Circuits and Expressions with Non-Associative Gates

Joshua Berman[1], Arthur Drisko[2], Francois Lemieux[3], Cristopher Moore[4], and Denis Thérien[3]

[1] State University of New York at Binghamton
[2] National Security Agency, USA
[3] McGill University, Montréal, Québec
[4] The Santa Fe Institute, Santa Fe, New Mexico (corresponding author)

Abstract. We consider circuits and expressions whose gates carry out multiplication in a non-associative algebra such as a quasigroup or loop. We define a class we call the *polyabelian* algebras, formed by iterated quasidirect products of Abelian groups. We show that a quasigroup can express arbitrary Boolean functions if and only if it is not polyabelian, in which case its Expression Evaluation and Circuit Value problems are $\mathbf{NC}^1$-complete and $\mathbf{P}$-complete respectively. This is not true for algebras in general, and we give a counter-example.
We show that Expression Evaluation is also $\mathbf{NC}^1$-complete if the algebra has a non-solvable multiplication group or semigroup, but is in $\mathbf{TC}^0$ if the algebra is both polyabelian and has a solvable multiplication semigroup, e.g. for a nilpotent loop or group.
Thus, in the non-associative case, earlier results about the role of solvability in circuit complexity generalize in several different ways.

1 Introduction: algebraic circuits and expressions

Boolean expressions and circuits are well-known constructs in logic and computer science. A Boolean expression ϕ is either a variable x_i, or is formed from shorter expressions as $(\phi_1 \wedge \phi_2)$, $(\phi_1 \vee \phi_2)$ or $\neg\phi_1$. If all the x_i have truth values TRUE or FALSE, then ϕ's truth value is determined by interpreting $\wedge$, $\vee$ and $\neg$ as AND, OR and NOT respectively.

A Boolean circuit is an acyclic directed graph with source nodes (inputs) x_i and a single sink node (output), and three kinds of intermediate nodes: AND and OR gates with two inputs, and NOT gates with one input. (Expressions are simply circuits whose graph is a tree.) Then the truth value of the output is defined in the obvious way.

Given an expression or circuit, and the truth values of its variables or inputs, determining the truth value of its output is called the Expression Evaluation or Circuit Value problem respectively. These problems are deeply related to two important complexity classes, $\mathbf{NC}^1$ and $\mathbf{P}$.

$\mathbf{NC}^1$ is the set of problems solvable by parallel circuits of polynomial size and logarithmic depth as a function of length of the input. $\mathbf{P}$ is the set of problems solvable in polynomial time by a deterministic serial computer such as a Turing machine.

More generally, circuits of polynomial size and polylogarithmic depth ($\mathcal{O}(\log^k n)$ for inputs of size n) recognize the following complexity classes [9, 19], where the *fan-in* is the number of inputs to each gate:

- $\mathbf{NC}^k$ if the gates are AND's and OR's with fan-in 2
- $\mathbf{AC}^k$ if the gates are AND's and OR's with unbounded fan-in
- $\mathbf{TC}^k$ if the gates are *threshold gates* with unbounded fan-in, $\theta_t = \text{TRUE}$ if t or more of their inputs are TRUE
- $\mathbf{ACC}^k[p]$ if the gates are AND's, OR's, and "sum mod p" with unbounded fan-in
- $\mathbf{ACC}^k = \cup_p \mathbf{ACC}^k[p]$.

The union of any of these classes over all k is $\mathbf{NC}$, the class of problems solvable by parallel circuits of polylogarithmic depth (which is also polylogarithmic time on an idealized parallel computer).

Since combinations of threshold gates can compute any function that depends only on the sum of the inputs, including sum mod k, we have

$$\mathbf{NC}^k \subseteq \mathbf{AC}^k \subseteq \mathbf{ACC}^k \subseteq \mathbf{TC}^k \subseteq \mathbf{NC}^{k+1}$$

for all k. The classes we'll be interested in are

$$\mathbf{AC}^0 \subset \mathbf{ACC}^0[2] \subset \mathbf{ACC}^0 \subseteq \mathbf{TC}^0 \subseteq \mathbf{NC}^1 \subseteq \mathbf{ACC}^1 \subseteq \cdots \subseteq \mathbf{NC} \subseteq \mathbf{P}$$

It is believed, but not known, that $\mathbf{P} \neq \mathbf{NC}$: in other words, that there are *inherently sequential* problems in $\mathbf{P}$ that cannot be efficiently parallelized. Some small progress has been made towards proving this [1, 8, 21, 23]: parity is in $\mathbf{ACC}^0[2]$ but not $\mathbf{AC}^0$, $\mathbf{ACC}^0[p]$ and $\mathbf{ACC}^0[q]$ are incomparable if p and q are distinct primes, and majority is in $\mathbf{TC}^0$ but not $\mathbf{ACC}^0[2]$. Thus the first two inclusions in this series are proper, but $\mathbf{ACC}^0[6]$ and $\mathbf{P}$ (or even $\mathbf{NP}$) could be identical for all anyone has been able to prove.

A *reduction* from a problem A to a problem B is a mapping of instances of A to instances of B. If the mapping is computationally easy compared to B, then any fast algorithm for B becomes a fast algorithm for A; thus B is at least as hard as A. A problem B is *complete* for a complexity class if every other problem in that class can be reduced to it.

For $\mathbf{P}$ it is common to use $\mathbf{LOGSPACE}$ ($\mathcal{O}(\log n)$ memory in a Turing machine) or $\mathbf{NC}^1$ reductions; for $\mathbf{NC}$ we will use $\mathbf{AC}^0$ or $\mathbf{NC}^0$ reductions, the latter being essentially local replacement rules.

We also have

$$\mathbf{NC}^1 \subseteq \mathbf{DET} \subseteq \mathbf{NC}^2$$

where $\mathbf{DET}$ is the class of problems $\mathbf{NC}^1$-reducible to calculating the determinant of an integer matrix. $\mathbf{DET}$ is not known to be comparable with $\mathbf{AC}^1$ or $\mathbf{ACC}^1$.

Then we have the classical results that, for Boolean gates, EXPRESSION EVALUATION and CIRCUIT VALUE are $\mathbf{NC}^1$-complete and $\mathbf{P}$-complete respectively, under $\mathbf{AC}^0$ and $\mathbf{NC}^1$ reductions [6, 12].

We will consider circuits and expressions where the sole operation is multiplication in some finite algebra $(A, \cdot)$, rather than the usual Boolean operations. Thus our expressions are polynomials like $(x_1 \cdot x_2) \cdot (x_2 \cdot x_3)$, and our circuits have one kind of node whose output is the product $a \cdot b$ of its two inputs.

Then depending on the algebraic properties of $(A, \cdot)$, such as associativity, commutativity, solvability and so on, EXPRESSION EVALUATION and CIRCUIT VALUE can have varying complexities. Previous results for the associative case (groups and semigroups) include the following [2, 4, 17]:

	EXPRESSION EVALUATION	CIRCUIT VALUE
non-solvable	$\mathbf{NC}^1$-complete	$\mathbf{P}$-complete
solvable	$\mathbf{ACC}^0$	$\mathbf{ACC}^1 \cap \mathbf{DET}$

In this paper, we will extend these results to non-associative algebras such as quasigroups and loops, and to some extent to algebras in general. We will show that the idea of solvability generalizes in two important ways in the non-associative case: *polyabelianness*, the property of being an iterated quasidirect product of Abelian groups; and $\mathcal{M}$*-solvability*, the property of having a solvable multiplication group or semigroup.

We hope that these results are interesting in their own right, and that they might help illuminate the internal structure of $\mathbf{P}$ and $\mathbf{NC}$ and the relationship between different circuit models.

In addition, the problem of predicting a cellular automaton for a polynomial number of steps corresponds to a special case of CIRCUIT VALUE where the circuit has a periodic structure. Thus these results will also help us tell when there are fast algorithms for predicting cellular automata whose rules correspond to certain algebras, as in [17, 18].

2 Algebraic preliminaries

We will use the following terms. For the theory of quasigroups and loops, we recommend [20] as an introduction.

An *algebra* (also called a *groupoid* or *magma*) $(G, \cdot)$ is a binary operation $f : G \times G \to G$, written $f(a, b) = a \cdot b$ or simply ab. The *order* of an algebra is the number of elements in G, written $|G|$. Throughout the paper, we will assume that our algebras are finite.

A *quasigroup* is an algebra whose multiplication table is a *Latin square*, in which each symbol occurs once in each row and each column. Equivalently, for every a, b there are unique elements a/b and $a \backslash b$ such that $(a/b) \cdot b = a$ and $a \cdot (a \backslash b) = b$; thus the left (right) *cancellation property* holds, that $bc = bd$ (resp. $cd = bd$) implies $c = d$.

An *identity* is an element 1 such that $1 \cdot a = a \cdot 1 = a$ for all a. A *loop* is a quasigroup with an identity. In an Abelian group we will call the identity 0 instead of 1.

In a loop, the *left (right) inverse* of an element a is $a^\lambda = 1/a$ (resp. $a^\rho = a\backslash 1$) so that $a^\lambda \cdot a = 1$ (resp. $a \cdot a^\rho = 1$). If these are the same, we will refer to them both as a^{-1}.

An algebra is *associative* if $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ for all a, b, c. A *semigroup* is an associative algebra, and a *monoid* is a semigroup with identity. A *group* is an associative quasigroup; groups have inverses and an identity.

Two elements a, b *commute* if $a \cdot b = b \cdot a$. An algebra is *commutative* if all pairs of elements commute. Commutative groups are also called *Abelian*. We will use $+$ instead of $\cdot$ for products in an Abelian group.

In a group, the *order* of an element a is the smallest $p > 0$ such that $a^p = 1$ (or $pn = 0$ in an Abelian group).

A *homomorphism* is a function ϕ from one algebra $(A, \cdot)$ to another $(B, \star)$ such that $\phi(a \cdot b) = \phi(a) \star \phi(b)$. An *isomorphism* is a one-to-one and onto homomorphism; we will write $A \cong B$ if A and B are isomorphic. Homomorphisms and isomorphisms from an algebra into itself are called *endomorphisms* and *automorphisms* respectively; the automorphisms of an algebra A form a group $\mathrm{Aut}(A)$. Endomorphisms of Abelian groups can be represented by matrices.

A *subalgebra* (*subquasigroup, subloop*, etc.) of G is a subset $H \subseteq G$ such that $b_1 \cdot b_2 \in H$ for all $b_1, b_2 \in H$. The subalgebra generated by a set S, consisting of all possible products of elements in S, is written $\langle S \rangle$.

The *left (right) cosets* of a subloop $H \subseteq G$ are the sets $aH = \{ah \mid h \in H\}$ and $Ha = \{ha \mid h \in H\}$ for each $a \in G$. A subloop H is *normal* if the following hold for all $a, b \in G$:

$$aH = Ha, \quad a(bH) = (ab)H, \quad \text{and} \quad (aH)b = a(Hb)$$

Then the set of cosets of H is the *quotient loop* or *factor* G/H; it has identity $1H = H$, and is the image of G under the homomorphism $\phi(a) = aH$. Conversely, any homomorphic image $\phi(G)$ of a loop is a quotient $G/(\ker \phi)$ where the *kernel* $\ker \phi = \{g \in G \mid \phi(g) = 1\}$ is a normal subloop of G.

A subloop of G is *proper* if it is neither $\{1\}$ nor all of G. A *minimal* normal subloop of G is one which does not properly contain any proper normal subloops of G. A *simple* loop is one with no proper normal subloops.

The *commutator* of two elements in a loop is $[a, b] = ab \,/\, ba$, and the *associator* of three elements is $[a, b, c] = (ab)c \,/\, a(bc)$. The subloop generated by all possible commutators and associators in a loop G is called the *commutator-associator subloop* or *derived subloop* G'. It is normal, and it is the smallest subloop such that the quotient G/G' is an Abelian group.

A loop G is *solvable* if its *derived series* $G = G_0 \supset G_1 \supset \cdots$, where $G_{i+1} = G_i'$ for all i, ends in $G_k = \{1\}$ after a finite number of steps. More generally, we call an algebra solvable if it contains no non-solvable loops.

A *divisor* of an algebra is a factor of a subalgebra. Any non-solvable loop has a simple non-Abelian divisor.

The *center* of a loop is the set of elements that associate and commute with everything, $Z(G) = \{c \mid cx = xc,\, c(xy) = (xc)y = x(yc) \text{ for all } x, y \in G\}$. It is a normal subloop of G.

The *upper central series* of a loop is $\{1\} = Z_0 \subset Z_1 \subset \cdots$ where Z_{i+1}/Z_i is the center of G/Z_i. A loop is *nilpotent* (of class k) if $Z_k = G$ for some k. Inductively, G is nilpotent if it has a nontrivial center $Z(G)$, and $G/Z(G)$ has a non-trivial center, and so on until we get an Abelian group for which $Z(G) = G$. The nilpotent loops are a proper subclass of the solvable ones.

A *variety* is a class of algebras V such that subgroups, factors, and direct products of algebras in V are also in V. A *pseudovariety* is closed under finite direct products, but not necessarily infinite ones. Solvable and nilpotent loops both form pseudovarieties.

In a quasigroup Q, we can define left and right multiplication as functions $L_a(b) = a \cdot b$ and $R_a(b) = b \cdot a$. These are permutations on Q (the rows and columns of the multiplication table), and the *multiplication group* $\mathcal{M}(Q)$ is the group of permutations they generate. More generally, any algebra has a *multiplication semigroup* generated by the L_a and R_a, which are not necessarily one-to-one functions on Q. If we have more than one operation we will refer to $L_a^{\odot}$, $\mathcal{M}(Q, \odot)$, and so on.

Finally, we refer to the identity function $\mathbf{1}(x) = x$, the *cyclic group* $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$ with addition mod p, and the groups S_n and A_n of permutations and even permutations respectively on n elements.

3 Solvability and Boolean-completeness in groups and loops

Let us define the set of functions that can be expressed as circuits whose gates carry out multiplication in an algebra A, and whose inputs can be variables or constant elements of A. This is equivalent to the set of polynomial expressions definable in A with constants and variables, such as $\phi(x_1, x_2, x_3) = (x_1(ax_2))(x_3 b)^2$:

Definition. The *polynomial closure* of an algebra A is the smallest set $\mathcal{P}(A)$ of functions ϕ on an arbitrary number of variables $x_1, \ldots, x_k$ containing the following:

- (constants) a for all $a \in A$.
- (projections) x_i for all i.
- (products) $\phi_1 \cdot \phi_2$ for all $\phi_1, \phi_2 \in \mathcal{P}(A)$.

Since $\mathcal{P}(A)$ is closed under composition and substitution of one function for a variable of another, it is a *clone* [25]. We will sometimes refer to the set of functions on k variables as $\mathcal{P}^k(A)$: for instance, $\mathcal{P}^1(A)$ contains the multiplication semigroup $\mathcal{M}(A)$, as well as functions like $\phi(x) = x^2$.

We are interested in whether an algebra can express arbitrary Boolean functions. For instance, in the quasigroup

$\star$	1	2	3	4
1	1	3	2	4
2	3	2	4	1
3	2	4	1	3
4	4	1	3	2

if FALSE $= 1$ and TRUE $= 2$, then

$$a \wedge b = (a \star b)^2 \quad \text{and} \quad \neg a = 3 \star (1 \star a)$$

are expressions of these Boolean functions as polynomials in $\star$. We can combine these to make any other Boolean function. Formally:

Definition. An algebra $(A, \star)$ is *strongly Boolean-complete* if there exist elements TRUE and FALSE in A, and functions $\phi_\wedge, \phi_\neg$ in $\mathcal{P}(A)$, such that $\phi_\wedge(a, b) = a \wedge b$ and $\phi_\neg(a) = \neg a$ whenever $a, b \in \{\text{TRUE}, \text{FALSE}\}$.

In general, we will allow TRUE and FALSE to be sets, rather than single elements:

Definition. An algebra $(A, \star)$ is *Boolean-complete* if there exist disjoint subsets $T, F \subset A$ of "true" and "false" elements respectively, and functions $\phi_\wedge, \phi_\neg$ in $\mathcal{P}(A)$ such that $\phi_\wedge(a, b) \sim a \wedge b$ and $\phi_\neg(a) \sim \neg a$ whenever $a, b \in T \cup F$, where $x \sim y$ if x and y are both true or both false.

Then a strongly Boolean-complete algebra is a Boolean-complete one where T and F are the singletons $\{\text{TRUE}\}$ and $\{\text{FALSE}\}$.

We will often use the following lemma:

Lemma 1. *If a divisor of an algebra G is Boolean-complete, then G is also.*

Proof. If a subalgebra $H \subset G$ is (strongly) Boolean-complete, then G is also since $\mathcal{P}(H) \subset \mathcal{P}(G)$. If a factor $\phi(G)$ is Boolean-complete with T and F, simply let T' and F' in G be the inverse images $\phi^{-1}(T)$ and $\phi^{-1}(F)$. Since divisors are factors of subalgebras, the lemma is proved. □

Corollary. *The set of non-Boolean-complete algebras forms a pseudovariety.*

Proof. It is easy to show that finite direct products of non-Boolean-complete algebras are non-Boolean-complete; lemma 1 completes the proof, since it shows that (contrapositively) factors and subalgebras of non-Boolean-complete algebras are non-Boolean-complete. □

Then our fundamental motivation for this work is the following:

Lemma 2. *If an algebra is Boolean-complete, then its* Expression Evaluation *and* Circuit Value *problems are* $\mathbf{NC}^1$*-complete and* $\mathbf{P}$*-complete (under* $\mathbf{AC}^0$ *and* $\mathbf{NC}^1$ *reductions) respectively.*

Proof. Boolean Expression Evaluation and Circuit Value problems are reducible to their algebraic counterparts, since a local rule can replace AND, OR and NOT gates with complexes of algebraic gates. $\square$

Then we briefly re-state a theorem of Barrington [2], with a slightly different proof. The construction hinges on the fact that the commutator has the character of an AND gate: if FALSE $= 1$, then $[a, b]$ is FALSE if either a or b is, since the identity commutes with everything.

First we show two useful lemmas:

Lemma 3. *If* Q *is a finite quasigroup, the divisions* a/b *and* $a\backslash b$ *are in* $\mathcal{P}^2(Q)$ *as functions of* a *and* b*. Therefore, functions that yield the commutator, associator, and left and right inverses are in* $\mathcal{P}$ *as well.*

Proof. Recall the definition of L_a and R_a above. If Q is a quasigroup of order n, L_a and R_a are permutations on its elements, and $L_a^{n!}$ and $R_a^{n!}$ are the identity $\mathbf{1}$. Then $aL_a^{n!-1}(b) = L_a^{n!}(b) = b$, so

$$a\backslash b = L_a^{n!-1}(b) = \underbrace{a(a(\cdots(a}_{n!-1 \text{ times}} \cdot b)))$$

is in $\mathcal{P}(Q)$. Similarly for a/b; then by composition we can define $[a, b] = ab \,/\, ba$, $[a, b, c] = (ab)c \,/\, a(bc)$, $a^\lambda = 1/a$ and $a^\rho = a\backslash 1$. $\square$

Lemma 4. *If* G *is a simple loop or group, for any* $x, y \neq 1$ *there is a (not necessarily unique) function* $\pi_{x\to y}$ *in* $\mathcal{P}^1(G)$ *that sends* x *to* y *and keeps the identity fixed.*

Proof. Let U be the set of functions ϕ in $\mathcal{P}^1(G)$ such that $\phi(1) = 1$. For an element $x \in G$, let $U(x) = \{\phi(x) \mid \phi \in U\}$ be the set of y's that we can send x to, while fixing 1.

Then $U(x)$ is a subloop, since $\phi'(1) = 1$ if we define $\phi'(a) = \phi_1(a) \cdot \phi_2(a)$. Moreover, $U(x)$ is a normal subloop, since we can fulfill each requirement for normality with a suitable definition of ϕ':

$$\begin{aligned} a\,U(x) = U(x)\,a &: \ \phi'(x) = a\,\phi(x) \,/\, a \\ a(b\,U(x)) = (ab)\,U(x) &: \ \phi'(x) = (ab) \setminus a(b\,\phi(x)) \\ a(U(x)\,b) = (a\,U(x))b &: \ \phi'(x) = a \setminus \big(a(\phi(x)\,b) \,/\, b\big) \end{aligned}$$

In each case ϕ' is in $\mathcal{P}(G)$ by lemma 3, and in U since $\phi'(1) = 1$ if and only if $\phi(1) = 1$.

If G is simple, then it has no proper normal subloops and $U(x) = G$ for any $x \neq 1$. Thus for any y, there exists $\phi \in U$ such that $\phi(x) = y$; this is our $\pi_{x\to y}$. $\square$

Theorem 5 (Barrington). *Simple non-Abelian groups are strongly Boolean-complete, and non-solvable groups are Boolean-complete.*

Proof. Let G be simple and non-Abelian, so that $G = G'$. Then choose any non-commuting pair of elements x, y with $[x, y] \neq 1$. Let FALSE $= 1$ and TRUE be any non-identity element $t \neq 1$, and define

$$a \wedge b = \pi_{[x,y] \to t}\big([\pi_{t \to x}(a), \pi_{t \to y}(b)] \big)$$

This expression evaluates to t if $a = b = t$, and 1 if either a or b is 1; in other words, it is an AND gate. Finally, we can express negation as $\neg a = t \cdot a^{-1}$.

So simple non-Abelian groups are strongly Boolean-complete; and since every non-solvable group has a simple non-Abelian divisor, non-solvable groups are Boolean-complete by lemma 1. □

(In fact, [2] seems to state that non-solvable groups are strongly Boolean-complete; we believe this is a slight mistake.)

By using an associator instead of a commutator, this generalizes to loops [13]. Like the commutator, the associator $[x, y, z]$ is 1 if any of its arguments is 1, since the identity associates with everything.

Theorem 6. *Simple non-Abelian loops are strongly Boolean-complete, and non-solvable loops are Boolean-complete.*

Proof. Assume without loss of generality that G is simple and non-associative, since theorem 5 treats the associative case. Choose a triplet of non-associating elements $x, y, z \in G$ with $[x, y, z] \neq 1$. Let FALSE $= 1$ and TRUE $= t \neq 1$ as before. Then

$$a \wedge b = \pi_{[x,y,z] \to t}\big([\pi_{t \to x}(a), \pi_{t \to y}(b), z] \big)$$

and $\neg a = t/a$ or $t \cdot a^{\rho}$. So simple non-Abelian loops are strongly Boolean-complete; and since non-solvable loops have simple non-Abelian divisors, they are Boolean-complete by lemma 1. □

In fact, simple non-Abelian groups [15], simple loops [13], and non-affine simple quasigroups [16] have a stronger property, that their polynomial closure contains *all* possible n-ary functions on their elements. This is called *functional completeness*, and is of interest in the field of multi-valued logic [22, 25]. However, Boolean-completeness is sufficient for our purposes.

Since the CIRCUIT VALUE problem for solvable groups is in **NC** [4, 17], non-solvability is both necessary and sufficient for Boolean-completeness in the case of groups (or semigroups, in fact). This also means that in the associative case, Boolean-completeness and strong Boolean-completeness are equivalent.

However, a loop can be solvable and still be (strongly) Boolean-complete. Let $(G, \cdot)$ be

$\cdot$	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8
2	2	3	4	1	6	7	8	5
3	3	4	1	2	7	8	5	6
4	4	1	2	3	8	5	6	7
5	5	6	7	8	1	3	2	4
6	6	7	8	5	3	2	4	1
7	7	8	5	6	2	4	1	3
8	8	5	6	7	4	1	3	2

Here G' is the normal subloop $\{1,2,3,4\} \cong \mathbb{Z}_4$. But the lower right-hand block is the Boolean-complete quasigroup $\star$ given above, and $\star$ can be expressed in $\mathcal{P}(G)$ as $a \star b = (5 \cdot a) \cdot (5 \cdot b)$. Then if FALSE $= 1$ and TRUE $= 2$ as before, we can write $a \wedge b = [5 \cdot ((5a)(5b))]^2$ and G is Boolean-complete. We also note that this loop has a solvable multiplication group, which we will discuss below.

A loop has the *inverse property* [20] if $a^{-1}(ab) = (ba)a^{-1} = b$, where $a^{-1} = a^\lambda = a^\rho$. Solvable loops with the inverse property can also be Boolean-complete: we believe the smallest example is

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	3	4	5	1	7	9	10	8	6
3	3	4	5	1	2	8	10	7	6	9
4	4	5	1	2	3	9	8	6	10	7
5	5	1	2	3	4	10	6	9	7	8
6	6	7	8	9	10	1	2	3	4	5
7	7	9	10	8	6	2	4	5	3	1
8	8	10	7	6	9	3	5	2	1	4
9	9	8	6	10	7	4	3	1	5	2
10	10	6	9	7	8	5	1	4	2	3

Here $G' = \{1,2,3,4,5\} \cong \mathbb{Z}_5$, so G is solvable. However, if FALSE $= 1$ and TRUE $= 2$, we can define $a \wedge b = [a, b, 6]$ since $[2,2,6] = 2$ (and $\neg a = 2/a$).

In both these examples, the lower right-hand block can be any quasigroup whatsoever: clearly the standard definition of "solvable" for loops is not a very meaningful constraint for our purposes. In the next section we will show that for loops and quasigroups, a property slightly more subtle than solvability is the dividing line between Boolean-completeness and -incompleteness.

4 Polyabelian algebras

4.1 Definition and properties

The *direct product* of two algebras $A \times B$ is the set of pairs (a, b) with pairwise multiplication, $(a_1, b_1) \cdot (a_2, b_2) = (a_1 a_2, b_1 b_2)$. Consider the following generalization:

Definition. A *quasidirect product* [7] of two algebras A and B is the set of pairs (a, b), under an operation of the form

$$(a_1, b_1) \cdot (a_2, b_2) = (a_1 a_2,\ b_1 \odot_{a_1,a_2} b_2)$$

where each a_1, a_2 defines a *local operation* $\odot_{a_1,a_2}$ on B. We will denote such a product $A \otimes B$.

If the local operations are of the form

$$b_1 \odot_{a_1,a_2} b_2 = f_{a_1,a_2}(b_1) \cdot g_{a_1,a_2}(b_2)$$

we will call them *separable.* Furthermore, if B is an Abelian group and the $\odot$'s are of the form

$$b_1 \odot_{a_1,a_2} b_2 = f_{a_1,a_2}(b_1) + g_{a_1,a_2}(b_2) + h_{a_1,a_2}$$

where f and g are endomorphisms on B and h is an element of B, depending arbitrarily on a_1 and a_2, then we will call them *affine.* We will call a quasidirect product $A \otimes B$ separable or affine on B if all its local operations are.

Lemma 7. *If $A \otimes B$ is a quasigroup, then A and B are quasigroups and all the $\odot$'s are quasigroup operations on B. If $A \otimes B$ is a quasigroup and is affine on B, then all the f's and g's are automorphisms on B. If $A \otimes B$ is a loop affine on $(B, +)$, then A is a loop, and for all $a \in A$ we have $f_{a,0} = g_{0,a} = \mathbf{1}$ and $h_{a,0} = h_{0,a} = 0$ (here we are using 0 for the identities of both A and B.) Thus $b_1 \odot_{0,0} b_2 = b_1 + b_2$ for $b_1, b_2 \in B$, and B is a subloop of $A \otimes B$ on which $+$ and $\cdot$ coincide.*

Proof. Easy; left to the reader. □

The quasidirect product is a rather general way of extending to a loop from a normal subloop:

Lemma 8. *If a loop G has a normal subloop N, then G is isomorphic to a quasidirect product $(G/N) \otimes N$. Furthermore, all the local operations, and (if N is Abelian and G is affine on N) the f's, g's and h's, are expressible in $\mathcal{P}(G)$.*

Proof. Choose a set T with one element in each coset of N, and define an operation $\bullet$ on T where $t_1 \bullet t_2$ is the element of T in the same coset as $t_1 \cdot t_2$. Then clearly $T \cong G/N$.

Every element can be uniquely written $g = tn$ where $t \in T$ and $n \in N$. Then

$$(t_1 n_1) \cdot (t_2 n_2) = (t_1 \bullet t_2) \cdot (n_1 \odot_{t_1,t_2} n_2)$$

where

$$n_1 \odot_{t_1,t_2} n_2 = (t_1 \bullet t_2) \setminus ((t_1 n_1) \cdot (t_2 n_2))$$

which is in N since N is normal. Thus G is a quasidirect product $T \otimes N$, and all the local operations are in $\mathcal{P}(G)$. If N is an Abelian group and G is affine on N, then

$$\begin{aligned} f_{a_1,a_2}(n) &= (n \odot_{a_1,a_2} 0) - (0 \odot_{a_1,a_2} 0) \\ g_{a_1,a_2}(n) &= (0 \odot_{a_1,a_2} n) - (0 \odot_{a_1,a_2} 0) \\ h_{a_1,a_2} &= (0 \odot_{a_1,a_2} 0) \end{aligned}$$

where we abuse notation by writing $+$ and 0, instead of $\cdot$ and 1, for products in N. Finally, $(N, +)$ is in $\mathcal{P}(G)$ since $a \odot_{0,0} b = a + b$ by lemma 7. $\square$

Then define the following class of algebras:

Definition. An algebra is *polyabelian* if it is an iterated quasidirect product of Abelian groups A_i:

$$((A_0 \otimes A_1) \otimes A_2) \otimes \cdots \otimes A_k$$

where all the products are affine.

It is easy to show that subalgebras, factors, and finite direct products of polyabelian algebras are polyabelian, so this class forms a pseudovariety. The next few lemmas show inclusions between the polyabelian loops and some common classes of groups and loops.

Lemma 9. *Polyabelian loops are solvable.*

Proof. Let $H_i = (A_i \otimes A_{i+1}) \otimes \cdots \otimes A_k$ with $H_0 = G$. Then the reader can show that all the H_i are normal subloops of G, and $H_i/H_{i+1} = A_i$ is Abelian. Therefore, $H_i' \subseteq H_{i+1}$ and the derived series ends after at most k steps. $\square$

The converse is not true for loops in general (for instance, the solvable Boolean-complete loops above, since the local operations in their lower right-hand blocks are not affine) but it is true for groups:

Lemma 10. *Solvable groups are polyabelian.*

Proof. Any solvable group G has a normal subgroup N which is Abelian, namely the last non-trivial group in its derived series such that $N' = \{1\}$. Then the local operation in $G/N \otimes N$ (from lemma 8) can be written

$$\begin{aligned} n_1 \odot_{t_1,t_2} n_2 &= (t_1 \bullet t_2)^{-1} t_1 n_1 t_2 n_2 \\ &= ((t_1 \bullet t_2)^{-1} t_1 t_2) + (t_2^{-1} n_1 t_2) + n_2 \end{aligned}$$

where we use $+$ for products within N. Thus G is affine on N where

$$\begin{aligned} f_{t_1,t_2}(n) &= t_2^{-1} n t_2 \\ g_{t_1,t_2}(n) &= n \\ h_{t_1,t_2} &= (t_1 \bullet t_2)^{-1} t_1 t_2 \end{aligned}$$

Then G/N has an Abelian normal subgroup, and so on; by induction G is polyabelian.

(If $t_1 \bullet t_2 = t_1 t_2$ so that $h = 0$, then T is a subgroup of G isomorphic to G/N, the quasidirect product reduces to the *semidirect product* on groups, and G is a *split extension* of N by T [24]. In [17] we defined polyabelianness with semidirect products only, in which case any solvable group is a subgroup of a polyabelian group by a theorem regarding wreath products.) □

Lemma 11. *Nilpotent loops are polyabelian.*

Proof. Let G be a nilpotent loop with center $Z(G)$. Then the local operation in $G/Z(G) \otimes Z(G)$ is

$$n_1 \odot_{t_1,t_2} n_2 = ((t_1 \bullet t_2)\backslash t_1 t_2) + n_1 + n_2$$

since n_1 and n_2 associate and commute with everything. So G is affine on $Z(G)$ with $f = g = \mathbf{1}$ and $h = (t_1 \bullet t_2)\backslash t_1 t_2$. Then $G/Z(G)$ has a non-trivial center, and so on; by induction G is polyabelian. □

Thus polyabelianness coincides with solvability for groups, and lies properly between nilpotence and solvability for loops.

We wish to show that, for purposes of Boolean-completeness, polyabelianness is the correct generalization of solvability in the non-associative case: that is, an algebra is Boolean-complete if and only if it is not polyabelian. This will turn out to be true for loops and quasigroups, but not for algebras in general.

4.2 Polyabelian algebras aren't Boolean-complete

In one direction, we can prove this for all algebras. In [17] we show that CIRCUIT VALUE for polyabelian algebras is in $\mathbf{ACC}^1$, and a simple modification of the proof for solvable semigroups in [4] shows that it is also in **DET**. We now show directly that polyabelian algebras cannot express the AND function. First, two lemmas from [25]:

Definition. A function ϕ on variables $x_1, \ldots, x_n$ is *affine* if there exists an Abelian group A, endomorphisms $f_1, \ldots, f_n$, and an element h such that $\phi(x_1, \ldots, x_n) = \sum_i f_i(x_i) + h$.

Lemma 12. *If A is an Abelian group, then any function in $\mathcal{P}(A)$ is affine, and the affine functions are closed under composition.*

Proof. Obvious: $\phi(a, b) = a + b$ is affine, and if ϕ_1 and ϕ_2 are both affine, then so are $\phi_1 + \phi_2$ and $\phi_1 \circ \phi_2$. □

Lemma 13. *If $\phi(a, b)$ is an affine function, then $\phi(a_1, b_1) = \phi(a_1, b_2)$ if and only if $\phi(a_2, b_1) = \phi(a_2, b_2)$ for any four elements a_1, a_2, b_1, b_2.*

Proof. We can write $\phi(a,b) = f(a) + g(b) + h$ where f and g are endomorphisms. Then $\phi(a_1,b_1) = \phi(a_1,b_2)$ implies that $g(b_1) = g(b_2)$, which in turn implies that $\phi(a_2,b_1) = \phi(a_2,b_2)$ for any a_2. □

Theorem 14. *Polyabelian algebras cannot express the* AND *function, and so are not Boolean-complete.*

Proof. If G is Boolean-complete, then it can express an n-ary AND function for any n, i.e. $\phi(a_1, a_2, \ldots, a_n) \in T$ if and only if $a_i \in T$ for all i (assuming that $a_i \in T \cup F$ for all i). We will show that this is impossible for n sufficiently large.

If $G = (A_0 \otimes A_1) \otimes \cdots \otimes A_k$, then any $g \in G$ has a unique vector of components $(g_0, g_1, \ldots, g_k)$ where $g_i \in A_i$ for all i. Call g_i the A_i*-component* of g. We will proceed through the A_i by induction, showing that there are elements of T and F matching on all their components, and therefore equal; then T and F are not disjoint, a contradiction.

We will use the following fact: if $\psi(a_1, \ldots, a_n) = \sum_i f_i(a_i) + h$ is an n-ary affine function on an Abelian group A of order p, and if A has k distinct endomorphisms (for instance, $k = p^{m^2}$ for $A = \mathbb{Z}_p^m$ since its endomorphisms are $m \times m$ matrices with entries in $\mathbb{Z}_p$), and if n is greater than $(p-1)k$, then at least p of the variables have the same $f_i = f$. Then if these p variables are all equal, they contribute nothing to ψ since pf=0. In particular, if the $n-p$ other variables are true, ψ has the same value whether these p variables are true or false. As shorthand for this, we write $\psi(f^p t^{n-p}) = \psi(t^n)$. Thus ψ cannot be an AND function.

So assume that there is an n-ary AND function ϕ in $\mathcal{P}(G)$. To start the induction, since A_0 is a factor of G, then ϕ's A_0-component ϕ_0 is a function of the A_0-components of the a_i, expressible in $\mathcal{P}(A_0)$ and therefore affine by lemma 12. Choose $t \in T$ and $f \in F$; then for n sufficiently large $\phi_0(f^p t^{n-p}) = \phi_0(t^n)$. Let $f_0 = \phi(f^p t^{n-p})$ and $t_0 = \phi(t^n)$; then $t_0 \in T$ and $f_0 \in F$ by hypothesis, and they have the same A_0-component.

Now suppose that $t_m \in T$ and $f_m \in F$ agree on their A_j-components for all $j \leq m$. Think of ϕ as a tree where each node outputs the product of two subexpressions according to some local product. Now the A_j-components at each node depend only on the $A_{j'}$-components of its two subexpressions for $j' \leq j$ (since $A_0 \otimes \cdots \otimes A_j$ is a factor of G for all j) and t_m and f_m have the same A_j-component for all $j \leq m$, so inductively ϕ and each of its subexpressions have constant A_j-components for $j \leq m$ when restricted to inputs in $\{t_m, f_m\}$.

Furthermore, each node applies an affine local operation on A_{m+1}, and which one it applies depends only on its subexpressions' A_j-components for $j \leq m$. Since these are constant in this restriction, each node always applies the same local operation; the composition of all of these make ϕ_{m+1} an affine function on the A_{m+1}-components of its inputs.

Then if we let $f_{m+1} = \phi(f_m^p t_m^{n-p}) \in F$ and $t_{m+1} = \phi(t_m^n) \in T$, we see that f_{m+1} and t_{m+1} agree on their A_j-components for all $j \leq m+1$. After k steps of this induction, t_k and f_k agree on all their components, and so are equal; so T and F are not disjoint. □

4.3 Non-polyabelian loops and quasigroups are Boolean-complete

Theorem 5 and lemma 10 show that non-polyabelianness implies Boolean-completeness in the case of groups; we will now show this for loops and then for quasigroups, using slightly different techniques.

Theorem 15. *Non-polyabelian loops are Boolean-complete.*

Proof. Let H be the smallest non-polyabelian divisor of G. We will show that H is strongly Boolean-complete.

Assume without loss of generality that H is solvable, since we have already treated the non-solvable case with theorem 6. Then H has a normal subloop K which is an Abelian group, namely the last non-trivial subloop in its derived series with $K' = \{1\}$. Let N be a minimal normal subloop of H contained in K; then N is also Abelian. We know that H is not affine on N; otherwise H/N would be a smaller non-polyabelian divisor of G.

Recall the definition of $U(x)$ from lemma 4. Since N is minimal, $U(n) \supset N$ for any $n \in N$; otherwise $U(n) \cap N$ would be a smaller normal subloop since the intersection of normal subloops is normal. So for any $n_1, n_2 \in N$, there exists a function $\pi_{n_1 \to n_2} \in \mathcal{P}(H)$ that sends n_1 to n_2 and preserves the identity.

Since H is not affine on N, some local operation $\odot$ is either not separable or not affine. Define the *separator*

$$K_\odot(n_1, n_2) = (n_1 \odot n_2) - (n_1 \odot 0) - (0 \odot n_2) + (0 \odot 0)$$

where we use $+$ and $-$ for products in N. If $\odot$ is not separable, then $K_\odot(n_1, n_2) = k \neq 0$ for some n_1, n_2; however, $K_\odot(0, n) = K_\odot(n, 0) = 0$ for any n. But this gives us our AND gate: let FALSE $= 0$ and choose TRUE $= t \in N$, and let

$$a \wedge b = \pi_{k \to t}\big(K_\odot(\pi_{t \to n_1}(a), \pi_{t \to n_2}(b))\big)$$

If all the local operations are separable, then one must not be affine: that is, some f or g is not a endomorphism of N. Let $f(n) = (n \odot 0) - (0 \odot 0)$ as in lemma 8, and define the *affinator*

$$L_f(n_1, n_2) = f(n_1 + n_2) - f(n_1) - f(n_2)$$

If f is not a endomorphism, then $L_f(n_1, n_2) = k \neq 0$ for some n_1, n_2; but $L_f(n, 0) = L_f(0, n) = 0$ for all n, so

$$a \wedge b = \pi_{k \to t}\big(L_f(\pi_{t \to n_1}(a), \pi_{t \to n_2}(b))\big)$$

is an AND gate. Similarly if some g is not a endomorphism.

So any nonlinearity in the local operations can be used to construct an AND gate, and we can express negation $\neg a = t/a$ as before. Thus H is strongly Boolean-complete, and G is Boolean-complete by lemma 1. □

For the quasigroup case, we need some additional definitions and lemmas. The idea of a normal subloop generalizes in the following way:

Definition. A *normal congruence* on a quasigroup Q is an equivalence $\sim$ such that if $a_1 \sim a_2$ and $b_1 \sim b_2$, then $a_1 a_2 \sim b_1 b_2$. In other words, the equivalence class of the product depends only on the equivalence classes of a and b, so the map from Q to the set of equivalence classes $Q/\sim$ is a homomorphism. (This definition works for finite quasigroups; for infinite ones, additional requirements are needed to make sure $Q/\sim$ is also a quasigroup [20].) For a loop with a normal subloop, for instance, the equivalence classes are simply its cosets.

A *proper* congruence is one other than the identity ($a \sim b$ only if $a = b$) or all of Q ($a \sim b$ for all a, b). A quasigroup is *simple* if it has no proper congruences.

A normal congruence $\sim$ is *minimal* if there is no proper normal congruence whose equivalence classes are properly contained in those of $\sim$.

Then lemma 4 generalizes in the following way:

Lemma 16. *If Q is a simple quasigroup, then for any x, y, z, w with $x \neq z$, there exists a (not necessarily unique) function $\pi_{x \to y, z \to w}$ in $\mathcal{P}^1(Q)$ that sends x to y and z to w.*

Proof. Fix z, and let $U_{z \to w}$ be the set of functions ϕ in $\mathcal{P}^1(Q)$ such that $\phi(z) = w$. Then $U_{z \to w}(x)$ is the set of y's that we can send x to, while sending z to w.

Suppose $y_1 \in U_{z \to w_1}(x)$ and $y_2 \in U_{z \to w_2}(x)$; that is, suppose there are $\phi_1, \phi_2 \in \mathcal{P}(Q)$ such that $\phi_1(x) = y_1$, $\phi_1(z) = w_1$, $\phi_2(x) = y_2$ and $\phi_2(z) = w_2$. Then if $\phi'(a) = \phi_1(a) \cdot \phi_2(a)$, we have $\phi'(x) = y_1 \cdot y_2$ and $\phi'(z) = w_1 \cdot w_2$. Therefore, $\phi' \in U_{z \to w_1 \cdot w_2}$ and $y_1 \cdot y_2 \in U_{z \to w_1 \cdot w_2}(x)$.

So for each x and z, the sets $U_{z \to w}(x)$ for different w are equivalence classes of a normal congruence. Moreover, each one has more than one element: for instance, w and $(w/z)x$ are both in $U_{z \to w}(x)$ by the functions $\phi(a) = w$ and $(w/z)a$, and are distinct in a quasigroup if $x \neq z$.

Since Q is simple, it has no proper normal congruences, and $U_{z \to w}(x) = Q$; so for every y, there exists a function $\phi \in U_{z \to w}(x)$ that sends x to y, which is our $\pi_{x \to y, z \to w}$. □

Two more lemmas will prove useful.

Lemma 17. *If a quasigroup $(Q, \star)$ is separable on a loop operation $\cdot$, i.e. if $a \star b = f(a) \cdot g(b)$ for some f, g, then $a \cdot b$ is expressible in $\mathcal{P}^2(Q, \star)$.*

Proof. Let the identity of $\cdot$ be 1. Then $f(a) = a \star g^{-1}(1)$ and $g(a) = f^{-1}(1) \star a$. Since Q is a quasigroup, f and g are one-to-one, and $f^{-1} = f^{n!-1}$ and $g^{-1} = g^{n!-1}$ if Q has order n. Then $a \cdot b = f^{-1}(a) \star g^{-1}(b)$. (Note that it is not necessary for f and g to be automorphisms.) □

Lemma 18. *Simple non-affine quasigroups are strongly Boolean-complete.*

Proof. This is just a special case of the functional completeness result mentioned above [16]. We rely on the following fact, the proof of which can be found is [25]: if Q is not affine, then there is a function $\phi \in \mathcal{P}(Q)$ that violates the statement of lemma 13, i.e. $\phi(a_1,b_1) = \phi(a_1,b_2)$ but $\phi(a_2,b_1) \neq \phi(a_2,b_2)$ for some a_1,a_2,b_1,b_2.

Then choose an element x, and define

$$\psi(a,b) = (x \cdot \phi(a,b))/\phi(a,b_1)$$

The reader can check that $\psi(a,b) = x$ for three out of four combinations of a_1, a_2, b_1 and b_2, except $\psi(a_2,b_2)$ which is some $y \neq x$. This has the shape of an AND gate, and using lemma 16 we can define

$$a \wedge b = \pi_{x\to\text{FALSE},y\to\text{TRUE}}\big(\psi(\pi_{\text{FALSE}\to a_1,\text{TRUE}\to a_2}(a),\ \pi_{\text{FALSE}\to b_1,\text{TRUE}\to b_2}(b))\big)$$

and

$$\neg a = \pi_{\text{TRUE}\to\text{FALSE},\text{FALSE}\to\text{TRUE}}(a)$$

of which one example derived from a *Mal'cev operation* [25] is

$$(\text{TRUE}/\text{TRUE})\backslash((\text{TRUE}/a)\cdot\text{FALSE})$$

since $(t/t)\backslash t = (t/t)\backslash((t/t)t) = t$. □

Finally, we note that lemma 8 holds for quasigroups as well: if Q has a normal congruence $\sim$, then $Q \cong (Q/\sim)\otimes N$ where N is any one of the equivalence classes, and the local operations on N are in $\mathcal{P}(Q)$. (Unlike the loop case, N might not be a subquasigroup of Q.) Lemma 1 also holds: if $Q/\sim$ is Boolean-complete, then so is Q. Then we can prove

Theorem 19. *Non-polyabelian quasigroups are Boolean-complete.*

Proof. As before, replace Q with its smallest non-polyabelian divisor R, which we will show is strongly Boolean-complete. Since lemma 18 takes care of the case when R is simple, we assume without loss of generality that R has a minimal normal congruence $\sim$ such that R is not affine on its equivalence classes.

If there is a particular local operation which is not affine, then it is a simple quasigroup operation expressible in $\mathcal{P}(R)$, and lemma 18 applies. However, it could be the case that every local operation is affine but not all on an identical Abelian group. For instance, in the Boolean-complete loop of order 10 above, every local operation is isomorphic to $\mathbb{Z}_5$, but in a way which is not affine on each other: the lower right-hand block is the upper left-hand block with 3 and 4 switched, but there is no automorphism of $\mathbb{Z}_5$ that only switches two elements.

In this case, we use lemma 17: if any of the operations is affine on an Abelian group A, then $(A,+)$ is expressible in $\mathcal{P}(R)$. Then if any other local operation is not also affine on A, we use its separator or affinator with respect to $+$ to construct an AND gate as in theorem 15.

Then R is strongly Boolean-complete and Q is Boolean-complete by lemma 1. □

Corollary. *The non-Boolean-complete quasigroups and the polyabelian quasigroups form the same pseudovariety.*

4.4 Algebras in general

It would be very nice if non-polyabelianness were the criterion for Boolean-completeness for algebras in general (presumably with 'Abelian group' replaced by 'Abelian semigroup' in the definition). However, we can give a counterexample. Let $(A, \star)$ be

$\star$	0	1	2	3
0	0	1	2	3
1	1	0	2	3
2	2	2	0	0
3	3	3	0	1

Theorem 20. *$(A, \star)$ is non-polyabelian, but cannot express the* AND *function.*

Proof. The equivalence classes $\{0, 1\}$ and $\{2, 3\}$ form the only proper normal congruence $\sim$ of A. There is no Abelian semigroup on which the lower-right and upper-left local operations are both affine, so A is not polyabelian.

We could not have TRUE and FALSE in different equivalence classes, since $(A/\!\sim) \cong \mathbb{Z}_2$. So we will assume first that $\text{TRUE}, \text{FALSE} \in \{0, 1\}$.

The lower right-hand block looks like an AND gate with $\text{TRUE} = 1$ and $\text{FALSE} = 0$, in which case we could use the upper left-hand block to say $\neg a = 1 \star a$. But there is no way to map, say, 0 to 2 and 1 to 3, since $a \star b = b \star a = a$ whenever $a \in \{2, 3\}$ and $b \in \{0, 1\}$; that is, 2 and 3 dominate 0 and 1.

Formally, consider a polynomial expression ϕ in $\mathcal{P}(A)$. The following rules preserve the value of ϕ on inputs restricted to $\{0, 1\}$:

- Replace $a \star \phi$ and $\phi \star a$ with a if $a \in \{2, 3\}$ and ϕ contains only elements of $\{0, 1\}$, and
- Replace $a \star b$ with the appropriate constant in $\{0, 1\}$ if $a, b \in \{2, 3\}$.

Applying these rules repeatedly will leave us either with a constant in $\{2, 3\}$ or with variables and constants in $\{0, 1\}$, on which $\star$ is a subalgebra isomorphic to $\mathbb{Z}_2$. So any function in $\mathcal{P}(A)$ is constant or affine when restricted to variables in $\{0, 1\}$, and cannot express an AND.

Finally, assume that $\text{TRUE}, \text{FALSE} \in \{2, 3\}$. Any node whose output is 2 or 3 is equal to either its left or its right input (whichever one is 2 or 3), which in turn is equal to one of its inputs, and so on back up to a single constant or variable. Since $\sim$ is normal, for inputs in $\{2, 3\}$ the same subexpressions always have values in $\{2, 3\}$, and this path always leads back to the same input or constant. So any function in $\mathcal{P}(A)$ with an output in $\{2, 3\}$ is either constant or equal to one of its inputs when restricted to variables in $\{2, 3\}$, and cannot be an AND. $\square$

Note that we have not shown that this algebra's EXPRESSION EVALUATION or CIRCUIT VALUE problem is in **NC**, simply that it is not Boolean-complete. Perhaps the reader can find some more subtle analog of polyabelianness that works for all algebras.

5 $\mathcal{M}$-solvability and expression evaluation

We now consider the relationship between expressions in an algebra and words in that algebra's multiplication group (or semigroup).

Definition. An algebra is $\mathcal{M}$*-solvable* if its multiplication semigroup is solvable.

For groups, solvability and $\mathcal{M}$-solvability coincide since one can show that $\mathcal{M}(G)$ is isomorphic to $(G \times G)/Z(G)$ and is solvable if G is. For loops, $\mathcal{M}$-solvability implies solvability by a recent result of Vesanen [26], and nilpotent loops are $\mathcal{M}$-solvable by a theorem of Bruck [5]. Thus, like polyabelianness, $\mathcal{M}$-solvability is a generalization of solvability which lies properly between nilpotence and solvability for loops.

However, the $\mathcal{M}$-solvable loops and polyabelian loops are incomparable. First we show that $\mathcal{M}$-solvability is related to the solvability of the automorphisms generated by the f's and g's in a quasidirect product:

Definition. For an affine quasidirect product $A \otimes B$, define $\langle fg(B)\rangle$ as the subgroup of $\mathrm{Aut}(B)$ generated by all the f's and g's in the local operations, and define $\langle fgh(B)\rangle$ as the group of affine operations on B generated by all the rows and columns of the local operations.

We need the following definition from [10]:

Definition. The *wreath product* of B by A, written $A \wr B$, is a particular kind of semidirect product $A \otimes B^A$ where B^A is the set of functions β from A to B, with multiplication defined as

$$(a_1, \beta_1) \cdot (a_2, \beta_2) = (a_1 a_2, \beta') \quad \text{where} \quad \beta'(a) = \beta_1(a_2 a) \cdot \beta_2(a)$$

In other words, each element of $A \wr B$ consists of an element of A and a vector β of $|A|$ elements of B; the β's are multiplied componentwise, but with the components of β_1 permuted by a_2. Thus the A-component is affected by $a \in A$, while the B-component within a block $\{a\} \times B$ is affected by $\beta(a) \in B^A$.

Then we have

Lemma 21. *If $G = A \otimes B$ is a quasigroup, then $\mathcal{M}(B, \odot)$ for every local operation $\odot$ is contained in a divisor of $\mathcal{M}(G)$. Furthermore, if G is affine on B, the following are equivalent:*

- *G is $\mathcal{M}$-solvable*
- *$\langle fg(B)\rangle$ is solvable*
- *$\langle fgh(B)\rangle$ is solvable.*

Proof. Let $(G, \cdot) = A \otimes B$. Let H be the subgroup of $\mathcal{M}(G)$ consisting of those multiplications that preserve the blocks $\{a\} \times B$ (i.e. that leave the A-component unchanged), and let N be the subgroup of H that also fixes the B-components of elements in a particular block $\{a_0\} \times B$. It is easy to see that N is normal in H, and that H/N is isomorphic to the set of permutations of B that can be carried out with multiplications in G.

Although, unlike the loop case, $\{a_0\} \times B$ might not be a subalgebra, we can identify b with (a_0, b) and define the local operations as $b_1 \odot b_2 = a \setminus (a_1 b_1)(a_2 b_2)$ where a is chosen so that the A-component of $b_1 \odot b_2$ is a_0. Then multiplications in $\odot$ are compositions of multiplications in G: for instance, $R_b^{\odot} = L_a^{-1} R_{(a_2 b)} L_{a_1}$. So $\mathcal{M}(B, \odot) \subseteq H/N$.

Now suppose G is affine on B. Since the rows and columns of all the $\odot$'s generate exactly $\langle fgh(B)\rangle$, we have $\langle fgh(B)\rangle = H/N$. The subgroup of $\langle fgh(B)\rangle$ that preserves the identity of B is simply $\langle fg(B)\rangle$. Since divisors of solvable groups are solvable, both these groups are solvable if $\mathcal{M}(G)$ is.

Conversely, since elements of $\mathcal{M}(G)$ permute the blocks with each other by elements of A while permuting them internally by the rows and columns of the $\odot$'s, $\mathcal{M}(G)$ is a subgroup of the wreath product $\mathcal{M}(A) \wr \langle fgh(B)\rangle$, and the wreath product of two solvable groups is solvable.

Finally, since two affine functions $\phi_1(b) = f_1(b) + h_1$ and $\phi_2(b) = f_2(b) + h_2$ compose as

$$(\phi_1 \circ \phi_2)(b) = (f_1 \circ f_2)(b) + h_1 + \phi_1(h_2)$$

we see that $\langle fgh(B)\rangle$ is a semidirect product $\langle fg(B)\rangle \otimes B$. The semidirect product of two solvable groups is solvable, and B is Abelian; so $\langle fgh(B)\rangle$ is solvable if $\langle fg(B)\rangle$ is. ☐

Theorem 22. *The classes of polyabelian and $\mathcal{M}$-solvable loops are incomparable.*

Proof. First, we construct a polyabelian loop which is not $\mathcal{M}$-solvable. Let B be an Abelian group with a non-solvable automorphism group: for instance, $\mathrm{Aut}(\mathbb{Z}_2^3)$ is the simple group of order 168, and is generated by two elements [27]

$$f = \begin{pmatrix} 0\,1\,0 \\ 0\,0\,1 \\ 1\,1\,0 \end{pmatrix} \quad \text{and} \quad g = \begin{pmatrix} 0\,0\,1 \\ 1\,0\,0 \\ 0\,1\,0 \end{pmatrix}$$

If we let $G = \mathbb{Z}_2 \otimes \mathbb{Z}_2^3$ where $b_1 \odot_{1,1} b_2 = f(b_1) + g(b_2)$, then $\langle fg(B)\rangle = \mathrm{Aut}(B)$ is non-solvable and G is not $\mathcal{M}$-solvable by lemma 21. (This produces a loop of order 16; since Abelian groups smaller than $\mathbb{Z}_2^3$ all have solvable automorphism groups, we believe this is the smallest possible.)

Conversely, the Boolean-complete loop of order 8 given in section 3 above is $\mathcal{M}$-solvable: $\mathcal{M}(G)$ consists of those permutations of eight elements which either preserve or switch the blocks $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$. This is the wreath product $\mathbb{Z}_2 \wr S_4$, and its derived subgroup $\mathcal{M}(G)'$ is the subset of $S_4 \times S_4$ consisting of pairs of permutations whose total parity is even. The rest of the derived series is

$$\mathcal{M}(G)' \supset A_4 \times A_4 \supset \mathbb{Z}_2^4 \supset \{1\}$$

Thus non-polyabelian, and Boolean-complete, loops can be $\mathcal{M}$-solvable. ☐

We now give a simple result relating expressions in $\mathcal{M}(G)$ to expressions in G.

Lemma 23. *For any algebra G, the* Expression Evaluation *problem of $\mathcal{M}(G)$ is $\mathbf{NC}^0$-reducible to that of G.*

Proof. Suppose we have a word in $\mathcal{M}(G)$ such as $L_a R_b L_c$. If G has n elements, the product is determined by its action on each element $1, \ldots, n$. But this just means evaluating n expressions in G, namely $a((c1)b), a((c2)b), \ldots, a((cn)b)$. This is easily seen to be an $\mathbf{NC}^0$ reduction. □

Then we immediately have

Theorem 24. *The* Expression Evaluation *problem for non-$\mathcal{M}$-solvable algebras is $\mathbf{NC}^1$-complete under $\mathbf{AC}^0$ reductions.*

Proof. By lemma 23 and the fact that Expression Evaluation for non-solvable semigroups is $\mathbf{NC}^1$-complete under $\mathbf{AC}^0$ reductions [2]. □

No analogue of lemma 23 seems to exist in the case of circuits: since Circuit Value is $\mathbf{P}$-complete for non-solvable algebras but in $\mathbf{ACC}^1$ for non-$\mathcal{M}$-solvable ones that are polyabelian, circuits over $\mathcal{M}(G)$ are not easy to simulate with circuits over G, or vice versa, unless $\mathbf{P} = \mathbf{ACC}^1$.

As mentioned above, Expression Evaluation is in $\mathbf{ACC}^0$ for solvable groups [2]. In this case, the correct generalization of solvability in the non-associative case consists of being *both* polyabelian and $\mathcal{M}$-solvable, but the need to parse the expression makes the problem somewhat harder:

Theorem 25. *For an algebra which is both polyabelian and $\mathcal{M}$-solvable,* Expression Evaluation *is in $\mathbf{TC}^0$, or in $\mathbf{ACC}^0$ if the expression's parse tree is already known.*

Proof. Consider a polyabelian algebra $A = (A_0 \otimes A_1) \otimes \cdots \otimes A_k$. An expression such as $\phi = (x_1 \cdot (x_2 \cdot x_3)) \cdot x_4$ on variables $x_1, \ldots, x_n$ can be inductively calculated in the following way (similar to the algorithm in [17] for Circuit Value): the A_0-component is just a sum in an Abelian group, and the A_{m+1} component is an expression with affine local operations $\odot_i$,

$$(x_1 \odot_1 (x_2 \odot_2 x_3)) \odot_3 x_4$$

where the $\odot_i$ are determined by the A_m-components of the subexpressions to their left and right.

If $a \odot_i b = f_i(a) + g_i(b) + h_i$ for all i, this is a sum

$$\sum_{i=1}^{n} F_i(x_i) + \sum_{j=1}^{n-1} G_j(h_j)$$

where the F_i and G_j are endomorphisms of A_{m+1} composed of less than n of the f's and g's. In this case, the reader can verify that

$$\begin{array}{ll} F_1 = f_3 f_1 & G_1 = f_3 \\ F_2 = f_3 g_1 f_2 & G_2 = f_3 g_1 \\ F_3 = f_3 g_1 g_2 & G_3 = \mathbf{1} \\ F_4 = g_3 & \end{array}$$

Each one of these is a word in $\langle fg(A_{m+1})\rangle$; by lemma 21 this is solvable if A is $\mathcal{M}$-solvable. Since Expression Evaluation is in $\mathbf{ACC}^0$ for solvable groups [2], these words and sums can be evaluated in $\mathbf{ACC}^0$. We can calculate all of ϕ's components with k induction steps, and we're done.

However, how the f's and g's contribute to the F's and G's depends on the expression's parse tree. For each subexpression $(\phi \odot_k \phi')$ that x_i is contained in, F_i gains an f_k or g_k depending on whether x_i is to the left or right of $\odot_k$. Similarly, G_i gains an f_k or g_k for each subexpression $(\phi \odot_k \phi')$ that $\odot_i$'s subexpression is contained in.

The set of well-formed expressions is a *structured context-free language* [11], and such expressions can be parsed in $\mathbf{TC}^0$ as shown in [3]. In particular, threshold gates can count the *ascent* of a string in the parse tree, defined as the number of)'s minus the number of ('s; then x is in the subexpression to the left of $\odot$ if the ascent of the string between them is at least as great as the ascent of any of its initial substrings.

Thus we can compute ϕ by first parsing it with a $\mathbf{TC}^0$ circuit, and then calculating the above sum with k levels of $\mathbf{ACC}^0$ circuits.

(Since the expression only has to be parsed once, the *majority-depth* of the circuit, defined as the maximum number of threshold gates that any path traverses, is constant for all algebras. Thus this problem is in $\widehat{\mathbf{TC}}^0_k$ for some small k as defined in [14].) $\square$

Since nilpotent loops are both polyabelian (by lemma 11) and $\mathcal{M}$-solvable [5], we have the corollary:

Corollary. Expression Evaluation *is in* $\mathbf{TC}^0$, *or* $\mathbf{ACC}^0$ *if the expression's parse tree is already known, for nilpotent loops.*

Cases where the parse tree is already known could include uniform families of expressions, one of each length. For instance, the problem of predicting a cellular automaton amounts to evaluating a uniform family of circuits, one of each size. This uniformity can significantly simplify the Circuit Value problem as in [17], where cellular automata based on nilpotent groups are shown to be predictable in $\mathbf{ACC}^0$.

6 Conclusion and directions for further work

We have shown that the relationship between solvability and circuit complexity generalizes in non-trivial ways in the non-associative case: solvability becomes polyabelianness for Boolean-completeness and CIRCUIT VALUE, and a combination of polyabelianness and $\mathcal{M}$-solvability for EXPRESSION EVALUATION. The table given in the Introduction becomes the following in the case of quasigroups or loops:

	EXPRESSION EVALUATION	CIRCUIT VALUE
non-polyabelian	$\mathbf{NC}^1$-complete	$\mathbf{P}$-complete
polyabelian but not $\mathcal{M}$-solvable	$\mathbf{NC}^1$-complete	$\mathbf{ACC}^1 \cap \mathbf{DET}$
polyabelian and $\mathcal{M}$-solvable (including nilpotent)	$\mathbf{TC}^0$	$\mathbf{ACC}^1 \cap \mathbf{DET}$

For algebras in general, non-polyabelianness needs to be replaced with some further generalization as the necessary and sufficient condition for Boolean-completeness. However, the second and third rows of this table hold for all algebras.

These results also have a language-theoretic interpretation. A regular language can be characterized by its *syntactic monoid*, the semigroup of allowed transitions of its finite-state machine. Thus the regular languages that are $\mathbf{NC}^1$-complete are exactly those whose syntactic monoid is non-solvable. Similarly, the set of expressions in a non-associative algebra that evaluate to a particular element is a structured context-free language, generated by the productions $a \to (bc)$ for all b, c such that $b \cdot c = a$. Thus it seems that we may be close to showing exactly which (structured) context-free languages are $\mathbf{NC}^1$-complete.

Overall, the fact that algebras with differing properties have EXPRESSION EVALUATION and CIRCUIT VALUE problems with (probably) differing circuit complexities may help us learn more about the internal structure of $\mathbf{NC}$, and hopefully make some progress towards proving that $\mathbf{ACC}^k$, $\mathbf{TC}^k$, and $\mathbf{NC}^k$ form rich, distinct hierarchies within $\mathbf{P}$, rather than all being equal to it.

Acknowledgements. J.B. assisted with this work as an intern at the Santa Fe Institute, funded by the Research Experience for Undergraduates program of the National Science Foundation. A.D. is grateful to the Santa Fe Institute for an enjoyable visit. C.M. is grateful to McGill University for their hospitality, and to Claire Riley for spicing the visit up a bit.

References

1. M. Ajtai, "Σ_1^1 formulae on finite structures." *Ann. Pure Appl. Logic* **24** (1983) 1–48.
2. D. A. Barrington, "Bounded-width polynomial-size branching programs recognize exactly those languages in $\mathbf{NC}^1$." *J. Comput. System Sci.* **38** (1989) 150–164.
3. D. A. Mix Barrington and J. Corbett, "On the relative complexity of some languages in $\mathbf{NC}^1$." *Info. Proc. Lett.* **32** (1989) 251–256.
4. M. Beaudry, P. McKenzie, P. Péladeau, and D. Thérien, "Circuits with monoidal gates." Springer LNCS, *Proc. STACS* (1993) 555–565.

5. R.H. Bruck, "Contributions to the theory of loops." *Trans. Amer. Math. Soc.* **60** (1946) 245–354.
6. S.R. Buss, "The Boolean formula value problem is in ALOGTIME." In *Proc. 18th ACM Symp. on the Theory of Computing* (1987) 123–131.
7. O. Chein, H.O. Pflugfelder, and J.D.H. Smith (eds.), *Quasigroups and Loops: Theory and Applications.* Heldermann Verlag, 1990.
8. M. Furst, J.B. Saxe, and M. Sipser, "Parity, circuits, and the polynomial-time hierarchy." *Math. Syst. Theory* **17** (1984) 13–27.
9. R. Greenlaw, H.J. Hoover and W.L. Ruzzo, *Limits to Parallel Computation:* **P**-*Completeness Theory.* Oxford University Press, 1995.
10. M. Hall, *The Theory of Groups.* Chelsea, 1976.
11. O.H. Ibarra, T. Jiang and B. Ravikumar, "Some subclasses of context-free languages in $\mathbf{NC}^1$." *Info. Proc. Lett.* **29** (1988) 111-117.
12. R.E. Ladner, "The circuit value problem is **LOGSPACE**complete for **P**." *SIGACT News* **7** (1975) 18-20.
13. F. Lemieux, *Finite Groupoids and their Applications to Computational Complexity.* Ph. D. Thesis, School of Computer Science, McGill University, Montréal (1996).
14. A. Maciel and D. Thérien, "Threshold circuits of small majority-depth." Technical Report SOCS–95.5, School of Computer Science, McGill University, Montréal (1995).
15. W.D. Maurer and J.L. Rhodes, "A property of finite simple non-Abelian groups." *Proc. Am. Math. Soc.* **16** (1965) 552–554.
16. R. McKenzie, "On minimal, locally finite varieties with permuting congruence relations." Preprint (1976).
17. C. Moore, "Non-Abelian cellular automata." Santa Fe Institute Working Paper 95-09-81, or "Predicting non-linear cellular automata quickly by decomposing them into linear ones," submitted to *Physica* **D**.
18. C. Moore, "Quasi-linear cellular automata." Santa Fe Institute Working Paper 95-09-078, to appear in *Physica* **D**, *Proceedings of the International Workshop on Lattice Dynamics.*
19. C.H. Papadimitriou, *Computational Complexity.* Addison-Wesley, 1994.
20. H.O. Pflugfelder, *Quasigroups and Loops: An Introduction.* Heldermann Verlag, 1990.
21. A.A. Razborov, "Lower bounds for the size of circuits of bounded depth with basis $\{\&, \oplus\}$." *Math. Notes Acad. Sci. USSR* **41(4)** (1987) 333–338.
22. I.G. Rosenberg, "Completeness properties of multiple–valued logic algebras." In D.C. Rhine, ed., *Computer Science and Multiple-Valued Logic: Theory and Application.* North-Holland, 1977, 144–186
23. R. Smolensky, "Algebraic methods in the theory of lower bounds for Boolean circuit complexity." *Proc. 19th ACM Symposium on the Theory of Computing* (1987) 77–82.
24. M. Suzuki, *Group Theory I.* Springer-Verlag, 1982.
25. A. Szendrei, *Clones in Universal Algebra.* Les Presses de L'Université de Montréal, 1986.
26. A. Vesanen, "Solvable groups and loops." *Journal of Algebra* **180** (1996) 862–876.
27. M. Weinstein, *Examples of Groups.* Polygonal Publishing House, Passaic, New Jersey, 1977.