An Evaluation of Automated Theorem Proving in Regular Algebra (Extended Abstract)

Alasdair Armstrong

Department of Computer Science, The University of Sheffield, UK
a.armstrong@dcs.shef.ac.uk

Introduction The Isabelle/HOL environment [8] combines the power of automated reasoning with higher-order features for theory engineering and proof management. Its built-in *Sledgehammer* tool integrates state of the art ATP and SMT tools, allowing for powerful automated reasoning in proofs [2]. Theory engineering features such as typeclasses and locales support the effective design of large theory hierarchies and allow for theorem propagation in these hierarchies [6]. They also allow for the connection between abstract algebras and concrete models.

When using regular algebras such as $*$-continuous Kleene algebras or quantales, many proof obligations can be discharged using only first-order axioms. There are many first-order regular algebras, such as Kleene algebras, Pratt's action algebras and Boffa's algebras. Action algebras are particularly interesting as they have an axiomatisation based on Galois connections and an equivalent purely equational axiomatisation [9]. This result is interesting, as one might expect that purely equational axioms might be more amenable to ATP than other sets of axioms. However, this purely equational axiomatisation requires a large signature. This paper attempts to ascertain which of action or Kleene algebras are better from an ATP standpoint, providing insight into the trade-off between an equational axiomatisation and a larger signature.

To achieve this, Isabelle's built in benchmarking tool *Mirabelle* is used on facts from a large repository[1] for algebraic methods in Isabelle which has been documented in previous papers [5]. Recently the repository has been extended to support higher-order regular algebra, and many higher-order order concepts such as Galois connections have been implemented in this setting [1]. We have also begun work on extending this repository with explicit carrier sets, allowing us to formalise concepts such as Galois connections between different partial orders, subalgebras and other concepts from universal algebra. Even in this higher-order setting, automated reasoning remains a valuable tool. However, a question remains—exactly *how* much does the carrier set based axiomatisation these features require impact the usefulness of ATP and SMT tools?

Regular Algebra A *dioid* is a structure $(D, +, \cdot, 0, 1)$ where $(D, +, 0)$ is a semilattice with least element 0, $(D, \cdot, 1)$ is a monoid, $\cdot$ distributes over $+$ from

[1] http://staffwww.dcs.shef.ac.uk/people/G.Struth/isa/

both the left and right, and $0 \cdot x = 0 = x \cdot 0$. An *action algebra* [9] is an algebra $(A, +, 0, \cdot, 1, \leftarrow, \rightarrow, ^*)$ such that $(A, +, \cdot, 0, 1)$ is a dioid, and satisfying

$$x \leq z \leftarrow y \overset{L}{\Leftrightarrow} xy \leq z \overset{R}{\Leftrightarrow} y \leq x \rightarrow z, \tag{1}$$

$$1 + x^*x^* + x \leq x^*, \qquad 1 + yy + x \leq y \Rightarrow x^* \leq y \tag{2, 3}$$

where the natural partial order $\leq$ is defined as $x \leq y \Leftrightarrow x + y = y$.

A *Galois connection* between two posets $(A, \leq_A)$ and $(B, \leq_B)$ is a pair of functions $f : A \rightarrow B$ and $g : B \rightarrow A$ such that the following equivalence holds

$$f(x) \leq_B y \Leftrightarrow x \leq_A g(y).$$

The function f is called the *lower adjoint* of the Galois connection, while g is the upper adjoint. Axiom (1) defines the left and right *residuals*, $\leftarrow$ and $\rightarrow$ as the upper adjoints of $(\cdot y)$ and $(x \cdot)$. (1L) and (1R) thus describe families of Galois connections indexed by y and x respectively. Galois connections have many interesting properties, from which the properties of residuation are derived.

Pratt's main result is that there exists an equivalent set of axioms for action algebra which are purely equational. These axioms are shown below:

$$\begin{array}{cc} x \rightarrow y \leq x \rightarrow (y + z), & x(x \rightarrow y) \leq y \leq x \rightarrow xy, \\ y \leftarrow x \leq (y + z) \leftarrow x, & (y \leftarrow x) \cdot x \leq y \leq yx \leftarrow x, \\ x^* \leq (x + y)^*, & 1 + x^*x^* + x \leq x^*, \end{array}$$
$$(x \rightarrow x)^* \leq x \rightarrow x.$$

A Kleene algebra [7] is a structure $(K, +, \cdot, 0, 1, ^*)$ where $(K, +, \cdot, 0, 1)$ is a dioid, satisfying the following 4 axioms:

$$1 + xx^* \leq x^*, \qquad 1 + x^*x \leq x^*, \tag{4, 5}$$

$$z + xy \leq y \Rightarrow x^*z \leq y, \qquad z + yx \leq y \Rightarrow zx^* \leq y. \tag{6, 7}$$

The axioms of Kleene algebra can be derived from those of action algebra.

A *quantale* (also called a standard Kleene algebra or S-Algebra by Conway [4]) is a structure $(Q, \bigvee, \cdot)$ such that $(Q, \bigvee)$ is a complete lattice, $\cdot$ is associative, and satisfying the infinite distributivity laws

$$x\left(\bigvee_{y \in Y} y\right) = \bigvee_{y \in Y} xy \qquad \text{and} \qquad \left(\bigvee_{y \in Y} y\right) x = \bigvee_{y \in Y} yx,$$

where $Y \subseteq Q$. The residuation operators from action algebra are defined as

$$x \rightarrow y = \bigvee \{z | xz \leq y\} \qquad \text{and} \qquad y \leftarrow x = \bigvee \{z | zx \leq y\}.$$

It is easy to show that these residuals satisfy (1L) and (1R). A quantale is *unital* if $\cdot$ has an identity element 1. The star operation can then be naturally defined in two ways, either as

$$x^* = \mu(\lambda y. 1 + xy) \qquad \text{or} \qquad x^* = \bigvee_{n \in \mathbb{N}} x^n$$

both of which can be shown to be equivalent by way of Kleene's fixed point theorem. It can then be shown that $(Q, \vee, 0, \cdot, 1, \leftarrow, \rightarrow, ^*)$ is an action algebra and $(Q, \vee, \cdot, 0, 1, ^*)$ is a Kleene algebra. ATP tools are usually limited to first-order logic, and as such are unable to deal with higher-order axioms such as the infinite distributivity laws above. Nevertheless, in a quantale many proof goals can be discharged with first order statements derived from these first order axioms [1]. It is therefore interesting to know which first order regular algebras are most useful from an ATP perspective, as many can be defined and used in this higher-order setting.

Experiments In the remainder of this paper, two experiments are conducted. First, action algebras and Kleene algebras are benchmarked to find if either allows for faster automated proofs. Some analysis is then conducted to ascertain whether there are many theorems of both action algebra and Kleene algebra which can be proven automatically in one but not the other.

The second experiment concerns the performance impact of using explicit carrier sets when formalising regular algebra in Isabelle. By formalising Kleene algebra with and without explicit carrier sets, the effect they have on ATP support can be investigated.

First Experiment The first experiment analyses the time efficiency of the provers used by Isabelle's Sledgehammer tool when applied to 20 problems in both action algebra and Kleene algebra. Sledgehammer itself has already been extensively tested and benchmarked in [3].

Why only 20 facts? To ensure fairness only properties that could be derived directly from the axioms within a 300 second period were considered. The reason for this is simple—in an ordinary Isabelle workflow, one starts by proving useful lemmas which are used in later proofs. For example, one might want to prove that * is isotone, $x \leq y \Rightarrow x^* \leq y^*$. In both Kleene and action algebra this proof might require several auxiliary lemmas, however, the ideal set of lemmas would be different in both cases. The order in which results are proved is quite important, and a specific order would bias the results towards a certain algebra.

While this approach ensures fairness, it has a downside. There are relatively few properties which can be derived directly from the axioms in both algebras in a reasonable timeframe, limiting the sample size to a small set of 20 lemmas.

Isabelle's built in benchmarking tool Mirabelle was used to measure the performance for each of the provers for these 20 problems. Table 1 displays the running times in seconds for each of the various provers. In practice all the provers are run together, with the fastest prover 'winning', so for each problem we take the minimum time given by any prover. This is shown in Table 2.

There is clearly some correlation between the time it takes to find a proof in action algebra and the time it takes in Kleene algebra. The correlation coefficient is roughly 0.45 — i.e. not uncorrelated, but not perfectly correlated either.

The provers used in these experiment are E, Z3, Vampire and SPASS. Z3 and Vampire were run remotely. This set of provers is essentially the default set of provers used by Sledgehammer. The only prover which appears noticeably worse at a specific algebra is SPASS. This is misleading, as SPASS tends to either

	E			remote_z3			remote_vampire			spass		
#	KLE	ACT	diff	KLE	ACT	diff	KLE	ACT	diff	KLE	ACT	diff
1	110.61	110.2	-0.41	**F**	**F**	**F**	7.25	10.15	2.90	193.06	1.39	-191.67
2	103.36	0.78	-102.58	**F**	1.53	**F**	6.02	2.00	-4.02	100.47	0.10	-100.37
3	1.58	**F**	**F**	1.44	**F**	**F**	1.10	72.61	71.51	0.12	100.42	100.30
4	1.06	116.19	115.13	1.42	**F**	**F**	1.08	3.78	2.71	0.09	3.95	3.85
5	1.11	116.52	115.42	1.41	**F**	**F**	1.07	13.2	12.13	0.09	3.97	3.89
6	100.66	111.01	10.35	**F**	**F**	**F**	41.24	44.77	3.53	134.07	**F**	**F**
7	36.45	100.33	63.93	1.49	**F**	**F**	37.95	2.21	-35.74	100.41	100.25	-0.16
8	0.84	0.94	0.1	1.42	1.45	0.03	1.86	3.00	1.15	0.09	1.50	1.41
9	0.80	1.91	1.12	1.39	1.4	0.01	1.89	1.98	0.10	0.08	0.11	0.02
10	105.05	0.99	-104.06	22.57	1.4	-21.18	29.36	28.01	-1.35	104.88	0.89	-103.10
11	139.5	100.98	-38.52	**F**	3.24	**F**	27.04	29.14	2.10	113.64	100.21	-13.43
12	1.97	41.36	39.4	**F**	**F**	**F**	1.50	3.20	1.60	135.05	**F**	**F**
13	105.2	0.87	-104.29	25.2	1.39	-23.82	3.90	1.95	-1.93	100.77	0.13	-100.64
14	102.1	39.72	-62.38	3.27	**F**	**F**	1.39	7.66	6.27	114.47	153.76	39.29
15	100.89	0.91	-99.98	1.47	1.41	-0.06	**F**	2.01	**F**	0.49	0.23	-0.27
16	114.02	100.83	-13.2	**F**	**F**	**F**	125.65	40.98	-84.67	**F**	51.97	**F**
17	**F**	**F**	**F**	**F**	3.14	**F**	**F**	31.29	**F**	**F**	10.37	**F**
18	61.82	100.85	39.03	1.82	**F**	**F**	29.31	2.23	-27.07	100.35	100.46	0.10
19	0.36	0.37	0.02	1.38	1.34	-0.04	1.09	**F**	**F**	0.11	0.09	-0.01
20	0.82	0.84	0.03	1.6	**F**	**F**	1.13	1.09	-0.04	0.10	0.11	0.01
			-140.92			-45.09			-50.75			-361.66

Table 1. Prover running times (s) for action and Kleene algebras

perform extremely well for some problems or extremely slowly for others. The small sample size could therefore skew the result in one direction or the other.

What is clear however, is that having a variety of provers is undoubtedly a good thing. Each prover has different strengths and weaknesses in this problem domain. Z3 is usually extremely fast when it succeeds, even though it sometimes cannot find a proof (represented by an **F** in Table 1). E and Vampire tend to be the most reliable, finding proofs when other provers fail.

At first glance action algebra might appear to be slightly faster. However, there is no statistically significant difference overall between using action algebra and Kleene algebra when all the provers are taken into account (in Table 2).

#	Kleene	action	difference	#	Kleene	action	difference
1	7.25	1.39	-5.86	11	27.04	3.24	-23.80
2	6.02	0.10	-5.91	12	1.50	3.20	1.70
3	0.12	72.61	72.49	13	3.88	0.13	-3.75
4	0.09	3.78	3.69	14	1.39	7.66	6.27
5	0.09	3.97	3.89	15	0.49	0.23	-0.27
6	41.24	44.77	3.53	16	114.02	40.98	-73.04
7	1.49	2.21	0.72	17	0.00	3.14	3.14
8	0.09	0.94	0.85	18	1.82	2.23	0.42
9	0.08	0.11	0.02	19	0.10	0.09	-0.01
10	22.57	0.89	-21.69	20	0.10	0.11	0.01

Table 2. Minimum prover times (s) for action and Kleene algebras

Another useful way to compare the two algebras is to compare how much can be automatically proven from the axioms, not just how quickly. Using the axioms of Kleene algebra, the sliding rule, $(xy)^*x \leq x(yx)^*$ and a variant $x^*x \leq xx^*$ can be automatically proven. Properties such as $x^* \leq (x^*)^*$, and that the star is isotone can also be automatically derived using just the axioms of Kleene algebra. These properties cannot be automatically derived in action algebra. Furthermore one can show that $xy \leq y \Rightarrow x^*y \leq y$ and $yx \leq y \Rightarrow yx^* \leq y$. These two properties are also easy to show in action algebra, provided that one has already proven that the star is isotone—which, as mentioned, cannot automatically be derived from the axioms of action algebra via Sledgehammer.

Overall, the choice of first order regular algebra seems to matter little. The performance difference is not significance, and one could simply derive the axioms of Kleene algebra from those of action algebra anyway.

Second Experiment While there is no significant difference between action algebras and Kleene algebras (both without explicit carrier sets) in terms of performance, is there a significant difference in ATP usefulness when formalising algebras with and without explicit carrier sets in Isabelle?

An algebra, such as a dioid $(D, +, \cdot, 0, 1)$, has an explicit carrier set if we represent D using an actual set in Isabelle rather than the set being implicit in the type. This requires us to add additional axioms to our algebras, stating that the operations are closed with respect to the carrier set. For example, in the dioid case, axioms would be needed stating that if $x \in D$ and $y \in D$ then $x + y \in D$ and $x \cdot y \in D$, and also that $0, 1 \in D$.

This increase in the number of axioms might lead to a significant slowdown in ATP search times, as more axioms inevitably create a larger search space. For this experiment, equivalent facts about dioids have been proven with and without explicit carrier sets, and then Kleene algebras have been formalised on top of both. As above, 18 problems provable directly from the axioms in both algebras within 300 seconds were selected. Table 3 shows the time it took for Sledgehammer to return a proof for each of the 18 problems, both for the explicit carrier set and non-explicit carrier set algebra. As with above, the sample size is again necessarily quite small—there are only so many facts we can automatically prove just from the axioms.

Analysing the results, it is clear that there is a statistically significant difference between using carrier sets and not using carrier sets, with the non-carrier

#	non-explicit	explicit	difference
1	0.19	0.45	-0.26
2	0.15	0.38	-0.23
3	2.30	21.86	-19.56
4	0.13	0.46	-0.33
5	0.11	0.78	-0.67
6	0.11	1.25	-1.15
7	0.12	0.77	-0.65
8	0.11	1.27	-1.16
9	0.60	1.38	-0.77
10	0.12	0.32	-0.20
11	76.47	31.50	44.96
12	1.16	4.38	-3.22
13	1.32	102.46	-101.14
14	1.12	1.14	-0.01
15	1.22	102.00	-100.78
16	0.12	0.24	-0.12
17	1.21	66.15	-64.94
18	1.22	63.60	-62.38

Table 3. Minimum prover times (s) for explicit and non-explicit carrier sets

set version being much more efficient. However, this is not a huge obstacle, as ATP is still very useful even with explicit carrier sets. Furthermore, carrier sets are absolutely essential for formalising many mathematical concepts.

The only problem out of the 18 which was more efficient with explicit carrier sets than without was that $1 + x + x^*x^* \leq x^*$. The reason for this was that it is quite a difficult problem for most of the provers, as only SPASS was able to prove it for the carrier set based case, while E was able to solve the problem without carrier sets. For each of the provers there are usually several cases where the explicit carrier set based algebra wins out over the non-carrier set based variant. This is most likely because even though the search space with carrier sets is larger, there is still the chance that a prover may still find a proof more quickly. By using four provers, this effect is diminished, but it can appear when only few provers are capable of tackling a problem.

Conclusion These results show that while the use of different first-order regular algebras has little effect on prover performance, adding explicit carrier sets does impact performance significantly. On the other hand, previous experiments in formalising concepts such as Galois connections indicates that carrier sets provide a significant boost in expressivity [1], which can make the trade-off worthwhile. Unfortunately there does not seem to be an easy way in Isabelle to connect explicit carrier set and non-explicit carrier set algebras and get the best of both worlds. These results could be made more conclusive with a larger set of examples, but this is difficult to achieve while ensuring fairness.

References

1. A. Armstrong and G. Struth. Automated reasoning in higher-order regular algebra (to appear). In T. G. Griffin and W. Kahl, editors, *RAMiCs 2012*, volume 7560 of *LNCS*. Springer, 2012.
2. J. C. Blanchette, L. Bulwahn, and T. Nipkow. Automatic proof and disproof in Isabelle/HOL. In C. Tinelli and V. Sofronie-Stokkermans, editors, *FroCos 2011*, LNCS, pages 12–27. Springer, 2011.
3. S. Böhme and T. Nipkow. Sledgehammer: Judgement day. In J. Giesl and R. Hähnle, editors, *Automated Reasoning (IJCAR 2010)*, volume 6173 of *LNCS*, pages 107–121. Springer, 2010.
4. J. H. Conway. *Regular Algebra and Finite Machines*. Chapman and Hall, 1971.
5. S. Foster, G. Struth, and T. Weber. Automated engineering of relational and algebraic methods in Isabelle/HOL – (invited tutorial). In H. de Swart, editor, *RAMiCS 2011*, volume 6663 of *LNCS*, pages 52–67. Springer, 2011.
6. F. Haftmann and M. Wenzel. Local theory specifications in Isabelle/Isar. In S. Berardi, F. Damiani, and U. de'Liguoro, editors, *TYPES 2008*, volume 5497 of *LNCS*, pages 153–168. Springer, 2008.
7. D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. *Information and Computation*, 110(2):366–390, 1994.
8. L. Paulson, T. Nipkow, and M. Wenzel. Isabelle. `http://www.cl.cam.ac.uk/research/hvg/Isabelle/index.html`, 2011.
9. V. R. Pratt. Action logic and pure induction. In J. van Eijck, editor, *JELIA '90*, volume 478 of *LNCS*, pages 97–120. Springer, 1990.