MINIMA
A Symbolic Approach to Qualitative Algebraic Reasoning

Brian C. Williams *
MIT Artificial Intelligence Lab
545 Technology Square
Cambridge, MA 02139

Abstract

The apparently weak properties of a qualitative algebra have lead some to conclude that we must turn instead to extra-mathematical properties of physical systems. We propose instead that a more powerful qualitative algebra is needed, one that merges the algebras on signs and reals. We have invented a hybrid algebra, called Q1, allows us to select abstractions intermediate between traditional qualitative and quantitative algebras.

The power of our algebra is demonstrated in three ways: First, analysis of Q1 shows that the algebra is robust, sharing many properties of reals, but including several that are unique. Second, these properties enable symbolic manipulation techniques for canonicalization and factorization distinct from those applied to the reals. Finally, these manipulation techniques hold much promise for tasks like design and verification, as suggested by a simple design example.

1 Introduction

Many systems analyze the behavior of physical devices using qualitative equations derived from models and device structure. For tasks like explanation and diagnosis, a device's behavior is predicted using the equations as a network of conduits through which values are propagated. In contrast, the task of design is to construct a network of conduits (i.e, qualitative equations) that produces some desired behavior. Modeling this process requires a theory of composition – how qualitative equations combine to produce the design's aggregate behavior. Creating such a theory requires a careful analysis of the algebraic properties of qualitative equations together with a set of techniques for algebraic manipulation.

We have invented a powerful hybrid algebra called *Q1*, that captures qualitative and quantitative information about physical devices. We have implemented a qualitative symbolic algebra system based on this hybrid algebra called *MINIMA*, that provides facilities for combining, simplifying, canonicalizing and factoring qualitative equations. These facilities are demonstrated on a simple design/verification example and play a central role in the system for novel design described in [8].

Very little has been written about the properties of qualitative algebras, i.e., algebras defined on abstractions of the reals. The few existing studies have focused on the properties of "confluences" – equations involving only addition and subtraction on the signs of quantities[2]. It has been known for some time [1] that confluences are weak – e.g., there is no additive inverse. Recently Struss [7] has performed a detailed analysis of confluences: in replacing signs with a variety of interval representations he came to some disheartening conclusions: "Since the uncovered drawbacks [of confluences] turn out to be very severe, this should motivate a search for additional concepts and approaches of a completely different nature." As a consequence of this weakness few qualitative reasoning systems manipulate qualitative equations symbolically.[1]

Our analysis of a more extensive qualitative algebra *(Q1)* is much more optimistic. Focusing on both the interaction between qualitative and real expressions and a broad set of qualitative operators results in a powerful algebra, one that shares many important properties with the reals, as well as offering additional important properties (section 4). These additional properties allow efficient algebraic manipulation (e.g., canonicalization and factoring) of qualitative expressions without resorting to the expensive procedures needed to manipulate real expressions (section 6). Q1's algebraic properties are also sufficient to account for simple designs (sections 5, 7).

2 Example: Culinary Design

Our overall agenda is to develop a theory of design innovation that accounts for designs using technologies where little is known beyond the physics underlying a few simple devices. A key component of this theory is a qualitative algebra used to describe and compose behavioral relations of primitive devices at an abstract level. The designer uses this algebra to reformulate a desired behavioral relation by combining it with known relations, until he finds one that he knows how to produce (fully or in part) through additions to the physical structure (e.g., adding components or connections). To understand the requirements of this qualitative algebra we consider a simple design example. Since the focus of this paper is the Q1 algebra rather than design, neither the example nor the reasoning strategy presented are particularly sophisticated. [8] presents

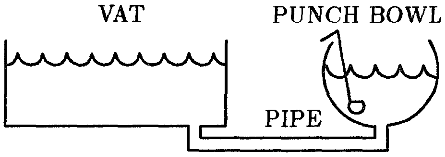
*©1988 Brian C. Williams. This paper describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the author's artificial intelligence research is provided by an Analog Devices Fellowship, the Digital Equipment Corporation, Wang Corporation, and the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-85-K-0124.

[1] The only current exceptions are the qualitative gauss rule [3] and the composition of the M operator[5].

a much more sophisticated design strategy using Q1 that constructs designs of significantly greater complexity.

Suppose you are throwing a major party that includes beverages. Having waiters manually refill the punch bowl from a large vat would intrude on the ambiance of the event. Thus you would like the level of the punch bowl to be restored to the level of the vat automatically. That is, whenever there is a height difference between the bowl and vat, a device should automatically change the bowl height to meet that of the vat.

You reason as follows: First, the height of the punch in the bowl is raised or lowered by having punch flow in or out of it. Second, the pressure at the bottom of a container is proportional to the height and density of the liquid in the container. Since the same type of liquid is in both the vat and bowl, a difference in height corresponds to a difference in pressure. Thus our goal is reformulated as having punch flow into the bowl whenever its pressure drops relative to the vat. Further we know for a pipe that fluid flows to the end that has the lower pressure. Thus our task is completed simply by attaching a pipe between the bottoms of the vat and bowl (and, for aesthetics, hiding the vat behind a tasteful and rare tapestry).



The qualitative vocabulary used above is similar to that found in the literature, involving signs of quantities, differences and their derivatives. The reasoning process, however, is a bit different.

First, the example does not involve reasoning about specific numbers or qualitative values (e.g., positive or increasing). Instead we reasoned about the composition of qualitative relations, using the process of reformulating an initial goal relation with known relations until a goal is found that can be met by augmentations to physical structure (e.g., addition of the pipe).

Second, the example at times requires the designer to reason about the precise relationship between quantities, rather than simply relating the signs of quantities as with confluences[2] (e.g., the exact quantitative relationship between fluid density, height and pressure must be known to relate height and pressure difference).

Capturing this reasoning process requires a hybrid qualitative - quantitative algebra coupled with a theory of the designer's algebraic manipulation skills.

3 The Qualitative Algebra Q1

The qualitative algebra explored here (Q1) is similar to those used elsewhere in the literature (e.g., [9],[2]), but differs in two important respects.

Most importantly, our equations combine qualitative and quantitative information by allowing a combination of qualitative and real operators. Traditionally, real quantities are immediately abstracted to qualitative values (e.g., sign of the quantity), then operated on by the qualitative operators. As a consequence of this early abstraction the result of the qualitative operations is often ambiguous.

In Q1 quantities may first be operated on using the standard real operators, the result abstracted to a qualitative value, and then operated on further using qualitative operators. This produces a result that is less ambiguous than that produced by qualitative operators alone. A "hybrid" algebra of qualitative and quantitative operators thus allows us to express constraints spanning the spectrum from weak constraints expressible by traditional qualitative algebras to quantitative constraints expressible by the standard algebra on the reals.

A second property of Q1 is that qualitative expressions include a full complement of operators on signs analogous to real addition, subtraction, multiplication, division and exponentiation. Although previous work has included many of these operators in qualitative simulation systems (e.g., [9],[4],[5]), algebraic analysis has focused only on the properties of sign addition and subtraction [2],[7], [3].

The remainder of this section defines the domain, operators and syntax of the algebra. Algebraic properties are explored in the next section.

3.1 Domain, Operators and Syntax

Qualitative descriptions operate on two sets, the reals $\Re$ and the set $\mathcal{S} = \{-, 0, +\}$ denoting the sign of real quantities. The relation between $\Re$ and $\mathcal{S}$ is defined by the mapping $[\,]$: $\Re \rightarrow \mathcal{S}$ where:

$$\text{For any } x \in \Re,\ [x] = \begin{cases} + & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ - & \text{if } x < 0 \end{cases}$$

Thus the operator [ ] partitions $\Re$ into three intervals, $(0, +\infty)$, $[0, 0]$ and $(-\infty, 0)$ corresponding to $+, 0$ and $-$.

The set $\mathcal{S}' = \{-, 0, +, ?\}$ extends $\mathcal{S}$ with the value ?, used to represent an undetermined sign (i.e., the value may lie in any one of the three intervals). Thus ? corresponds to the interval $(-\infty, +\infty)$.

Qualitative expressions are composed from the standard operators on $\Re$, $(+, -, \times, /)$, an analogous set of operators on $\mathcal{S}'$ $(\oplus, \ominus, \otimes, \oslash)$ and the operator $[\,]$.

The operator $\oplus : \mathcal{S}' + \mathcal{S}' \rightarrow \mathcal{S}'$ is the qualitative analog of real addition, answering the question: "What is the sign of $x + y$, given only the signs of x and y?". The operators $(\ominus, \otimes, \oslash)$ have similar analogs to their corresponding real operators $(-, \times, /)$. These operators, called sign operators, are defined by the following tables:

⊕	−	0	+	?
−	−	−	?	?
0	−	0	+	?
+	?	+	+	?
?	?	?	?	?

⊖	
−	+
0	0
+	−
?	?

⊗	−	0	+	?
−	+	0	−	?
0	0	0	0	0
+	−	0	+	?
?	?	0	?	?

⊘	−	0	+	?
−	+	U	−	U
0	0	U	0	U
+	−	U	+	U
?	?	U	?	U

(U denotes values for which an operator is undefined, i.e., division by an interval containing zero).

Allowable qualitative expressions (qexp) and real expressions (rexp) are of the form:

qexp ::= [rexp] | sign | qexp sop qexp | $\ominus$ qexp
| $\text{qexp}^{integer}$ | (qexp)
rexp ::= symbol | real | rexp rop rexp | $-$ rexp
| $\text{rexp}^{integer}$ | (rexp)
sop ::= $\oplus$ | $\ominus$ | $\otimes$ | $\oslash$
rop ::= + | − | × | /
sign ::= [+] | [0] | [−] | [?]

where square brackets are placed around the elements of S' to distinguish them from real operators.

Constraints on a system's behavior are described by a combination of qualitative and real equations (qeqn and reqn, respectively):

qeqn ::= qexp = qexp
reqn ::= rexp = rexp

In most qualitative algebras (e.g., [2],[9], [7],[3], and [5]) equality takes on a non-standard meaning - a qualitative equation $q1 = q2$ is satisfied if $q1$ and $q2$ denote the same value *or if either q1 or q2 = [?]*. Thus the expression $[f1] + [f2] = [0]$ is not satisfied by $[f1] = [+]$ and $[f1] = [-]$ since $[+] + [-] = [?]$. This definition results in considerable confusion since qualitative equality is not an equivalence relation - it is not transitive (e.g., $[+] = [?], [?] = [-] \not\Rightarrow [-] = [+]$). As a consequence the fundamental operation of substitution of equals for equals is not generally permitted. This significantly weakens the traditional qualitative algebra.

In Q1, equality has the standard meaning (i.e, qualitative or real expressions that are equated denote the same value). For example, the expression $[f1] + [f2] = [0]$ is satisfied only if $[f1] = [0]$ and $[f2] = [0]$. The equation is not satisfied by $[f1] = [+], [f1] = [-]$ since $[+] + [-] = [?] \neq [0]$.[2]

Q1 can be used to describe the relations in the design scenario. The desired relationship between vat and bowl heights (H_v, H_b) is captured by the hybrid equation:

$$[H_v - H_b] = [d/dt(H_b)]$$

(i.e., the height of the bowl should change in the direction of the height difference between the bowl and vat).

The required relation between height, pressure (P), fluid density (d) and gravity (g) is a real equation, while the relations between height, volume (V) and flow (Q) can be real or qualitative:

$$\begin{aligned} P_i &= d \times g \times H_i \text{ for open container } i \\ Q_b &= d/dt(V_b) \\ [V_c] &= [H_c] \end{aligned}$$

These equations combine to produce the desired relation:

$$[P_v - P_b] = [Q_b]$$

The relation for a pipe with ends e1 and e2 is:

$$[P_{e1} - P_{e2}] = -[Q_{e2}]$$

Thus the desired relation is achieved by connecting a pipe between the bottoms of the vat and bowl.

4 Properties of Q1

In this section we demonstrate the power of Q1 by examining its most important properties (see [8] for a more complete discussion). Q1 is defined as the structure $\langle \Re \cup S', +, \times, \oplus, \otimes, [\ ] \rangle$, where $-$, $/$ are defined in terms of $+, *$; $\ominus$, $\oslash$ are defined in terms of $\otimes$. To understand Q1 we explore the properties of the real and sign algebras ($\langle \Re, +, \times \rangle$ and $\langle S', \oplus, \otimes \rangle$) and then the interactions between them.

In the remainder of the paper s,t and u denote elements of S' and a,b,c denote elements of $\Re$. A table below summarizes the basic properties of $\langle \Re, +, \times \rangle$ (i.e. the field axioms) and the corresponding properties of $\langle S', \oplus, \otimes \rangle$.

	$\langle S', \oplus, \otimes \rangle$	$\langle \Re, +, \times \rangle$
Associative:		
	$(s \oplus t) \oplus u = s \oplus (t \oplus u)$	$(a + b) + c = a + (b + c)$
	$(s \otimes t) \otimes u = s \otimes (t \otimes u)$	$(a \times b) \times c = a \times (b \times c)$
Identity:		
	$s \oplus [0] = s$	$a + 0 = a$
	$s \otimes [+] = s$	$a \times 1 = a$
Inverse:		
	none	$a - a = 0$
	$s \oslash s = [+]$ if $s \neq [0]$ or $[?]$	$a/a = 1$ if $a \neq 0$
Commutativity:		
	$s \oplus t = t \oplus s$	$a + b = b + a$
	$s \otimes t = t \otimes s$	$a \times b = b \times a$
Distributivity:		
	$s \otimes (t \oplus u) = s \otimes t \oplus s \otimes u$	$a \times (b + c) = a \times b + a \times c$

The discussion of these properties are broken into properties of $\langle \Re, +, \times \rangle$ missing in $\langle S', \oplus, \otimes \rangle$, shared properties, and properties of $\langle S', \oplus, \otimes \rangle$ beyond those of $\langle \Re, +, \times \rangle$. We focus on the consequences of these properties most relevant to symbolic manipulation.

4.1 Weaknesses of S'

$\langle S', \oplus, \otimes \rangle$ contains most of the field axioms. The major weakness is the lack of an additive inverse for any element of S' except [0] (i.e., if $v \in \{+, -, ?\}$, there is no $w \in S'$ such that $v \oplus w = [0]$). Thus, whatever $\ominus$ is, it can't be the inverse of $\oplus$. As a result the sign algebra does not meet any of the normal classifications of field, ring, or even group. One major consequence is that there is no cancellation law for $\oplus$:

$$s \oplus u = t \oplus u \quad \not\Rightarrow \quad s = t$$

Without it we cannot in general solve systems of sign equations by subtracting equations and canceling terms. Furthermore, addends cannot be moved between sides of an equation:

$$s \oplus t = u \quad \not\Rightarrow \quad s = u \ominus t$$

Consequently we cannot always solve for a particular variable in a qualitative expression using standard techniques for real expressions. This is why an algebra based only on confluences (i.e., $\langle S', \oplus \rangle$) is so impoverished.

4.2 Commonalities Between S' and $\Re$

In spite of a missing identity, $\langle S', \oplus, \otimes \rangle$ is still quite strong because it shares most of the remaining properties of $\langle \Re, +, \times \rangle$. $\oplus$ has an identity ([0]), is commutative and associative. In addition, $\otimes$ has essentially all the properties of $\times$ (an identity ([+]), inverse operator ($\oslash$), associativity

[2] Q1 provides two additional relations, "contains" and "overlaps", that are important for expressing weaker constraints. [8] discusses their properties, together with techniques for determining if one constraint is an abstraction of another.

and commutativity). Like ×, $\otimes$ has no inverse for $[0]$, but $\otimes$ also has no inverse for $[?]$ (since $[?]$ contains $[0]$). This does not present a problem in practice since a subexpression denoting $[?]$ provides no information.

Since $\otimes$ has an inverse, $\otimes$ has a cancellation rule analogous to × and multiplicands can be moved between sides of an equation:

$$\begin{array}{ll} s \otimes u = t \otimes u \Leftrightarrow s = t & a \times c = b \times c \Leftrightarrow a = b \\ \text{For } t \neq [0], [?]: & \text{For } b \neq 0: \\ s \otimes t = u \Leftrightarrow s = u \oslash t & a \times b = c \Leftrightarrow a = c/b \end{array}$$

The second property above allows us to solve for certain variables or subexpressions of qualitative equations in many situations. Also $\otimes$ distributes over $\oplus$. This, combined with the commutativity and associativity of $\oplus$ and $\otimes$, allows us to represent expressions in a canonical form similar to polynomials on $\Re$.

Earlier we pointed out that $\ominus$ cannot be defined as the inverse of $\oplus$. However $\ominus$ is related to $\otimes$ in a manner similar to − and ×:

$$\begin{array}{rlrl} \ominus u & \equiv [-] \otimes u & -a & \equiv (-1) \times a \\ u \ominus v & \equiv u \oplus (\ominus v) & a - b & \equiv a + (-b) \end{array}$$

As a result $\langle \mathcal{S}', \oplus, \otimes \rangle$ and $\langle \Re, +, \times \rangle$ share the following properties important for simplification:

$$\begin{array}{ll} \ominus(\ominus s) \ = s & -(-a) \ = a \\ \ominus(s \oplus t) \ = \ominus s \ominus t & -(a + b) = -a - b \\ [+] \oslash ([+] \oslash s) = s & 1/(1/a) \ = a, \ \text{if } a \neq 0 \\ \quad \text{if } s \neq [0], [?] & \\ [+] \oslash (s \times t) \ = & 1/(a \times b) = (1/a) \times (1/b) \\ \quad ([+]/s) \otimes ([+] \oslash t) & \quad \text{if } a, b \neq 0 \\ \quad \text{if } s, t \neq [0], [?] & \end{array}$$

4.3 Properties of $\mathcal{S}'$ not in $\Re$

$\langle \mathcal{S}', \oplus, \otimes \rangle$ has three important properties that allow simplifications not possible in $\langle \Re, +, \times \rangle$, and that are fundamental to the canonicalization and factoring algorithms described in section 6.

First, since $[+] \otimes [+] = [+]$ and $[-] \otimes [-] = [+]$, $\otimes$ *is its own multiplicative inverse*:

$$[s] \oslash [t] = [s] \otimes [t] \ \text{ for } \ [t] \neq [0], [?] \qquad (1)$$

A major consequence is that all occurrences of $\oslash$ in an expression can be replaced with $\otimes$ (as long as the denominator doesn't contain $[0]$). In addition:

$$\begin{array}{lll} [s] \otimes [s] & = [+] & \text{for } s \neq [0], [?] \\ s \otimes t = u & \Leftrightarrow s = u \otimes t & \text{for } t \neq [0], [?] \end{array}$$

The second property relates to exponentiation. For a qualitative expression s and integer n, let s^n denote $\underbrace{s \otimes s \cdots \otimes s}_{|n|}$ if n is positive or negative, and $[+]$ if $n = 0$. Then the following holds:

$$s^{2i} \otimes s = s \text{ for } i \in \text{integers} \qquad (2)$$

Thus all expressions raised to a positive/negative odd power are equivalent; likewise for positive/negative even powers. This allows all exponents i to be reduced to $0 \leq i \leq 2$. This is used later in section 6.1 to reduce all sign expressions to quadratics.

Third, there is a cancellation rule for addition:

$$[s] + [s] = [s]$$

As a result of these three properties, common subexpressions are often "absorbed" into a single expression during the simplification process. This results in expressions that are far simpler than their counterpart would be in $\Re$. We return to this issue in section 6.1.

4.4 Relating $\mathcal{S}'$ and $\Re$

The remaining task is to examine expressions that use $[\,]$ to combine properties of $\mathcal{S}'$ and $\Re$ (i.e., $\langle \Re \cup \mathcal{S}', +, \times, \oplus, \otimes, [\,] \rangle$).

$[\,]$ is a homomorphism of $\Re$ onto $\mathcal{S}$ for the operations of multiplication, division, minus and exponentiation:

$$\begin{array}{ll} [a * b] & \Leftrightarrow [a] \otimes [b] \\ [a/b] & \Leftrightarrow [a] \oslash [b] \\ [-a] & \Leftrightarrow \ominus[a] \\ [a^n] & \Leftrightarrow [a]^n \end{array}$$

However, this is not the case for addition or subtraction. For example, expressing height difference as $[H_v] \ominus [H_b]$ is weaker than $[H_v - H_b]$ (e.g., consider $H_v = 8, H_b = 7$).

This sheds light on a crucial problem with the standard approaches to qualitative reasoning – they over-abstract. The mistake is that a qualitative equation is traditionally produced from a real equation by replacing each operator with its sign equivalent and each variable v with $[v]$. Thus, in the punch bowl example we would be forced to represent height difference as $[H_v] \ominus [H_b]$. But this expression is useless – since height is never negative and rarely zero, the value will almost always be $[?]$. We solve this problem by allowing a hybridization of real and sign expressions.

5 Using a Qualitative Algebra for Design

The next step is to incorporate the above properties into a symbolic algebra system adequate to capture the reasoning steps in designs similar to our example. To do this we consider what design entails.

Given a desired behavior, a designer examines the behavioral constraints imposed by the design's existing structure, then uses the models of available components to determine where and what additional augmentations are necessary to meet the desired behavior. A good designer exploits constraints imposed by the existing structure to reduce the additions necessary and identify novel additions. Here we focus only on the algebraic manipulations used in this process; coordinating the overall design process coherently is a subtle task described in [8].

The types of behavior used in the above process can be expressed by equations in our qualitative algebra. The basic algebraic inference performed by the designer is to reformulate an equation describing a desired behavior (*goal*), by combining it with equations describing either existing physical structure or augmentations to that structure he is willing to make (*constraints*). This process is repeated until either a reformulated goal is met by an existing constraint or it is proven unachievable.

Combining a goal with a constraint involves 1) identifying shared variables (or subexpressions), 2) solving for a variable/subexpression in the constraint, 3) substituting the result into the goal, and 4) simplifying the combined result.

In the punch bowl example the original goal $[H_v - H_b] = [d/dt(H_b)]$, and the constraint $H_b * A_b = V_b$ share the variable H_b. Solving for H_b in the constraint we get $H_b = V_b/A_b$. Substituting for H_b in the right hand side of the goal produces $[H_v - H_b] = [d/dt(V_b/A_b)]$; simplifying results in $[H_v - H_b] = [d/dt(V_b)] \oslash [A_b]$. This example is completed in section 7.

Next the reformulated goal is checked for failure or success. Success occurs if the goal is a tautology (e.g., $s \oplus t = s \oplus t$) or equivalent to an existing constraint. Failure occurs if the equation is inconsistent (e.g., $[+] = [-]$).

6 MINIMA

MINIMA is a symbolic algebra system for Q1 that supports the operations identified above. MINIMA is a qualitative analog of the symbolic algebra system Macsyma[6], and in fact uses Macsyma to manipulate subexpressions in $\langle \Re, +, \times \rangle$.

We discuss the two most important operations performed by MINIMA: simplification and equation solving (i.e., solving for a variable or subexpression). By making the simplifier sufficiently powerful (i.e, reducing expressions to a unique canonical form), identifying tautologies and equivalent equations is reduced to determining syntactic equivalence.

Like Macsyma, MINIMA provides two approaches to simplification and equation solving. The first approach is restricted to "obvious" transformations of the equations, using a subset of the properties mentioned above. For example, given $c_1 = [+]$ and $c_2 = 8$, the equation $([a^2] \oplus [c_2]) \otimes [-a/(b-4)*(-c_1)]$ simplifies to $([a] \oplus [+]) \otimes [a]^2 \oslash [b-4]$ by 1) substituting for constants with known values, 2) applying the homomorphisms for $+, \times, -$, 3) evaluating $[\ ]$ on known values, 4) cancelling identities and double negations, and 5) using associativity and commutativity to canonicalize the order of operator arguments. Completeness is traded for faster, more intuitive deductions. The approach is sufficient for many designs, including the punch bowl example.

The second approach performs less obvious transformations, using techniques for qualitative canonicalization and factorization. For example, $([a] \oplus [+]) \otimes [a]^2 \oslash [b-4]$ is further simplified to the multivariate quadratic[3] $[b-4] \otimes [a]^2 \oplus [b-4] \otimes [a]$ for $[b-4] \neq [0], [?]$. Canonicalization and factorization are prohibitively expensive in traditional symbolic algebraic systems on $\Re$. However, this is not the case for a sign algebra – MINIMA exploits the properties of Q1 described in section 4.3 to make canonicalization and factorization very efficient in practice.

6.1 Simplification and Canonicalization

The purpose of simplification is to eliminate irrelevant structure in the equations. This facilitates the process of both comparing and combining equations. The simplifier eliminates structure through a combination of cancellation (e.g., $a/a \Rightarrow 1$), evaluation ($[+] \otimes [-] \Rightarrow [-]$), substitution of known constants, and reduction of subexpressions to a standard form (e.g.,$(b * a) * c \quad \Rightarrow \quad a * b * c$). The operator definitions and properties described in sections 3 and 4 provide the tools to perform simplification.

Simplifying equations in MINIMA involves three steps. First, the real subexpressions of an equation (i.e., expressions contained within $[\ ]$) are simplified using the properties described for $\langle \Re, +, \times \rangle$. Next, real operators are transformed into sign equivalents whenever possible using the homomorphisms of section 4.4. Finally, the surrounding sign expressions are simplified using the properties described for $\langle \mathcal{S}', \oplus, \otimes \rangle$. Mapping from real to sign operators has two advantages: First, the sign of a quantity is often known when the real value isn't. For example, we know density and gravity are positive, independent of substance and planet, thus $[P] = [d*a*g*H]$ simplifies to $[P] = [H]$. Second, the properties of section 4.3 allow significant simplifications in $\mathcal{S}'$ not possible in $\Re$.

The "obvious" simplification approach involves making *local* changes to an equation's structure. Most of the properties of section 4 are applied as simple rewrite rules during simplification. Commutativity and associativity are used together to convert binary expressions into n-ary expressions whose arguments are sorted lexicographically (as in $(b * a) * c \Rightarrow a * b * c$). The main property *not* used is distributivity since expanding expressions using distribution can radically change an equation's structure.

The second simplification approach reduces an expression or equation to a pseudo-canonical form[4] analogous to a multivariate polynomial. In traditional symbolic systems real expressions can be reduced to a unique rational form – a fraction consisting of two multivariate polynomials with common factors removed.

Although constructing polynomials is fast, constructing rationals is expensive for large expressions. The cost is in factorization, which relies heavily on computing greatest common denominator (GCD).

A similar approach is taken for Q1, but one that is significantly faster in practice. Operators in $\mathcal{S}'$ are distributive, commutative, and associative. This is sufficient to construct polynomials from sign expressions. Furthermore, by equation 4.3 all exponents can be reduced to degree 1 or 2, thus the polynomials are at most quadratic. Finally, by equation 4.3 division can be replaced by multiplication, thus all expressions in $\langle \mathcal{S}', \oplus, \otimes \rangle$ can be represented as quadratic, multivariate polynomials.[5] Since $\oslash$ is eliminated, factorization and GCD is unnecessary to perform canonicalization.

To canonicalize a hybrid equation we convert the real subexpressions to rationals, apply the homomorphisms, and then canonicalize the sign expressions as above. [6]

[3] A multivariate polynomial is a polynomial in variable v whose coefficients are polynomials not in v. A multivariate quadratic is a multivariate polynomial of degree 2.

[4] We use the modifier "pseudo" only because the canonical form has not yet been proven unique.

[5] An additional stage involves eliminating irrelevant terms in the quadratic equation (e.g.,$[x]^2 \oplus [x] \oplus [+] \quad \Leftrightarrow \quad [x] \oplus [+]$). See [8] for a complete list of elimination rules.

[6] A more general approach involves constructing the prime factorization of the real expression before applying the homomorphisms.

6.2 Equation Solving and Factoring

Given an equation in $\langle \Re, +, \times \rangle$ it is possible to solve for any variable. This is not the case for $\langle S', \oplus, \otimes \rangle$: Since there is no cancellation for addition, addends cannot be moved between the left and right sides of an equation.

However, it is often possible to solve for certain subexpressions. Cancellation can be performed for multiplication (section 4.2), thus we can solve for any subexpression that is an argument of a top-level multiplication For example, solving for $s \oplus t$:

$$(s \oplus t) \otimes u = v \quad \Rightarrow \quad (s \oplus t) = v \oslash u \quad \text{for } u \neq [0], [?]$$

More generally, we can compute the prime factors of the top-level expressions in a qualitative equation, and then solve for any of the factors. Traditionally factorization requires computing GCD, which is very expensive. However, factorization using standard GCD algorithms cannot be used for sign expressions since GCD algorithms rely on cancellation. Instead we use a much simpler approach. Since sign expressions can be reduced to quadratics, it is relatively inexpensive to determine the factors by generate and test.

The factorization of a quadratic is of the form:

$$s_2 \otimes [x]^2 \oplus s_1 \otimes [x] \oplus s_0 = (a \otimes [x] \oplus c) \otimes (b \otimes [x] \oplus d)$$

where s_2, s_1, s_0, a, b, and c are qualitative expressions and $s_2 = a \otimes b$, $s_1 = (a \otimes c) \oplus (b \otimes d)$ and $s_0 = c \otimes d$. Thus to factor a quadratic we generate a,b,c and d by factoring the coefficients s_2 and s_0, and then distribute s_2's factors between a and b, and s_0's factors between c and d. To test we compute the polynomial corresponding to $(a \otimes c) \oplus (b \otimes d)$ and compare it with s_1. Quadratics are sufficiently infrequent that this strategy is quite acceptable in practice.

7 Example Revisited

MINIMA's facilities for simplification, substitution and equation solving provide the algebraic tools necessary to walk through the punch bowl example which, as the trace below suggests, is more complex than our intuitions might at first suggest. The following is a simplified trace of the deductions going from the initial goal $[H_v - H_b] = [dH_b/dt]$ to the reformulation $[P_v - P_b] = [Q_b]$, which is the key to recognizing the solution involving a pipe. In the example, Gn, Fn, and Cn denote Goals, given Facts and Consequences, respectively.

G1)	$[H_v - H_b] = [d/dt(H_b)]$	Original Design Goal
F2)	$H_b * A_b = V_b$	Container Model
C3)	$H_b = V_b/A_b$	Solve for H_b in F2
G4)	$[H_v - H_b] = [d/dt(V_b/A_b)]$	Substitute for H_b in G1 using C3
G7)	$[H_v - H_b] = [d/dt(V_b)/A_b]$	Differentiate G4
G8)	$[H_v - H_b] = [d/dt(V_b)] \oslash [A_b]$	Simplify G7
F9)	$[A_b] = [+]$	Container Model
G10)	$[H_v - H_b] = [d/dt(V_b)]$	Substitute for A_b in G8 using F9
F11)	$Q_b = d/dt(V_b)$	Container Model
G12)	$[H_v - H_b] = [Q_b]$	Substitute for $d/dt(V_b)$ in G10 using F11
F13)	$P_v = d * g * H_v$	Container Model
C14)	$H_v = P_v/m * g$	Solve for H_v in F13
G15)	$[P_v/d * g - H_b] = [Q_b]$	Substitute for H_v in G12 using C14
F16)	$P_b = d * g * H_b$	Container Model
C17)	$H_b = P_b/d * g$	Solve for H_b in F16
G18)	$[P_v/d * g - P_b/d * g] = [Q_b]$	Substitute for H_b in G15 using C17
G19)	$[(P_v - P_b)]/([d] \otimes [g]) = [Q_b]$	Simplify G18
F20)	$[d] = [+]$	Property of fluids
F21)	$[g] = [+]$	Property of gravity
G22)	$[(P_v - P_b)]/([+] \otimes [+]) = [Q_b]$	Substitute g,d into G19 using F20,F21
G23)	$[P_v - P_b] = [Q_b]$	Simplify G22

The reformulated goal G23 enables the crucial insight – the close similarity between G23 and a pipe's behavior (i.e., $[P_{t1} - P_{t2}] = [Q_{t2}]$ where t1,t2 are the ends of a pipe) suggests connecting a pipe between the two containers.

8 Discussion

The apparently weak properties of a qualitative algebra have lead some to conclude that we must turn instead to extra-mathematical properties of physical systems. We have instead proposed a new qualitative algebra, Q1, that merges the algebras on signs and reals, allowing us to select abstractions intermediate between traditional qualitative and quantitative algebras.

The power of our algebra is demonstrated in three ways: First, Q1 is a robust algebra sharing many properties of reals, but several that are unique. Second, these properties enable symbolic manipulation techniques for canonicalization and factorization, distinct from those applied to the reals. Finally, these manipulation techniques hold much promise for tasks like design and verification, as suggested by our example.

The qualitative symbolic algebra system MINIMA has been fully implemented and tested on a Symbolics 3600. A design system based on MINIMA is partially implemented.

Acknowledgments: I would like to thank Leah Williams, Dan Weld, Jeff Van Baalen, Paul Resnick, Walter Hamscher, Johan de Kleer and Randy Davis.

References

[1] J. de Kleer. *Causal and Teleological Reasoning in Circuit Recognition.* AI-TR-529, MIT AI Lab, Sep. 1979.

[2] J. de Kleer and J. Brown. A Qualitative Physics Based on Confluences. *Artificial Intelligence*, 24, Dec. 1984.

[3] J. Dormoy and O. Raiman. Assembling a Device. *AI in Engineering*, To Appear 1988.

[4] K. Forbus. Qualitative Process Theory. *Artificial Intelligence*, 24, Dec. 1984.

[5] B. Kuipers. Commonsense Reasoning about Causality. *Artificial Intelligence*, Dec. 1984.

[6] J. Moses. Algebraic simplification: a guide for the perplexed. *Comm. ACM*, 14(8), Aug. 1971.

[7] P. Struss. Mathematical Aspects of Qualitative Reasoning. *AI in Engineering*, To Appear 1988.

[8] B. Williams. *Principled Design Based on Topologies of Interaction.* PhD Thesis, MIT AI Lab, To Appear 1988.

[9] B. Williams. Qualitative Analysis of MOS Circuits. *Artificial Intelligence*, Dec. 1984.