Записки научных
семинаров ПОМИ
Том 402, 2012 г.

T. Sallinen

FUNCTION TRANSFORMATIONS WITH AUTOMATA

Abstract. We use conventional models of computations to define rather unconventional computational processes. Specifically, we use one-tape automata to compute real-valued functions and two-tape automata to describe transformations of those functions. As transformations we consider the integration and the derivation of a function.

§1. Introduction

Finite automata are among the simplest models of conventional computing. They operate on words over a finite alphabet Σ. As real numbers in the unit interval can be interpreted as infinite words, these automata can be used with infinite inputs to compute real-valued functions. For an ordinary finite nondeterministic automaton $\mathcal{A}$ we can associate to each accepted word the number of different ways it can be accepted, that is the multiplicity associated to the input. Further, instead of mere acceptance, we can use nondeterministic automata equipped also with weights, first introduced by Schützenberger in [8], to compute real-valued functions $f_{\mathcal{A}} : \Sigma^* \to \mathbb{R}_+$. Combining this with infinite computations we can use finite automata to define real functions $f : [0,1] \to \mathbb{R}_+$ (or $\mathbb{R}$), see [2].

Ordinary two-tape automata called transducers – that is automata with inputs and outputs – compute word functions $f : \Sigma_1^* \to \Sigma_2^*$ from an input alphabet to an output one. These constitute relations between words. With a weighted variant we can compute a weighted relation between words, $\rho : \Sigma_1^* \times \Sigma_2^* \to \mathbb{R}_+$. Finally, as we can see, a weighted relation on infinite words, that is of $\Sigma_1^\omega \times \Sigma_2^\omega$, can be interpreted as a transformation of a function into another one, as first formalized in [3].

In Sections 2 and 3 we describe how finite automata can be used to define real functions. In particular, we consider how the continuity of the function is related to the automaton defining it. Then in Section 4 we use finite (weighted) transducers as models to define transformations of real functions. Two particularly interesting such transformations are the

Key words and phrases: weighted finite automaton, weighted finite transducer, real-valued function, derivative, integral.

integration and the derivation of a given function. We analyze how this can be done on the level of automata representations of these functions. In particular, in turns out that although functions defined by finite automata are typically fractal type functions, and hence analytically complicated, they can be easily integrated, that is the automata-theoretic representation of the integral is easy to compute from the automata-theoretic presentation of the function.

The goal of this paper is to provide complete proofs for some results published earlier without proofs or with not sufficiently detailed proofs.

§2. Preliminaries

We consider words over the alphabet Σ, with ε being the empty word of length 0. Every infinite word can be interpreted as a real number in the interval $[0,1]$. The mapping $\widehat{\ }$:

$$\widehat{\ }\colon \Sigma^\omega \to [0,1], \widehat{\ }(w) = \widehat{w}$$

interpretes the word $w = w_1 w_2 \ldots$, where $w_i \in \Sigma = \{0, \ldots, n-1\}$, as the number $\widehat{w}$:

$$\widehat{w} = \sum_{i=1}^{\infty} w_i n^{-i},$$

which in the binary case $\Sigma = \{0,1\}$ can be written as:

$$\widehat{w} = \sum_{i=1}^{\infty} w_i 2^{-i}. \tag{1}$$

The only real numbers in the interval $[0,1]$ that have two representations $w = w_1 w_2 \ldots$, where $w_i \in \Sigma = \{0,1\}$, are those that have a finite representation (1), that is those that have a finite binary representation. For these numbers the representations are $v10^\omega$ and $v01^\omega$ for some finite word v. Let the former be the *standard representation*. Now the mapping $\widehat{\ }$:

$$\widehat{\ }\colon \Sigma^\omega \setminus \Sigma^* 1^\omega \to [0,1], \widehat{\ }(w) = \widehat{w}$$

is an injection. The values of (1) are obtained by iteratively halving the unit interval $[0,1]$: w_1 is 0 or 1 depending on whether the corresponding number is in the first or second half, and similarly w_i is 0 or 1 if the number is in the corresponding ith step on the left or right part, respectively. For example the words $0110w$ represent numbers in the interval illustrated in Figure 1.

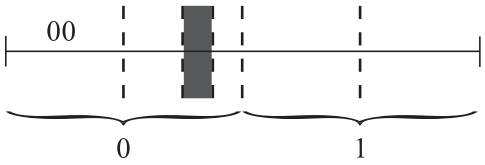


Fig. 1. Recursive subdivision and addressing of the words $0110w$.

We use automata defined in [2], whose transitions have been labeled with real numbers. A *Weighted Finite Automaton* (WFA) is a 5-tuple

$$\mathcal{A} = (Q, \Sigma, \mathcal{W}, I, F),$$

where

(i) $Q = \{q_1, q_2, \ldots, q_n\}$ is a finite ordered set of *states*,
(ii) Σ is a finite alphabet,
(iii) $\mathcal{W} = \{W_a \mid a \in \Sigma\}$, where $W_a : Q \times Q \to \mathbb{R}$ is a *weight function* of the transitions labeled by symbol a,
(iv) $I : Q \to \mathbb{R}$ is an *initial distribution* and
(v) $F : Q \to \mathbb{R}$ is a *final distribution*.

As the set of states is defined as linearly ordered we may use vector notations logically. The weight functions W_a can be interpreted as $n \times n$ matrices. If $W_a(p, q) = \alpha$, the element (p, q) of matrix W_a is α. In this formalization the initial distribution I is a row vector and the final distribution F a column vector. The *underlying automaton* of $\mathcal{A}$ is the NFA, whose transition relation contains exactly the 3-tuples (p, a, q), for which $W_a(p, q) \neq 0$ and whose initial and final states are those that have a non-zero value of images of I and F.

Weighted finite automata compute functions $\Sigma^* \to \mathbb{R}$. A word $u \in \Sigma^*$ defines a *distribution* $P_{\mathcal{A}}(u)$ on $\mathcal{A}$. It is defined recursively with matrix multiplication:

$$P_{\mathcal{A}}(\varepsilon) = I,$$
$$P_{\mathcal{A}}(va) = P_{\mathcal{A}}(v) \cdot W_a, \quad \text{for all } v \in \Sigma^*.$$

The word function $F_{\mathcal{A}}$ defined by $\mathcal{A}$ is

$$F_{\mathcal{A}} : \Sigma^* \to \mathbb{R}, \quad F_{\mathcal{A}}(u) = P_{\mathcal{A}}(u) \cdot F.$$

Weighted finite automata may compute real functions $[0, 1] \to \mathbb{R}$ when we extend the word function to infinite words and interpret real numbers

as words. Now $\mathcal{A}$ defines an *ω-word function*

$$f_{\mathcal{A}} : \Sigma^{\omega} \to \mathbb{R}, \quad f_{\mathcal{A}}(w) = \Big(\lim_{n\to\infty} P_{\mathcal{A}}\big(\mathrm{pref}_n(w)\big) \Big) \cdot F \tag{2}$$

and a *real function*

$$\widehat{f_{\mathcal{A}}} : [0,1] \to \mathbb{R}, \quad \widehat{f_{\mathcal{A}}}(x) = f_{\mathcal{A}}(w),$$

where w is the standard representation of x.

The definition (2) requires some convergence analysis. For that reason we define a type of automaton, for which $f_{\mathcal{A}}$ is always defined and finite. A WFA $\mathcal{A}$ is a *level automaton*, if and only if

(i) the only cycles in the underlying automaton are of the form $p \xrightarrow{a} p$,
(ii) $W_a(p,q) \geq 0$ when $p, q \in Q$ and $a \in \Sigma$,
(iii) for all $p \in Q$: if there exist such $q \in Q$, $b \in \Sigma$, $q \neq p$, that $W_b(p,q) \neq 0$, then $0 \leq W_a(p,p) < 1$ for all $a \in \Sigma$; otherwise $W_a(p,p) = 1$ for all $a \in \Sigma$,
(iv) the elements of I and F are non-negative real numbers, except for some applications, and
(v) the underlying automaton does not have unused states.

The condition (i) implies, that for every state of a level automaton $\mathcal{A}$, there exists a unique number, which we will call the *degree* of the state. States which at the underlying automaton lead only to themselves, are of degree 0. For a state q_i that leads to other states, let the smallest degree of those other states be j. Then q_i is of degree $j+1$. A level automaton $\mathcal{A}$ is a *line automaton* if and only if for each number $0, \ldots, n-1$ there exists exactly one state of that degree. A level automaton $\mathcal{A}$ with $\Sigma = \{0,1\}$ is 0-*faithful*, if and only if

$$\sum_{q\in Q} W_0(p,q) = 1 \quad \text{for all } p \in Q.$$

If $F = (1, 1, \ldots, 1)$, we have

Lemma 1. *If a level automaton $\mathcal{A}$ is 0-faithful, then*

$$F_{\mathcal{A}}(v) = f_{\mathcal{A}}(v0^{\omega}) \quad \textit{for all } v \in \Sigma^*.$$

Proof. By definition, the sum of the elements on every row of W_0 is 1. Therefore, since all elements in the final distribution are ones, we have $W_0 F = F$. □

Consequently, if $x \in [0,1)$ has a finite binary representation $v1$, then for a 0-faithful level automaton we have $F_{\mathcal{A}}(v1) = \widehat{f}_{\mathcal{A}}(x)$. Finally, a WFA $\mathcal{A}$ is called *average preserving* if for all $p \in Q$,

$$\sum_{a \in \Sigma, q \in Q} W_a(p,q) \cdot F(q) = |\Sigma| \cdot F(p),$$

where $F(p)$ is the element in the final distribution corresponding to the state p.

Both the property of being 0-faithful and the property of being average preserving provide a normal form for level automata computing real functions as any function computed by a level automaton can be computed also with a 0-faithful or average preserving automaton, for details, see [2].

§3. Computing real functions

As an important result of imposing the level structure, we have:

Theorem 2. *Let $\mathcal{A}$ be a level automaton. The value of the function $f_{\mathcal{A}}(w)$ is defined and finite for all $w \in \{0,1\}^\omega$. The function $\widehat{f}_{\mathcal{A}}$ is continuous at all points x, which do not have a finite binary representation.*

A detailed proof can be found in [2]. The theorem immediately yields a condition for continuity.

Corollary 3. *Let $\mathcal{A}$ be a level automaton. The function $\widehat{f}_{\mathcal{A}}$ is continuous, if and only if it is continuous at all points that have a finite binary representation.*

The point $\frac{1}{2}$ has a special role.

Lemma 4. *Let $\mathcal{A}$ be a level automaton. The function $\widehat{f}_{\mathcal{A}}$ is continuous at point $\frac{1}{2}$, if and only if*

$$f_{\mathcal{A}}(01^\omega) = f_{\mathcal{A}}(10^\omega).$$

Proof. Obviously the equality is a necessary condition. The sufficiency follows from Theorem 2, for details, see [2]. □

Deciding continuity at any point having a finite binary representation can be reduced to deciding continuity at point $\frac{1}{2}$ for another automaton.

Lemma 5. *Let $x \in (0,1)$ have a finite binary representation $v1$. Let $\mathcal{A}$ be a level automaton and $\mathcal{A}_v$ the automaton $\mathcal{A}$ with the initial distribution $P_{\mathcal{A}}(v)$. Then $\widehat{f}_{\mathcal{A}}$ is continuous at x, if and only if $\widehat{f}_{\mathcal{A}_v}$ is continuous at point $\frac{1}{2}$.*

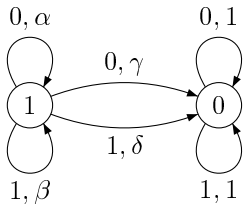


Fig. 2. The two state automaton $\mathcal{A}_\ell$.

Proof. Follows from Lemma 4, when we notice that

$$P_{\mathcal{A}}(vu) = P_{\mathcal{A}_v}(u),$$

for all $u \in \Sigma^*$. □

If an automaton defines a continuous function with every initial distribution, we call it *strongly continuous.*

For any two state line automaton $\mathcal{A}_\ell$ that has weight matrices

$$W_0 = \begin{pmatrix} \alpha & \gamma \\ 0 & 1 \end{pmatrix} \text{ and } W_1 = \begin{pmatrix} \beta & \delta \\ 0 & 1 \end{pmatrix},$$

depicted in Figure 2, we have a simple continuity condition. The proof can again be found in [2].

Theorem 6. *With initial distribution* $(1,0)$ *and final distribution* $(1,1)$*, the line automaton* $\mathcal{A}_\ell$*, for which* $\alpha, \beta \in [0,1)$ *and* $0 \leq \gamma, \delta$*, defines a continuous function* $f_{\mathcal{A}_\ell}$*, if and only if at least one of the following conditions is fulfilled:*

(i) $\alpha + \beta = 1$,
(ii) $\delta(1-\alpha) = \gamma(1-\beta)$.

Example 1. We consider the automaton $\mathcal{A}_\ell$ with $\alpha = \beta = \delta = \frac{1}{2}$ and $\gamma = 0$. The component of the distribution corresponding to the state 1 equals $\frac{1}{2}^{|v|}$ for an input word v. For every input symbol 1 that component is added to the distribution in state 0 multiplied with the coefficient $\frac{1}{2}$. As a result, we have

$$f_{\mathcal{A}_\ell}(w) = \lim_{n\to\infty} \left(2^{-n} + \sum_{i=1}^{n} w_i 2^{-i} \right) = \sum_{i=1}^{\infty} w_i 2^{-i},$$

for $w = w_1 w_2 \ldots$. Comparing this with (1) it is easy to see that $\widehat{f}_{\mathcal{A}_\ell}(x) = x$.

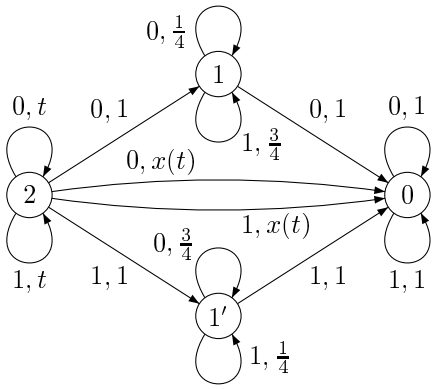


Fig. 3. The four state automaton $\mathcal{A}(t)$.

Now we start to analyze a particular parameterized family of automata $\mathcal{A}(t)$, depicted in Figure 3, with weight matrices (4), initial distribution $(1,0,0,0)$ and final distribution $(0,0,0,1)$. The states are in the order $(2,1,1',0)$. The choice of $x(t)$ is explained later.

Theorem 7. *The automaton $\mathcal{A}(t)$ is strongly continuous.*

Proof. The automaton $\mathcal{A}(t)$ computes the function $g : \Sigma^\omega \to \mathbb{R}$ with an initial distribution (a,b,c,d). Then, because the state 0 is not connected back to other states and the states 1 and $1'$ are not connected to each other, we have

$$g = af_{\mathcal{A}(t)} + bf_{\mathcal{A}_1} + cf_{\mathcal{A}_{1'}} + d,$$

where $\mathcal{A}_1$ and $\mathcal{A}_{1'}$ are the subautomata containing only the states $\{1,0\}$ and $\{1',0\}$, respectively. According to the definition of strong continuity and Lemma 5, the automaton $\mathcal{A}(t)$ is strongly continuous, if $f_{\mathcal{A}(t)}$, $f_{\mathcal{A}_1}$ and $f_{\mathcal{A}_{1'}}$ are continuous. Because the automata $\mathcal{A}_1$ and $\mathcal{A}_{1'}$ satisfy the condition (i) in Theorem 6, they define continuous functions. The continuity of $f_{\mathcal{A}(t)}$ follows from the symmetry of $\mathcal{A}(t)$ with respect to the labels 0 and 1, as $f_{\mathcal{A}(t)}(01^n) = f_{\mathcal{A}(t)}(10^n)$ for any $n \geq 0$. □

However, the subautomata $\mathcal{A}'$, containing three states in the order $(2,1,0)$, and $\mathcal{A}''$ with the states $(2,1',0)$, define continuous functions only for some values of $x(t)$. This guides us to fix the value $x(t)$. The continuity

condition

$$f_{\mathcal{A}'}(10^\omega) = f_{\mathcal{A}'}(01^\omega)$$

by the weight matrices

$$W_0 = \begin{pmatrix} t & 1 & x(t) \\ 0 & \frac{1}{4} & 1 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } W_1 = \begin{pmatrix} t & 0 & x(t) \\ 0 & \frac{3}{4} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

of the automaton $\mathcal{A}'$ is equivalent to

$$t\left(\frac{1}{1-t}x(t) + \frac{1}{1-t}\cdot 1 \cdot \frac{4}{3}\right) = x(t),$$

and further to

$$x(t) = \frac{4t}{2t-1},$$

whenever $t \neq \frac{1}{2}$. Negative values of the function $x(t)$ bring the automaton $\mathcal{A}(t)$ outside of the class of level automata, but as the negative value is merely a weight of a noncycle, this is only a technicality.

The automaton $\mathcal{A}(t)$ in Figure 3 has been examined originally in [4], with the values $t \in \left\{\frac{1}{4}, \frac{2}{3}, \frac{3}{4}\right\}$, and later in [7], [5] and [6]. Therein it was reported:

Theorem 8. *The function $\widehat{f}_{\mathcal{A}(t)}$, for $t \in (0,1)$, with $t \notin \left\{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}\right\}$ and*

$$x(t) = \frac{4t}{2t-1},$$

is continuous, but does not have a derivative at any point.

Proof. The continuity follows from Theorem 7. Let now $x \in [0,1)$ be an arbitrary point and w its binary representation with w_n its prefix of length n. We will examine the following ω-words:

$$\begin{aligned} w_0(n) &= w_n 0^\omega, \\ w_1(n) &= w_n 10^\omega, \\ w_2(n) &= w_n 110^\omega \text{ and} \\ w_3(n) &= w_n 1^\omega. \end{aligned}$$

We notice that

$$|\widehat{w} - \widehat{w_i(n)}| \leq \frac{1}{2^n}, \tag{3}$$

for $i = 0, 1, 2, 3$. We denote the distribution given by w_n as

$$P_{\mathcal{A}(t)}(w_n) = (\alpha_n, \beta_n, \gamma_n, \delta_n)$$

and compute the values $f_{\mathcal{A}(t)}\big(w_i(n)\big)$.

The weight matrices associated to the automaton $\mathcal{A}(t)$ are

$$W_0 = \begin{pmatrix} t & 1 & 0 & x(t) \\ 0 & \frac{1}{4} & 0 & 1 \\ 0 & 0 & \frac{3}{4} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad W_1 = \begin{pmatrix} t & 0 & 1 & x(t) \\ 0 & \frac{3}{4} & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{4}$$

We need the limits $\lim_{n\to\infty}(W_0)^n$ and $\lim_{n\to\infty}(W_1)^n$. Focusing merely to the uppermost row and disregarding the nonessential values marked with stars, we have

$$\begin{aligned} (W_0)^k \cdot W_0 &= \begin{pmatrix} t^k & p_k(t) & * & q_k(t) \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \cdot \begin{pmatrix} t & 1 & 0 & x(t) \\ 0 & \frac{1}{4} & 0 & 1 \\ 0 & 0 & \frac{3}{4} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} t^{k+1} & t^k + \frac{1}{4}p_k(t) & * & t^k \cdot x(t) + p_k(t) + q_k(t) \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \\ &= (W_0)^{k+1}, \end{aligned}$$

and the recursions

$$\begin{aligned} q_{k+1}(t) &= t^k \cdot x(t) + p_k(t) + q_k(t) \text{ and} \\ p_k(t) &= t^{k-1} + \frac{1}{4}p_{k-1}(t) \end{aligned}$$

with the initial values $p_1(t) = 1$ and $q_1(t) = x(t)$. Then, generally

$$\begin{aligned} q_k(t) &= \sum_{i=0}^{k-1} \left(t^i \cdot x(t) + \sum_{j=1}^{i} \left(t^{j-1} \cdot \left(\tfrac{1}{4}\right)^{i-j} \right) \right) \\ &= \sum_{i=0}^{k-1} \left(t^i \cdot x(t) + \frac{(4t)^i - 1}{4^{i-1}(4t-1)} \right), \end{aligned}$$

where, by definition, the empty inner sum, when $i = 0$, equals zero. The case for W_1 is symmetric. We now have, for $0 \le t < 1$,

$$\lim_{n\to\infty}(W_0)^n = \begin{pmatrix} 0 & 0 & 0 & r(t) \\ 0 & 0 & 0 & \frac{4}{3} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ and } \lim_{n\to\infty}(W_1)^n = \begin{pmatrix} 0 & 0 & 0 & r(t) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{4}{3} \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

where

$$r(t) = \sum_{i=0}^{\infty}\left(t^i \cdot x(t) + \frac{(4t)^i - 1}{4^{i-1}(4t-1)}\right) = \frac{x(t)}{1-t} - \frac{4}{3(t-1)}.$$

Now we are ready to compute the values $f_{\mathcal{A}(t)}\big(w_i(n)\big)$. We have

$$\begin{aligned}
f_{\mathcal{A}(t)}\big(w_0(n)\big) &= r(t)\alpha_n + \tfrac{4}{3}\beta_n + \delta_n, \\
f_{\mathcal{A}(t)}\big(w_1(n)\big) &= \big(t \cdot r(t) + x(t)\big)\alpha_n + \beta_n + \gamma_n + \delta_n, \\
f_{\mathcal{A}(t)}\big(w_2(n)\big) &= \big(t^2 \cdot r(t) + t \cdot x(t) + x(t) + 1\big)\alpha_n + \tfrac{3}{4}\beta_n + \tfrac{5}{4}\gamma_n + \delta_n \text{ and} \\
f_{\mathcal{A}(t)}\big(w_3(n)\big) &= r(t)\alpha_n + \tfrac{4}{3}\gamma_n + \delta_n.
\end{aligned}$$

We also note that $\alpha_n = t^n$ for all n. Next we examine the differences

$$\begin{aligned}
f_{\mathcal{A}(t)}\big(w_0(n)\big) - f_{\mathcal{A}(t)}\big(w_3(n)\big) &= \tfrac{4}{3}(\beta_n - \gamma_n) \text{ and} \\
f_{\mathcal{A}(t)}\big(w_1(n)\big) - f_{\mathcal{A}(t)}\big(w_2(n)\big) &= (t \cdot r(t) - t^2 \cdot r(t) - t \cdot x(t) - 1)\alpha_n \\
&\quad + \tfrac{1}{4}(\beta_n - \gamma_n),
\end{aligned} \tag{5}$$

where the coefficient of α_n can further be modified to

$$\left(\frac{x(t)}{1-t} - \frac{4}{3(t-1)}\right)(t - t^2) - t \cdot x(t) - 1 = \frac{4}{3}t - 1.$$

Thus the coefficient is given by a linear function depending on a parameter t. The function is continuous in the interval $[0,1)$ except where the functions $x(t)$ and $r(t)$ are not continuous. The coefficient for α_n is zero, if and only if $t = \frac{3}{4}$.

Now, regardless of the values of β_n and γ_n, for every n at least one of the differences (5) has an absolute value of at least $\frac{1}{2}\left|\frac{4}{3}t - 1\right|\alpha_n$. Then for all n there exists an $i_n \in \{0,1,2,3\}$ for which

$$|f_{\mathcal{A}(t)}(w) - f_{\mathcal{A}(t)}\big(w_0(n)\big)| \ge \frac{1}{4}\left|\frac{4}{3}t - 1\right|\alpha_n. \tag{6}$$

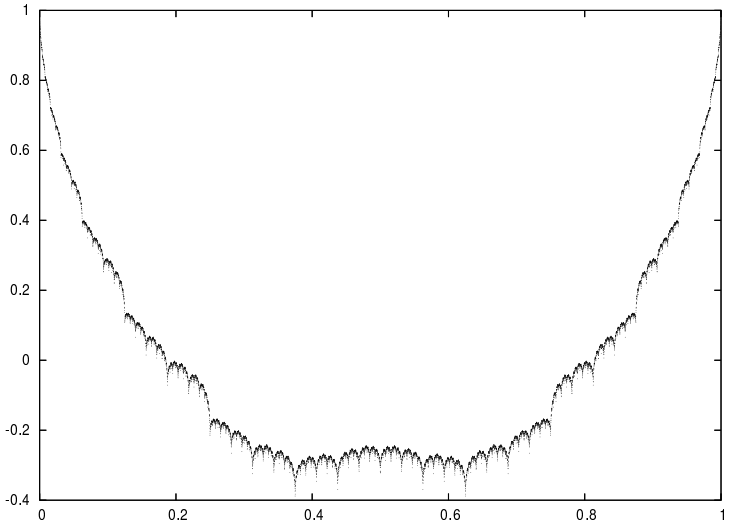


Fig. 4. Graph of the function $\widehat{f}_{\mathcal{A}(\frac{2}{3})}(x)$.

Because the values i_n are taken from a finite set, the inequality (6) holds with a fixed value of i_n infinitely many times. Thus there exists an infinite subset $I_0 \subseteq \mathbb{N}$ and a number $i_n \in \{0,1,2,3\}$ for which

$$|f_{\mathcal{A}(t)}(w) - f_{\mathcal{A}(t)}(w_{i_0}(n))| \geq \left|\frac{1}{3}t - \frac{1}{4}\right| \alpha_n.$$

Combining this with (3) we have

$$\frac{|f_{\mathcal{A}(t)}(w) - f_{\mathcal{A}(t)}(w_{i_0}(n))|}{|\widehat{w} - \widehat{w_{i_0}(n)}|} \geq \left|\frac{1}{3}t - \frac{1}{4}\right| (2t)^n. \tag{7}$$

Because the set I_0 is infinite, the function $\widehat{f}_{\mathcal{A}}(t)$ does not have a derivative at the point defined by w, if the right-hand side of (7) grows unbounded when n tends to infinity. This is what happens, when $\frac{1}{2} < t < \frac{3}{4}$ or $\frac{3}{4} < t < 1$. The case for $0 < t < \frac{1}{2}$ is symmetric. □

An example of the function with $t = \frac{2}{3}$ is shown in Figure 4. For $t = \frac{1}{4}, \frac{3}{4}$ the derivative exists and equals zero at all points having a finite binary representation, see [4]. For $t = \frac{1}{2}$ the function $x(t)$ is undefined.

§4. Transformation of functions defined by level automata

Our goal here is to take functions computed by level automata, make desirable modifications to them and find new automata computing the resulting functions. As a tool we use transducers. They are a special type of automata, which in their basic form transform an input from an input alphabet into an output in an output alphabet.

We use transducers computing weights, or more specifically weighted relations between two words. Essentially, they transform a function by associating the value computed for each input word of the original function to the value computed for each input word (that is, output of the transducer) of the new function. The new value is the original value multiplied by the weighted relation between the words.

Formally, as defined in [3], a *Weighted Finite Transducer* (WFT) is a 6-tuple

$$\mathcal{T} = (Q, \Sigma_1, \Sigma_2, \mathcal{W}, I, F),$$

where

(i) $Q = \{q_1, q_2, \ldots, q_n\}$ is a finite ordered set of states,
(ii) Σ_1 and Σ_2 are two finite alphabets,
(iii) $\mathcal{W} = \{W_{a,b} \mid a \in \Sigma_1 \cup \varepsilon, b \in \Sigma_2 \cup \varepsilon\}$, where $W_{a,b} : Q \times Q \to \mathbb{R}$ is a weight function of the transitions labeled by symbols a and b,
(iv) $I : Q \to \mathbb{R}$ is an initial distribution and
(v) $F : Q \to \mathbb{R}$ is a final distribution.

If $W_{a,b}(p, q) = 0$ whenever $a = \varepsilon$ or $b = \varepsilon$ for every $p, q \in Q$, we will call the WFT *ε-free*. If $W_{a,b}(p, p) = 0$ whenever $a = \varepsilon$ or $b = \varepsilon$ for every $p \in Q$, we will call the WFT *ε-loop free*.

The *path* σ from state p to state q is a sequence of adjacent transitions

$$\sigma = p \xrightarrow{a_1,b_1} p_1 \xrightarrow{a_2,b_2} p_2 \to \cdots \to p_{n-1} \xrightarrow{a_n,b_n} q.$$

If $p = q$, we call such a path a *loop*. Let $u = a_1 \cdots a_n$ and $v = b_1 \cdots b_n$ with $u \in \Sigma_1^*$ and $v \in \Sigma_2^*$. We define the *projections* π^1 and π^2 of the path σ into Σ_1^* and Σ_2^* as $\pi^1(\sigma) = u$ and $\pi^2(\sigma) = v$, respectively. The weight of the path σ denoted by $W(\sigma)$ is the product of all weights of all transitions

of σ:

$$\begin{aligned} W(\sigma) &= W\left(p \xrightarrow{a_1,b_1} p_1\right) \cdot W\left(p_1 \xrightarrow{a_2,b_2} p_2\right) \cdot \ \cdots \ \cdot W\left(p_{n-1} \xrightarrow{a_n,b_n} q\right) \\ &= W_{a_1,b_1}(p,p_1) \cdot W_{a_2,b_2}(p_1,p_2) \cdot \ \cdots \ \cdot W_{a_n,b_n}(p_{n-1},q). \end{aligned} \tag{8}$$

Let $\mathcal{T} = (Q, \Sigma_1, \Sigma_2, \mathcal{W}, I, F)$ be a WFT and let

$$T_{p,q}(u,v) = \sum W(\sigma), \tag{9}$$

where the sum is over all paths σ from p to q such that $\pi^1(\sigma) = u$ and $\pi^2(\sigma) = v$, provided that the sum converges. We define the *weighted relation* $\rho_{\mathcal{T}}$ as

$$\rho_{\mathcal{T}}(u,v) = \sum_{p,q \in Q} I(p) T_{p,q}(u,v) F(q), \tag{10}$$

for $u \in \Sigma_1^*$ and $v \in \Sigma_2^*$. If the sum does not converge, $\rho_{\mathcal{T}}(u,v)$ remains undefined.

In the special case of an ε-free WFT $\mathcal{T}$, for words $u = a_1 a_2 \dots a_k \in \Sigma_1^k$ and $v = b_1 b_2 \dots b_k \in \Sigma_2^k$ of equal length we have, by (8), (9) and (10)

$$\rho_{\mathcal{T}}(u,v) = I \cdot W_{a_1,b_1} \cdot W_{a_2,b_2} \cdot \ \cdots \ \cdot W_{a_k,b_k} \cdot F, \tag{11}$$

and $\rho_{\mathcal{T}}(u',v') = 0$, if $|u'| \neq |v'|$.

Let ρ be a weighted relation over $\Sigma_1^* \times \Sigma_2^*$, that is a function $\Sigma_1^* \times \Sigma_2^* \to \mathbb{R}$, and let f be a function $\Sigma_1^* \to \mathbb{R}$. We define the application of ρ to f to be the function $g : \Sigma_2^* \to \mathbb{R}$, $g = \rho(f)$, where

$$g(v) = \sum_{u \in \Sigma_1^*} \rho(u,v) f(u), \tag{12}$$

for $v \in \Sigma_2^*$, if the sum, which can be infinite, converges. The application $\mathcal{T}(f)$ of the WFT $\mathcal{T}$ to f is defined as the application of the weighted relation $\rho_{\mathcal{T}}$ to f, that is $\mathcal{T}(f) = \rho_{\mathcal{T}}(f)$. This is how we use weighted relations to transform real-valued functions. Similarly, we define $\mathcal{T}(\mathcal{A})$, the application of a transducer $\mathcal{T}$ to an automaton $\mathcal{A}$, as applying the weighted relation $\rho_{\mathcal{T}}$ to $f_{\mathcal{A}}$. Next we show how weighted transducers are used to define $\rho_{\mathcal{T}}(f)$ in terms of automata-theoretic representation of real functions.

We denote by $\otimes$ the ordinary *tensor product* of matrices, which is also called the *Kronecker product* or *direct product*. It is defined as follows. Let K and M be matrices of sizes $k \times \ell$ and $m \times n$, respectively. Then their

tensor product is the matrix:

$$K \otimes M = \begin{pmatrix} K_{11}M & \cdots & K_{1\ell}M \\ \vdots & & \vdots \\ K_{k1}M & \cdots & K_{k\ell}M \end{pmatrix},$$

of size $km \times \ell n$. Using the tensor product we can establish a very natural relation between automata and transducers.

The core of our transformation is shown in the next result, originally stated in [1].

Theorem 9. *Let $\mathcal{A} = (Q_{\mathcal{A}}, \Sigma_1, \mathcal{W}_{\mathcal{A}}, I_{\mathcal{A}}, F_{\mathcal{A}})$ be a WFA and $\mathcal{T} = (Q, \Sigma_1, \Sigma_2, \mathcal{W}, I, F)$ be an ε-free transducer. Then there exists a WFA $\mathcal{A}'$ such that $f_{\mathcal{A}'} = \rho_{\mathcal{T}} \circ f_{\mathcal{A}}$, that is an automaton which directly computes the function that results in applying the weighted relation $\rho_{\mathcal{T}}$ to the original function $f_{\mathcal{A}}$.*

Outline of proof. Let $\mathcal{A}$ have m states and $\mathcal{T}$ have n states. Then we construct an mn-state WFA $\mathcal{A}' = \mathcal{T}(\mathcal{A})$ over alphabet $a \in \Sigma_2$ with initial distribution $I_{\mathcal{A}'} = I \otimes I_{\mathcal{A}}$, final distribution $F_{\mathcal{A}'} = F \otimes F_{\mathcal{A}}$ and weight matrices

$$W_{\mathcal{A}',b} = \sum_{a \in \Sigma_1} W_{a,b} \otimes W_{\mathcal{A},a},$$

for all $b \in \Sigma_2$, where $W_{\mathcal{A}',b}$ and $W_{\mathcal{A},a}$ are the matrices W_b and W_a for the automata $\mathcal{A}'$ and $\mathcal{A}$, respectively. As the transducer $\mathcal{T}$ is ε-free, the weight matrices are uniquely determined. Now, by (11) and definition (12), we can see that $\mathcal{A}'$ computes the function $f_{\mathcal{A}'} = \rho_{\mathcal{T}}(f_{\mathcal{A}})$, that is the function defined by $\mathcal{A}'$ is the same as the result of application of the WFT $\mathcal{T}$ to the function computed by the WFA $\mathcal{A}$. □

Next we illustrate the power of transformations transducers are capable of. The following two results are reported without proofs in [3]. We begin by analyzing the transducer $\mathcal{T}_{\mathrm{d}}$ in Figure 5 that has weight matrices

$$W_{0,0} = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}, W_{0,1} = \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix}, W_{1,0} = \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix} \text{ and } W_{1,1} = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}$$

and initial and final distributions $(1, 1)$ and $(-1, 1)$, respectively. The states of the transducer $\mathcal{T}_{\mathrm{d}}$ are ordered from left to right in Figure 5.

Theorem 10. *Let $\Sigma = \{0, 1\}$ and $\mathcal{A}$ be a 0-faithful level automaton over Σ. Then we can construct such a WFA $\mathcal{A}_{\mathrm{d}}$ that if $\widehat{f_{\mathcal{A}}}$ has a derivative at*

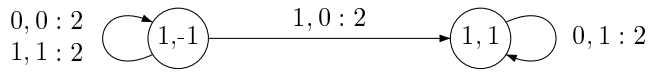


Fig. 5. A WFT $\mathcal{T}_{\mathrm{d}}$ computing the right derivative $\frac{\mathrm{d}\widehat{f}(x)}{\mathrm{d}x}$.

point $x = \widehat{w}$, we have

$$\widehat{f}_{\mathcal{A}_{\mathrm{d}}}(x) = \frac{\mathrm{d}\widehat{f}_{\mathcal{A}}(x)}{\mathrm{d}x}.$$

Proof. For a sufficiently large i let $w_i = \mathrm{pref}_i(w)$ and $x_i = \widehat{w_i 0^\omega}$. Then we have $x - 2^{-i} \le x_i \le x$. By Lemma 1 we have $F_{\mathcal{A}}(w_i) = f_{\mathcal{A}}(w_i 0^\omega) = \widehat{f}_{\mathcal{A}}(x_i)$. We apply the transducer $\mathcal{T}_{\mathrm{d}}$ in Figure 5 to the automaton $\mathcal{A}$. With input w_i the construction resulting from the left-hand side state of $\mathcal{T}_{\mathrm{d}}$ computes $-2^i \cdot \widehat{f}_{\mathcal{A}}(x_i)$, because each transition weight is doubled and the final distribution negated. The construction resulting from the right-hand side state always deduces the lexicographically smallest binary word of equal length larger than w_{i+j} and uses it as input computing with input $w_{i-1}01^j$ the function $2^{i+j} \cdot F_{\mathcal{A}}(w_{i-1}10^j)$. As

$$\widehat{w_{i-1}10^j}0^\omega - \widehat{w_{i-1}01^j}0^\omega = 2^{-i-j},$$

with input w_i the right-hand side state computes $2^i \cdot \widehat{f}_{\mathcal{A}}\big(x_i + 2^{-i}\big)$ and the whole construction computes

$$F_{\mathcal{T}_{\mathrm{d}}(\mathcal{A})}(w_i) = -2^i \cdot \widehat{f}_{\mathcal{A}}(x_i) + 2^i \cdot \widehat{f}_{\mathcal{A}}\big(x_i + 2^{-i}\big) = \frac{\widehat{f}_{\mathcal{A}}\big(x_i + 2^{-i}\big) - \widehat{f}_{\mathcal{A}}(x_i)}{2^{-i}}. \tag{13}$$

Let now $\mu > 0$. As we assumed the existence of the derivative, the right derivative then exists and equals the derivative at x. Then there exists $\delta > 0$ such that for every h satisfying $|h| < \delta$ we have

$$\left| \frac{\widehat{f}_{\mathcal{A}}(x+h) - \widehat{f}_{\mathcal{A}}(x)}{h} - \widehat{f}'_{\mathcal{A}}(x) \right| < \mu.$$

Let $i > -\log_2 \delta$ so that $2^{-i} < \delta$. Then we have

$$\left| \frac{\widehat{f}_{\mathcal{A}}(x_i) - \widehat{f}_{\mathcal{A}}(x)}{x_i - x} - \widehat{f}'_{\mathcal{A}}(x) \right| < \mu$$

and

$$\left|\frac{\widehat{f}_{\mathcal{A}}(x_i+2^{-i})-\widehat{f}_{\mathcal{A}}(x)}{x_i+2^{-i}-x}-\widehat{f}'_{\mathcal{A}}(x)\right|<\mu.$$

Using these and the triangle inequality we can find an upper bound for the difference of (13) and the derivative:

$$\left|\frac{\widehat{f}_{\mathcal{A}}(x_i+2^{-i})-\widehat{f}_{\mathcal{A}}(x_i)}{2^{-i}}-\widehat{f}'_{\mathcal{A}}(x)\right|=\left|\frac{\widehat{f}_{\mathcal{A}}(x_i+2^{-i})-\widehat{f}_{\mathcal{A}}(x_i)-2^{-i}\cdot\widehat{f}'_{\mathcal{A}}(x)}{2^{-i}}\right|$$

$$=\left|\frac{\left(\widehat{f}_{\mathcal{A}}(x_i+2^{-i})-\widehat{f}_{\mathcal{A}}(x)-(x_i+2^{-i}-x)\cdot\widehat{f}'_{\mathcal{A}}(x)\right)-\left(\widehat{f}_{\mathcal{A}}(x_i)-\widehat{f}_{\mathcal{A}}(x)-(x_i-x)\cdot\widehat{f}'_{\mathcal{A}}(x)\right)}{2^{-i}}\right|$$

$$\leq\left|\frac{\left|\widehat{f}_{\mathcal{A}}(x_i+2^{-i})-\widehat{f}_{\mathcal{A}}(x)-(x_i+2^{-i}-x)\cdot\widehat{f}'_{\mathcal{A}}(x)\right|+\left|\widehat{f}_{\mathcal{A}}(x_i)-\widehat{f}_{\mathcal{A}}(x)-(x_i-x)\cdot\widehat{f}'_{\mathcal{A}}(x)\right|}{2^{-i}}\right|$$

$$<\frac{\mu|x_i+2^{-i}-x|+\mu|x_i-x|}{2^{-i}}=\frac{\mu(x_i+2^{-i}-x-x_i+x)}{2^{-i}}=\frac{\mu\cdot 2^{-i}}{2^{-i}}=\mu,$$

that is, for any $\mu>0$ the value of $F_{\mathcal{T}_{\mathrm{d}}(\mathcal{A})}(w_i)$, with sufficiently large i, differs from the derivative at $\widehat{w}$ by less than μ. It follows that $\mathcal{A}_{\mathrm{d}}=\mathcal{T}_{\mathrm{d}}(\mathcal{A})$. □

In Theorem 10 we assumed the derivability in the given point. We do not know how much weaker assumptions would be sufficient here.

Next we move on to the transducer INT in Figure 6 that has weight matrices

$$W_{0,0}=\left(\tfrac{1}{2}\right),W_{0,1}=\left(\tfrac{1}{2}\right),W_{1,0}=\left(\tfrac{1}{2}\right)\text{ and }W_{1,1}=\left(\tfrac{1}{2}\right),$$

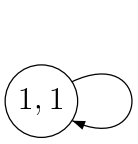
along with $I=F=(1)$, and the transducer $\mathcal{T}_{\mathrm{R}}$ in Figure 7 with

$$W_{0,0}=\begin{pmatrix}\frac{1}{2}&0\\0&\frac{1}{2}\end{pmatrix},W_{0,1}=\begin{pmatrix}0&\frac{1}{2}\\0&\frac{1}{2}\end{pmatrix},W_{1,0}=\begin{pmatrix}0&0\\0&\frac{1}{2}\end{pmatrix}\text{ and }W_{1,1}=\begin{pmatrix}\frac{1}{2}&0\\0&\frac{1}{2}\end{pmatrix}$$

and initial and final distributions $(1,0)$ and $(1,1)$, respectively. The states of the transducer $\mathcal{T}_{\mathrm{R}}$ are ordered from left to right in Figure 7.

Theorem 11. *Let $\Sigma=\{0,1\}$ and $\mathcal{A}$ be an average preserving level automaton over Σ defining a Riemann-integrable function. Then we can construct such a WFA $\mathcal{A}_{\mathrm{R}}$ that:*

$$\widehat{f}_{\mathcal{A}_{\mathrm{R}}}(x)=\int_0^x\widehat{f}_{\mathcal{A}}(t)\ \mathrm{d}t.$$



$0, 0 : \frac{1}{2}$
$0, 1 : \frac{1}{2}$
$1, 0 : \frac{1}{2}$
$1, 1 : \frac{1}{2}$

Fig. 6. A WFT INT.

Proof. We begin by analyzing the transducer INT in Figure 6. Given a WFA $\mathcal{A}$ that has weight matrices W_0 and W_1, the WFA $\mathrm{INT}(\mathcal{A})$ computes the distributions

$$P_{\mathrm{INT}(\mathcal{A})}(0) = \frac{1}{2}P_{\mathcal{A}}(0) + \frac{1}{2}P_{\mathcal{A}}(1) = P_{\mathrm{INT}(\mathcal{A})}(1),$$
$$P_{\mathrm{INT}(\mathcal{A})}(v0) = \frac{1}{2}P_{\mathrm{INT}(\mathcal{A})}(v)W_0 + \frac{1}{2}P_{\mathrm{INT}(\mathcal{A})}(v)W_1 = P_{\mathrm{INT}(\mathcal{A})}(v1)$$

and thus, with any finite word v,

$$F_{\mathrm{INT}(\mathcal{A})}(v) = \sum_{\substack{u\in\Sigma^* \\ |u|=|v|}} \frac{1}{2^{|u|}}F_{\mathcal{A}}(u),$$

that is the average of $F_{\mathcal{A}}$ over every input word of length $|v|$. With infinite words w, the average becomes

$$f_{\mathrm{INT}(\mathcal{A})}(w) = \lim_{i\to\infty}\left(\frac{1}{2^i}\cdot\sum_{j=0}^{2^i-1}\widehat{f}_{\mathcal{A}}\left(\frac{j}{2^i}\right)\right) = \int_0^1 \widehat{f}_{\mathcal{A}}(t)\ \mathrm{d}t, \tag{14}$$

where the second equality follows from our assumption of integrability.

Let $uv = w \in \Sigma^\omega$ and $f_{\mathrm{INT}(\mathcal{A}),u}(v)$ be the function the automaton $\mathrm{INT}(\mathcal{A})$ computes with initial distribution $P_{\mathcal{A}}(u)$ on input v. Then

$$f_{\mathrm{INT}(\mathcal{A}),u}(v) = 2^{|u|}\int_{\widehat{u0^\omega}}^{\widehat{u1^\omega}} \widehat{f}_{\mathcal{A}}(t)\ \mathrm{d}t, \tag{15}$$

as only words containing the prefix u contribute to the sum in (14).

We apply the transducer $\mathcal{T}_{\mathrm{R}}$ in Figure 7 to the automaton $\mathcal{A}$. The right-hand side state of $\mathcal{T}_{\mathrm{R}}$ works analogically to the transducer INT earlier, except its initial distribution component is 0. Let $w = a_1a_2\ldots$ and its prefix of length i be $w_i = a_1\ldots a_i$. Then, for $1 \le j \le i$, we have $w_{j-1}a_jv = w_i$. The left-hand side state of the construction now computes with input

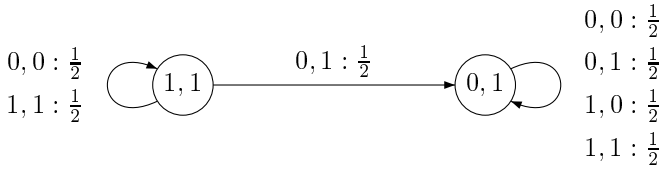


Fig. 7. A WFT $\mathcal{T}_{\mathrm{R}}$ computing $\int_0^x \widehat{f}(t)\ \mathrm{d}t$.

w_i the distribution $\frac{1}{2^i}P_{\mathcal{A}}(w_i)$, which is needed for (15). The coefficients $\frac{1}{2^i}$ and $2^{|u|}$ cancel each other out leaving us only the integral. With input $w_i 0^\omega$ the automaton $\mathcal{T}_{\mathrm{R}}(\mathcal{A})$ then, for every $a_j = 1$, flips a_j into 0 and adds to the computation in the right-hand side of the construction the function

$$\frac{1}{2^j} f_{\mathrm{INT}(\mathcal{A}),w_{j-1}0}(v0^\omega) = \int_{\widehat{w_{j-1}0^\omega}}^{\widehat{w_{j-1}01^\omega}} \widehat{f}_{\mathcal{A}}(t)\ \mathrm{d}t,$$

thus adding the interval $[\widehat{w_{j-1}0^\omega}, \widehat{w_{j-1}01^\omega})$ to the integral. To put this in other terms, for every $a_j = 1$ it is computing the average of $f_{\mathcal{A}}$ over every infinite input word lexicographically between $w_{j-1}0^\omega$ and $w_{j-1}01^\omega$. Overall, the construction computes

$$\begin{aligned} f_{\mathcal{T}_{\mathrm{R}}(\mathcal{A})}(w_i 0^\omega) &= \lim_{n\to\infty} \frac{1}{2^n} F_{\mathcal{A}}(w_i 0^n) + \sum_{w_{j-1}1 \in \mathrm{pref}(w_i)} \frac{1}{2^{|w_{j-1}1|}} f_{\mathrm{INT}(\mathcal{A}),w_{j-1}0}(v0^\omega) \\ &= \int_0^{\widehat{w_i 0^\omega}} \widehat{f}_{\mathcal{A}}(t)\ \mathrm{d}t \end{aligned}$$

and finally, with $w = uav$ and $a \in \Sigma$,

$$f_{\mathcal{T}_{\mathrm{R}}(\mathcal{A})}(w) = \lim_{n\to\infty} \frac{1}{2^n} f_{\mathcal{A}}(w) + \sum_{u1 \in \mathrm{pref}(w)} \frac{1}{2^{|u1|}} f_{\mathrm{INT}(\mathcal{A}),u0}(v) = \int_0^{\widehat{w}} \widehat{f}_{\mathcal{A}}(t)\ \mathrm{d}t.$$

It follows that $\mathcal{A}_{\mathrm{R}} = \mathcal{T}_{\mathrm{R}}(\mathcal{A})$. □

Finite automata typically compute fractal type of functions. If they are Riemann-integrable, we can apply the previous process. The integral of the function in Figure 4, computed by an automaton $\mathcal{A}_{\mathrm{R}}(t)$, is shown in Figure 8.

In above the integration was done within the framework of general weighted transducers with a uniform representation. With some additional

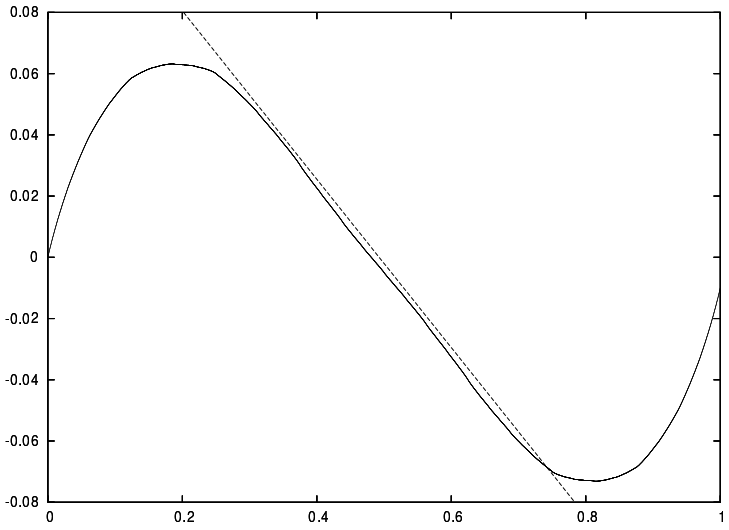


Fig. 8. Graph of the function $\int_0^x \widehat{f}_{\mathcal{A}\left(\frac{2}{3}\right)}(t)\ \mathrm{d}t$. The dashed linear function is added for reference.

considerations the result can extend also to other WFA than level automata, as the level structure is only needed to guarantee the finiteness of the values of the original automaton. The process always duplicates the number of states. However, at least for level automata, this can be done straightforwardly and even without the assumption of existence of the integral, as shown in [2]. Moreover, only one additional state is needed.

Theorem 12. *For each level automaton $\mathcal{A}$ of degree n there exists a level automaton $\mathcal{A}_{\mathrm{I}}$ of degree $n+1$, such that*

$$\widehat{f}_{\mathcal{A}_{\mathrm{I}}}(x) = \int_0^x \widehat{f}_{\mathcal{A}}(t)\ \mathrm{d}t. \tag{16}$$

Moreover, given the automaton $\mathcal{A}$ the automaton $\mathcal{A}_{\mathrm{I}}$ can be constructed by uniform rules. We first find an average preserving level automaton $\mathcal{A}_{ap}$ of the same degree defining the same function. Next, we define a level automaton $\mathcal{A}_{\mathrm{I}}$ of degree $n+1$ as follows:

(i) *Its states are those of $\mathcal{A}_{ap}$ with an added state t.*
(ii) *Its transitions contain those of $\mathcal{A}_{ap}$ with weights divided by 2.*

Fig. 9. A one state WFA $\mathcal{A}_0$ computing a constant function.

(iii) *From each original state q of $\mathcal{A}_{ap}$ there is a transition in $\mathcal{A}_1$ to the state t labeled by 1 and with the weight $\frac{1}{2}\sum_{p\in Q} W_0(q,p)$, where W_0 is the weight function of $\mathcal{A}_{ap}$ and Q its set of states.*

(iv) *The state t contains loops of weight 1 with both labels 0 and 1.*

The usefulness of the theorem is illustrated by the fact that it alone yields automata representations for every polynomial.

Example 2. We begin with the simple one state automaton $\mathcal{A}_0$ in Figure 9. Given that both inputs 0 and 1 have weight 1 attached to them, the function computed is the same, $I \cdot F$, for every input word. Fixing $F = (1)$ and $I = (a_0)$, we have

$$\widehat{f}_{\mathcal{A}_0}(x) = a_0,$$

the constant function. For now, let us fix $a_0 = 1$.

We then apply the process of Theorem 12 to $\mathcal{A}_0$. It is average preserving and thus $\mathcal{A}_{\mathrm{ap}} = \mathcal{A}_0$. The result is the automaton $\mathcal{A}_1$ in Figure 10, with $I = (1, a_1)$ and $F = (0, 1)$, which is not average preserving. It can be modified, however, so that it computes the same function and is average preserving. Fixing $a_1 = 0$ and applying Theorem 12 to the result we then receive the second automaton in Figure 10, with $I = (\frac{1}{2}, 0, a_2)$ and $F = (0, 0, 1)$. By (16) we have

$$\widehat{f}_{\mathcal{A}_1}(x) = \int_0^x 1 \, \mathrm{d}t = x, \quad \widehat{f}_{\mathcal{A}_2}(x) = \int_0^x t \, \mathrm{d}t = \frac{1}{2}x^2.$$

With the initial distribution $I = (1, 0, 0)$ the automaton $\mathcal{A}_2$ then computes $\widehat{f}_{\mathcal{A}_2}(x) = x^2$.

Looking at the three automata we notice that the automaton $\mathcal{A}_0$ is contained in $\mathcal{A}_1$ at the state 1. Also, the automaton $\mathcal{A}_1$ is contained in $\mathcal{A}_2$

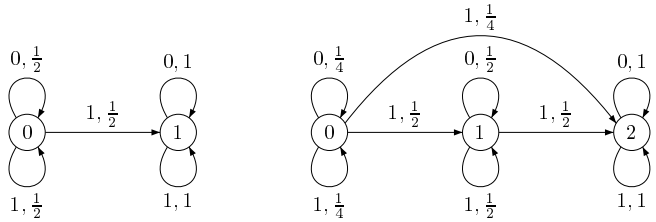


Fig. 10. A two state WFA $\mathcal{A}_1$ computing a linear function and a three state WFA $\mathcal{A}_2$ computing a quadratic function.

at states 1 and 2. We can then compute

$$\widehat{f}_{\mathcal{A}_0}(b_0) + \widehat{f}_{\mathcal{A}_1}(b_1) + \widehat{f}_{\mathcal{A}_2}(b_2) = b_2x^2 + b_1x + b_0$$

using only the automaton $\mathcal{A}_2$ with the initial distribution (b_2, b_1, b_0). Continuing further we can acquire an automaton for any at most nth degree polynomial with $n+1$ states.

Acknowledgements

The author is grateful to professors Juhani Karhumäki and Jarkko Kari for helpful discussions and suggestions.

References

1. K. Culik II, I. Fris, *Weighted finite transducers in image processing.* — Discrete Applied Mathematics **58**, No. 3 (1995), 223–237;
2. K.Culik II, J. Karhumäki, *Finite automata computing real functions.* — SIAM J. Comput. **23** (1994), 789–814.
3. K. Culik, II, J. Kari, *Finite state transformation of images.* — Computers & Graphics **20**, No. 1 (1996), 125–135.
4. D. Derencourt, Juhani Karhumäki, M. Latteux, A. Terlutte, *On continuous functions computed by finite automata.* — RAIRO-Theor. Inf. Appl. **29** (1994), 387–403.
5. J. Karhumäki, J. Kari, *Finite automata, image manipulation and automatic real functions.* — In: Handbook of Automata, European Mathematical Society, (to appear).
6. J. Karhumäki, T. Sallinen, *Weighted finite automata: computing with different topologies.* — In: Calude, Cristian and Kari, Jarkko and Petre, Ion and Rozenberg, Grzegorz (eds.), Unconventional Computation, Lect. Notes Computer Sci. **6714** (2011), 14–33.
7. T. Sallinen, *Reaalifunktioiden laskennasta automaateilla.* — University of Turku (2009).

8. M. P. Schützenberger, *On the definition of a family of automata.* — Information and Control **4** (1961), 245–270.

Department of Mathematics
and Turku Centre for Computer Science (TUCS)
University of Turku
FIN-20014 Turku
Finland
E-mail: thtsal@utu.fi

Поступило 7 сентября 2012 г.