Journal of Computer and System Sciences **60**, 368–394 (2000)
doi:10.1006/jcss.1999.1673, available online at http://www.idealibrary.com on IDEAL®

Circuits and Expressions with Nonassociative Gates

Cristopher Moore

The Santa Fe Institute, Santa Fe, New Mexico

Denis Thérien

McGill University, Montréal, Québec, Canada

François Lemieux

Université du Québec à Chicoutimi, Québec, Canada

Joshua Berman

State University of New York at Binghamton, New York

and

Arthur Drisko

National Security Agency, Washington, DC

We consider circuits and expressions whose gates carry out multiplication in a nonassociative groupoid such as a quasigroup or loop. We define a class we call the *polyabelian* groupoids, 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 Circuits Value problems are $\mathbf{NC}^1$-complete and **P**-complete, respectively. This is not true for groupoids in general, and we give a counterexample. We show that Expression Evaluation is also $\mathbf{NC}^1$-complete if the groupoid has a nonsolvable multiplication group or semigroup, but is in $\mathbf{TC}^0$ if the groupoid both is polyabelian and has a solvable multiplication semigroup, e.g., for a nilpotent loop or group. Interestingly, in the nonassociative case, the criteria for making Circuit Value **P**-complete and for making Expression Evaluation $\mathbf{NC}^1$-complete—nonpolyabelianness and nonsolvability of the multiplication group—are different. Thus, earlier results about the role of solvability in complexity generalize in several different ways.

0022-0000/00 $35.00

1. INTRODUCTION: ALGEBRAIC CIRCUITS AND EXPRESSIONS

Boolean expressions and circuits are well-known constructs in logic and computer science. A Boolean expression ϕ either is 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 uniform[1] 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 [14, 24], 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, where $\theta_t(x_1, ..., x_n) =$ TRUE iff t or more of the inputs are TRUE;

— $\mathbf{ACC}^k[p]$ if the gates are AND's, OR's, and MOD_p with unbounded fan-in, where $\text{MOD}_p(x_1, ..., x_n) =$ TRUE iff the number of true inputs is not a multiple of p;

— $\mathbf{ACC}^k = \bigcup_p \mathbf{ACC}^k[p]$.

The union of any of these families over all k is $\mathbf{NC}$, the class of problems solvable by parallel circuits of polylogarithmic depth and polynomial width. This is equivalent to polylogarithmic time on an idealized parallel computer with shared memory, constant communication delays, and a polynomial number of processors.

Since combinations of threshold gates can compute any function that depends only on the sum of the inputs, including MOD_p, and since any threshold gate can be realized in $\mathbf{NC}^1$, 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 will be interested in are

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

[1] For the most part, we will disregard uniformity questions. Interested readers are referred to [28, 5].

It is believed, but not known, that $\mathbf{P} \neq \mathbf{NC}$: in other words, that there are *inherently sequential* problems in **P** that cannot be efficiently parallelized. Some small progress has been made toward proving this [1, 13, 26, 29]: 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 **P** (or even **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 **P** it is common to use **LOGSPACE** ($\mathcal{O}(\log n)$ memory in a Turing machine) or $\mathbf{NC}^1$ reductions; for $\mathbf{NC}^1$ we will use $\mathbf{NC}^0$ reductions, this being essentially local replacement rules.

We also have

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

where **DET** is the class of problems log-space reducible to calculating the determinant of an integer matrix. **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 **P**-complete, respectively, under $\mathbf{NC}^0$ and LOGSPACE reductions [10, 17].

We will consider circuits and expressions where the sole operation is multiplication in some finite groupoid $(A, \cdot)$, rather than the usual Boolean operations. Thus, our circuits have one kind of node whose output is the product $a \cdot b$ of its two inputs, and our expressions are strings like $(x_1 \cdot x_2) \cdot (x_2 \cdot x_3)$. For technical reasons, we allow the presence of blanks in the encoding of an expression: this will be useful for getting weak reductions from problems in $\mathbf{NC}^1$ to the expression evaluation problem.

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 [3, 7, 22].

	Expression evaluation	Circuit value
Nonsolvable	$\mathbf{NC}^1$-complete	**P**-complete
Solvable	$\mathbf{ACC}^0$	$\mathbf{ACC}^1 \cap \mathbf{DET}$

In this paper, we will extend these results to nonassociative groupoids such as quasigroups and loops and to some extent to groupoids in general. We will show that the idea of solvability generalizes in two important ways in the nonassociative case, *polyabelianness*, the property of being an iterated affine quasidirect product of

Abelian groups, and *$\mathscr{M}$-solvability*, the property of having a solvable multiplication group or semigroup.

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 groupoids, as in [22, 23].

The paper is structured as follows. In Section 2 we give an introduction to the algebraic terms and concepts we will use. In Section 3 we define Boolean-completeness, the ability to express arbitrary Boolean functions as circuits or expressions. We review existing results on solvability in groups and loops and show that in the nonassociative case, even solvable loops and quasigroups can be Boolean-complete. Section 4 introduces the notion of polyabelianness, which coincides with solvability in the case of groups. We show that for quasigroups the Boolean-complete algebras are precisely the nonpolyabelian ones. In Section 5 we discuss $\mathscr{M}$-solvability, the property of having a solvable multiplication semigroup, and show that the non-$\mathscr{M}$-solvable groupoids are precisely those for which Expression Evaluation is $\mathbf{NC}^1$-complete. Finally, in Section 6 we give our conclusions and suggest directions for further work.

2. ALGEBRAIC PRELIMINARIES

For the theory of quasigroups and loops, the reader is referred to [2, 8, 9, 25]. We will use the following terms.

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

A *quasigroup* is a groupoid 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, where $bc = bd$ (resp. $cd = bd$) implies $c = d$.

An *identity* is an element 1 such that $1 \cdot a = a \cdot 1$ for all a. A *loop* is a quasigroup with an identity.

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. $a \cdot a^\rho = 1$). A loop has the left (right) *inverse property* if $a\backslash b = a^\lambda \cdot b$ (resp. $b/a = b \cdot a^\rho$). If a loop has both the left and the right inverse property, it has the *inverse property* and $a^\lambda = a^\rho$, in which case we will refer to them both as a^{-1}.

A groupoid is *associative* if $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ for all a, b, c. A *semigroup* is an associative groupoid, 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 $a \cdot b = b \cdot a$. A groupoid 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 and call the identity 0 instead of 1.

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

A *homomorphism* is a function ϕ from one groupoid $(A, \cdot)$ 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 a groupoid into itself are called *endomorphisms* and *automorphisms*, respectively; the automorphisms of a groupoid A form a group $\mathrm{Aut}(A)$.

A *subgroupoid* (*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 subgroupoid 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 $a, b \in G$:

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

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 and which is not $\{1\}$. A *simple* loop is one with no proper normal subloops.

The *commutator* of two elements in a loop is $[a, b] = ab/ba$, i.e., the unique element such that $ab = [a, b](ba)$. The *associator* of three elements is $[a, b, c] = (ab)\,c/a(bc)$, i.e, the unique element such that $(ab)\,c = [a, b, c](a(bc))$. 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. A groupoid is solvable if it has no subsets which are nonsolvable groups under the groupoid's operation.

A *divisor* of a groupoid is a factor of a subgroupoid. Any nonsolvable loop has a simple non-Abelian divisor. A divisor is not necessarily a subgroupoid, even for groups.

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 and is always an Abelian group.

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 if $G/Z(G)$ has a nontrivial center, and so on until we get an Abelian group H for which $Z(H) = H$. The nilpotent loops are a proper subclass of the solvable ones.

A *pseudovariety* is a class of groupoids V such that subgroups, factors, and finite direct products of groupoids in V are also in V. 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 groupoid 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* $\mathbb{Z}_p = \{0, 1, ..., p-1\}$ 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 a groupoid A and whose inputs can be variables or constant elements of A. This is equivalent to the set of expressions definable in A with constants and variables, such as $\phi(x_1, x_2, x_3) = (x_1(ax_2))(x_3 b)$, regardless of their size.

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

— (constants) a for all $a \in A$.

— (projections) x_i for all i.

— (products) $(\phi_1 \cdot \phi_2)$ for all $\phi_1, \phi_2 \in \mathcal{P}(A)$.

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 $\phi(x) = xx$. (Since $\mathcal{P}(A)$ is closed under composition and substitution of one function for a variable of another, it is a *clone* in the nomenclature of [31].)

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

★	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

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

are expressions of these Boolean functions as expressions in ★. We can combine these to make any other Boolean function. Formally:

DEFINITION. A groupoid $(A, \star)$ is *strongly Boolean-complete* if there exist elements TRUE and FALSE in A and functions $\phi_\wedge$, $\phi_\neg$ in $\mathscr{P}(A)$, 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. A groupoid $(A, \star)$ 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 $\mathscr{P}(A)$ 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 groupoid 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. *The set of non-Boolean-complete groupoids forms a pseudovariety. Therefore, if a divisor of a groupoid G is Boolean-complete, then G is also.*

Proof. If a subgroupoid $H \subset G$ is (strongly) Boolean-complete, then G is also since $\mathscr{P}(H) \subset \mathscr{P}(G)$. 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 $\phi^{-1}(F)$. This shows that non-Boolean-complete groupoids are closed under division. It remains to prove that finite direct products of non-Boolean-complete groupoids are non-Boolean-complete.

Let G and H be two non-Boolean-complete groupoids and suppose that $G \times H$ is Boolean-complete. Let T and F be two subsets of $G \times H$ containing true and false values. Since $G \times H$ is Boolean-complete but G and H are not, there must exist elements $a, b \in G$ and $c, d \in H$ such that either $(a, c) \in T$ and $(a, d) \in F$ or $(a, c) \in T$ and $(b, c) \in F$. Assume the first case, the other one being symmetric.

Let $f(x, y) \in \mathscr{P}^2(G \times H)$ represent the function that computes NAND(x, y). By fixing the first component of x and y to a, we get that the first component of $f(x, y)$ is fixed to a_0 for some $a_0 \in G$. Hence, we only have to look at the second component to determine if f evaluates to TRUE or FALSE. Observe that we do not have a contradiction yet, since we can have a situation where $(a, c) \in T$ and $(a_0, c) \in F$.

Let $g_1(x, y) = f(x, y)$ and, for any $k \geqslant 1$, define

$$g_{k+1}(x, y) = g_k(f(f(x, x), f(x, x)), f(f(y, y), f(y, y))).$$

Then, for any $k \geqslant 1$, $g_k(x, y)$ computes NAND(x, y), since $f(f(x, x), f(x, x))$ has the same truth value as x.

If the first component of x and y is a_0, then we can define a_i as the first component of $g_i(x, y)$. Since G and H are both finite, there must exist two integers $0 \leqslant i < j$ such that $a_i = a_j$. Hence, if we use only true and false values whose first component is a_i, then the first component of g_{j-i} is also a_i. Let $S = \{(a_i, c) \mid c \in H\}$. We have that the sets $T' = S \cap T$ and $F' = S \cap F$ are disjoint, and so, H is Boolean-complete, a contradiction. ∎

Then our fundamental motivation for this work is the following:

LEMMA 2. *If a groupoid is Boolean-complete, then its Expression, Evaluation and Circuit Value problems are* $\mathbf{NC}^1$*-complete and* $\mathbf{P}$*-complete, under* $\mathbf{NC}^0$ *and* $\mathbf{NC}^1$ *reductions respectively.*

Proof. The Boolean Circuit Value problem is reducible to its algebraic counterparts, since a local rule can replace AND, OR, and NOT gates with "gadgets" of algebraic gates.

The case of Expression Evaluation is slightly more complicated. Actually we can show that it is complete for $\mathbf{NC}^1$ under DLOGTIME-uniform projection (see [5]).

We observe first that the circuits involved in the definition of $\mathbf{NC}^1$ can be restricted to be DLOGTIME-uniform full binary trees with 2^k input gates (for some k depending on the length of the input) and with AND–OR gates alternating between each level (see [5]). Moreover, we can replace all AND and OR gates with NAND gates without loss of generality by using the equalities

$$\text{AND}(x, y) = \text{NAND}(\text{NAND}(x, y), \text{NAND}(x, y))$$

$$\text{OR}(s, y) = \text{NAND}(\text{NAND}(x, x), \text{NAND}(y, y)).$$

Let $L \in \mathbf{NC}^1$ be a language recognized by such a family of circuits. Given a binary string w, the reduction from w to the string v corresponding to the 2^k input gates of the circuit can be done in DLOGTIME since the circuits are DLOGTIME uniform. Hence, $w \in L$ if and only if v can be evaluated through a full tree using only NAND gates.

So, we only have to show that evaluating v using a full tree of NAND gates is equivalent to evaluating an expression ϕ over a Boolean complete groupoid and that each symbol in ϕ can be determined in DLOGTIME.

Let G be a Boolean complete groupoid. Then there exists an expression $f \in \mathscr{P}^2(G)$ that represents the NAND function. Without loss of generality we can assume that the number M of symbols in f is a power of 2 (pad it with blanks if necessary). Let $f^{(i)}$ be the expression obtained from f by replacing each symbol s, that is not a variable, with $sB^{M^{i-1}-1}$, where B is the blank symbol.

The rest of the idea is similar to that in [6]. We construct an expression for each gate in the circuit so that ϕ is the expression corresponding to the output gate. A gate g on level 1 whose inputs are x and y is represented by the expression $\phi_g = f(x, y)$. A gate g on level $i > 1$ whose inputs are g_1 and g_2 (which are on level $i-1$) is represented by the expression $\phi_g = f^{(i)}(\phi_{g_1}, \phi_{g_2})$ whose length is M^i. For any j, the jth symbol of ϕ can be determined simply by looking at the binary expansion of j. Whenever this symbol is a variable, we use the uniformity of the circuits to determine its position in the input. ∎

Then we briefly restate a theorem of Barrington [3], 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 $\mathscr{P}^2(Q)$ as functions of a and b. Therefore, when Q is a loop, functions that yield the commutator, associator, and left and right inverses are in $\mathscr{P}$ 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 **1**. Then $aL_a^{n!-1}(b) = L_a^{n!}(b) = b$, so

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

is in $\mathscr{P}(Q)$. 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 $a\backslash 1$. ▮

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 $\mathscr{P}^1(G)$ that sends x to y and keeps the identity fixed.*

Proof. Let U be the set of functions ϕ in $\mathscr{P}^1(G)$ 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.

It is easy to see that U is closed under product, since if ϕ_1 and ϕ_2 are in U and we define $\phi' = \phi_1 \cdot \phi_2$, then $\phi'(1) = \phi_1(1) \cdot \phi_2(1) = 1 \cdot 1 = 1$ and ϕ' is in U also. For the same reason, $U(x)$ is a subloop, since $\phi'(x) = \phi_1(x) \cdot \phi_2(x)$ is in $U(x)$ whenever $\phi_1(x)$ and $\phi_2(x)$ are. Moreover, $U(x)$ is a normal subloop, since for any ϕ in U the following ϕ' is also in U:

$$aU(x) = U(x)\, a : \phi'(x) = a\phi(x)/a$$

$$a(bU(x)) = (ab)\, U(x) : \phi'(x) = (ab)\backslash a(b\phi(x))$$

$$a(U(x)\, b) = (aU(x))\, b : \phi'(x) = a\backslash(a(\phi(x)\, b)/b).$$

In each case ϕ' is in $\mathscr{P}(G)$ 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 $\pi_{x \to y}$. ▮

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

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

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 nonsolvable groups have simple non-Abelian divisors, nonsolvable groups are Boolean-complete by Lemma 1. ▮

In fact, [3] seems to state that nonsolvable groups are strongly Boolean-complete; we believe this is a slight mistake. A counterexample would be a group such as SL(2, 5) whose simple divisor $\mathrm{PSL}(2,5) \cong A_5$ is not a subgroup [33] (recall that a divisor is not generally a subgroup). In this case, PSL(2, 5) is strongly Boolean-complete, while SL(2, 5) is not.

By using an associator instead of a commutator, this generalizes to loops [11, 18]. 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 nonsolvable loops are Boolean-complete.*

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

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

and let (say) $\neg a = t/a$ or $t \cdot a^{\rho}$ (note that these are not the same for all a unless the loop has the right inverse property). So simple non-Abelian loops are strongly Boolean-complete; and since nonsolvable loops have simple non-Abelian divisors, they are Boolean-complete by Lemma 1.

In fact, simple non-Abelian groups [20], simple loops [11, 18], and nonaffine simple quasigroups [21] have a stronger property, that their closure contains *all* possible n-ary functions on their elements. This is called *functional completeness* and is of interest in the field of multivalued logic [27, 31]. However, Boolean-completeness is sufficient for our purposes.

Since the Circuit Value problem for solvable groups is in **NC** [7, 22], nonsolvability is both necessary and sufficient for Boolean-completeness in the case of groups (or semigroups, in fact).

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 $\mathscr{P}(G)$ 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.

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 GROUPOIDS

4.1. Definition and Properties

The *direct product* of two groupoids $A \times B$ 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* [12] of two groupoids 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 $A \otimes B$.

Observe that the quasidirect product, as defined above, makes no use of the product in B and so it makes sense to talk of the quasidirect product of A and S even when S is a set with no underlying operation. In order to take into account the algebraic structure of B, we have to restrict the local operations.

If the local operations are of the form

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

where f and g are functions from B to B, 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 7.1. 1. *If a groupoid G is a quasidirect product $A \otimes B$, then A is a factor of G.*

2. *If G is a quasigroup, then A and B are quasigroups and all the $\odot$'s are quasigroup operations on B.*

3. *If G is a quasigroup and is affine on B, then all the f's and g's are automorphisms on B.*

4. *If G is a loop, then A is a loop and $B \cong \{1\} \times B$ is a normal subloop (where 1 is the identity of A).*

5. *If G is a loop affine on $(B, +)$, then for all $a \in A$ we have $f_{a,1} = g_{1,a} = \mathbf{1}$ and $h_{a,1} = h_{1,a} = 0$ (where we use 1 and 0 for the identities of A and B, respectively). Thus $b_1 \odot_{1,1} b_2 = b_1 + b_2$ for $b_1, b_2 \in B$, and $+$ and $\cdot$ coincide in B.*

Proof. For the most part, we leave this to the reader. For (5), the identity of G must be $(1, b)$ where 1 is the identity of A and b is some element of B. Without loss of generality, we can assume that $b = 0$, since otherwise we can redefine the operation $+$ by adding a constant. Then $(a, b) \cdot (1, 0) = (a, f_{a,1}(b) + h_{a,1})$ and $(1, 0) \cdot (a, b) = (a, g_{1,a}(b) + h_{1,a})$. Setting the B component of both of these equal to b and using the fact that f and g are endomorphisms yields $f_{a,1} = g_{1,a} = \mathbf{1}$ and $h_{a,1} = h_{1,a} = 0$. ∎

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 are expressible in $\mathscr{P}(G)$. If the local operations are separable, then the f's and g's are expressible, and if N is Abelian and G is affine on N, the f's, g's, and h's are expressible.*

Proof. The first part of the proof comes from [2]. Choose a set T with one element in each coset of N (such a set is often called a *transversal*), 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

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

which is in N since N is normal. Thus, G is a quasidirect product $T \otimes N$, and all the local operations $\odot$ are in $\mathscr{P}(G)$.

If the $\odot$'s are separable, then

$$f_{t_1, t_2}(n) = n \odot_{t_1, t_2} g^{-1}_{t_1, t_2}(1),$$

where 1 is the identity of N. Thus, f_{t_1, t_2} is expressible for each t_1 and t_2, and similarly for g_{t_1, t_2}.

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 $\mathscr{P}(G)$ since $a \odot_{0,0} b = a + b$ by Lemma 7. ▌

Then define the following class of groupoids.

DEFINITION. A groupoid 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 subgroupoids, factors, and finite direct products of polyabelian groupoids 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. ▌

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 nontrivial group in its derived series such that $N' = \{1\}$. Since factors of solvable groups are solvable, G/N is solvable if G is, so we can assume by induction on smaller groups that G/N is polyabelian. Now express G as a quasidirect product of G/N and N using Lemma 8, with the local operation

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

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 [30]. In [22] 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 nontrivial 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 nonassociative case; that is, a groupoid 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 groupoids in general.

4.2. Polyabelian Groupoids Are Not Boolean-complete

In one direction, we can prove this for all groupoids. In [22] we show that Circuit Value for polyabelian groupoids is in $\mathbf{ACC}^1$, and a simple modification of the proof for solvable semigroups in [7] shows that it is also in **DET**. We now

show directly that polyabelian groupoids cannot express the AND function. First, two lemmas from [31].

DEFINITION. Let A be an Abelian group. A function $\phi: A^n \to A$ is *affine* if there exist endomorphisms $f_1, ..., f_n$ and an element h such that $\phi(x_1, ..., x_n) = \sum_i f_i(x_i) + h$.

Recall the definition of the closure $\mathscr{P}(A)$ from Section 3. The closure of an Abelian group consists only of affine functions.

LEMMA 12. *If A is an Abelian group, then any function in $\mathscr{P}(A)$ 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 groupoids 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., $\phi(a_1, a_2, ..., a_n) \in T$ 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 $x \in G$ has a unique vector of components $(x_0, x_1, ..., x_k)$ where $x_i \in A_i$ for all i. Call x_i the *A_i-component* of x. 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.

Since A is finite, it has a finite number $k \leqslant |A|^{|A|}$ of endomorphism.[2] Therefore, if $\psi(a_1, ..., a_n) = \sum_i g_i(a_i) + h$ is an n-ary affine function on an Abelian group A of order p, and if n is greater than $(p-1)k$, then at least p of the variables have the same $g_i = g$. Then if these p variables are all equal, they contribute nothing to ψ since $pg = 0$. 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 $\mathscr{P}(G)$. To start the induction, since A_0 is a factor of G by Lemma 7, ϕ's A_0-component ϕ_0 is a function of the A_0-components of the a_i, expressible in $\mathscr{P}(A_0)$ 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 $j \leqslant m$. Think of ϕ as a tree where each node corresponds to a subexpression equal to the

[2] If $A = \mathbb{Z}_p^m$, for instance, $k = p^{m^2}$ since the endomorphisms of A are $m \times m$ matrices with entries in $\mathbb{Z}_p$.

product of its daughters according to some local product. Then the A_j-components at each node depend only on the $A_{j'}$-components of its two subexpressions for $j' \leqslant j$ (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 $j \leqslant m$, so inductively ϕ and each of its subexpressions have constant A_j-components for $j \leqslant m$ 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 $j \leqslant m$. Since these are constant in this restriction, each node always applies the same local operation; the composition of all of these makes 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 $j \leqslant m+1$. 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.

Thus, by contradiction, G cannot express an n-ary AND and is not Boolean-complete. ∎

4.3. *Nonpolyabelian Loops and Quasigroups Are Boolean-Complete*

Theorem 5 and Lemma 10 show that nonpolyabelianness 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. *Nonpolyabelian loops are Boolean-complete.*

Proof. Let H be the smallest nonpolyabelian divisor of G. We will show that H (which is also a loop) is strongly Boolean-complete.

Assume without loss of generality that H is solvable, since we have already treated the nonsolvable case with Theorem 6. Then H has a normal subloop K which is an Abelian group, namely the last nontrivial 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. Note that N can be smaller than K. We know that H is not affine on N; otherwise H/N would be a smaller nonpolyabelian 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 $\pi_{n_1 \to n_2} \in \mathscr{P}(H)$ 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 $K_{\odot} = 0$, then $n_1 \odot n_2 = f(n_1) + g(n_2)$ where $f(n_1) = (n_1 \odot 0)$ and $g(n_2) = (0 \odot n_2) - (0 \odot 0)$, so $\odot$ is separable. Conversely, if $\odot$ is separable, then all the terms cancel and $K_{\odot} = 0$. Therefore, 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

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

If all the local operations are separable, then one must not be affine; that is, some f or g is not an 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

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

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 in Theorem 6. Thus, H is strongly Boolean-complete, and G is Boolean-complete by Lemma 1. ∎

This proof fails for the quasigroup case, since there might be no Abelian subquasigroup N to define the separator or affinator over. Instead, we will use some results from clone theory and the study of multioperation algebras [31], after giving some new definitions.

DEFINITION. An *algebra* (A, S), or A for short, is a set A with a set of operations S of any arity. Its closure $\mathscr{P}(A)$ is the set of all functions that can be written with these operations and constants. For instance, if $S = \{f, \cdot, g\}$ where f, $\cdot$, and g are unary, binary, and ternary, respectively, then $\phi(x_1, x_2) = g(f(x_1 \cdot a), x_2, b)$ is in $\mathscr{P}(A)$.

An algebra is *functionally complete* if $\mathscr{P}(A)$ includes all possible functions, of whatever arity, on A. Clearly a functionally complete algebra with more than one element is also strongly Boolean-complete.

DEFINITION. An algebra (A, S) is *affine* if there is some Abelian group $+$ defined on A such that every operation in S is affine on $(A, +)$.

DEFINITION. A *congruence* on an algebra (A, S) is an equivalence $\sim$ on A such that, for every operation f in S of arity k, if $a_1 \sim b_1$, $a_2 \sim b_2$, ..., and $a_k \sim b_k$, then $f(a_1, ..., a_k) \sim f(b_1, ..., b_k)$.

In other words, the equivalence class of f depends only on the equivalence classes of the a_i. Then the map ϕ from A to the set of equivalence classes $A/\sim$ is a *homomorphism*, i.e., $\phi(f(a_1, ..., a_k)) = f(\phi(a_1), ..., \phi(a_k))$ for all f in S. This generalizes the idea of normality; if A is a loop with a normal subloop N, then the equivalence classes are simply N's cosets.

A congruence $\sim$ is *proper* if it is other than the identity ($a \sim b$ only if $a = b$) or all of A($a \sim b$ for a, b). It is *minimal* if there is no proper congruence whose

equivalence classes are properly contained in those of $\sim$. An algebra is *simple* if it has no proper congruences.

DEFINITION. An algebra A is *Mal'cev* if $\mathscr{P}(A)$ contains a *Mal'cev operation*, a ternary function ϕ such that $\phi(x, y, y) = \phi(y, y, x) = x$ for all $x, y \in A$.

Any algebra with a quasigroup operation is Mal'cev, since if we define $\phi(x, y, z) = (x/x)\backslash((x/y)\, z)$, then $\phi(x, y, y) = (x/x)\backslash x = (x/x)\backslash((x/x)\, x) = x$ and $\phi(y, y, x) = (y/y)\backslash((y/y)\, x) = x$.

Lemma 1 clearly generalizes to algebras in general; if $A/\sim$ is Boolean-complete, then so is A. In addition, Lemma 8 generalizes to quasigroups: if Q has a congruence $\sim$, then $Q \cong (Q/\sim) \otimes N$ where N is any one of $\sim$'s equivalence classes and the local operations on N are in $\mathscr{P}(Q)$. (However, unlike in the loop case, it is possible that none of $\sim$'s equivalence classes are subquasigroups.)

Finally, we import the following corollary to Theorem 4.7 and Corollary 4.12 of [31].

LEMMA 16. *If an algebra is Mal'cev, simple, and nonaffine, then it is functionally complete.*

This allows us to reach our goal, namely

THEOREM 17. *Nonpolyabelian quasigroups are Boolean-complete.*

Proof. Let Q be nonpolyabelian, and let R be its smallest nonpolyabelian divisor. As before, we will show R is strongly Boolean-complete.

Lemma 16 takes care of the case when R is simple, since simple and nonpolyabelian implies nonaffine. Thus, we can assume that R has proper congruences; let $\sim$ be its minimal congruence and N one of its equivalence classes. We now consider the multioperation algebra $(N, \{\odot\})$ consisting of all the local operations on N.

If $(N, \{\odot\})$ is affine on the equivalence classes of $\sim$, there is an Abelian group $(N, +)$ on which all the local operations $\odot$ are affine. Then the quasidirect product $(R/\sim) \otimes N$ is affine, and $R/\sim$ is a smaller nonpolyabelian divisor of Q. So $(N, \{\odot\})$ is nonaffine.

Similarly, any proper congruence of $(N, \{\odot\})$ is also a proper congruence of each of the quasigroups $(N, \odot)$ dividing the equivalence classes of $\sim$ into smaller ones. Since $\sim$ is R's minimal congruence, $(N, \{\odot\})$ is simple.

Then N is functionally complete by Lemma 16, R is strongly Boolean-complete since $\mathscr{P}(N) \subset \mathscr{P}(R)$ by Lemma 8, and A is Boolean-complete by Lemma 1. ∎

While this proof is quite powerful and includes Theorems 5, 6, and 15 as special cases, it is considerably less constructive. To actually build an AND gate, one can use Theorem 2.4 of [31], a kind of converse to Lemmas 12 and 13:

LEMMA 18. *If A is a nonaffine Mal'cev algebra, then $\mathscr{P}(A)$ contains a function ϕ such that, for some $a_1, a_2, b_1, b_2 \in A$, $\phi(a_1, b_1) = \phi(a_1, b_2)$ but $\phi(a_2, b_1) \neq \phi(a_2, b_2)$.*

In essence, this shows that a nonlinear function always exists if a Mal'cev algebra is not affine. We also need the following generalization of Lemma 4.

LEMMA 19. *If A is a simple algebra, 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 $\mathscr{P}(A)$ 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 $\mathscr{P}(A)$ 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 functions $\phi_1,\ \phi_2 \in \mathscr{P}(A)$ 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 $y_1 \cdot y_2 \in U_{z \to w_1 \cdot w_2}(x)$.

Fix x and z. Then the sets $U_{z \to w}(x)$ for different w are equivalence classes of a 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 A is simple, it has no proper congruences, and $U_{z \to w}(x) = A$; so for every y, there exists a function $\phi \in U_{z \to w}(x)$ that sends x to y, which is our $\pi_{x \to y,\, z \to w}$. ∎

Since $(N, \{\odot\})$ is simple, we can construct an AND gate by choosing an element $x \in N$ and defining

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

where ϕ is the function of Lemma 18. 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, all except $\psi(a_2, b_2)$ which is some $y \neq x$. This has the shape of an AND gate, and using Lemma 19 we can define

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

In any case, we have shown the following:

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

4.4. Groupoids in General

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

$\bigstar$	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 nonpolyabelian, but cannot express the* AND *function.*

Proof. The equivalence classes $\{0, 1\}$ and $\{2, 3\}$ form the only proper 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 TRUE, FALSE $\in \{0, 1\}$.

The lower right-hand block looks like an AND gate with TRUE $= 1$ and 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 an expression ϕ in $\mathscr{P}(A)$. 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 subgroupoid isomorphic to $\mathbb{Z}_2$. So any function in $\mathscr{P}(A)$ is constant or affine when restricted to variables in $\{0, 1\}$ and cannot express an AND.

Finally, assume that TRUE, 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 $\mathscr{P}(A)$ 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. ∎

Note that we have not shown that this groupoid's Expression Evaluation or Circuit Value problem is in **NC**, but simply that it is not Boolean-complete. Perhaps the reader can find some more subtle analog of polyabelianness that works for all groupoids.

5. $\mathscr{M}$-SOLVABILITY AND EXPRESSION EVALUATION

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

DEFINITION. A groupoid is *$\mathscr{M}$-solvable* if its multiplication semigroup is solvable.

For groups, solvability and $\mathscr{M}$-solvability coincide since one can show that $\mathscr{M}(G)$ is isomorphic to $(G \times G)/Z(G)$ and is solvable if G is. For loops, $\mathscr{M}$-solvability implies solvability by a recent result of Vesanen [32], and nilpotent loops are $\mathscr{M}$-solvable by a theorem of Bruck [8]. Thus, like polyabelianness, $\mathscr{M}$-solvability is a generalization of solvability which lies properly between nilpotence and solvability for loops.

However, the $\mathscr{M}$-solvable loops and polyabelian loops are incomparable. First we show that $\mathscr{M}$-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; these are automorphisms by (3) in Lemma 7. 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 [15].

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

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

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 the following.

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

— *G is $\mathscr{M}$-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 $\mathscr{M}(G)$ consisting of those multiplications that preserve each block $\{a\} \times B$, i.e., that leave the A-component unchanged; in a loop, for instance, this includes multiplications by elements of B. Choose an element $a_0 \in A$, and let N be the subgroup of H that also fixes the B-components of elements in $\{a_0\} \times B$. 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 subgroupoid, we can identify b with (a_0, b) and define the local operations as $b_1 \odot b_2 = a \backslash ((a_1 b_1)(a_2 b_2))$ where a is chosen so that the A-component of $b_1 \odot b_2$ is a_0; since A is a factor of G by Lemma 7, a depends only on a_1 and a_2.

Then multiplications in $\odot$ are compositions of multiplications in G, since $L_b^{\odot} = L_a^{-1} L_{(a_1 b)} L_{a_2}$ and $R_b^{\odot} = L_a^{-1} R_{(a_2 b)} L_{a_1}$. So $\mathscr{M}(B, \odot) \subseteq H/N$.

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

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

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 nonsolvable automorphism group; for instance, if $B = \mathbb{Z}_2^3$, its automorphisms can be thought of as invertible 3×3 matrices over $\mathbb{Z}_2$. The group of these, $\mathrm{Aut}(\mathbb{Z}_2^3)$, is the simple group of order 168, and it is generated by two elements [33],

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

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 nonsolvable 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 S 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 groupoid G, the Expression Evaluation problem of $\mathcal{M}(G)$ is $\mathbf{NC}^0$-reducible to that of G.*

Proof. Since $\mathcal{M}(G)$ is the closure of $S = \{L_a, R_a \mid a \in G\}$, there exists an integer $k > 0$ such that any element $P \in \mathcal{M}(G)$ can be expressed as $P = D_1 \cdots D_k$, where

$D_i \in S \cup \{B\}$ and B is the blank symbol. Clearly, words over $\mathscr{M}(G)$ can be reduced to equivalent words over $S \cup \{B\}$ using $\mathbf{NC}^0$ circuits.

It remains to show how to reduce words over $S \cup \{B\}$ to expressions over G. Let w be a word over $S \cup \{B\}$ and let X be a variable. For each prefix v of w we construct an expression $\phi_v[X]$. If v is the empty word then $\phi_v[X] = X$. Otherwise, we can write $v = uD$, where $D \in S \cup \{B\}$ and we define $\phi_v[X] = (D_1(\phi_u[X]\ D_2))$, where D_1 (resp. D_2) is a if D is L_a (resp. R_a) for some $a \in G$ and B otherwise. Then, evaluating w can be done by evaluating the expressions $\phi_w[g]$, for all $g \in G$. Since G is finite, this is easily seen to be an $\mathbf{NC}^0$ reduction. ▮

Then we immediately have the following.

THEOREM 24. *The Expression Evaluation problem for non-$\mathscr{M}$-solvable groupoids is $\mathbf{NC}^1$-complete under $\mathbf{NC}^0$ reductions.*

Proof. By Lemma 23 and the fact that Expression Evaluation for nonsolvable semigroups is $\mathbf{NC}^1$-complete under $\mathbf{NC}^0$ reductions (actually, DLOGTIME-uniform projections) [3]. ▮

No analogue of Lemma 23 seems to exist in the case of circuits; since Circuit Value is $\mathbf{P}$-complete for nonsolvable groupoids but in $\mathbf{ACC}^1$ for non-$\mathscr{M}$-solvable ones that are polyabelian, circuits over $\mathscr{M}(G)$ 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 [3]. In this case, the correct generalization of solvability in the nonassociative case consists of being *both* polyabelian and $\mathscr{M}$-solvable, but the need to convert the expression into a tree of subexpressions makes the problem somewhat harder. Define the *subexpression tree* of an expression as a tree with one node for each subexpression, whose daughters are its subsubexpressions and which is labeled with the left–right path that reaches it from the root. Then,

THEOREM 25. *For a groupoid which is both polyabelian and $\mathscr{M}$-solvable, Expression Evaluation is in $\mathbf{TC}^0$ or in $\mathbf{ACC}^0$ if its subexpression tree is already known.*

Proof. Consider a polyabelian groupoid $A = (A_0 \otimes A_1) \otimes \cdots \otimes A_k$. An expression such as $\phi = (x_1 \cdot (x_2 \cdot x_3)) \cdot x_4$ on variables $x_1, ..., x_n$ can be inductively calculated in the following way (similar to the algorithm in [22] 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

$$F_1 = f_3 f_1 \qquad G_1 = f_3$$
$$F_2 = f_3 g_1 f_2 \qquad G_2 = f_3 g_1$$
$$F_3 = f_3 g_1 g_2 \qquad G_3 = \mathbf{1}$$
$$F_4 = g_3$$

Each one of these is a word in $\langle fg(A_{m+1})\rangle$; by Lemma 21 this is solvable if A is $\mathscr{M}$-solvable. Since Expression Evaluation is in $\mathbf{ACC}^0$ for solvable groups [3], 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 are done.

However, how the f's and g's contribute to the F's and G's depends on the subexpression 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. Thus, the left–right paths in the subexpression tree determine the F's and G's.

The set of well-formed expressions is a *structured context-free language* [16]. Threshold circuits of constant depth can count the *ascent* of a string in the parse tree [4], 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 convert the expression into its subexpression tree in $\mathbf{TC}^0$, and then compute the F's and G's, and so ϕ, by calculating the above sum with k levels of $\mathbf{ACC}^0$ circuits.

(Since we only need to find the subexpression tree once, the *majority-depth* of the circuit, defined as the maximum number of threshold gates that any path traverses, is constant for all groupoids. Thus, this problem is in $\widehat{\mathbf{TC}}^0_k$ for some small k as defined in [19].) ▮

Since nilpotent loops are both polyabelian (by Lemma 11) and $\mathscr{M}$-solvable [8], we have the following corollary.

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

Cases where the subexpression 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 [22], 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 nontrivial ways in the nonassociative case: solvability becomes

polyabelianness for Boolean-completeness and Circuit Value and a combination of polyabelianness and $\mathscr{M}$-solvability for Expression Evaluation. The table given in the Introduction becomes the following in the case of quasigroups or loops:

	Expression evaluation	Circuit value
Nonpolyabelian	$\mathbf{NC}^1$-complete	$\mathbf{P}$-complete
Polyabelian but not $\mathscr{M}$-solvable	$\mathbf{NC}^1$-complete	$\mathbf{ACC}^1 \cap \mathbf{DET}$
Polyabelian and $\mathscr{M}$-solvable (including nilpotent)	$\mathbf{TC}^0$	$\mathbf{ACC}^1 \cap \mathbf{DET}$

For groupoids in general, nonpolyabelianness 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 groupoids. It is also interesting to note that our investigations provide examples of groupoids which are not Boolean-complete, while their Expression Evaluation is still $\mathbf{NC}^1$-complete; this may point to a potential source of difficulty in the question of separating $\mathbf{ACC}^0$ from $\mathbf{NC}^1$, since one may have intuitively believed that solvable monoids, which correspond to $\mathbf{ACC}^0$, could not possibly handle $\mathbf{NC}^1$ computations, since a solvable group cannot be Boolean-complete. The final answer will have to be more subtle than that.

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 nonsolvable. Similarly, the set of expressions in a nonassociative groupoid 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 groupoids 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 toward 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.

ACKNOWLEDGMENTS

We are thankful to Agnes Szendrei for helping us complete the proof of Theorem 17 and to an anonymous referee for improving Lemmas 1 and 8. D. T. and F. L. acknowledge support from NSERC and FCAR. 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. A. A. Albert, Quasigroups I, *Trans. Amer. Math. Soc.* **54** (1943), 507–519; Quasigroups II, *Trans. Amer. Math. Soc.* **55** (1944), 401–419.
3. 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.
4. D. A. Mix and J. Corbett, On the relative complexity of some languages in $\mathbf{NC}^1$, *Info. Proc. Lett.* **32** (1989), 251–256.
5. D. M. Barrington, N. Immerman, and H. Straubing, On uniformity within $\mathbf{NC}^1$, *J. Comput. System Sci.* **41** (1990), 274–306.
6. F. Bédard, F. Lemieux, and P. McKenzie, Extension to Barrington's M-program model, *Theoret. Comput. Sci.* **107** (1993), 31–61.
7. M. Beaudry, P. McKenzie, P. Péladeau, and D. Thérien, Circuits with monoidal gates, *in* "Proc. STACS, 1993," pp. 555–565, Springer-Verlag, Berlin.
8. R. H. Bruck, Contributions to the theory of loops, *Trans. Amer. Math. Soc.* **60** (1946), 245–354.
9. R. H. Bruck, "A Survey of Binary Systems," Springer-Verlag, Berlin/New York, 1966.
10. S. R. Buss, The Boolean formula value problem is in ALOGTIME, *in* "Proc. 18th ACM Symp. on the Theory of Computing, 1987," pp. 123–131.
11. H. Caussinus and F. Lemieux, The complexity of computing over quasigroups, *in* "Proc. 14th annual FST&TCS, 1994," pp. 36–47.
12. O. Chein, H. O. Pflugfelder, and J. D. H. Smith, (Eds.) "Quasigroups and Loops: Theory and Applications," Heldermann-Verlag, Berlin, 1990.
13. M. Furst, J. B. Saxe, and M. Sipser, Parity, circuits, and the polynomial-time hierarchy, *Math. System Theory* **17** (1984), 13–27.
14. R. Greenlaw, H. J. Hoover, and W. L. Ruzzo, "Limits to Parallel Computation: **P**-Completeness Theory," Oxford University Press, Oxford, 1995.
15. M. Hall, "The Theory of Groups," Chelsea, New York, 1976.
16. O. H. Ibarra, T. Jiang, and B. Ravikumar, Some subclasses of context-free languages in $\mathbf{NC}^1$, *Info. Process. Lett.* **29** (1988), 111–117.
17. R. E. Ladner, The circuit value problem is **LOGSPACE**-complete for **P**, *SIGACT News* **7** (1975), 18–20.
18. F. Lemieux, "Finite Groupoids and Their Applications to Computational Complexity," Ph.D. thesis, School of Computer Science, McGill University, Montréal, 1996.
19. A. Maciel and D. Thérien, "Threshold Circuits of Small Majority-Depth," Technical Report SOCS-95.5, School of Computer Science, McGill University, Montréal, 1995.
20. W. D. Maurer and J. L. Rhodes, A property of finite simple non-Abelian groups, *Proc. Amer. Math. Soc.* **16** (1965), 552–554.
21. R. McKenzie, On minimal, locally finite varieties with permuting congruence relations, preprint, 1976.
22. C. Moore, Predicting non-linear cellular automata quickly by decomposing them into linear ones, *Physica D* **111** (1998), 27–41.
23. C. Moore, Quasi-linear cellular automata, *in* "Proceedings of the International Workshop on Lattice Dynamics," Physica D, Vol. 103, pp. 100–132, 1997.
24. C. H. Papadimitriou, "Computational Complexity," Addison–Wesley, Reading, MA, 1994.
25. H. O. Pflugfelder, "Quasigroups and Loops: An Introduction," Heldermann-Verlag, Berlin, 1990.
26. A. A. Razborov, Lower bounds for the size of circuits of bounded depth with basis $\{\&, \oplus\}$, *Math. Notes Acad. Sci. USSR* **41** (1987), 333–338.

27. I. G. Rosenberg, Completeness properties of multiple-valued logic algebras, *in* "Computer Science and Multiple-Valued Logic: Theory and Application" (D. C. Rhine, Ed.), pp. 144–186, North-Holland, Amsterdam, 1977.
28. W. L. Ruzzo, On uniform circuit complexity, *J. Comput. System Sci.* **22** (1981), 365–383.
29. R. Smolensky, Algebraic methods in the theory of lower bounds for Boolean circuit complexity, *in* "Proc. 19th ACM Symposium on the Theory of Computing, 1987," pp. 77–82.
30. M. Suzuki, "Group Theory I," Springer-Verlag, Berlin/New York, 1982.
31. A. Szendrei, "Clones in Universal Algebra," Les Presses de L'Université de Montréal, Montréal, 1986.
32. A. Vesanen, Solvable groups and loops, *J. Algebra* **180** (1996), 862–876.
33. M. Weinstein, "Examples of Groups and loops," Polygonal, Passaic, NJ, 1977.