Systems for equational additivity*

Mike Behrisch[†‡§] Bernardo Rossi[‡¶] Erhard Aichinger[‡‖]

20th June 2023

Overview

This data set provides supplementary material supporting the verification of the solution sets of several systems of equations appearing in [1] that ensure a property of algebraic systems called *equational additivity*, cf. [4]. By [2, Theorem 2.5], for an algebraic system $\mathbf{A} = (A; (f)_{f\in\mathcal{F}})$ on a finite set A, equational additivity is characterised by the fact that the relation

$$\Delta_A^{(4)} := \left\{ (x, y, u, v) \in A^4 \mid x = y \lor u = v \right\}$$

is the set of solutions of a finite system of quaternary term equations over $\mathbf{A}$; in other words, $\mathbf{A}$ is equationally additive if there is $n \in \mathbb{N} \setminus \{0\}$ and there are quaternary terms $s_1, \dots, s_n, t_1, \dots, t_n$ over the language of $\mathbf{A}$ such that

$$\Delta_A^{(4)} = \left\{ \boldsymbol{x} \in A^4 \mid s_1(\boldsymbol{x}) = t_1(\boldsymbol{x}) \land \cdots \land s_n(\boldsymbol{x}) = t_n(\boldsymbol{x}) \right\}.$$

The article [1] relies on the (non-)expressibility of $\Delta_A^{(4)}$ via certain systems of equations in the subsequently listed sections (for the meaning of the respective function symbols we refer the esteemed reader to [1], the systems of equations are merely listed here for better recognisability in [1]):

Section 6 Remark 6.7

$$h(x_3, x_4, x_1) = h(x_3, x_4, x_2) \tag{1}$$

and

$$\begin{aligned} \tau(x_3, x_4, x_1) &= \tau(x_3, x_4, x_2),\\ \tau(x_4, x_3, x_1) &= \tau(x_4, x_3, x_2); \end{aligned} \tag{2}$$

moreover

$$\forall f, g \in \mathsf{S}_{00}^{[4]}\colon \qquad \Delta_{\{0,1\}}^{(4)} \neq \left\{ \boldsymbol{x} \in \{0,1\}^4 \;\middle|\; f(\boldsymbol{x}) = g(\boldsymbol{x}) \right\}. \tag{3}$$

*Support by the Austrian Science Fund (FWF), grant P33878 is gratefully acknowledged.
†Institute of Discrete Mathematics and Geometry, TU Wien, Wien, Austria
‡Institut für Algebra, Johannes Kepler Universität Linz, Austria
§https://orcid.org/0000-0003-0050-8085
¶https://orcid.org/0000-0002-0404-6634
‖https://orcid.org/0000-0001-8998-4138

Section 7 Lemma 7.1,

$$\begin{aligned} f(x_1,x_2,x_3) &\approx f(x_1,x_2,x_4),\\ f(x_2,x_1,x_3) &\approx f(x_2,x_1,x_4),\\ f(x_3,x_4,x_1) &\approx f(x_3,x_4,x_2),\\ f(x_4,x_3,x_1) &\approx f(x_4,x_3,x_2). \end{aligned} \tag{4}$$

Theorem 7.6

$$\begin{aligned} m(x_1,x_2,x_3) &\approx m(x_1,x_2,x_4),\\ m(x_2,x_1,x_3) &\approx m(x_2,x_1,x_4). \end{aligned} \tag{5}$$

Section 8 Proposition 8.2

$$\begin{aligned} f(x_1,x_2,x_3) &\approx f(x_1,x_2,x_4),\\ f(x_2,x_1,x_3) &\approx f(x_2,x_1,x_4),\\ f(x_3,x_4,x_1) &\approx f(x_3,x_4,x_2),\\ f(x_4,x_3,x_1) &\approx f(x_4,x_3,x_2). \end{aligned} \tag{6}$$

We observe that the systems (4) and (6) are syntactically identical, they only appear in different contexts where the function symbol f is instantiated with two different concrete functions, cf. [1].

The solution sets of the systems (1)-(6) are mainly computed and verified in the script `checking_systems_of_equations.py`; note, however that the code for the verification of (3) relies on results arising from the script `write_uacalc_files.py`.

List of files

We now give a brief overview of the files in the data set, before focusing on the content of `checking_systems_of_equations.py` in a little more detail.

`checking_systems_of_equations.py`	python script verifying the definability of $\Delta_A^{(4)}$ from the systems (1)–(6); outputs `check_of_systems_of_equations.txt`; the code partially depends on `F4_over_S00.txt`.
`check_of_systems_of_equations.txt`	output log of running the script `checking_systems_of_equations.py`.
`write_uacalc_files.py`	python script writing source files (`*.alg`) for the universal algebra calculator [3]; outputs `output_write_uacalc_files.txt`; results in the files `aP.alg`, `AP.alg`, `L2.alg`, `S00.alg`, `S10.alg`, `SL.alg`. The script mainly defines functions for the generators of these clones according to the literature (referenced in the code) and then outputs the value tables of these functions in a format[1] suitable to be read by [3].
`output_write_uacalc_files.txt`	output log of running the script `write_uacalc_files.py`

`aP.alg`, `AP.alg`, `L2.alg`, `SL.alg`, `S00.alg`, `S10.alg`	input files[1] for the universal algebra calculator [3], representing algebras the clone of term operations of which equals $\mathbf{aP}, \mathbf{AP}, \mathbf{L}_2, \mathbf{SL}$ appearing in [1, Section 7] and S_{00} and S_{10} from [1, Section 6], respectively; these have been written by running `write_uacalc_files.py`.
`F4_over_S00.csv`, `F4_over_S10.csv`	result of computing in UACalc [3] the free algebra on four generators (using all coordinates, no decomposition, no thinning) over the algebra represented by `S00.alg` and `S10.alg`, respectively.
`F4_over_S00.txt`, `F4_over_S10.txt`	the same file as `F4_over_S00.csv` and `F4_over_S10.csv`, respectively, but after slight manual editing to enhance readability. Columns 3 to 18 of the table in `F4_over_S00.txt` are used as part of the code of the function `check_implication_re_F4_S00_in_67()` in `checking_systems_of_equations.py`.
`systems4_eqn_add.pdf`	this documentation
`systems4_eqn_add.tex`	LaTeX source file to produce this documentation

Details regarding the script `checking_systems_of_equations.py`

The initial functions in `checking_systems_of_equations.py` implement generator functions as explained in [1] or as presented in the literature referenced in the comments of `checking_systems_of_equations.py`. This is rather self-explanatory given the comments. We briefly explain the remaining functions:

`get_Deltarelation(k)`	computes a sorted list containing the quadruples in $\Delta^{(4)}_{\{0,\ldots,k-1\}}$ where $k \in \mathbb{N}$.
`get_solution_set_71_82(f)`	computes a sorted list containing the solutions of the systems (4) (and (6), respectively); the parameter is the ternary operation that instantiates the symbol f in the systems (4) and (6).
`get_solution_set_76(f)`	computes a sorted list containing the solutions of the system (5); the parameter is the ternary operation that instantiates the symbol f in the system.
`get_solution_set_67h(f)`	computes a sorted list containing the solutions of the system (1) over $\{0, 1\}$; the parameter is the ternary Boolean function that instantiates the symbol h in the system.

[1] The format of `.alg`-files is briefly explained in the final section of this document.

`get_solution_set_67t(f)`

computes a sorted list containing the solutions of the system (2) over $\{0,1\}$; the parameter is the ternary Boolean function that instantiates the symbol τ in the system.

`check_solution_set_equals_delta(sol,k)`

checks whether the solution set given as an ordered list of quadruples over $\{0,\dots,k-1\}$ via the parameter `sol` is identical to $\Delta^{(4)}_{\{0,\dots,k-1\}}$.

`aux_fg_agree_on_delta(f,g)`

an auxiliary function checking if two quaternary Boolean functions $f,g\colon \{0,1\}^4 \to \{0,1\}$, represented by their list of 16 values on $\{(0,0,0,0),(0,0,0,1),(0,0,1,0),(0,0,1,1),\dots,(1,1,1,1)\}$ satisfy $f\restriction_{\Delta^{(4)}_{\{0,1\}}} = g\restriction_{\Delta^{(4)}_{\{0,1\}}}$. This function is used in `check_implication_re_F4_S00_in_67()`.

`aux_fg_agree_somewhere_outside_delta(f,g)`

an auxiliary function checking if two quaternary Boolean functions $f,g\colon \{0,1\}^4 \to \{0,1\}$, represented by their list of 16 values on $\{(0,0,0,0),(0,0,0,1),(0,0,1,0),(0,0,1,1),\dots,(1,1,1,1)\}$ satisfy

$$
\begin{aligned}
&f(0,1,0,1) = g(0,1,0,1) \vee\\
&f(0,1,1,0) = g(0,1,1,0) \vee\\
&f(1,0,0,1) = g(1,0,0,1) \vee\\
&f(1,0,1,0) = g(1,0,1,0).
\end{aligned}
$$

This function is used in `check_implication_re_F4_S00_in_67()`.

`check_implication_re_F4_S00_in_67()`

a function checking the truth of the universally quantified inequality (3) by verifying the following universal implication

$$
\forall f,g \in \mathsf{S}^{[4]}_{00}\colon \quad \Delta^{(4)}_{\{0,1\}} \subseteq \left\{ \boldsymbol{x} \in \{0,1\}^4 \;\middle|\; f(\boldsymbol{x}) = g(\boldsymbol{x}) \right\}
\Rightarrow \Delta^{(4)}_{\{0,1\}} \subsetneq \left\{ \boldsymbol{x} \in \{0,1\}^4 \;\middle|\; f(\boldsymbol{x}) = g(\boldsymbol{x}) \right\}.
$$

For this check the algorithm iterates over all pairs $f,g \in \mathsf{S}^{[4]}_{00}$, finds those where the inclusion $\Delta^{(4)}_{\{0,1\}} \subseteq \left\{ \boldsymbol{x} \in \{0,1\}^4 \;\middle|\; f(\boldsymbol{x}) = g(\boldsymbol{x}) \right\}$ holds (via `aux_fg_agree_on_delta(f,g)`), and then checks (via `aux_fg_agree_somewhere_outside_delta(f,g)`) that each time this happens the proper inclusion $\Delta^{(4)}_{\{0,1\}} \subsetneq \left\{ \boldsymbol{x} \in \{0,1\}^4 \;\middle|\; f(\boldsymbol{x}) = g(\boldsymbol{x}) \right\}$ holds, as well. When this is true, we either have proper inclusion, or no inclusion at all, hence never equality, as claimed in (3).

We remark that the iteration over all $f,g \in \mathsf{S}^{[4]}_{00}$ is achieved by iterating over the rows in the hard-coded list `F4`, which contains the functions in the free four-generated algebra as taken from `F4_over_S00.txt`.

remaining code	uses the previously discussed functions to answer the questions that can be read from the log file `check_of_systems_of_equations.txt`.

Format description of `.alg`-files

This section contains a short explanation of the format used in `.alg`-input files for the universal algebra calculator [3]. A finite algebra **A** with $n \geq 1$ elements is represented in UACalc by the standard carrier set $n = \{0, 1, \ldots, n-1\}$. An `.alg`-file contains, line by line, the following integer numbers.

- The first line contains the value n, the cardinality of the algebra.
- For each fundamental operation f of **A**, say with $k \geq 0$ arguments, the following lines are included:
  - a line with the value k
  - n^k lines with the values of f in the following order:

$$
\begin{array}{l}
f(0,0,0,\ldots,0)\\
f(1,0,0,\ldots,0)\\
\vdots\\
f(n-1,0,0,\ldots,0)\\
f(0,1,0,\ldots,0)\\
f(1,1,0,\ldots,0)\\
\vdots\\
f(n-1,1,0,\ldots,0)\\
f(0,2,0,\ldots,0)\\
\vdots\\
f(0,n-1,0,\ldots,0)\\
\vdots\\
f(n-1,n-1,0\ldots,0)\\
f(0,0,1,\ldots,0)\\
\quad\ddots\\
f(n-1,n-1,\ldots,n-1),
\end{array}
$$

    that is to say, the first argument changes fastest, the last argument changes slowest.

Example. *The left-zero semigroup on* $\{0, 1, 2\}$*, that is,* $(\{0, 1, 2\}; f)$ *where* f *is the binary operation* $f\colon \{0, 1, 2\}^2 \to \{0, 1, 2\}$ *given as* $f(x, y) := x$ *for each*

$x, y \in \{0,1,2\}$, having the operation table

$x \backslash y$	0	1	2
0	0	0	0
1	1	1	1
2	2	2	2 ,

would be stored in `.alg`-format as follows:

```
3
2
0
1
2
0
1
2
0
1
2
```

References

[1] Erhard Aichinger, Mike Behrisch, and Bernardo Rossi. On when the union of two algebraic sets is algebraic. `arXiv:2309.00478 [math.RA]`:1–50, September 2023. doi: `https://doi.org/10.48550/arXiv.2309.00478`.

[2] Èvelina Yur'evna Daniyarova, Alexei Georgievich Myasnikov, and Vladimir Nikanorovich Remeslennikov. Algebraic geometry over algebraic structures. IV. Equational domains and codomains. *Algebra i Logika*, 49(6):715–756, 844, 847, 2010. Translation in *Algebra and Logic*, 49(6):483–508, 2011. doi: `https://doi.org/10.1007/s10469-011-9112-2`.

[3] Ralph Freese, Emil Kiss, and Matthew Valeriote Universal Algebra Calculator, 2022. Available on-line from `http://www.uacalc.org`.

[4] Aleksandr Georgievich Pinus. Algebraically equivalent clones. *Algebra i Logika*, 55(6):760–768, 2016. Translation in *Algebra and Logic*, 55(6):501–506, 2017. doi: `https://doi.org/10.1007/s10469-017-9420-2`.