Abstract of the Ph.D. Thesis

The Complexity of Star-Freeness and the Free Conway Theories

by

László Bernátsky

József Attila University
Department of Computer Science
Szeged, Árpád tér 2.
H-6720 Hungary

1999

1 Preliminaries

The thesis has two main topics: the problem of deciding if a regular language is star-free, and the description of the free Conway theories. I would like to note here that I first considered the second problem, and the solution to that problem gave the motivation to consider the aperiodicity problem of finite automata. Since that was already solved by Cho and Huynh [11] Zoltán Ésik suggested to study the complexity of deciding if a regular expression denotes a star-free language.

Star-free languages form an important subclass of regular languages. By definition, they are the languages that can be obtained from the singleton languages by a finite number of applications of the operations of union, complement and product. By Schützenberger's [26] famous theorem, a regular language $L \subseteq \Sigma^*$ is star-free if and only if its syntactic monoid is aperiodic, i.e., if there exists an integer $k \geq 0$ such that for all words $u, v, w \in \Sigma^*$,

$$uv^k w \in L \iff uv^{k+1} w \in L.$$

This is the same as to say that L is recognized by an aperiodic DFA. There also exists a logical characterization by which a language is star-free if and only if it can be defined by a first-order formula of a suitable formal language, see [29]. Thus, it is interesting to know how hard it is to decide if a given regular language is star-free.

Naturally, when considering the problem of deciding whether a language has some property, we implicitly assume that the language is given by some finite representation. The thesis studies the complexity of the star-freeness problem when the regular language is represented by a (nondeterministic) finite automaton, and, more importantly, when it is represented by a regular expression. Some restricted versions of these decision problems are also considered, and it is proved that all of them are **PSPACE**-complete. As a byproduct, we obtain some new complexity results on some restricted versions of the automata intersection problem, first considered by Kozen [21].

The second main topic of my thesis is the description of the free algebras in the variety of Conway theories—a kind of many-sorted algebra axiomatized by a finite set of equation schemes, or *meta-equations*. The description uses flowchart schemes.

The algebraic study of flowchart schemes and flowchart algorithms was initiated in [15] and further developed in [6, 27, 10], to mention only a few references. Schemes may be defined as locally ordered, vertex labeled, finite digraphs with distinguished begin and exit nodes, each numbered by a nonnegative integer, so that each scheme has source n and target p for sonie nonnegative integers n, p. (We use N to denote the set of nonnegative integers.) The other nodes are consistently labeled by letters in a ranked alphabet, or signature. Schemes over a *signature* Σ, or Σ-*schemes for short*, are equipped with several constants and the operations of sequential composition, pairing or separated sum—which may be viewed as some sort of parallel composition—and a looping operation called iteration. (The paper [10] uses feedback instead of iteration.) In [6], Σ-schemes have been characterized as the free algebra generated by the signature Σ in a variety of $\mathrm{N} \times \mathrm{N}$-sorted algebras axiomatized by a finite number of meta-equations. See also [27, 10] for refinements of this result.

Σ-schemes with source n and target p may also be viewed as morphisms $n \to p$ in a small category whose objects are the nonnegative integers. Unless Σ is trivial, coproducts do not exist in this category, so that Σ-schemes do not form an algebraic theory in the sense of Lawvere [23]. Nevertheless, schemes are commonly interpreted in such theories which are *enriched by a fixed-point operation modeling iteration. For example*, the theories $\mathbf{Seq}_A$ of sequacious functions [13] on a set A are used to model the stepwise behavior of flowchart algorithms, while the theories of partial functions $\mathbf{Pfn}_A$ serve as semantic models for input-output behavior. Another common class of interpretations of schemes is as continuous functions over cpo's. In this approach, a scheme is a graphical representation of a recursive system of equations. When A is a cpo with a bottom element, each letter in Σ may be interpreted as a continuous function on A. Then the semantics of a Σ-scheme is a morphism in the theory $\mathbf{Th}_A$ of continuous functions over A, obtained as the least solution of the system of equations corresponding to the scheme.

The theories $\mathbf{Seq}_A$, $\mathbf{Pfn}_A$, and $\mathbf{Th}_A$ are all examples of *iteration theories*, which were originally defined in [4, 5] and [16]. The book [8] and the paper [9] give a summary of the results on iteration theories (and the properties of the fixed point operation in general).

It is known for example that the variety of iteration theories is generated by the theories $\mathbf{Seq}_A$, where A is a set, or by the theories $\mathbf{Th}_A$, where A is a cpo with a bottom element. (The theories of the form $\mathbf{Pfn}_A$ generate the subvariety consisting of the iteration theories with a unique morphism $1 \to 0$.) Thus, two schemes are equivalent under all interpretations in iteration theories (or *strongly equivalent*, for short) if and only if they are equivalent under all interpretations in the theories $\mathbf{Seq}_A$, or in the theories $\mathbf{Th}_A$. For this reason, iteration theories may be considered as the "standard" interpretations of flowchart schemes.

It is also well known that the equational theory of iteration theories (that is, the problem of deciding whether an equation holds in all iteration theories) is solvable in polynomial time. It is also decidable in polynomial time whether two schemes are strongly equivalent.

The thesis contains similar results about "nonstandard" interpretations of flowchart schemes, first published in [3]. By a nonstandard interpretation we mean a theory enriched with an iteration operation satisfying all equations true of flowcharts. One of the main results states that these theories are exactly the *Conway theories*.

Aside from serving as interpretation domains for flowchart schemes, our interest in (the free) Conway theories stems from several mathematical facts. First, the complete description of a variety of algebras should include (at least) an equational axiomatization, and also a concrete description of the free algebras. For example, the papers [2, 18] give an axiomatization and a concrete description of the free algebras for the variety generated by all algebras of binary relations with operations of union, composition, conversion and reflexive-transitive closure, and neutral elements 0 (the empty relation) and 1 (the identity relation). Second, the equational theory of iteration theories is axiomatized by the Conway theory axioms together with a complicated equation scheme, the commutative identity [16], or the group identities [17], or the Scott induction principle formulated to involve only equations [19]. (The second of the latter results may be seen as a generalization of Krob's result [22] on the axiomatization of the regular identities.) Comparing the structure of the free Conway theory with that of the free iteration theory, we obtain a clear picture of that part of the equational theory of iteration theories which is captured by the commutative identity, or the group identities. Also, our work explains the role of the commutative identity: it separates nonstandard models from the standard ones by equations. And finally, Conway theories are interesting in themselves for the following reasons.

- In a matrix theory [14, 8] equipped with a unary operation $a \mapsto a^*$, the Conway axioms are the two well-known sum and product identities

  $$(a+b)^* = (a^*b)^*a^*$$
  $$(ab)^* = a(ba)^*b + 1.$$

  Conway's book [12] contains many interesting identities which are consequences of just the Conway axioms. See also [22].

- A general Kleene-type theorem is a logical consequence of just the Conway axioms, see [8].

- It was proved in [7] that the soundness, and relative completeness of the Floyd-Hoare calculus in expressive models, is a consequence of the Conway theory axioms. Thus, even under nonstandard interpretations, one can reason about the correctness of flowchart programs using the Floyd-Hoare rules.

2 Results on the complexity of star-freeness

The results on the complexity of star-freeness contained in the thesis were published in [1]. The only exception is Theorem 2.3.2, which I proved later and have not published yet. I also have to note that, due to a minor modification of Construction 2.2.2, Theorem 2.3.1 is slightly stronger than the corresponding theorem published in [1].

We say that a finite automaton is a reset automaton if it has a unique initial state, and each input symbol of the automaton induces either the identity function or a partial constant function (that is, a partial function having a range of cardinality at most one) on the states. A 1-reset automaton is a reset automaton with a unique final state in which the inverse of each relation induced by an input symbol is either the identity function or a partial constant function. In other words, a 1-reset automaton has a unique initial state and a unique final state, and each input symbol induces either the identity function or a singleton relation or the empty relation on its states.

A deterministic finite automaton (DFA) is called minimal if it has a minimal number of states among all DFAs recognizing the same language. A DFA is called complete if it has a unique initial state and each input symbol induces a total function on its states.

The thesis provides an analysis of the computational complexity of the following decision problems.

1. The automata intersection problem (**AIP**):

   INSTANCE: A sequence $\mathcal{A}_1, \dots, \mathcal{A}_n$ ($n \geq 2$) of nondeterministic finite automata with a common set of input symbols.

   QUESTION: Does $\bigcap_{i\in[n]} L(\mathcal{A}_i) \neq \emptyset$ hold?

2. The intersection problem of minimal 1-reset automata ($\mathbf{AIP}_R$):

   INSTANCE: A sequence $\mathcal{A}_1, \dots, \mathcal{A}_n$ ($n \geq 2$) of minimal 1-reset automata with a common set of input symbols.

   QUESTION: Does $\bigcap_{i\in[n]} L(\mathcal{A}_i) \neq \emptyset$ hold?

3. The intersection problem of complete reset automata ($\mathbf{AIP}_C$):

   INSTANCE: A sequence $\mathcal{A}_1, \dots, \mathcal{A}_n$ ($n \geq 2$) of complete reset automata with a common set of input symbols.

   QUESTION: Does $\bigcap_{i\in[n]} L(\mathcal{A}_i) \neq \emptyset$ hold?

4. Automaton star-freeness (**ASF**):

   INSTANCE: A nondeterministic finite automaton $\mathcal{A}$.

   QUESTION: Does $\mathcal{A}$ recognize a star-free language?

5. A restricted version of automaton star-freeness ($\mathbf{ASF}_R$):

INSTANCE: A minimal DFA $\mathcal{A}$ with input symbols {0,1}.

QUESTION: Does $\mathcal{A}$ recognize a star-free language?

6. Regular expression star-freeness (**RSF**):

   INSTANCE: A regular expression E.

   QUESTION: Does E denote a star-free language?

7. A restricted version of regular expression star-freeness ($\mathbf{RSF}_R$):

   INSTANCE: A regular expression E of star-height 2 over the 2-element set $\{0, 1\}$.

   QUESTION: Does E denote a star-free language?

Assuming some efficient encoding of automata and regular expressions (see [24, 20]) with words over a fixed finite set of symbols, all these problems can be considered as languages.

The first result on the complexity of **ASF** is due to Jacques Stern [28], who proved that the problem of deciding whether a deterministic finite automaton recognizes a star-free language is **coNP**-hard and belongs to **PSPACE**. A few years later Shang Cho and Dung T. Huynh [11] proved that the problem $\mathbf{ASF}_R$ is **PSPACE**-complete.

- We prove that all the other problems above are also **PSPACE**-complete, except for $\mathbf{AIP}_C$, for which we present a polynomial time algorithm in Theorem 2.3.2.

The remaining results are separated into three theorems.

- In Theorem 2.3.1 we show that

  $$\mathbf{PSPACE} \leq_{log} \mathbf{AIP}_R \leq_{log} \mathbf{AIP} \in \mathbf{PSPACE},$$

  so that both $\mathbf{AIP}_R$ and **AIP** are **PSPACE**-complete. Since **PSPACE** is closed under taking complements, it follows that the complementary problems $\overline{\mathbf{AIP}_R}$ and $\overline{\mathbf{AIP}}$ are also **PSPACE**-complete. The initial reduction $\mathbf{PSPACE} \leq_{log} \mathbf{AIP}_R$ (see Construction 2.2.1) is one of the most important results. It is a sharpening of the one given by Kozen [21] in the sense that this construction yields a collection of much simpler finite automata, namely, 1-reset automata.

- In Theorem 2.3.3 we show that

  $$\overline{\mathbf{AIP}_R} \leq_{log} \mathbf{ASF}_R \leq_{log} \mathbf{ASF} \in \mathbf{PSPACE},$$

  so that both $\mathbf{ASF}_R$ and **ASF** are **PSPACE**-comlete. The reduction $\overline{\mathbf{AIP}_R} \leq_{log} \mathbf{ASF}_R$ is closely related to the one of Cho and Huynh, but it is simpler due to the fact that the input of the problem $\mathbf{ASF}_R$ is a sequence of 1-reset automata, which are a very special kind of aperiodic automata. (Cho and Huynh were using Kozen's aforementioned construction as a starting point in proving that the problem $\mathbf{ASF}_R$ is **PSPACE**-hard. Since the result of that construction is a sequence of automata which are not aperiodic in general, there is an additional step in the proof of Cho and Huynh in order to get a sequence of aperiodic automata. No such modification is needed in my argument.) It is a new result that the problem **ASF** is solvable in polynomial space.

- Then in Theorem 2.3.4 we prove that

$$\overline{\mathbf{AIP}_R} \leq_{log} \mathbf{RSF}_R \leq_{log} \mathbf{RSF} \leq_{log} \mathbf{ASF}.$$

  Here the essential part is the reduction $\overline{\mathbf{AIP}_R} \leq_{log} \mathbf{RSF}_R$. It is again heavily based on the fact that the problem $\mathbf{AIP}_R$ deals with a collection of 1-reset automata.

It remains an interesting open problem how hard it is to decide if a regular expression of star-height 1 denotes a star-free language.

3 Results on the free Conway theories

The results on the free Conway theories were published in [3].

As we noted before, the standard interpretations of flowcharts are iteration theories, for example, the theory $\mathbf{Seq}_A$ of sequacious functions over a set A [13], the theory $\mathbf{Th}_A$ of continuous functions over a cpo A, or the theory of partial functions $\mathbf{Pfn}_A$.

A nonstandard interpretation of flowcharts is a theory enriched with an iteration operation satisfying all equations true of flowcharts.

- The first result, Theorem 3.3.2 states that the nonstandard interpretations are exactly the *Conway theories* defined in [8], axiomatized by a small set of meta-equations including the well-known composition identity, which implies Elgot's fixed point identity [13]. Thus the least congruence on Σ-schemes whose quotient is a theory gives the free Conway theory.
- The second main result, Theorem 4.3.2, provides an explicit description of the free Conway theories. The description uses aperiodic simulations of flowchart schemes, a concept borrowed from automata theory (see [25]). It follows that the equations that hold in Conway theories are exactly the valid "group-free" equations of iteration theories.

The proof of Theorem 4.3.2 is based on Theorem 3.3.2, as well as two other auxiliary results, Corollary 4.1.30 and Lemma 4.3.1, which are interesting in themselves.

In the thesis, we start by introducing the notion of aperiodic simulations of flowchart schemes. Briefly, a simulation from a flowchart scheme S to a flowchart scheme S' is a binary relation ρ from the states of S to the states of S' preserving the transitions and the labeling of the states. A simulation from a scheme to itself is called a congruence if it is an equivalence relation. A congruence ρ on a scheme S is called aperiodic if none of the possible transition sequences of S induces a nontrivial permutation of any subset of an equivalence class of ρ. A simulation is called a homomorphism if it is function. It is easy to see that the kernel of a homomorphism ϕ from a scheme S to a scheme S' is a congruence on S. We say that a homomorphism is aperiodic if its kernel is an aperiodic congruence. We also introduce some special kinds of aperiodic congruences in Definition 4.1.6, and prove several properties of these congruences in a sequence of lemmas. For example, an aperiodic homomorphism $\phi : S \to S'$ is called simple if for any two nonsingleton equivalence classes C, D of its kernel, any relation from C to D induced by a transition sequence of S is either a constant function or a bijective function from C to D.

Then we consider three equivalence relations on the class of flowchart schemes.

1. The least congruence on the algebra of schemes such that the quotient is a preiteration theory. In other words, this is the smallest equivalence relation which identifies two schemes S and S' with n input nodes and p output nodes if either $n = 0$, or S consists of two disjoint copies of the internal nodes of S' such that for some $0 \leq k \leq n$ the first k input nodes of S are connected to the first copies of the corresponding internal nodes of S', the last $n - k$ input nodes are connected to the second copies of the corresponding internal nodes of S', and both copies of the internal nodes of S' are connected to the same output nodes in S as in S'.

2. The smallest equivalence identifying any two schemes S and S' such that there is a simple aperiodic homomorphisms from S to S'.

3. The smallest equivalence identifying any two schemes S and S' such that there is an aperiodic homomorphism from S to S'.

- After proving a few technical lemmas—the most important being Lemma 4.1.29—we establish in Corollary 4.1.30 that the second of the above equivalences is the same as the third one.

- In the course of this proof we obtain a concrete description of the third equivalence, stated as Theorem 4.1.27.

- Then we prove in Lemma 4.3.1 that the second of the above equivalences is the same as the first one. Consequently, we obtain a concrete description of the free Conway theories as the quotient of the algebra of schemes under the third equivalence relation.

- Finally, we use the explicit description stated in Theorem 4.1.27 to prove in Theorem 4.3.3 that the following problems are **PSPACE**-complete for an arbitrary signature Σ containing at least one symbol of rank at least two.

1. Aperiodic Σ-schemes ($\mathbf{ASch}_\Sigma$):

   INSTANCE: A (strongly accessible) Σ-scheme S.

   QUESTION: Is $S \times S$ an aperiodic congruence on S?

2. Aperiodic congruences of Σ-schemes ($\mathbf{ACong}_\Sigma$):

   INSTANCE: A (strongly accessible) Σ-scheme S and a Σ-sorted relation $\rho \subseteq S \times S$.

   QUESTION: Is ρ an aperiodic congruence on S?

3. The Conway-equivalence problem of Σ-schemes ($\mathbf{SchEq}_\Sigma$):

   INSTANCE: A pair (S, S') of Σ-schemes.

   QUESTION: Does $S \equiv S'$ hold?

4. The equational theory of Conway theories with variables in Σ ($\mathbf{Eq}_\Sigma(\mathit{Conway})$):

   INSTANCE: A pair (t, t') of terms over the signature of Conway theories consisting of variables in Σ.

   QUESTION: Does the equation $t = t'$ hold in all Conway theories?

Theorems 3.3.2 and 4.3.2 answer open problems raised in [6] and [8].

References

[1] L. Bernátsky. Regular expression star-freeness is PSPACE-complete. *Acta Cybernetica*, 13:1–21, 1997.

[2] L. Bernátsky, S. L. Bloom, Z. Ésik, and G. Stefanescu. Equational theories of relations and regular sets, Extended abstract. In *Proceedings of the 2nd Colloquium on Words, Combinatorics and Semigroups*, 1992.

[3] L. Bernátsky and Z. Ésik. Semantics of flowchart programs and the free Conway theories. *Theoretical Informatics and Applications*, 32(1-2-3):35–78, 1998.

[4] S. L. Bloom, C. C. Elgot, and J. B. Wright. Solutions of the iteration equation and extensions of the scalar iteration operation. *SIAM Journal of Computing*, 9:24–65, 1980.

[5] S. L. Bloom, C. C. Elgot, and J. B. Wright. Vector iteration in pointed iterative theories. *SIAM Journal of Computing*, 9:525–540, 1980.

[6] S. L. Bloom and Z. Ésik. Axiomatizing schemes and their behaviours. *Journal of Computing and System Sciences*, 31:375–393, 1985.

[7] S. L. Bloom and Z. Ésik. Floyd–Hoare logic in iteration theories. *JACM*, 38:887–934, 1991.

[8] S. L. Bloom and Z. Ésik. *Iteration theories: the equational logic of iterative processes.* EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1993.

[9] S. L. Bloom and Z. Ésik. The equational logic of fixed points. *Theoretical Computer Science*, 179(1–2):1–60, 1997.

[10] V. E. Cazanescu and G. Stefanescu. Towards a new algebraic foundation of flowchart scheme theory. *Fundamenta Informaticae*, 13:171–210, 1990.

[11] S. Cho and D. T. Huynh. Finite-automaton aperiodicity is PSPACE-complete. *Theoretical Computer Science*, 88:99–116, 1991.

[12] J. C. Conway. *Regular algebra and finite machines.* Chapman and Hall, 1971.

[13] C. C. Elgot. Monadic computation and iterative algebraic theories. In J. Shepherdson, editor, *Logic Colloquium 1973*, volume 80 of *Studies in Logic*, Amsterdam, 1975. North Holland.

[14] C. C. Elgot. Matricial theories. *Journal of Algebra*, 42:391–421, 1976.

[15] C. C. Elgot. Structured programming with and without goto statements. *IEEE Transactions on Software Engineering*, 2(1):41–54, Mar. 1976.

[16] Z. Ésik. Identities in iterative and rational algebraic theories. *Computational Linguistics and Computer Languages*, 14:183–207, 1980.

[17] Z. Ésik. Group axioms for iteration. *Information and Computation*, 148:131–180, 1999.

[18] Z. Ésik and L. Bernátsky. Equational properties of Kleene algebras of relations with conversion. *Theoretical Computer Science*, 137:237–251, 1995.

[19] Z. Ésik and L. Bernátsky. Scott induction and equational proofs. In *MFPS XI, 1995*, volume 1 of *Electronic Notes in Theoretical Computer Science (ENTCS)*, 1995. http://www.elsevier.nl/locate/entcs/volume1/esik.ps, 28 pages.

[20] M. R. Garey and D. S. Johnson. *Computers and intractability, A guide to the theory of NP-completeness.* W. H. Freeman and Company, New York, 1979.

[21] D. Kozen. Lower bounds for natural proof systems. In *Proc. 18th Ann. Symp. on Foundations of Computer Science*, pages 254–266, Long Beach, CA, 1977. IEEE Computer Society.

[22] D. Krob. Complete systems of B-rational identities. *Theoretical Computer Science*, 89:207-343, 1991.

[23] F. W. Lawvere. Functorial semantics of algebraic theories. *Proceedings of the National Academy of Sciences USA*, 50:869–873, 1963.

[24] C. H. Papadimitriou. *Computational complexity.* Addison-Wesley, New York, 1994.

[25] J.-E. Pin. *Varieties of formal languages.* North Oxford Academic, 1986.

[26] M. P. Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8:190–194, 1965.

[27] G. Stefanescu. On flowchart theories: part I. The deterministic case. *Journal of Computing and System Sciences*, 35:163–191, 1987.

[28] J. Stern. Complexity of some problems from the theory of automata. *Information and Control*, 66:163–176, 1985.

[29] H. Straubing. *Finite automata, formal languages and circuit complexity.* Birkhäuser, 1994.