An Evaluation of Automated Theorem Proving in Regular Algebra (Student Paper)

Alasdair Armstrong

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

September 20, 2012

Overview

Overview

Experiment 1

- Many different first-order regular algebras
- Kleene algebras are commonly used, but action algebras permit a purely equational axiomatisation

Experiment 1

- Many different first-order regular algebras
- Kleene algebras are commonly used, but action algebras permit a purely equational axiomatisation

Which of action algebras or Kleene algebras is better from an ATP standpoint?

Experiment 1 - Background

- A dioid is an algebra $(D, +, \cdot, 0, 1)$ where $(D, +, 0)$ is a semilattice with least element 0, $(D, \cdot, 1)$ is a monoid, $\cdot$ distributes over $+$ from both the left and right, and $0 \cdot x = 0 = x \cdot 0$.
- We can prove many properties about dioids, which can be used in both Kleene algebra and action algebra
- A Kleene algebra is an algebra $(K, +, \cdot, 0, 1, ^*)$ where $(K, +, \cdot, 0, 1)$ is a dioid, satisfying the following 4 axioms:

$$\begin{aligned} 1 + xx^* \leq x^*, \qquad & 1 + x^*x \leq x^* \\ z + xy \leq y \Rightarrow x^*z \leq y, \qquad & z + yx \leq y \Rightarrow zx^* \leq y. \end{aligned}$$

Experiment 1 - Background

An action algebra is an algebra $(A, +, 0, \cdot, 1, \leftarrow, \rightarrow, ^*)$ such that $(A, +, \cdot, 0, 1)$ is a dioid, and satisfying

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

$$1 + x^* x^* + x \leq x^*, \qquad 1 + yy + x \leq y \Rightarrow x^* \leq y$$

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

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

Experiment 1 - Hypothesis

- One might expect that purely equational axioms would be more amenable to ATP

Experiment 1 - Hypothesis

- One might expect that purely equational axioms would be more amenable to ATP
- On the other hand, a larger set of axioms and a larger signature may slow down the prover

Experiment 1 - Hypothesis

- One might expect that purely equational axioms would be more amenable to ATP
- On the other hand, a larger set of axioms and a larger signature may slow down the prover
- Maybe there is no difference between the two algebras?

Experiment 2

- Two ways of formalising algebras in Isabelle
- With and without explicit carrier sets
- Carrier sets are necessary for real mathematics

Experiment 2

- Two ways of formalising algebras in Isabelle
- With and without explicit carrier sets
- Carrier sets are necessary for real mathematics

Exactly how much do carrier sets impact the usefulness of ATP and SMT tools?

Experiment 2 - Background

```
class kleene_algebra = dioid + star_op +
  fixes star :: "'a ⇒ 'a" ("_*" [101] 100)
  assumes star_unfoldl: "1 + xx* ≤ x*"
  and star_unfoldr: "1 + x*x ≤ x*"
  and star_inductl: "z+xy ≤ y ⟶ x*z ≤ y"
  and star_inductr: "z+yx ≤ y ⟶ zx* ≤ y"
```

Experiment 2 - Background

```
record 'a kleene_algebra = "'a dioid" +
  star :: "'a ⇒ 'a" ("_*ᵢ" [101] 100)

locale kleene_algebra = dioid K for K (structure) +
  assumes star_closed: "x ∈ carrier K ⟹ x* ∈ carrier K"
  and star_unfoldl: "x ∈ carrier K ⟹ 1 + xx* ≤ x*"
  and star_unfoldr: "x ∈ carrier K ⟹ 1 + x*x ≤ x*"
  and star_inductl: "⟦ x ∈ carrier K; y ∈ carrier K; z ∈ carrier K ⟧
                     ⟹ z+xy ≤ y ⟶ x*z ≤ y"
  and star_inductr: "⟦ x ∈ carrier K; y ∈ carrier K; z ∈ carrier K ⟧
                     ⟹ z+yx ≤ y ⟶ zx* ≤ y"
```

Experiment 2 - Hypothesis

- Explicit carrier sets make our axioms more expressive
- But also more complicated
- We can reasonably assume that we will pay a price of this increased expressivity in terms of ATP usefulness and performance

Overview

Method

- Isabelle's built in benchmarking tool Mirabelle is used to benchmark the ATP systems
- Our interest is in comparing the algebras, not various provers, so we stick to using the default set of provers sledgehammer uses
  - E
  - Z3 (Remote)
  - Vampire (Remote)
  - SPASS
- Each prover is still tested individually, so the results on still a per prover basis

Method

- To ensure fairness only properties that could be derived directly from the axioms within a 300 second period were considered.
- This approach has a downside – only a small amount of lemmas can be derived fully automatically from both the axioms of Kleene and action algebra
- For the first experiment, there are 20 available properties satisfying this criterion
- For the second, there are only 18

Method

- To ensure fairness only properties that could be derived directly from the axioms within a 300 second period were considered.
- This approach has a downside – only a small amount of lemmas can be derived fully automatically from both the axioms of Kleene and action algebra
- For the first experiment, there are 20 available properties satisfying this criterion
- For the second, there are only 18

Why is this restriction necessary?

Method

- In an ordinary Isabelle workflow, one starts by proving useful lemmas which are then used in later proofs.
- For example, I want to prove $x \leq y \Rightarrow x^* \leq y^*$
- To prove this easily, I might need some auxiliary lemmas
- However, depending on which axiom set I start with, the ideal set of lemmas for the shortest proof may be different
- In practice, the order in which things are proved is very important
- Selecting a specific order would invariably favour one algebra

Overview

Experiments and Hypothesis

Method

Results

Conclusion

	e			remote_z3		
#	KLE	ACT	diff	KLE	ACT	diff
1	110.61	110.2	-0.41	F	**F**	**F**
2	103.36	0.78	-102.58	**F**	1.53	**F**
3	1.58	**F**	**F**	1.44	**F**	**F**
4	1.06	116.19	115.13	1.42	**F**	**F**
5	1.11	116.52	115.42	1.41	**F**	**F**
6	100.66	111.01	10.35	**F**	**F**	**F**
7	36.45	100.33	63.93	1.49	**F**	**F**
8	0.84	0.94	0.1	1.42	1.45	0.03
9	0.80	1.91	1.12	1.39	1.4	0.01
10	105.05	0.99	-104.06	22.57	1.4	-21.18
11	139.5	100.98	-38.52	**F**	3.24	**F**
12	1.97	41.36	39.4	**F**	**F**	**F**
13	105.2	0.87	-104.29	25.2	1.39	-23.82
14	102.1	39.72	-62.38	3.27	**F**	**F**
15	100.89	0.91	-99.98	1.47	1.41	-0.06
16	114.02	100.83	-13.2	**F**	**F**	**F**
17	**F**	**F**	**F**	**F**	3.14	**F**
18	61.82	100.85	39.03	1.82	**F**	**F**
19	0.36	0.37	0.02	1.38	1.34	-0.04
20	0.82	0.84	0.03	1.6	**F**	**F**
			-140.92			-45.09

remote_vampire				spass		
#	KLE	ACT	diff	KLE	ACT	diff
1	7.25	10.15	2.90	193.06	1.39	-191.67
2	6.02	2.00	-4.02	100.47	0.10	-100.37
3	1.10	72.61	71.51	0.12	100.42	100.30
4	1.08	3.78	2.71	0.09	3.95	3.85
5	1.07	13.2	12.13	0.09	3.97	3.89
6	41.24	44.77	3.53	134.07	**F**	**F**
7	37.95	2.21	-35.74	100.41	100.25	-0.16
8	1.86	3.00	1.15	0.09	1.50	1.41
9	1.89	1.98	0.10	0.08	0.11	0.02
10	29.36	28.01	-1.35	104.88	0.89	-103.10
11	27.04	29.14	2.10	113.64	100.21	-13.43
12	1.50	3.20	1.60	135.05	**F**	**F**
13	3.90	1.95	-1.93	100.77	0.13	-100.64
14	1.39	7.66	6.27	114.47	153.76	39.29
15	**F**	2.01	**F**	0.49	0.23	-0.27
16	125.65	40.98	-84.67	**F**	51.97	**F**
17	**F**	31.29	**F**	**F**	10.37	**F**
18	29.31	2.23	-27.07	100.35	100.46	0.10
19	1.09	**F**	**F**	0.11	0.09	-0.01
20	1.13	1.09	-0.04	0.10	0.11	0.01
			-50.75			-361.66

Results - Action Algebra vs KA

- At first glance action algebras appear slightly faster

Results - Action Algebra vs KA

- At first glance action algebras appear slightly faster

#	KLE	ACT	difference
1	7.25	1.39	-5.86
2	6.02	0.10	-5.91
3	0.12	72.61	72.49
4	0.09	3.78	3.69
5	0.09	3.97	3.89
6	41.24	44.77	3.53
7	1.49	2.21	0.72
8	0.09	0.94	0.85
9	0.08	0.11	0.02
10	22.57	0.89	-21.69

#	KLE	ACT	diff
11	27.04	3.24	-23.80
12	1.50	3.20	1.70
13	3.88	0.13	-3.75
14	1.39	7.66	6.27
15	0.49	0.23	-0.27
16	114.02	40.98	-73.04
17	0.00	3.14	3.14
18	1.82	2.23	0.42
19	0.10	0.09	-0.01
20	0.10	0.11	0.01

- However, there is no statistically significant difference when we look at the results with all provers taken into account

Results - Action Algebra vs KA

- The null hypothesis was correct—there is no difference in ATP performance between action algebras and Kleene algebras
- How did the individual provers perform?
  - SPASS tends to perform either extremely well, or extremely poorly
  - It seems to be noticeably better at Kleene algebra (but the small sample size could make this misleading)
  - Z3 is always very fast when it can find a proof
  - E is often the fastest
  - Vampire is the most consistent
- The time it takes to find a proof in action algebra and Kleene algebra is somewhat correlated

Results - Action Algebra vs KA

- These results compare how quickly properties can be derived from the axioms
- However, we are also interested in what can be automatically derived from the axioms, and not just how fast
- For example in Kleene algebra one can automatically derive:
  - The sliding rule, $(xy)^*x \leq x(yx)^*$
  - $x^*x \leq xx^*$
  - $x^* \leq (x^*)^*$
  - $x \leq y \Longrightarrow x^* \leq y^*$
  - $xy \leq y \Rightarrow x^*y \leq y$ and $yx \leq y \Rightarrow yx^* \leq y$

Results - Action Algebra vs KA

- These results compare how quickly properties can be derived from the axioms
- However, we are also interested in what can be automatically derived from the axioms, and not just how fast
- For example in Kleene algebra one can automatically derive:
  - The sliding rule, $(xy)^*x \leq x(yx)^*$
  - $x^*x \leq xx^*$
  - $x^* \leq (x^*)^*$
  - $x \leq y \Longrightarrow x^* \leq y^*$
  - $xy \leq y \Rightarrow x^*y \leq y$ and $yx \leq y \Rightarrow yx^* \leq y$
- Kleene algebra is clearly superior here

Results - Explicit vs Non-Explicit Carrier Sets

#	NE	E	diff
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

#	NE	E	diff
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

Results - Explicit vs Non-Explicit Carrier Sets

- It is clear that there is a statistically significant difference between using explicit carrier sets and not using explicit carrier sets
- ATP systems work much better without carrier sets
- One one problem that was better with carrier sets was showing $1 + x + x^*x^* \leq x^*$
- This is probably because it was quite hard for the provers
  - SPASS proved it with carrier sets
  - E proved it without
- Using multiple provers minimises cases like this

Overview

Conclusion

- Choice of regular algebra axioms largely irrelevant from an ATP performance perspective
- Kleene algebra axioms seem more usable though
- Explicit carrier sets have a negative impact on ATP performance
- However, in some cases they are necessary