s' = s, \\ 0, & \text{otherwise.} \end{array} \right. \end{aligned}$$

Often, 1s is denoted by s or $\mathbf{1}_s$.

Next, we introduce several operations on $P\langle\langle \Sigma^* \rangle\rangle$. For $r, r_1, r_2 \in P\langle\langle \Sigma^* \rangle\rangle$, $a \in P$, we define the sum $r_1 + r_2$, the (Cauchy) product $r_1 \odot r_2$, the Hadamard product $r_1 \cdot r_2$, and scalar product ar, ra, each as a seies belonging to $P\langle\langle \Sigma^* \rangle\rangle$, as follows:

 $\begin{array}{l} (1) \ (r_1+r_2,s)=(r_1,s)+(r_1,s).\\ (2) \ (r_1\odot r_2,s)=\sum_{s_1s_2=s}(r_1,s_1)\bullet(r_2,s_2).\\ (3) \ (r_1\cdot r_2,s)=(r_1,s)\bullet(r_2,s). \end{array}$

(4) $(ar, s) = a \bullet (r, s).$

- (5) $(ra,s) = (r,s) \bullet a.$
- $a\varepsilon \odot r = r$ for any $a \in P$ and $r \in P\langle \langle \Sigma^* \rangle \rangle$.

It can be checked that $(P\langle\langle \Sigma^* \rangle\rangle, +, \cdot, \mathbf{0}, \varepsilon)$, $(P\langle\Sigma^* \rangle, +, \cdot, \mathbf{0}, \varepsilon)$ and $(P\langle\langle\Sigma^* \rangle\rangle, +, \odot, \mathbf{0}, char(\Sigma^*))$ are strong bimonoids. Since P does not satisfies the distributive laws in general, the (Cauchy) product $r_1 \odot r_2$ does not satisfies the associative law in general. But we have the following conclusion.

Proposition 1: Let P be a strong bimonoid and Σ an alphabet.

(i) $\forall r_i = \sum_{k_i=1}^{N_i} a_{ik_i} \sigma_{ik_i} \in P\langle \Sigma_{\varepsilon} \rangle (i = 1, 2, 3; N_i \text{ are positive integers}), it holds that$

$$(r_1 \odot r_2) \odot r_3 = r_1 \odot (r_2 \odot r_3) \in P\langle \Sigma^* \rangle.$$

(ii) $\forall r_i = \sum_{k_i=1}^{N_i} a_{ik_i} \sigma_{ik_i} \in P\langle \Sigma_{\varepsilon} \rangle (i = 1, 2, \cdots, n; N_i)$ are positive integers), then the value of $r_1 \odot r_2 \odot \cdots \odot r_n$ is unique in every way of adding brackets. This value is denoted by $r_1 \odot r_2 \odot \cdots \odot r_n (\in P\langle \Sigma^* \rangle)$, and

$$r_{1} \odot r_{2} \odot \cdots \odot r_{n} = \sum_{k_{1}=1}^{N_{1}} a_{1k_{1}} \sigma_{1k_{1}} \odot \sum_{k_{2}=1}^{N_{2}} a_{2k_{2}} \sigma_{2k_{2}} \odot \cdots \odot \sum_{k_{n}=1}^{N_{n}} a_{nk_{n}} \sigma_{nk_{n}} = \sum_{k_{1}=1}^{N_{1}} \sum_{k_{2}=1}^{N_{2}} \cdots \sum_{k_{n}=1}^{N_{n}} (a_{1k_{1}} \bullet a_{2k_{2}} \bullet \cdots \bullet a_{nk_{n}})$$

Proof: (i) It is immediate by the definition of the (Cauchy) product \odot .

(ii) it followed by (i), the definition of the (Cauchy) product \odot and mathematical induction.

From the definition of scalar product, we have the following theorem.

Proposition 2: Let P be a strong bimonoid and Σ an alphabet. For any $a \in P, r = a_1s_1 + a_2s_2 + \cdots + a_ns_n \in P\langle \Sigma^* \rangle$, we have ar and $ra \in P\langle \Sigma^* \rangle$ and

(i) $ar = a(a_1s_1 + a_2s_2 + \dots + a_ns_n) = (a \bullet a_1)s_1 + (a \bullet a_2)s_2 + \dots + (a \bullet a_n)s_n.$

(ii)
$$ra = (a_1s_1 + a_2s_2 + \dots + a_ns_n)a = (a_1 \bullet a)s_1 + (a_2 \bullet a)s_2 + \dots + (a_n \bullet a)s_n.$$

By the definitions of the sum +, pruduct \odot and proposition 2, it is straighforward to see the following two conclusions.

Proposition 3: Let P be a strong bimonoid and Σ an alphabet. $\forall r_1 = a_1 s, r_2 = a_2 s, \cdots, r_n = a_n s \in P\langle \Sigma^* \rangle$, it hold that $r_1 + r_2 + \cdots + r_n \in P\langle \Sigma^* \rangle$, and

 $r_1 + r_2 + \dots + r_n = a_1 s + a_2 s + \dots + a_n s = (a_1 + a_2 + \dots + a_n)s.$

Proposition 4: For any $a, b \in P$ and $r \in P\langle\langle \Sigma^* \rangle\rangle$, then (i) $a \varepsilon \odot r = ar$.

(ii) $r \odot b\varepsilon = rb$.

(iii) $a\varepsilon \odot r \odot b\varepsilon = (ar)b = a(rb)$.

III. WEIGHTED TRANSDUCERS OVER STRONG BIMONOIDS

Now, we will recall the definition of nondeterministic weighted finite automata over strong bimonoids without outputs.

Let P be a strong bimonoid. A nondeterministic weighted finite automaton over P (which is also called a nondeterministic P-valued weighted finite automaton, P-NFA, for short) is a five-tuple $\mathcal{A} = (Q, \Sigma, \delta, I, F)$, where

(i) Q is a non-empty finite set of states, Σ is a non-empty finite set of symbols.

(ii) $I: Q \longrightarrow P$ is a *P*-valued initial state, $F: Q \longrightarrow P$ is a *P*-valued finial (accepted) state.

(iii) $\delta: Q \times \Sigma \times Q \longrightarrow P$ is a *P*-valued transition function.

The behavior of \mathcal{A} in run semantics way is defined as, $\mathcal{R}_{\mathcal{A}} \in P^{\Sigma^*}, \forall s = \sigma_1 \cdots \sigma_n \in \Sigma^*,$

$$\mathcal{R}_{\mathcal{A}}(s) = \sum_{\substack{q_0, q_1, \cdots, q_n \in Q \\ \bullet \delta(q_{n-1}, \sigma_n, q_n) \bullet F(q_n)}} I(q_0) \bullet \delta(q_0, \sigma_1, q_1) \bullet \cdots$$

 $\mathcal{R}_{\mathcal{A}} \in P^{\Sigma^*}$ ia also been called the *P*-valued languages accepted (recognized) by \mathcal{A} .

Next, we will give the formal definition of nondeterministic weighted finite automata over strong bimonoids with outputs, i.e. weighted transducers over strong bimonoids.

Definition 3: Let P be a strong bimonoid. A weighted transducer over P (which is also called a P-valued weighted transducer, P-WT, for short) is a six-tuple $\mathcal{T} = (Q, \Sigma, \Omega, \delta, I, F)$, where

(i) Q is a non-empty finite set of states, Σ is a non-empty finite set of input alphabet, Ω is a non-empty finite set of output alphabet.

(ii) $I: Q \longrightarrow P$ is a *P*-valued initial state, $F: Q \longrightarrow P$ is a *P*-valued finial (accepted) state.

(iii) $\delta: Q \times \Sigma_{\varepsilon} \times \Omega_{\varepsilon} \times Q \longrightarrow P$ is a *P*-valued inputoutput transition function, where, $\Sigma_{\varepsilon}, \Omega_{\varepsilon}$ denote respectively $\Sigma \cup \{\varepsilon\}, \Omega \cup \{\varepsilon\}.$

Let $E_{\mathcal{T}} = \{(p, x, y, q) \in Q \times \Sigma_{\varepsilon} \times \Omega_{\varepsilon} \times Q \mid \delta(p, x, y, q) \neq 0\}$. For any $e = (p, x, y, q) \in E_{\mathcal{T}}$, the weight of e is $\delta(e)$, the input-output symbols of e is (x, y), written as alp(e), that is alp(e) = (x, y), and the input symbol of e is x, written as Ialp(e), the output symbol of e is y, written as Oalp(e). The current state of e is p denoted by c(e). The successor state of e is q, denoted by s(e).

Let $p(\mathcal{T}) = E_{\mathcal{T}} \cup \{\pi \mid \pi = e_1 e_2 \cdots e_n (n \ge 2), e_i \in E_{\mathcal{T}}, i = 1, 2, \cdots, n \text{ and } \pi \text{ satisfies } s(e_{i+1}) = c(e_i), i = 1, 2 \cdots n - 1\}.$

For any $\pi \in p(\mathcal{T})$, π is called a path of \mathcal{T} . For example, $\pi_1 = (q_3, \sigma_1, \varepsilon, q_1)$ and $\pi_2 = (q_0, \sigma, \varepsilon, q_1)(q_1, \varepsilon, \omega, q_2)$ are two paths of \mathcal{T} , but $\pi_3 = (q_0, \sigma_1, \varepsilon, q_1)(q_3, \sigma_2, \omega, q_2)$ is not a path of \mathcal{T} .

For any $\pi = e_1 e_2 \cdots e_n (n \ge 1) \in p(\mathcal{T})$, the inial state of e denoted by o(e), the finial state of e denoted by d(e). The weight of e, denoted by w(e), is defined by

$$w(\pi) = \delta(e_1) \bullet \delta(e_2) \bullet \dots \bullet \delta(e_n). \tag{1}$$

The input string and output string of e, denoted by Istr(e) and $Ostr(\pi)$ is defined as follows, respectively.

$$Istr(\pi) = Ialp(e_1)Ialp(e_2)\cdots Ialp(e_n).$$
$$Ostr(\pi) = Oalp(e_1)Oalp(e_2)\cdots Oalp(e_n).$$
(2)

Therefore, the input-output string of e, denoted by str(e), defined by

$$tr(\pi) = alp(e_1)alp(e_2)\cdots alp(e_n) = (Ialp(e_1)Ialp(e_2)\cdots Ialp(e_n), Oalp(e_1)Oalp(e_2)\cdots Oalp(e_n)) = (Istr(\pi), Ostr(\pi)).$$

Let $str(p(\mathcal{T})) = \{str(\pi) | \pi \in p(\mathcal{T})\}.$ $Ostr(p(\mathcal{T})) = \{Ostr(\pi) | \pi \in p(\mathcal{T})\}.$ $Istr(p(\mathcal{T})) = \{Istr(\pi) | \pi \in p(\mathcal{T})\}.$ $\forall (s_1, v_1), (s_2, v_2) \in \Sigma^* \times \Omega^*,$

$$(s_1, v_1) = (s_2, v_2) \iff s_1 = s_2 \text{ and } v_1 = v_2.$$

Hence, $\forall \pi_1, \pi_2 \in p(\mathcal{T})$,

s

$$str(\pi_1) = str(\pi_2) \iff Istr(\pi_1) = Istr(\pi_2)$$
 and
 $Ostr(\pi_1) = Ostr(\pi_2).$

Based the above notations, we will give the related definitions of P-WT.

Definition 4: Let $\mathcal{T} = (Q, \Sigma, \Omega, \delta, I, F)$ be a P - WT. The P-valued input-output-function of $\mathcal{T}, f_{\mathcal{T}} : \Sigma^* \times \Omega^* \longrightarrow P$, defined by $\forall (s, v) \in \Sigma^* \times \Omega^*$,

$$f_{\mathcal{T}}(s,v) = \begin{cases} a, & \text{if } \exists \pi \in p(\mathcal{T}) \text{ such that } str(\pi) = (s,v), \\ 0, & \text{otherwise.} \end{cases}$$
(3)

Where, $a = \sum_{\pi \in p(\mathcal{T}), str(\pi) = (s,v)} [I(o(\pi)) \bullet w(\pi) \bullet F(d(\pi))].$ It is noted that, for a *P*-NFA $\mathcal{A} = (Q, \Sigma, \delta, I, F)$, we may

It is noted that, for a P-NFA $\mathcal{A} = (Q, \Sigma, \delta, I, F)$, we may use the notations similar to those of P-WT. And $\mathcal{R}_{\mathcal{A}} \in P^{\Sigma^*}$ are written equivalently as

$$\mathcal{R}_{\mathcal{A}}(s) = \begin{cases} b, & \text{if } \exists \pi \in p(\mathcal{T}) \text{ such that } str(\pi) = s, \\ 0, & \text{otherwise.} \end{cases}$$
(4)

Where, $b = \sum_{\pi \in p(\mathcal{T}), str(\pi) = s} [I(o(\pi)) \bullet w(\pi) \bullet F(d(\pi))].$

IV. REALIZATION OF INPUT-FUNCTIONS AND OUTPUT-FUNCTIONS

Next, we give the definition of input-functions and outputfunctions induced by input-output-functions.

Definition 5: Let $\mathcal{T} = (Q, \Sigma, \Omega, \delta, I, F)$ be a P-WT. The P-valued input-output-function of $\mathcal{T}, f_{\mathcal{T}} : \Sigma^* \times \Omega^* \longrightarrow P$, induced the following two functions

(1) The output-function: $f_o: \Sigma^* \longrightarrow P\langle \langle \Omega^* \rangle \rangle$ is defined by $\forall s \in \Sigma^*$,

$$f_o(s) = \sum_{v \in \Omega^*} f_{\mathcal{T}}(s, v) v.$$

 $f_o(s)$ are viewed as the all outputs with input string s.

(2) The input-function: $f_i : \Omega^* \longrightarrow P\langle \langle \Sigma^* \rangle \rangle$ is defined by $\forall v \in \Omega^*$,

$$f_i(v) = \sum_{s \in \Sigma^*} f_{\mathcal{T}}(s, v)s.$$

 $f_i(v)$ are viewed as the all inputs with output string v.

Since the terms with coefficients equal 0 can be omit. $f_o(s)$ and $f_i(v)$ can be simplified the following forms, respectively

$$f_o(s) = \sum_{v \in Ostr(P(\mathcal{T}))} f_{\mathcal{T}}(s, v)v \tag{5}$$

$$f_i(v) = \sum_{s \in Istr(P(\mathcal{T}))} f_{\mathcal{T}}(s, v)s \tag{6}$$

These simplification forms are help for the realization of input-functions and output-functions. The realization of input-functions (output-functions) can be solved through using the relevant weighted automata over strong bimonoids. Firstly, we consider the realization of output-functions.

Theorem 1: Let $\mathcal{T} = (Q, \Sigma, \Omega, \delta, I, F)$ be a P-WT. There exists a weighted finite automaton over a strong bimonoid $(P\langle \langle \Omega^* \rangle \rangle, +, \odot, \mathbf{0}, \varepsilon) \mathcal{O}$, such that $\mathcal{R}_{\mathcal{O}} = f_o$.

Proof : Let $\mathcal{T} = (Q, \Sigma, \Omega, \delta, I, F)$ be a *P*-WT. We construct a weighted finite automaton over a strong bimonoid $(P\langle \langle \Omega^* \rangle \rangle, +, \odot, \mathbf{0}, \varepsilon), \mathcal{O} = (Q, \Sigma, \delta_{\mathcal{O}}, I_{\mathcal{O}}, F_{\mathcal{O}}), \text{ as follows}$

$$I_{\mathcal{O}}(q) = I(q)\varepsilon, F_{\mathcal{O}}(q) = F(q)\varepsilon, \text{ for any } q \in Q,$$

$$\delta_{\mathcal{O}} : Q \times \Sigma \times Q \longrightarrow P\langle \Omega_{\varepsilon} \rangle, \forall p, q \in Q, x \in \Sigma_{\varepsilon},$$

$$\delta_{\mathcal{O}}(p, x, q) = \sum_{(p, x, y, q) \in E_{\mathcal{T}}} \delta(p, x, y, q) y.$$

We will shown that $\mathcal{R}_{\mathcal{O}} = f_o$ by the following steps. (1) By the definition of δ_o , we can give a surjection from $E_{\mathcal{T}}$ to $E_{\mathcal{O}}$, i.e. $\varphi: E_{\mathcal{T}} \longrightarrow E_{\mathcal{O}}, \forall e \in E_{\mathcal{T}}$

$$\varphi(e) = (c(e), Ialp(e), s(e)) \in E_{\mathcal{O}}.$$

 $\forall e_{\mathcal{O}} \in E_{\mathcal{O}}, \text{ let } \varphi^{-1}(e_{\mathcal{O}}) = \{e \in E_{\mathcal{T}} \mid \varphi(e) = e_{\mathcal{O}}\}, \text{ then}$

$$\varphi(E_{\mathcal{T}}) = \bigcup_{e \in E_{\mathcal{T}}} \varphi(e), \ \varphi^{-1}(E_{\mathcal{O}}) = \bigcup_{e_{\mathcal{O}} \in E_{\mathcal{O}}} \varphi^{-1}(e).$$

We can verify that

(1-1)
$$\forall e_{\mathcal{O}} \in E_{\mathcal{O}}, \forall e \in \varphi^{-1}(e_{\mathcal{O}}),$$

$$c(e_{\mathcal{O}}) = c(e), s(e_{\mathcal{O}}) = s(e), alp(e_{\mathcal{O}}) = Ialp(e),$$

$$\delta_{\mathcal{O}}(e_{\mathcal{O}}) = \sum_{e \in \varphi^{-1}(e_{\mathcal{O}})} \delta(e)Oalp(e) \in P\langle \Omega_{\varepsilon} \rangle.$$

(1-2) $\varphi(E_{\mathcal{T}}) = E_{\mathcal{O}}, \ \varphi^{-1}(E_{\mathcal{O}}) = E_{\mathcal{T}}.$

(2) $\varphi: E_{\mathcal{T}} \longrightarrow E_{\mathcal{O}}$ may induce a homomorphic mapping from $p(\mathcal{O})$ to $p(\mathcal{T})$ denoted also by φ . That is $\varphi: p(\mathcal{T}) \longrightarrow$ $p(\mathcal{O}), \forall \pi = e_1 e_2 \cdots e_n \in p(\mathcal{T}),$

$$\varphi(\pi) = \varphi(e_1 e_2 \cdots e_n) = \varphi(e_1)\varphi(e_2) \cdots \varphi(e_n).$$

 $\forall \pi_{\mathcal{O}} = e_{\mathcal{O}1} e_{\mathcal{O}2} \cdots e_{\mathcal{O}n} \in p(\mathcal{O}), \ \varphi^{-1}(\pi_{\mathcal{O}}) \text{ is defined as}$

$$\varphi^{-1}(\pi_{\mathcal{O}}) = \varphi^{-1}(e_{\mathcal{O}1}e_{\mathcal{O}2}\cdots e_{\mathcal{O}n})$$

= $\varphi^{-1}(e_{\mathcal{O}1})\varphi^{-1}(e_{\mathcal{O}2})\cdots\varphi^{-1}(e_{\mathcal{O}n})$
= $\{e_1e_2\cdots e_n \mid e_i \in \varphi^{-1}(e_{\mathcal{O}i}), i=1,\cdots,n\}$
= $\{\pi \mid \varphi(\pi) = \pi_{\mathcal{O}}\}$
 $\varphi(p(\mathcal{T})) = \cup_{\pi \in p(\mathcal{T})}\varphi(\pi)$

$$\varphi^{-1}(p(\mathcal{O})) = \bigcup_{\pi_{\mathcal{O}} \in p(\mathcal{O})} \varphi^{-1}(\pi_{\mathcal{O}}).$$

And we also can verify that (2-1) $\forall \pi_{\mathcal{O}} \in n(\mathcal{O}) \ \pi \in (\mathcal{O}^{-1}(\pi_{\mathcal{O}}))$

$$o(\pi_{\mathcal{O}}) = o(\pi), d(\pi_{\mathcal{O}}) = d(\pi), str(\pi_{\mathcal{O}}) = Istr(\pi).$$

$$(2-2) \ \varphi(p(\mathcal{T})) = p(\mathcal{O}), \ \varphi^{-1}(p(\mathcal{O})) = p(\mathcal{T}).$$

$$(2-3) \ \forall \pi_{\mathcal{O}} \in p(\mathcal{O}), \text{ Let } \pi_{\mathcal{O}} = e_{\mathcal{O}1}e_{\mathcal{O}2}\cdots e_{\mathcal{O}n}, \text{ then}$$

$$\begin{split} w_{\mathcal{O}}(\pi_{\mathcal{O}}) &= \delta_{\mathcal{O}}(e_{\mathcal{O}1}) \odot \delta_{\mathcal{O}}(e_{\mathcal{O}2}) \odot \cdots \odot \delta_{\mathcal{O}}(e_{\mathcal{O}n}) \\ &= \sum_{e_1 \in \varphi^{-1}(e_{\mathcal{O}1})} \delta(e_1) Oalp(e_1) \odot \\ &\sum_{e_2 \in \varphi^{-1}(e_{\mathcal{O}2})} \delta(e_2) Oalp(e_2) \odot \cdots \\ &\odot \sum_{e_n \in \varphi^{-1}(e_{\mathcal{O}n})} \delta(e_n) Oalp(e_n) \\ &= \sum_{e_1 \in \varphi^{-1}(e_{\mathcal{O}n})} \sum_{e_2 \in \varphi^{-1}(e_{\mathcal{O}2})} \cdots \\ &\sum_{e_n \in \varphi^{-1}(e_{\mathcal{O}n})} (\delta(e_1) \bullet \delta(e_2) \bullet \cdots \bullet \delta(e_n)) \\ &(Oalp(e_1) Oalp(e_2) \cdots Oalp(e_n)) \\ &(\text{by proposition 1}) \\ &= \sum_{e_1 \in \varphi^{-1}(e_{\mathcal{O}1}), e_2 \in \varphi^{-1}(e_{\mathcal{O}2}), \cdots, e_n \in \varphi^{-1}(e_{\mathcal{O}n}) \\ &(w(e_1 e_2 \cdots e_n) Ostr(e_1 e_2 \cdots e_n)) \\ &= \sum_{\pi \in \varphi^{-1}(\pi_{\mathcal{O}})} w(\pi) Ostr(\pi) \in P\langle \Omega^* \rangle \end{split}$$

(3) Finally, we shown that $\mathcal{R}_{\mathcal{O}} = f_o$. In fact, $\forall s \in \Sigma^*$,

$$\begin{aligned} \mathcal{R}_{\mathcal{O}}(s) &= \sum_{\pi_{\mathcal{O}} \in p(\mathcal{O}), str(\pi_{\mathcal{O}}) = s} [I_{\mathcal{O}}(o(\pi_{\mathcal{O}})) \odot w_{\mathcal{O}}(\pi_{\mathcal{O}}) \odot \\ F(d(\pi_{\mathcal{O}}))] \\ &= \sum_{\pi_{\mathcal{O}} \in p(\mathcal{O}), str(\pi_{\mathcal{O}}) = s} [I(o(\pi_{\mathcal{O}})\varepsilon) \odot \\ (\sum_{\pi \in \varphi^{-1}(\pi_{\mathcal{O}})} w(\pi) Ostr(\pi)) \odot F(d(\pi_{\mathcal{O}}))\varepsilon] \\ &\quad (by (2-1)) \\ &= \sum_{\pi_{\mathcal{O}} \in p(\mathcal{O}), str(\pi_{\mathcal{O}}) = s} [\sum_{\pi \in \varphi^{-1}(\pi_{\mathcal{O}})} (I(o(\pi)) \bullet \\ w(\pi) \bullet F(d(\pi))) Ostr(\pi)] \\ &\quad (by (2-3) \text{ and proposition } 1) \\ &= \sum_{\pi \in \bigcup_{\pi_{\mathcal{O}} \in p(\mathcal{O})} \varphi^{-1}(\pi_{\mathcal{O}}), Istr(\pi) = str(\pi_{\mathcal{O}}) = s} \\ (I(o(\pi)) \bullet w(\pi) \bullet F(d(\pi))) Ostr(\pi) \\ &\quad (by (2-1)) \\ &= \sum_{\pi \in \varphi^{-1}(p(\mathcal{O})), Istr(\pi) = s} (I(o(\pi)) \bullet w(\pi) \bullet \\ F(d(\pi))) Ostr(\pi) \\ &= \sum_{\pi \in p(\mathcal{T}), Istr(\pi) = s} (I(o(\pi)) \bullet w(\pi) \bullet \\ F(d(\pi))) Ostr(\pi) (by (2-2)) \\ &= \sum_{v \in Ostr(p(\mathcal{T}))} (\sum_{\pi \in p(\mathcal{T}), str(\pi) = (s,v)} I(o(\pi)) \bullet \\ w(\pi) \bullet F(d(\pi))) v (by proposition 2) \\ &= \sum_{v \in Ostr(p(\mathcal{T}))} f_{\mathcal{T}}(s, v) v = f_o(s). \end{aligned}$$

The following theorem will give the realization of inputfunctions.

Theorem 2: Let $\mathcal{T} = (Q, \Sigma, \Omega, \delta, I, F)$ be a P-WT. There exists a weighted finite automaton over a strong bimonoid $(P\langle \langle \Sigma^* \rangle \rangle, +, \odot, \mathbf{0}, \varepsilon) \mathcal{I}$, such that $\mathcal{R}_{\mathcal{I}} = f_i$.

proof: Let $\mathcal{T} = (Q, \Sigma, \Omega, \delta, I, F)$ be a P-WT. We construct a weighted finite automaton over a strong bimonoid $(P\langle\langle \Sigma^* \rangle\rangle, +, \odot, \mathbf{0}, \varepsilon), \mathcal{I} = (Q, \Sigma, \delta_{\mathcal{I}}, I_{\mathcal{I}}, F_{\mathcal{I}}), \text{ as follows}$

$$\begin{split} I_{\mathcal{I}}(q) &= I(q)\varepsilon, F_{\mathcal{I}}(q) = F(q)\varepsilon, \text{for any } q \in Q, \\ \delta_{\mathcal{I}} : Q \times \Omega \times Q \longrightarrow P\langle \Sigma_{\varepsilon} \rangle, \, \forall p, q \in Q, y \in \Omega_{\varepsilon}, \\ \delta_{\mathcal{I}}(p, y, q) &= \sum_{(p, x, y, q) \in E_{\mathcal{T}}} \delta(p, x, y, q) x. \end{split}$$

 \mathcal{R}_{0}

Then, it is similar to theorem 1 to shown that $\mathcal{R}_{\mathcal{I}} = f_i$.

Example 1: Let $\mathbf{R}^{\infty}_{+} = \{a \in \mathbf{R} | a \ge 0\} \cup \{\infty\}$, then $P = (\mathbf{R}^{\infty}_{+}, \wedge, +, \infty, 0)$ a is trong bimonoid with $\infty \wedge a = a \wedge \infty = a, \infty \wedge \infty = \infty$ and $a + \infty = \infty + a = \infty$. It is noted that P is not distributive. Give a P-TM $\mathcal{T} = (Q, \Sigma, \Omega, \delta, I, F)$, where

$$\begin{split} Q &= \{q_0, q_1, q_2, q_3\}, \Sigma = \{A, B\}, \Omega = \{a_1, a_2, b\} \\ I &= \frac{q_0}{0}, F = \frac{q_2}{0} + \frac{q_3}{0} \\ \delta(q_0, A, a_1, q_1) &= 2, \delta(q_0, A, a_2, q_1) = 3, \\ \delta(q_1, A, a_1, q_2) &= 1, \delta(q_1, A, a_1, q_3) = 2, \\ \delta(q_1, A, b, q_3) &= 1, \delta(q_1, B, b, q_3) = 2, \end{split}$$

and $\delta = \infty$ for the rest. We have

$$f_{\mathcal{T}} = 3(AA, a_1a_1) + 3(AA, a_1b) + 4(AB, a_1b) + 4(AA, a_2a_1) + 4(AA, a_2b) + 4(AB, a_2b).$$

and then

$$f_o = (3a_1a_1 + 3a_1b + 4a_2a_1 + 4a_2b)AA + (4a_1b + 5a_2b)AB$$

$$f_i = (3AA)a_1a_1 + (4AA)a_2a_1 + (3AA + 4AB)a_1b + (4AA + 5AB)a_2b$$

We construct a weighted finite automaton over a strong bimonoid $(P\langle\langle \Omega^* \rangle\rangle, +, \odot, \mathbf{0}, 0\varepsilon), \mathcal{O} = (Q, \Sigma, \delta_{\mathcal{O}}, I_{\mathcal{O}}, F_{\mathcal{O}}),$ where

$$I_{\mathcal{O}} = \frac{q_0}{0\varepsilon}, F_{\mathcal{O}} = \frac{q_2}{0\varepsilon} + \frac{q_3}{0\varepsilon}$$

$$\delta_{\mathcal{O}}(q_0, A, q_1) = 2a_1 + 3a_2,$$

$$\delta_{\mathcal{O}}(q_1, A, q_2) = 1a_1,$$

$$\delta_{\mathcal{O}}(q_1, A, q_3) = 2a_1 + 1b,$$

$$\delta_{\mathcal{O}}(q_1, B, q_3) = 2b,$$

and $\delta_{\mathcal{O}} = \mathbf{0}$ for the rest. we can compute that $\mathcal{R}_{\mathcal{O}}$ as follows.

$$\begin{aligned} \mathcal{R}_{\mathcal{O}}(AA) &= & \delta_{\mathcal{O}}(q_0, A, q_1) \odot \delta_{\mathcal{O}}(q_1, A, q_2) + \\ & & \delta_{\mathcal{O}}(q_0, A, q_1 \odot \delta_{\mathcal{O}}(q_1, A, q_3)) \\ &= & (2a_1 + 3a_2) \odot 1a_1 + (2a_1 + 3a_2) \odot 2b \\ &= & (2 + 1)a_1a_1 + (3 + 1)a_2a_1 \\ & & + (2 + 2)a_1b + (3 + 2)a_2b \\ &= & 3a_1a_1 + 3a_1b + 4a_2a_1 + 4a_2b \end{aligned}$$

$$\mathcal{R}_{\mathcal{O}}(AB) = \delta_{\mathcal{O}}(q_0, A, q_1) \odot \delta_{\mathcal{O}}(q_1, B, q_3) = (2a_1 + 3a_2) \odot 2b = (2 + 2)a_1b + (3 + 2)a_2b = 4a_1b + 5a_2b$$

and $\mathcal{R}_{\mathcal{O}} = \mathbf{0}$ for the rest. That is $\mathcal{R}_{\mathcal{O}} = f_o$.

Similarly, We construct a weighted finite automata over a strong bimonoid $(P\langle\langle \Sigma^*\rangle\rangle, +, \odot, \mathbf{0}, 0\varepsilon), \mathcal{I} = (Q, \Sigma, \delta_{\mathcal{I}}, I, F)$, where

$$\begin{split} &\delta_{\mathcal{I}}(q_0, a_1, q_1) = 2A, \\ &\delta_{\mathcal{I}}(q_0, a_2, q_1) = 3A, \\ &\delta_{\mathcal{I}}(q_1, a_1, q_2) = 1A, \\ &\delta_{\mathcal{I}}(q_1, a_1, q_3) = 2A, \\ &\delta_{\mathcal{I}}(q_1, b, q_3) = 1A + 2B, \end{split}$$

and $\delta_{\mathcal{I}} = \mathbf{0}$ for the rest. we can compute that $\mathcal{R}_{\mathcal{I}} = f_i$.

V. CONCLUSIONS

In this work, we introduced weighted transducers over strong bimonoids and solved some realizations problems related to them. For a strong bimonoid P, we proposed the notions of the input-function f_i and output-function f_o induced by the input-output-function of P-WT. Then we constructed a finite weighted automaton \mathcal{I} over strong bimonoid $(P\langle\langle \Sigma^* \rangle\rangle, +, \odot, \mathbf{0}, \varepsilon)$ such that $\mathcal{R}_{\mathcal{I}} = f_i$ and a finite weighted automaton \mathcal{O} over strong bimonoid $(P\langle\langle \Omega^* \rangle\rangle, +, \odot, \mathbf{0}, \varepsilon)$ such that $\mathcal{R}_{\mathcal{O}} = f_o$. By the realizations of input-function f_i and output-function f_o . Next, the further research on the applications of P-WT in uncertain data management will be given.

REFERENCES

- M. Čirić, M. Droste, J. Ignjatović, H. Vogler, Determinization of weighted finite automata over srong bimonoids, *Information sciences*, 180(2010) 3497-3520.
- [2] M. Droste, T. Stüber, H. Vogler, Weighted finite automata over strong bimonoids, *Information Sciences*, 180(2010)156-166.
- [3] M.S. Ying. A theory of computation based on quantum logic (I), *Theoretical Computer Science*, 344(2005) 134-207.
- [4] Yun Shang, Xian Lu, Ruqian Lu, A theory of computation based on unsharp quantum logic: Finite automata ang pushdoow automata *Theoretical Computer Science*, 434(2012) 53-86.
- [5] M. Droste, W. Kuich, H. Vogler(eds), Handbook of Weighted Automata, *EATCS Monographs in Theoretical Computer Sciences*, Springger-Verlag, Birlin, 2009.
- [6] D.W. Qiu. Automata theory based on completed residuated latticevalued logic (I), *Science in China Serial F*, 44(6)(2001) 419-429.
- [7] Y.M. Li, W. Pedrycz. Fuzzy finite automata and fuzzy regular expressions withe membership values in lattice-ordered monoids, *Fuzzy sets* and systems, 156(2005) 68-92.
- [8] Y.M. Li, Lattice-valued fuzzy Turinng machines: Computing power, universality and efficiency, *Fuzzy sets and systems*, 160(2009)3453-3474.
- [9] Z. H. Li, P. Li, Y. M. Li, The relationships among several types of fuzzy automata, *Information Sciences*, 176(2006)2208-2226.
- [10] Ping Li, Yongming Li. Algeberaic Properties of LA-languages, *Infor*mation Sciences, 176(2006) 3232-3255.
- [11] M. Mohri. Finite-state transducers in language and speech processing, *Computational Linguistics*, 23(1997)269-311.
- [12] M. Mohri. Minimization algorithms for sequential transducers, *Theoretical Computer Science*, 234(2000)177-201.
- [13] M. Mohri. Statisstical natural language processsing. In M. Lothaire, editor, Applied Combinatorics on Words, *Cambridge University Press*, Cambridge, 2005.
- [14] M. Mohri, F.C. N. Pereira, and M. Riley. Weighted automata in text and speech processing. In Proceedings of the 12th Biennial European Conference on Artificial Intelligence (ECAI-96), Workshop on Extended Finite State Models of Language, Budapest, Hungary, 1996. Wiley, Chichester, 1996.
- [15] M. Mohri, F. Pereira, M. Riley. The design principles of weighted finite-state transducer library, *Theoretical Computer Science*, 231(2000)17-32.