Automated Analysis of Regular Algebra

Simon Foster and Georg Struth

Department of Computer Science, The University of Sheffield
{s.foster,g.struth}@dcs.shef.ac.uk

Abstract. Regular algebras axiomatise the equational theory of regular expressions. We use Isabelle/HOL's automated theorem provers and counterexample generators to study the regular algebras of Boffa, Conway, Kozen and Salomaa, formalise their soundness and completeness (relative to a deep result by Krob) and engineer their hierarchy. Proofs range from fully automatic axiomatic and inductive calculations to integrated higher-order reasoning with numbers, sets and monoid submorphisms. In combination with Isabelle's simplifiers and structuring mechanisms, automated deduction provides powerful support to the working mathematician beyond first-order reasoning.

1 Introduction

Regular languages, regular expressions and finite automata belong to the foundations of computing. Regular algebras are the mathematical structures that underly these formalisms. Originally proposed for axiomatising the equational theory of regular expressions, they have since found wide applications in various fields of computing.

Work on regular algebras has spanned decades. Salomaa gave two axiom systems, proved completeness of the first and conjectured it of the second [12]. Conway, in his influential monograph, conjectured completeness of several alternative axiomatisations [6]. Krob gave a long and intricate completeness proof of Conway's so-called classical axioms extended by a system of monoid identities [10]. Boffa proved completeness of two particularly simple algebras relative to Krob's result [3,4]. Relative to Boffa's algebras, Krob, in turn, verified some of Conway's remaining conjectures. Kozen proved completeness of a simplified algebra of Conway [8], which under the name *Kleene algebra* has been widely studied and applied since. Boffa, in turn, showed completeness of a simplified version of Kleene algebra.

Within the programme of enhancing mathematics by theorem provers, regular algebras yield an interesting test case: they include pure first-order as well as higher-order structures axiomatised by inductive families of identities and with elements generated by finite monoids via submorphisms. Proofs include equational calculations and integrated higher-order reasoning about algebra, numbers, sets (of lists), infinite suprema and functions. Our main motivation is the following question: *How far can first-order automated theorem provers support the working mathematician in such a heterogeneous environment?*

B. Gramlich, D. Miller, and U. Sattler (Eds.): IJCAR 2012, LNAI 7364, pp. 271–285, 2012.

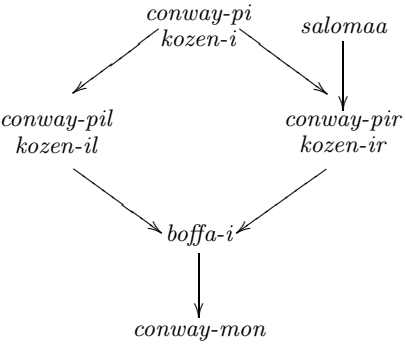

Fig. 1. Fine structure of regular algebra

Automated theorem proving alone, is of course too limited for our study. It is only possible due to Isabelle/HOL's [11] recent integration of first-order proof and counterexample search technology into a higher-order interactive theorem proving environment (cf. [2] for an overview). In a nutshell, Isabelle's Sledgehammer tool delegates proof goals to external automated theorem provers (ATPs) and satisfiability modulo theories (SMT) solvers. A relevance filter gathers hypotheses for the external tools. Their proof outputs are reconstructed within Isabelle to increase trustworthiness. ATP in Isabelle is complemented by the Nitpick and Quickcheck counterexample search tools. This integration supports a very natural new style of computer enhanced mathematics. Traditionally, work with Isabelle was driven by its simplifiers and direct applications of theorems from its libraries. Now, paper and pencil proofs can be typed directly into Isabelle's proof scripting language and verified step by step by an ATP system using the hypotheses it gathers. With this approach, an Isabelle repository for Kleene and relation algebras with more than 2000 facts has already been implemented[1]. But the fine structure of regular algebras with their higher-order features has not yet been considered. Our main contributions are as follows:

We implement the algebras of Boffa, Conway (without monoid identities), Kozen and Salomaa as abstract type classes in Isabelle and develop a library of regular identities and auxiliary concepts for Boffa's algebras.

We use Isabelle's locale mechanism in combination with Nitpick to capture meta-theorems that relate these algebras. We reconstruct known completeness results for regular algebras and add some new ones: equipollence (mutual deducibility) of Boffa's algebras, and of some of Conway's algebras and Kleene algebras, a simple completeness proof for Salomaa's first algebra, a gap in Boffa's completeness proof for his second one, and proofs that various subclasses are proper. The main relationships are shown in Figure 1. Nodes represent equipollent algebras; arrows the implication preorder. All completeness proofs are relative to Krob's result; they are based on implications between axiom systems.

[1] `http://www.dcs.shef.ac.uk/~georg/isa`

We establish soundness of regular algebras relative to regular languages. Soundness of Salomaa's algebra and Kleene algebra is automatically propagated down the hierarchy by Isabelle's sublocale mechanism.

We reconstruct Boffa's completeness result relative to Conway's classical axioms with monoid identities. This requires an alternative implementation of Boffa's algebras with explicit carrier sets and additional theory infrastructure. In this case, the sublocale mechanism propagates completeness up the hierarchy.

Most subclass and equipollence proofs are fully automatic. This demonstrates the impressive power of ATP in algebraic reasoning. Automating more complex results requires specific elimination rules for higher-order structure and an interplay with Isabelle's simplifier. As soon as supporting libraries were developed, all proofs could be implemented at least at textbook-level granularity in a natural mathematical style. Some formalisation tasks, in particular the construction of infinite counterexamples, are deliberately left open to demonstrate not only the potential, but also the limitations of our lightweight ATP-based approach.

This paper can only highlight some main features of our work. The complete Isabelle implementation can be accessed through our repository. We must also assume familiarity with the basics of Isabelle. We refer to the excellent online documentation, in particular the locale tutorial [1] and the references given therein, for further information. The paper itself has been processed by Isabelle's document preparation system, including the verification of its technical results. The following numbers underpin the success of ATP in analysing regular algebra: our implementation contains 303 proof goals. 242 were fully automatic (apart perhaps from calling an induction or case analysis tactic); 35 were fully automatic after invoking a simplifier; 26 required moderate user interaction.

2 Dioids, Powers and Finite Sums

All regular algebras can be based on dioids or idempotent semirings. Implementations of these structures and a library of facts can be found in the repository.

Formally, a *semiring* is a structure $(S, +, \cdot, 0, 1)$ where $(S, +, 0)$ is a commutative monoid, $(S, \cdot, 1)$ is a monoid, and the distributivity laws $x \cdot (y + z) = x \cdot y + x \cdot z$ and $(x + y) \cdot z = x \cdot z + y \cdot z$, and annihilation laws $x \cdot 0 = 0$ and $0 \cdot x = 0$ hold. A semiring is *idempotent*—a *dioid*—if $x + x = x$. In this case the reduct $(S, +)$ forms a semilattice, and can be endowed with the usual semilattice order $x \leq y \leftrightarrow x + y = y$. The least element of this order is 0 and the operations of addition and multiplication are isotone. An important concept in semiring theory is duality with respect to opposition. It is based on the opposite multiplication $x \circ y = y \cdot x$. We have implemented this duality in Isabelle and shown that $(S, +, \circ, 0, 1)$ is a dioid whenever $(S, +, \cdot, 0, 1)$ is. Duals of theorems in dioids are available for free in Isabelle. This also yields automatic completeness proofs for the duals of all structures in this paper (e.g. the righthanded algebras in Figure 1).

For most of the development in this paper, implementing algebras by axiomatic type classes is sufficient. Consequently, their carrier sets are left implicit. This is common mathematical practice, beneficial to automation, but insufficient

for more advanced mathematics (cf. Section 9). Some axiomatisations of regular algebras require powers and finite sums. Powers can be defined recursively.

primrec *power* :: $'a \Rightarrow nat \Rightarrow 'a$ (-$^{\wedge}$ [*101,50*] *100*)
where $x^{0} = 1$
| $x^{Suc\ n} = x \cdot x^{n}$

We have developed a basic library for powers. Typical facts are $x^n \cdot x = x \cdot x^n$ or $y \cdot x \leq y \rightarrow y \cdot x^n \leq y$. Apart from induction, proofs are mostly automatic. The following example illustrates the style of reasoning.

Lemma *power-add*: $x^m \cdot x^n = x^{m+n}$
Proof (*induct m*)
case *0* **show** *?case* **by** (*metis add-0-left mult-onel power.simps*(*1*))
case (*Suc m*) **show** *?case* **by** (*smt Suc add-Suc mult-assoc power.simps*(*2*))
qed

Isabelle's induction tactic is called to generate proof obligations for the base case and the induction step. Sledgehammer is then called on both cases. The first case is discharged by Metis, an internally verified ATP system. The second one uses SMT proof reconstruction. The proof uses the clauses in the definition of *power*, induction hypothesis *Suc* and facts about dioids and numbers. All have been gathered by the relevance filter.

Next we define a function that sums up powers: $x_m^n = \sum_{i=m}^{n+m} x^i$. Avoiding Isabelle's library function *setsum* yields better control over proof automation, but ultimately, an integration with existing Isabelle libraries is desirable (cf. [7]).

primrec *powsum* :: $'a \Rightarrow nat \Rightarrow nat \Rightarrow 'a$ (-$_{-}^{\wedge}$ [*101,50,50*] *100*)
where ${x_n}^{0} = x^{n}$
| ${x_n}^{Suc\ m} = {x_n}^{m} + x^{n+Suc\ m}$

Again we have proved a number of basic facts by ATP, often by induction, and sometimes calling Isabelle's simplifier before Sledgehammer.

3 Conway's Classical Axioms

Regular algebras are dioids expanded by the regular operation *. We implement Conway's classical axioms (p.25 in his monograph) using Isabelle's axiomatic type classes; hence again without explicit carrier sets.

Class *regalg-base* = *dioid-one-zero* + *star-op* + *plus-ord* +
assumes *C11*: $(x+y)^* = (x^* \cdot y)^* \cdot x^*$
and *C12*: $(x \cdot y)^* = 1 + x \cdot (y \cdot x)^* \cdot y$

Class *conway* = *regalg-base* +
assumes *C13*: $(x^*)^* = x^*$

Class *conway-classical* = *conway* +
assumes *C14*: $x^* = (x^{n+1})^* \cdot {x_0}^{n}$

The class *regalg-base* is reused for Boffa's first axiomatisation. In class *conway*, axiom schema $C14$—also called *powerstar* axiom—has been removed from the classical axioms, since it is not needed for most of our results.

Conway himself uses semirings instead of dioids. He shows that $x + x = x$ can be derived from that basis; hence both variants are equipollent. We use dioids for the sake of uniformity across the paper. Conway has shown that the classical axioms are incomplete with respect to (the equational theory of) regular languages (p. 118). He has also analysed the role of axiom $C13$ (p. 104). We could easily automate his analysis with Nitpick: a 3-element counterexample shows irredundancy of $C13$ in the semiring setting; in its absence, $x+x = x$ (3-element counterexample) and $x^* \cdot x^* = x^*$ (5-element counterexample) could be refuted. In the dioid setting, however, we could neither prove nor refute $x^* \cdot x^* = x^*$ automatically in the absence of $C13$ within Isabelle's default time limits. In the presence of powerstar, Nitpick uniformly failed. In fact, Conway constructs an infinite model of a semiring in which the classical axioms except $C13$ hold and in which $C13$ fails (p.104). We have not attemtped to formalise his model.

4 Boffa's Axioms

Boffa [3,4] presented two axiom systems for regular algebra. His first axiomatisation adds a very simple quasi-identity to Conway's classical axioms. In his second paper he shows that some of Conway's axioms—including powerstar—are redundant. He also shows that his second axiomatisation implies the first. We can base the first axiomatisation on *regalg-base*.

Class *boffa-1* = *regalg-base* +
assumes *R*: $x{\cdot}x = x \rightarrow x^* = 1{+}x$

Class *boffa-2* = *dioid-one-zero* + *star-op* +
assumes *B1*: $1{+}x \leq x^*$
and *B2*: $x^*{\cdot}x^* = x^*$
and *B3*: $1{+}x \leq y \wedge y{\cdot}y = y \rightarrow x^* \leq y$

Boffa algebras are closed under duality since all axioms are self-dual.

We first show that *boffa-1* and *boffa-2* are equipollent (*boffa-1* = *boffa-2*). Boffa has already shown that *boffa-2* $\subseteq$ *boffa-1*—the first is a subclass of the second—whereas the converse inclusion is new. Following Boffa, we then relate Boffa's algebras with Conway's classical axioms. In Isabelle, subclass relationships can be captured by subclass or sublocale proofs. We use sublocales simply because the associated syntax leads to more readable statements. In general, an understanding of Isabelle's subclass and locale mechanisms is not needed to grasp the mathematical statements in this paper.

Sublocale *boffa-1* $\subseteq$ *boffa-2*

Isabelle dictates the proof obligations: all *boffa-2* axioms must be derived from *boffa-1*. In fact, only $B1$-$B3$ need to be verified. Isabelle recognises that both algebras extend the class *dioid-one-zero*. All proof obligations were discharged by ATP. All theorems for *boffa-2* are now automatically available for *boffa-1*.

Proving the converse sublocale relationship is more involved. A direct automated proof was impossible within Isabelle's time limits. First, we therefore verified all regular identities that have been proved for Kleene algebras in the repository in the weaker context of *boffa-2*. These 46 facts include well known identities such as $1 \leq x^*$, $x \leq x^*$, $x^* \cdot x^* = x^*$, $x^{**} = x^*$, $1^* = 1$, $0^* = 1$, $1 + x \cdot x^* = x^*$, $(x \cdot y)^* \cdot x = x \cdot (x \cdot y)^*$, and $(x + y)^* = x^* \cdot (y \cdot x^*)^*$. 41 proofs were automatic; for the remaining ones, paper and pencil proofs could be translated. Consider the following proof of $C12$ as an example.

Proof –
have $\forall\, x\ y.\ 1{+}x{\cdot}(y{\cdot}x)^*{\cdot}y = (1{+}x{\cdot}(y{\cdot}x)^*{\cdot}y){\cdot}(1{+}x{\cdot}(y{\cdot}x)^*{\cdot}y)$ — by smt
hence $\forall\, x\ y.\ (x{\cdot}y)^* \leq 1{+}x{\cdot}(y{\cdot}x)^*{\cdot}y$ — by metis
hence $1{+}x{\cdot}(y{\cdot}x)^*{\cdot}y \leq 1{+}x{\cdot}y{+}x{\cdot}y{\cdot}(x{\cdot}y)^*{\cdot}(x{\cdot}y)$ — by smt
hence $1{+}x{\cdot}(y{\cdot}x)^*{\cdot}y \leq (x{\cdot}y)^*$ — by smt ...
thus *?thesis* — by metis ...
qed

The remaining half of *boffa-1* = *boffa-2* is then fully automatic.

Sublocale *boffa-2* ⊆ *boffa-1*

All regular identities are now available also in *boffa-1*.

Deriving Conway's classical axioms from Boffa's algebras again requires some preparation. We need a few general lemmas about the interaction of the star with (sums of) powers, for instance, that $x^n \leq x^*$, $x^k \cdot (x^n)^* = (x^n)^* \cdot x^k$ and $x_0^k \cdot (x^n)^* = (x^n)^* \cdot x_0^k$ for $k \leq n$, and $x_m^n \leq x^*$. Most of them are automatic up to induction. Finally, to derive *powerstar* from $B3$, it suffices to prove the following two facts.

Lemma *conway-powerstar1*: $(x^{n+1})^* \cdot {x_0}^n \cdot (x^{n+1})^* \cdot {x_0}^n = (x^{n+1})^* \cdot {x_0}^n$

Lemma *conway-powerstar2*: $1{+}x \leq (x^{n+1})^* \cdot {x_0}^n$

Their proofs require a case analysis on n. While the $n = 0$ cases are automatic, those for $n \neq 0$ translate paper and pencil proofs. *powerstar* can then be derived automatically in two steps ($\leq$ and $\geq$), and the desired sublocale statement is automatic as well.

Theorem *powerstar*: $x^* = (x^{n+1})^* \cdot {x_0}^n$

Sublocale *boffa-2* ⊆ *conway-classical*

All theorems of Boffa's algebras are now available for Conway's classical axioms.

The subclass relationship is strict. Boffa has shown that his algebras are complete (relative to Krob's result, cf. Section 9); Conway has shown that his classical axioms are not (p. 118). This implies that R cannot be derived in *conway-classical*. We have tried unsuccessfully to test this fact with Sledgehammer and Nitpick. This is not surprising because Conway's counterexample is constructed inductively. Again we have not further attempted to formalise Conway's proof.

5 Conway's Conjectures

Conway presents several extensions of his classical axioms and conjectures their completeness (p. 103). Boffa has verified one of them, Krob the remaining ones relative to *boffa-1* (p. 329f). All completeness results are relative to Krob's completeness proofs of Conway's classical axioms with monoid indentities. Following Boffa, these axioms are derived from Boffa's algebras in Section 9, which shows that Boffa's algebras are complete as well. The (relative) completeness results in this section are obtained by deriving Boffa's axioms. We automatically reconstruct Boffa and Krob's results in the weaker setting of *conway* without *powerstar* by deriving the axioms of *boffa-1* from them. We also establish new equipollence results for Conway's variants. Conway considers dual lefthanded and righthanded variants as well as their combinations. Here we only present the lefthanded ones. Their duals and all dual statements can be found in the repository.

Class *conway-p0* = *conway* +
assumes *P0*: $x{\cdot}y = y{\cdot}z \to x^*{\cdot}y = y{\cdot}z^*$

Class *conway-p1l* = *conway* +
assumes *P1l*: $x{\cdot}y \leq y{\cdot}z \to x^*{\cdot}y \leq y{\cdot}z^*$

Class *conway-p2l* = *conway* +
assumes *P2l*: $x = y{\cdot}x \to x = y^*{\cdot}x$

Class *conway-p3l* = *conway* +
assumes *P3l*: $x{\cdot}y \leq y \to x^*{\cdot}y \leq y$

The rule *P3l* and its dual will reappear in Kozen's axiomatisation.

We establish two results. First, we show that *conway-p2l* is complete. Second, we prove that all lefthanded variants are equipollent, hence complete as well. The following result is automatic:

Sublocale *conway-p2l* $\subseteq$ *boffa-1*

The question whether *conway-p2l* = *boffa-1* remains open. We could neither prove nor refute the remaining inclusion within Isabelle's default time limits, despite the fact that all our regular identities are available in Boffa's algebras.

The regular identities can now be used in *conway-p2l* to prove equipollence of Conway's variants in a completely automatic fashion. As usual, the sublocale mechanism takes care of metalogical aspects such as theorem propagation.

Sublocale *conway-p2l* $\subseteq$ *conway-p3l*

Sublocale *conway-p3l* $\subseteq$ *conway-p1l*

Sublocale *conway-p1l* $\subseteq$ *conway-p2l*

Finally we show for $i = 1, 2, 3$ that the combination of *conway-pil* and *conway-pir* is equipollent to *conway-p0*. Here we only present the result for $i = 2$.

Class *conway-p2* = *conway-p2l* + *conway-p2r*

Sublocale *conway-p0* ⊆ *conway-p2*

Sublocale *conway-p2* ⊆ *conway-p0*

6 Kozen's Kleene Algebras

Kozen's Kleene algebras are essentially *conway-p3* with $C11$-$C13$ replaced by a simpler axiom. Kozen gave an elementary completeness proof for his variant based on Conway's trick of encoding finite automata in terms of a matrix regular algebra over a regular algebra. This proof has recently been formalised in the proof assistant Coq [5]. Boffa proved completeness for left Kleene algebras, where axiom $P3r$ is absent, relative to *boffa-2* (it seems that Kozen's proof does not go through in this weaker context).

We reconstruct Boffa's completeness result and prove new results that establish equipollence of Kleene algebras and Conway's variants. Finally, we reproduce well known equipollence results between two variants of Kleene algebra introduced by Kozen. As usual, we stick to the left. Dual classes and statements can be found in the repository.

Class *kozen-base-l* = *dioid-one-zero* + *star-op* +
assumes *star-unfoldl'*: $1+x\cdot x^* \leq x^*$

Class *kozen-1l* = *kozen-base-l* +
assumes *star-inductl*: $x\cdot y \leq y \rightarrow x^*\cdot y \leq y$

Class *kozen-2l* = *kozen-base-l* +
assumes *star-inductl-var*: $z+x\cdot y \leq y \rightarrow x^*\cdot z \leq y$

Class *kozen* = *kozen-1l* + *kozen-1r*

Conceptually, completeness of *kozen-1l* (and its dual) follows from the equipollence results below. Technically, however, the corresponding sublocale proof is particularly simple and automatic; it also brings the regular identities into the scope of *kozen-1l* for equipollence proofs.

Sublocale *kozen-1l* ⊆ *boffa-2*

Sublocale *kozen-1l* ⊆ *conway-p2l*

Sublocale *conway-p2l* ⊆ *kozen-1l*

Sublocale *kozen* ⊆ *conway-p0*

Sublocale *conway-p0* ⊆ *kozen*

All proofs are fully automatic. They show that *conway-pil* = *kozen-1l* and *conway-pi* = *kozen*. Finally, we establish equipollence of Kozen's variants.

Sublocale *kozen-1l* ⊆ *kozen-2l*

Sublocale *kozen-2l* ⊆ *kozen-1l*

Once more we were unsuccessful in testing whether Kozen's algebras are equipollent to Boffa's within Isabelle's default time limits.

7 Salomaa's Axioms

Salomaa's axioms are based on dioids without 1, since in the presence of the Kleene star, 1 can be defined as 0^*. Boffa has observed that idempotency is redundant in this setting. As before we base Salomaa's axiomatisation on dioids to keep the development simple and uniform.

Salomaa presents two axiom systems, proves completeness for the first and conjectures that property for the second one. His completeness proof uses an algebraic abstraction of Arden's well known rule for solving linear equations over regular languages (axiom *salomaa*). Since a precondition of Arden's rule is the absence of the empty word property—some language must not contain the empty word—Salomaa inductively defines the negation of property *ewp* for regular algebra terms (or regular expressions). Due to this, one of his axioms is not defined for first-order variables, but for substitution instances of terms.

To circumvent this complication we define *ewp* abstractly with respect to a property that holds in the case of regular languages, as we show in the next section. This property suffices for our completeness proof. It can safely be replaced by stronger (inductive) properties that imply it.

Class *salomaa-ewp* = *dioid-one-zero* + *star-op* +
  fixes *ewp* :: $'a \Rightarrow bool$
  assumes *S11*: $(1+x)^* = x^*$
  and *S12*: $x^* = 1+x^*\cdot x$
  and *ewp-form* : $ewp\ x \leftrightarrow (\exists\, y.\ x = 1+y \wedge \neg\ ewp\ y)$

Class *salomaa* = *salomaa-ewp* +
  assumes *salomaa* : $(\neg\ ewp\ y) \wedge x = x\cdot y+z \rightarrow x = z\cdot y^*$

Class *salomaa-conj* = *salomaa-ewp* +
  assumes *salomaa-small* : $(\neg\ ewp\ y) \wedge x = x\cdot y+1 \rightarrow x = y^*$

Property *ewp-form* states that the empty word can be isolated from every language that contains it. We can easily reconstruct the following relationship [3].

Sublocale *salomaa* ⊆ *salomaa-conj*

salomaa = *salomaa-conj* could be refuted by a 3-element counterexample. We have not tested whether this would still hold for stronger variants of *ewp*.

Boffa has presented a completeness proof of *salomaa-conj* relative to *boffa-1*. We provide a new direct completeness proof of *salomaa* relative to *kozen-1r* and briefly argue why Boffa's proof contains a gap.

Proving *star-inductr-var* from *salomaa* yields completeness automatically.

Lemma *kozen-induct*: $y{\cdot}x{+}z \leq y \rightarrow z{\cdot}x^* \leq y$
Proof (*cases ewp x*)
case *False* **thus** *?thesis* — one step by metis
next
case *True* **thus** *?thesis* — several steps by metis and smt, using *ewp-form*
qed

Sublocale *salomaa* $\subseteq$ *kozen-2r*

The proof of *kozen-induct* illustrates the fact that reasoning with Salomaa's axioms typically requires case analyses on *ewp* and the trick of using *ewp-form* to reduce the negative case to one where *salomaa* can again be applied.

Such a case analysis is needed in the completeness proof of *salomaa-conj*, but omitted by Boffa [3]. We have attempted a complete case analysis for $C11$ but failed with manual proofs based on Boffa's paper as well as with automated and interactive attempts. Also, Nitpick could not find a counterexample. As far as we can tell, completeness of *salomaa-conj* therefore remains open.

8 Soundness

We now prove soundness of Salomaa's axioms and Kleene algebras, which in this context means that the regular languages form models of these axioms. By our sublocale relationships this implies soundness of all the other regular algebras investigated (cf. Figure 1). The main step is proving Arden's lemma (i.e. axiom *salomaa*) at the language level, for which we could have reused a previous formalisation in Isabelle [9]. Access to the algebraic level, however, significantly simplifies this previous development. Only a few non-automatic non-algebraic proofs are needed.

As usual in Isabelle, words are represented as lists; @ denotes word concatenation. To enhance automation we introduce elimination rules for higher-order concepts. They can be used for simplification before calling Sledgehammer.

type-synonym *'a lan* = *'a list set*

Definition *l-prod* :: *'a lan* $\Rightarrow$ *'a lan* $\Rightarrow$ *'a lan* (**infixr** $\cdot$ *75*)
where $X{\cdot}Y = \{v@w \mid v\ w.\ v{\in}X \wedge w{\in}Y\}$

Lemma *l-prod-elim*: $w{\in}X{\cdot}Y \leftrightarrow (\exists\, u\ v.\ w = u@v \wedge u{\in}X \wedge v{\in}Y)$

We can directly show by an interpretation statement that regular languages form dioids (though that might not be immediately evident from Isabelle's syntax).

Interpretation *dioid-one-zero* (*op* $\cup$) *l-prod* (*op* $\subseteq$) (*op* $\subset$) {[]} {}

We can now use the function *power* from *dioid-one-zero* to define the Kleene star of a language as usual (*powsum* would only yield finite sums). We also define the empty word property in the obvious way.

Definition *star* :: *'a lan* $\Rightarrow$ *'a lan* (-* [*101*] *100*)

where $X^* = (\bigcup n.\ X^n)$

Definition *l-ewp* $X \leftrightarrow \{[]\} \subseteq X$

Lemma *star-elim*: $x{\in}X^* \leftrightarrow (\exists k.\ x{\in}X^k)$

To show that regular languages form Kleene algebras, only two continuity properties are needed. Both are automatic after calling Isabelle's simplifier.

Lemma *star-contl*: $X{\cdot}Y^* = (\bigcup n.\ X{\cdot}Y^n)$

Lemma *star-contr*: $X^*{\cdot}Y = (\bigcup n.\ X^n{\cdot}Y)$

Interpretation *kozen* (*op* $\cup$) *l-prod* (*op* $\subseteq$) (*op* $\subset$) $\{[]\}$ $\{\}$ *star*

Only the verification of the unfold rules required a few interactions. All regular identities are now available for regular languages and can be used in the remaining step; the derivation of Arden's rule, which verifies axiom *salomaa*.

In fact, only an inequality remains to be shown since one half of the proof is already covered by axiom *star-inductr-var* of Kleene algebra. Part of this inequality can be captured at the abstract algebraic level as well.

Lemma (**in** *boffa-1*) *arden-aux*: $y \leq y{\cdot}x{+}z \rightarrow y \leq y{\cdot}x^{Suc\ n}{+}z{\cdot}x^*$

Its proof translates an inductive paper and pencil argument. It now suffices to show that—under the conditions of axiom *salomaa* interpreted in regular languages— the term $Y \cdot X^{Suc\ n}$ vanishes. Following the textbook proofs of Arden's lemma, this is the case since the length of minimal words in $Y \cdot X^n$ grows proportionally to n, hence all words in Y die out in $Y{\cdot}X^n$ for n sufficiently large. We formalise this using two elementary facts about lower bounds of word lengths in languages.

Lemma *prod-lb*: $(\forall\, w{\in}X.\ m{\leq}|w|\ ) \rightarrow (\forall\, w{\in}Y.\ n{\leq}|w|\ ) \rightarrow (\forall\, w{\in}X{\cdot}Y.\ m{+}n{\leq}|w|\ )$

Lemma *power-lb*: $(\forall\, w{\in}X.\ k{\leq}|w|\ ) \rightarrow (\forall\, w.\ w{\in}X^{Suc\ n} \rightarrow n{*}k{\leq}|w|\ )$

Lemma *word-suicide*: $\neg$ *l-ewp* $X \rightarrow Y \neq \{\} \rightarrow (\forall\, w{\in}Y.\ \exists\, n.\ w{\notin}Y{\cdot}X^{Suc\ n})$

Only *power-lb* requires induction and some user interaction in the induction step. The proof of *word-suicide* is calculational with 3 intermediate steps. Together with *arden-aux* it is used in the following soundness result, which now is completely automatic.

Interpretation *salomaa op* $\cup$ *l-prod op* $\subseteq$ *op* $\subset$ $\{[]\}$ $\{\}$ *star l-ewp*

9 Relative Completeness

Krob has proved completeness of Conway's classical axioms extended by the following rule: If $x_i \cdot x_j \leq x_{i\circ j}$ and $(x_{i,i})^* = x_{i,i}$ hold for all $i, j \in I$, then $(\sum x_i)^* = \sum x_i$ (p. 116 of Conway's monograph). In this definition, I is a finite monoid, $\sum$ indicates summation over I and $x_{i,j} = \sum_{ik=j} x_k$. The discussion

of this schematic rule—which has been called *monoid identities* by Krob—and of Krob's proof requires group theory beyond the scope of this paper; a short sketch can be found in Conway's monograph.

Perhaps surprisingly, Boffa has shown that the monoid identities are derivable from *boffa-1* by purely elementary reasoning. Relative to Krob's result, this establishes completeness of *boffa-1* (hence of all algebras in Figure 1).

We now reconstruct Boffa's proof in Isabelle. While his original proof covers just a few lines, a certain amount of theory infrastructure must be developed in Isabelle beforehand. First, abstract axiomatic reasoning—as in the previous sections—is no longer sufficient; an axiomatisation of Boffa's algebras based on carrier sets is needed. Second, finite sums need to be implemented for algebras with carrier sets since they are not available in Isabelle's standard library. Third, elements of Boffa's algebras must be modelled as functions from a finite monoid into a Boffa algebra in order to capture indexing.

Algebras with explicit carrier sets can be found in the Isabelle library, however, the associated syntax is not well documented and some constructions used in this section may therefore remain somewhat obscure. We have implemented dioids along these lines and proved some essential properties.

locale *dioid* = *weak-partial-order D* **for** *D* (**structure**) +
  assumes *add-closed*: ⟦*x*∈*carrier D*; *y*∈*carrier D*⟧ ⇒ *x*+*y*∈*carrier D*
  — and further closure conditions
  and *mult-assoc*: ⟦*x*∈*carrier D*; *y*∈*carrier D*; *z*∈*carrier D*⟧ ⇒ *x*·(*y*·*z*) = (*x*·*y*)·*z*
  — and the remaining dioid axioms

Algebraic structures are now parametrised with respect to their carrier set, and closure conditions for all operations must be added. ATP systems must check these additional conditions, which involve some simple set expressions. At the level of dioids, however, this has little impact on their performance.

The most natural way of defining finite sums over dioids with carrier sets would be using a fold function, as does Isabelle's *setsum* operator without carriers. For automated theorem proving, however, it turns out to be much simpler to define this (partial) recursive function by locale extension.

locale *dioid-finsup* = *dioid D* **for** *D* (**structure**) +
  assumes *finsup-closed* : ⟦*finite A*; *A* ⊆ *carrier D*⟧ ⇒ Σ*A*∈*carrier D*
  and *finsup-empty*: Σ{} = **0**
  and *finsup-insert*: ⟦*A* ⊆ *carrier D*; *finite A*; *x*∈*carrier D*⟧ ⇒ Σ(*insert x A*) = *x*+Σ*A*

We have developed a basic library for sums, in particular for their interaction with the dioid operations. Typical examples are $\sum A \leq y \leftrightarrow \forall x \in A.x \leq y$, $\sum(A \cup B) = (\sum A) + (\sum B)$, and $(\sum A) \cdot (\sum B) = \sum\{a \cdot b \mid a \in A, b \in B\}$, whenever A and B are finite sets. Their proofs are the least automatic ones in the paper, since side conditions on the elements and sets involved need to be processed. All individual proof steps, however, could still be discharged automatically, sometimes after simplifying. We expect that the degree of automation can significantly be increased in a more thoroughly designed library.

Next we axiomatise *boffa1* with carrier sets.

locale *boffa1* = *dioid-finsup B* **for** *B* (**structure**) +

assumes *star-closed*: $x \in carrier\ B \Rightarrow x^* \in carrier\ B$
and *C11*: $[\![x \in carrier\ B;\ y \in carrier\ B]\!] \Rightarrow (x+y)^* = (x^* \cdot y)^* \cdot x^*$
and *C12*: $[\![x \in carrier\ B;\ y \in carrier\ B]\!] \Rightarrow (x \cdot y)^* = \mathbf{1} + x \cdot (y \cdot x)^* \cdot y$
and *R*: $x \in carrier\ B \Rightarrow x \cdot x = x \rightarrow x^* = \mathbf{1} + x$

We now link this algebra with the index monoid I. First, we define I—parametrised by the carrier of the algebra—as an arbitrary set that is mapped by a function x—again parametrised by the carrier—into the regular algebra. The record *'a boffa* provides the signature for the locale *boffa1*.

record $('a, 'b)$ *boffa-gen* $=$ *'a boffa* $+$
gen-set :: *'b set* (I_1)
gen :: *'b* $\Rightarrow$ *'a* $(x_{1\_})$

locale *boffa-gen* $=$ *boffa1 G* **for** *G* (**structure**) $+$
assumes *gen-closed*: $i \in I \Rightarrow x_i \in carrier\ G$

We can then impose the monoid structure and finiteness constraint on I.

record $('a, 'b)$ *boffa-monoid* $= ('a, 'b)$ *boffa-gen* $+$
comp :: $['b, 'b] \Rightarrow 'b$ (**infix** $\circ_1$ *80*)
unit :: *'b* (e_1)

locale *boffa-monoid* $=$ *boffa-gen G* **for** *G* (**structure**) $+$
assumes *gen-finite*: *finite I*
and *comp-closed*: $[\![i \in I;\ j \in I]\!] \Rightarrow i \circ j \in I$
and *unit-closed*: $e \in I$
and *comp-assoc*: $[\![i \in I;\ j \in I;\ k \in I]\!] \Rightarrow i \circ (j \circ k) = (i \circ j) \circ k$
and *unit-left*: $i \in I \Rightarrow e \circ i = i$
and *unit-right*: $i \in I \Rightarrow i \circ e = i$

This infrastructure allows us to write down Conway's monoid identities in Isabelle. Deriving them requires about 10 additional lemmas on the interaction of the monoid and the regular algebra. To shorten expressions we write $\{x_i\}$ instead of $\{x_i \mid i \in I\}$ and similarly $\{x_i \cdot x_j\}$ or $\{x_{i \circ j}\}$ when indices range over I. We have shown, for instance, that the set $\{x_i \mid i \in A\}$ is a finite subset of the carrier G of our algebra and that $\sum\{x_i \mid i \in A\} \in G$, for every $A \subseteq I$. Another example is that $\{x_i \cdot x_j\}$ is a finite subset of G and the sum over this set an element of G. Finally, we have shown that the image of the monoidal unit e under x can be isolated from sums: $\sum\{x_i\} = x_e + \sum\{x_i \mid i \in (I - \{e\})\}$. Most corresponding proofs are fully automatic.

The final missing step is the implementation of the pair notation $x_{i,j}$.

Definition *mon-pair* :: $('a, 'b, 'c)$ *boffa-monoid-scheme* $\Rightarrow$ *'b* $\Rightarrow$ *'b* $\Rightarrow$ *'a* $(x_{1\_,\_})$
where $x_{G\,i,j} = \Sigma_G\{x_{G\,k} \mid k.\ k \in I_G \wedge i \circ_G k = j\}$

For syntactic reasons, the index G refers to the underlying carrier set. The following lemma corresponds to the first step in Boffa's proof [3].

Lemma *mon-pair-split*: $(\forall\, i \in I.\ \forall\, j \in I.\ x_{i,j}{}^* = x_{i,j}) \Rightarrow \Sigma\{x_i\} = \mathbf{1} + \Sigma\{x_i\}$

Its proof translates Boffa's reasoning more or less directly. The remaining two lemmas formalise properties that have been left implicit in Boffa's next steps.

Lemma *aux1*: $\{x_{i \circ j}\} = \{x_i\}$

Lemma *aux2*: $(\forall\, i{\in}I.\ \forall\, j{\in}I.\ x_i{\cdot}x_j \leq x_{i \circ j}) \ \Rightarrow \Sigma\{x_i{\cdot}x_j\} \leq \Sigma\{x_{i \circ j}\}$

By Lemma *aux1*, summing over all elements $i \circ j$ of I means summing over all elements i. Lemma *aux2* helps to lift the assumption in Conway's rule that x is a submorphism to the level of suprema. Therefore, the map x from the monoid I into the Boffa algebra B is "almost" an embedding.

Finally, these three lemmas allow us to feed Boffa's remaining proof of Conway's rule directly into Isabelle, verifying all his proof steps automatically.

Theorem *mon-id*: $(\forall\, i{\in}I.\ \forall\, j{\in}I.\ x_i{\cdot}x_j \leq x_{i \circ j} \wedge {x_{i,j}}^* = x_{i,j}) \Rightarrow (\Sigma\{x_i\})^* = \Sigma\{x_i\}$
Proof –
assume $\forall\, i{\in}I.\ \forall\, j{\in}I.\ x_i{\cdot}x_j \leq x_{i \circ j} \wedge {x_{i,j}}^* = x_{i,j}$
— preparatory steps on the assumption
have $(\Sigma\{x_i\}){\cdot}(\Sigma\{x_i\}) = (\mathbf{1}{+}\Sigma\{x_i\}){\cdot}(\mathbf{1}{+}\Sigma\{x_i\})$ — by smt
also have ... $= \mathbf{1}{+}(\Sigma\{x_i\}){+}(\Sigma\{x_i\}){\cdot}(\Sigma\{x_i\})$ — by smt
also have ... $= \mathbf{1}{+}(\Sigma\{x_i\}{+}\Sigma\{x_i{\cdot}x_j\})$ — by simplification
ultimately have $(\Sigma\{x_i\}){\cdot}(\Sigma\{x_i\}) = \Sigma\{x_i\}$ — by smt
thus $(\Sigma\{x_i\})^* = \Sigma\{x_i\}$ — by smt, essentially R and *mon-pair-split*
qed

This last theorem establishes completeness of all regular algebras in our hierarchy relative to Krob's proof (cf. Figure 1). Formalising this result fully in Isabelle would require linking our abstract implementations of algebras with the carrier based ones. Unfortunately, to our knowledge, this is impossible. Alternatively, we could have based the entire development on carrier sets. But that seems mathematically rather unnatural and it hampers proof automation.

10 Conclusion

We have reconstructed the fine structure of regular algebras within Isabelle based on the Sledgehammer tool for automated theorem proving and on automated counterexample search. The main emphasis was on known completeness results, yet some new findings clarify the overall picture in Figure 1.

As an exercise in computer enhanced mathematics, our study underlines the impressive potential of integrated automated and interactive proof technology for the working mathematician. Automation of axiomatic algebraic reasoning left little to desire; that of moderately difficult higher-order and integrated reasoning (e.g. by induction, with algebra, numbers or sets) was still reasonably high. The most complex proofs could be translated directly and rather quickly from paper and pencil proofs and automated step by step. The hardest work was certainly in library design. Overall, formalising regular algebras in this new kind of integrated environment seems reasonably lightweight and natural from a mathematician's point of view. Results that eminent scientists found worth publishing could be reconstructed with relative ease and a high degree of automation.

We end with some remarks on proof technology. Isabelle proof reconstruction often requires proof search. This remains a bottleneck. Standardised detailed

ATP output would support fast microstep proof reconstruction even when proof search takes time. Standardised type support for ATP seems desirable for heterogeneous mathematical reasoning. Sledgehammer calls five ATP systems and the SMT solver Z3 (cf. [2]). Having them all is certainly a gain, but Z3 showed definitely the most consistent performance. In Isabelle, the gap between abstract and carrier-based structures inhibits smooth mathematical reasoning. A less rigid proof scripting language could yield simpler and less verbose ATP-based proofs: assumption contexts are managed by the relevance filter; hence detailed control at command level—which determines the scripting syntax— seems unnecessary.

Acknowledgements. We are grateful to Geoff Sutcliffe and the München Isabelle group for making ATP/SMT systems freely available over the Internet.

References

1. Ballarin, C.: Tutorial to locales and locale interpretation. In: Lambán, L., Romero, A., Rubio, J. (eds.) Contribuciones Científicas en honor de Mirian Andrés. Servicio de Publicationes de la Universidad de La Rioja (2010)
2. Blanchette, J.C., Bulwahn, L., Nipkow, T.: Automatic Proof and Disproof in Isabelle/HOL. In: Tinelli, C., Sofronie-Stokkermans, V. (eds.) FroCos 2011. LNCS (LNAI), vol. 6989, pp. 12–27. Springer, Heidelberg (2011)
3. Boffa, M.: Une remarque sur les systèmes complets d'identités rationnelles. Informatique théorique et Applications 24(4), 419–423 (1990)
4. Boffa, M.: Une condition impliquant toutes les identités rationnelles. Informatique théorique et Applications 29(6), 515–518 (1995)
5. Braibant, T., Pous, D.: An Efficient Coq Tactic for Deciding Kleene Algebras. In: Kaufmann, M., Paulson, L.C. (eds.) ITP 2010. LNCS, vol. 6172, pp. 163–178. Springer, Heidelberg (2010)
6. Conway, J.H.: Regular Algebra and Finite Machines. Chapman and Hall (1971)
7. Guttmann, W., Struth, G., Weber, T.: Automating Algebraic Methods in Isabelle. In: Qin, S., Qiu, Z. (eds.) ICFEM 2011. LNCS, vol. 6991, pp. 617–632. Springer, Heidelberg (2011)
8. Kozen, D.: A completeness theorem for Kleene algebras and the algebra of regular events. Information and Computation 110(2), 366–390 (1994)
9. Krauss, A., Nipkow, T.: Proof pearl: Regular expression equivalence and relation algebra. J. Autom. Reasoning (2011)
10. Krob, D.: Complete systems of $\mathcal{B}$-rational identities. Theoretical Computer Science 89, 207–343 (1991)
11. Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL. LNCS, vol. 2283. Springer, Heidelberg (2002)
12. Salomaa, A.: Two complete axiom systems for the algebra of regular events. J. ACM 13(1), 158–169 (1966)