CDMTCS Research Report Series

Algebraic Constraints, Automata, and Regular Languages (Revised Version)

Bakhadyr Khoussainov
Department of Computer Science
University of Auckland

CDMTCS-126
November 2001

Centre for Discrete Mathematics and
Theoretical Computer Science

On Algebraic and Logical Specifications of Classes of Regular Languages

Bakhadyr Khoussainov[1]

Abstract

The paper studies classes of regular languages based on algebraic constraints imposed on transitions of automata and discusses issues related to specifications of these classes from algebraic, computational and logical points of view.

Key Words: Finite Automata, Tree Automata, Universal Algebra, Computably Enumerable Sets, Σ_1–Algebra, Π_1–Algebra, Finitely Presented Algebra, Regular Language, Homomorphism, Congruence Relation, Isomorphism.

1 Introduction

In this paper we develop a theory devoted to investigating issues of specifying classes of regular languages over a given signature (alphabet). By a regular language we mean one recognized by either a finite automaton or a tree automaton. In the former case the underlying language consists of strings, in the latter case the underlying language consists of trees, or generally, ground terms. Two natural questions arise immediately. What classes of regular languages do we want to specify? How can we specify a given class of regular languages?

To answer the first question, we use (universal) algebra. It is well known that finite deterministic automata can be viewed as finite unary algebras. Similarly, tree automata can be viewed as finite universal algebras [2] [6]. This observation suggests the idea of considering those automata whose underlying sets of states form natural algebraic structures. Thus, these structures can, for example, be defined by universally quantified systems of formulas of the first order or other logics. Examples of such structures are groups, lattices, rings, boolean algebras or semigroups. When the formulas are of the form of equations or conditional equations, the corresponding classes of automata are well-behaved in the sense that they are closed under known automata-theoretic and algebraic constructions such as Cartesian

[1]Computer Science Department, The University of Auckland, New Zealand. E-mail: bmk@cs.auckland.ac.nz

products, homomorphisms, complementations, etc. From a computational point of view a given set of formulas can be thought of as algebraic and logical constraints put on transitions of finite state systems. To illustrate this, consider a run of a finite state system occurring during the execution of program instructions. The run is simply a sequence of states. There are usually certain constraints specified by the system software and hardware that the run must satisfy. Some of these constraints are often of an algebraic and logical nature. For example, during the run, two consecutive executions of an instruction I can produce the same result obtained by an execution of another instruction J. This can algebraically be presented as an equation $II = J$. Constraints that force two instructions, say I and J, to be executed in parallel, can be presented by an algebraic equation $IJ = JI$ which is a natural algebraic presentation of parallelism. More generally, an algebraic (or logical) expression of the type $I = J \rightarrow T = S$, can be understood as a constraint with the following meaning. Whenever the executions of the instructions I and J produce the same result then the results of executions of instructions T and S coincide. All these considerations motivate the idea of studying regular languages recognized by automata whose transitions satisfy universally quantified sets of formulas, in particular sets equations or more generally conditional equations. This view also suggests a fruitful ground for the interplay between tree or finite automata and the concepts of algebra, e.g. finitely presented algebra, free algebra, equations, conditional equations.

To answer the second question, suppose that we are given a class of languages. In the theory of formal languages, a traditional question that arises about the class is whether or not the class can be specified in an appropriate terminology. For example, the class of all finite automata or tree automata recognizable languages can be specified as the class of languages determined by regular expressions of appropriate types. Similarly, the class of all pushdown automata recognizable languages can be thought of as being specified by context free grammars. There has also been research in characterizing other known classes of languages, e.g. classes of problems decidable in polynomial time, by using formal systems of the first order logic and its extensions. Thus, the notion of specification is a general concept, and each time when one talks about a specification this notion should be given a precise formalization. As our approach to defining classes of regular languages is algebraic and uses the language of the first order logic (e.g. systems of universally quantified equations or conditional equations), our specification of classes of regular languages will also be of an algebraic and logical nature.

The basic idea is twofold. On the one hand, we will concentrate on specifying the classes of regular languages by the isomorphism types of certain algebras naturally induced by the classes. In particular, we will show the uniqueness of such algebras. On the other hand, we will use the logical language to investigate whether or not a given class of regular languages can possess a first order definition in a certain precise sense.

Here is a brief outline of the paper. The paper consists of three parts. In the first part, Section 2, we present basic definitions and results concerning automata, their languages and introduce classes of regular languages defined by conditional equations. The proofs of results in this part are relatively easy, and generally follow classical automata theory. However, we provide the proofs as we would like to give an intuition for the reader and make the paper self-contained. In the second part, Section 3, we study algebras naturally associated with the classes of regular languages. This will be done by using well-known concepts in universal algebra, e.g. finitely presented algebra and residually finite algebra. We will show that among all these algebras there is one unique up to isomorphism induced by any given class of regular languages. We call this algebra the canonical algebra of the class. The idea here is that the canonical algebras can be thought of as purely algebraic specifications of the classes of regular languages. The section ends with a study of computability theoretic properties of canonical algebras. In the last part, Section 4, we discuss issues related to specifying the classes of regular languages by using conditional equations with an emphasis on equations. This approach will lead us to natural interactions between the equational specifications, formal languages and the theory of effective algebras. Ideologically, our approach in this part of the paper is related to the approach of Bergstra and Tucker on specifications of abstract data types from [1]. However, our approach is based on the study of classes of regular languages rather than abstract data types. As a consequence our definitions, results and questions are obtained in rather different settings (see for example Comment 1 and Comment 2 in Section 4.2). Finally, in the paper we will discuss and motivate some of our definitions and theorems and relate them to known results where possible.

We assume that the reader is familiar with the basics of finite automata, tree automata, regular languages [9], view of finite automata and tree automata as finite algebras [2], basics of the theory of universal algebras, e.g. finitely presented algebra, free algebra, congruence relations [7]. In addition, we use notions from computability theory [15], e.g. computably enumerable (c.e.) set, simple set, immune set; and computable algebra [14], e.g. Σ_1–algebra, Π_1–algebra. Many of these notions will be defined as needed. A related paper discussing complexity issues is [3].

2 Automata with Algebraic Constraints

This section is introductory and provides basic definitions and results. Some of the results use standard constructions from automata theory but to our knowledge not explicitly stated in the literature. We present short proofs of these results to make the paper self-contained and give a basic intuition to the reader.

2.1 Basic Definitions

In this section, using terminology from universal algebra, we recall definitions of automata, regular languages, and introduce the concept of automata with algebraic constraints. Throughout the paper we fix the **signature** $\sigma =< f_1, \ldots, f_n, c_1, \ldots, c_m >$, where $c_1, \ldots, c_m$ are **constant symbols** , and $f_1, \ldots, f_n$ are **function symbols**. An **algebra** $\mathcal{A}$ of this signature is a system $< A, f_1, \ldots, f_n, c_1, \ldots, c_m >$, where A is a non-empty set called the **domain** of the algebra, each f_i is an operation on A and each c_j is a constant that interpret the appropriate symbols of the signature[2]. The algebra is **finite** if its domain A is finite. From now on all algebras we consider will be assumed to be generated by constants c_1, …, c_n unless explicitly stated otherwise.

The **terms** of σ are defined by induction: each variable x and constant c_j are terms; if $t_1, \ldots, t_k$ are terms and f is a k–ary function symbol then $f(t_1, \ldots, t_k)$ is a **term**. The set G of **ground terms** is the set of all terms without variables. Each ground term g defines a **finite labeled tree** t_g as follows: the leaves of the tree are labeled with the constants, other nodes are labeled with the function symbols, and any node labeled with symbol f of arity k has exactly k immediate successors. Thus, if $g = f(g_1, \ldots, g_n)$ then t_g can be constructed as follows. The root of the tree is labeled with f, the root has exactly n immediate successors ordered from left to right, and each ith successor is the root of the tree t_{g_i} for $i = 1, \ldots, n$.

[2]We abuse notation and denote the function (constant) symbols and their interpretations with the same letters.

Definition 2.1 *A* **language** *is a subset of the set G of ground terms.*

If one identifies the ground terms g with trees t_g then any language can be thought as a set of trees. A basic notion of this paper is the following.

Definition 2.2 *A* **finite automaton** *is a pair $M = (\mathcal{A}, F)$ consisting of a finite algebra $\mathcal{A}$ and the set $F \subseteq A$. The elements of A are called* **states**, *F is called the* **set of final states** *, and the constants $c_1, \ldots, c_m \in A$ are the initial states of M. The* **algebra associated with** *M is $\mathcal{A}$.*

If $m = 1$ then M is of course a standard deterministic finite automaton over the alphabet $\{f_1, \ldots, f_n\}$.

From now on all automata will be assumed to be finite. Let g be a ground term and $M = (\mathcal{A}, F)$ be an automaton. The automaton M evaluates g in a natural way: it is simply the value of the term g in $\mathcal{A}$. Procedurally this can be thought as follows. Think of g as the labeled tree t_g. The leaves of t_g are values of the constants of the signature in the associated algebra $\mathcal{A}$. These are the initial states of M. If a node of the tree is labeled with f and the values of the immediate successors of the node are states $s_1, \ldots s_k$ then label the node with the state $f(s_1, \ldots, s_k)$. Thus the automaton works from the leaves to the root of t_g, and labels the nodes with states of M. The root is then labeled with the state which is the value of g in the algebra $\mathcal{A}$.

Definition 2.3 *The automaton $M = (\mathcal{A}, F)$* **accepts** *the ground term g if the value of g in $\mathcal{A}$ is in F. Let $L(M)$ be the set of all ground terms accepted by M. The language $L(M)$ is called a* **regular** *language.*

Any regular language is a decidable language. Moreover, it is known that the class of all regular languages is a **Boolean class**, that is closed under the set–theoretic operations of union, intersection, and the complementation. Below, using the concept of algebraic constraint, we provide some other examples of Boolean classes of regular languages.

Definition 2.4 *A* **conditional constraint** *is the universal closure of a formula of the type $t_1 = q_1$ & $\ldots$ & $t_n = q_n \rightarrow t = q$, where t_i, q_i, t and q are terms of the signature. An* **equational constraint** *is the universal closure of a formula of the type $t = q$, where t and q are terms.*

Clearly, every equational constraint is also a conditional constraint. The idea behind this definition is that we want to consider those automata whose transitions satisfy constraints which are of algebraic nature. We formalize this as follows.

Definition 2.5 *Let C be a set of conditional constraints and let $M = (\mathcal{A}, F)$ be an automaton. The algebra $\mathcal{A}$ is a C–**algebra** if it satisfies all the formulas from C. The automaton $M = (\mathcal{A}, F)$ is a C–**automaton** if $\mathcal{A}$ is a C–algebra. The language accepted by a C–automaton is a C–**language**. Define R_C to be the set of all C–languages.*

In order to explicitly distinguish conditional constraints from equational ones, we use the letter E to denote sets of equational constraints. Thus, by replacing C with E, one can naturally talk about E-algebras, E-automata, E-languages, and the class R_E of all E-languages

2.2 Preliminary Results

Let C be a set of conditional constraints. Our goal is to study the class R_C of all C–languages. Note that C can be infinite, and moreover, every language from R_C is regular. Also note that R_C always contains G (the set of all ground terms) and $\emptyset$. We now prove that the class R_C is a Boolean class. The proof uses the standard constructions from automata theory for recognizing the union, intersection, and the complements of regular languages (see for example [6]), and we present the proof to provide some intuition to the reader. We also point out that the proof of this theorem uses a well-known fact from universal algebra that states that any class of algebras that satisfy a set C of conditional equations is closed under the Cartesian product operation and subalgebras.

Theorem 2.1 *The class R_C of all C–languages is closed under the operations of union, intersection, and complementation.*

Proof. Let L_1 and L_2 be C–languages. There exist finite C–automata $M_1 = (\mathcal{A}_1, F_1)$ and $M_2 = (\mathcal{A}_2, F_2)$ that accept L_1 and L_2, respectively. Let $a_1^1, \dots, a_m^1$, and $a_1^2, \dots, a_m^2$ be the values of the constant symbols $c_1, \dots, c_m$ in the algebras $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively. Consider the Cartesian product $\mathcal{A}_1 \times \mathcal{A}_2$ of the two algebras. This algebra contains the subalgebra generated by the pairs $(a_1^1, a_1^2), \dots, (a_m^1, a_m^2)$. Denote this subalgebra by $\mathcal{A}$. Thus, the algebra $\mathcal{A}$ is an algebra of the given signature. The algebra $\mathcal{A}$ satisfies all the algebraic constraints from C because the algebras $\mathcal{A}_1$ and $\mathcal{A}_2$ do so. Hence $\mathcal{A}$ is a C–algebra. Now consider the following two automata, $(\mathcal{A}, A \cap (F_1 \times F_2))$ and $(\mathcal{A}, A \cap (F_1 \times A_2 \cup A_1 \times F_2))$. Both automata are C–automata. The first automaton accepts the language $L_1 \bigcap L_2$, and the second one accepts the language $L_1 \bigcup L_2$. The automaton $(\mathcal{A}_1, A_1 \setminus F_1)$ accepts the complement of L_1 and is clearly a C–automaton. The theorem is proved.

We present one more theorem that shows a difference between the classes of R_C and R_E. The difference exploits the fact that, as opposed to conditional constraints, equational constraints are preserved under homomorphisms. Let $\mathcal{M} = (\mathcal{A}, F)$ be an automaton. A **homomorphism** of M onto an automaton $M_1 = (\mathcal{A}_1, F_1)$ is a mapping h from $\mathcal{A}$ onto $\mathcal{A}_1$ such that h preserves the basic operations and for all states $s \in A$, $s \in F$ if and only if $h(s) \in F_1$. Note that in this case M and M_1 accept the same language. Equational constraints are always preserved under homomorphisms. Recall that a **minimal automaton** for a regular language L is the automaton with the fewest states that accepts L.

Theorem 2.2 *Let L be an E–language. Then a minimal automaton for L is unique and is an E–automaton.*

Proof. The following are known facts (see for example [6]). Any regular language L has a minimal automaton accepting it. Moreover, the automaton is unique up to isomorphism. Additionally, any automaton that accepts L can be homomorphically mapped onto the minimal automaton. So let M_1 be the minimal automaton for L. Since L is an E–language there exists an E–automaton M that accepts L. Since M_1 is minimal, the automaton M_1 is a homomorphic image of M. Thus, M_1 is an E–automaton since equational constraints are preserved under homomorphisms. The theorem is proved.

The theorem can not be strengthen by replacing equations with conditional equations. Here is a counterexample. Consider the signature $< f_1, f_2, c >$, where f_1 and f_2 are unary function symbols. Consider the language $\{f_1^n f_2^m \mid n, m \in \omega\}$. The minimal automaton recognizing this language is $\mathcal{M} = (\mathcal{A}, F)$, where $A = \{0, 1, 2\}$, 0 is the initial state, $F = \{0, 1\}$, and $f_1(0) = 0$, $f_2(0) = f_2(1) = 1$, $f_1(1) = 2$, $f_1(2) = f_2(2) = 2$. Let C be the set consisting of the following conditional equations:

$$\forall x \forall y (f_1(f_2(0)) = f_1(f_1(f_2(0)) \to x = y)$$
$$\forall x \forall y (f_1(f_2(0)) = f_2(f_1(f_2(0)) \to x = y).$$

Clearly $\mathcal{M}$ is not a C-automaton. However, the language L is a C-language and there are two nonisomorphic minimal C-automaton $\mathcal{M}_1 = (\mathcal{A}_1, F_1)$ and $\mathcal{M}_2 = (\mathcal{A}_2, F_2)$ accepting L. $\mathcal{M}_1 = (\mathcal{A}_1, F_1)$ is defined as follows:

$A_1 = \{0,1,2,3\}$, 0 is the initial state, $F_1 = \{0,1\}$, and $f_1(0) = 0$, $f_2(0) = 1$, $f_1(1) = 2$, $f_2(1) = 1$, $f_1(2) = 2$, $f_2(2) = 3$, $f_1(3) = f_2(3) = 2$. $\mathcal{M}_2 = (\mathcal{A}_2, F_2)$ is defined as follows: $A_2 = \{0,1,2,3\}$, 0 is the initial state, $F = \{0,1\}$, and $f_1(0) = 0$, $f_2(0) = 1$, $f_1(1) = 2$, $f_2(1) = 1$, $f_1(2) = 3$, $f_2(2) = 2$, $f_1(3) = f_2(3) = 2$.

A natural relation defined by the the set C of conditional constraints is the following. Ground terms t and q are C**–equivalent** if the equality $t = q$ can be proved (in the first order logic) from C. We denote C–equivalent terms t and q by $t \sim_C q$. We single out this equivalence relation in the following definition:

Definition 2.6 *For a set C of conditional constraints, define*

$$\sim_C = \{(p,q) \mid \ C \textit{ proves } p = q\}.$$

The following lemma follows immediately.

Lemma 2.1 *The relation $\sim_C$ is a computably enumerable relation with an oracle for C. In particular, if C is a decidable set then $\sim_C$ is a c.e. relation.*$\Box$

For a set C of algebraic constraints any C-language possesses a natural C-closeness property with respect to the relation $\sim_C$. Formally, a language L is C**–closed** if for all $t, q \in G$ the condition $t \in L$ and $t \sim_C q$ implies that $q \in L$. Thus, any C-closed language is a union of some $\sim_C$–equivalence classes. These considerations now imply the following result.

Corollary 2.1 *For any set C of conditional constraints all C–languages are C–complete. Similarly, for any set E of equational constraints all E–languages are E–complete.* $\Box$

2.3 The Global Structure of R_E–Classes

In this subsection we study the global structure of all R_E–classes, that is, we investigate the set $K = \{R_E \mid E \text{ is a set of equational constraints}\}$. The set K forms a natural partially ordered set $\mathcal{K} = (K, \subseteq)$. Informally, $R_{E_1} \subset R_{E_2}$ represents the fact that the computations with constraints E_2 are more powerful than those with constraints E_1. Here is a simple lemma that states several properties of the partially ordered set $\mathcal{K}$.

Lemma 2.2 *1. The class $R_\emptyset$ is the maximum element of $\mathcal{K}$.*

2. *Let $E = \{\forall \bar{x}(t(\bar{x}) = c_1) \mid t(\bar{x})$ is a term$\}$. Then R_E is the minimum element of $\mathcal{K}$.*

3. *If $E_1 \subseteq E_2$ then $R_{E_2} \subseteq R_{E_1}$.*

Proof. For the first part it suffices to note that the class $R_\emptyset$ consists of all regular languages. For the second part, note that any E-algebra consist of one element only. Hence any E-automaton recognizes either the set G of all ground terms or the empty $\emptyset$. Thus, $R_E = \{\emptyset, G\}$. We have already mentioned that any class R_C contains $\emptyset$ and G. This proves the second part. For the last part note that any E_2-automaton is an E_1-automaton. Hence any E_2-language is an E_1-language. The lemma is proved.

It is not hard to see that it may be the case that $E_1 \bigcap E_2 = \emptyset$ but $R_{E_1} = R_{E_2}$. A trivial example would be $E_1 = \emptyset$ and $E_2 = \{\forall x(x = x)\}$. So the converse of the last part of the lemma above does not hold true. The next lemma shows that $\mathcal{K}$ is a complete lower lattice, that is any subset of K has the least upper bound.

Lemma 2.3 *Let X be a set of regular languages. Then there exists a minimal class $R \in K$ such that $X \subset R$.*

Proof. Consider the set $I = \{E' \mid X \subset R_{E'}\}$. The set I is not empty since $X \subseteq R_\emptyset$. Let $E = \bigcup_{E' \in I} E'$. We want to show that R_E is the desired class. It suffices to show that $R_E = \bigcap_{E' \in I} R_{E'}$. From part 3 of the lemma above we see that $R_E \subseteq R_{E'}$ for all $E' \in I$. Hence $R_E \subseteq \bigcap_{E' \in I} R_{E'}$. Now assume that $L \in \bigcap_{E' \in I} R_{E'}$. Hence for each $E' \in I$ there exists an E'–automaton $M(E')$ that recognizes L. By Theorem 2.2 we can assume that $M(E')$ is minimal. By the same theorem, all automata $M(E')$ are isomorphic to each other. Hence, $M(E')$ is in fact an E–automaton. Hence $L \in R_E$. This proves the lemma.

From this lemma we conclude that for all $R_{E_1}, R_{E_2} \in K$ there exists a minimal R_E such that $R_{E_1} \subseteq R_E$ and $R_{E_2} \subseteq R_E$. To see this let X be equal to $R_{E_1} \cup R_{E_2}$. We now combine these lemmas into the following theorem.

Theorem 2.3 *The partially ordered set $\mathcal{K}$ forms a complete lattice, where for all $R_{E_1}, R_{E_2} \in K$ the meet $R_{E_1} \wedge R_{E_2}$ coincides with $R_{E_1} \bigcap R_{E_2}$ and equals to $R_{E_1 \cup E_2}$, and the join $R_{E_1} \vee R_{E_2}$ is the minimal R_E that contains both R_{E_1} and R_{E_2}.* $\square$

3 Specifications by Isomorphism Types

The goal of this section is to provide a purely algebraic specification of the classes of regular languages defined by equational constraints. We introduce the notion of relative algebra for a given class R_E and study properties of relative algebras in relation to the class R_E. We give a precise meaning to the concept of specification by introducing the notion of character. The subsection will also show that the class R_E can uniquely be specified by the isomorphism type of a character called a canonical algebra. We will also study algebraic and computability-theoretic properties of the canonical algebras.

3.1 Characters and Canonical Algebras

The set G of all ground terms can naturally be transformed into the following algebra: for any functional symbol f of arity k and ground terms $t_1, \ldots, t_k$, the value of f on $(t_1, \ldots, t_k)$ is $f(t_1, \ldots, t_k)$. The algebra $\mathcal{F}$ thus obtained is called the **absolutely free algebra with generators** $c_1, \ldots, c_m$. We recall that an equivalence relation η on $\mathcal{F}$ is a **congruence** relation on $\mathcal{F}$ if for all $a_1, \ldots, a_k, b_1, \ldots, b_k \in G$ and a basic k–ary operation f, the condition $(a_1, b_1), \ldots, (a_k, b_k) \in \eta$ implies that $(f(a_1, \ldots, a_k), f(b_1, \ldots, b_k)) \in \eta$.

Let E be a set of equational constraints. It is not hard to see that the equivalence relation $\sim_E$ induced by the equational constraints E (see Definition 2.6) is a congruence relation of the absolutely free algebra $\mathcal{F}$. Factorizing $\mathcal{F}$ by $\sim_E$, we obtain the algebra called the **free algebra** $\mathcal{F}_E$ defined by E. The algebra $\mathcal{F}_E$ possesses several natural properties. Any algebra that satisfies E and whose generators are $c_1, \ldots, c_m$ is a homomorphic image of $\mathcal{F}_E$, and moreover this property defines $\mathcal{F}_E$ uniquely up to an isomorphism (see for example [7] and [13]).

Definition 3.1 *The algebra $\mathcal{F}_E$ is an* **initial algebra for the class** R_E.

From the properties of $\mathcal{F}_E$ mentioned above, we obviously obtain the following lemma:

Lemma 3.1 *For any E–automaton $M = (\mathcal{A}, F)$, the algebra $\mathcal{A}$ is a homomorphic image of $\mathcal{F}_E$. Moreover, if $\mathcal{F}_{E_1}$ is isomorphic to $\mathcal{F}_{E_2}$ then $R_{E_1} = R_{E_2}$.* $\square$

This lemma suggests the idea of specifying the class R_E by the isomorphism type of the initial algebra $\mathcal{F}_E$. This idea does not work because there are examples of nonisomorphic $\mathcal{F}_{E_1}$ and $\mathcal{F}_{E_2}$ such that $R_{E_1} = R_{E_2}$. Indeed, take for example two infinite algebras $\mathcal{A}_1$ and $\mathcal{A}_2$ with no nontrivial congruence relations. Let E_1 and E_2 be the set of all equations satisfied by $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively. Now note that any (finite) E_1 or E_2-algebra contains exactly one element. Hence $R_{E_1} = R_{E_2} = \{\emptyset, G\}$. Now we refine the idea of characterizing the class R_E by the isomorphism types of algebras by introducing the following new notions.

Definition 3.2 *For an algebra $\mathcal{A}$ define the set $FH(\mathcal{A})$ to be the set containing the isomorphism types of all finite homomorphic images of $\mathcal{A}$.*

For example, consider the algebra $\mathcal{A} = (\omega, 0, S)$, where $\omega = \{0, 1, 2, \ldots\}$ and $S(x) = x + 1$. Any homomorphic image of this algebra is of the form $\mathcal{A}_{n,k} = (\{0, 1, \ldots, n, n+1, \ldots, n+k\}, f)$, where $f(i) = i+1$ and $f(n+k) = n$ for $i \leq n+k-1$ and $n, k \in \omega$. Then $FH(\mathcal{A})$ has infinitely many elements and contains the isomorphism types of the algebras $\mathcal{A}_{n,k}$. The next definition "identifies" those algebras that have the same finite homomorphic images.

Definition 3.3 *Two algebras $\mathcal{A}$ and $\mathcal{B}$ are* **relative** *if $FH(\mathcal{A}) = FH(\mathcal{B})$.*

Thus, relative algebras can not be distinguished from each other by their finite homomorphic images. Relative algebras are not always isomorphic, as for example, any two algebras with no nontrivial congruences are relative. Now we prove the following theorem that shows usefulness of the notions introduced.

Theorem 3.1 *Two classes R_{E_1} and R_{E_2} coincide if and only if the initial algebras $\mathcal{F}_{E_1}$ and $\mathcal{F}_{E_2}$ are relative.*

Proof. Assume that the initial algebras $\mathcal{F}_{E_1}$ and $\mathcal{F}_{E_2}$ are relative. Take any language $L \in R_{E_1}$. There exists an E_1–automaton $M = (\mathcal{A}, F)$ that accepts the language. Then $\mathcal{A}$ is a homomorphic image of $\mathcal{F}_{E_1}$. Hence $\mathcal{A}$ must be a homomorphic image of $\mathcal{F}_{E_2}$ as well. We conclude that L is an E_2–language. Assume now that $R_{E_1} = R_{E_2}$. We want to show that $FH(\mathcal{F}_{E_1}) = FH(\mathcal{F}_{E_2})$. Suppose, without loss of generality, that there exists a finite homomorphic image $\mathcal{A}$ of $\mathcal{F}_{E_1}$ which does not belong to the set $FH(\mathcal{F}_{E_2})$. This implies that there exists an equation $t(x_1, \ldots, x_k) = q(x_1, \ldots, x_k)$ that is not satisfied in $\mathcal{A}$ such that the equation belongs to E_2. Let $a_1, \ldots, a_k$ be elements in $\mathcal{A}$ that make this equation false in $\mathcal{A}$. There exist ground terms $p_1, \ldots, p_k$ such that a_i equals to the value of the term p_i in $\mathcal{A}$. Let $F = \{t(a_1, \ldots, a_n)\}$. Consider the automaton $(\mathcal{A}, F)$. This automaton accepts the ground term $t(p_1, \ldots, p_k)$, but does not accept the term $q(p_1, \ldots, p_k)$. Let L be the language accepted by this automaton. Clearly, $t(p_1, \ldots, p_k) \in L$ and $q(p_1, \ldots, p_k) \notin L$. Since $R_{E_1} = R_{E_2}$ it must be the case that $L \in R_{E_2}$ as L, by the choice of $\mathcal{A}$, is accepted by an E_1–automaton. By Corollary 2.1 the language L is E_2–complete. Therefore $q(p_1, \ldots, p_k)$ must belong to L since the equality $t(x_1, \ldots, x_k) = q(x_1, \ldots, x_k)$ belongs to E_2. This is a contradiction. The theorem is proved.

Definition 3.4 *An algebra $\mathcal{A}$ is a* **character** *of the class R_E if $FH(\mathcal{A})$ consists of all E-algebras.*

Thus, for example the algebra $\mathcal{F}_E$ and, by the theorem above, any algebra $\mathcal{F}_{E'}$ that is relative to $\mathcal{F}_E$ are characters of the class R_E. We now give an example of relative but not isomorphic $\mathcal{F}_{E_1}$ and $\mathcal{F}_{E_2}$. Note that for these algebras $E_1 \neq E_2$. Let $\mathcal{A}_1 = (\omega, 0, S, S^{-1})$, where $S(x) = x + 1$ for all $x \in \omega$, and $S^{-1}(x) = x - 1$ if $x \geq 1$, and $S^{-1}(x) = 0$ if $x = 0$. Let $\mathcal{A}_2 = (Z, 0, S, S^{-1})$, where S^{-1} is the reverse of S in Z. Let E_1 and E_2 be the set of all equations satisfied in $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively. Clearly $E_1 \neq E_2$ as $S^{-1}(0) = 0$ holds in $\mathcal{A}_1$ but not in $\mathcal{A}_2$. However, both algebras are relatives and are characters of the class R_E, where $E = \{\forall x \forall y (x = y)\}$.

The next lemma shows that the notion of a character is complete in the sense that any algebra can be viewed as a character of some class R_E.

Lemma 3.2 *Any algebra is a character for some class R_E.*

Proof. Let $\mathcal{A}$ be an algebra. Consider the set $E(\mathcal{A})$ of all equations satisfied by $\mathcal{A}$. Then the algebra $\mathcal{A}$ is the initial algebra defined by $E(\mathcal{A})$. Therefore the algebra $\mathcal{A}$ is a character of the class $R_{E(\mathcal{A})}$. This proves the lemma.

Corollary 3.1 *Any two relative algebras are characters of the same class of regular languages. Particularly for any E, the initial algebra $\mathcal{F}_E$ and any algebra relative to $\mathcal{F}_E$ are characters of the class R_E.*

Proof. We note that no algebra can be a character of two distinct classes R_{E_1} and R_{E_2}. Now in order to prove the corollary, let $\mathcal{A}$ and $\mathcal{B}$ be relative algebras. Then $\mathcal{A}$ is a character for the class $R_{E(\mathcal{A})}$. Therefore $\mathcal{B}$

is a character of the class $R_{E(\mathcal{A})}$ since $\mathcal{B}$ is relative to $\mathcal{A}$. This proves the corollary.

For a given set E of equational constraints, consider the set $Ch(R_E)$ of all isomorphism types of algebras relative to $\mathcal{F}_E$. A natural question is whether one can define an algebra in the set $Ch(R_E)$ which, in certain sense, is a canonical character for R_E. One way to do this is the following. On $Ch(R_E)$ introduce the relation $\leq_h$: for all $\mathcal{A}, \mathcal{B} \in Ch(R_E)$, $\mathcal{A} \leq_h \mathcal{B}$ if and only if there exists a homomorphism from $\mathcal{B}$ onto $\mathcal{A}$. This relation is a partial order (because all algebras considered are generated by the constants). The next theorem shows that $(Ch(R_E), \leq_h)$ has a unique minimal element. Thus, one can say that the minimal element is the canonical character of the class R_E.

Theorem 3.2 *For any R_E there exists a character $\mathcal{C}_E$ of the class R_E such that every character of the class R_E is homomorphically mapped onto $\mathcal{C}_E$.*

Proof. Consider the absolutely free algebra $\mathcal{F}$. Consider the class of all finite E-algebras. This class coincides with the class of all finite homomorphic images of $\mathcal{F}_E$. Let

$$\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2 \dots$$

be a list of all these finite algebras from the class. Define the following equivalence relation $\sim_E^r$ on the set G of ground terms:

> Two terms t and q are $\sim_E^r$–equivalent, written $t \sim_E^r q$, if in the algebra $\mathcal{A}_i$ the equality $t = q$ holds for all i[3].

One now checks that $\sim_E^r$ is a congruence relation on $\mathcal{F}$. Hence factorizing $\mathcal{F}$ by $\sim_E^r$, we obtain the algebra which we denote by $\mathcal{C}_E$. We want to show that $\mathcal{C}_E$ satisfies the properties stated by the theorem. First we show that $\mathcal{C}_E$ is relative to the initial algebra $\mathcal{F}_E$. Let $\mathcal{B}$ be a finite algebra from $FH(\mathcal{F}_E)$. We define a mapping h from $\mathcal{C}_E$ to $\mathcal{B}$ as follows. Take an $a \in C_E$. There exists a ground term t whose value in $\mathcal{C}_E$ equals to a. Let b be the value of the ground term t in the algebra $\mathcal{B}$. Then, one can check that the mapping $h(a) = b$ is a homomorphism from $\mathcal{C}_E$ onto $\mathcal{B}$. Now we want to show that any finite homomorphic image of $\mathcal{C}_E$ is also a homomorphic image of $\mathcal{F}_E$. It suffices to show that $\mathcal{C}_E$ is is a homomorphic image of $\mathcal{F}_E$. Since

[3]Note that this equivalence relation does not necessarily coincides with $\sim_E$ defined in Definition 2.6. For example, if $\mathcal{F}_E$ contains no nontrivial congruences and is infinite (see for instance the algebra $\mathcal{F}_{E_1}$ provided right after Definition 3.4) then $\sim_E^r = \{(p, q) \mid p, q \in G\}$ and $\sim_{E_1}$ is clearly not equal to $\sim_E^r$.

$\mathcal{F}_E$ is the initial algebra for E, it suffices to prove that any equality $t = q$ between ground terms that is true in $\mathcal{F}_E$ is also true in $\mathcal{C}_E$. Let $t = q$ be an equality between ground terms that are true in $\mathcal{F}_E$. Then $t = q$ holds in every finite algebra $\mathcal{A}_i$. Hence, by the definition of $\sim_E^r$, the terms t and q are $\sim_E^r$–equivalent. Hence $t = q$ is true in $\mathcal{C}_E$. Therefore $\mathcal{C}_E$ is, in fact, a homomorphic image of $\mathcal{F}_E$. Hence any finite homomorphic image of $\mathcal{C}_E$ is also a homomorphic image of $\mathcal{F}_E$. This shows that $\mathcal{C}_E$ and $\mathcal{F}_E$ are relative algebras.

To prove the second part of the theorem we need to show that any algebra $\mathcal{B}$ relative to $\mathcal{F}_E$ can be homomorphically mapped onto $\mathcal{C}_E$. Let b be an element of $\mathcal{B}$. Take a term t whose value in $\mathcal{B}$ is b. Map b onto the value of the term t in $\mathcal{C}_E$. This mapping does not depend on the choice of t. Hence there exists a homomorphism from $\mathcal{B}$ onto $\mathcal{C}_E$. The theorem is proved.

The following definition is suggested by the theorem above:

Definition 3.5 *The* **canonical character** *of the class R_E of regular languages is the algebra $\mathcal{C}_E$ which is the minimal element of $(Ch(R_E), \leq_h)$.*

The next section studies some computational properties of the canonical characters for certain classes of R_E. The section provides a necessary and sufficient condition for the canonical character of R_E to coincide with the initial algebra $\mathcal{F}_E$.

3.2 On Canonical Characters

All the characters of the class R_E of regular languages that satisfy E are among homomorphic images of the algebra $\mathcal{F}_E$. Thus, the partially ordered set $(\{A \mid \mathcal{A} \leq_h \mathcal{F}_E\}, \leq_h)$ has the minimal element $\mathcal{C}_E$ and the maximal element $\mathcal{F}_E$. In this section we find conditions when $\mathcal{F}_E$ coincides with $\mathcal{C}_E$, and study some computability-theoretic properties of the canonical characters. To do this, we need to introduce a couple of notions from universal and computable algebra.

Definition 3.6 *An algebra $\mathcal{A}$ is* **residually finite** *if for all $a, b \in A, a \neq b$ there is a homomorphism h of $\mathcal{A}$ onto a finite algebra such that $h(a) \neq h(b)$.*

Residually finite algebras are fundamental in the study of universal algebra and play an important role in classifying and studying algebraic and algorithmic properties of algebraic structures (see for example [7], [8], [13]). We also refer the reader to an excellent survey [11] that includes results related to residually finite algebras. A few results in this subsection will naturally have an intersection with the results in the papers mentioned. However, in our study the use of residually finite algebras arises in a different setting and shows a new dimension of applications of residually finite algebras.

Now we introduce standard notions from computable algebra. Consider an algebra $\mathcal{A}$ of the signature σ generated by the constants $c_1, \ldots, c_n$. There is a congruence relation η on $\mathcal{F}$ such that $\mathcal{A}$ is isomorphic to the algebra obtained by factorizing $\mathcal{F}$ by η.

Definition 3.7 *The algebra $\mathcal{A}$ is a* Π_1**–algebra** *if the relation η is a complement of a c.e. set. Similarly, an algebra $\mathcal{A}$ is a* Σ_1**–algebra** *if the relation η is a c.e. set. If $\mathcal{A}$ is both a Σ_1–algebra and Π_1–algebra then $\mathcal{A}$ is a* **computable algebra**[4]

Examples of Σ_1–algebras are the initial algebras $\mathcal{F}_E$ for computably enumerable sets of constraints E. In general, it is not hard to obtain natural examples of Σ_1–algebras. These algebras have been studied in computable algebra, logic as well as in computer science (see for example [14] or [4] or [16]). We also point out that Σ_1-objects (in one or another sense) often arise in other areas of computer science, computability and logic. For example, Herbrand models of logic programs are Σ_1-objects, Lindenbaum Boolean algebras of computably enumerable theories (e.g. Peano arithmetic) are Σ_1-objects. However, there has not been much study of Π_1–objects mainly because of the small number of natural examples. It turns out that canonical characters are the source of natural examples of Π_1-algebras. Here is a simple result.

Lemma 3.3 *If the class of all finite homomorphic images of $\mathcal{F}_E$ is computably enumerable then the canonical character $\mathcal{C}_E$ for the class R_E is a Π_1–algebra.*

Proof. By the assumption, there exists a sequence $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots$ of all finite homomorphic images of $\mathcal{F}_E$ such that the set $\{(x, y) \mid x, y \in A_i\}$ is computably enumerable. Consider the congruence $\sim_E^r$ (see Theorem 3.2) that defines the canonical algebra $\mathcal{C}_E$. By the definition, $t \sim_E^r q$ if and only if for $\forall i(t = q \text{ in } \mathcal{A}_i)$. Therefore η is a Π_1–relation. We conclude that the algebra $\mathcal{C}_E$ is a Π_1–algebra. The lemma is proved. □

[4]In the literature, Σ_1–algebras are also called as computably enumerable, semicomputable or positive algebras.

Corollary 3.2 *For any finite set E, the canonical character $\mathcal{C}_E$ for the class R_E is a Π_1–algebra.*

Proof. The set E is finite. So, effectively list all finite algebras that satisfy E. These algebras are homomorphic images of $\mathcal{F}_E$. Hence the hypothesis of the lemma above holds true. Therefore $\mathcal{C}_E$ is a Π_1–algebra. The corollary is proved.

The next theorem gives a criteria as when the partially ordered set $(\{\mathcal{A} \mid \mathcal{A} \leq_h \mathcal{F}_E\}, \leq_h)$ has a unique element, that is when $\mathcal{F}_E = \mathcal{C}_E$.

Theorem 3.3 *For a given class R_E of regular languages, the initial algebra $\mathcal{F}_E$ is residually finite if and only if the algebras $\mathcal{F}_E$ and $\mathcal{C}_E$ coincide.*

Proof. Consider the class R_E. Assume that $\mathcal{F}_E$ is a residually finite algebra. We want to show that the minimal character $\mathcal{C}_E$ for the class R_E is isomorphic to $\mathcal{F}_E$. From the proof of Theorem 3.2, we know that $\mathcal{C}_E$ is a homomorphic image of $\mathcal{F}_E$. Let h be the homomorphism. We want to show that h is a one to one mapping. Indeed, let a, b be two distinct elements in $\mathcal{F}_E$. Then, there exist ground terms $t(p_1, \ldots, p_k)$ and $q(r_1, \ldots, q_s)$ such that the values of these terms in the algebra $\mathcal{F}_E$ are a and b, respectively. Since $\mathcal{F}_E$ is a residually finite algebra there exists a finite homomorphic image $\mathcal{A}_i$ of $\mathcal{F}_E$ in which the images of a and b are also distinct. Therefore the ground terms $t(p_1, \ldots, p_k)$ and $q(r_1, \ldots, r_s)$ are not $\sim_E^r$–equivalent, where $\sim_E^r$ is the congruence relation that defines the algebra $\mathcal{C}_E$ (see the proof of Theorem 3.2). Hence the mapping h must be a one to one mapping since $h(a) \neq h(b)$ by the definition of $\sim_E^r$.

Assume now that $\mathcal{F}_E$ and the minimal character $\mathcal{C}_E$ coincide. For the sake of contradiction, also assume that $\mathcal{F}_E$ is not residually finite. Hence there exist two distinct elements a and b in $\mathcal{F}_E$ such that in any finite homomorphic image of $\mathcal{F}_E$ the images of a and b are equal. Let $t(p_1, \ldots, p_k)$ and $q(r_1, \ldots, r_s)$ be ground terms whose values in $\mathcal{F}_E$ are a and b, respectively. Then the images of these elements in any finite homomorphic image of $\mathcal{F}_E$ are equal. Therefore, by the definition of the equivalence relation $\sim_E^r$, the ground terms $t(p_1, \ldots, p_k)$ and $q(r_1, \ldots, r_s)$ must be equal in the algebra $\mathcal{C}_E$. But this is not possible because $\mathcal{C}_E$ and $\mathcal{F}_E$ coincide. Contradiction. The theorem is proved.

Corollary 3.3 *For any finite set E, if the initial algebra $\mathcal{F}_E$ is residually finite then the minimal character $\mathcal{C}_E$ of the class R_E is a computable algebra.*

Proof. The initial algebra $\mathcal{F}_E$ and the canonical character $\mathcal{C}_E$ are isomorphic. Since E is finite, $\mathcal{F}_E$ is a Σ_1–algebra. By Corollary 3.2 the canonical character $\mathcal{C}_E$ is a Π_1–algebra. So, the algebra $\mathcal{F}_E$ is both a Σ_1–algebra and Π_1–algebra. Hence $\mathcal{F}_E = \mathcal{C}_E$, and $\mathcal{C}_E$ is a computable algebra. The corollary is proved.

4 Equational Specifications

In the previous two sections we introduced the notion of character as a tool to specify a given class R_E. This is an algebraic approach to the specification problem of the class R_E. As a dual to this algebraic approach, one can study the specification problem from computational and logical points of view as well. By its essence the isomorphism types of algebras are infinite objects. Therefore from a computational point of view it is quite natural to ask whether or not a given class of (regular) languages has some sort of finite formal specification. This sections deals with the question related to finding finite specifications for classes of regular languages from a logical point view.

4.1 Finite Equational Specifications

Let R be a class of regular languages. We would like to specify R by giving a finite definition to R using a formal system (e.g first order logic). For instance, assume that R consists of all languages recognized by automata of signature $< f_1, \ldots, f_n >$, where all f_i are unary, so that the automata can process the input symbols f_i and f_j at any given state with the same result. This class of regular languages can then be specified by the formula $\forall x(f_i(f_j(x)) = f_j(f_i(x))$.

There are two approaches in trying to find formal specifications of a given class R_E. The first approach consists of finding an E' such that E and E' have the same proof–theoretic power, that is $\sim_E = \sim_{E'}$. This essentially corresponds to the algebraic specification problem of Bergstra and Tucker [1] on specifying the algebra $\mathcal{F}_E$ without adding any additional sorts or expanding the original language. Of course if an E' is found such that $\sim_E = \sim_{E'}$, then $R_E = R_{E'}$. We single out such specifications in the following definition.

Definition 4.1 *The pair (R_E, E) has a* **a finite specification** *if there exists a finite E' for which $\sim_E = \sim_{E'}$.*

Clearly, this definition is primarily concerned with preserving the proof–theoretic power of E by finite means. The second approach consists of weakening this condition. Thus, for a given class R_E of regular languages we would like to find a finite E' such that $R_E = R_{E'}$. We formally define this approach in the following definition.

Definition 4.2 *The class R_E has a* **finite specification** *if for some finite set E' of equational constraint we have $R_E = R_{E'}$.*

Thus, the former definition is essentially a definition that requires the initial algebra $\mathcal{F}_E$ to be finitely presented in the variety of algebras satisfying the equation E. The latter definition weakens the former one and basically requires some relative of $\mathcal{F}_E$ to be finitely presented. It is not hard to find a pair (R_E, E) without any finite specification so that R_E has a finite specification. For example, take a non Σ_1–algebra $\mathcal{A}$ without nontrivial congruence relations. Let E be the set of all equations true in $\mathcal{A}$. The pair (R_E, E) does not have a specification (as Lemma 4.2 below shows) while R_E has a finite specification, e.g. $\{\forall x \forall y (x = y)\}$.

We present one simple example. Let R be the class consisting of all languages recognized by automata of the type $\mathcal{M} = (\mathcal{A}, F)$, where $\mathcal{A}$ is an algebra of the form $(\{0, 1, \ldots, n-1\}, 0, S, +, \times)$ with the $mod(n)$ addition $+$ and the $mod(n)$ product $\times$ operations. Then a finite specification E of this class R consists of the following equations:

$x + 0 = x;$
$x + S(y) = S(x + y);$
$x \times 0 = 0;$
$x \times S(y) = x + x \times y.$

Below we provide a theorem that gives examples of classes that have finite specifications. But first we need the following lemma. Recall from the previous section that $E(\mathcal{A})$ is the set of all equations true in $\mathcal{A}$.

Lemma 4.1 *Let $\mathcal{A}$ be a finite algebra, and $E(\mathcal{A})$ be the set of all equations satisfied by $\mathcal{A}$. Then the pair $(R_{E(\mathcal{A})}, E(\mathcal{A}))$ has a finite specification.*

Proof. To prove the lemma we introduce the notion of **height** $h(t)$ for ground terms t. The height is inductively defined as follows. The height of any constant term c, $h(c)$, is 0. If the heights $h(t_1), \ldots, h(t_m)$ have been defined, then $h(f(t_1,\ldots,t_m)) = max\{h(t_i) \mid i = 1,\ldots,m\} + 1$. Since the algebra $\mathcal{A}$ is finite, it is not hard to see that there exists a minimal s such that every term of height s equals, in the algebra $\mathcal{A}$, to a term whose height is less than s. The number of terms of height $\leq s$ is finite. Define

$$E' = \{t = q \mid h(t), h(q) \leq s \text{ and the algebra } \mathcal{A} \text{ satisfies the universal closure of the equality } t = q\}.$$

Note that E' is finite. Now $\mathcal{F}_{E'}$ is isomorphic to the algebra $\mathcal{A}$. Therefore $\sim_{E(\mathcal{A})} = \sim_{E'}$. The lemma is proved.

Theorem 4.1 *For any finite set X of regular languages, the minimal class $R(X) \in K$ that contains X has a finite specification.*

Proof. Note that by Lemma 2.3 the class $R(X)$ exists. Let $X = \{L_1,\ldots,L_k\}$. Consider the minimal automaton $M_i = (\mathcal{A}_i, F_i)$ that accepts L_i, $i = 1,\ldots,k$. Consider the congruence relation η_X on $\mathcal{F}$ defined as follows: $(t,q) \in \eta_X$ iff $t = q$ in $\mathcal{A}_i$ for all $i = 1,\ldots,k$. Let $\mathcal{F}(X)$ be the algebra obtained by factorizing $\mathcal{F}$ by η_X. The algebra $\mathcal{F}(X)$ is the minimal algebra with respect to $\leq_h$ in the class of all algebras $\mathcal{A}$ such that $\{\mathcal{A}_1,\ldots,\mathcal{A}_k\} \subset FH(\mathcal{A})$. Note that $\mathcal{F}(X)$ is isomorphic to the subalgebra $\mathcal{B}$ generated by the constants of to the Cartesian product $\mathcal{A}_1 \times \ldots \times \mathcal{A}_k$ because $(t,q) \in \eta_X$ if and only if $t = q$ holds in $\mathcal{A}_1 \times \ldots \times \mathcal{A}_k$. Hence $\mathcal{F}(X)$ is finite. Thus, from the lemma above we conclude that the theorem is proved.

Now we provide some simple facts that give us necessary conditions for a class R_E or pair (R_E, E) to have finite specifications.

Lemma 4.2 *For a class R_E the following are true:*

1. *If the pair (R_E, E) has a finite specification E' then the algebra $\mathcal{F}_E$ is a Σ_1–algebra. Moreover, if $\mathcal{F}_E$ is residually finite then $\mathcal{F}_E$ is a computable algebra.*

2. *If R_E has a finite specification then the set $\{\ A \mid \mathcal{A}$ is an E-algebra$\}$ is decidable. Hence the canonical algebra of $\mathcal{R}_E$ is a Π_1-algebra.*

Proof. For part one, note that the algebras $\mathcal{F}_E$ and $\mathcal{F}_{E'}$ are isomorphic. Since E' is finite, the congruence relation $\sim_{E'}$ is a c.e. relation. Hence the algebra $\mathcal{F}_E$ is a Σ_1–algebra. If $\mathcal{F}_E$ is residually finite then, by Corollary 3.3, the algebra $\mathcal{F}_E$ is computable.

For part two let $R_E = R_{E'}$ for some finite E'. Then a finite algebra is an E-algebra if and only if it is an E'-algebra. Checking whether or not a finite algebra satisfies E' is clearly decidable. The rest is proved in Corollary 3.2. The lemma is proved.

Corollary 4.1 *If (R_E, E) has a finite specification and $\mathcal{F}_E$ is not computable then $\mathcal{F}_E$ is not residually finite.* $\Box$

The results above lead us to the following question. Does the pair (R_E, E) have a finite specification if the initial algebra $\mathcal{F}_E$ is computable and residually finite? The theorem below answers the question.

Theorem 4.2 *There exists an E such that $\mathcal{F}_E$ is computable and residually finite but the pair (R_E, E) does not have a finite specification.*

Proof. Consider the signature is $< f_1, f_2, c >$, where f_1, f_2 are unary function symbols. Define the congruence relation η on $\mathcal{F}$ as follows: $t\eta q$ iff $t = q$ or $h(t) = h(q) = 2^n$ for some n. It is not hard to see that the algebra $\mathcal{A}$, obtained by factorizing $\mathcal{F}$ by η, is computable. Moreover, one can check that $\mathcal{A}$ is a residually finite algebra. Consider $E = E(\mathcal{A})$, the set of all equations true in $\mathcal{A}$. We claim that the pair (R_E, E) does not have a finite specification. To show this we analyze the equations true in $\mathcal{A}$. Let the universal closure of the equation $t = q$ be true in $\mathcal{A}$. Then, from the definition of $\mathcal{A}$, one can see that $h(t) = h(q)$. Suppose t and q contain variables x and y, respectively. So we write $t(x)$ and $q(y)$ instead of t and q. Then $x = y$, otherwise, as easily seen, the equation would not be true in $\mathcal{A}$. We claim that if $x = y$ then the terms t and q are in fact (syntactically) equal terms. There exists an n such that the height of t and q are equal to 2^n. Otherwise, the equality $t(c) = q(c)$ would not be true in the algebra. Let m be any positive number less than 2^{n+1}. Then, since the universal closure of $t(x) = q(x)$ is true in $\mathcal{A}$, the equation $t(f_1^m(c)) = q(f_1^m(c))$ is also true in $\mathcal{A}$. By the definition of $\mathcal{A}$, this is not possible. Also, it is not the case that only one of the terms t, q contains a variable. Now assume that for some finite E' we have $\mathcal{F}_E = \mathcal{F}_{E'}$. Then, as we have already proved, we can assume that no equation $t = q$ in E' contains a variable. Set $s = max\{h(t) | t = q \in E'\}$. Let $r = 2^s$. Then the equality $f_1^r(c) = f_2^r(c)$ can not be derived from E'. This is a contradiction. The theorem is proved.

It turns out that the class R_E constructed in the proof of the theorem above gives us a stronger statement.

Corollary 4.2 *There exists a class R_E without a finite specification so that the canonical character $\mathcal{C}_E$ is computable and residually finite.*

Proof. Consider E defined in the theorem above. Assume that for some finite E', $R_E = R_{E'}$. Then it must be the case that $\sim_{E'} \subset \sim_E$. Hence for all $(t, q) \in E'$, the height of t equals to the height of q. Since E' is finite there exist two terms t, q such that $(t, q) \notin E'$ but $(t, q) \in E$. Then there exists a homomorphic finite image of $\mathcal{F}_{E'}$ in which t and q are also distinct. Hence $\mathcal{F}_{E'}$ is not relative to $\mathcal{F}_E$. Therefore $R_E \neq R_{E'}$ by Theorem 3.1. The corollary is proved.

4.2 Expansionary Specified Classes

Theorem 4.2 and its corollary show that it is not always possible to find a finite specification for a class R_E even when the initial algebra $\mathcal{F}_E$ is computable and residually finite. This motivates us to consider the idea of refining the notion of finite specification. We do this by considering expansions of the original language with the goal of increasing the expressive power of our language[5]. An **expansion** of the signature σ is obtained by adding finitely many new function symbols to the signature. The goal here is to have more powerful language than the original one and thus to attack the specification problem by means of additional tools but within the first order logic. These tools are new functional symbols and their interpretations in algebras. If $\mathcal{A}$ is an algebra of σ then by taking interpretations of the new function symbols in the domain A, we obtain a new algebra $\mathcal{B}$ which is called an **expansion** of $\mathcal{A}$. Then the original algebra $\mathcal{A}$ is called a σ–**reduct** of the expansion. Thus, one can think of new functions as those that were hidden from us when we used the original language

Definition 4.3 *Let σ_1 be an expansion of the signature σ. Let E, E_1 be sets of algebraic constraints of the signatures σ, σ_1 respectively. Then R_{E_1} is a* **refinement** *of R_E if R_{E_1} is infinite and the σ–reduct of any E_1–algebra is an E–algebra.*

The basic motivation for this definition is to give a finite specification for an infinite subset of R_E by using an expansion of the original signature

[5]We note that considering expansions of the original language is a standard and powerful method often used in classical model theory, modern finite model theory, computable model theory and algebra, and the theory of algebraic specifications.

in case when R_E can not be specified in its own language. In other words, the aim is to weaken the original specification problem for classes of regular languages in two ways. On the one hand, we allow to use expanded signature, and hence to possess more syntactic and expressive power. On the other hand, instead of trying to specify the whole class R_E we would like to choose a non-trivial and sufficiently rich (that is infinite) subclass of R_E that can be specified in some expanded signature. We now formalize this concept in the following definition:

Definition 4.4 *A class R_E of regular languages is* **expansionary specified** *if there exists a refinement R_{E_1} of R_E such that R_{E_1} has a finite specification.*

Thus, our specification problem asks whether or not a given class R_E of regular languages possesses an expansionary specification. From this definition we obtain the following proposition that gives us a necessary condition for a class R_E to have an expansionary specification.

Proposition 4.1 *If R_E is expansionary specified then there exists a Σ_1–algebra which is initial for some refinement of R_E.* $\Box$

In relation to the introduced concepts, we would like to make the following two comments.

Comment 1. In [1] it is proved that for any computable algebra $\mathcal{A}$ there exists an expansion $\mathcal{A}^\star$ of $\mathcal{A}$ so that $\mathcal{A}^\star$ is isomorphic to $\mathcal{F}_E$ for some finite set E of equations of the expanded signature. We do not know whether or not this result can be strengthen so that every algebra in $FH(\mathcal{A})$ is a reduct of some algebra in $\mathcal{F}_E$. Such strengthening would require constructing expansions of $\mathcal{A}$ which preserve the structure of congruences of finite index of the algebra $\mathcal{A}$. This of course would show that any class R_E of regular languages which has a computable character can have an expansionary specification.

Comment 2. In light of the result mentioned in the comment above, in [1] Bergstra and Tucker pose the problem as to whether or not any Σ_1-algebra $\mathcal{A}$ can have an expansion $\mathcal{A}^\star$ so that $\mathcal{A}^\star$ is in fact the initial algebra of some finite set E of equations in the expanded language. In [10] and [12] this problem is solved negatively by using computability-theoretic arguments and constructions. Our specification problem for the class R_E is significantly weaker than that of Bergstra-Tucker. Therefore a counterexample to our specification problem is harder to provide. In the next section, however, we provide such a counterexample. The ideas used in the counterexample are similar but more elaborate than those provided in [10] and [12].

4.3 A Counterexample

We fix the following signature $\sigma =< f_1, f_2, c >$, where f_1, f_2 are unary function symbols and c is a constant. We now provide some notions from computability theory. An infinite subset of the set G of ground terms (of signature σ) is **immune** if it contains no infinite c.e. subsets. A c.e. set $X \subset G$ with immune complement $\bar{X}$ is called **simple**. Simple sets exist, see for example [15]. A subset X of the set G of all ground terms is a **weak subalgebra** if $f_1(x), f_2(x) \in X$ for all $x \in X$. Note that in case $c \in X$ for a weak subalgebra X then $X = G$.

Any weak subalgebra X of the algebra $\mathcal{F} = (G, f_1, f_2, c)$ of ground terms defines a congruence relation $\eta(X)$ as follows:

$$(t_1, t_2) \in \eta(X) \text{ iff } t_1, t_2 \in X \vee t_1 = t_2.$$

We denote the factor algebra defined by this equivalence relation by $\mathcal{A}_X$. A weak subalgebra X is **simple** if X is a simple set.

The next lemma shows that simple weak subalgebras exist. In the proof of the lemma we use the following notation. Let Y be a subset of the set G. Consider $Cl(Y)$ which consists of all terms which have subterms from Y. Thus, it is easy to see that

$$Cl(Y) = \{t(y) \mid y \in Y \text{ and } t \text{ is a term with one variable } x\}.$$

Clearly, $Cl(Y)$ is a weak subalgebra of $\mathcal{F}$.

Lemma 4.3 *There exists a simple weak subalgebra of $\mathcal{F}$.*

Proof. Let $W_0, W_1, \ldots$ be a standard enumeration of all c.e. subsets of G. We construct the weak subalgebra X by stages. At stage s we define a set X_s, then put $X = \bigcup_s X_s$. In order to construct the desired weak subalgebra we need to satisfy the following list of requirements:

$$R_i : \quad W_i \bigcap \bar{X} \neq \emptyset,$$

where $i \in \omega$, W_i is infinite and $W_i \not\subset X$. We say that the requirement R_i **attracts the attention at stage** s if

$$W_{i,s} \bigcap X_s = \emptyset \quad \text{and} \quad W_{i,s} \neq \emptyset.$$

Here is now the construction of X. At the initial stage, *Stage* 0, we set $X_0 = \emptyset$. At *Stage* s we proceed as follows. Assume that X_{s-1} has been constructed. Find the minimal R_i, $i \leq s$, that requires attention. Take the first term $t \in W_{i,s}$ such that $h(t) > i+1$, and set $X_s = Cl(X_{s-1} \bigcup \{t\})$. Go to the next stage. If no $i \leq s$ requires attention then go to the next stage. This ends the construction at this stage.

Let $X = \bigcup_s X_s$. Clearly, X is computably enumerable.

It is not hard to see that for each i there is a term $t \notin X$ of length $i+1$. Therefore the complement $\bar{X}$ of the set X is infinite. If X is not simple, then take the minimal i for which $W_i \subset \bar{X}$ and W_i is infinite. Consider the stage t, after which no r_j, $j < i$, requires attention. Then there must exist a stage $s > t$ at which r_i requires attention. Hence $W_{i,s} \bigcap X_s \neq \emptyset$, and therefore $W_i \bigcap X \neq \emptyset$ which is a contradiction. We conclude that the set X is simple. By the construction, the set X forms a weak subalgebra. The lemma is proved.

No we are ready to prove a theorem that provides a counterexample to our specification problem.

Theorem 4.3 *There exists a class R_E which has the following properties:*

1. *The initial algebra $\mathcal{F}_E$ for the class is a Σ_1-algebra.*
2. *The class R_E has no expansionary specification.*

Proof. Consider the absolutely free algebra $\mathcal{F}$ of the signature $\sigma = < f_1, f_2, c >$. Let X be the weak subalgebra X constructed in the lemma. Define the following c.e. congruence relation $\eta = \{(t,q) \mid t = q \text{ or } t, q \in X\}$. Take the algebra $\mathcal{A}$ obtained by factorizing $\mathcal{F}$ by η. Define $E = E(\mathcal{A})$, where $E(\mathcal{A})$ is the set of all equations true in $\mathcal{A}$. Now our goal is to show that R_E is the required class.

Note that the congruence relation η that defines $\mathcal{A}$ is a c.e. relation. Therefore $\mathcal{F}_E$ is a Σ_1-algebra. This proves the first part of the theorem.

To prove the second part we need some notions. Let f be a basic $n-$ary operation of an algebra $\mathcal{B}$. A **transition of** $\mathcal{B}$ is any of the mappings $f(a_1, \ldots, a_{n-1}, x)$, $\ldots$, $f(x, a_1, \ldots, a_{n-1})$, where $a_1, \ldots, a_{n-1} \in B$ are fixed. Let $Tr(\mathcal{B})$ be the algebra whose basic operations are all transitions of $\mathcal{B}$. Then any binary relation α is a congruence relation of $\mathcal{B}$ if and only if α is a congruence of the algebra $Tr(\mathcal{B})$ (see for example [7]).

Assume that there exists a refinement R_{E_1} of R_E which has a finite specification. Hence there exists a finite E' such that $R_E = R_{E'}$. Let $\mathcal{A}' = (A', f_1, \ldots, f_n)$ be the initial algebra $\mathcal{F}_{E'}$ defined by E'. Since E' is finite, the algebra $\mathcal{A}'$ is a Σ_1–algebra. It is not hard to see that the $< f_1, f_2, c >$–reduct $\mathcal{A}'_\sigma$ of $\mathcal{A}'$ is a homomorphic image of $\mathcal{A}$ because $\mathcal{A}'_\sigma$ satisfies all the equations from E. Let $t \to t'$ be the homomorphism from $\mathcal{F}$ into $\mathcal{A}'_\sigma$. Let X' be the image of the weak algebra X in $\mathcal{A}'$, and Y be the preimage of X' in the algebra $\mathcal{F}$. The set Y is a c.e. superset of X and therefore is simple. Thus, for any $t' \notin X'$ the set of all ground terms q equal to t' in $\mathcal{A}'$ is finite.

Our goal is to show that $\mathcal{A}'$ is residually finite. This would lead to a contradiction, as in this case by Corollary 3.3, the algebra $\mathcal{A}'$ would be computable.

In order to prove that $\mathcal{A}'$ is residually finite consider an effective list $F_0, F_1, \ldots$ of all transitions of the expanded algebra $\mathcal{A}'$. Consider **the transition algebra of** $Tr(\mathcal{A}')$. As noted above, it suffices to prove that $Tr(\mathcal{A}')$ is residually finite.

Let $t'_1, t'_2 \notin X'$ such that $t'_1 \neq t'_1$. We will show that there exists a finite set S' in the complement of X' such that $t'_1, t'_2 \in S'$ and the relation $eq(S') = \{(x', y') \mid x, y \in G \setminus S\} \bigcup \{(x', y') \mid x = y\}$ induces a congruence of the transition algebra $Tr(\mathcal{A}')$.

If such a set S' exists, then the mapping $h : t' \to \{s' \mid (t', s') \in eq(S')\}$ will be a homomorphism from $\mathcal{A}'$ onto a finite algebra in which $h(t'_1) \neq h(t'_2)$.

To prove that there exists a set S' with the above properties we need to make several notes. Fix a term $u \in Y$, a transition F_i, and a finite $S' \subset \bar{X}'$. Let S be the set of all ground terms t such that $t' \in S'$. Note that S is finite. If $F_i(u') \notin S'$ then $\{t \mid F_i(t') \in S'\} \subset \bar{Y}$. This set of ground terms is computable and hence, since $\bar{Y}$ is immune, is finite. If $F_i(u') \in S'$ then $F_i(q') = F_i(u')$ for all $q \in Y$, and again the set $\{t \mid F_i(t') \neq F_i(u')\}$ of ground terms is computable and hence is finite.

Note the following fact. Let S' be a subset of the complement of X'. The equivalence relation $eq(S') = \{(x', y') \mid x, y \in G \setminus S\} \bigcup \{(x', y') \mid x = y\}$ is a congruence for the transition F_i if and only if for all $t' \notin S'$ we have the following: *a*) $F_i(u') \in S'$ if and only if $F_i(t') = F_i(u')$, and *b*) $F_i(u') \notin S'$ if and only if $F_i(t') \notin S'$.

Now we give a stagewise construction of S'. At *Stage* 0 we let $S'_0 = \{t'_1, t'_2\}$. Clearly $S_0 \subset \bar{Y}$ and is finite. *Stage* $j + 1$ proceeds as follows.

Suppose that S'_j has been constructed and $S'_j \subset \bar{X}'$. Consider the transitions $F_0, \ldots, F_{j+1}$. For each $i \leq j+1$, consider $F_i(u')$. If $F_i(u') \notin S'_j$, then let $S'_{j+1,i} = S'_j \bigcup \{t' \mid F_i(t') \in S'_j\}$. Otherwise, let $S'_{j+1,i} = S'_j \bigcup \{t' \mid F_i(t') \neq F_i(u')\}$. Define $S'_{j+1} = S'_{j+1,0} \bigcup \ldots \bigcup S'_{j+1,j+1}$. Clearly $S_{j+1} \subset \bar{Y}$, and at this stage we can effectively contract an algorithm to decide S_{j+1}.

By the remarks given before the construction, the set $S = \bigcup_j S_j$ is a finite subset of $\bar{Y}$. There exists a stage j_0 such that $S = S_{j_0}$. The terms t_1 and t_2 belong to S. We have to show that $eq(S')$ induces a congruence relation for every transition F_i. It suffices to prove that if t' does not belong to S', then $(F_i(u'), F_i(t')) \in eq(S')$. Consider any stage $j \geq j_0$. Suppose that $F_i(u') \notin S'_j$. Then $F_i(u') \notin S'_j$, otherwise $u' \in S'_j$ and hence $S_{j_0} \neq S_j$. Similarly, if $F_j(u') \in S'_j$, then $F_j(t') = F_j(u')$, otherwise $t' \in S'_j$ and hence $S'_{j_0} \neq S'_j$. Thus, the homomorphism h defined by $h : t \rightarrow \{s' \mid (t', s') \in eq(S')\}$ maps $\mathcal{A}'$ onto a finite algebra in which $h(t'_1) \neq h(t'_2)$. Thus, $\mathcal{A}'$ is residually finite. The theorem is proved.

5 Acknowledgment

The author would like to thank the referees for their careful reading of the preliminary version of this paper, excellent suggestions for improvements, and pointing out a number of corrections in the text.

References

[1] J.A. Bergstra, J.V.Tucker, Algebraic Specifications of Computable and Semicomputable Data Types, Theoretical Comp. Science, 50, 1987.

[2] J. R. Büchi. *Finite Automata, Their Algebras and Grammars: Towards a Theory of Formal Expressions*, D. Siefkes (editor), Springer-Verlag, 1989.

[3] M. J. Dinneen and B. Khoussainov. *Automata with Equational Constraints*, Proceedings of the Australasian Workshop on Constraint Programming and Applications, CPA-99, p.42-51, 12th Australian Joint Conference on AI held at UNSW, Sydney, Australia, December.

[4] Computability Theory and Its Applications. Current Trends and Open Problems. Contemporary Mathematics Series, **257**. Eds: P. A. Cholak, S. Lempp, M. Lerman, R.A.Shore. 2000.

[5] Handbook of Formal Languages, Volume 1, G. Rosenberg and A. Salomaa (eds.), Springer-Verlag, 1997.

[6] F. Gécseg and M. Steinby. *Tree Automata*, Akadémiai Kiadó, Budapest, 1984.

[7] G. Grätzer. *Universal Algebra*, Springer-Verlag, New York and Heidelberg, 2nd edn., 1979.

[8] T. Evans. Some connections between Residual Finiteness, Finite Embaddability, and the Word Problem. *J. London Math. Soc.*, **1**, 399-403, 1969.

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

[10] N.K. Kassimov. On Finitely Approximable and R.E. Representable Algebras, Algebra and Logic, 26, No 6,1986.

[11] O.G. Kharlampovich and M.V. Sapir. Algorithmic Problems in Varieties. *International J. Algebra Comput.*, **5**, 379-6-2, 1995.

[12] B. Khoussainov. Randomness, Computability, and Algebraic Specifications, *Annals of Pure and Applied Logic* 91, no1, p.1-15, 1998.

[13] A.I. Malcev. *Constructive Algebras*, Uspekhi Matem. Nauk, 16, No 3, 1961, 3-60.

[14] *Handbook of Recursive Mathematics*, Volume 1. Studies in Logic and Foundations of Mathematics, edited by Yu. Ershov, S. Goncharov, A. Nerode, J. Remmel. Associate Editor V. Marek, Elsevier, 1998.

[15] R. Soare. *Recursively Enumerable Sets and Degrees.* The Study of Computable Functions and Computably Generated Sets. Perspect. in Math. Logic, 1987.

[16] M. Wirsing. *Algebraic Specifications.* In Handbook of Theoretical Computer Science, Volume B. Editor Jan Van Leeuwen. MIT Press. p.675-788. 1990.