On Level-Compatible Algebra Functions and Their Transformers

ZHUHAN JIANG
School of Computing and Information Technology
University of Western Sydney, Parramatta Campus, Sydney
AUSTRALIA
zhuhan@cit.uws.edu.au

Abstract: - We develop the mathematical theory of algebra functions on A^* for a finite alphabet A, such that the functions are level-compatible when they are restricted to the different levels m according to A^m. We shall first establish the fundamentals and the structure of such function spaces and the related properties, and propose and study the all important level-preserving transformers there. We shall then look into the the automaton representations of the level-compatible algebra functions, the characterisation of the level-preserving transformers, as well as the potential applications at calculation of real-valued functions, fractal generations and even the compression of natural images.

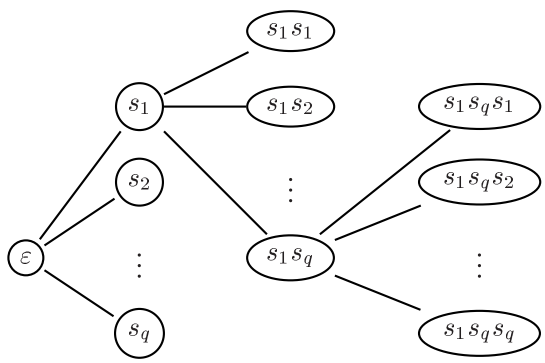
Keywords: - alphabet, algebra, mapping and transformer, level preservation, automaton representation.

1. Introduction

Well establisehd function spaces and their inner structures and properties are often invaluable and even instrumental in the theory and applications that shrive on such spaces. From the classical space of continuously differentiable functions $C^m[a, b]$ to the metric space of $L^p[a, b]$, and to the generalised functions of distributions, a rich collection of functions spaces have been created and studied in thorough details to assist the respective applications in both mathematics and elsewhere. Enormous advances in computer science and technology in recent decades have also substantially extended the landscape of the traditional mathematics, and have subsequently cultivated a steady stream of new methodologies and tools, including various ad-hoc algorithms, computability and automata if we name a few. Our purpose here is therefore to follow up this expansion and look in great details into the structure and theory deriving from the space $\mathfrak{F}$ of algebra functions of $A^* \mapsto \mathbb{R}$, where A is a finite alphabet and $\mathbb{R}$ is the set of all real numbers. There could be abundant use of such functions. For instance, a continuous real-valued function $F : [0, 1] \mapsto \mathbb{R}$ could be represented [1] by an $f : A^* \mapsto \mathbb{R}$ if one identifies a string $\omega = a_1 \cdots a_m$ with the value $\sum_{i=1}^{m} \chi(a_i)q^{-i}$, where $q = |A|$ is the number of symbols in A, and $\chi(a)$ is the 0-based index of the symobl a in A. As another example, any string $\omega \in A^*$ can be identified with the address of the quadrant subimages, see for example [2–5] for some of the existing applications related to images and videos. In this connection, images can be represented naturally by such algebra functions. For easy distinction and the fact that A^* is algebraic and somewhat of a tree type, we will often refer such functions of type $A^* \mapsto \mathbb{R}$ as *algebra functions* or *tree functions*. The term *tree* is actually a reflection of the fact that A^* can be enumerated in a tree structure, see Fig. 1, meaningful especially in the context [5] of treating A^* as the addresses of quadrant subimages. Since such application context is often more involved with the so-called level-compatible subspace $\mathbf{F}$ of the algebra function $\mathfrak{F}$, a sizable attention will be directed to explore finer details and properties there.

The study of the space $\mathbf{F}$ of algebra functions then naturally lead us to the study of the more intrinsic linear mappings on $\mathbf{F}$. It turns out that the mappings that essentially map all $f \in \mathbf{F}$ level by level, thus termed level-preserving transformers, are most interesting and useful. As we will show that many algebra functions may be represented and reconstructed from weighted finite automata, a rich collection of such transformers along with their complete characterisation will enable one to reduce the number of the automaton states so as to achieve the desired application objective of having as smaller an automaton as possible. A smaller automaton, for example, will in general induce a better compression ratio if a weighted finite automaton is used to represent an original image, see e.g. [5]. Mathematically, this paper has also solved a thorny issue that was conspicuously absent in all the automata based compression schemes, that is, what transformers can generate legitimate

Figure 1. Tree Structure of A^*



new images to enlarge the range image pool, and how to construct them systematically. We hasten to add that our transformers bear no connections with the so-called transducers [6] that are often encountered in a similar context. Another obvious application of our level-preserving transformers is that it may enable one to easily generate rich fractal patterns. Incidentally, we have chosen to use the terms *level-compatible* and *level-preserving* instead of the original *area-preserving* [2] tied up more closely with the applications. This is largely because the concept of *area* is removed from our abstract function space and our new terms may be more indicative in the present structure.

This paper is thus organised as follows. First in section 2, we introduce the fundamentals of the level-compatible algebra functions on A^*, including the corresponding linear space **F** of such functions. We then propose and study the level-preserving transformers, and the structure and the spaces of such transformers. The fundamental results of complete characterisation of the level-preserving transformers are then presented in section 3 in terms of the explicit construction of all such transformers. The geometric interpretations of such construction are also provided via multiresolutional images for their intuitiveness. Section 4 is then allocated to the study of the representation and generation of the algebra functions through the use of weighted finite automata. We illustrate there how and what level-preserving transformers can be utilised to reduce the number of the states for the corresponding automata. Finally in section 5, the applicability of the theory established in the earlier sections is discussed in terms of function evaluation, fractal generation as well as image compression.

2. Algebra Functions and Level-Preserving Transformers

We first set up the necessary basic notations. Let $A = \{s_1, s_2, ..., s_q\}$ be the alphabet, s_i there be the symbols and $|A| = q$ be the cardinality of the set A. As is the convention, we take $A^0 = \{\varepsilon\}$, where ε denotes the empty string, and take $A^m = \{a_1 \cdots a_m \,|\, a_i \in A\}$ and $A^{(m)} = \bigcup_{i=0}^{m} A^i$ for any $m \in \mathbb{Z}^+$, the set of non-negative integers, and set $A^* = \bigcup_{m=0}^{\infty} A^m$. Any string $\omega \in A^*$ thus belongs to A^m for a unique non-negative integer m, and that m, denoted by $|\omega|$, will be termed the *length* of ω. With $\mathbb{R}$ being the set of all real numbers, an algebra function $f : A^* \mapsto \mathbb{R}$ is said to be *level-compatible* if it satisfies the following *level-invariance* condition

$$\sum_{a\in A} f(\omega a) = f(\omega)\cdot|A|, \quad \forall\omega \in A^* . \tag{1}$$

The essence of the level-invariance is that it ensures a truthful multiscale representation in the sense that the restriction $f_{|_{A^{(m)}}} \mapsto \mathbb{R}$ of f to the "scale" $A^{(m)}$ gives an increasingly better details of function f's global behaviour as m grows larger.

Let $\mathfrak{F}$ be the set of all algebra functions $A^* \mapsto \mathbb{R}$, and **F**$=\{f \in \mathfrak{F} \,|\, f$ is level-compatible$\}$. We endow a natural addition "$+$" on $\mathfrak{F}$ by

$$(\alpha f + \beta g)(\omega) = \alpha f(\omega) + \beta g(\omega),$$

$$\forall a, b \in \mathbb{R},\ f, g \in \mathfrak{F},\ \omega \in A^*.$$

Then both $\mathfrak{F}$ and **F** are linear spaces over $\mathbb{R}$. This is because the level-invariance is well preserved under the linear addition. For any $f \in$ **F**, we say f is of *finite precision* N, or is just *finite*, if $f(\omega\omega') = f(\omega)$ holds for all $\omega \in A^N$ and all $\omega' \in A^*$.

On the linear space **F**, we now denote by $\mathfrak{L}$ the space of linear mappings from **F** to itself. In fact we will make $\mathfrak{L}$ an algebra by endowing the composition product "$\circ$" by

$$(\sigma \circ \tau)(f) = \sigma(\tau(f)), \quad \forall\sigma, \tau \in \mathfrak{L},\ f \in \mathbf{F}$$

and defining the *zero* **0** and *one* **1** in $\mathfrak{L}$ by $\mathbf{0}(f) = 0$ and $\mathbf{1}(f) = f$ for all $f \in$ **F**. In what follows we shall thus present the basic structure and properties of the space $\mathfrak{L}$, before moving into further in-depth analysis in the next section.

First let us examine some of the simpler mappings in $\mathfrak{L}$. To start with, for any $\Omega \in A^*$, if we define $\hat{\Omega} : \mathbf{F} \mapsto \mathfrak{F}$ by $f^{\Omega}(\omega) \stackrel{\text{def}}{=} (\hat{\Omega} f)(\omega) = f(\Omega\omega)$ for all $\omega \in A^*$ and all $f \in$ **F**, then the identity

$$\begin{aligned}\sum_{a\in A} (\hat{\Omega} f)(\omega a) \quad &= \quad \sum_{a\in A} f(\Omega\omega a) \;=\; f(\Omega\omega)\cdot|A| \\ &= \quad (\hat{\Omega} f)(\omega)\cdot|A|\end{aligned}$$

implies $\hat{\Omega}(f)$ is also level-compatible and thus $\hat{\Omega} \in \mathfrak{L}$ for all $\Omega \in A^*$. If, for any $\Omega \in A^*$, we define $\widehat{\Omega^{-1}} : \mathbf{F} \mapsto \mathfrak{F}$ by

$$(\widehat{\Omega^{-1}}f)(\omega) = \begin{cases} f(\omega') & \text{if } \omega = \Omega\omega',\, \omega' \in A^* \\ 0 & \text{if } |\omega| \geq |\Omega|,\, \omega \neq \Omega\omega' \\ & \text{ for all } \omega' \in A^* \\ \text{average} & \text{if } |\omega| < |\Omega| \end{cases}$$

where the average can be simply calculated recursively through the use of the level-invariance property, then it is also easy to show $\widehat{\Omega^{-1}} \in \mathfrak{L}$. In terms of the address tree in Fig. 1, $\hat{\Omega}$ and $\widehat{\Omega^{-1}}$ amount to zoom-in a tree branch or zoom out of the whole tree. Since functions in the space $\mathbf{F}$ are level-oriented, mappings of level-preservation are often naturally desired. For this purpose, a mapping $\sigma \in \mathfrak{L}$ will be termed *level-preserving* if, for all $f \in \mathbf{F}$, the restriction of f to $A^{(m)}$ *completely* determines the restriction of $\sigma(f)$ to $A^{(m)}$, i.e. $f_{|_{A^{(m)}}}$ completely determines $\sigma(f)_{|_{A^{(m)}}}$. Because the level-invariance of the functions in $\mathbf{F}$, a $\sigma \in \mathfrak{L}$ is level-preserving if and only if $f_{|_{A^m}}$ completely determines $\sigma(f)_{|_{A^m}}$ for all $m \in \mathbb{Z}^+$. Let $\mathbf{L}$=$\{\sigma \in \mathfrak{L} \,|\, \sigma$ is level-preserving$\}$. Then the mappings in $\mathbf{L}$ will in general be referred to as *transformers*. If we simply identify $\hat{\Omega}$ with Ω and likewise, $\widehat{\Omega^{-1}}$ with Ω^{-1}, then we have the following

Proposition 1 *Both $\mathfrak{L}$ and $\mathbf{L}$ are linear spaces, and both are algebras. Moreover,*

(i) $A^* \subset \mathfrak{L}, \quad A^{-*} \subset \mathbf{L}$, *where* $A^{-*} = \{\omega^{-1} \,|\, \omega \in A^*\}$.

(ii) All mappings in A^{-*} are invertible transformers.

(iii) $\omega \circ \omega^{-1} = \mathbf{1}$ *holds for all* $\omega \in A^*$.

(iv) $\sum_{a \in A} a^{-1} \circ a = \mathbf{1}$.

PROOF: Most of the proof is already covered in the above, the rest is also easy to fill in. Q.E.D.

We note that the set of invertible transformers, $\mathbf{I}$=$\{\sigma \in \mathbf{L} \,|\, \sigma$ is invertible$\}$, is incidentally a very useful subset of $\mathbf{L}$: it is a group with respect to the composition product "$\circ$" although it is not a linear space. We also note that although our transformers are all defined to be linear because they come from $\mathfrak{L}$, it is possible to have a nonlinear level-preserving mapping from $\mathbf{F}$ to itself. For instance, if we take $A = \{a, b\}$ and define a δ by

$$\delta(\omega) = \begin{cases} 1 & \text{if } \omega = a\omega',\, \omega' \in A^* \\ 0 & \text{if } \omega = b\omega',\, \omega' \in A^* \\ 1/2 & \text{if } \omega = \varepsilon \end{cases}$$

then $a^{-1} + \delta$ is level-preserving and is nonlinear. Nonlinear level-preserving mappings on $\mathbf{F}$ tend to break the mathematical logic in potential applications, and will thus not be considered any further in the current work. We also note that most concepts in the earlier part of this section can be generalised to multidimensional cases. The counterparts for $\mathfrak{L}$ and $\mathbf{L}$ in multidimensions are, for instance,

$$\mathfrak{L}_m \stackrel{\text{def}}{=} \{\sigma : \mathbf{F}^m \mapsto \mathbf{F} \mid \sigma \text{ is linear}\}$$
$$\mathbf{L}_m \stackrel{\text{def}}{=} \{\sigma \in \mathfrak{L}_m \mid \sigma \text{ is level-preserving}\}$$

where σ is level-preserving means for all $\mathbf{f}$=$(f_1, ..., f_m) \in \mathbf{F}^m$, the values in $\{\mathbf{f}(\omega) \,|\, \omega \in A^{(m)}\}$ completely determine those in $\{(\sigma\mathbf{f})(\omega) \,|\, \omega \in A^{(m)}\}$. Immediately, each element $\mathbf{t}$=$(t_1, ..., t_m) \in \mathbf{L}^m$ can be identified with an element in $\mathbf{L}_m$ via $\mathbf{F}^m \mapsto \mathbf{F} : (f_1, ..., f_m) \to \sum_{i=1}^m t_i(f_i)$. In other words, we can embed $\mathbf{L}^m$ into $\mathbf{L}_m$, or simply write $\mathbf{L}^m \subset \mathbf{L}_m$, which is often all that one may need to use in some typical applications.

3. Characterisation of Transformers

In order to understand better the structure of the level-preserving transformers in $\mathbf{L}$, we first show how we can construct an easy family of linear mappings in $\mathfrak{L}$. The main observation is that an algebra function $f \in \mathbf{F}$ can be just defined on the *level* A^m because $f_{|_{A^m}}$ uniquely and completely determines $f_{|_{A^{(m)}}}$ and can also naturally extend to $f_{|_{A^*}}$. This extension is done by the following *default completion process*

(i) For $k = m - 1$ to 0, construct $f_{|_{A^k}}$:
$f_{|_{A^k}}(\omega) = \sum_{a \in A} f_{|_{A^{k+1}}}(\omega a)/|A|, \quad \forall \omega \in A^k$.

(ii) Set $f(\omega) = f_{|_{A^m}}(\Omega)$ for all $\omega = \Omega\omega'$ with $\Omega \in A^m$ and $\omega' \in A^*$.

For any set B, we denote by S_B the permutation group of all the elements in B. Then every permuation $s \in S_{A^m}$ induces a linear mapping $\sigma \in \mathfrak{L}$ via

$$(\sigma f)\Big|_{A^m}(\omega) = f(s(\omega)), \quad \forall f \in \mathbf{F}, \omega \in A^m \ .$$

The above described default completion process thus completes the definition of $\sigma f = \sigma(f) \in \mathfrak{L}$. Although an $s \in S_{A^m}$ does not in general induce a level-preserving transformer, being simple and illustrative, the above does provide the insights into the structure of $\mathbf{L}$ to some extent.

To facilitate our complete characterisation of the structure of $\mathbf{L}$, we first introduce the following set of coefficient matrices

$$\mathfrak{C} = \left\{ (c_{ij})_{|A| \times |A|} \;\Big|\; \sum_{i=1}^{|A|} c_{ij} = 1 \text{ for } j = 1, ..., |A| \right\},$$

where (c_{ij}) is an $|A|$ by $|A|$ matrix whose columns all add up to the unit 1. Suppose $\sigma \in \mathbf{L}$ is an arbitrary transformer, then for all $f \in \mathbf{F}$ we have

$$(\sigma f)(s_i) = \sum_{j=1}^{|A|} c_{ij} f(s_j) + c_0$$

because $(\sigma f)|_{A^{(1)}}$ is linear, and is completely determined by f at level A^1. Obviously the inhomogeneous term c_0 must be 0 because $\sigma(f)$ also needs to be level-compatible. Hence

$$
\begin{aligned}
|A|(\sigma f)(\varepsilon) &= \sum_{i=1}^{|A|}(\sigma f)(s_i) = \sum_{i=1}^{|A|}\sum_{j=1}^{|A|} c_{ij} f(s_j) \\
&= \sum_{j=1}^{|A|}\Big[\sum_{i=1}^{|A|} c_{ij}\Big] f(s_j)
\end{aligned}
$$

and

$$
(\sigma f)(\varepsilon) = |A|\rho f(\varepsilon) = \rho \sum_{j=1}^{|A|} f(s_j) \quad \text{for some } \rho \in \mathbb{R}
$$

hold for all $f \in \mathbf{F}$ implies $\sum_{i=1}^{|A|} c_{ij} = \rho|A|$ for all $j = 1, ..., |A|$. In other words, there exists a $c^\varepsilon = (c^\varepsilon_{ij}) \in \mathfrak{C}$ such that

$$
(\sigma f)(s_i) = \rho \sum_{j=1}^{|A|} c^\varepsilon_{ij} f(s_j) \ .
$$

Because of the level-invariance of the functions in $\mathbf{F}$, the refinement at a higher level should be done in a localised fashion. That is, the contribution for $f(s_j)$ should come from the $f(s_j s_k)$'s. Hence we obtain

$$
(\sigma f)(s_i s_{i'}) = \rho \sum_{j=1}^{|A|}\sum_{j'=1}^{|A|} c^\varepsilon_{ij} c^{s_j}_{i'j'} f(s_j s_{j'}) \ .
$$

The fact that $\sigma f \in \mathbf{F}$ then implies $c^{s_j} \in \mathfrak{C}$. Inductively following this procedure, we can derive the following

Proposition 2 *For any $\sigma \in \mathbf{L}$, there exists a sequence $\{\rho \in \mathbb{R}, c^\omega \in \mathfrak{C}\}_{\omega\in A^*}$ such that $\forall f \in \mathbf{F}$ and $m \in \mathbb{Z}^+$, the restriction of $\sigma(f)$ on A^m is given by*

$$
(\sigma f)(a_{i_1} a_{i_2} \cdots a_{i_m}) = \rho \sum_{j_1, j_2, \cdots, j_m=1}^{|A|} c^\varepsilon_{i_1 j_1} c^{a_{j_1}}_{i_2 j_2} \cdots c^{a_{j_1}\cdots j_{m-1}}_{i_m j_m} f(a_{j_1} a_{j_2} \cdots a_{j_m}) \ . \tag{2}
$$

Moreover, the inverse of the transformer σ exists and is also level-preserving if and only if $\det(c^\omega) \neq 0$ for all $\omega \in A^*$.

PROOF: The level-preservation is obvious from the explicit expression (2). That is, the restriction of f to $A^{(m)}$ completely determines $\sigma(f)$ at $A^{(m)}$. We now need to verify that the $\sigma(f)$ defined by (2) is indeed level-compatible, i.e. $(\sigma f) \in \mathbf{F}$. However, the following identity

$$
\begin{aligned}
(\sigma f)(a_{i_1} \cdots a_{i_{m-1}}) &= \frac{1}{|A|} \sum_{a_{i_m} \in A} (\sigma f)(a_{i_1} \cdots a_{i_m}) \\
&= \rho \sum_{j_1, j_2, \cdots, j_m=1}^{|A|} c^\varepsilon_{i_1 j_1} c^{a_{j_1}}_{i_2 j_2} \cdots c^{a_{j_1}\cdots j_{m-2}}_{i_{m-1} j_{m-1}} \times \\
&\qquad \frac{1}{|A|} \sum_{i_m=1}^{|A|} c^{a_{j_1}\cdots j_{m-1}}_{i_m j_m} f(a_{j_1} a_{j_2} \cdots a_{j_m}) \\
&= \rho \sum_{j_1, j_2, \cdots, j_m=1}^{|A|} c^\varepsilon_{i_1 j_1} c^{a_{j_1}}_{i_2 j_2} \cdots c^{a_{j_1}\cdots j_{m-2}}_{i_{m-1} j_{m-1}} \times \\
&\qquad \frac{1}{|A|} \sum_{j_m=1}^{|A|} f(a_{j_1} a_{j_2} \cdots a_{j_m}) \\
&= \rho \sum_{j_1, j_2, \cdots, j_m=1}^{|A|} c^\varepsilon_{i_1 j_1} c^{a_{j_1}}_{i_2 j_2} \times \\
&\qquad c^{a_{j_1}\cdots j_{m-2}}_{i_{m-1} j_{m-1}} f(a_{j_1} a_{j_2} \cdots a_{j_{m-1}})
\end{aligned}
$$

shows that equation (2) for m implies exactly that for $m-1$ if $m \geq 1$. Hence the level-invariance of (σf) is guaranteed by the constructive defination (2). As for the inverse of σ, we observe that the nonsingularity of c^ω implies the existence of (d^ω_{ij}) such that $c^\omega d^\omega$ is an identity matrix. Hence we have

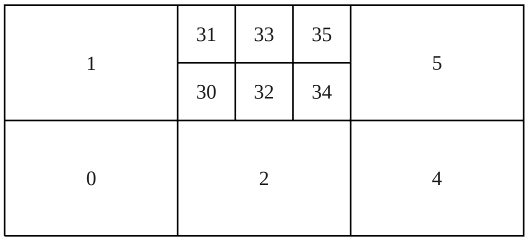
$$
f(a_{i_1} \cdots a_{i_m}) = \rho^{-1} \sum_{j_1, j_2, \cdots, j_m=1}^{|A|} d^\varepsilon_{i_1 j_1} d^{a_{j_1}}_{i_2 j_2} \cdots d^{a_{j_1}\cdots j_{m-1}}_{i_m j_m} (\sigma f)(a_{j_1} \cdots a_{j_m}) \ , \tag{3}
$$

and therefore the inverse is also level-compatible. If c^Ω is singular for some $\Omega \in A^*$, then it is possible to construct $f, g \in \mathbf{F}$ such that $f \neq g$, f and g are both of finite precision $|\Omega| + 1$, but $\sigma(f) = \sigma(g)$. Q.E.D.

An immediate corollary of the above proposition, due to (3) for $m = 1$, is that nonsingular matrices in $\mathfrak{C}$ are closed under the inverse operation. That is, for any $c \in \mathfrak{C}$, $\det(c) \neq 0$ implies $c^{-1} \in \mathfrak{C}$. This corollary can also be easily proved directly. In fact we can furthermore show that the set of all invertible matrices in $\mathfrak{C}$ comprise a *group* under the matrix multiplication.

In order to have a more intuitive understanding of the transformer σ in (2), we assume that we have chosen an alphabet A such that any rectangular mathematical image can be divided into $|A|$ equal-sized rectangular subimages, each addressed uniquely by a single symbol in the alphabet A. All such subimages can then be recursively partitioned further into even smaller subimages. This way, any image can be identified with a mapping $f \in \mathbf{F}$ in terms of the

Figure 2. A^* **as the Addresses**



following correspondence: $f(\varepsilon)$=intensity of the whole image, $f(\omega)$=intensity of the subimage addressed by $\omega \in A^*$. Any transformer $\sigma \in$ **L** can then be interpreted as a mapping that transforms a multiresolutional image to another multiresolutional image, such that the transformed image at any given level, A^m, is completely determined by that of the original image at the same level. For example, let $A = \{0, 1, 2, 3, 4, 5\}$ and the recursive partition of a rectangular image be exemplified in Fig. 2. In general, if an $f \in$ **F** represents an image, then for any $\omega = a_1 a_2 \cdots a_m \in A^*$ (treated as a member of $\mathfrak{L}$), function $f^{\omega} \stackrel{\text{def}}{=} \omega(f)$ belongs to **F** and represents the a_m-th quadrant of a_{m-1}-th quadrant of $\cdots$ of a_2-th quadrant of a_1-th quadrant of the original image $f \in$ **F**. Now we are ready to intuitively examine how to construct level-preserving transformers.

To achieve the level-preservation, we will construct the transformers level by level. At level A^1, the level-preservation can only be achieved by $(\sigma f)(s_i) = \sum_j \rho c^{\varepsilon}_{ij} f(s_j)$. Then we go to each of the quadrant transformed subimage, $(\sigma f)^{s_i}$, apply a level-preserving transform there to 1 level deeper. This transformer will be characterised by a $c^{s_i} \in \mathfrak{C}$, and the ρ in this case has to be 1 because of the required level-invariance. In other words, we apply ρc^{ε} to image f (just at the 1st level), then apply $c^a \in \mathfrak{C}$ to the transformed subimages addressed by a, and then, in general, apply $c^{\omega} \in \mathfrak{C}$ to the previously cumulatively-transformed resulting subimage addressed by ω. This, therefore, implies that a level-preserving transformer can be represented by a $\rho \in \mathbb{R}$ and a sequence of matrices $\{c^{\omega} \in \mathfrak{C}\}_{\omega \in A^*}$, which is basically the same result as Proposition 2.

There is a more interpretable subset of **L**, which contains some simpler and invertible transformers. These transformers can be represented by a sequence of permutations $\{s_{\omega} \in S_A\}_{\omega \in A^*}$ through the use of (2) in which $\rho = 1$ and $c^{\omega}_{ij} = 1$ if $j = s_{\omega}(i)$ and =0 if otherwise. These transformers are thus called *permutative transformers*. As an example, if we make use of the alphabet and the partition scheme illustrated in Fig. 2, then the permutative transformer $\{s_{\omega} = (0, 5)\}_{\omega \in A^*}$ represents simply an image reflection against the diagonal axis. In fact, permutative transformers will emcompass a wide range of transformers, including almost all intuitive transformers such as reflections, rotations, and block permuations.

Proposition 3 *Let $A = \{s_1, s_2, ..., s_q\}$ be the alphabet. For any transformer $\sigma \in$ **L**, there exist a matrix $c \in \mathfrak{C}$ and transformers $\tau_j \in$ **L** for $j = 1, ..., q$ such that*

$$s_i \circ \sigma = \sum_{j=1}^{q} c_{ij} \tau_j \circ s_j \, , \quad i = 1, ..., q \, . \tag{4}$$

*If σ is permutative, then there are permutative transformers $\tau_j \in$ **L** such that*

$$s_i \circ \sigma = \tau_j \circ s_{j_i} \, , \quad j = 1, ..., q \, . \tag{5}$$

That is, for any $a \in A$, there exists a symbol $b \in A$ and a permutative transformer τ such that $a \circ \sigma = \tau \circ b$.

Before we supply the proof, we note that the properties here, other than analysing the mathematical structure of the transformers, may serve an important role in the theory of representing an algebra function $f \in$ **F** by a weighted finite automata of *minimum* number of states.

PROOF: Since (2) can be written as

$$\begin{aligned}
&(\sigma f)^{s_i}(a_{i_2} \cdots a_{i_m}) \\
&= \sum_{j=1}^{|A|} c_{ij} \left[\rho \sum_{j_2, \cdots, j_m = 1}^{|A|} c^{a_j}_{i_2 j_2} \cdots c^{a_{j_1} \cdots j_{m-1}}_{i_m j_m} f^{s_j}(a_{j_2} \cdots a_{j_m}) \right] \\
&= \sum_{j=1}^{|A|} c_{ij} (\tau_j f^{s_j})(a_{i_2} \cdots a_{i_m}),
\end{aligned}$$

we see that (4) follows from the above immediately. If σ is permutative, then for each i, there is only one j such that $c_{ij} \neq 0$. In fact such c_{ij} must be 1 due to the permutative nature of σ. Hence (5) follows from (4). Q.E.D.

4 Automata Representations of Algebra Functions

Among all the functions in **F**, the ones of finite precision do play more important roles, particularly in the potential applications. This is because it is easy to introduce a metric **d** on the linear space **F** so that any $f \in$ **F** can be approximated under the metric **d** by a sequence $\{f_n\}$ of functions $f_n \in$ **F** of finite precision. One such metric can be defined by

$$\begin{aligned}
\|f\| &\stackrel{\text{def}}{=} \sum_{m=0}^{\infty} \frac{1}{2^m} \|f\|_m, \\
\|f\|_m &\stackrel{\text{def}}{=} \frac{1}{|A|^m} \sum_{\omega \in A^m} |f(\omega)|, \qquad \forall f \in \mathbf{F}
\end{aligned}$$

because the triangle inequality $\|f+g\| \leq \|f\|+\|g\|$ holds for all $f, g \in \mathbf{F}$. Moreover, $\|f-g\|_m = 0$ means f and g are identical at the level of A^m. Since $\|f\|_m = 0$ implies $\|f\|_k = 0$ for all $0 \leq k \leq m$ due to the level-invariance of f, the definition of $\|f\|$ well respects the level-consistency of the functions in $\mathbf{F}$. If f is to represent an image in an application, for instance, then $\|f-g\|_m = 0$ means f and g are identical at the resolution of A^m.

Proposition 4 *Any $f \in \mathbf{F}$ of finite precision can be represented by a weighted finite automaton.*

PROOF: Suppose f is of finite precision N. Then for $S \stackrel{\text{def}}{=} \{f^\omega \,|\, \omega \in A^{(N)}\}$ we have $\phi^a \in S$ for all $\phi \in S$ and $a \in A$. This means that if S is enumerated by $S = \{\phi_i\}_{i=1}^M$, then we have $\phi_i(\varepsilon) = \gamma_i$ and $\phi_i^a = \sum_j w_{ij}^a \phi_j$ for some real values γ_i and w_{ij}^a. If we set the M states, represented by the state functions ϕ_i respectively, assign the γ_i to the corresponding state as the initial distribution, and assign the value w_{ij}^a as the weight associated with the transition from state i to state j on the symbol $a \in A$, then the entity $(\gamma_i, w_{ij}^a : i, j = 1, .., M, a \in A)$ comprises a weighted finite automaton, and the values $f(\omega)$ for all $\omega \in A^*$ can be derived from running this automaton. More precisely, for any $\omega = a_1 \cdots a_m \in A^m$, the value $\phi(\omega)$ for any state function ϕ_i is $\gamma_i \times$the sum of the weight product of all the paths of ω starting from the state i. Since f is among the state functions, the proposition is thus proven. Q.E.D.

We note that the automaton constructed in the above proof is overly bloated. There are algorithms which can enable one to derive a much smaller automaton. Also this proposition is essentially already known, see e.g. [2], albeit in different forms. It is included here for completeness, and for the fact that the above bloated proof actually provides good insights into the next proposition. To this end, a broader version, utilising level-preserving transformers, is described in the following

Proposition 5 *Let $f \in \mathbf{F}$ be finite and $\mathbf{T} \subset \mathbf{L}$ is nonempty. Then the following inference procedure will construct a weighted finite automaton of near-minimum number of states.*

(i) initialise: queue $S \leftarrow \{f_1\}$, length of the queue $N \leftarrow 1$, current pointer in the queue $i \leftarrow 1$

(ii) for each $a \in A$, if $f_i^a = \sum_{j=1}^N \alpha_{j,\tau} \sum_{\tau \in \mathbf{T}} \tau f_k$, then set $w_{ij}^{a,\tau} = \alpha_{j,\tau}$; otherwise set $N \leftarrow N+1$ and append f_N to S.

The inferred weighted finite automaton thus reads $f_i^a = \sum_{\tau \in \mathbf{T}} \sum_{j=1}^N w_{ij}^{a,\tau} \, \tau f_j$, where N is the total number of the states.

PROOF: The fact that the derived automaton regenerates the original algebra function f is manifested in the description of the procedure itself. In the extreme case of $\mathbf{T} = \{\mathbf{1}\}$, one can show that the above generated automaton has indeed the minimum number of states. For more general $\mathbf{T}$, some additional conditions such as $\mathbf{T}$ be a group under the composition "$\circ$" may be needed to ensure theoretically that the automaton has the minimum number of states. For the universality at this stage, we hence choose the term *near-minimum* in the proposition. Q.E.D.

We note that the main role of introducing $\mathbf{T} \in \mathbf{L}$ in the construction of an automaton representation of a level-compatible algebra function is that it may significantly reduce the number of states in the resulting automaton. We also note that if $\mathbf{T}$ is a *group* of permutative transformers, then, with the help of (5), we can show that the near-minimum in Proposition 5 is in fact a true minimum.

5 Applications

A simple application of $\mathbf{F}$, or more precisely, $\mathfrak{F}$, is the representation of continuous real function $F : [0,1] \mapsto \mathbb{R}$ via the composition of an $f : A^* \mapsto \mathbb{R}$ and a $\Lambda : A^* \mapsto [0,1)$ by $\Lambda(\omega) = \sum_{i=1}^\infty a_i q^{-i}$ for $\omega = a_1 a_2 \cdots$ with $A = A_q \stackrel{\text{def}}{=} \{0, 1, ..., q-1\}$. Real valued functions may be calculated through the use of weighted finite automata corresponding to the algebra function f. For further pursuit in this connection, readers may consult [1] for the coverage related to the alphabet $A_2 = \{0, 1\}$ and to the line and level automata. We shall however not follow this direction here any further.

A more pertinent application is the generation of fractal images through the use of weighted finite automata and the level-preserving transformers. In the trivialest but illustrative cases in Fig. 3, we have two weighted finite automata on alphabet $A_4 = \{0, 1, 2, 3\}$ and alphabet $A_2 = \{0, 1\}$ respectively, both of which can regenerate the same image (the leftmost). The partition scheme on the 2nd automaton is somewhat involved because the bisection needs to take turn to be done horizontally and vertically. In fact, each image may be assigned a positive or negative orientation initially, the subimages obtained by a bisection will then be assigned the opposite orientation. A bisection will be done horizontally or vertically, depending on whether the image to be bisected is positively or negatively oriented. This adjustment is due to the directional imbalance caused by the choice of A_2, and is thus not needed for the case of A_4 in the other automaton in Fig. 3.

A slightly less trivial fractal image, mathematically an algebra function $f \in \mathbf{F}$ of non-finite precision, can be generated by the weighted finite automaton in Fig. 4 with $A = A_4$, where the level-preserving transformers $\mathfrak{h}$, $\mathfrak{v}$ and $\mathfrak{d}$ are reflection transformers against the vertical, horizontal

Figure 3. Automata on A_4 and A_2

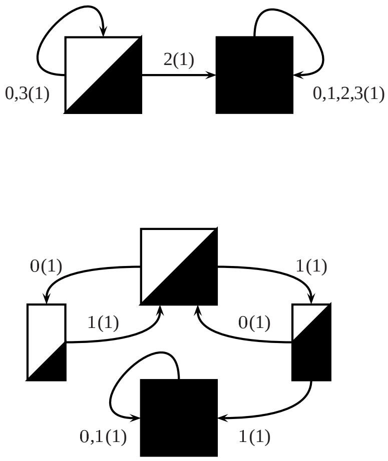
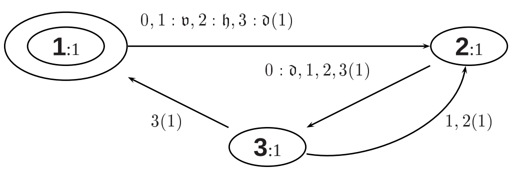


Figure 4. Graph Representation of an Automaton on A_4



and diagonal axises respectively. The label "3 : $\mathfrak{d}(1)$" on an edge, for instance, refers to the transition on the symbol "3" with weight "1", and to the fact that the state function needs to be transformed under "$\mathfrak{d}$" before being used. The label "2 : 1" inside an oval box, for instance, says it represents state "2" and has the average intensity value 1. The generated fractal image then becomes that in Fig. 5, and tiles up well with itself too. A natural image can also be represented or approximated by a weighted finite automaton whenever the image is regarded as an algebra function in **F** of finite precision. For instance, the images in Fig. 6 are the algebra function that can be generated from a weighted finite automaton of 1609, and respectively 941, states and 3 level-preserving transformers. If we save the automaton as the representation of the original image, we obtain in effect an image compression similar to that in [2–5]. In image compression based on weighted finite automata, the use of certain symmetry mappings are instrumental at reducing the total number of states, improving compression ratio in the course. Our level-preserving transformers are therefore the utmost legitimate generalisation of such symmetry mappings. We note that symmetries or similarities are im-

Figure 5. Fractal Image as an Algebra Function of Non-finite Precision

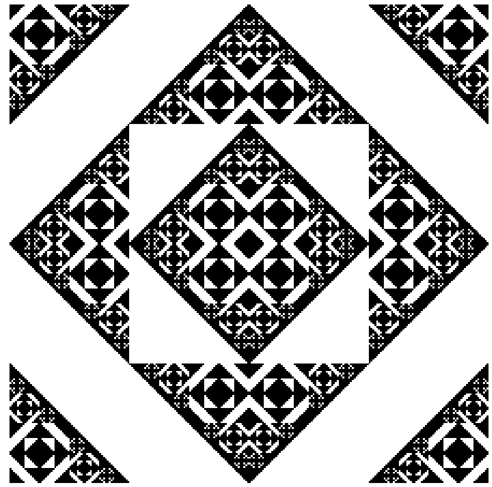
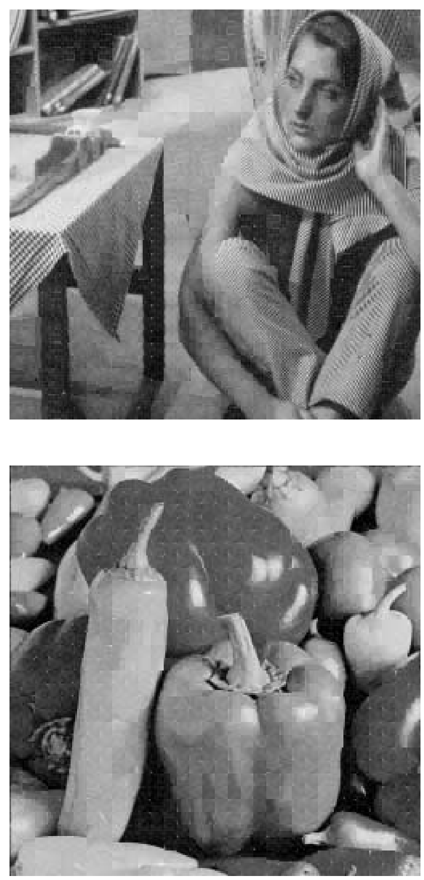


Figure 6. Image Generated by Weighted Finite Automata



mensely useful in various fields including for example the group theoretical analysis in both image processing and solutions of dynamical systems [7–9]. So are the weighted finite automata, see [10] for the application to the speech recognition.

6. Conclusions

We have studied in great details the structure and the characterisation of level-compatible algebra functions as well as the level-preserving transformers on such functions. It is shown that weighted finite automata may be used to represent or generate level-compatible algebra functions, and that the use of level-preserving transformers could significantly reduce the number of states in the resulting automata. We have also discussed the potential use of the theory developed here to other fields such as fractal generation and image compression.

References

[1] Culik II K and Karhumaki J, Finite automata computing real functions, SIAM Journal on Computing **23** (1994)789-814.

[2] Culik II K and Kari J, Inference algorithm for WFA and image compression, *Fractal Image Compression: Theory and Application*, edr. Fisher Y, Springer, New York 1995.

[3] Hafner U, Refining image compression with weighted finite automata, Proceedings of Data Compression Conference , edrs. Storer J and Cohn M, (1996)359–368.

[4] Hafner U, Weighted finite automata for video compression, IEEE Journal on Selected Areas in Communications , edrs. Enami K, Krikelis A and Reed T R, (1998)108–119.

[5] Jiang Z, Litow B and De Vel O, An inference implementation based on extended weighted finite automata, Proceedings of Australian Computer Science Conference, vol **23**, (2001)100–108.

[6] Culik II K and Kari J, Digital Images and Formal Languages, *Handbook of Formal Languages* **v3**, edrs. Rozenberg G and Salomaa A, (1997)599-616.

[7] Kanataki K, *Group Theoretical Methods in Image Understanding*, Springer Series in Information Sciences **20**, Springer, Berlin 1990.

[8] Jiang Z, Lie symmetries and their local determinancy for a class of differential-difference euqations, Physics Letters A **240** (1998)137–143.

[9] Jiang Z, The intrinsicality of Lie symmetries of $u_n^{(k)}(t) = F_n(t, u_{n+a}, ..., u_{n+b})$, Journal of Mathematical Analysis and Applications **227** (1998)396–419.

[10] Pereira F C and Riley M D, Speech recognition by composition of weighted finite automata, AT&T Labs, Murray Hill 1996.