Automata, Boolean Matrices, and Ultimate Periodicity

Guo-Qiang Zhang
Department of Computer Science
University of Georgia
Athens, Georgia 30602, U. S. A.
Phone: 706-542-3444 Fax: 706-542-2966 Email:gqz@cs.uga.edu

Abstract. This paper presents a Boolean-matrix based method to automata theory, with an application to the study of regularity-preserving functions. A new characterization of such functions is derived in terms of their ultimate periodicity with respect to powers of Boolean matrices. This characterization reveals the intrinsic algebraic nature of regularity-preserving functions. It facilitates a concise proof of known, as well as previously unknown properties of regularity-preserving functions, leading to the solution of the "subtraction problem" left open by Kosaraju.

1 Introduction

Finite automata are one of the most widely used mathematical structures in computer science. Since its inception [10], automata theory has been dominated by a combinatorial approach, grounded on the intuitive notions of machines and states [20, 18, 8, 15]. During the last two decades, however, there have been significant developments in an algebraic approach (see, e.g., [6, 7, 13, 17, 19]). The algebraic approach uses ideas from linear algebra to specify a language via a system of linear equations over a semiring. The beauty of this approach comes from its higher level of abstraction: the definitions are concise, proofs elegant, and more appropriate treatments can be provided to advanced topics about formal languages, such as the inherent ambiguity of context-free languages [17].

This paper presents a Boolean *matrix approach to automata theory* (abbreviated as MAAT) which combines the style of the combinatorial approach with that of the algebraic approach. MAAT is based on the observation that many problems in the area can be reduced to the construction of a new automaton from a given one. To construct an automaton, one can use states of the form (p, Γ), where p is an original state, and Γ is a tuple of adjacency matrices determined by the original automaton. The desired transition function can be defined in terms of straightforward matrix operations, and the correctness of the construction can often be proved by induction.

This way of constructing an automaton is consistent with the traditional concept of machines and states; the novelty lies in the importation of matrices to states, rendering greater expressiveness in the specification of transition functions and final states. By taking advantage of the intuition provided by the combinatorial approach together with the conciseness and mathematical elegance brought about by the algebraic approach, we obtain a method which will be demonstrated in this paper to be well suited for tackling problems about regularity-preserving functions.

A function f on non-negative integers is called *regularity-preserving* if for any regular language L, the language $\mathsf{T}(L, f)$ is also regular, where

$$\mathsf{T}(L, f) := \{x \mid \exists y \ \ |y| = f(|x|) \ \& \ xy \in L\}.$$

The study of regularity-preserving functions dates back to the early days of automata theory (see Stearns and Hartmanis [24]). The characterization of Seiferas and McNaughton [22] shows that this study does not merely serve the purpose of satisfying our intellectual curiosity; they are closely related to semilinear sets, which play an important role in the study of monadic second order logic [4, 23], of Presburger arithmetic, and recently, of learning theory [1].

An important contribution of MAAT is a new characterization theorem (Theorem 4.1), stating that a function is regularity-preserving if and only if it is ultimately periodic with respect to powers of Boolean matrices. This result facilitates a concise proof of known, as well as many previously unknown, properties of regularity-preserving functions. In addition to proving all the structural properties suggested by Kosaraju [11], we also prove the following new properties of regularity-preserving functions (Theorem 5.1):

- if f is regularity-preserving then so is the "sum function" $\lambda x.\ \Sigma_{i \leq x} f(i)$;
- if f is regularity-preserving then so is the "product function" $\lambda x.\ \Pi_{i \leq x} f(i)$.

MAAT is equally capable of treating functions associated with other language operators not considered before, such as

- $\mathsf{P}(L, f) := \{x \mid x^{f(|x|)} \in L\}$, and
- $\mathsf{Q}(L, f) := \{a_0 a_1 a_2 \cdots a_n \mid a_0^{f(0)} a_1^{f(1)} a_2^{f(2)} \cdots a_n^{f(n)} \in L\}$.

These operators are more sophisticated, because they are not only length-related, but also pattern-related. We will capture regularity-preserving functions with respect to these operators (see Section 5.2).

In studying regularity-preserving functions under various language operators mentioned above, it is important to differentiate languages over a singleton alphabet from those in general. We say, for example, that a function is T_1-regularity-preserving if it preserves regularity for languages over the singleton alphabet under T. Similarly, a function is T-regularity-preserving if it preserves regularity for languages under T in general. We prove that a function is T_1-regularity-preserving if and only if it is ultimately periodic with respect to powers of *permutation matrices*, a property not shared by T-regularity-preserving functions. A careful analysis leads to an example showing that (T-) regularity-preserving functions are not preserved by subtraction. This is a property claimed to hold in [11], but the claim was soon withdrawn [12]. Our example finally settles this issue.

1.1 Related work

A recent paper of Kozen [14] contains a lucid exposition to regularity-preserving functions. Kozen [14, 15] uses Boolean matrices to provide an elegant proof that polynomials are regularity-preserving.

With respect to matrix-based approaches to automata theory, the important work of Kuich and Salomaa [17] should be pointed out. Kuich, Salomaa, and others present a uniform treatment of many results in automata theory and formal languages through the study of formal power series and linear algebra over semirings. Kozen's LICS'91 paper [13] uses matrix-based ideas as well in showing the completeness of an axiomatization of Kleene algebras (see also [2, 5, 16, 21]). In contrast, MAAT is not so general, although the use of Boolean matrices as states turns out to be the right middle ground for the study of regularity-preserving functions.

It is important to note that the acronym MAAT is created as a convenient way for reference to the specific method of construting finite automata via Boolean matrices. As indicated elsewhere, we claim neither the originality of the idea, nor the originality of general matrix-based approaches to automata theory.

1.2 Organization of the paper

The rest of the paper is organized as follows. Section 2 reviews some basic terminology and fixes notations. Section 3 contains motivating examples illustrating the ideas behind forthcoming proofs. Section 4 presents the main technical result of the paper, which is the characterization of regularity-preserving functions as the ones that are ultimately periodic with respect to powers of Boolean matrices. Section 5 applies the main result to derive structural properties of regularity-preserving functions, to present properties of some new language operators, and to resolve the subtraction problem. Section 6 discusses regularity-preserving relations. Finally, Section 7 points out some related directions.

2 Notation and preliminaries

2.1 Boolean matrices

A *Boolean matrix* is a matrix (of size $m \times n$) whose elements are either 0 or 1. Such matrices are also called $(0,1)$-matrices, but the name *Boolean* makes it clear that when the standard operations on matrices are used, the internal operations on elements obey Boolean laws such as $1+1=1$. In this paper, we are concerned with square Boolean matrices indexed over a set Q. We write $\mathcal{M}(Q)$ for the set of such matrices, which are nothing but functions from $Q \times Q$ to $\{0,1\}$.

A square Boolean matrix is called a *permutation matrix* if every row and every column of it contains exactly one 1. Such a matrix has the nice property that its inverse is just its transpose. (In fact, a Boolean matrix has an inverse if and only if it is a permutation matrix [9].)

A *Boolean vector* of dimension n is an n-tuple $(b_1, b_2, \ldots, b_n)$ of 0s and 1s. This is also called a row vector. A column vector is the (matrix) transpose of a row vector (an $n \times 1$ matrix). We are mostly interested in vectors indexed over a set Q as well, and so such vectors $\mathcal{V}(Q)$ are nothing but functions from Q to $\{0,1\}$. The *inner product* of two vectors is the sum of the component-wise product of their elements. This sum is just the matrix product of a $1 \times n$ vector and an $n \times 1$ vector, which is indistinguishable from a Boolean value.

Given a directed graph over a vertex set Q, we let $\Delta(p,q) = 1$ if there is an edge from p to q, and otherwise let $\Delta(p,q) = 0$. Such a Boolean matrix Δ is called the *adjacency matrix* [9] of the graph. The *characteristic vector* of a subset A of Q is the row vector I_A such that $\mathrm{I}_A(p) = 1$ if and only if $p \in A$. The characteristic vector of a singleton set $\{p\}$ is written as I_p. In general, we write I for an identity matrix, and O for a matrix with all its elements being 0.

2.2 Automata and Boolean matrices

We follow [8] for notations related to automata theory. By a deterministic finite automaton (dfa; we use this for both singular and the plural in the rest of the paper) M we mean a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where Q is a state set, Σ is an alphabet, δ is a transition function, q_0 is the starting state, and F is a set of final states. Throughout this paper, when there is a transition from state p to state q in an automaton using a string x, we will say that x *drives* the machine from p to q; we will also say there is a *path* of length $|x|$ from p to q.

Each dfa determines a *Boolean matrix system* $\{\Delta^a \mid a \in \Sigma\}$, where $\Delta^a \in \mathcal{M}(Q)$ is the adjacency matrix of the a-labeled subgraph associated with the dfa. In other words, $\Delta^a(p,q) = 1$ if and only if $\delta(p,a) = q$. Since M is a dfa, each Δ^a has the property that every row contains exactly one 1. The sum Δ of all members Δ^a in the Boolean matrix system of a dfa is called the *adjacency matrix of the dfa*. Such a Δ ignores the labels of the underlying graph of the dfa. The adjacency matrix is useful because of this well-known fact:

Lemma 2.1 *For every $i \geq 0$, $\Delta^i(p,q) = 1$ if and only if there is a length-i transition in M from state p to state q. Here, Δ^0 is defined to be the identity matrix.*

For a string $x = a_1 a_2 \cdots a_n$ over Σ, we write Δ^x for the matrix product $\Delta^{a_1}\Delta^{a_2}\cdots\Delta^{a_n}$. This is unambiguous, as we insist that a_i's be symbols form Σ. We can describe the language accepted by a dfa as follows.

Lemma 2.2 *Let M be a dfa, and $\{\Delta^a \mid a \in \Sigma\}$ its Boolean matrix system. Then the language accepted by M is the set $\{x \mid \mathrm{I}_{q_0}\Delta^x\mathrm{I}_F^t = 1\}$, where q_0 is the starting state of M, F is the set of final states of M, and $(\ )^t$ stands for matrix transpose.*

Again, the simple proof for this well-known lemma is omitted.

2.3 Regularity-preserving functions

Let ω be the set $\{0, 1, 2, 3, \ldots\}$ of non-negative integers and $f : \omega \to \omega$ be a function. In the literature, there are two notions of regularity-preserving functions. One is what we use in this paper: f is *regularity-preserving* if the set

$$\mathsf{T}(L, f) := \{x \mid \exists y \ \ |y| = f(|x|) \ \& \ xy \in L\}$$

is regular for every regular language L. The second uses, instead,

$$\mathsf{T}'(L, f) := \{x \mid \exists y \in L \ \ |y| = f(|x|)\}.$$

However, Kozen ([14], Theorem 5, page 6) shows that the two notions are equivalent.

To give a summary of known characterizations of regularity-preserving functions [14, 11, 12, 22], we recall some definitions. A subset A of ω is called *ultimately periodic* (u.p.) if there exists a number $m > 0$ such that $i \in A \iff i + m \in A$ holds for all but finitely many i. A well-known property of a regular language is that its length-set is always ultimately periodic. A function $f : \omega \to \omega$ is called *ultimately periodic modulo* m if there exists a $p > 0$ such that $f(i) \equiv f(i + p) \bmod m$ holds for all but finitely many $i \geq 0$.

Theorem 2.1 (Kozen, Seiferas and McNaughton, Kosaraju) *The following four conditions are equivalent:*

1. $\mathsf{T}(L, f)$ *is regular for every regular* L.
2. $\mathsf{T}'(L, f)$ *is regular for every regular* L.
3. *the inverse image* $f^{-1}(A)$ *of any u.p. set* A *is again u.p.*
4. *for any* $m \geq 1$, f *is ultimately periodic modulo* m, *and the inverse image of any singleton set under* f *is u.p.*

This characterization, however, is not completely satisfactory for several reasons. One is that the proof of f's preservation of u.p. sets under inverse image does not render the explicit construction of a dfa on which our computational intuition may rest. The second is that such a characterization involves the use of inverse image and u.p. sets, making it inconvenient to apply. One only needs to spend a few minutes attempting to prove, for example, item 4 of Theorem 5.1 to realize this. Perhaps the more important reason is its limitations in treating pattern-related language operations such as the operators P, Q mentioned earlier.

In comparison, our characterization states that a function is regularity-preserving if and only if it is ultimately periodic with respect to powers of Boolean matrices (Theorem 4.1). The rest of the paper contains a proof of this result, as well as applications showing the advantages of the characterization. Operator P, in particular, includes many interesting new examples:

$$\begin{aligned} L_1 &:= \{w \mid www \in L\}, \\ L_2 &:= \{w \mid w^{|w|} \in L\}, \\ L_3 &:= \{w \mid w^{2^{|w|}} \in L\}. \end{aligned}$$

All of these are regularity-preserving, as a consequence of Theorem 5.2.

3 Examples

This section presents three examples to illustrate the key ideas of MAAT. The same ideas are used in the subsequent proofs. We mention in passing that many problems at the end of Chapter 3 of [8], usually considered difficult, can now be solved through MAAT in a straightforward, uniform manner.

The exponential function $\lambda x.\, 2^x$ is regularity-preserving. Given a language L and an accepting dfa $M = (Q, \Sigma, \delta, q_0, F)$ with adjacency matrix Δ, we construct the dfa

$$M' = (Q', \Sigma, \delta', q_0', F'),$$

where

- $Q' = \{(p, \mathrm{A}) \mid p \in Q \ \&\ \mathrm{A} \in \mathcal{M}(Q)\}$,
- $q_0' = (q_0, \Delta)$,
- $\delta'((p, \mathrm{A}), a) = (\delta(p, a), \mathrm{A}^2)$, and
- $F' = \{(p, \mathrm{A}) \mid \mathrm{I}_p \mathrm{A} \mathrm{I}_F^t = 1\}$.

There are only finitely many Boolean matrices in $\mathcal{M}(Q)$, so we do have a dfa. The fact that M' indeed accepts the language $\mathsf{T}(L, \lambda x.\, 2^x)$ can be done by an easy mathematical induction on the lengths of strings accepted by it. It suffices to note that

$$(q_0, \Delta) \xrightarrow{a_1} (q_1, \Delta^2) \xrightarrow{a_2} (q_2, \Delta^4) \xrightarrow{a_3} \cdots \xrightarrow{a_n} (q_n, \Delta^{2^n})$$

is a transition sequence in M' ending with a final state if and only if the string $a_1 a_2 \cdots a_n$ drives M from q_0 to q_n and, moreover, from q_n, there is a path of length 2^n in M leading to an accepting state. (A careful reader can check that M' behaves appropriately with respect to the empty string as well.)

Small variations of the construction are possible to make M' accept different languages. For example, if we define the final states to be $F'' = \{(p, \mathrm{A}) \mid I_{q_0} \mathrm{A} I_F^t = 1\}$, we get a dfa accepting $\mathsf{T}'(L, \lambda x.\, 2^x)$; if we change the definition of the transition function to $\delta''((p, \mathrm{A}), a) = (\delta(p, a), \mathrm{A}^9)$, we get a dfa accepting the language $\mathsf{T}(L, \lambda x.\, 9^x)$.

The square root of a language L is given as $\sqrt{L} := \{w \mid w^2 \in L\}$. (Note that a closely related operator, $\{w^2 \mid w \in L\}$, clearly does not preserve regularity.) We show that the square root operation preserves regular sets. The idea is similar: given L and an accepting dfa $M = (Q, \Sigma, \delta, q_0, F)$ with matrix system $\{\Delta^a \mid a \in \Sigma\}$, we construct the dfa

$$M' = (Q', \Sigma, \delta', q_0', F'),$$

where

- $Q' = \{(p, \mathrm{A}) \mid p \in Q \ \&\ \mathrm{A} \in \mathcal{M}(Q)\}$,

- $q'_0 = (q_0, \mathrm{I})$,
- $\delta'((p, \mathrm{A}), a) = (\delta(p, a), \mathrm{A}\Delta^a)$, and
- $F' = \{(p, \mathrm{A}) \mid \mathrm{I}_p \mathrm{A} \mathrm{I}_F^t = 1\}$.

To see why this construction works, note that a string x drives the machine M' from state (q_0, I) to state (p, A) if and only if x drives the machine M from q_0 to p, and $\mathrm{A} = \Delta^x$. However, $\mathrm{I}_p \Delta^x \mathrm{I}_F^t = 1$ means that, by Lemma 2.2, x drives M from state p to a final state. Therefore, x is accepted by M' if and only if xx is accepted by M.

The Fibonacci sequence $F_0, F_1, \ldots F_n, \ldots$ is defined inductively as

$$F_0 = 0,\ F_1 = 1,\ \text{ and } F_n = F_{n-2} + F_{n-1} \text{ for } n \geq 2.$$

To show that $\lambda x.\, F_x$ is regularity-preserving, let L be a regular language and

$$M = (Q, \Sigma, \delta, q_0, F)$$

an accepting dfa with adjacency matrix Δ. We construct the new dfa

$$M' = (Q', \Sigma, \delta', q'_0, F'),$$

where

- $Q' = \{(p, \mathrm{A}, \mathrm{B}) \mid p \in Q \ \&\ \mathrm{A}, \mathrm{B} \in \mathcal{M}(Q)\}$,
- $q'_0 = (q_0, \mathrm{I}, \Delta)$,
- $\delta'((p, \mathrm{A}, \mathrm{B}), a) = (\delta(p, a), \mathrm{B}, \mathrm{AB})$,
- $F' = \{(p, \mathrm{A}, \mathrm{B}) \mid \mathrm{I}_p \mathrm{A} \mathrm{I}_F^t = 1\}$.

The following sequence depicts a typical computation with M':

$$(q_0, \Delta^0, \Delta^1) \xrightarrow{a_1} (q_1, \Delta^1, \Delta^1) \xrightarrow{a_2} \cdots \xrightarrow{a_n} (q_n, \Delta^{F_n}, \Delta^{F_{n+1}})$$

where, by Lemma 2.1, Δ^{F_n} determines the set of all state pairs (p, q) such that there exists a "path" of length F_n from p to q in M. Thus, according to the construction of M', a string $a_1 a_2 \cdots a_n$ is accepted by M' if and only if the string can drive M from q_0 to q_n, and, moreover, there exists a path of length F_n from q_n to a final state of M. This shows the machine M' has exactly the desired properties (a more formal proof should follow the style of mathematical induction).

The idea behind these examples can be applied to many other functions as well, *without resorting to* ultimate periodicity. In general, suppose k is the total number of function names in the summation of the right hand side of a recurrence relation system. We can use states of the form $(p, \mathrm{A}_1, \mathrm{A}_2, \ldots, \mathrm{A}_k)$ with $\mathrm{A}_i \in \mathcal{M}(Q)$, $1 \leq i \leq k$, to *simulate the recursive definition* of a function in a dfa. Thus we have the following result.

Proposition 3.1 *A function is regularity-preserving if it can be defined by a recurrence relation using bounded summation of linear forms of function names.*

As an immediate corollary of this result, we see that all linear functions ($\lambda x.kx$), polynomials, and exponentials ($\lambda x.k^x$) are regularity preserving (these are well-known facts). This result is, however, rather restricted, in light of Theorem 5.1 later.

4 Ultimate periodicity and regularity preservation

The main result of this section is a new characterization of regularity-preserving functions, stated as follows.

Theorem 4.1 *A function f is regularity-preserving if and only if for any (square) Boolean matrix Δ, there exists a positive number m such that*

$$\Delta^{f(i)} = \Delta^{f(i+m)}$$

for all but finitely many $i \geq 0$.

Notation. A function which is ultimately periodic with respect to powers of Boolean matrices will be abbreviated as m.u.p.

Example. The identity function is m.u.p. To see this, note that there are only finitely many different matrices of a given dimension, and so the list $\Delta^1, \Delta^2, \ldots, \Delta^i \ldots$ will contain a repetition, by the Pigeonhole principle. Such a repetition determines a desired period m and a number $k > 0$ such that $\Delta^{k+i} = \Delta^{k+i+m}$ for all $i \geq 0$. This shows that the identity function is m.u.p.

□

Example. The exponentiation $\lambda x.\, 2^x$ is m.u.p. Consider the list $\Delta^{2^0}, \Delta^{2^1}, \Delta^{2^2} \ldots$ and we know, again by the Pigeonhole principle, that there must be some $k, m > 0$ such that $\Delta^{2^k} = \Delta^{2^{k+m}}$. Now, for any $i \geq 0$ we calculate: $\Delta^{2^{k+i}} = (\Delta^{2^k})^{2^i} = (\Delta^{2^{k+m}})^{2^i} = \Delta^{2^{k+i+m}}$.

□

The following lemmas relate the regularity-preservation of a function to its ultimate periodicity with respect to powers of Boolean matrices.

Lemma 4.1 *A function $f : \omega \to \omega$ is regularity-preserving if it is ultimately periodic with respect to powers Boolean matrices.*

Proof. Suppose f is ultimately periodic with respect to powers of Boolean matrices. We want to show that it is regularity-preserving. The idea is this: if $M = (Q, \Sigma, \delta, q_0, F)$ is a dfa accepting L, then we can construct a dfa $M' = (Q', \Sigma, \delta', q'_0, F')$ accepting $\mathsf{T}(L, f)$. To specify M', we fix a few notations first. Let Δ be the adjacency matrix associated with M. By the given property of f, there exist $m, k \geq 1$ such that $\Delta^{f(i+k)} = \Delta^{f(i+k+m)}$ for all $i \geq 0$. We write $[i]_k^m$ for i if $i \leq k$, and $k + [(i-k) \mathsf{\ mod\ } m]$ if $i > k$. Note that we have $0 \leq [i]_k^m < k + m$. Now, M' can be given as follows.

- $Q' = \{(p, i, \Delta^{f(i)}) \mid p \in Q \;\&\; 0 \leq i < m + k\}$,
- $q_0' = (q_0, 0, \Delta^{f(0)})$,
- $\delta'((p, i, \Delta^{f(i)}), a) = (\delta(p, a), [i+1]_k^m, \Delta^{f([i+1]_k^m)})$, and
- $F' = \{(p, i, \mathrm{A}) \mid \mathrm{I}_p \mathrm{A} \mathrm{I}_F^t = 1\}$.

Clearly, we have a well-defined transition function and M' is a dfa. According to the definition of M', a string x is accepted by M' if and only if x drives M' from the initial state $(q_0, 0, \Delta^{f(0)})$ to a final state $(p, j, \Delta^{f(|x|)})$. But $(p, j, \Delta^{f(|x|)})$ is a final state if and only if $\Delta^{f(|x|)}(p, q) = 1$ for some $q \in F$. Therefore, x is accepted by M' if and only if x can drive the original M from q_0 to some state p, from which there is a string of length $f(|x|)$ to continue driving M to a final state, by Lemma 2.1.

□

By changing the definition of the final states of M' to $F'' = \{(p, i, \mathrm{A}) \mid \mathrm{I}_{q_0} \mathrm{A} \mathrm{I}_F^t = 1\}$, we get a dfa which accepts $\mathsf{T}'(L, f)$. Therefore, if a function $f : \omega \to \omega$ is ultimately periodic with respect to Boolean matrices, then for every regular language L, the language $\mathsf{T}'(L, f)$ is also regular.

We now prove the converse of Lemma 4.1.

Lemma 4.2 *If a function $f : \omega \to \omega$ is regularity-preserving, then it is ultimately periodic with respect to powers of Boolean matrices.*

Proof. Suppose f is regularity-preserving. We want to show that it is ultimately periodic with respect to powers of Boolean matrices. Given a Boolean matrix Δ, we associate with it the singleton alphabet $\{1\}$ and a state set Q whose size is the same as the dimension of Δ. This way, we get a collection of *nondeterministic finite automata* by varying the starting state and the final state and so we get a finite collection of regular languages: $\{1^i \mid \Delta^i(p, q) = 1\}$, with $p, q \in Q$. By the regularity-preserving property of f and by Theorem 2.1, the languages $\{1^i \mid \exists k\, (k = f(i) \;\&\; \Delta^k(p, q) = 1)\}$ are regular for all $p, q \in Q$. From this we know that $\{i \mid \Delta^{f(i)}(p, q) = 1\}$ are u.p. for every $p, q \in Q$. By the definition of ultimately periodic sets, for each p, q, there exists an $m > 0$ such that $\Delta^{f(j)}(p, q) = 1$ if and only if $\Delta^{f(j+m)}(p, q) = 1$ for all but finitely many $j > 0$. Now let k be a common period of all these sets (a multiplication of the periods will do), we have $\Delta^{f(j)} = \Delta^{f(j+k)}$ for all but finitely many $j > 0$, and this was what we wanted.

□

The previous two lemmas constitute a proof of Theorem 4.1. Note that although we need to consider only languages over $\{1\}$ in the proof of the last lemma, the assumption that f is regularity-preserving in general, rather than just for languages over $\{1\}$, is essential. The equivalence of T-regularity-preserving and T'-regularity-preserving is needed in the proof; but this is only true with respect to general languages.

5 Applications

5.1 Structural properties of regularity-preserving functions

Our new characterization makes it relatively simple to prove structural properties of regularity-preserving functions. In many cases, it becomes just a matter of a short proof by induction.

Theorem 5.1 *Suppose f and g are regularity-preserving functions. Then the following functions also preserve regularity: (1) composition $f(g)$; (2) addition $f+g$; (3) multiplication $f \cdot g$; (4) exponentiation $\lambda x.\, f(x)^{g(x)}$, provided that f is positive; (5) $\lambda x.\, \Sigma_{i \leq x} f(i)$; and (6) $\lambda x.\, \Pi_{i \leq x} f(i)$.*

The first four items have been suggested by Kosaraju [11], and the last two items seem to be new – as far as we know. Note that Item 6 implies that the factorial function $\lambda x.x!$ is regularity-preserving.

Proof of Theorem 5.1. Item 1 is straightforward if one considers $\mathsf{T}(\mathsf{T}(L,f),g)$. Proof of Items 2 and 3 amounts to an easy mathematical induction in light of Theorem 4.1.

To prove Item 4, suppose f, g are regularity-preserving functions. Let Δ be a Boolean matrix, and let $\Delta_1, \Delta_2, \ldots, \Delta_n$ be all Boolean matrices of the same size as Δ. By applying Theorem 4.1, we can find $k, m > 0$ such that

$$\forall i \geq 0,\ \Delta_j^{f(i+k)} = \Delta_j^{f(i+k+m)} \tag{1}$$

holds uniformly for all $1 \leq j \leq n$.

Clearly, every function in the list $\lambda x.f(k)^{g(x)}, \lambda x.f(k+1)^{g(x)}, \ldots, \lambda x.f(k+m-1)^{g(x)}$ is m.u.p. Hence, we can choose $\alpha > k$ and β, a multiple of m, such that

$$\forall y \geq 0,\ \Delta^{f(i+k)^{g(y+\alpha)}} = \Delta^{f(i+k)^{g(y+\alpha+\beta)}} \tag{2}$$

holds uniformly for all $0 \leq i \leq m-1$. We can now calculate, for any number $y \geq 0$,

$$\begin{aligned}
\Delta^{f(y+\alpha)^{g(y+\alpha)}} &= \Delta^{f(i_0+k)^{g(y+\alpha)}} && 0 \leq i_0 \leq m-1;\ \text{Equation 1} \\
&= \Delta^{f(i_0+k)^{g(y+\alpha+\beta)}} && \text{Equation 2} \\
&= \Delta^{f(y+\alpha)^{g(y+\alpha+\beta)}} && \text{Equation 1, again} \\
&= \Delta^{f(y+\alpha+\beta)^{g(y+\alpha+\beta)}} && \beta \text{ is a multiple of } m
\end{aligned}$$

This proves that the function f^g is m.u.p.

Among Items 5 and 6, we prove the relatively harder one, Item 6. By Theorem 4.1, it suffices to prove that the function $\pi = \lambda x.\, \Pi_{i \leq x} f(i)$ is m.u.p. given that f is. Let Δ be a Boolean matrix, and let $\Delta_1, \Delta_2, \ldots, \Delta_n$ be all Boolean matrices of the same size as Δ. By applying Theorem 4.1, we can find $k, m > 0$ such that

$$\forall i \geq 0,\ \Delta_j^{f(i+k)} = \Delta_j^{f(i+k+m)} \tag{3}$$

holds uniformly for all $1 \leq j \leq n$. Consider the sequence

$$\Delta^{\pi(k)}, \Delta^{\pi(k+m)}, \Delta^{\pi(k+2m)}, \ldots, \Delta^{\pi(k+t \cdot m)}, \ldots$$

and let $\alpha > k$ and β, a multiple of m, be such that

$$\Delta^{\pi(\alpha)} = \Delta^{\pi(\alpha+\beta)} \tag{4}$$

The existence of such α and β is guaranteed by the Pigeonhole principle. Our goal is to prove that for any number $y \geq 0$,

$$\Delta^{\pi(\alpha+y)} = \Delta^{\pi(\alpha+y+\beta)} \tag{5}$$

and we do this by mathematical induction. Equation 4 furnishes the basis. For the induction step, assuming

$$\Delta^{\pi(\alpha+y)} = \Delta^{\pi(\alpha+y+\beta)} \tag{6}$$

for some y and we calculate

$$\begin{aligned}
\Delta^{\pi(\alpha+y+1)} &= (\Delta^{\pi(\alpha+y)})^{f(\alpha+y+1)} & \\
&= \Delta_{j_0}^{f(\alpha+y+1)} & 1 \leq j_0 \leq n;\ \text{Equation 3} \\
&= \Delta_{j_0}^{f(\alpha+y+1+\beta)} & \alpha > k \text{ and } \beta \text{ a multiple of } m \\
&= (\Delta^{\pi(\alpha+y)})^{f(\alpha+y+1+\beta)} & \\
&= (\Delta^{\pi(\alpha+y+\beta)})^{f(\alpha+y+1+\beta)} & \text{induction hypothesis: Equation 6} \\
&= \Delta^{\pi(\alpha+y+1+\beta)},
\end{aligned}$$

which finishes the proof.

□

5.2 Other language operators

Consider language operators P and Q, where

- $\mathsf{P}(L, f) := \{x \mid x^{f(|x|)} \in L\}$, and
- $\mathsf{Q}(L, f) := \{a_0 a_1 a_2 \cdots a_n \mid a_0^{f(0)} a_1^{f(1)} a_2^{f(2)} \cdots a_n^{f(n)} \in L\}$.

In general, we say that a function f is X-regularity-preserving if for any regular language L, $\mathsf{X}(L, f)$ is regular. We call f X_1-regularity-preserving if for any regular language L over $\{1\}$, $\mathsf{X}(L, f)$ is regular. Clearly, any X-regularity-preserving function is also X_1-regularity-preserving; but the converse is not true.

Theorem 5.2 *(1) If a function $f : \omega \to \omega$ is P-regularity-preserving, then the function $\lambda x.\, x f(x)$ is m.u.p.; (2) If $f : \omega \to \omega$ is m.u.p., then it is P-regularity-preserving.*

Proof. The proof of Item 1 is similar to that of Lemma 4.2; hence it is omitted.

For Item 2, suppose f is ultimately periodic with respect to powers of Boolean matrices. We wanted to show that it is P-regularity-preserving. The idea is to construct a dfa $M' = (Q', \Sigma, \delta', q_0', F')$ accepting $\mathsf{P}(L, f)$ given a dfa $M = (Q, \Sigma, \delta, q_0, F)$ accepting L. To specify M', we fix a few notations first. Let $\{\Delta^a \mid a \in \Sigma\}$ be the adjacency matrix system associated with M. By the given property of f , there exist $m, k \geq 1$ such that $\Delta_j^{f(i+k)} = \Delta_j^{f(i+k+m)}$ uniformly for all $i \geq 0$ and $j = 1, \ldots n$, with $\{\Delta_j \mid 1 \leq j \leq n\}$ the set of all distinct Boolean matrices of the same dimension as Δ. Recall that we write $[i]_k^m$ for i if $i \leq k$, and $k + [(i - k) \bmod m]$ if $i > k$; this means $0 \leq [i]_k^m < k + m$. Now, M' can be given as follows.

- $Q' = \{(p, i, \Delta_j) \mid p \in Q \;\&\; 0 \le i < m+k \;\&\; 1 \le j \le n\}$,
- $q'_0 = (q_0, 0, \mathrm{I})$,
- $\delta'((p, i, \mathrm{A}), a) = (\delta(p, a), [i+1]_k^m, \mathrm{A}\Delta^a)$, and
- $F' = \{(p, i, \mathrm{A}) \mid \mathrm{I}_p \mathrm{A}^{f(i)} \mathrm{I}_F^t = 1\}$.

Clearly, we have a well-defined transition function and M' is a dfa. According to the definition of M', a string x is accepted by M' if and only if x drives M' from the initial state $(q_0, 0, \mathrm{I})$ to a final state (p, i, A), with $\mathrm{A} = \Delta^x$. But (p, i, Δ^x) is a final state if and only if $(\Delta^x)^{f(|x|)}(p, q) = 1$ for some $q \in F$. Therefore, x is accepted by M' if and only if x can drive the original M from q_0 to some state p, from which the string $x^{f(|x|)}$ drives M to a final state, by Lemma 2.1.

$\square$

It may be the case that f is P-regularity-preserving if and only if $\lambda x.\, x f(x)$ is m.u.p. However, we were unable to close the gap between the m.u.p. of $\lambda x.\, x f(x)$ and that of f. It boils down to the following problem, which has to be left unsolved here.

> Suppose, for a function $f : \omega \to \omega$, the function $\lambda x.\, x f(x)$ is m.u.p. Is it necessary that f itself is m.u.p.?

Although T-regularity-preserving functions are Q-regularity-preserving, we have not found a good characterization of Q-regularity-preserving functions. However, for a singleton alphabet, we do have this.

Proposition 5.1 *A function $f : \omega \to \omega$ is Q_1-regularity-preserving if and only if it is T_1-regularity-preserving.*

To better understand why this is true, we provide a characterization of language operators restricted to the singleton alphabet.

Theorem 5.3 *A function $f : \omega \to \omega$ is T_1-regularity-preserving if and only if it is ultimately periodic with respect to powers of* **permutation matrices**.

Proof. (Only if) Suppose f is T_1-regularity-preserving. We want to show that it is ultimately periodic with respect to powers permutation matrices. Given such a matrix Δ, we associate with it the singleton alphabet $\{1\}$ and a state set Q whose size is the same as the dimension of Δ. This way, we get a collection of finite automata by varying the starting state and the final state and so we get a finite collection of regular languages: $\{1^i \mid \Delta^i(p, q) = 1\}$, with $p, q \in Q$. By the T_1-regularity-preserving property of f, the languages $\{1^i \mid \exists k\, (k = f(i) \;\&\; \Delta^{i+k}(p, q) = 1)\}$ are regular for all $p, q \in Q$. From this we know that $\{i \mid \Delta^{i+f(i)}(p, q) = 1\}$ are u.p. for every $p, q \in Q$. By the definition of ultimately periodic sets, for each p, q, there exists an $m > 0$ such that $\Delta^{j+f(j)}(p, q) = 1$ if and only if $\Delta^{(j+m)+f(j+m)}(p, q) = 1$ for all but finitely many $j > 0$. Since Δ is a permutation matrix, for some $\beta > 0$ we have $\Delta^\beta = \mathrm{I}$, the identity matrix. Now let k, a multiple of β, be a common period (a multiplication of the periods will do) of all the sets

$$\{i \mid \Delta^{i+f(i)}(p,q) = 1\}$$

with $p, q \in Q$. We have $\Delta^{j+f(j)} = \Delta^{(j+k)+f(j+k)}$ for all but finitely many $j > 0$. Now, by the invertibility of Δ and the fact that $\Delta^k = \mathrm{I}$, we obtain the fact that $\Delta^{f(j)} = \Delta^{f(j+k)}$ for all but finitely many $j > 0$. Therefore, f is ultimately periodic with respect to powers permutation matrices.

(If) In this part we prove that if a function $f : \omega \to \omega$ is ultimately periodic with respect to powers of permutation matrices, then it preserves regularity for langauges over the singleton alphabet $\{1\}$.

Suppose f is one such function, and L is a regular langauge over $\{1\}$. We know that there exists a dfa $(Q, \{1\}, \delta, q_0, F)$ accepting L. We can, in fact, further assume that this dfa has a minimal number of states, so it shapes like a lollipop. We can label the states as $Q = Q_1 \cup Q_2$ with $Q_1 = \{q_0, q_2, q_3, \ldots, q_{k-1}\}$ for the "stick" part of the "lollipop"; and $Q_2 = \{q_k, \ldots, q_{k+m-1}\}$ for the "disk" part of the "lollipop". Let P be the permutation matrix (of size $m \times m$) which represents the "disk". By the given property of f, there exist $\alpha, \beta > k$ such that $\mathrm{P}^{\alpha+i} = \mathrm{P}^{\alpha+i+\beta}$ for all $i \geq 0$. We construct the dfa $M' = (Q', \{1\}, \delta', q_0', F')$, where (recall that the notation $[i]_k^m$ was introduced in the proof of Lemma 4.1)

- $Q' = \{(q, i, \mathrm{P}^{f(i)}) \mid q \in Q_2 \ \& \ 0 \leq i < \alpha + \beta\}$,
- $q_0' = (q_k, k, \mathrm{P}^{f(k)})$,
- $\delta'((p, i, \mathrm{P}^{f(i)}), 1) = (\delta(p, 1), [i+1]_\alpha^\beta, \mathrm{P}^{f([i+1]_\alpha^\beta)})$, and
- $F' = \{(p, i, \mathrm{A}) \mid \mathrm{I}_p \mathrm{A} \mathrm{I}_G^t = 1\}$, where $G = F \cap Q_2$.

Similar to the proof of Lemma 4.1, we can show that the language accepted by M' is $\{1^i \mid 1^{i+k+f(i+k)} \in L \ \& \ i \geq 0\}$, and so the langauge $\{1^i \mid 1^{i+f(i)} \in L \ \& \ i \geq k\}$ is regular. However, $\mathsf{T}_1(L, f)$ differs from this language by only a finite set, and so $\mathsf{T}_1(L, f)$ is regular, too.

□

Since permutation matrices are invertible, we can use the previous theorem to prove that

Corollary 5.1 *(i) A function $f : \omega \to \omega$ is T_1-regularity-preserving if and only if the function $\lambda x.\ \Sigma_{i \leq x} f(i)$ is. (ii) T_1-regularity-preserving functions are closed under subtraction (whenever the resulting function is non-negative).*

Here we have an interesting connection between Theorem 5.3 and condition (4) of Theorem 2.1.

Proposition 5.2 *A function $f : \omega \to \omega$ is ultimately periodic modulo m for any $m \geq 1$ if and only if it is ultimately periodic with respect to powers of permutation matrices.*

Proof. (Only if) Suppose $f : \omega \to \omega$ is ultimately periodic modulo m for any $m \geq 1$. Given any permutation matrix Δ, there exists an integer $\beta \geq 1$ such that $\Delta^\beta = \mathrm{I}$, the identity matrix. We know that f is ultimately periodic modulo β; so there exists a $p > 0$ such that $f(i) \equiv f(i+p) \bmod \beta$ holds for all but finitely many $i \geq 0$. Therefore, $\Delta^{f(i)} = \Delta^{f(i+p)}$ for all but finitely many $i \geq 0$. This proves that f is ultimately periodic with respect to powers of permutation matrices.

(If) Suppose f is ultimately periodic with respect to powers of permutation matrices. For any $m \geq 1$, it is easy to check that there exists a permutation matrix Δ such that m is the least integer other than 0 for which $\Delta^m = \mathrm{I}$ holds. For this Δ, there exists a $p > 0$ such that $\Delta^{f(i)} = \Delta^{f(i+p)}$ for all but finitely many $i \geq 0$. By the given property of Δ, this can only happen if $f(i) \equiv f(i+p) \bmod m$ holds for all but finitely many $i \geq 0$.

□

5.3 The subtraction problem

The subtraction problem is this: if f and g are regularity-preserving (under T), is $f - g$ also regularity-preserving (assuming that it is non-negative)? An affirmative answer was announced in [11] but soon withdrawn [12]. This leaves us wonder what is the status of the problem, as it has not been discussed in the literature since then.

We need a result which is helpful in the identification of regularity-preserving functions.

Proposition 5.3 *A function $f : \omega \to \omega$ is regularity preserving if (i) it is ultimately periodic with respect to powers of permutation matrices, and (ii) $\lim_{n\to\infty} f(n) = \infty$.*

Proof. For any Boolean matrix Δ, there exist $k, m > 0$ such that $\Delta^i = \Delta^{i+m}$ for all $i \geq k$, as the identity function is m.u.p. By Proposition 5.2, there exist $s, n > 0$ such that $f(i+n) \equiv f(i) \bmod m$ for all $i \geq s$. By condition (ii), for all but finitely many i, we have $f(i) \geq k$. Therefore, for all sufficiently large i, we have $\Delta^{f(i)} = \Delta^{f(i+n)}$.

□

We have now achieved an understanding of regularity-preserving functions to the extent that a simple counterexample can be found to the subtraction problem.

Let P be the set of prime numbers, $g(x) = x!$ (the factorial) and let f be defined as: $f(x) = x!$ for $x \in P$, and $f(x) = 2(x!)$ for $x \notin P$. By item (6) of Theorem 5.1 and the previous proposition, f and g are both regularity-preserving. However, $h = f - g$ *is not* regularity-preserving (although it *is* T_1-regularity-preserving). To see that h is not regularity-preserving, all we need to consider is the 2×2 matrix with the only non-zero entry being $(1, 1)$, and the fact that prime numbers are not ultimately periodic.

It is interesting to point out that the new results of this paper are not critical for the counterexample. The same conclusion can be made by considering $h^{-1}(\{0\})$ in light of Theorem 2.1. However, it is puzzling why such a simple counterexample had not been discovered earlier.

6 Regularity-preserving relations

The paper of Seiferas and McNaughton [22] actually considered *regularity-preserving relations*. Given a relation $r \subseteq \omega \times \omega$, r is said to be regularity-preserving if for any regular L, the set

$$\mathsf{R}(L, r) := \{x \mid \exists y\, (|x|, |y|) \in r \;\&\; xy \in L\}$$

is regular as well. The main result of [22] is that r is regularity-preserving if and only if for any u.p. set $A \subseteq \omega$, the set $r^{-1}(A)$ is u.p., where

$$r^{-1}(A) := \{i \mid \exists j \in A\; (i, j) \in r\}.$$

Regularity-preserving relations are useful in expressing some language constructions. For example, one can see that the relation $r = \omega \times \omega$ preserves the inverse image of u.p. sets, and for this particular r, $\mathsf{R}(L, r)$ corresponds to the prefix-closure of L. It is well-known that prefix-closure preserves regular sets; but here we can see this as a special case of a more general fact.

We can rephrase the result of Seiferas and McNaughton in terms of infinite Boolean vectors and infinite Boolean matrices. This will lead to an intrinsic, unresolved issue about infinite Boolean matrices.

An infinite Boolean vector is an infinite "tuple"

$$\mathbf{v} = (v_1, v_2, \ldots, v_n, \ldots)$$

such that each v_n is either 0 or 1. Clearly, there is a 1-1 correspondence between such vectors and subsets of integers. Given an infinite Boolean vector $\mathbf{v}$, we get a subset $\{i \in \omega \mid v_i = 1\}$. Conversely, given any subset A of ω, we get an infinite Boolean vector $\mathbf{v}$ by letting $v_i = 1$ if and only if $i \in A$.

Based on this observation, we will say that an infinite Boolean vector is ultimately periodic (u.p.) if the corresponding set is u.p. Clearly, u.p. Boolean vectors are closed under addition $(1 + 1 = 1)$ and scalar multiplication by a Boolean value.

On the other hand, there is also a 1-1 correspondence between binary relations $r \subseteq \omega \times \omega$ and infinite Boolean matrices $M = [M_{ij}]$ by letting $(i, j) \in r$ exactly when $M_{ij} = 1$.

Proposition 6.1 *A relation $r \subseteq \omega \times \omega$ is regularity-preserving if and only if the transpose of the corresponding infinite Boolean matrix M determines a transformation which preserves u.p. Boolean vectors.*

Proof. Let $r \subseteq \omega \times \omega$ be a relation and M its corresponding infinite Boolean matrix. Suppose $\mathbf{v} = \mathbf{u}M^t$. Then $v_i = \Sigma_{j \geq 1} u_j M_{ij}$ for every $i \geq 1$. Therefore,

$$\begin{aligned} v_i = 1 &\iff \exists j \geq 1, u_j = 1 \;\&\; M_{ij} = 1 \\ &\iff \exists j \geq 1, u_j = 1 \;\&\; (i, j) \in r. \end{aligned}$$

We can now invoke the result of Seiferas and McNaughton to deduce the desired conclusion. □

There is, of course, not much substance in this proposition. However, it does suggest a much deeper question: *is there an "internal" characterization of infinite Boolean matrices which transform u.p. vectors to u.p. vectors?*

By rephrasing the fact that the exponential function is regularity-preserving, we see that the relation defined by $(i, j) \in r$ if and only if $j = 2^i$ transforms u.p. vectors to u.p. vectors. However, there doesn't seem to be anything "regular" about the corresponding infinite Boolean matrix.

The desired internal characterization should further generalize Theorem 4.1. It should provide a way for us to recognize infinite Boolean matrices that transform u.p. vectors to u.p. vectors by inspecting properties of their entries. We have to leave the issue unresolved.

7 Concluding remarks

We have successfully applied MAAT, a matrix-based approach to automata theory, to the study of regularity-preserving functions. This approach can be considered as a drastic generalization of the powerset construction [20], or what is often referred to as the "pebble" game. An important result proved through MAAT is the characterization of regularity-preserving functions in terms of their ultimate periodicity with respect to powers of Boolean matrices. By virtue of the result, many properties of regularity-preserving functions – both old and new – have been treated in an uniform way, and Kosaraju's subtraction problem has been settled.

Since our development is largely independent of the inner workings of Boolean matrices, one wonders if a theory of ultimate periodicity can be developed over finite semirings, or other algebraic structures in general.

The application of MAAT to other areas of automata theory has not been fully explored, although Kleene's theorem can be treated in a concise manner without going through non-deterministic finite automata. We believe that MAAT should be able to shed light on other topics as well, such as minimization or even regular ω-languages. It remains to be seen if MAAT will be useful in areas related to semilinear sets, including Presburger arithmetic, monadic second order arithmetic with successor, and learning theory.

Acknowledgments. I'd like to thank Bob Robinson for valuable feedback on an earlier version and pointing out to me the connection stated as Proposition 5.2. Rod Canfield was responsible for getting me interested in regularity-preserving functions; discussions with him are much appreciated. Finally, thanks go to John Kececioglu for keeping the CATS seminar going; various versions of the paper were presented at the seminar.

References

[1] N. Abe. Characterizing PAC-learnability of semilinear sets. *Inform. and Comput.* 116, no. 1, 81–102, 1995.

[2] S. L. Bloom and Z. Ésik. Equational axioms for regular sets. *Math. Structures Comput. Sci.* 3, 1–24, 1993.

[3] J. Brzozowski. Open problems about regular languages. In R.V. Book (eds). *Formal Language Theory: perspectives and open problems*. Academic Press, 1980.

[4] J. Büchi. On a decision method in restricted second order arithmetic. In *Proc. Internat. Congr. Logic, Method, and Philosophy of Sci. 1960*, pp 1-12 Stanford University Press, 1962.

[5] J.H. Conway. *Regular algebra and finite machines*. chapman and Hall, London, 1971.

[6] S. Eilenberg. *Automata, languages, and machines*. Vol. 58 and 59, in the series *Pure and Applied Mathematics*. Academic Press, New York, 1974.

[7] F. Gécseg. *Products of automata.* EATCS Monographs on Theoretical Computer Science, vol 7. Springer-Verlag, Berlin-New York, 1986.

[8] J.E. Hopcroft, J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[9] K. H. Kim. *Boolean Matrix Theory and Applications*. Marcel Dekker, INC., New York, 1982.

[10] S.C. Kleene. Representation of events in nerve nets and finite automata. *Automata Studies*, 3–42, Princeton Univ. Press, 1956.

[11] S.R. Kosaraju. Regularity preserving functions. *SIGACT News*, 6 (2), 16–17, 1974.

[12] S.R. Kosaraju. Correction to "Regularity preserving functions". *SIGACT News*, 6 (3), 22, 1974.

[13] D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. *Information and Computation*, 110:366–390, 1994.

[14] D. Kozen. On regularity-preserving functions. Technical Report TR95-1559, Department of Computer Science, Cornell University, 1995.

[15] D. Kozen. *Automata and Computability*. Springer-Verlag, 1997.

[16] D. Krob. Complete systems of B-rational identities. Theoret. Comput. Sci. 89, 207–343, 1991.

[17] W. Kuich and A. Salomaa. *Semirings, Automata, Languages*. Springer-Verlag, Berlin, 1986.

[18] D. Perrin. Finite Automata. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, vol B. 133-191. MIT Press and Elsiver, 1990.

[19] J.-E. Pin. *Varieties of formal languages*. Plenum Publishing Corp., New York, 1986.

[20] M. Rabin, D. Scott. Finite automata and their decision problems. *IBM J. Res.*, 3:2, 115–125, 1959.

[21] A. Salomaa. Two complete axiom systems for the algebra of regular events. *J. Assoc. Comput. Mach.*, 13 (1), 158 – 169, 1966.

[22] J.I. Seiferas, R. McNaughton. Regularity preserving relations. Theoretical Computer Science 2, 147–154, 1976.

[23] D. Siefkes. Decidable extensions of monadic second order successor arithmetic. In *Automaten-theorie und formale Sprachen*, J. Dorr and G. Hotz (eds), 441–472, Mannheim, 1970.

[24] R.E. Stearns and J. Hartmanis, Regularity preserving modifications of regular expressions. *Information and Control*, 6 (1), 55–69, 1963.