Real Functions computed by Parametric Weighted Finite Automata

German Tischler

Universität Würzburg, Lehrstuhl für Informatik II, Am Hubland,
97074 Würzburg, Germany
tischler@informatik.uni-wuerzburg.de
http://www2.informatik.uni-wuerzburg.de/staff/tischler

Abstract. Weighted finite automata (WFA) are an extension of finite automata that can be used to compute real functions and parametric WFA (PWFA) are a multidimensional variant of WFA. This paper shows that each PWFA computable set is computable by a PWFA that has an alphabet cardinality of two. Further we show that real polynomials with solely rational coefficients can be computed by PWFA that do not use non-rational numbers. We study a class of PWFA that behaves like WFA on mirrored input words and obtain the result that the only smooth real functions definable by PWFA in this family are polynomials.

1 Introduction

A first definition of weighted finite automata (WFA) was given in [5]. A constructive proof that these automata can be used to display polynomials on the unit interval can be found in [3]. The statement that the only smooth (that means derivable infinitely often everywhere in the unit interval) functions computable by WFA are polynomials was first formulated in [4]. There is a new proof of this result by Jarkko Kari et al. The result is used in section 4 but independent of the method that is used to prove it. A formal definition of WFA is:

A WFA X is a quintuple (Q, Σ, W, I, F) where

1. Q is a finite set of states
2. Σ is a finite alphabet
3. $W : Q \times \Sigma \times Q \mapsto \mathbb{R}$ is a weight function
4. $I : Q \mapsto \mathbb{R}$ is the initial distribution
5. $F : Q \mapsto \mathbb{R}$ is the final distribution.

Usually the weight function and initial and final distribution are written in matrix and vector form. For each $a \in \Sigma$ there is a transition matrix A_a where $A_a(i, j) = W(i, a, j)$. I is understood as a row vector and F as a column vector. The real value $f_X(w)$ that X computes for an input word $w = w_1 w_2 \dots w_k \in \Sigma_X^*$ is

$$f_X(w) = I \prod_{i=1}^{k} A_{w_i} F. \tag{1}$$

The word $w = w_1 w_2 \ldots \in \{0,1\}^\omega$ is interpreted as the real number

$$\mathrm{bin}(w) = \sum_{i=1}^{\infty} w_i 2^{-i}. \tag{2}$$

The extension of f_X to words of infinite length $w = w_1 w_2 \ldots \in \Sigma^\omega$ is by using a limit construction

$$f_X(w) = \lim_{n \to \infty} f_X(w_1 w_2 \ldots w_n). \tag{3}$$

If this limit does not exist, the function f_X is not defined for w.

Parametric weighted finite automata (PWFA) were first defined in [1] as a multidimensional variant of WFA. A formal definition is: A PWFA X is a quintuple (Q, Σ, W, I, F) where

1. Q is a finite set of states
2. $\Sigma = \{0, 1, \ldots, l-1\}$ is a finite alphabet
3. $W = (A_0, A_1, \ldots, A_{|\Sigma|-1}), A_i \in \mathbb{R}^{|Q| \times |Q|}$ are the transition matrices
4. $I = (I_0, I_1, \ldots, I_{d-1}), I_i \in \mathbb{R}^{|Q|}$ is a set of d initial distributions, the I_i are the rows of the matrix I
5. $F \in \mathbb{R}^{|Q|}$ is the final distribution.

The point that the automaton computes on a word $w = w_1 w_2 \ldots w_n \in \Sigma^*$ is

$$f_X(w) = I \prod_{i=1}^{n} A_{w_i} F \tag{4}$$

and the computed set $S(X)$ is

$$S(X) = \bigcap_{n=0}^{\infty} \overline{S_{\geq n}(X)} \tag{5}$$

where

$$S_{\geq n}(X) = \bigcup_{i=n}^{\infty} S_i(X) \tag{6}$$

and

$$S_n(X) = \{f_X(w) | w \in \Sigma^n\} \tag{7}$$

where the overline notation in (5) denotes the topological closure of the set under the line. Thus a vector x is in $S(X)$ if and only if there is a sequence $v_1, v_2, \ldots$ of words of increasing length such that $x = \lim_{i \to \infty} f_X(v_i)$. The transition relation $T_X \subseteq Q_X \times \Sigma_X \times Q_X \times \mathbb{R}$ of a PWFA X is defined as

$$T_X = \{(p_0, l, p_1, w) \,|p_0, p_1 \in Q_X, l \in \Sigma_X, w = A_{X_l}(p_0, p_1)\} \ . \tag{8}$$

We can understand WFA as a subset of PWFA in the sense that every WFA computable function f can be computed as the set

$$\{(x, f(x)) | x \in [0,1]\}. \tag{9}$$

PWFA are known to be able to compute non-polynomial functions, e.g. the exponential, sine and cosine functions and their inverse functions ([1]), some rational functions and real polynomials on $\mathbb{R}$ ([6]). It was also shown in [6] that the set of PWFA computable sets is closed under set union and invertible affine transformation.

In section 2 of this paper we show that every PWFA computable set can be computed by a PWFA that has an alphabet cardinality of two. Section 3 provides a construction method for PWFA computing real polynomials on $\mathbb{R}$ without the need for non-rational weights if the polynomial to be displayed has only rational coefficients. In section 4 we consider a family of PWFA that use a certain family of hyperbolic iterated function systems (see [2] for reference) for computing their first result component and show that the only smooth real functions computable by automata in this family are polynomials.

2 PWFA alphabet cardinality

We examine the influence of the alphabet cardinality on PWFA computability of sets. No PWFA computable set requires a PWFA with an alphabet cardinality greater than 2 to be displayed. The proof is based on the following lemma.

Lemma 1. *Let X be a PWFA with $|\Sigma_X| = 2^{k+2}$ for some $k \in \mathbb{N}$. Then there is a PWFA Y with $S(Y) = S(X)$ and $|\Sigma_Y| = 2^{k+1}$.*

Proof. Y can be constructed as follows:

- $Q_Y = \{0, 1, \ldots, 3|Q_X| - 1\}$
- $\Sigma_Y = \{0, 1, \ldots, |\Sigma_X|/2 - 1\}$
- The first $|Q_X|$ rows and columns of I_Y are I_X, the rest is filled with zero elements.
- $(F_Y)^T$ is $({F_X}^T\ {F_X}^T\ {F_X}^T)$

For every state $q \in Q_X$ we introduce two new states $q' = |Q_X| + q$ and $q'' = 2|Q_X| + q$. The edges in Y are defined in statements 10 and 11.

$$q \in Q_X \Rightarrow \begin{array}{l} \{(q, l, q', 1) | l \in \{0, 1, \ldots, 2^k - 1\}\} \subseteq T_Y \text{ and} \\ \{(q, l, q'', 1) | l \in \{2^k, \ldots, 2^{k+1} - 1\}\} \subseteq T_Y \end{array} \tag{10}$$

and

$$(q, l, p, w) \in T_X \Rightarrow \begin{array}{ll} (q', \lfloor l/2 \rfloor, p, w) \in T_Y & \text{if } l \bmod 2 = 0 \text{ and} \\ (q'', \lfloor l/2 \rfloor, p, w) \in T_Y & \text{if } l \bmod 2 = 1. \end{array} \tag{11}$$

Define

$$g(a, m) = (a \bmod 2)2^k + m \text{ for } a \in \Sigma_X, m \in \mathbb{N} \tag{12}$$

and

$$h(a) = \left\lfloor \frac{a}{2} \right\rfloor \text{ for } a \in \Sigma_X. \tag{13}$$

Let $w_1 w_2 \ldots w_m \in {\Sigma_X}^*$, then

$$f_X(w_1 w_2 \ldots w_m) = f_Y\left(g(w_1, i)h(w_1)g(w_2, i)h(w_2) \ldots g(w_m, i)h(w_m)\right) \tag{14}$$

for $i \in \{0, 1, \ldots 2^k - 1\}$. (14) describes the behavior of Y for all input words of Y. Y produces the same set of output vectors as X. The number of words that produce the vector v as an output of X is finite iff this is also true for Y. Thus $S(Y) = S(X)$. □

It is clear that for every PWFA with alphabet cardinality k there is a PWFA of alphabet cardinality $k+1$ that computes the same set. The transition matrix of the new symbol can be chosen to be any of the matrices already present in the PWFA. We can thus assume that every PWFA computable set is computable by a PWFA that has an alphabet cardinality that is a power of 2. So the following statement holds.

Theorem 1. *Let X be a PWFA. There is a PWFA Y of alphabet cardinality 2 with $S(Y) = S(X)$.*

3 Real Polynomials computed by PWFA

WFA are known to be able to compute polynomials on the unit interval. In [6] two ways were presented for computing real polynomials on $\mathbb{R}$. Both of these have in common that they use at least one weight that is not rational, even if the polynomial to be displayed has only rational coefficients, so from a certain point of view they are not finite automata. In this section we show that there are PWFA computing polynomials on $\mathbb{R}$ without using non-rational weights. We start by providing a PWFA that computes a polynomial p on $\mathbb{N}$.

Lemma 2. *Let $p(x) = x^k$ with $k \in \mathbb{N}$ for $x \in \mathbb{R}$. There is a PWFA X that computes the set $S(X) = \overline{\{(n, n^k) \,|n \in \mathbb{N}\}}$.*

Proof. We construct a PWFA X' that has the alphabet $\Sigma_X = \{0, 1, 2\}$. Let $c(w)$ denote the word that is obtained from $w \in \Sigma_X^*$ by deleting all occurrences of the symbol 2. For each input word w we interpret the word $c(w) = b_1 b_2 \ldots b_m$ as the natural number $\mathrm{nat}_2(c(w)) = \sum_{i=1}^{m} b_i 2^i$. Let $x_1 = b_1 b_2 \ldots b_m \in \{0, 1\}^*$ and $x_2 = b_2 b_3 \ldots b_m$. Then

$$\begin{aligned} {x_1}^k &= (2b_1 + 2x_2)^k \\ &= 2^k \left(b_1 + x_2\right)^k \\ &= 2^k \textstyle\sum_{i=0}^{k} \binom{k}{i} b_1 x_2^i \end{aligned} \tag{15}$$

that means

$${x_1}^k = \begin{cases} 2^k x_2^k & \text{for } \; b_1 = 0 \\ \sum_{i=0}^{k} 2^k \binom{k}{i} x_2^i & \text{for } \; b_1 = 1 \end{cases}. \tag{16}$$

So the construction of X' follows the scheme shown in figure 1. The label 2 is used to give the automaton the possibility to keep the current result and produce it infinitely often, so the set computed by the automaton is not empty, as it would be, if every point were produced only once. X' produces the set

$$\overline{\{(\mathrm{nat}_2(c(w)), \mathrm{nat}_2(c(w))^k) | w \in \Sigma_X^*\}} = \{(n, n^k) | n \in 2\mathbb{N}\} \tag{17}$$

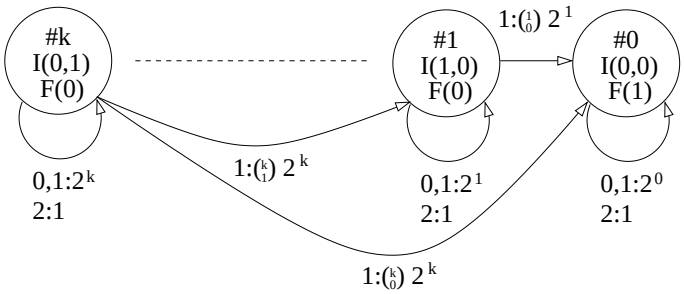

Fig. 1. Polynomial automaton X' computing set $\overline{\{(n, n^k)|n \in 2\mathbb{N}\}}$

by construction. As $\left(2n, (2n)^k\right) = (2n, 2^k n^k)$, the automaton X can be obtained by applying the invertible affine transformation $\begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2^k} \end{pmatrix}$ to X'. □

The proof provided for lemma 2 is based on that given for the construction of WFA computing polynomials in [3]. As WFA are closed under sum and multiplication by a scalar, every real polynomial can be computed by a PWFA on $\mathbb{N}$. The construction given in lemma 2 can be extended to computing real polynomials on $\overline{\mathbb{R}}$. Firstly we show this for $\overline{\mathbb{R}_0^+}$.

Lemma 3. *Let $p(x) = x^k$ be a real function with $k \in \mathbb{N}$. There is a PWFA Z that computes the set $\overline{\{(x, p(x))|x \in \mathbb{R}, x \geq 0\}}$*

Proof. The automaton Z is constructed by combining two automata. The first is an automaton computing the set $\{x, p(x))|x \in [0; 1]\}$ by using the standard WFA construction for polynomials, let the name of this subautomaton be Y. The second is the automaton X as defined for $p(x) = x^k$ in the proof of lemma 2 but stripping away label 2 and setting its final distribution to zero. We remap X so it uses label 2 instead of 0 and 3 instead of 1. If Y makes a transition, we want X not to change its configuration, so for every state $q \in Q_X$ we introduce the edges $(q, 0, q, 1)$ and $(q, 1, q, 1)$. Both X and Y have states for x^0 to x^k in their context, let them be denoted by x_X^0 to x_X^k and x_Y^0 to x_Y^k. For every transition (x_X^i, l, x_X^j, v) where $l \in \{2, 3\}$ we add a transition (x_X^i, l, x_Y^j, v). The complete scheme is shown in figure 2. Let $\text{bin}(w) = \sum_{i=1}^{m} w_i 2^{-i}$ for $w = w_1 w_2 \dots w_m \in (0|1)^*$. We observe the behavior of Z for two different classes of input words w:

1. $w \in (0|1)^*$ that means there is no occurrence of the symbols 2 and 3 in w. Then $f_Z(w) = (\text{bin}(w), \text{bin}(w)^k)$ as for usual WFA, thus $\{(x, x^k)|x \in [0; 1]\} \subset S(Z)$.
2. $w \in \Sigma_Z^*(2|3)(0|1)^*$ that means there is at least one occurrence of the symbol 2 or 3 in w. As the configuration of Y is overwritten by that of X by every occurrence of the symbols 2 and 3, any occurrence of the symbols 0 and 1 before the last 2 or 3 does not have any effect on the result of the computation. Let $w' \in (2|3)^*(2|3)(0|1)^*$ be the word w with all occurrences of 0 and 1 before

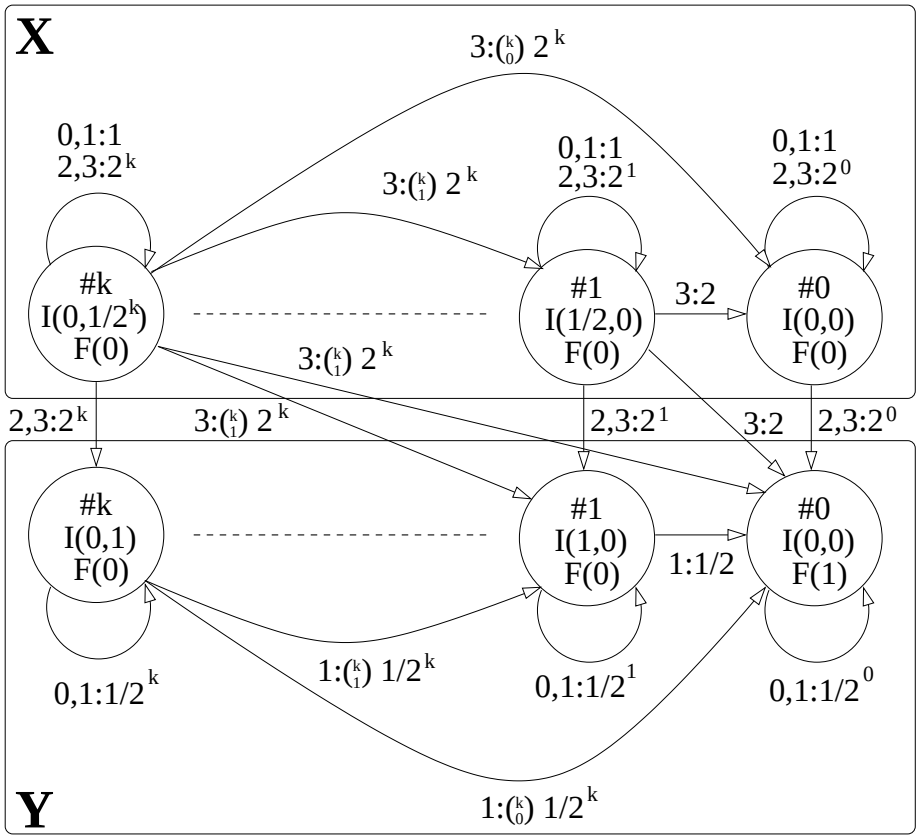

Fig. 2. Polynomial automaton Z that computes the set $S(Z) = \overline{\{x, x^k | x \in \mathbb{R}_0^+\}}$

the last 2 or 3 erased. Let $\mathrm{nat}_1(v) = \sum_{i=1}^{m}(v_i - 2)2^{i-1}$ for $v = v_1 v_2 \dots v_m \in \{2,3\}^*$. After reading the prefix of maximal length p of w in $(2|3)^*(2|3)$, Y is initialized to compute the set $\{(x + \mathrm{nat}_1(p), (x + \mathrm{nat}_1(p))^k) | x \in [0,1]\}$. As $\mathrm{nat}_1(p)$ can be any natural number, the computed set is

$$\overline{\bigcup_{i=0}^{\infty}\{(x+i,(x+i)^k)|x \in [0,1]\}} \subseteq S(Z). \tag{18}$$

As the output for class 2 contains the output for class 1, the proof is completed. □

Again, as WFA are closed under sum and multiplication by a scalar we can obtain PWFA computing the set $\overline{\{(x, p(x)) | x \in \mathbb{R}_0^+\}}$ where p is an arbitrary real polynomial. An automaton computing the set $\{(x, p(x)) | x \in \mathbb{R}\}$ for the same polynomial can be constructed by using the set union construction presented in

[6] to build an automaton Z that computes the set

$$S(Z) = \overline{\{(x, p(x))|x \in \mathbb{R}_0^+\} \cup \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \{(x, p(-x))|x \in \mathbb{R}_0^+\}} \tag{19}$$

implying proposition 1.

Proposition 1. *Let $p(x)$ be a real polynomial with $k \in \mathbb{N}$. There is a PWFA Z that computes the set $\overline{\{(x, p(x))|x \in \mathbb{R}\}}$.*

Let $p(x) = \sum_{i=0}^{k} a_i x^i$ be a real polynomial. Let Z be the PWFA constructed above to compute p on $\mathbb{R}$. All weights in Z are natural multiples of an integer power of 2. The initial and final distribution contain products of the coefficients a_i and integer powers of 2. Thus if $a_i \in \mathbb{Q}$ for $i = 0, 1, \ldots, k$, then Z does not contain any non-rational numbers.

4 WFA input reversion

In this section we consider the set of smooth functions computable by a certain family of PWFA on the unit interval. Let $d \in (0, 1)$. Then we define the iterated function system (IFS, see e.g. [2] for reference) D as the set of functions

$$\begin{aligned} d_0(x) &= dx \\ d_1(x) &= dx + (1 - d) \ . \end{aligned} \tag{20}$$

The system D is hyperbolic and has the attractor $[0, 1]$. We first show that polynomials on the unit interval can be computed while using this IFS for the first component of a PWFA.

Lemma 4. *Let $p(x) = x^k$ be a real function for $k \in \mathbb{N}$. There is a PWFA X that computes the set $\{(x, p(x))|x \in [0, 1]\}$ while the first component of X is computed as the IFS D.*

Proof. The automaton for the IFS D is shown in figure 3. The recursion for x^k

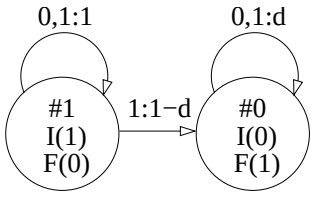

Fig. 3. IFS automaton for system D

can be written as

$$\begin{aligned} e_0(x_0^k) &\mapsto (dx_0)^k = d^k x_0^k \\ e_1(x_0^k) &\mapsto (dx_0 + (1 - d))^k = \textstyle\sum_{i=0}^{k} \binom{k}{i} d^i x_0^i (1 - d)^{k-i} \ . \end{aligned} \tag{21}$$

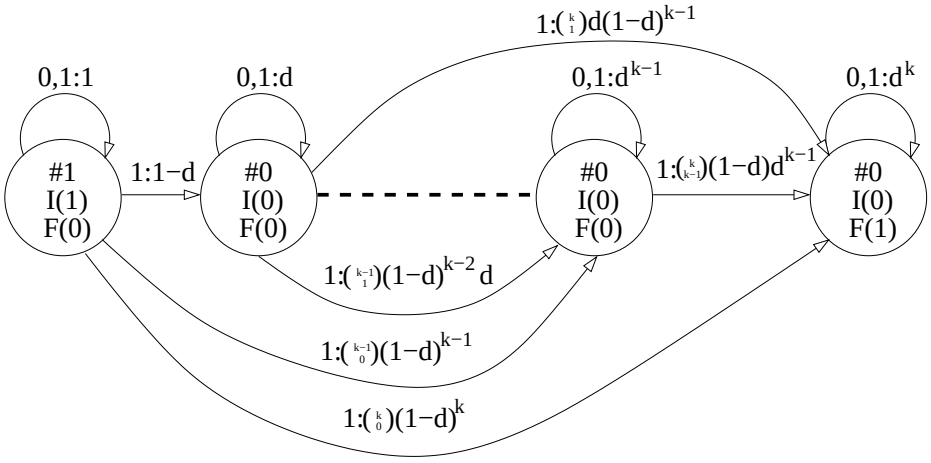

Fig. 4. x^k automaton for first component computed as the IFS D

The construction scheme is shown in figure 4. Thus the automaton X can be built by combining the automata in figure 3 for the first component and 4 for the second. □

As WFA are closed under sum and multiplication by a constant, we have the following.

Theorem 2. *Let $p(x) = \sum_{i=0}^{k} a_i x^i$ a real polynomial for $k \in \mathbb{N}$. For each $d \in (0,1)$ there is a PWFA X that computes the set $\{(x, p(x)) | x \in [0,1]\}$ while computing the first component as the IFS D.*

It is remarkable that unlike the usual WFA construction for polynomials we cannot use one single line-automaton to display a complete polynomial with more than one non-zero coefficient. Clearly the construction for $x^{k+1}, k \in \mathbb{N}$ contains the construction for x^k as a subgraph. It differs in two features:

1. The subautomaton for x^k contained in x^{k+1} has outgoing edges.
2. Building the automaton for x^{k+1} from x^k the final distribution is changed instead of the initial distribution.

We now show that polynomials are the only smooth (that means having all derivates everywhere on the unit interval) real functions computable by PWFA using the IFS D for computing the first component if $d = \frac{1}{2}$. This is done by reducing such automata to WFA. We assume that the proof for the statement that the only smooth functions computable by WFA are real polynomials would also hold for a modified definition of WFA where the splitting of the unit interval is not performed at the point $\frac{1}{2}$ but at some $d \in (0,1)$.

Lemma 5. *Let X be a PWFA computing the smooth real function f on the unit interval as the set $S(X) = \{(x, f(x)) | x \in [0,1]\}$ while computing the first*

component as the IFS D with $d = \frac{1}{2}$*. There is a WFA Y that computes the same function.*

Proof. We assume without loss of generality that there is an upper bound to the norms of all vectors produced by computations of X. Let $w = w_1 w_2 \dots w_m \in \Sigma_X^*$. The first component of the result vectors of X is computed as

$$\begin{aligned} x(w) &= \tfrac{1}{2}\left(\left(\dots \tfrac{1}{2}\left(\tfrac{1}{2}w_1\right) + \tfrac{1}{2}w_2 \dots\right) + \tfrac{1}{2}w_{m-1}\right) + \tfrac{1}{2}w_m \\ &= \textstyle\sum_{i=1}^{m} \frac{1}{2}^{m-i+1} w_i \ . \end{aligned} \tag{22}$$

That means the input order is reversed in comparison to WFA where the function

$$\mathrm{bin}(w) = \sum_{i=1}^{m} \frac{1}{2}^{i} w_i \tag{23}$$

is used for the interpretation of input words as abscissae. Let $r(w)$ denote the reversed word of w. Apparently $x(r(w)) = \mathrm{bin}(w)$ and $\mathrm{bin}(r(w)) = x(w)$ for w in Σ_X^*. Without loss of generality we assume that the set of states X uses to compute its first component is disjoint from the set of states it uses to compute the second. Then we can decompose X into two independent WFA computing the components of the result vectors of X. Let these subgraphs of X be named Y' for the first component and Y'' for the second. Then equation 24 holds for $w = w_1 w_2 \dots w_m \in \Sigma_X^*$.

$$\begin{aligned} (x(w), f(x(w))) &= (I_{Y'} \textstyle\prod_{i=1}^{m} A_{Y'_{w_i}} F_{Y'}, I_{Y''} \prod_{i=1}^{m} A_{Y''_{w_i}} F_{Y''}) \\ &= \left(I_{Y'} \left(\textstyle\prod_{i=1}^{m} A^T_{Y'_{w_{m-i+1}}}\right)^T F_{Y'}, I_{Y''} \left(\prod_{i=1}^{m} A^T_{Y''_{w_{m-i+1}}}\right)^T F_{Y''}\right) \\ &= \left(F^T_{Y'} \left(\textstyle\prod_{i=1}^{m} A^T_{Y'_{w_{m-i+1}}}\right) I^T_{Y'}, F^T_{Y''} \left(\prod_{i=1}^{m} A^T_{Y''_{w_{m-i+1}}}\right) I^T_{Y''}\right) \\ &= \left(\mathrm{bin}(r(w)), F^T_{Y''} \left(\textstyle\prod_{i=1}^{m} A^T_{Y''_{w_{m-i+1}}}\right) I^T_{Y''}\right) \end{aligned} \tag{24}$$

So we obtain the WFA Y from Y'' by transposing the transition matrices of Y'' and swapping the initial and final distribution of Y''. Possible differences due to the different definitions of convergence for WFA and PWFA are ruled out here, because f is

- smooth (especially continuous) and
- a function.

□

Every smooth real function computable by a WFA is a polynomial, so proposition 2 follows.

Proposition 2. *Let X be a PWFA computing the smooth real function f on the unit interval as the set $S(X) = \{(x, f(x)) | x \in [0,1]\}$ while computing the first component as the IFS D with $d = \frac{1}{2}$. Then f is a polynomial.*

5 Conclusion

In this paper we showed that every PWFA computable set is computable by a PWFA that has an alphabet cardinality of 2. We further showed that real polynomials on $\mathbb{R}$ with solely rational coefficients can be displayed by PWFA using only rational weights and rational initial and final distributions. PWFA do not gain computational power regarding the representability of smooth real functions if the usual interpretation of words for WFA as real numbers happens on reversed words.

Acknowledgment

We thank Paula Steinby for some early discussions on the proof of lemma 5 and the two referees for their valuable comments on a previous version of this paper.

References

1. J. Albert, J. Kari, *Parametric Weighted Finite Automata and Iterated Function Systems* Proceedings L'Ingenieur et les Fractales - Fractals in Engineering, Delft, 248-255, 1999.
2. M. Barnsley, *Fractals Everywhere*, 2nd ed. Boston, Academic Press, 1993.
3. K. Culik II, J. Karhumäki, *Finite automata computing real functions* SIAM Journal on Computing 23/4, 789-814, 1994.
4. D. Derencourt, J. Karhumäki, M. Latteux, A. Terlutte, *On Computational Power of Weighted Finite Automata* Lecture Notes in Computer Science 629, 236-245, 1992.
5. M. P. Schützenberger, *On the definition of a family of automata* Information and computation 4, 245-270, 1961.
6. G. Tischler, *Properties and applications of parametric weighted finite automata* submitted.